ON Semiconductor

Is Now

To learn more about onsemi™, please visit our website at
www.onsemi.com

onsemi and onsemi and other names, marks, and brands are registered and/or common law trademarks of Semiconductor Components Industries, LLC dba "**onsemi**" or its affiliates and/or subsidiaries in the United States and/or other countries. **onsemi** owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of **onsemi** product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent-Marking.pdf. **onsemi** reserves the right to make changes at any time to any products or information herein, without notice. The information herein is provided "as-is" and **onsemi** makes no warranty, representation or guarantee regarding the accuracy of the information, product features, availability, functionality, or suitability of its products for any particular purpose, nor does **onsemi** assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using **onsemi** products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by **onsemi**. "Typical" parameters which may be provided in **onsemi** data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. **onsemi** does not convey any license under any of its intellectual property rights nor the rights of others. **onsemi** products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use **onsemi** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **onsemi** and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that **onsemi** was negligent regarding the design or manufacture of the part. **onsemi** is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner. Other names and brands may be claimed as the property of others.

AND9625/D

System Functions User's Manual for LC823450 Software Development

ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

ARM®

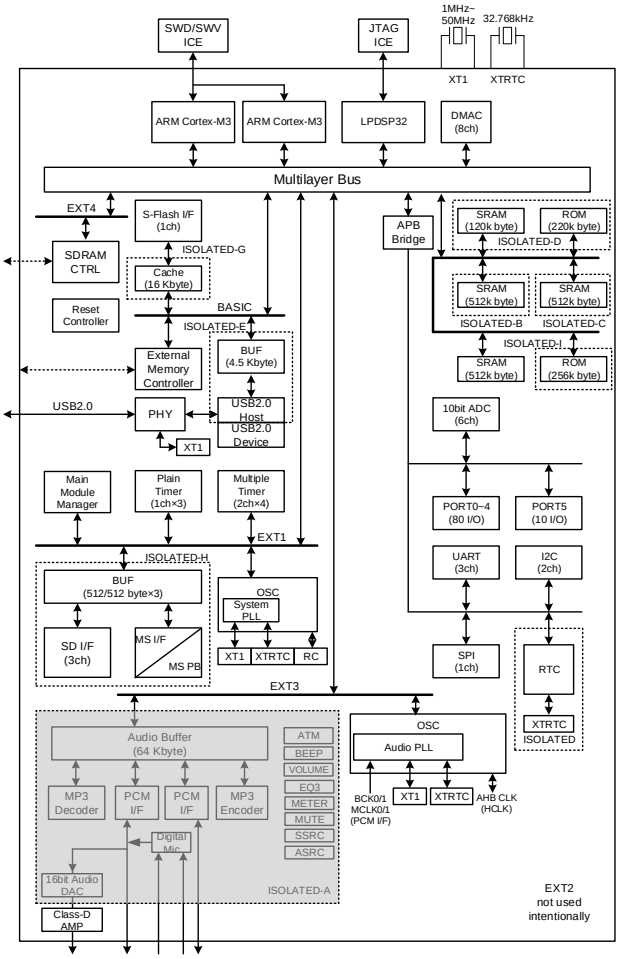
Overview

This application note describes the system functions of LC823450 Series (called LC823450 hereafter) for software development using this device. Each functions as follow blocks are described in this manual.

This user manual is designed to provide a good understanding of the system functions of LC823450 as a reference document.

Intended audience is customers who are building audio application using LC823450.

OSC
SystemPLL
AudioPLL
INTC
SystemController
Multi-core
CORE_REG
MUTEX_REG
DSP_CMDIF
PORT
PTM
MTM
RTC
SPI
UART
I2C
S-Flash
ADC
ROMC
XMC
SDRAMC
USB20DC
USB20HC



System block on LC823450

© Semiconductor Components Industries, LLC, 2017

1 OSC 12

1-1 Introduction 12

1-1-1 The Outline of Oscillation Controller 13

1-2 Functional Overview 15

1-2-1 Working State 16

1-2-2 Reset State 17

1-2-3 Run state 18

1-2-4 Sleep state 19

1-2-5 WIC sleep state 20

1-3 Programmer's Model 22

1-3-1 About the programmer's model 23

1-3-2 Summary of ABUF registers 23

1-3-3 Register Descriptions 24

1-4 Operation 35

1-4-1 System clock generation circuit 35

1-4-2 Function clock generation circuit 37

1-4-3 Audio clock generation circuit 38

1-4-4 WIC sleep condition 39

1-4-5 Hardware interrupt 40

2 SystemPLL 42

2-1 Introduction 42

2-1-1 The Outline of SysPLL 43

2-2 Functional Overview 44

2-2-1 SysPLL Circuit Configuration 45

2-3 Programmer's Model 46

2-3-1 About the Programmer's Model 47

2-3-2 Summary of SysPLL Registers 47

2-3-3 Register Descriptions 48

2-4 Operation 51

2-4-1 Output Frequency Setting 51

2-4-2 Precautions for use 51

3 AudioPLL 52

3-1 Introduction 52

3-1-1 The outline of AudPLL 53

3-2 Functional Overview 54

3-2-1 AudPLL circuit configuration 55

3-3 Programmer's Model 56
3-3-1 About the Programmer's Model 57
3-3-2 Summary of AudPLL Registers 57
3-3-3 Register Descriptions 58
3-4 Operation 61
3-4-1 Output Frequency Setting 61
3-4-2 Precautions for use 61

4 INTC 62

4-1 Introduction 62
4-1-1 Brief Summary of INTC 63
4-2 Functional Overview 66
4-3 Programmer's Model 67
4-3-1 About the programmer's model 68
4-3-2 Summary of Interrupt Controller registers 68
4-3-3 Register Descriptions 72
4-4 Operation 121

5 SystemController 122

5-1 Introduction 122
5-1-1 The outline of System Controller 123
5-2 Functional Overview 125
5-2-1 System Controller circuit configuration 126
5-3 Programmer's Model 127
5-3-1 About the Programmer's model 128
5-3-2 Summary of System Controller Registers 128
5-3-3 Register Descriptions 130
5-4 Operation 196
5-4-1 Reset Functions 196
5-4-2 Clock control functions 203
5-4-3 The setting flow of software reset control and clock ON/OFF control 208
5-4-4 The setting flow of power supply control and isoration control for internal function blocks 209
5-4-5 Operating Precautions 210

6 Multi-Core 211

6-1 Introduction 211
6-1-1 The outline of Multi-Core 212
6-2 Functional Overview 213
6-2-1 Multi-Core Circuit Configuration 214

6-2-2 Cortex-M3 Core0/Cortex-M3 Core1 function 216
6-2-3 LPDSP function 220
6-3 Operation 221
6-3-1 Reset function 221
6-3-2 Clock Function 228
6-3-3 LPDSP Program download and Start sequence 233

7 CORE_REG 234

7-1 Introduction 234
7-1-1 The Outline of CORE_REG 235
7-2 Functional Overview 236
7-2-1 CORE_REG Circuit Configuration 237
7-3 Programmer's Model 238
7-3-1 About the Programmer's model 239
7-3-2 Summary of CORE_REG registers 239
7-3-3 Register Descriptions 240
7-4 Operation 242

8 MUTEX_REG 243

8-1 Introduction 243
8-1-1 Brief Summary of MUTEX_REG 244
8-2 Functional Overview 245
8-2-1 MUTEX_REG Circuit Configuration 246
8-3 Programmer's Model 247
8-3-1 About the programmer's model 248
8-3-2 Summary of MUTEX_REG registers 248
8-3-3 Register Descriptions 249
8-4 Operation 250
8-4-1 Common resource access lock 250
8-4-2 Common resource access unlock 250

9 DSP_CMDIF 251

9-1 Introduction 251
9-1-1 The Outline of DSP Command IF 252
9-2 Functional Overview 253
9-2-1 DSP Command IF Circuit Configuration 254
9-3 Programmer's Model 255
9-3-1 About the Programmer's Model 256
9-3-2 Summary of DSP Command IF Registers 256

9-3-3 Register Descriptions 258
9-4 Operation 272
9-4-1 Ex: Command notification from HOST0 to DSP and the response 272
9-4-2 Ex: Command notification from HOST1 to DSP and the response 272

10 PORT 273

10-1 Introduction 273
10-1-1 The Outline of PORTs 274
10-2 Functional Overview 275
10-2-1 PORTs Circuit Configuration 276
10-3 Programmer's Model 277
10-3-1 About the Programmer's model 278
10-3-2 Summary of PORTs registers 278
10-3-3 Register Descriptions 279
10-4 Operation 284
10-4-1 PORTn Pin Assignment 284
10-4-2 Precautions for use 284

11 PTM 285

11-1 Introduction 285
11-1-1 Brief Summary of PTMs 286
11-2 Functional Overview 288
11-2-1 BTMn Circuit Configuration 289
11-2-2 WTMn Circuit Configuration 290
11-3 Programmer's Model 291
11-3-1 About the programmer's model 292
11-3-2 Summary of PTMs registers 292
11-3-3 Register Descriptions 293
11-4 Operation 301
11-4-1 Function of basic timer 301
11-4-2 Operation of basic timer 302
11-4-3 Function of watchdog timer 303
11-4-4 Operation of watchdog timer 305
11-4-5 Directions 307

12 MTM 308

12-1 Introduction 308
12-1-1 Brief Summary of MTMs 309
12-2 Functional Overview 311

12-2-1 MTMs Circuit Configuration 312
12-3 Programmer's Model 313
12-3-1 About the programmer's model 314
12-3-2 Summary of MTMs registers 314
12-3-3 Register Descriptions 315
12-4 Operation 348
12-4-1 Counter source clock control function 348
12-4-2 Counter synchronous function 350
12-4-3 Standard timer operation 352
12-4-4 Pulse length reader function 356
12-4-5 Sentinel inform function 360
12-4-6 PWM output function 365
12-4-7 Multiple timers interrupt 370
12-4-8 Multiple timers standby function 371
12-4-9 Precautions for use 371

13 RTC 372

13-1 Introduction 372
13-1-1 The outline of RTC 373
13-2 Functional Overview 374
13-2-1 RTC circuit configuration 375
13-2-2 RTC Sequence 376
13-3 Programmer's Model 378
13-3-1 About the programmer's model 379
13-3-2 Summary of RTC registers 380
13-3-3 Register Descriptions 381
13-4 Operation 419
13-4-1 Explanation about operations 419
13-4-2 Control flow 425
13-4-3 Precautions for use 426

14 SPI 427

14-1 Introduction 427
14-1-1 Brief Summary of SPI 428
14-2 Functional Overview 430
14-2-1 SPI circuit configuration 431
14-3 Programmer's Model 432
14-3-1 About the programmer's model 433

14-3-2 Summary of SPI registers 433
14-3-3 Register Descriptions 434
14-4 Operation 461
14-4-1 Baud rate generator 461
14-4-2 DMA transfer 461
14-4-3 Chip selection setup/holding time 462
14-4-4 Polarity of SCLK 463
14-4-5 Timing chart 464
14-4-6 Directions 466
15 UART 467
15-1 Introduction 467
15-1-1 Brief Summary of UARTn 468
15-2 Functional Overview 470
15-2-1 UARTn Circuit Configuration 471
15-3 Programmer's Model 472
15-3-1 About the programmer's model 473
15-3-2 Summary of UART registers 473
15-3-3 Register Descriptions 474
15-4 Operation 500
15-4-1 Start-stop synchronous operation mode 500
15-4-2 Transmission/Reception Timing 504
15-4-3 Continuous Transmission Timing 507
15-4-4 Break Transmission and Reception Detection 509
15-4-5 Reception Register Overwrite Error Detection Function 511
15-4-6 Parity Error Detection Function 512
15-4-7 Framing Error Detection Function 514
15-4-8 Transmission Timing at Active CTS Function 516
15-4-9 Reception timing at Active RTS Function 524
15-4-10 Asynchronous Serial Transmission Reception 527
16 I2C 532
16-1 Introduction 532
16-1-1 The Outline of I2C 533
16-2 Functional Overview 534
16-2-1 I2C Circuit Configuration 535
16-3 Programmer's Model 536
16-3-1 About the Programmer's model 537

16-3-2 Summary of I2C Registers 537
16-3-3 Register Descriptions 538
16-4 Operation 544
16-4-1 SCL and SDA Timing 544
16-4-2 Operating Sequence 545
16-4-3 Control Flow 546
16-4-4 Precautions for use 549
17 S-Flash 550
17-1 Introduction 550
17-1-1 Brief Summary of S-Flash 551
17-2 Functional Overview 552
17-2-1 S-Flash Circuit Configuration 553
17-3 Programmer's Model 554
17-3-1 About the programmer's model 555
17-3-2 Summary of S-Flash registers 555
17-3-3 Register Descriptions 556
17-4 Operation 578
17-4-1 Read transfer procedure 578
17-4-2 Write transfer procedure 579
18 ADC 580
18-1 Introduction 580
18-1-1 Brief Summary of ADC Block 581
18-1-2 ADC Features 582
18-2 Functional Overview 583
18-2-1 ADC Circuit Configuration 584
18-3 Programmer's Model 585
18-3-1 About the programmer's model 586
18-3-2 Summary of ADC registers 586
18-3-3 Register Descriptions 587
18-4 Operation 599
18-4-1 ADC Function Description 599
18-4-2 AD Converter Operation 605
18-4-3 Operating Precautions 615
19 XMC 616
19-1 Introduction 616
19-1-1 Brief Summary of XMC 617

19-2 Functional Overview 619
19-2-1 XMC Circuit Configuration 620
19-3 Programmer's Model 621
19-3-1 About the programmer's model 622
19-3-2 Summary of XMC registers 622
19-3-3 Register Descriptions 623
19-4 Operation 637
19-4-1 Function explanation of XMC 637
19-4-2 XMC's operation 643

20 SDRAMC 664

20-1 Introduction 664
20-1-1 Brief Summary of SDRAMC 665
20-1-2 Feature of SDRAMC 665
20-2 Functional Overview 666
20-2-1 SDRAMC circuit configuration 667
20-3 Programmer's Model 668
20-3-1 About the programmer's model 669
20-3-2 Summary of SDRAMC registers 669
20-3-3 Register Descriptions 670
20-4 Operation 683
20-4-1 About the initialization operation 683
20-4-2 Setting method of mode register 684
20-4-3 Setting method of extended mode register (SDRTYPE.MSDRAM register is "1"). 684
20-4-4 Control method of auto refresh 685
20-4-5 Setting method of SDRAM standby mode 686
20-4-6 About the setting of waiting time between SDRAM commands 687
20-4-7 About write and read access to SDRAM 688

21 USB20DC 689

21-1 Introduction 689
21-1-1 The outline of USB20DC 690
21-2 Functional Overview 707
21-2-1 USB20DC circuit configuration 708
21-2-2 USB20DC Function 709
21-3 Programmer's Model 719
21-3-1 About the programmer's model 720
21-3-2 Summary of USB20DC Registers 720

21-3-3 Register Descriptions 725
21-4 Operation 789
21-4-1 Precautions for use 789
22 USB20HC 855
22-1 Introduction 855
22-1-1 The outline of USB20HC 856
22-2 Functional Overview 869
22-2-1 USB20HC circuit configuration 870
22-3 Programmer's Model 871
22-3-1 About the programmer's model 872
22-3-2 Summary of USB20HC registers 872
22-3-3 Register Descriptions 875
22-4 Operation 942
22-4-1 Precautions for use 942
Appendix 943

Preface

This user manual is designed to provide a good understanding of the hardware and software functions of LC823450.

This can also be used as a reference document for hardware and software development of the system using these devices.

Introductory explanation

Data notation	:	High-order digit is on the left, low-order digit is on the right	
Note	:	Explanation of the note in the context	
Attention	:	Please be sure to read this comment with care	
Remarks	:	Remarks of the context	
Numerical notation	:	Binary digit	…XXXX b
	:	Decimal digit	…XXXX
	:	Hexadecimal digit	…0xXXXX
Data type	Word		…32 Bit
	Half word		…16 Bit
	Byte		…8 Bit

1 OSC

1-1 Introduction

This chapter provides the outline of Oscillation controller in the section below.

- The Outline of Oscillation Controller

1-1-1 The Outline of Oscillation Controller

This section provides the outline of Oscillation controller.

This LSI has the registers below for clock control.

- OSC control

OSC control register

- PLL reference control

PLL reference setting register

- System clock dividing control

MAIN clock dividing register

Function clock dividing register

Peripheral clock dividing register

- Interrupt mode control

Interrupt mode control register

- Standby control

Software standby control register

- Audio PLL (PLL2) clock dividing control

Audio PLL clock dividing register

- Audio clock control

Audio clock setting register

1-1-1-1 Oscillation Control System

This LSI has three oscillators: RC OSC (1 MHz), XTRTC OSC (32 kHz) and XT1 OSC (1 to 50 MHz) and two PLLs: PLL1 for system clock and PLL2 for audio clock. Controlling these with registers can generate clocks for the system and the audio blocks.

- RC OSC

Built-in oscillation circuit (with resister and capacitor) generates a clock usable in the system.

- XTRTC OSC

LC823450, when connected to an external oscillator or supplied with an externally-generated clock, generates a clock usable in the system.

- XT1 OSC

LC823450, when connected to an external oscillator or supplied with an externally-generated clock, generates a clock used for the system.

In addition, LC823450 has a PLL controller to control system clocks in a versatile way.

1-2 Functional Overview

This chapter provides the outline of ABUF function block in the sections below.

- Working State
- Reset State
- Run state
- Sleep state
- WIC sleep state

1-2-1 Working State

This LSI manages 4 working states listed below, which are closely related to the oscillation control systems.

- Reset state
- Run state
- Sleep state
- WIC sleep state

The working state transition diagram is as shown below.

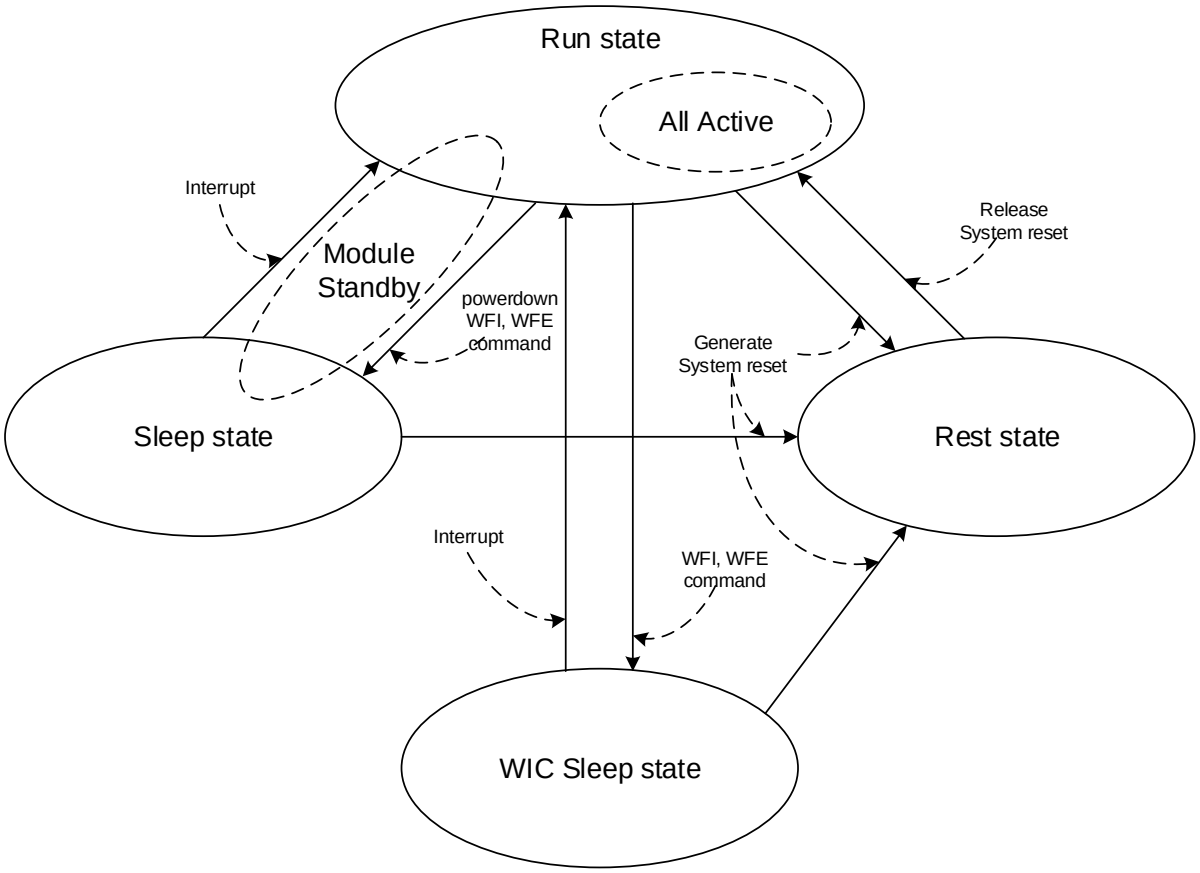


Figure 1-1 Working state transition diagram

1-2-2 Reset State

This state is when this LSI is being system-reset.

1-2-2-1 Transition to the Reset state

Any one of the following items can be the factor for the transition to the reset state.

- A reset signal supplied from the outside
- A system reset generated by the watchdog timer

When either one of these factors occur, regardless of which working state it is in, it will transition to the reset state and initialize this LSI.

1-2-2-2 Reset state cancellation

Factors to release the reset state and the working state that is transitioned to after the release is as below.

- System reset release

In this case, it transitions to the run state.

1-2-3 Run state

This is a state which shows this LSI is active. In this state, commands can be executed and power consumption is the highest of the 4 working states.

In other states, Cortex-M3 and LPDSP will stop and commands cannot be executed.

1-2-3-1 Transition to the Run state

Any one of the following items can be the factor of the transition to the run state.

- System reset release during the reset state
- Hardware interrupt during the sleep state
- Hardware interrupt during WIC sleep state

The system will transit to the run state when any one of the above factors occurs. In the run state, each module can be separately put on standby.

1-2-3-2 Run state cancellation

Factor to cancel the run state and the working state that is transitioned to after cancellation are as below.

- System reset

In this case, it transitions to the reset state.

- When bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) is set to “0” and WFI command or WFE command is executed by program (for Cortex-M3).
- When power down command is executed by program (for LPDSP).

In these cases, it transitions to the sleep state. Each module set on standby remains on standby.

- When bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) is set to “1” and WFI command or WFE command is executed by program.

In this case, it transitions to WIC sleep state. Each module set on standby remains on standby. LPDSP cannot transition to WIC sleep state.

1-2-4 Sleep state

This is a state which only Cortex-M3 or LPDSP will stop. Transition to this state is made by the program.

1-2-4-1 Transition to the sleep state

Any one of the following items can be the factor for the transition to the sleep state.

- When, in the run state, bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) is set to “0” and WFI command or WFE command is executed (for Cortex-M3).
- When, in the run state, power down command is executed (for LPDSP).

It will transit to the sleep state with either one of the factor’s occurrence. In the sleep state, only functions that will stop are Cortex-M3 or LPDSP so other functions will continue working. Each core can separately transition to the sleep state.

1-2-4-2 Sleep state cancellation

Factors to cancel the sleep state and the working state that is transitioned to after cancellation are as below.

- System reset

In this case, it transitions to the reset state.

- Hardware interrupt

In this case, it transitions to the run state. Each module set on standby remains on standby.

1-2-5 WIC sleep state

The entire LSI stops in this state because the oscillation system stops. However, as an exception, XTRTC OSC (32 kHz) and RTC will continue working. Note that PLL1, PLL2 and 10 bit ADC are put on standby, but USB PHY and D-AMP are not put on standby. Transition to this state is made by the program.

1-2-5-1 Transition to WIC sleep state

Transition factor to WIC sleep state is as below.

- When, in the run state, bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) is set to “1” and WFI command or WFE command is executed by the program.

Having this factor will make a transition to WIC sleep state. Note that the audio block is not put on standby. LPDSP cannot transition to WIC sleep state.

*Attention 1:

SLEEPDEEP signal from Cortex-M3 is used for transition to WIC sleep state. But when it is connected to a debugger, Cortex-M3 cancels the SLEEPDEEP signal to prevent transition to WIC sleep state. When a debugger is connected, software standby register (SSTBY) can be used to virtually debug WIC sleep state.

*Attention 2:

When WFI command or WFE command and an interrupt occur at the same time, Cortex-M3 does not generate SLEEPDEEP signal so it cannot transition to WIC sleep state.

1-2-5-2 WIC sleep state cancellation

Factors to cancel WIC sleep state and the working state that is transitioned to after cancellation are as below.

- System reset

In this case, it transitions to the reset state.

- Hardware interrupt

In this case, it transitions to the run state. Each module set on standby remains on standby.

1-2-5-3 WIC sleep state sequence

The timing of transition to and return from WIC sleep state is as shown in Figure 1-2.

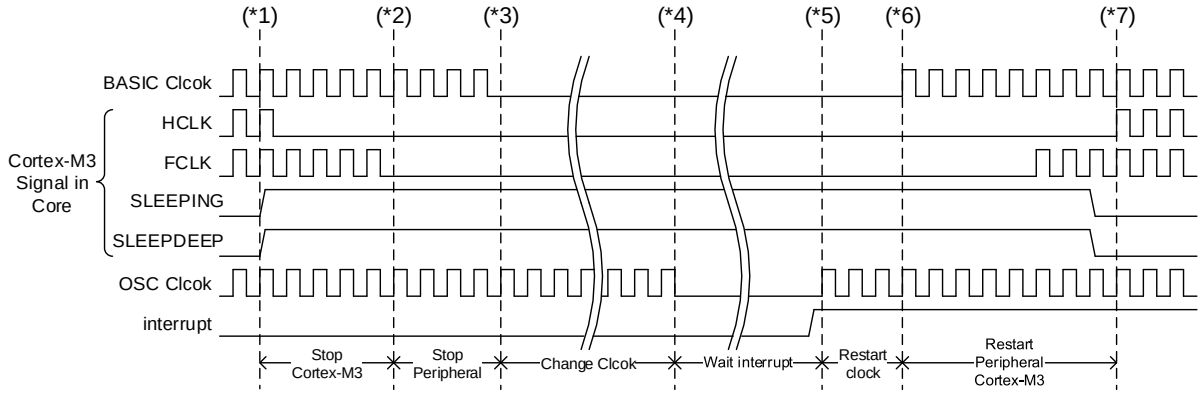


Figure 1-2 WIC sleep state sequence

When bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) is set to “1” and WFI command or WFE command is executed, SLEEPING and SLEEPDEEP signals of Cortex-M3 become active at the timing of (*1) in the figure above. At this time, Cortex-M3 stops HCLK and FCLK to be supplied to Cortex-M3 Core as shown from (*1) to (*2) in the figure above.

After Cortex-M3 completely stops, Oscillation controller stops Peripheral clock as shown from (*2) to (*3) in the figure. After (*3), very few circuits in Oscillation controller are supplied with clock.

The period from (*3) to (*4) is when clock change sequence is being executed. In this period, the clock source is changed to RC OSC as BASIC Clock source.

All the clock stop sequence finishes at the timing of (*4). After that, all the clocks except RTC remain stopped until a hardware interrupt is input. At this timing, PLL1, PLL2 and 10 bit ADC are forcibly put on standby, RC OSC stops and XT1 OSC, if enabled to oscillate, is forcibly stopped from oscillating.

If a hardware interrupt is input at the timing of (*5) in the figure, Oscillation controller detects it to start RC OSC oscillation, cancel the forced standby of PLL1, PLL2 and 10 bit ADC and restart XT1 OSC oscillation if enabled to oscillate. During the period form (*5) to (*6), the system-clock-supply-restart sequence is being executed.

At a timing after (*6) in the figure above, Oscillation controller starts to supply clock to the whole system. At a timing after (*6), LPDSP can work. At a timing after (*7), Cortex-M3 can work because WIC restarts HCLK and FCLK to be supplied to Core between (*6) and (*7).

1-3 Programmer's Model

This chapter provides Oscillation controller registers and setting via microcontroller in the sections below.

- About the programmer's model
- Summary of ABUF registers
- Register Descriptions

1-3-1 About the programmer’s model

Oscillation controller base address is not fixed and varies by the system implementation.

See the system specification for more information about the base address.

Each register's offset is fixed.

1-3-2 Summary of ABUF registers

Oscillation controller is controlled by the registers in “Table 1-1”.

Remark: Do not access any unspecified address from Base+0x0000 to Base+0xFFFF.

Table 1-1 Oscillation Controller registers

Offset	Abbrev.	Attribute	Register name	Initial value
0x000	OSCCNT	R/W	OSC control register	0x0000_0040
0x004	PLLREFCNT	R/W	PLL reference control register	0x0000_0000
0x008	PERICLKDIV	R/W	Peripheral clock frequency dividing register	0x0000_0000
0x00C	FCLKCNT	R/W	Function clock control register	0x0000_0000
0x014	AUDCLKCNT	R/W	Audio clock setting register	0x0008_0002
0x020	IMCNT	R/W	Interrupt mode control register	0x0000_0000
0x030	SSTBY	R/W	Software standby register	0x0000_0000
0x040	CORESTS	R/W	Core status register	0x0000_0000

1-3-3 Register Descriptions

1-3-3-1 OSC control register: OSCCNT

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	XT1ACT	-	-	SPLLDIV	-	MAINDIV2	MAINDIV1	MAINDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	XT1EN	RCOSC EN	XT1DIV SYS	-	MC FORCE	MCSEL	SCKSEL1	SCKSEL0
Reset	0	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit15: XT1ACT
Indicates whether XT1 is oscillating or not.
0: XT1 is not oscillating.
1: XT1 is oscillating.

Attention: This bit does not ensure that XT1 is stably oscillating. Normally, it is used to check whether a crystal oscillator is connected or not.

Bit12: SPLLDIV
Controls the division immediately after System PLL output.
0: Divided by 2 immediately after System PLL output.
1: Divided by 4 immediately after System PLL output.

Attention: No change shall be made when System PLL output clock is used as the system clock source.

Bit10-8: MAINDIV2-0
Selects the dividing ratio of MAIN clock divider.
000 b: Not divided.
001 b: Divided by 2.
010 b: Divided by 4.
011 b: Divided by 8.
100 b: Divided by 16.
Others: Setting prohibited.

Remark: Change can be made even if MAIN clock is used as the system clock.

Attention: After setting, 4 cycles are needed for actual clock change.

Bit7: XT1EN
Controls XT1 oscillation. When XT1 is not used, it can be stopped. When XT1IEN flag of the interrupt mode control register (IMCNT) is “1” and a hardware interrupt occurs, it automatically changes into “1”.
0: Stops.
1: Oscillates.

Bit6: RCOSCEN
Controls RC OSC oscillation. When RC OSC is not used, it can be stopped. When RCCHGEN flag of the interrupt mode control register (IMCNT) is “1”, a hardware interrupt occurs and XTRTC (32.768 kHz) clock is selected as the system clock, or when it returns from WIC sleep mode, it automatically changes into “1”.
0: Stops.
1: Oscillates.

Bit5: XT1DIVSYS
Controls the WT1 division (Whether to divide it by 4 or not).
0: Uses XT1 clock.
1: Uses XT1 clock divided by 4.

Attention: No change shall be made when XT1 clock or XT1 divided-by-4 clock is used as the system clock source, or as the reference clock with System PLL working.

Bit3: MCFORCE
When the system does not have XT1, Bit2: MCSEL setting is ignored and MAIN clock source is forcibly changed to PLL1.
0: No change.
1: Changes MAIN clock source to PLL1 forcibly.

Attention: Normally, this bit is used for a system without XT1. No change shall be made when MAIN clock is used as the system clock.

Bit2: MCSEL
Selects the clock source for MAIN clock.
0: XT1 output clock is the clock source of MAIN clock.
1: PLL1 output clock is the clock source of MAIN clock.

Attention: Change can be made when the output clocks of PLL1 and XT1 are stable. After setting, 4 cycles are needed for actual clock change.

Bit1-0: SCKSEL1-0
Selects the system clock. When RCCHGEN flag of the interrupt mode control register (IMCNT) is “1”, a hardware interrupt occurs and XTRTC (32.768 kHz) clock is selected as the system clock, or when it returns from WIC sleep mode, it automatically changes into “00 b”.
00 b: RC OSC clock is the system clock.
01 b: MAIN clock is the system clock.
10 b: XTRTC (32.768 kHz) clock is the system clock.

Attention: Change can be made when the selected clock is stabled before or after change. Note that the operation clocks of Cortex-M3 and LPDSP will also change. After setting, 4 cycles are needed for actual clock change.

1-3-3-2 PLL reference setting register: PLLREFCNT

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	P2REF SEL2	P2REF SEL1	P2REF SEL0	-	-	-	P1REF SEL
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R/W

Bit6-4: P2REFSEL2-0
Selects the reference clock of PLL2.
000 b: XT1
001 b: XTRTC (32.768 kHz)
010 b: MCLK
011 b: BCK0
100 b: BCK1
Others: Setting prohibited.

Attention:
If this setting is changed while PLL2 output clock is working as the audio clock, PLL2 gets unlocked and the audio clock gets into an abnormal state. The changes shall be made when PLL2 is not used as the audio clock source. To change this setting while PLL2 is being used as the audio clock source, change the setting and after PLL2 becomes stable enough, reset the whole audio.

Bit0: P1REFSEL
Selects PLL1 reference clock.
0: XT1
1: XTRTC (32.768 kHz)

Attention:
If this setting is changed while PLL1 output clock is working as the system clock, PLL1 gets unlocked and the system clock gets into an abnormal state. It can be changed only when PLL1 is not used as the system clock source.

1-3-3-3 Peripheral clock frequency dividing register: PERICLKDIV

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	EXT4 DIV5	EXT4 DIV4	EXT4 DIV3	EXT4 DIV2	EXT4 DIV1	EXT4 DIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	PCLK DIV5	PCLK DIV4	PCLK DIV3	PCLK DIV2	PCLK DIV1	PCLK DIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	HCLK DIV5	HCLK DIV4	HCLK DIV3	HCLK DIV2	HCLK DIV1	HCLK DIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit21-16: EXT4DIV5-0
Sets the clock division to be used for EXT4 block in AHB Peripheral.
AHB clock (EXT4 only) = BASIC clock ÷ (EXT4DIV + 1)

Attention: The same amount of cycles as the division ratio before the change will be necessary, at the maximum, from after the setting until the clock change. For example, when the setting before change is BASIC clock ÷ 8, it will take maximum of 8 cycles for the clock to change.

Bit13-8: PCLKDIV5-0
Sets the clock division to be used for APB Peripheral.
APB clock = BASIC clock ÷ (PCLKDIV + 1)

Attention: The same amount of cycles as the division ratio before the change will be necessary, at the maximum, from after the setting until the clock change. For example, when the setting before change is BASIC clock ÷ 8, it will take maximum of 8 cycles for the clock to change.

Bit5-0: HCLKDIV5-0
Sets the clock division to be used for AHB Peripheral (except EXT4 block).
AHB clock = BASIC clock ÷ (HCLKDIV + 1)

Attention: The same amount of cycles as the division ratio before the change will be necessary, at the maximum, from after the setting until the clock change. For example, when the setting before change is BASIC clock ÷ 8, it will take maximum of 8 cycles for the clock to change.

1-3-3-4 Function clock control register: FCLKCNT

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	PHI1SEL2	PHI1SEL1	PHI1SEL0	PHI0SEL2	PHI0SEL1	PHI0SEL0	SFDIV1	SFDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	UARTSEL	-	UDIV INT2	UDIV INT1	UDIV INT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	UDIV FRAC5	UDIV FRAC4	UDIV FRAC3	UDIV FRAC2	UDIV FRAC1	UDIV FRAC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	MTM3 RTC	-	TMSEL1	TMSEL0	-	TMDIV2	TMDIV1	TMDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R/W	R/W	R/W

Bit31-29: PHI1SEL2-0
Selects the clock to be output to PHI1 pin.
000 b: Outputs XTRTC output clock.
001 b: Outputs BASICCLK.
010 b: Outputs XT1 output clock.
011 b: Outputs 1/2 output clock of XT1.
100 b: Outputs 1/4 output clock of XT1.

Bit28-26: PHI0SEL2-0
Selects the clock to be output to PHI0 pin.
000 b: Outputs XTRTC output clock.
001 b: Outputs BASICCLK.
010 b: Outputs XT1 output clock.
011 b: Outputs 1/2 output clock of XT1.
100 b: Outputs 1/4 output clock of XT1.

Bit25-24: SFDIV1-0
Sets the clock division to be used for S-Flash I/F function clock.
00 b: BASIC clock ÷ 1
01 b: BASIC clock ÷ 2
10 b: BASIC clock ÷ 4
11 b: BASIC clock ÷ 8

Bit20: UARTSEL
Selects UART function clock source.
0: XT1 output clock is UART function clock source.
1: PLL1 output clock is UART function clock source.

Attention: No change shall be made when using UART.

Bit18-6: UDIVINT2-0
Sets the integer part of the dividing ratio of UART function clock divider.

Bit13-8: UDIVFRAC5-0
Sets the decimal part of the dividing ratio of UART function clock divider.

The frequency dividing ratio of UART function clock divider is as below.
UARTCLK = Clock source frequency / { (1 + UDIVINT) + (UDIVFRAC / 64) }

Bit7: MTM3RTC
Selects MTM3 function clock.
0 b: The clock selected in TMSEL1-0 is the function clock.
1 b: XTRTC clock is the function clock.

Attention: No change shall be made when MTM3 is used.

Bit5-4: TMSEL1-0
Selects PTM and MTM function clock.
00 b: XT1 clock is the function clock.
01 b: XT1 divided by 4 is the function clock.
10 b: PLL1 output clock divided is the function clock.

Attention: No change shall be made when PTM or MTM is used.

Bit2-0: TMDIV2-0
Selects the dividing ratio of PTM and MTM function clock divider.
000 b: Not divided.
001 b: Divided by 2.
010 b: Divided by 4.
011 b: Divided by 8.
100 b: Divided by 16.
Others: Setting prohibited.

Remark: Changes can be made while PTM or MTM is working.

1-3-3-5 Audio clock setting register: AUDCLKCNT

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	ASRC DIV1	ASRC DIV0	-	-	SSRC DIV1	SSRC DIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	ENCDIV1	ENCDIV0	reserved	reserved	DECDIV1	DECDIV0
Reset	0	0	0	0	1	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	reserved	reserved	-	-	AUDDIV1	AUDDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	XT1DIV AUD	-	-	AUDCK EN	AUDCK SEL
Reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R/W	R	R	R/W	R/W

Bit29-28: ASRCDIV1-0
Sets the clock division to be used for ASRC.
00 b: 1/1 dividing
01 b: 1/2 dividing
10 b: 1/4 dividing
11 b: 1/8 dividing

Remark: Change can be made while ASRC is working.

Bit25-24: SSRCDIV1-0
Sets the clock division to be used for SSRC.
00 b: 1/1 dividing
01 b: 1/2 dividing
10 b: 1/4 dividing
11 b: 1/8 dividing

Remark: Change can be made while SSRC is working.

Bit21-20: ENCDIV1-0
Sets the clock division to be used for MP3 Encoder.
00 b: 1/4 dividing
01 b: 1/8 dividing
10 b: 1/16 dividing
Others: Setting prohibited.

Remark: Change can be made while MP3 Encoder is working.

Bit17-16: DECDIV1-0
Sets the clock division to be used for MP3 Decoder.
00 b: 1/2 dividing
01 b: 1/4 dividing
10 b: 1/8 dividing
Others: Setting prohibited.

Remark: Change can be made while MP3Decoder is working.

Bit9-8: AUDDIV1-0
Sets the clock division to be used for Audio block.
00 b: 1/1 dividing
01 b: 1/2 dividing
10 b: 1/4 dividing
Others: Setting prohibited.

Remark: Change can be made while Audio block is working.

Bit4: XT1DIVAUD
Divides XT1 by 4 or not.
0: Uses XT1 clock.
1: Uses XT1 clock divided by 4.

Attention: No change shall be made when XT1 clock or XT1 divided-by-4 clock is used as the audio clock source, or is used as the reference clock with Audio PLL working.

Bit1: AUDCKEN
Controls the clock supply to the audio block.
0 b: Stops supplying clock to the audio.
1 b: Supplies clock to the audio.

Bit0: AUDCKSEL
Selects the clock for the audio block.
0 b: AHB clock
1 b: Audio PLL output clock

Attention: Change can be made only when AHB clock and PLL output cock are stabled. After setting, 4 cycles are needed for actual clock change. When AHB clock is used, AHB clock shall be 147.456 MHz or lower.

1-3-3-6 Interrupt mode control register: IMCNT

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	IMDIV2	IMDIV1	IMDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	XT1IEN	RC CHGEN	IMDIVEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit10-8: IMDIV2-0

When a hardware interrupt is generated, a value set here is automatically set in MAIN clock dividing register (MAINCLKDIV).

Bit2: XT1IEN

Controls XT1 oscillation start automatically when a hardware interrupt occurs. When it is set to "1", a hardware interrupt updates XT1EN flag of OSC control register (OSCCNT) to be “1” and starts XT1 oscillation. A hardware interrupt does nothing when XT1EN flag is “1”.

0: No action

1: An interrupt starts XT1 oscillation.

Bit1: RCCHGEN

Controls the system clock change from XTRTC (32.768 kHz) clock to RC OSC clock automatically when a hardware interrupt occurs. When it is set to “1” and XTRTC (32.768 kHz) clock is selected as the system clock at the time of a hardware interrupt, SCKSEL flag of OSC control register (OSCCNT) is updated to be “00” and RC OSC clock becomes the system clock. At this time, if RCOSCEN flag of OSC control register (OSCCNT) is "0," the flag is set to “1” at once and RC OSC oscillation starts. When SCKSEL flag does not select XTRTC (32.768 kHz) as the system clock at the time of a hardware interrupt, nothing is done.

0: No action

1: RC OSC clock becomes the system clock with the interrupt occurrence.

Bit0: IMDIVEN

Controls the update of the value in bit10-8 MAINDIV of OSC control register (OSCCNT) automatically when a hardware interrupt occurs. When it is set to "1", a hardware interrupt sets IMDIV value in MAINDIV.

0: No action

1: A hardware interrupt updates the value in bit10-8 MAINDIV of OSC control register (OSCCNT).

1-3-3-7 Software standby register: SSTBY

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	SSTBY
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0: SSTBY
This register can make forcible transition to WIC sleep. This register is automatically cleared at the time of return from WIC sleep state. Normally, do not use this register because it is for debug.
0: No action.
1: Forcible transition to WIC sleep.

When a debugger is connected, it cannot transition to WIC sleep state because Cortex-M3 cancels SLEEPDEEP signal. In such a case, set this register right before executing WFI command or WFE command for the transition to WIC sleep state. It can virtually transition to WIC sleep state to execute debugging.

1-3-3-8 Core status register: CORESTS

[Base + 0x0040]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	SLPD1	SLP1	SLPD0	SLP0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit3: SLPD1
Checks the power down state of Cortex-M3 core1. It is equivalent to SLEEPDEEP signal output from Cortex-M3.
0: Core is working.
1: Core is in WIC sleep (Both HCLK and FCLK are stopped.)

Bit2: SLP1
Checks the power down state of Cortex-M3 core1. It is equivalent to SLEEPING signal output from Cortex-M3.
0: Core is working.
1: Core is in sleep. (Only HCLK is stopped.)

Bit1: SLPD0
Checks the power down state of Cortex-M3 core0. It is equivalent to SLEEPDEEP signal output from Cortex-M3.
0: Core is working.
1: Core is in WIC sleep. (Both HCLK and FCLK are stopped.)

Bit0: SLP0
Checks the power down state of Cortex-M3 core0. It is equivalent to SLEEPING signal output from Cortex-M3.
0: Core is working.
1: Core is in sleep. (Only HCLK is stopped.)

1-4 Operation

This section provides Oscillation controller setting flow.

1-4-1 System clock generation circuit

The system clock configuration diagram is as shown below.

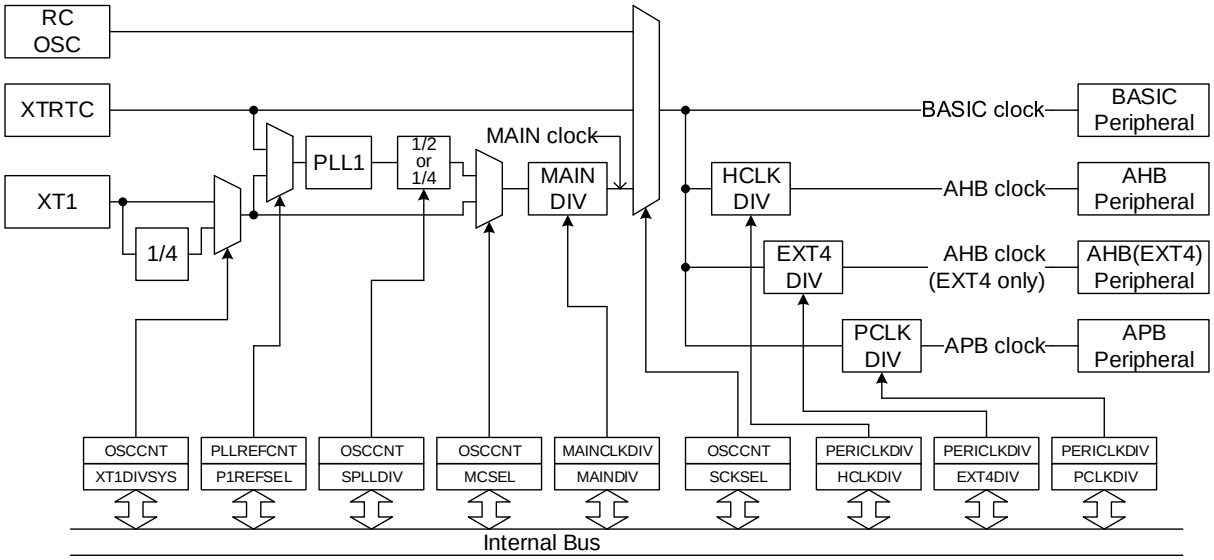


Figure 1-3 System clock configuration diagram

There are 4 clock sources for the system clock.

- RC OSC

A clock that is generated by LSI built-in RC OSC. This clock is selected after reset release or return from WIC sleep state.

- XTRTC

A clock (32.768 KHz) that is generated by an oscillator connected externally to LSI.

- XT1

A clock (1 to 50 MHz) that is generated by an oscillator connected externally to LSI.

- PLL1 output clock

A clock that is generated by LSI built-in PLL.

When PLL1 is the clock source, it is necessary to select the reference clock by setting bit0 P1REFSEL of PLL reference control register (PLLREFCNT), and to select the PLL1 output clock division by setting bit10-8 MAINDIV of OSC control register (OSCCNT). When PLL1 is selected as the system clock

source, P1REFSEL cannot be changed because PLL1 gets unlocked, but MAINDIV can be changed. To use PLL1, it is necessary to work PLL1 by using PLL1 controller. See *PLL1 Programmer's Model* for more information about PLL1 controller.

The system clock can be selected from 4 clock sources by setting bit1-0 SCKSEL and bit2 MCSEL of OSC control register (OSCCNT). To set MCSEL and SCKSEL, it is necessary that currently-selected clock and next-selected clock are stable and one of the remaining clocks is stable or stopped.

AHB Clock, AHB Clock (EXT4 only) and APB clock are used for AHB Peripheral and APB Peripheral, respectively. AHB Clock (EXT4 only) is AHB clock that is used only in EXT4 block where SDRAM controller exists. These BASIC Clock dividing ratio are determined by setting bit5-0 HCLKDIV, bit13-8 PCLKDIV and bit21-16 EXT4DIV of the peripheral clock dividing register (PERICLKDIV).

1-4-2 **Function clock generation circuit**

S-Flash I/F, PTM0–2, MTM0–3 and UART0–2 need a function clock for functional operations, apart from HCLK and PCLK.

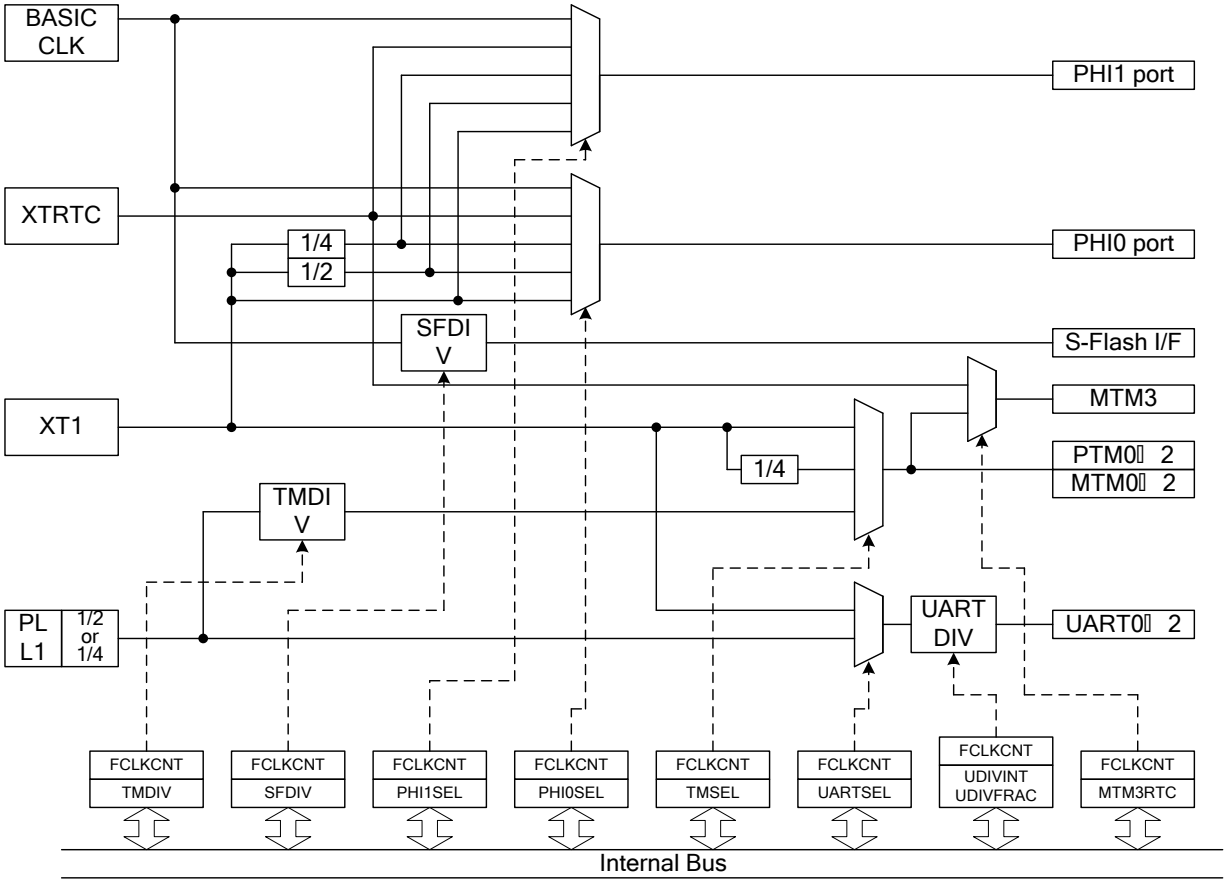


Figure 1-4 Function clock configuration diagram

The function clock used for S-Flash I/F is a clock to operate an external serial flash module. To make it operate at a frequency lower than the BASIC, change the external operation clock by setting SFDIV.

All PTM0–2 and MTM0–3 function clocks are the same and this clock is used for the count operation. Normally, XT1 clock or XT1 divided-by-4 clock is used, but when the system does not have XT1, PLL1 output clock can be used. XTRTC can be selected for MTM3.

UART0–2 function clock affects the communication clock. Normally, XT1 is used as the clock source, but when a system does not have XT1 or when high-speed communication is performed, PLL1 can be also used as the clock source.

1-4-3 Audio clock generation circuit

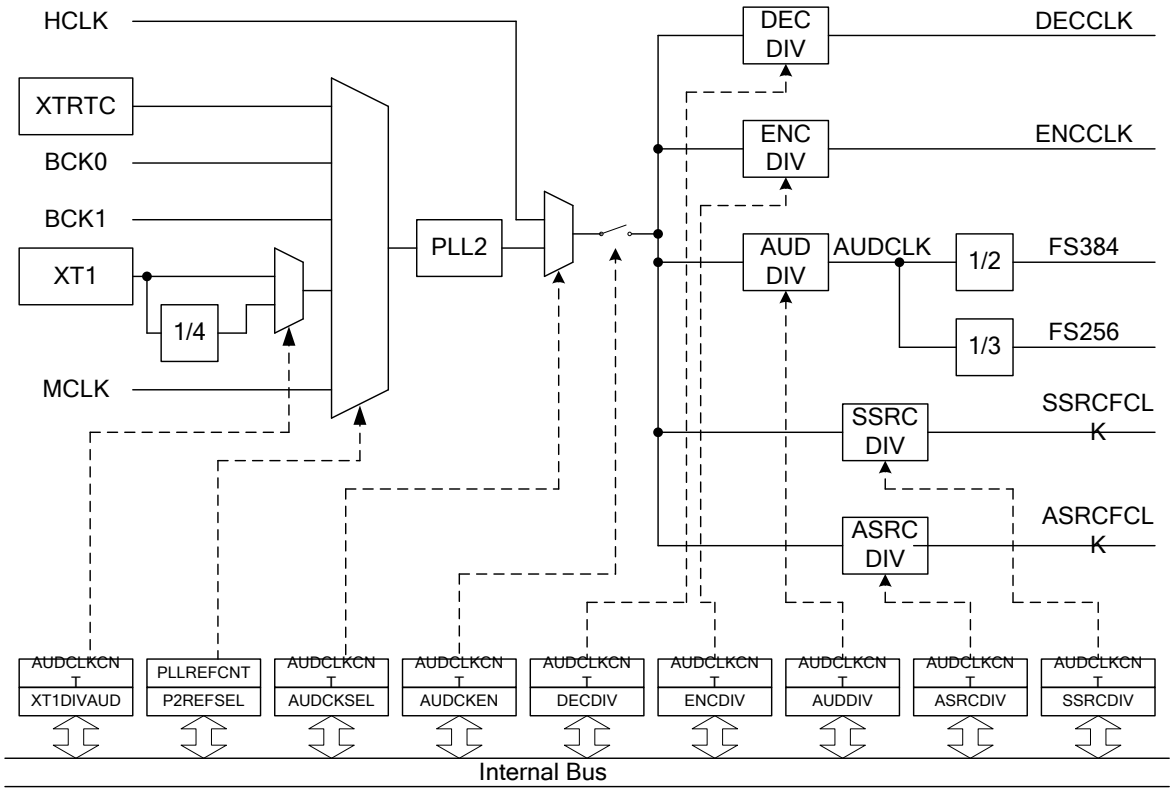


Figure 1-5 Audio clock configuration diagram

Normally, Audio clock uses PLL2 output clock as the clock source. Almost all modules work at a clock of FS384 or FS256 in the figure. Some modules work at other clocks. Clocks correspond to modules as shown below.

Table 1-2 Correspondence with clocks

Clock name	Module name
DECCLK	MP3 hardware decoder
ENCCLK	MP3 hardware encoder
SSRCFCLK	Synchronous sampling rate converter
ASRCFCLK	Asynchronous sampling rate converter

Normally, HCLK is not used as the clock source because its frequency is not related to the sampling frequency, but selecting HCLK allows register setting during the period until PLL2 oscillation becomes stable.

Attention: When HCLK is selected, HCLK shall be 147.456 MHz or lower.

1-4-4 WIC sleep condition

Because this LSI has two Cortex-M3 cores: Core0 and Core1, conditions for transition to WIC sleep state depend on whether Core1 is used or not. Here, the use of Core1 means that both bit0 C1CLKEN and bit1 C1RSTN of System Controller core control register (CORECNT) are set to “1”.
The differences are as listed below.

When Core1 is used.	When both Core0 and Core1 set bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) to “1” and WFI command or WFE command is executed by program, it transitions to WIC sleep state.
When Core1 is not used.	When Core0 sets bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) to “1” and WFI command or WFE command is executed by program, it transitions to WIC sleep state.

When Core1 is used, only one of Core0 or Core1 sets bit2 SLEEPDEEP of Cortex-M3 system control register (SCR) to “1” and WFI command or WFE command is executed by program, the core that has issued WFI command or WFE command stops the clock and the other core normally works.

1-4-5 Hardware interrupt

Oscillation controller is worked partly by a hardware interrupt as below.

Return from the sleep state to the run state

Return from WIC sleep state to the run state

Operation related to the interrupt mode control register (IMCNT)

1-4-5-1 Return from the sleep state to the run state

It differs between Cortex-M3 and LPDSP.

In the case of Cortex-M3

In the case of LPDSP

LPDSP core has 15 interrupt inputs that are controlled by the interrupt controller (INTC). When LPDSP returns from the sleep state to the run state, it is necessary that a hardware interrupt that the interrupt controller (INTC) has assigned to LPDSP core becomes active and the hardware interrupt is not masked.

1-4-5-2 Return from WIC sleep state to the run state

It depends on whether Cortex-M3 Core1 is used or not. Here, the use of Core1 means that both bit0 C1CLKEN and bit1 C1RSTN of System Controller core control register (CORECNT) are set to “1”. The differences are as listed below.

When Core1 is used.	It is necessary that an external interrupt occurs in NVIC of either Core0 or Core1 or NMI to become active.
When Core1 is not used.	It is necessary that an external interrupt occurs in Core0 NVIC or NMI to become active.

It is necessary that the interrupt controller (INTC) assigns a hardware interrupt to NVIC external interrupt or NMI, and it is not masked. In WIC sleep state, the whole LSI is stopped except for XTRTC OSC (32 kHz) and RTC. So, interrupt causes are as below.

NVIC external interrupt, or external pin interrupt assigned to NMI (EXTINT00–EXTINT59)

RTC timer interrupt

RTC KEYINT interrupt

USB asynchronous interrupt

1-4-5-3 Operation related to the interrupt mode control register (IMCNT)

Conditions for the interrupt mode control register (IMCNT) to recognize a hardware interrupt occurrence depend on whether Cortex-M3 Core1 is used or not. Here, the use of Core1 means that both bit0 C1CLKEN and bit1 C1RSTN of System Controller core control register (CORECNT) are set to “1”. The differences are as listed below.

When Core1 is used.	It is necessary that an external interrupt occurs in NVIC of either Core0 or Core1 or NMI to become active.
When Core1 is not used.	It is necessary that an external interrupt occurs in Core0 NVIC or NMI to become active.

It is necessary that the interrupt controller (INTC) assigns a hardware interrupt to NVIC external interrupt or NMI, and it is not masked.

2 SystemPLL

2-1 Introduction

This chapter describes the outline of SysPLL in the section below.

- The Outline of SysPLL

2-1-1 The Outline of SysPLL

SysPLL controller generates a system clock.

2-2 Functional Overview

This chapter describes the outline of SysPLL function block in the section below.

- SysPLL Circuit Configuration

2-2-1 SysPLL Circuit Configuration

SysPLL circuit configuration is as shown in Figure 2-1

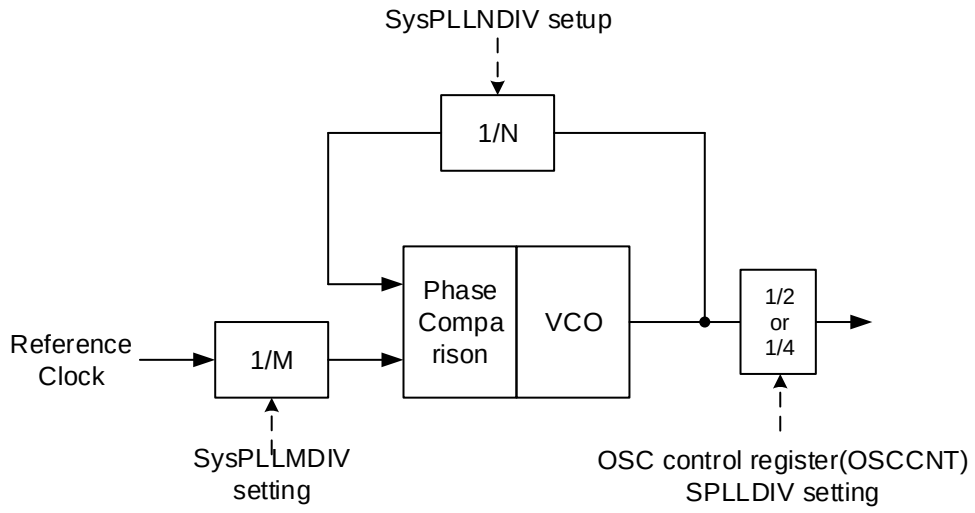


Figure 2-1

2-3 Programmer's Model

This chapter contains SysPLL register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the Programmer's Model
- Summary of SysPLL Registers
- Register Descriptions

2-3-1 About the Programmer's Model

SysPLL base address is not fixed and varies by the system implementation.

Please refer to the *System specifications* for more information about the base address.

Offset is fixed for all registers.

2-3-2 Summary of SysPLL Registers

SysPLL is controlled by the registers in Table 2-1.

Remarks: Do not access any unspecified address in the address range between Base+0x0000 and Base+0xFFFF.

Table 2-1 SysPLL registers

Offset	Abbrev.	Attribute	Contents	Initial value
0x00	SysPLLCNT	R/W	SysPLL control register	0x0000_0600
004	SysPLLMDIV	R/W	Reference clock divider's setting register	0x0000_0000
0x08	SysPLLNDIV	R/W	Feedback clock divider's setting register	0x0000_0000

2-3-3 Register Descriptions

2-3-3-1 SysPLL Control Register : SysPLLCNT

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	SysPLLS2	SysPLLS1	SysPLLS0
Reset	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	SysPLL STYB	SysPLL RESETB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit10-8: SysPLLS2-0
Selects SysPLL internal resistance
000 b: 10.5 k
001 b: 8.5 k
010 b: 7.5 k
011 b: 5.5 k
100 b: 6.5 k
101 b: 4.5 k
110 b: 3.5 k
111 b: 1.5 k

Remark: Please select 3.5 k for normal use.

Bit2-1: SysPLLSTYB
SysPLL standby setting
0: Standby
1: Standby cancel

Bit0: SysPLLRESETB
SysPLL reset setting
0: Reset
1: No reset

2-3-3-2 Reference Clock Divider's Setting Register : SysPLLMDIV

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	SysPLL MDIV7	SysPLL MDIV6	SysPLL MDIV5	SysPLL MDIV4	SysPLL MDIV3	SysPLL MDIV2	SysPLL MDIV1	SysPLL MDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: SysPLLMDIV7-0
Reference frequency dividing ratio
0x00: Not divided
0x01: 1/2 frequency
0x02: 1/3 frequency
0x03: 1/4 frequency

0xFC: 1/253 frequency
0xFD: 1/254 frequency
0xFE: 1/255 frequency
0xFF: 1/256 frequency

* To determine setting values, see Chapter 2-4-1 Output Frequency Setting and Chapter 2-4-2 Precautions for use and consider that VCO oscillation frequency is specified and that an output clock that has 1/2 of VCO oscillation frequency is used to generate the system clock.

2-3-3-3 Feedback Clock Divider's Setting Register: SysPLLNDIV

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	SysPLL NDIV13	SysPLL NDIV12	SysPLL NDIV11	SysPLL NDIV10	SysPLL NDIV9	SysPLL NDIV8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	SysPLL NDIV7	SysPLL NDIV6	SysPLL NDIV5	SysPLL NDIV4	SysPLL NDIV3	SysPLL NDIV2	SysPLL NDIV1	SysPLL NDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit13-0: SysPLLNDIV13-0
Feedback frequency dividing ratio
0x0000: Not divided.
0x0001: 1/2 frequency
0x0002: 1/3 frequency
0x0003: 1/4 frequency

0x3FFC: 1/16381 frequency
0x3FFD: 1/16382 frequency
0x3FFE: 1/16383 frequency
0x3FFF: 1/16384 frequency

* To determine setting values, see 2-4-1 Output Frequency Setting and 2-4-2 Precautions for use and consider that VCO oscillation frequency is specified and that an output clock that has 1/2 of VCO oscillation frequency is used to generate the system clock.

2-4 Operation

This section describes SysPLL setting flow.

2-4-1 Output Frequency Setting

The input clock and the output clock of SysPLL are related as follows:

$$\text{Phase comparison frequency [Hz]} = \frac{\text{Input clock frequency [Hz]}}{\text{Reference frequency dividing setting register value} + 1}$$

VCO oscillation frequency [Hz] = Phase comparison frequency [Hz] × (Feedback frequency dividing setting register value + 1)

$$\text{Output clock frequency [Hz]} = \frac{\text{VCO oscillation frequency [Hz]}}{\alpha}$$

The value α is determined by the setting in OSC control register (OSCCNT) and SPLLDIV of Oscillation Controller.

When 1/2 frequency is selected: α = 2.

When 1/4 frequency is selected: α = 4.

2-4-2 Precautions for use

Phase comparison frequency and VCO oscillation frequency have limits. Please refer to the datasheet for more information about the limits. The system clock is generated from an output clock generated by SysPLL. Therefore, the input clock frequency of the oscillation controller is 1/2 or 1/4 of PLL oscillation frequency (VCO oscillation frequency). Please pay attention to the relationship between system clock frequency and reference and feedback frequency dividing settings. See *Oscillation Controller User's Model* for the frequency divider setting.

3 AudioPLL

3-1 Introduction

This chapter provides the outline of AudPLL in the section below.

- The outline of AudPLL

3-1-1 The outline of AudPLL

AudPLL controller generates an Audio clock.

3-2 Functional Overview

This chapter provides the outline of functional blocks used for the AudPLL configuration in the section below.

- AudPLL circuit configuration

3-2-1 AudPLL circuit configuration

AudPLL circuit configuration is as shown in Figure 3-1

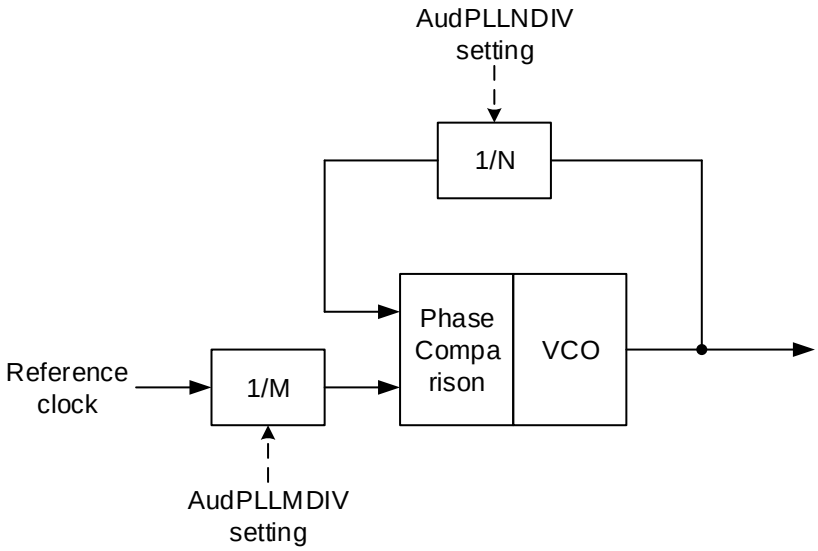


Figure 3-1

3-3 Programmer's Model

This chapter contains AudPLL register information and other information necessary for the register setup using microcontrollers in the sections below.

- About the Programmer's Model
- Summary of AudPLL Registers
- Register Descriptions

3-3-1 About the Programmer's Model

AudPLL base address is not fixed and varies by the system implementation.

Please refer to the *System specification* for more information about base address.

Offset is fixed for all registers.

3-3-2 Summary of AudPLL Registers

AudPLL is controlled by the registers listed in Table 3-1.

Remarks: Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 3-1 AudPLL registers

Offset	Abbrev.	Attribute	Register name	Initial value
0x00	AudPLLCNT	R/W	AudPLL control register	0x0000_0500
004	AudPLLMDIV	R/W	Reference clock divider's setting register	0x0000_0000
0x08	AudPLLNDIV	R/W	Feedback clock divider's setting register	0x0000_0000

3-3-3 Register Descriptions

3-3-3-1 AudPLL Control Register: AudPLLCNT

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	AudPLLS2	AudPLLS1	AudPLLS0
Reset	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	AudPLL STYB	AudPLL RESETB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit10-8: AudPLLS2-0
Sets AudPLL internal resistance
000 b: 10.5 k
001 b: 8.5 k
010 b: 7.5 k
011 b: 5.5 k
100 b: 6.5 k
101 b: 4.5 k
110 b: 3.5 k
111 b: 1.5 k

Remarks: Select 4.5 k for the normal use.

Bit1: AudPLLSTYB
AudPLL standby setting
0: Standby
1: Standby released

Bit0 : AudPLLRESETB
AudPLL reset setting
0: Reset
1: Does not reset.

3-3-3-2 Reference Clock Divider's Setting Register: AudPLLMDIV

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	AudPLL MDIV11	AudPLL MDIV10	AudPLL MDIV9	AudPLL MDIV8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	AudPLL MDIV7	AudPLL MDIV6	AudPLL MDIV5	AudPLL MDIV4	AudPLL MDIV3	AudPLL MDIV2	AudPLL MDIV1	AudPLL MDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit11-0 : AudPLLMDIV11-0
Reference frequency dividing ratio
0x000: No frequency division
0x001: 1/2 frequency division
0x002: 1/3 frequency division
0x003: 1/4 frequency division

0xFFC: 1/4093 frequency division
0xFFD: 1/4094 frequency division
0xFFE: 1/4095 frequency division
0xFFF: 1/4096 frequency division

3-3-3-3 Feedback Clock Divider's Setting Register: AudPLLNDIV

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	AudPLL NDIV15	AudPLL NDIV14	AudPLL NDIV13	AudPLL NDIV12	AudPLL NDIV11	AudPLL NDIV10	AudPLL NDIV9	AudPLL NDIV8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	AudPLL NDIV7	AudPLL NDIV6	AudPLL NDIV5	AudPLL NDIV4	AudPLL NDIV3	AudPLL NDIV2	AudPLL NDIV1	AudPLL NDIV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : AudPLLNDIV15-0
Feedback frequency dividing ratio
0x0000: No frequency division
0x0001: 1/2 frequency division
0x0002: 1/3 frequency division
0x0003: 1/4 frequency division

0xFFFC: 1/65533 frequency division
0xFFFD: 1/65534 frequency division
0xFFFE: 1/65535 frequency division
0xFFFF: 1/65536 frequency division

3-4 Operation

This section provides AudPLL setting flow.

3-4-1 Output Frequency Setting

The input clock and the output clock of AudPLL are related as shown below.

$$\text{Phase comparison frequency [Hz]} = \frac{\text{Input clock frequency [Hz]}}{\text{Reference frequency dividing setting register value} + 1}$$

VCO oscillation frequency [Hz] = Phase comparison frequency [Hz] × (Feedback frequency dividing setting register value + 1)

Output clock frequency [Hz] = VCO oscillation frequency [Hz]

3-4-2 Precautions for use

Phase comparison frequency and VCO oscillation frequency have limits. Please refer to the datasheet for more information about the limits. The audio clock is generated by dividing the clocks generated by AudPLL. Please refer to the Oscillation Controller User's Model *for the information about division settings.*

4 INTC

4-1 Introduction

This chapter provides the brief summary of INTC in the sections bellow:

- Brief Summary of INTC

4-1-1 Brief Summary of INTC

This INTC is a functional circuitry that assists mutual interrupt between ARM Cortex-M3 Core0, Core1, and LPDSP32 , and built-in Interrupt Controller(NVIC) of ARM Cortex-M3.
Moreover, it control the interrupt of LPDSP32 too.

The relation between Cortex-M3, NVIC and this INTC is shown by the following block chart.

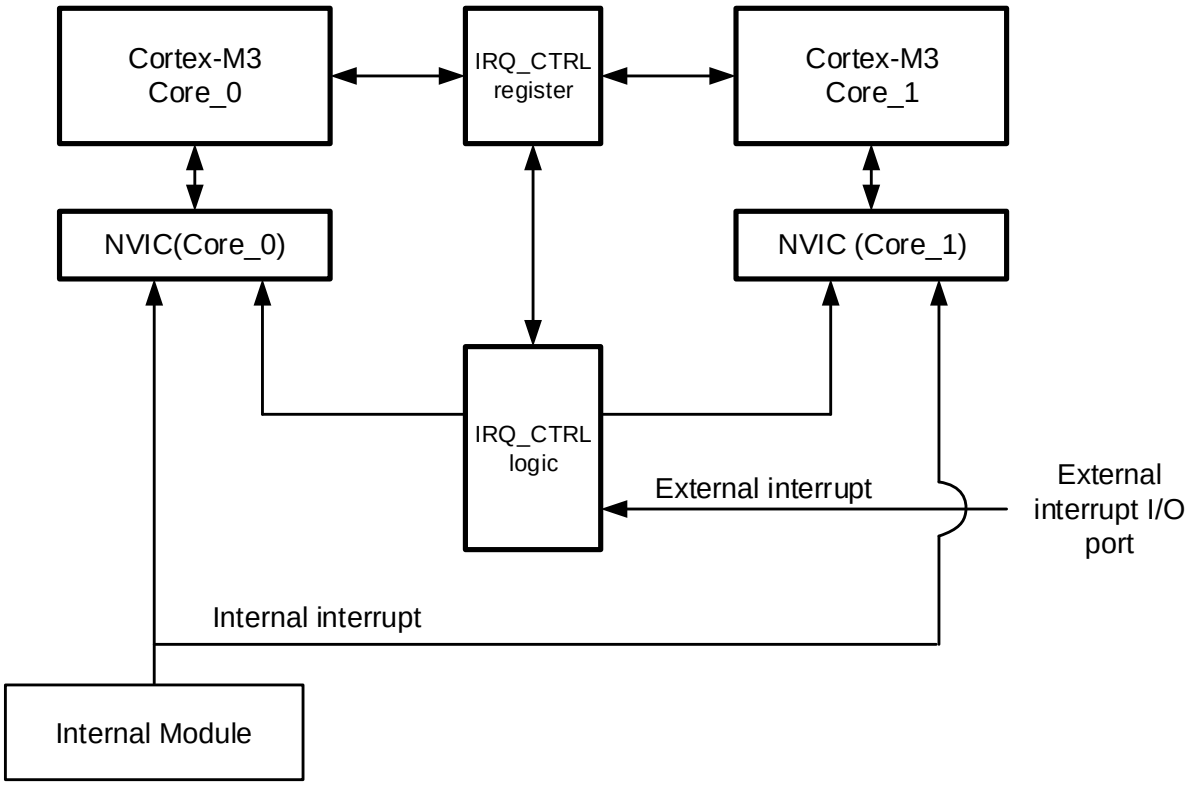


Figure 4-1

The interrupt handling of NVIC is shown by the following timing chart.

Because the registers which work with BASIC clock do handling of the interrupts between cores and external interrupts, the interrupt is asserted longer than 1 clock period. But when interrupts between cores are occurd repeatedly, Cortex-M3 can't accept interrupt if the bit of NVIC_ISPR register is high. Therefor software should control "Interrupt factor between cores confirmation & generation

register : IPIREG" of INTC.

* NVIC_ISPR is the register of Cortex-M3's NVIC. Please refer to Cortex-M3 technical reference maual if you need to understand.

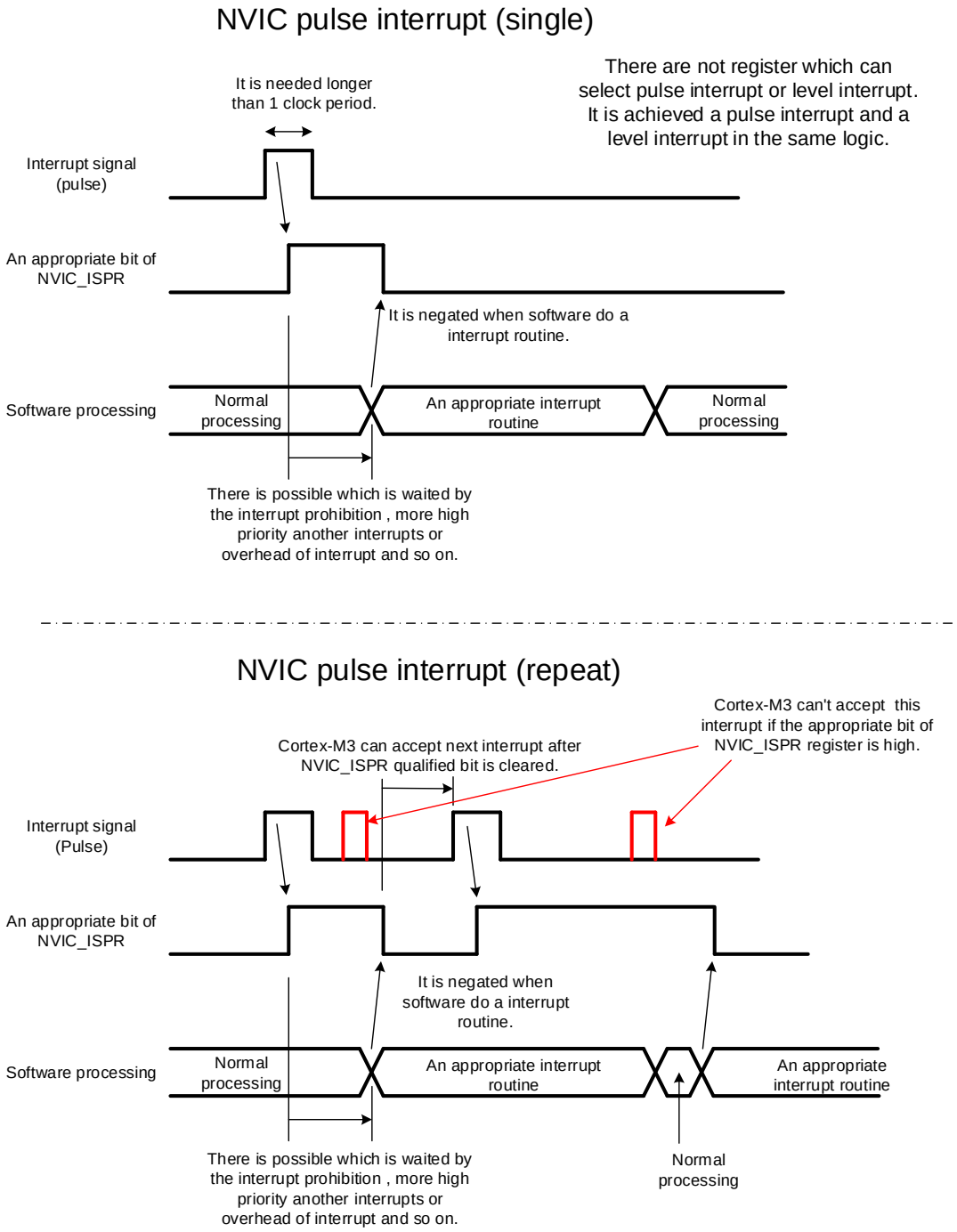

NVIC pulse interrupt (single)
There are not register which can select pulse interrupt or level interrupt. It is achieved a pulse interrupt and a level interrupt in the same logic.
It is needed longer than 1 clock period.
Interrupt signal (pulse)
An appropriate bit of NVIC_ISPR
It is negated when software do a interrupt routine.
Software processing
Normal processing
An appropriate interrupt routine
Normal processing
There is possible which is waited by the interrupt prohibition , more high priority another interrupts or overhead of interrupt and so on.
NVIC pulse interrupt (repeat)
Cortex-M3 can't accept this interrupt if the appropriate bit of NVIC_ISPR register is high.
Cortex-M3 can accept next interrupt after NVIC_ISPR qualified bit is cleared.
Interrupt signal (Pulse)
An appropriate bit of NVIC_ISPR
It is negated when software do a interrupt routine.
Software processing
Normal processing
An appropriate interrupt routine
An appropriate interrupt routine
There is possible which is waited by the interrupt prohibition , more high priority another interrupts or overhead of interrupt and so on.
Normal processing

Figure 4-2

4-2 Functional Overview

This chapter provides overview of functional blocks used for the INTC configuration in the sections below.

The remainder of this page is intentionally left blank.

4-3 Programmer's Model

This chapter contains INTC register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of Interrupt Controller registers
- Register Descriptions

4-3-1 About the programmer's model

INTC base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

4-3-2 Summary of Interrupt Controller registers

INTC is controlled by the registers shown in Table 4-1 below.

Remarks) Accessible addresses are the ones listed in between the range of “Base+0x0000” to “Base+0xFFFF”. Do not access any other address.

Table 4-1 Interrupt Controller registers

Offset	Abbreviation	Attribute	Content	Initial value
0x000	IPIREG	R/W	Interrupt factor confirmation & generation register between cores	0x0000_0000
0x004	IPICLR	R/W	Interrupt factor clear register between cores	0x0000_0000
0x008	WTMINTSET	R/W	Watch Dog Timer interrupt setting [arejisuta]	0x0000_0000
0x400	EXTINT0	R	0 external interrupt factor register	0x0000_0000
0x404	EXTINT1	R	One external interrupt factor register	0x0000_0000
0x408	EXTINT2	R	Two external interrupt factor register	0x0000_0000
0x40C	EXTINT3	R	Three external interrupt factor register	0x0000_0000
0x410	EXTINT4	R	Four external interrupt factor register	0x0000_0000
0x414	EXTINT5	R	Five external interrupt factor register	0x0000_0000
0x418	EXTINT0S	R	Source register of 0 external interrupt factors	0x0000_0000
0x41C	EXTINT1S	R	Source register of one external interrupt factor	0x0000_0000
0x420	EXTINT2S	R	Source register of two external interrupt factors	0x0000_0000
0x424	EXTINT3S	R	Source register of three external interrupt factors	0x0000_0000
0x428	EXTINT4S	R	Source register of four external interrupt factors	0x0000_0000
0x42C	EXTINT5S	R	Source register of five external interrupt factors	0x0000_0000
0x430	EXTINT0M	R/W	Mask register of 0 external interrupt factors	0x0000_FFFF
0x434	EXTINT1M	R/W	Mask register of one external interrupt factor	0x0000_FFFF
0x438	EXTINT2M	R/W	Mask register of two external interrupt factors	0x0000_FFFF
0x43C	EXTINT3M	R/W	Mask register of three external interrupt factors	0x0000_FFFF
0x440	EXTINT4M	R/W	Mask register of four external interrupt factors	0x0000_FFFF
0x444	EXTINT5M	R/W	Mask register of five external interrupt factors	0x0000_03FF
0x448	EXTINT0C0	R/W	0 external interrupt factor set register (core 0)	0x0000_0000
0x44C	EXTINT1C0	R/W	One external interrupt factor set register (core 0)	0x0000_0000
0x450	EXTINT2C0	R/W	Two external interrupt factor set register (core 0)	0x0000_0000
0x454	EXTINT3C0	R/W	Three external interrupt factor set register (core 0)	0x0000_0000
0x458	EXTINT4C0	R/W	Four external interrupt factor set register (core 0)	0x0000_0000
0x45C	EXTINT5C0	R/W	Five external interrupt factor set register (core 0)	0x0000_0000
0x460	EXTINT0C1	R/W	0 external interrupt factor set register (core 1)	0x0000_0000
0x464	EXTINT1C1	R/W	One external interrupt factor set register (core 1)	0x0000_0000
0x468	EXTINT2C1	R/W	Two external interrupt factor set register	0x0000_0000

			(core 1)	
0x46C	EXTINT3C1	R/W	Three external interrupt factor set register (core 1)	0x0000_0000
0x470	EXTINT4C1	R/W	Four external interrupt factor set register (core 1)	0x0000_0000
0x474	EXTINT5C1	R/W	Five external interrupt factor set register (core 1)	0x0000_0000
0x478	EXTINT0CND	R/W	0 external interrupt condition set register	0x0000_0000
0x47C	EXTINT1CND	R/W	One external interrupt condition set register	0x0000_0000
0x480	EXTINT2CND	R/W	Two external interrupt condition set register	0x0000_0000
0x484	EXTINT3CND	R/W	Three external interrupt condition set register	0x0000_0000
0x488	EXTINT4CND	R/W	Four external interrupt condition set register	0x0000_0000
0x48C	EXTINT5CND	R/W	Five external interrupt condition set register	0x0000_0000
0x490	EXTINT0CLR	R/W	External interrupt 0 clear register	0x0000_0000
0x494	EXTINT1CLR	R/W	External interrupt 1 clear register	0x0000_0000
0x498	EXTINT2CLR	R/W	External interrupt 2 clear register	0x0000_0000
0x49C	EXTINT3CLR	R/W	External interrupt 3 clear register	0x0000_0000
0x4A0	EXTINT4CLR	R/W	External interrupt 4 clear register	0x0000_0000
0x4A4	EXTINT5CLR	R/W	External interrupt 5 clear register	0x0000_0000
0x4A8	EXTINT0FEN	R/W	Permission register of 0 external interrupt noise filters	0x0000_0000
0x4AC	EXTINT1FEN	R/W	Permission register of one external interrupt noise filter	0x0000_0000
0x4B0	EXTINT2FEN	R/W	Permission register of two external interrupt noise filters	0x0000_0000
0x4B4	EXTINT3FEN	R/W	Permission register of three external interrupt noise filters	0x0000_0000
0x4B8	EXTINT4FEN	R/W	Permission register of four external interrupt noise filters	0x0000_0000
0x4BC	EXTINT5FEN	R/W	Permission register of five external interrupt noise filters	0x0000_0000
0x4C0	EXTINT0FSET	R/W	0 external interrupt noise filter set register	0x0000_0000
0x4C4	EXTINT1FSET	R/W	One external interrupt noise filter set register	0x0000_0000
0x4C8	EXTINT2FSET	R/W	Two external interrupt noise filter set register	0x0000_0000
0x4CC	EXTINT3FSET	R/W	Three external interrupt noise filter set register	0x0000_0000
0x4D0	EXTINT4FSET	R/W	Four external interrupt noise filter set register	0x0000_0000
0x4D4	EXTINT5FSET	R/W	Five external interrupt noise filter set register	0x0000_0000
0x800	DSPINT1STS	R	One LPDSP32 interrupt status register	0x0000_0000
0x804	DSPINT2STS	R	Two LPDSP32 interrupt status registers	0x0000_0000
0x808	DSPINT3STS	R	Three LPDSP32 interrupt status registers	0x0000_0000
0x80C	DSPINT4STS	R	Four LPDSP32 interrupt status registers	0x0000_0000
0x810	DSPINT5STS	R	Five LPDSP32 interrupt status registers	0x0000_0000
0x814	DSPINT6STS	R	Six LPDSP32 interrupt status registers	0x0000_0000
0x818	DSPINT7STS	R	Seven LPDSP32 interrupt status registers	0x0000_0000
0x81C	DSPINT8STS	R	Eight LPDSP32 interrupt status registers	0x0000_0000

0x820	DSPINT9STS	R	Nine LPDSP32 interrupt status registers	0x0000_0000
0x824	DSPINT10STS	R	Ten LPDSP32 interrupt status registers	0x0000_0000
0x828	DSPINT11STS	R	11 LPDSP32 interrupt status registers	0x0000_0000
0x82C	DSPINT12STS	R	12 LPDSP32 interrupt status registers	0x0000_0000
0x830	DSPINT13STS	R	13 LPDSP32 interrupt status registers	0x0000_0000
0x834	DSPINT14STS	R	14 LPDSP32 interrupt status registers	0x0000_0000
0x838	DSPINT15STS0	R	15 LPDSP32 interrupt status registers	0x0000_0000
0x83C	DSPINT15STS1	R	15 LPDSP32 interrupt status registers	0x0000_0000
0x840	DSPINTMASK0	R/W	LPDSP32 interrupt mask register	0x FFFF_ FFFF
0x844	DSPINTMASK1	R/W	LPDSP32 interrupt mask register	0x FFFF_ FFFF
0x848	DSPINTSRCSTS0	R	LPDSP32 interrupt source status register	0x0000_0000
0x84C	DSPINTSRCSTS1	R	LPDSP32 interrupt source status register	0x0000_0000
0x850	DSPINTSET0	R/W	LPDSP32 interrupt setting register 0	0xFFFF_FFFF
0x854	DSPINTSET1	R/W	LPDSP32 interrupt setting register 1	0xFFFF_FFFF
0x858	DSPINTSET2	R/W	LPDSP32 interrupt setting register 2	0xFFFF_FFFF
0x85C	DSPINTSET3	R/W	LPDSP32 interrupt setting register 3	0xFFFF_FFFF
0x860	DSPINTSET4	R/W	LPDSP32 interrupt setting register 4	0xFFFF_FFFF
0x864	DSPINTSET5	R/W	LPDSP32 interrupt setting register 5	0xFFFF_FFFF
0x868	DSPINTSET6	R/W	LPDSP32 interrupt setting register 6	0xFFFF_FFFF
0x86C	DSPINTSET7	R/W	LPDSP32 interrupt setting register 7	0xFFFF_FFFF
0x870	DSPDMACLR	R/W	LPDSP32 DMA [sute-tasukuriajisuta]	0x0000_0000
0x874	DSPDMBCLR	R/W	LPDSP32 DMB status clear register	0x0000_0000
0x878	DSPPMCLR	R/W	LPDSP32 PM status clear register	0x0000_0000
0x87C	DSPDMASTS	R/W	LPDSP32 DMA [sute-tasujisuta]	0x0000_0000
0x880	DSPDMBSTS	R/W	LPDSP32 DMB status register	0x0000_0000
0x884	DSPPMSTS	R/W	LPDSP32 PM status register	0x0000_0000
0x900	Reserved	R/W	-	0x0000_0000
0x904	Reserved	R/W	-	0x0000_0000
0x908	Reserved	R/W	-	0x0000_0000
0x90C	Reserved	R/W	-	0x0000_0000

4-3-3 Register Descriptions

4-3-3-1 Interrupt factor between cores confirmation & generation register : IPIREG

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	INTISR3_DSP	INTISR2_DSP	INTISR1_DSP	INTISR0_DSP
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	INTISR1_3	INTISR1_2	INTISR1_1	INTISR1_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	INTISR0_3	INTISR0_2	INTISR0_1	INTISR0_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Interrupt state confirmation register.

It can confirm from which core interrupt has been generated.

Moreover, it has the role for each core to generate interrupt from the corresponding core by writing "1" in this register.

It is invalid to write 0 in this register.

- At Read

Bit19-Bit16 : INTISRx_DSP

Monitor register of interrupt factor signal x from LPDSP32
0 : Interrupt signal x from LPDSP32 is "L" level.
1 : Interrupt signal x from LPDSP32 is "H" level.

Note) x is one from 0 to 3.
The relation to the interrupt number in Cortex-M3 core 0 and core 1 is as follows.
x=0: Interrupt number 8
x=1: Interrupt number 9
x=2: Interrupt number 10
x=3: Interrupt number 11

Moreover, it is the interrupt factor of LPDSP32 itself too.

The relation to the interrupt number is as follows.

x=0: Interrupt number 8
x=1: Interrupt number 9
x=2: Interrupt number 10
x=3: Interrupt number 11

Bit11-Bit8 : INTISR1_x
Monitor register of interrupt factor signal x from Cortex-M3 Core1
0 : Interrupt signal x from Cortex-M3 Core1 is "L" level.
1 : Interrupt signal x from Cortex-M3 Core1 is "H" level.

Note) x is one from 0 to 3.
The relation to the interrupt number of Cortex-M3 Core0 and LPDSP32 is as follows.
x=0: Interrupt number 4
x=1: Interrupt number 5
x=2: Interrupt number 6
x=3: Interrupt number 7

Moreover, it is the interrupt factor of Cortex-M3 Core1 itself too.

The relation to the interrupt number is as follows.

x=0: Interrupt number 4
x=1: Interrupt number 5
x=2: Interrupt number 6
x=3: Interrupt number 7

Bit3-Bit0 : INTISR0_x
Monitor regsiter of interrupt factor signal x from Cortex-M3 Core0
0 : Interrupt signal x from Cortex-M3 Core0 is "L" level.
1 : Interrupt signal x from Cortex-M3 Core0 is "H" level.

Note) x is one from 0 to 3.
The relation to the interrupt number of Cortex-M3 Core1 and LPDSP32 is as follows.
x=0: Interrupt number 0
x=1: Interrupt number 1
x=2: Interrupt number 2
x=3: Interrupt number 3

Moreover, it is the interrupt factor of Cortex-M3 Core0 itself too.

The relation to the interrupt number is as follows.

x=0: Interrupt number 0
x=1: Interrupt number 1
x=2: Interrupt number 2
x=3: Interrupt number 3

- At Write

Bit19-Bit16 : INTISRx_DSP
Interrupt generate register from LPDSP32 to each core
It is an interrupt signal x from LPDSP32.
0 : It is invalid. There is no effect.
1 : "H" level is supplied to interrupt factor signal x from LPDSP32.

Note) It is a prohibition that Cortex-M3 Core0 and Core1 write "1". Only writing "0" is allowed.
x is one from 0 to 3.

Bit11-Bit8 : INTISR1_x
Interrupt generate register from Cortex-M3 Core1 to each core
It is an interrupt signal x from Cortex-M3 Core1.
0 :It is invalid. There is no effect.
1 : "H" level is supplied to interrupt factor signal x from Cortex-M3 Core1.

Note) It is a prohibition that Cortex-M3 Core0 and LPDSP32 write "1". Only writing "0" is allowed.
x is one from 0 to 3.

Bit3-Bit0 : INTISR0_x
Interrupt generate register from Cortex-M3 Core0 to each core
It is an interrupt signal x from Cortex-M3 Core0. .
0 :It is invalid. There is no effect.
1 : "H" level is supplied to interrupt factor signal x from Cortex-M3 Core0.

Note) It is a prohibition that Cortex-M3 Core1 and LPDSP32 write "1". Only writing "0" is allowd.
x is one from 0 to 3.

4-3-3-2 The interrupt factor between cores clear register: IPICLR

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	INTISR3 CLR_DSP	INTISR2 CLR_DSP	INTISR1 CLR_DSP	INTISR0 CLR_DSP
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	INTISR1 CLR_3	INTISR1 CLR_2	INTISR1 CLR_1	INTISR1 CLR_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	INTISR0 CLR_3	INTISR0 CLR_2	INTISR0 CLR_1	INTISR0 CLR_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Interrupt in the corresponding core can be cleared by writing "1" in this register.

It is invalid to write 0 in this register. There is no effect.

It is possible to use it as an interrupt state confirmation register at Read.

- At Write

Bit19-Bit16 : INTISRxCLR_DSP

Clear register of the factor of the interrupt signal x of Cortex-M3 Core0 and Cortex-M3 Core1
0 : It is invalid. There is no effect.
1 : Interrupt signal x of Cortex-M3 Core0 and Cortex-M3 Core1 is cleared.

Note) x is one from 0 to 3.

Bit11-Bit8 : INTISR1CLR_x

Clear register of the factor of the interrupt signal x of LPDSP32 and Cortex-M3 Core0
0 : It is invalid. There is no effect.
1 : Interrupt signal x of LPDSP32 and Cortex-M3 Core0 is cleared.

Note) x is one from 0 to 3.

Bit3-Bit0 : INTISR0CLR_x

Clear register of the factor of the interrupt signal x of LPDSP32 and Cortex-M3 Core1
0 : It is invalid. There is no effect.
1 : Interrupt signal x of LPDSP32 and Cortex-M3 Core1 is cleared.

Note) x is one from 0 to 3.

- At Read

Bit19-Bit16 : INTISRxCLR_DSP
Monitor register of the factor of the interrupt signal x of Cortex-M3 Core0 and Cortex-M3 Core1
0 : Interrupt signal x of Cortex-M3 Core0 and Cortex-M3 Core1 is "L" levels.
1 : Interrupt signal x of Cortex-M3 Core0 and Cortex-M3 Core1 is "H" levels.

Note) x is one from 0 to 3.

Bit11-Bit8 : INTISR1CLR_x
Monitor register of the factor of the interrupt signal x of LPDSP32 and Cortex-M3 Core0
0 : Interrupt signal x of LPDSP32 and Cortex-M3 Core0 is "L" levels.
1 : Interrupt signal x of LPDSP32 and Cortex-M3 Core0 is "H" levels.

Note) x is one from 0 to 3.

Bit3-Bit0 : INTISR0CLR_x
Monitor register of the factor of the interrupt signal x of LPDSP32 and Cortex-M3 Core1
0 : Interrupt signal x of LPDSP32 and Cortex-M3 Core1 is "L" levels.
1 : Interrupt signal x of LPDSP32 and Cortex-M3 Core1 is "H" levels.

Note) x is one from 0 to 3.

4-3-3-3 Watch Dog Timer interrupt setting register: WTMINTSET

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	WTMINT1 SET2	WTMINT1 SET1	WTMINT1 SET0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	WTMINT0 SET2	WTMINT0 SET1	WTMINT0 SET0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit10-8 : WTMINT1SETx

Allocation register which set the interrupt of WTMx(Watch Dog Timer x) in normal interrupt or non-maskable interrupt of Cortex-M3 Core1

0 :It allocates it in normal interrupt.

1 :It allocates it in non-maskable interrupt (NMI).

Note) x is one from 0 to 2.

* When the Watch Dog Timer interrupt use as interrupt of LPDSP32, this setting doesn't influence.

Bit2-0 : WTMINT0SETx

Allocation register which set the interrupt of WTMx(Watch Dog Timer x) in normal interrupt or non-maskable interrupt of Cortex-M3 Core0

0 :It allocates it in usual interrupt.

1 :It allocates it in [nonmasukaraburu] interrupt (NMI).

Note) x is one from 0 to 2.

* When the Watch Dog Timer interrupt use as interrupt of LPDSP32, this setting doesn't influence.

4-3-3-4 External interrupt n(EXTINTn) monitor register: EXTINTn

[Base + 0x0400 - 0x0410]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ STATE15	EXTINTn_ STATE14	EXTINTn_ STATE13	EXTINTn_ STATE12	EXTINTn_ STATE11	EXTINTn_ STATE10	EXTINTn_ STATE9	EXTINTn_ STATE8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ STATE7	EXTINTn_ STATE6	EXTINTn_ STATE5	EXTINTn_ STATE4	EXTINTn_ STATE3	EXTINTn_ STATE2	EXTINTn_ STATE1	EXTINTn_ STATE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit15-0 : EXTINTnSTATE15-0
Monitor register of the state of external interrupt n
0: Interrupt has not been generated.
1: Interrupt has been generated.

Note) n = 0-4

4-3-3-5 External interrupt 5(EXTINT5) monitor register: EXTINT5

[Base + 0x0414]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINT5_ STATE9	EXTINT5_ STATE8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	EXTINT5_ STATE7	EXTINT5_ STATE6	EXTINT5_ STATE5	EXTINT5_ STATE4	EXTINT5_ STATE3	EXTINT5_ STATE2	EXTINT5_ STATE1	EXTINT5_ STATE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : EXTINT5STATE9-0

Monitor register of the state of external interrupt 5.

0: Interrupt has not been generated.

1: Interrupt has been generated.

4-3-3-6 Raw external interrupt n(EXTINTn) monitor register: EXTINTnS

[Base + 0x0418 - 0x0428]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ SRC15	EXTINTn_ SRC14	EXTINTn_ SRC13	EXTINTn_ SRC12	EXTINTn_ SRC11	EXTINTn_ SRC10	EXTINTn_ SRC9	EXTINTn_ SRC8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ SRC7	EXTINTn_ SRC6	EXTINTn_ SRC5	EXTINTn_ SRC4	EXTINTn_ SRC3	EXTINTn_ SRC2	EXTINTn_ SRC1	EXTINTn_ SRC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit15-0 : EXTINTnSRC15-0

Monitor register of the raw state of external interrupt n

0: Interrupt has not been generated.

1: Interrupt has been generated.

Note) n = 0-4

4-3-3-7 Raw external interrupt 5(EXTINT5) monitor register: EXTINT5S

[Base + 0x042C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINT5_ SRC9	EXTINT5_ SRC8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	EXTINT5_ SRC7	EXTINT5_ SRC6	EXTINT5_ SRC5	EXTINT5_ SRC4	EXTINT5_ SRC3	EXTINT5_ SRC2	EXTINT5_ SRC1	EXTINT5_ SRC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : EXTINT5SRC9-0

Monitor register of the raw state of external interrupt 5.

0: Interrupt has not been generated.

1: Interrupt has been generated.

4-3-3-8 External interrupt n(EXTINTn) mask register: EXTINTnM

[Base + 0x0430 - 0x0440]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ MSK15	EXTINTn_ MSK14	EXTINTn_ MSK13	EXTINTn_ MSK12	EXTINTn_ MSK11	EXTINTn_ MSK10	EXTINTn_ MSK9	EXTINTn_ MSK8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ MSK7	EXTINTn_ MSK6	EXTINTn_ MSK5	EXTINTn_ MSK4	EXTINTn_ MSK3	EXTINTn_ MSK2	EXTINTn_ MSK1	EXTINTn_ MSK0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : EXTINTnMSK15-0
Select register whether to mask the external interrupt n
0: The external interrupt is not masked.
1: The external interrupt is masked.

Note) n = 0-4

4-3-3-9 External interrupt 5(EXTINT5) mask register: EXTINT5M

[Base + 0x0444]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINTn_ MSK9	EXTINTn_ MSK8
Reset	0	0	0	0	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ MSK7	EXTINTn_ MSK6	EXTINTn_ MSK5	EXTINTn_ MSK4	EXTINTn_ MSK3	EXTINTn_ MSK2	EXTINTn_ MSK1	EXTINTn_ MSK0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0 : EXTINT5MSK9-0

Select register whether to mask the external interrupt n

0: The external interrupt is not masked.

1: The external interrupt is masked.

4-3-3-10 External interrupt n(EXTINTn) setting register: EXTINTnC0

[Base + 0x0448 - 0x0458]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ SET15	EXTINTn_ SET14	EXTINTn_ SET13	EXTINTn_ SET12	EXTINTn_ SET11	EXTINTn_ SET10	EXTINTn_ SET9	EXTINTn_ SET8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ SET7	EXTINTn_ SET6	EXTINTn_ SET5	EXTINTn_ SET4	EXTINTn_ SET3	EXTINTn_ SET2	EXTINTn_ SET1	EXTINTn_ SET0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : EXTINTnSET15-0
Interrupt setting register external interrupt n to the interrupt of Cortex-M3 Core0
0: It allocates it in normal interrupts.
1: It allocates it in non-maskable interrupt (NMI).

* When the external interrupt use as interrupt of LPDSP32, this setting doesn't influence.

Note) n = 0-4

4-3-3-11 External interrupt 5(EXTINT5) setting register: EXTINT5C0

[Base + 0x045C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINTn_ SET9	EXTINTn_ SET8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ SET7	EXTINTn_ SET6	EXTINTn_ SET5	EXTINTn_ SET4	EXTINTn_ SET3	EXTINTn_ SET2	EXTINTn_ SET1	EXTINTn_ SET0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0 : EXTINTnSET9-0

Intterupt setting register of external interrupt 5 to the interrupt of Cortex-M3 Core0

0:It allocates it in normal interrupts.

1:It allocates it in non-maskable interrupt (NMI).

* When the external interrupt use as interrupt of LPDSP32, this setting doesn't influence.

4-3-3-12 External interrupt n(EXTINTn) setting register: EXTINTnC1

[Base + 0x0460 - 0x0470]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ SET15	EXTINTn_ SET14	EXTINTn_ SET13	EXTINTn_ SET12	EXTINTn_ SET11	EXTINTn_ SET10	EXTINTn_ SET9	EXTINTn_ SET8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ SET7	EXTINTn_ SET6	EXTINTn_ SET5	EXTINTn_ SET4	EXTINTn_ SET3	EXTINTn_ SET2	EXTINTn_ SET1	EXTINTn_ SET0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : EXTINTnSET15-0
Intterupt setting register of external interrupt n to the interrupt of Cortex-M3 Core1
0: It allocates it in normal interrupts.
1: It allocates it in non-maskable interrupt (NMI).

* When the external interrupt use as interrupt of LPDSP32, this setting doesn't influence.

Note) n = 0-4

4-3-3-13 External interrupt 5(EXTINT5) setting register: EXTINT5C1

[Base + 0x0474]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINTn_ SET9	EXTINTn_ SET8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ SET7	EXTINTn_ SET6	EXTINTn_ SET5	EXTINTn_ SET4	EXTINTn_ SET3	EXTINTn_ SET2	EXTINTn_ SET1	EXTINTn_ SET0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0 : EXTINTnSET9-0

Intterupt setting register of external interrupt 5 to the interrupt of Cortex-M3 Core1

0:It allocates it in normal interrupts.

1:It allocates it in non-maskable interrupt (NMI).

* When the external interrupt use as interrupt of LPDSP32, this setting doesn't influence.

4-3-3-14 External interrupt n(EXTINTn) condition setting register: EXTINTnCND

[Base + 0x0478 - 0x0488]

Bit	31	30	29	28	27	26	25	24
Name	EXTINTn_CNDf1	EXTINTn_CNDf0	EXTINTn_CNDe1	EXTINTn_CNDe0	EXTINTn_CNDd1	EXTINTn_CNDd0	EXTINTn_CNDc1	EXTINTn_CNDc0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	EXTINTn_CNDb1	EXTINTn_CNDb0	EXTINTn_CNDa1	EXTINTn_CNDa0	EXTINTn_CND91	EXTINTn_CND90	EXTINTn_CND81	EXTINTn_CND80
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_CND71	EXTINTn_CND70	EXTINTn_CND61	EXTINTn_CND60	EXTINTn_CND51	EXTINTn_CND50	EXTINTn_CND41	EXTINTn_CND40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_CND31	EXTINTn_CND30	EXTINTn_CND21	EXTINTn_CND20	EXTINTn_CND11	EXTINTn_CND10	EXTINTn_CND01	EXTINTn_CND00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : EXTINTnCNDm1-m0
Intterupt condition setting register of external interrupt n bit m
00 b: Interrupt condition is a falling edge occurs.
01 b: Interrupt condition is a low level occurs.
10 b: Interrupt condition is a rising edge occurs.
11 b: Interrupt condition is a high level occurs.

Note) n = 0-4

4-3-3-15 External interrupt 5(EXTINT5) condition setting register: EXTINT5CND

[Base + 0x048C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	EXTINTn_ CND91	EXTINTn_ CND90	EXTINTn_ CND81	EXTINTn_ CND80
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ CND71	EXTINTn_ CND70	EXTINTn_ CND61	EXTINTn_ CND60	EXTINTn_ CND51	EXTINTn_ CND50	EXTINTn_ CND41	EXTINTn_ CND40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ CND31	EXTINTn_ CND30	EXTINTn_ CND21	EXTINTn_ CND20	EXTINTn_ CND11	EXTINTn_ CND10	EXTINTn_ CND01	EXTINTn_ CND00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit19-0 : EXTINT5CNDm1-m0
Intterupt condition setting register of external interrupt 5 bit m
00 b: Interrupt condition is a falling edge occurs.
01 b: Interrupt condition is a low level occurs.
10 b: Interrupt condition is a rising edge occurs.
11 b: Interrupt condition is a high level occurs.

4-3-3-16 External interrupt n(EXTINTn) clear register: EXTINTnCLR

[Base + 0x0490 - 0x04A0]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ CLR15	EXTINTn_ CLR14	EXTINTn_ CLR13	EXTINTn_ CLR12	EXTINTn_ CLR11	EXTINTn_ CLR10	EXTINTn_ CLR9	EXTINTn_ CLR8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ CLR7	EXTINTn_ CLR6	EXTINTn_ CLR5	EXTINTn_ CLR4	EXTINTn_ CLR3	EXTINTn_ CLR2	EXTINTn_ CLR1	EXTINTn_ CLR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : EXTINTnCLR15-0

Clear register of the factor of the external interrupt n bit 15-0

The factor of interrupt is cleared when "1" is written, and hardware returns 0 this register automatically.

0: There is no effect.

1: The factor of interrupt is cleared.

Note) n = 0-4

4-3-3-17 External interrupt 5(EXTINT5) clear register: EXTINT5CLR

[Base + 0x04A4]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINT5_ CLR9	EXTINT5_ CLR8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINT5_ CLR7	EXTINT5_ CLR6	EXTINT5_ CLR5	EXTINT5_ CLR4	EXTINT5_ CLR3	EXTINT5_ CLR2	EXTINT5_ CLR1	EXTINT5_ CLR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0 : EXTINT5CLR9-0

Clear register of the factor of the external interrupt 5 bit 9-0

The factor of interrupt is cleared when "1" is written, and hardware returns 0 this register automatically.

0: There is no effect.

1: The factor of interrupt is cleared.

4-3-3-18 External interrupt n(EXTINTn) noise filter permission register: EXTINTnFEN

[Base + 0x04A8 - 0x04B8]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ FEN15	EXTINTn_ FEN14	EXTINTn_ FEN13	EXTINTn_ FEN12	EXTINTn_ FEN11	EXTINTn_ FEN10	EXTINTn_ FEN9	EXTINTn_ FEN8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ FEN7	EXTINTn_ FEN6	EXTINTn_ FEN5	EXTINTn_ FEN4	EXTINTn_ FEN3	EXTINTn_ FEN2	EXTINTn_ FEN1	EXTINTn_ FEN0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : EXTINTnFEN15-0
Seting register whether the noise filter is used for each bit of external interrupt n
0:The noise filter is not used.
1:The noise filter is used.

The relation between interrupt when the noise filter is not used and the pulse width of an external input is shown below.

When you set the interrupt condition to the level interrupt.

If the width of the external interrupt input is more than 2*T, interrupt occurs.

When you set the interrupt condition to the edge interrupt.

If the both width before the change and after the change is more than 2*T, interrupt occurs.

If you need to know pulse width necessary to occur an interrupt when the noise filter is used, please refer to the following chapter.

4-3-3-20External interrupt n(EXTINTn) noise filter setting register:
EXTINTnFSET

Note) T = clock cycle (=1/ System Clock [Hz])

Note) n = 0-4

4-3-3-19 External interrupt 5(EXTINT5) noise filter permission register: EXTINT5FEN

[Base + 0x04BC]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	EXTINTn_ FEN9	EXTINTn_ FEN8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ FEN7	EXTINTn_ FEN6	EXTINTn_ FEN5	EXTINTn_ FEN4	EXTINTn_ FEN3	EXTINTn_ FEN2	EXTINTn_ FEN1	EXTINTn_ FEN0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0 : EXTINT5FEN9-0
Seting register whether the noise filter is used for each bit of external interrupt 5
0: The noise filter is not used.
1: The noise filter is used.

The relation between interrupt occurence when the noise filter is not used and the pulse width of an external input is shown below.

When you set the interrupt condition to the level interrupt.

If the width of the external interrupt input is more than 2*T, interrupt occurs.

When you set the interrupt condition to the edge interrupt.

If the both width before the change and after the change is more than 2*T, interrupt occurs.

If you need to know pulse width necessary to occur an interrupt when the noise filter is used, please refer to the following chapter.

4-3-3-21External interrupt 5(EXTINT5) noise filter setting register: EXTINT5FSET

Note) T = clock cycle (=1/ System Clock [Hz])

4-3-3-20 External interrupt n(EXTINTn) noise filter setting register: EXTINTnFSET

[Base + 0x04C0 - 0x04D0]

Bit	31	30	29	28	27	26	25	24
Name	EXTINTn_FSETf1	EXTINTn_FSETf0	EXTINTn_FSETe1	EXTINTn_FSETe0	EXTINTn_FSETd1	EXTINTn_FSETd0	EXTINTn_FSETc1	EXTINTn_FSETc0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	EXTINTn_FSETb1	EXTINTn_FSETb0	EXTINTn_FSETa1	EXTINTn_FSETa0	EXTINTn_FSET91	EXTINTn_FSET90	EXTINTn_FSET81	EXTINTn_FSET80
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_FSET71	EXTINTn_FSET70	EXTINTn_FSET61	EXTINTn_FSET60	EXTINTn_FSET51	EXTINTn_FSET50	EXTINTn_FSET41	EXTINTn_FSET40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_FSET31	EXTINTn_FSET30	EXTINTn_FSET21	EXTINTn_FSET20	EXTINTn_FSET11	EXTINTn_FSET10	EXTINTn_FSET01	EXTINTn_FSET00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : EXTINTnFSETm1-m0
Setting register of the noise removal ability of the noise filter for external interrupt n bit m
00 b: N = 128
01 b: N = 512
10 b: N = 1024
11 b: N = 2048

Note) N : Noise removal ability

The relation between interrupt occurence when the noise filter is used and the pulse width of an external input is shown below.

When you set the interrupt condition to the level interrupt.

If the width of the external interrupt input is less than N*T, interrupt dosen't occur.

If the width of the external interrupt input is more than (N+1)*T and less than (2*N-1)*T, interrupt might occur. (Depend on the state of the free run counter in the noise filter).

If the width of the external interrupt input is more than (2*N)*T, interrupt occurs.

When you set the interrupt condition to the edge interrupt.

If the width before the change or after the change is less than N*T, interrupt dosen't occur.

If the both width before the change and after the change is more than (2*N)*T, interrupt occur.

If the width before the change is more than (N+1)*T and it after the change is less than (2*N-1)*T, interrupt might occur. (Depend on the state of the free run counter in the noise filter).

If the width after the change is more than (N+1)*T and it before the change is less than (2*N-1)*T, interrupt might occur. (Depend on the state of the free run counter in the noise filter).

Note) T: Clock cycle (= 1/ System Clock [Hz])

Note) n = 0-4

4-3-3-21 External interrupt 5(EXTINT5) noise filter setting register: EXTINT5FSET

[Base + 0x04D4]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	EXTINTn_ FSET91	EXTINTn_ FSET90	EXTINTn_ FSET81	EXTINTn_ FSET80
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	EXTINTn_ FSET71	EXTINTn_ FSET70	EXTINTn_ FSET61	EXTINTn_ FSET60	EXTINTn_ FSET51	EXTINTn_ FSET50	EXTINTn_ FSET41	EXTINTn_ FSET40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EXTINTn_ FSET31	EXTINTn_ FSET30	EXTINTn_ FSET21	EXTINTn_ FSET20	EXTINTn_ FSET11	EXTINTn_ FSET10	EXTINTn_ FSET01	EXTINTn_ FSET00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : EXTINT5FSETm1-m0
Setting register of the noise removal ability of the noise filter for external interrupt 5 bit m
00 b: N = 128
01 b: N = 512
10 b: N = 1024
11 b: N = 2048

Note) N : Noise removal ability

The relation between interrupt occurence when the noise filter is used and the pulse width of an external input is shown below.

When you set the interrupt condition to the level interrupt.

If the width of the external interrupt input is less than N*T, interrupt dosen't occur.

If the width of the external interrupt input is more than (N+1)*T and less than (2*N-1)*T, interrupt might occur. (Depend on the state of the free run counter in the noise filter).

If the width of the external interrupt input is more than (2*N)*T, interrupt occurs.

When you set the interrupt condition to the edge interrupt.

If the width before the change or after the change is less than N*T, interrupt dosen't occur.

If the both width before the change and after the change is more than (2*N)*T, interrupt occur.

If the width before the change is more than (N+1)*T and it after the change is less than (2*N-1)*T, interrupt might occur. (Depend on the state of the free run counter in the noise filter).

If the width after the change is more than (N+1)*T and it before the change is less than (2*N-1)*T, interrupt might occur. (Depend on the state of the free run counter in the noise filter).

Note) T: Clock cycle (= 1/ System Clock [Hz])

4-3-3-22 LPDSP32 interrupt n status register: DSPINTnSTS

[Base + 0x0800 - 0x0834]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	DSPINTn STATE
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit0 : DSPINTnSTATE
Monitor register of the state of LPDSP32 interrupt n
0: Interrupt has not been generated.
1: Interrupt has been generated.

Note) n = 1-14

4-3-3-23 15 LPDSP32 interrupt status registers: DSPINT15STS0

[Base + 0x0838]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT STATE31	DSPINT STATE30	DSPINT STATE29	DSPINT STATE28	DSPINT STATE27	DSPINT STATE26	DSPINT STATE25	DSPINT STATE24
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	DSPINT STATE23	DSPINT STATE22	DSPINT STATE21	DSPINT STATE20	DSPINT STATE19	DSPINT STATE18	DSPINT STATE17	DSPINT STATE16
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	DSPINT STATE15	DSPINT STATE14	DSPINT STATE13	DSPINT STATE12	DSPINT STATE11	DSPINT STATE10	DSPINT STATE9	DSPINT STATE8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPINT STATE7	DSPINT STATE6	DSPINT STATE5	DSPINT STATE4	DSPINT STATE3	DSPINT STATE2	DSPINT STATE1	DSPINT STATE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-0 : DSPINTSTATE31-0
Monitor register of the state of Bit31-0 of LPDSP32 interrupt 15
0: Interrupt has not been generated
1: Interrupt has been generated.

* The interrupt number for LPDSP32 is the Bit number of this register.

4-3-3-24 LPDSP32 interrupt 15 status registers: DSPINT15STS1

[Base + 0x083C]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT STATE63	DSPINT STATE62	DSPINT STATE61	DSPINT STATE60	DSPINT STATE59	DSPINT STATE58	DSPINT STATE57	DSPINT STATE56
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	DSPINT STATE55	DSPINT STATE54	DSPINT STATE53	DSPINT STATE52	DSPINT STATE51	DSPINT STATE50	DSPINT STATE49	DSPINT STATE48
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	DSPINT STATE47	DSPINT STATE46	DSPINT STATE45	DSPINT STATE44	DSPINT STATE43	DSPINT STATE42	DSPINT STATE41	DSPINT STATE40
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPINT STATE39	DSPINT STATE38	DSPINT STATE37	DSPINT STATE36	DSPINT STATE35	DSPINT STATE34	DSPINT STATE33	DSPINT STATE32
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-0 : DSPINTSTATE63-32
Monitor register of the state of Bit63-32 of LPDSP32 interrupt 15
0: Interrupt has not been generated
1: Interrupt has been generated.

* The interrupt number for LPDSP32 is the Bit number of this register.

4-3-3-25 LPDSP32 interrupt mask register: DSPINTMASK0

[Base + 0x0840]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT MSK 31	DSPINT MSK 30	DSPINT MSK 29	DSPINT MSK 28	DSPINT MSK 27	DSPINT MSK 26	DSPINT MSK 25	DSPINT MSK 24
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT MSK 23	DSPINT MSK 22	DSPINT MSK 21	DSPINT MSK 20	DSPINT MSK 19	DSPINT MSK 18	DSPINT MSK 17	DSPINT MSK 16
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT MSK 15	DSPINT MSK 14	DSPINT MSK 13	DSPINT MSK 12	DSPINT MSK 11	DSPINT MSK 10	DSPINT MSK 9	DSPINT MSK 8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT MSK 7	DSPINT MSK 6	DSPINT MSK 5	DSPINT MSK 4	DSPINT MSK 3	DSPINT MSK 2	DSPINT MSK 1	DSPINT MSK 0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTMSK31-0
Select register whether to mask the LPDSP32 interrupt
0: The interrupt of LPDSP32 is not masked.
1: The interrupt of LPDSP32 is masked.

4-3-3-26 LPDSP32 interrupt mask register: DSPINTMASK1

[Base + 0x0844]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT MSK 63	DSPINT MSK 62	DSPINT MSK 61	DSPINT MSK 60	DSPINT MSK 59	DSPINT MSK 58	DSPINT MSK 57	DSPINT MSK 56
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT MSK 55	DSPINT MSK 54	DSPINT MSK 53	DSPINT MSK 52	DSPINT MSK 51	DSPINT MSK 50	DSPINT MSK 49	DSPINT MSK 48
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT MSK 47	DSPINT MSK 46	DSPINT MSK 45	DSPINT MSK 44	DSPINT MSK 43	DSPINT MSK 42	DSPINT MSK 41	DSPINT MSK 40
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT MSK 39	DSPINT MSK 38	DSPINT MSK 37	DSPINT MSK 36	DSPINT MSK 35	DSPINT MSK 34	DSPINT MSK 33	DSPINT MSK 32
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTMSK63-32
Select register whether to mask the LPDSP32 interrupt
0: The interrupt of LPDSP32 is not masked.
1: The interrupt of LPDSP32 is masked.

4-3-3-27 LPDSP32 interrupt source status register: DSPINTSRCSTS0

[Base + 0x0848]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT SRC 31	DSPINT SRC 30	DSPINT SRC 29	DSPINT SRC 28	DSPINT SRC 27	DSPINT SRC 26	DSPINT SRC 25	DSPINT SRC 24
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	DSPINT SRC 23	DSPINT SRC 22	DSPINT SRC 21	DSPINT SRC 20	DSPINT SRC 19	DSPINT SRC 18	DSPINT SRC 17	DSPINT SRC 16
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	DSPINT SRC 15	DSPINT SRC 14	DSPINT SRC 13	DSPINT SRC 12	DSPINT SRC 11	DSPINT SRC 10	DSPINT SRC 9	DSPINT SRC 8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPINT SRC 7	DSPINT SRC 6	DSPINT SRC 5	DSPINT SRC 4	DSPINT SRC 3	DSPINT SRC 2	DSPINT SRC 1	DSPINT SRC 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-0 : DSPINTSRC31-0
Monitor register of the raw interrupt status of LPDSP32
0: The raw interrupt of LPDSP32 has not been generated
1: The raw interrupt of LPDSP32 has been generated

4-3-3-28 LPDSP32 interrupt source status register: DSPINTSRCSTS1

[Base + 0x084C]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT SRC 63	DSPINT SRC 62	DSPINT SRC 61	DSPINT SRC 60	DSPINT SRC 59	DSPINT SRC 58	DSPINT SRC 57	DSPINT SRC 56
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	DSPINT SRC 55	DSPINT SRC 54	DSPINT SRC 53	DSPINT SRC 52	DSPINT SRC 51	DSPINT SRC 50	DSPINT SRC 49	DSPINT SRC 48
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	DSPINT SRC 47	DSPINT SRC 46	DSPINT SRC 45	DSPINT SRC 44	DSPINT SRC 43	DSPINT SRC 42	DSPINT SRC 41	DSPINT SRC 40
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPINT SRC 39	DSPINT SRC 38	DSPINT SRC 37	DSPINT SRC 36	DSPINT SRC 35	DSPINT SRC 34	DSPINT SRC 33	DSPINT SRC 32
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-0 : DSPINTSRC63-32
Monitor register of the raw interrupt status of LPDSP32
0: The raw interrupt of LPDSP32 has not been generated
1: The raw interrupt of LPDSP32 has been generated

4-3-3-29 LPDSP32 interrupt setting register 0: DSPINTSET0

[Base + 0x0850]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT7 SET3	DSPINT7 SET2	DSPINT7 SET1	DSPINT7 SET0	DSPINT6 SET3	DSPINT6 SET2	DSPINT6 SET1	DSPINT6 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT5 SET3	DSPINT5 SET2	DSPINT5 SET1	DSPINT5 SET0	DSPINT4 SET3	DSPINT4 SET2	DSPINT4 SET1	DSPINT4 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT3 SET3	DSPINT3 SET2	DSPINT3 SET1	DSPINT3 SET0	DSPINT2 SET3	DSPINT2 SET2	DSPINT2 SET1	DSPINT2 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT1 SET3	DSPINT1 SET2	DSPINT1 SET1	DSPINT1 SET0	DSPINT0 SET3	DSPINT0 SET2	DSPINT0 SET1	DSPINT0 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 0-7

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-30 LPDSP32 interrupt setting register 1: DSPINTSET1

[Base + 0x0854]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT15 SET3	DSPINT15 SET2	DSPINT15 SET1	DSPINT15 SET0	DSPINT14 SET3	DSPINT14 SET2	DSPINT14 SET1	DSPINT14 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT13 SET3	DSPINT13 SET2	DSPINT13 SET1	DSPINT13 SET0	DSPINT12 SET3	DSPINT12 SET2	DSPINT12 SET1	DSPINT12 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT11 SET3	DSPINT11 SET2	DSPINT11 SET1	DSPINT11 SET0	DSPINT10 SET3	DSPINT10 SET2	DSPINT10 SET1	DSPINT10 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT9 SET3	DSPINT9 SET2	DSPINT9 SET1	DSPINT9 SET0	DSPINT8 SET3	DSPINT8 SET2	DSPINT8 SET1	DSPINT8 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 8-15

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-31 LPDSP32 interrupt setting register 2: DSPINTSET2

[Base + 0x0858]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT23 SET3	DSPINT23 SET2	DSPINT23 SET1	DSPINT23 SET0	DSPINT22 SET3	DSPINT22 SET2	DSPINT22 SET1	DSPINT22 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT21 SET3	DSPINT21 SET2	DSPINT21 SET1	DSPINT21 SET0	DSPINT20 SET3	DSPINT20 SET2	DSPINT20 SET1	DSPINT20 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT19 SET3	DSPINT19 SET2	DSPINT19 SET1	DSPINT19 SET0	DSPINT18 SET3	DSPINT18 SET2	DSPINT18 SET1	DSPINT18 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT17 SET3	DSPINT17 SET2	DSPINT17 SET1	DSPINT17 SET0	DSPINT16 SET3	DSPINT16 SET2	DSPINT16 SET1	DSPINT16 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 16-23

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-32 LPDSP32 interrupt setting register 3: DSPINTSET3

[Base + 0x085C]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT31 SET3	DSPINT31 SET2	DSPINT31 SET1	DSPINT31 SET0	DSPINT30 SET3	DSPINT30 SET2	DSPINT30 SET1	DSPINT30 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT29 SET3	DSPINT29 SET2	DSPINT29 SET1	DSPINT29 SET0	DSPINT28 SET3	DSPINT28 SET2	DSPINT28 SET1	DSPINT28 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT27 SET3	DSPINT27 SET2	DSPINT27 SET1	DSPINT27 SET0	DSPINT26 SET3	DSPINT26 SET2	DSPINT26 SET1	DSPINT26 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT25 SET3	DSPINT25 SET2	DSPINT25 SET1	DSPINT25 SET0	DSPINT24 SET3	DSPINT24 SET2	DSPINT24 SET1	DSPINT24 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 24-31

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-33 LPDSP32 interrupt setting register 4: DSPINTSET4

[Base + 0x0860]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT39 SET3	DSPINT39 SET2	DSPINT39 SET1	DSPINT39 SET0	DSPINT38 SET3	DSPINT38 SET2	DSPINT38 SET1	DSPINT38 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT37 SET3	DSPINT37 SET2	DSPINT37 SET1	DSPINT37 SET0	DSPINT36 SET3	DSPINT36 SET2	DSPINT36 SET1	DSPINT36 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT35 SET3	DSPINT35 SET2	DSPINT35 SET1	DSPINT35 SET0	DSPINT34 SET3	DSPINT34 SET2	DSPINT34 SET1	DSPINT34 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT33 SET3	DSPINT33 SET2	DSPINT33 SET1	DSPINT33 SET0	DSPINT32 SET3	DSPINT32 SET2	DSPINT32 SET1	DSPINT32 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 32-39

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-34 LPDSP32 interrupt setting register 5: DSPINTSET5

[Base + 0x0864]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT47 SET3	DSPINT47 SET2	DSPINT47 SET1	DSPINT47 SET0	DSPINT46 SET3	DSPINT46 SET2	DSPINT46 SET1	DSPINT46 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT45 SET3	DSPINT45 SET2	DSPINT45 SET1	DSPINT45 SET0	DSPINT44 SET3	DSPINT44 SET2	DSPINT44 SET1	DSPINT44 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT43 SET3	DSPINT43 SET2	DSPINT43 SET1	DSPINT43 SET0	DSPINT42 SET3	DSPINT42 SET2	DSPINT42 SET1	DSPINT42 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT41 SET3	DSPINT41 SET2	DSPINT41 SET1	DSPINT41 SET0	DSPINT40 SET3	DSPINT40 SET2	DSPINT40 SET1	DSPINT40 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 40-47

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-35 LPDSP32 interrupt setting register 6: DSPINTSET6

[Base + 0x0868]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT55 SET3	DSPINT55 SET2	DSPINT55 SET1	DSPINT55 SET0	DSPINT54 SET3	DSPINT54 SET2	DSPINT54 SET1	DSPINT54 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT53 SET3	DSPINT53 SET2	DSPINT53 SET1	DSPINT53 SET0	DSPINT52 SET3	DSPINT52 SET2	DSPINT52 SET1	DSPINT52 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT51 SET3	DSPINT51 SET2	DSPINT51 SET1	DSPINT51 SET0	DSPINT50 SET3	DSPINT50 SET2	DSPINT50 SET1	DSPINT50 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT49 SET3	DSPINT49 SET2	DSPINT49 SET1	DSPINT49 SET0	DSPINT48 SET3	DSPINT48 SET2	DSPINT48 SET1	DSPINT48 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 48-55

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-36 LPDSP32 interrupt setting register 7: DSPINTSET7

[Base + 0x086C]

Bit	31	30	29	28	27	26	25	24
Name	DSPINT63 SET3	DSPINT63 SET2	DSPINT63 SET1	DSPINT63 SET0	DSPINT62 SET3	DSPINT62 SET2	DSPINT62 SET1	DSPINT62 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSPINT61 SET3	DSPINT61 SET2	DSPINT61 SET1	DSPINT61 SET0	DSPINT60 SET3	DSPINT60 SET2	DSPINT60 SET1	DSPINT60 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSPINT59 SET3	DSPINT59 SET2	DSPINT59 SET1	DSPINT59 SET0	DSPINT58 SET3	DSPINT58 SET2	DSPINT58 SET1	DSPINT58 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPINT57 SET3	DSPINT57 SET2	DSPINT57 SET1	DSPINT57 SET0	DSPINT56 SET3	DSPINT56 SET2	DSPINT56 SET1	DSPINT56 SET0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : DSPINTnSET3-0
Setting register of the interrupt factor n to LPDSP32
0: Prohibition setting.
1: Factor of interrupt n is allocated interrupt 1.
2: Factor of interrupt n is allocated interrupt 2.
3: Factor of interrupt n is allocated interrupt 3.
4: Factor of interrupt n is allocated interrupt 4.
5: Factor of interrupt n is allocated interrupt 5.
6: Factor of interrupt n is allocated interrupt 6.
7: Factor of interrupt n is allocated interrupt 7.
8: Factor of interrupt n is allocated interrupt 8.
9: Factor of interrupt n is allocated interrupt 9.
10: Factor of interrupt n is allocated interrupt 10.
11: Factor of interrupt n is allocated interrupt 11.
12: Factor of interrupt n is allocated interrupt 12.
13: Factor of interrupt n is allocated interrupt 13.
14: Factor of interrupt n is allocated interrupt 14.
15: Factor of interrupt n is allocated interrupt 15.

Note) n = 56-63

Attention) Interrupt 1-14 of LPDSP32 can be allocated only one factor. If two or more interrupt factors are allocated to one of the interrupt of LPDSP32, the smallest number(n) interrupt is allocated and other number(n) interrupts can not be detected as interrupt. Therefor please take care about this register setting.

4-3-3-37 LPDSP32 DMA status clear register: DSPDMACLR

[Base + 0x0870]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	DSPDMA CLR8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPDMA CLR7	DSPDMA CLR6	DSPDMA CLR5	DSPDMA CLR4	DSPDMA CLR3	DSPDMA CLR2	DSPDMA CLR1	DSPDMA CLR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit n : DSPDMACLR n

Clear register of the factor of interrupt DSPDMASTSn

The factor of interrupt DSPDMASTSn is cleared when "1" is written, and hardware returns 0 this register automatically.

0: There is no effect.

1: The factor of interrupt DSPDMASTSn is cleared.

Note) n = 0-8

4-3-3-38 LPDSP32 DMB status clear register: DSPDMBCLR

[Base + 0x0874]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	DSPDMB CLR8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPDMB CLR7	DSPDMB CLR6	DSPDMB CLR5	DSPDMB CLR4	DSPDMB CLR3	DSPDMB CLR2	DSPDMB CLR1	DSPDMB CLR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit n : DSPDMBCLR n

Clear register of the factor of interrupt DSPDMBSTSn

The factor of interrupt DSPDMBSTSn is cleared when "1" is written, and hardware returns 0 this register automatically.

0: There is no effect.

1: The factor of interrupt DSPDMBSTSn is cleared.

Note) n = 0-8

4-3-3-39 LPDSP32 PM status clear register: DSPPMCLR

[Base + 0x0878]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	DSPPMCLR9	DSPPMCLR8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSPPMCLR7	DSPPMCLR6	DSPPMCLR5	DSPPMCLR4	DSPPMCLR3	DSPPMCLR2	DSPPMCLR1	DSPPMCLR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bitn : DSPPMCLR n

Clear register of the factor of interrupt DSPPMSTSn

The factor of interrupt DSPPMSTSn is cleared when "1" is written, and hardware returns 0 this register automatically.

0: There is no effect.

1: The factor of interrupt DSPPMSTSn is cleared.

Note) n = 0-9

4-3-3-40 LPDSP32 DMA status register: DSPDMASTS

[Base + 0x087C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	DSPDMA STS8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPDMA STS7	DSPDMA STS6	DSPDMA STS5	DSPDMA STS4	DSPDMA STS3	DSPDMA STS2	DSPDMA STS1	DSPDMA STS0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn : DSPDMASTS n

When LPDSP32 accesses Segment(n) with DMA though it is prohibited that LPDSP32 accesses Segment(n) with DMA, this register is set "1". Moreover this register is a factor of a interrupt of LPDSP32.

0: The access has not been detected with the access prohibited.

1: The access has be detected with the access prohibited.

Note) n = 0-8

Remarks) The register that permits the access to Segment(n) for LPDSP32 DMA exists in System Controller.

4-3-3-41 LPDSP32 DMB status register: DSPDMBSTS

[Base + 0x0880]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	DSPDMB STS8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPDMB STS7	DSPDMB STS6	DSPDMB STS5	DSPDMB STS4	DSPDMB STS3	DSPDMB STS2	DSPDMB STS1	DSPDMB STS0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn : DSPDMBSTS n

When LPDSP32 accesses Segment(n) with DMB though it is prohibited that LPDSP32 accesses Segment(n) with DMB, this register is set "1". Moreover this register is a factor of a interrupt of LPDSP32.

0: The access has not been detected with the access prohibited.

1: The access has be detected with the access prohibited.

Note) n = 0-8

Remarks) The register that permits the access to Segment(n) for LPDSP32 DMB exists in System Controller.

4-3-3-42 LPDSP32 PM status register: DSPPMSTS

[Base + 0x0884]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	DSPPMSTS9	DSPPMSTS8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSPPMSTS7	DSPPMSTS6	DSPPMSTS5	DSPPMSTS4	DSPPMSTS3	DSPPMSTS2	DSPPMSTS1	DSPPMSTS0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn : DSPPMSTS n

When LPDSP32 accesses Segment(n) with PM though it is prohibited that LPDSP32 accesses Segment(n) with PM, this register is set "1". Moreover this register is a factor of a interrupt of LPDSP32.

0: The access has not been detected with the access prohibited.

1: The access has be detected with the access prohibited.

Note) n = 0-9

Remarks) The register that permits the access to Segment(n) for LPDSP32 PM exists in System Controller.

4-3-3-43 Reserved

[Base + 0x0900 - 0x0908]

Bit	31	30	29	28	27	26	25	24
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : Reserved
don't use

Reserved

[Base + 0x090C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit29-0 : Reserved
don't use

4-4 Operation

The remainder of this page is intentionally left blank.

5 SystemController

5-1 Introduction

This chapter provides the outline of System Controller in the section below.

- The outline of System Controller

5-1-1 The outline of System Controller

There are following registers in the System controller block as functions for this LSI's entire system control

System Controller Block has bellow registers for this LSI control.

- Cortex-M3 Core1,LPDSP Reset,Clock Control Register
  Core Control Register
  Power Down Clock Control Register

- SWV Setting Register
  SWV Setting Register

- Remap Register
  Remap Register

- Memory Access Control Register
  Memory Access Control Register0,1,2P,2A,2B,3,4

- Internal Block Power ON/OFF Control Register
  Power Control Register
  ISOLATION Control Register

- Watch Dog Reset and Lockup Control Register
  Watch Dog Reset Control Regster0,1,2
  Cortex-M3 Core1 LOCKUP Control Register
  Cortex-M3 LOCKUP History Register

- Version, Boot Mode Monitor Register
  Mode Monitor Register

- Module Control Register
  Module Clock Control BASIC,EXT1,EXT3,EXT4,APB Register
  Module Reset Control BASIC,EXT1,EXT3,EXT4,APB Register

- IO Control Register
  Port Function Switching Register n,5
  Drive Function Switching Register n,5,6

Resistance Control Register n,5,6

- Other Supporting Register

  SD IF Supporting Register 0,1

  DREQ Selection Register 0,1,2,3

  CACHE Control Register

  BMODE Pin I/O Control Register

  BMODE Data Register

  GPO Function Control Register

  GPO DAMP Control Register

  GPO Drive Capability Setting Register

  GPO Data Output Register

  USB Control Register

  USB Status Register

  S-Flash I/F Selection Register

  SDRAMC I/F Selection Register

  I2C Mode Selection Register

  AudioPLL Selection Register

5-2 Functional Overview

This chapter provides the outline of functional blocks used for the System Controller configuration in the section below.

- System Controller circuit configuration

5-2-1 System Controller circuit configuration

System Controller configuration is as shown in Figure 5-1 below:

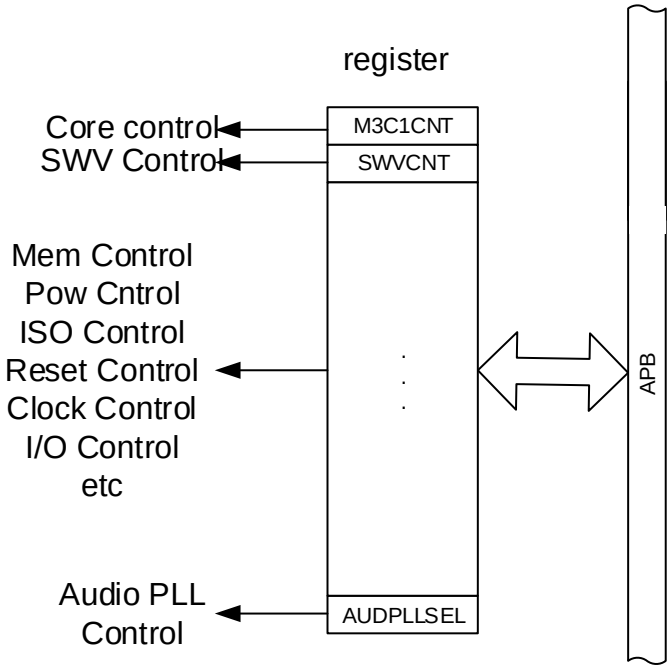


Figure 5-1 System Controller configuration

5-3 Programmer's Model

This chapter provides System Controller register information and other information necessary for the register setting using microcontroller in the sections below:

- About the Programmer's model
- Summary of System Controller Registers
- Register Descriptions

5-3-1 About the Programmer's model

System Controller base address is not fixed and varies by the system implementation.

Please refer to the *System specifications* for more information about the base address.

Offset is fixed for all registers.

5-3-2 Summary of System Controller Registers

System Controller is controlled by the registers in "Table 5-1".

Remarks: Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

,

Table 5-1 System Controller registers

Offset	Abbreviation	Attribute	Register name	Initial value
0x0000	M3C1CNT	R/W	Core control register	0x0000_0000
0x0004	SWVCNT	R/W	SWV setting register	0x0000_0000
0x0008	REMAP	R/W	Remap register	0x0000_0000
0x000C	MEMEN0	R/W	Memory access control register 0	0x0000_03FF
0x0010	MEMEN1	R/W	Memory access control register 1	0x0000_03FF
0x0014	MEMEN2P	R/W	Memory access control register 2P	0x0000_03FF
0x0018	MEMEN2A	R/W	Memory access control register 2A	0x0000_01FF
0x001C	MEMEN2B	R/W	Memory access control register 2B	0x0000_01FF
0x0020	MEMEN3	R/W	Memory access control register 3	0x0000_0000
0x0024	MEMEN4	R/W	Memory access control register 4	0x0000_0000
0x0028	LSISTBY	R/W	Power control register	0x0000_0000
0x002C	ISOCNT	R/W	ISOLATION control register	0x0000_01FF
0x0030	WDRCNT0	R/W	Watch Dog Reset control register 0	0x0000_0001
0x0034	WDRCNT1	R/W	Watch Dog Reset control register 1	0x0000_0001
0x0038	WDRCNT2	R/W	Watch Dog Reset control register 2	0x0000_0001
0x003C	C1LUPC	R/W	Cortex-M3 Core1 LOCKUP control register	0x0000_0000
0x0040	PDCLKCNT	R/W	Power down clock control register	0x0000_0000
0x0044	LOCKUPR	R/W	Cortex-M3 LOCKUP history register	0x0000_0000
0x00FC	MODEM	R	Mode monitor register	0x0100_0000
0x0100	MCLKCNTBASIC	R/W	Module clock control BASIC register	0x0000_0000
0x0104	MCLKCNTEXT1	R/W	Module clock control EXT1 register	0x0000_0000
0x0108	MCLKCNTEXT3	R/W	Module clock control EXT3 register	0x0000_0000
0x010C	MCLKCNTEXT4	R/W	Module clock control EXT4 register	0x0000_0000
0x0110	MCLKCNTAPB	R/W	Module clock control APB register	0x0000_0000
0x0114	MRSTCNTBASIC	R/W	Module reset control BASIC register	0x0000_0000
0x0118	MRSTCNTEXT1	R/W	Module reset control EXT1 register	0x0000_0000
0x011C	MRSTCNTEXT3	R/W	Module reset control EXT3 register	0x0000_0000
0x0120	MRSTCNTEXT4	R/W	Module reset control EXT4 register	0x0000_0000
0x0124	MRSTCNTAPB	R/W	Module reset control APB register	0x0000_0000

0x0400	PMDCNT0	R/W	Port function select register 0	0x0000_0000
0x0404	PMDCNT1	R/W	Port function select register 1	0x0000_0000
0x0408	PMDCNT2	R/W	Port function select register 2	0x0000_0000
0x040C	PMDCNT3	R/W	Port function select register 3	0x0000_0000
0x0410	PMDCNT4	R/W	Port function select register 4	0x0000_0000
0x0414	PMDCNT5	R/W	Port function select register 5	0x000F_0000
0x0430	PTDRVCNT0	R/W	Drive capacity switching register 0	0x0000_0000
0x0434	PTDRVCNT1	R/W	Drive capacity switching register 1	0x0000_0000
0x0438	PTDRVCNT2	R/W	Drive capacity switching register 2	0x0000_0000
0x043C	PTDRVCNT3	R/W	Drive capacity switching register 3	0x0000_0000
0x0440	PTDRVCNT4	R/W	Drive capacity switching register 4	0x0000_0000
0x0444	PTDRVCNT5	R/W	Drive capacity switching register 5	0x0000_0000
0x0448	PTDRVCNT6	R/W	Drive capacity switching register 6	0x0000_0000
0x044C	PUDCNT0	R/W	Resistance control register 0	0x0000_0040
0x0450	PUDCNT1	R/W	Resistance control register 1	0x0000_0000
0x0454	PUDCNT2	R/W	Resistance control register 2	0x0000_0000
0x0458	PUDCNT3	R/W	Resistance control register 3	0x0000_0000
0x045C	PUDCNT4	R/W	Resistance control register 4	0x0000_0000
0x0460	PUDCNT5	R/W	Resistance control register 5	0x0000_0000
0x0464	PUDCNT6	R/W	Resistance control register 6	0x0000_0000
0x0800	SDCTL	R/W	SD IF supporting register	0x3F80_8080
0x0804	SDCTL2	R/W	SD IF supporting register 2	0x0000_0000
0x0808	DRQSEL0	R/W	DREQ select register 0	0x0000_0000
0x080C	DRQSEL1	R/W	DREQ select register 1	0x0000_0000
0x0810	DRQSEL2	R/W	DREQ select register 2	0x0000_0000
0x0814	DRQSEL3	R/W	DREQ select register 3	0x0000_0000
0x0818	CACHE_CTL	R/W	CACHE control register	0x0000_0003
0x081C	BMODE_CNT	R/W	BMODE pin I/O control register	0x0000_0033
0x0820	BMODE_DT	R/W	BMODE data register	0x0000_00xx
0x0824	GPO_CNT	R/W	GPO function control register	0x0000_0000
0x0828	GPO_DZCTL	R/W	GPO DAMP control register	0x0000_0000
0x082C	GPO_DZINP	R/W	GPO drive capacity setting register	0x0000_0000
0x0830	GPO_DOUT	R/W	GPO data output register	0x0000_0000
0x0834	USBCNT	R/W	USB control register	0x0000_003A
0x0838	USBSTAT	R	USB status register	0x0000_0000
0x083C	SFIFSEL	R/W	S-Flash I/F select register	0x0000_0000
0x0840	SDRAMIFSEL	R/W	SDRAMC I/F select register	0x0000_0000
0x0844	I2CMODE	R/W	I2C mode select register	0x0000_0000
0x0F00	AUDPLLSEL	R/W	AudioPLL select register	0x0000_0000
0x0F04	Reserved	R/W	Reserved	0x0000_0000
0x0FFC	Reserved	R/W	Reserved	0x0000_0000

5-3-3 Register Descriptions

5-3-3-1 Core Control Register: CORECNT

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	DSPRSTN	DSPCLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	C1RSTN	C1CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit9: DSPRSTN
Controls LPDSP reset. When LPDSP is not used, reset LPDSP. When LPDSP is used, supply the clock and then release the reset.
0: Resets LPDSP.
1: release LPDSP reset.

Attention: Supply at least 3 clock cycles before releasing the reset.

Bit8: DSPCLKEN
Controls LPDSP clock supply. When LPDSP is not used, LPDSP clock can be stopped. When LPDSP is used, supply the clock before releasing the reset.
0: Stops LPDSP clock.
1: Supplies LPDSP clock.

Attention: Supply at least 3 clock cycles before cancelling the reset.

Bit1: C1RSTN
Controls the reset of Cortex-M3 core1. When Cortex-M3 core1 is not used, reset Cortex-M3 core1. When Cortex-M3 core1 is used, supply the clock and then cancel the reset.
0: Resets Cortex-M3 core1.
1: Releases the reset of Cortex-M3 core1.

Attention: Supply at least 3 clock cycles before cancelling the reset.

Bit0: C1CLKEN

Controls the clock supply of Cortex-M3 core1. When Cortex-M3 core1 is not used, Cortex-M3 core 1 clock can be stopped. When Cortex-M3 core1 is used, supply the clock before releasing the reset.

0: Stops the clock of Cortex-M3 core1.

1: Supply the clock of Cortex-M3 core1.

Attention: Supply at least 3 clock cycles before releasing the reset.

5-3-3-2 SWV Setting Register: SWVCNT

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	SWVSEL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0: SWVSEL
Selects SWV output of Cortex-M3 core0 and Cortex-M3 core1. In the reset state, Cortex-M3 core0 SWV output is selected.
0: Selects SWV output of Cortex-M3 core0.
1: Selects SWV output of Cortex-M3 core1.

5-3-3-3 Remap Register: REMAP

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	REMAP1	REMAP0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1-0: REMAP1-0
Sets whether to remap the program memory or not
00 b: No remap
01 b: Remap
10 b: Setting prohibited
11 b: Remap

Attention: This register reset is executed only at the time of hardware reset or watchdog timer reset. It will not reset by the JTAG debugger reset.

Refer to the separate datasheet for the address map of this REMAP register setting.

5-3-3-4 Memory Access Control Register 0: MEMEN0

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	MEMEN09	MEMEN08
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	MEMEN07	MEMEN06	MEMEN05	MEMEN04	MEMEN03	MEMEN02	MEMEN01	MEMEN00
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0: MEMEN0n

Sets the built-in SRAM (Seg0 to Seg9) that can be accessed from Cortex-M3 core0.
When disabled access is made, Cortex-M3 core1 will jump to Abort.
0: Disables the access from core0 to Seg n
1: Enables the access from core0 to Seg n.

5-3-3-5 Memory Access Control Register 1: MENEN1

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	MEMEN19	MEMEN18
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	MEMEN17	MEMEN16	MEMEN15	MEMEN14	MEMEN13	MEMEN12	MEMEN11	MEMEN10
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0: MEMEN1n

Sets built-in SRAM (Seg0 to Seg9) that can be accessed from Cortex-M3 core1.
When disabled access is made, Cortex-M3 core1 will jump to Abort.
0: Disables the access from core1 to Seg n
1: Enables the access from core1 to Seg n.

5-3-3-6 Memory Access Control Register 2P: MEMEN2P

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	MEMEN2P9	MEMEN2P8
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	MEMEN2P7	MEMEN2P6	MEMEN2P5	MEMEN2P4	MEMEN2P3	MEMEN2P2	MEMEN2P1	MEMEN2P0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-0: MEMEN2Pn

Sets built-in SRAM (Seg0 to Seg9) that can be accessed from LPDSP PM BUS.

When disabled access is made, the access itself is normally processed, but the relevant bit of INTC register DSPPMSTS will become “1”.

0: Disables the access from LPDSP to Seg n.

1: Enables the access from LPDSP to Seg n.

5-3-3-7 Memory Access Control Register 2A: MEMEN2A

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	MEMEN2A8
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	MEMEN2A7	MEMEN2A6	MEMEN2A5	MEMEN2A4	MEMEN2A3	MEMEN2A2	MEMEN2A1	MEMEN2A0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit8-0: MEMEN2An

Set built-in SRAM (Seg0 to Seg8) that can be accessed from LPDSP DMA bus. When disabled access is made, the access itself is normally processed, but the relevant bit of INTC register DSPDMASTS will become “1”.

0: Disables the access from LPDSP to Seg n.

1: Enables the access from LPDSP to Seg n.

* Seg9 is for PM. It cannot be accessed from DMA bus.

5-3-3-8 Memory Access Control Register 2B: MEMEN2B

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	MEMEN2B8
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	MEMEN2B7	MEMEN2B6	MEMEN2B5	MEMEN2B4	MEMEN2B3	MEMEN2B2	MEMEN2B1	MEMEN2B0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit8-0: MEMEN2Bn

Set the built-in SRAM (Seg0 to Seg8) that can be accessed from LPDSP DMB bus. When disabled access is made, the access itself is normally processed, but the relevant bit of INTC register DSPDMBSTS will become “1”.

0: Disables the access from LPDSP to Seg n.

1: Enables the access from LPDSP to Seg n.

* Seg9 is for PM. It cannot be accessed from DMA bus.

5-3-3-9 Memory Access Control Register 3: MEMEN3

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	MEMEN391	MEMEN390	MEMEN381	MEMEN380
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	MEMEN371	MEMEN370	MEMEN361	MEMEN360	MEMEN351	MEMEN350	MEMEN341	MEMEN340
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	MEMEN331	MEMEN330	MEMEN321	MEMEN320	MEMEN311	MEMEN310	MEMEN301	MEMEN300
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit19-0: MEMEN3n1-0

This register can make SRAM (Seg0 to Seg9) be exclusive to either LPDSP or AMBA.
When exclusive to LPDSP, DMAC0, 1, 2, LCDC and USB as well as Cortex-M3 core0 and core1 are inaccessible, write is disabled, and read-data is indefinite.
When exclusive to AMBA, LPDSP is inaccessible, write is disabled, and read-data is indefinite.
Use this for debug.
00 b: Seg n is not exclusive
01 b: Seg n is exclusive to LPDSP
10 b: Seg n is exclusive to AMBA
11 b: Setting prohibited

5-3-3-10 Memory Access Control Register 4: MEMEN4

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	DWAIT	HWAIT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1: DWAIT
Sets wait cycle for when LPDSP DMA bus read-accesses from the ROM.
0: No wait
1: 1 wait

Remark: When this register is set to “1”, 1 cycle wait is generated at BASIC clock.

Bit0: HWAIT
Sets wait cycle for when Cortex-M3 core0 or core1 read-accesses from ROM.
0: No wait
1: 1 wait

Remark: When this register is set to “1”, 1 cycle wait is generated at BASIC clock.

5-3-3-11 Power Control Register: LSISTBY

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	STBYI
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	STBYH	STBYG	reserved	STBYE	STBYD	STBYC	STBYB	STBYA
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit8-0: STBY (I–A)

Controls the power of internal blocks that can be turned off. There are 8 internal blocks (A to I) whose power can be turned off. Bit 0 to 8 correspond to A to I.
0: Supplies power to the relevant blocks
1: Stop supplying power to the relevant blocks

* After no-power-supplying state changes to power-supplying state, take enough time for the voltage to stabilize before activating the relevant blocks.

Before changing this register (STBY (I–A)), make sure the relevant blocks are set to “isolate” by using [Base + 0x0020] ISOCNT (I–A).

5-3-3-12 ISOLATION Control Register: ISOCNT

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	ISOCNTI
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	ISOCNTH	ISOCNTG	reserved	ISOCNTE	ISOCNTD	ISOCNTC	ISOCNTB	ISOCNTA
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit8-0: ISOCNT (I–A)

Controls the isolation of internal blocks that can be turned off. There are 8 internal blocks (A to I) that can be isolated. Bit 0 to 8 correspond to A to I.
0: Isolates the relevant blocks.
1: Does not isolate the relevant blocks.

* This register (ISOCNT) shall be set to “0” to control [Base + 0x001C] STBY (I–A).
Set “1” of ISOCNT (I–A) only for blocks that are power-supplied by [Base + 0x001C] STBY (I–A).

5-3-3-13 Watch Dog Reset Control Register 0: WDRCNT0

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	C0WDC
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Bit0: C0WDC

Controls whether to stop Watch Dog Timer0 counting when Cortex-M3 core0 falls into the break state during debugging

0: Does not stop the Watch Dog Timer0 counting at the time of break

1: Stops the Watch Dog Timer0 counting at the time of break

* Watch dog timer of the Plain Timer0 is the only one that is functional as a watchdog timer for Cortex-M3 core0.

5-3-3-14 Watch Dog Reset Control Register 1: WDRCNT1

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	C1WDC
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Bit0: C1WDC

Controls whether to stop Watch Dog Timer1 counting when Cortex-M3 core1 falls into the break state during debugging

0: Does not stop Watch Dog Timer1 counting at the time of break

1: Stops Watch Dog Timer1 counting at the time of break

* Watch dog timer of the Plain Timer1 is the only one that is functional as a watchdog timer for Cortex-M3 core1.

5-3-3-15 Watch Dog Reset Control Register 2: WDRCNT2

[Base + 0x0038]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	DSPWDC
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Bit0: DSPWDC

Controls whether to stop Watch Dog Timer2 counting when LPDSP falls into the break state during debugging

0: Does not stop Watch Dog Timer2 counting at the time of break

1: Stops Watch Dog Timer2 counting at the time of break

* Watch dog timer of the Plain Timer2 is the only one that is functional as a watchdog timer for LPDSP.

5-3-3-16 Cortex-M3 Core1 LOCKUP Control Register: C1LUPC

[Base + 0x003C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	C1LUPC
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0: C1LUPC

Controls whether to reset the whole chip when LOCKUP is occurring in Cortex-M3 core1

0: Does not reset the whole chip at the time of LOCKUP.

1: Resets the whole chip at the time of LOCKUP.

* When Cortex-M3 core1 is locked up, Cortex-M3 core1 will reset independently of the setting.

5-3-3-17 Power Down Clock Control Register: PDCLKCNT

[Base + 0x0040]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	PDCLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0: PDCLKEN

Selects whether to supply clock to LPDSP when LPDSP powerdown command is executed. Normally, when powerdown command is executed, the clock supply to LPDSP stops to reduce power consumption. However, during debugging, if needed, setting this flag to “1” enables the clock supply to LPDSP even when powerdown command is executed.

0: Does not supply clock to LPDSP when powerdown command is executed.
1: Supplies clock to LPDSP when powerdown command is executed.

5-3-3-18 Cortex-M3 LOCKUP History Register: LOCKUPR

[Base + 0x0044]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	LOCKUP R1	LOCKUP R0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

At the time of READ

Bit1-0: LOCKUPR1-0

Cortex-M3 may assert LOCKUP signal when a malfunction occurs. In that case, the whole system of this device will get reset. At the time of subsequent recovery, this register can check the history of whether LOCKUP signal was asserted or not as the reset factor.

0: No LOCKUP occurred.

1: LOCKUP occurred.

* Bit 1 and 0 correspond to core1 and core0 LOCKUP histories, respectively.

At the time of WRITE

Bit1-0: LOCKUPR1-0

Writing “1” will clear the LOCKUP history. There is no need to write “0”.

0: No effect.

1: Clears LOCKUP history.

* Bit 1 and 0 correspond to core1 and core0 LOCKUP histories, respectively.

5-3-3-19 Mode Monitor Register: MODEM

[Base + 0x00FC]

Bit	31	30	29	28	27	26	25	24
Name	UMAV3	UMAV2	UMAV1	UMAV0	MAV3	MAV2	MAV1	MAV0
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	MIV3	MIV2	MIV1	MIV0	REV3	REV2	REV1	REV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	BMODE1	BMODE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-16: UMAV3-0,MAV3-0,MIV3-0,REV3-0
Indicates the versions of LSI
UMAV3-0: Upper Major Version
MAV3-0: Major Version
MIV3-0: Minor Version
REV3-0: Revision

Bit1-0: BMODE1-0
Monitor register of boot mode switching pin. BMODE1, 0 input (Low/High) can be confirmed by reading this.
* When BMODE 1 and 0 of BMODE pin input control register (0x081C) are not set to input, the boot mode values cannot be read correctly. After releasing power-on reset, BMODE 1 and 0 will be in the input state.

5-3-3-20 Module Clock Control BASIC Register: MCLKCNTBASIC

[Base + 0x0100]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	USBDEV CLKEN	CACHE CLKEN	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	DMAC CLKEN	-	USBHOST CLKEN	SFIFCLK EN	EXTMEM CCLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit7-0: <Block Name> CLKEN
Controls the clock of <Block Name>.
0: Stops the clock to <BlockName>.
1: Supplies the clock to <BlockName>.

5-3-3-21 Module Clock Control EXT1 Register: MCLKCNTEXT1

[Base + 0x0104]

Bit	31	30	29	28	27	26	25	24
Name	-	PTM2C CLKEN	PTM1C CLKEN	PTM0C CLKEN	MTM3C CLKEN	MTM2C CLKEN	MTM1C CLKEN	MTM0C CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	MSPB CLKEN	MSHOST CLKEN	SDIF2 CLKEN	SDIF1 CLKEN	SDIF0 CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	PTM2 CLKEN	PTM1 CLKEN	PTM0 CLKEN	MTM3 CLKEN	MTM2 CLKEN	MTM1 CLKEN	MTM0 CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit30-28: PTMnCCLKEN
Controls the clock for the timer count of PTMn (n = 2 to 0).
0: Stops the count clock to PTMn.
1: Supplies the count clock to PTMn.

Bit27-24: MTMnCCLKEN
Controls the clock for the timer count of MTMn (n = 3 to 0).
0: Stops the count clock to MTMn.
1: Supplies the count clock to MTMn.

Bit12-0: <Block Name> CLKEN
Controls the clock of <Block Name>
0: Stops the clock to <BlockName>.
1: Supplies the clock to <BlockName>.

5-3-3-22 Module Clock Control EXT3 Register: MCLKCNTEXT3

[Base + 0x0108]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	AudioBuf CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit25-0: <Block Name> CLKEN
Controls the clock of <Block Name>.
0: Stops the clock to <BlockName>.
1: Supplies the clock to <BlockName.

5-3-3-23 Module Clock Control EXT4 Register: MCLKCNTEXT4

[Base + 0x010C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	SDRAMC CLKEN1	SDRAMC CLKEN0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit0: SDRAMCCLKEN0
Controls the clock of SDRAM Controller.
0: Stops the clock to SDRAM Controller.
1: Supplies the clock to SDRAM Controller.

Bit1: SDRAMCCLKEN1
Controls I/F clock between SDRAM Controller and SDRAM.
0: Stops SDRAM I/F clock.
1: Supplies SDRAM I/F clock.

5-3-3-24 Module Clock Control APB Register: MCLKCNTAPB

[Base + 0x0110]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	UART2IF CLKEN	UART1IF CLKEN	UART0IF CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	Reserved	GPO CLKEN	RTC CLKEN	UART2 CLKEN	UART1 CLKEN	UART0 CLKEN	I2C1 CLKEN	I2C0 CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	SPI CLKEN	ADC CLKEN	PORT5 CLKEN	PORT4 CLKEN	PORT3 CLKEN	PORT2 CLKEN	PORT1 CLKEN	PORT0 CLKEN
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit26-24: UARTnIFCLKEN
Controls the interface (communication) clock of UARTn (n = 2 to 0).
0: Stops the communication clock of UARTn.
1: Supplies the communication clock of UARTn.

Bit14-0: <Block Name> CLKEN
Controls the clock of <Block Name>.
0: Stops the clock to <BlockName>.
1: Supplies the clock to <BlockName>.

5-3-3-25 Module Reset Control BASIC Register: MRSTCNTBASIC

[Base + 0x0114]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	USBDEV RSTB	CACHE RSTB	DSPCMD RSTB	MUTEX RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	DMAC RSTB	IRQCNT RSTB	USB RSTB	SFIF RSTB	EXTMEMC RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit11-0: <Block Name> RSTB
Controls the reset of <Block Name>.
0: Resets <BlockName>.
1: Releases the reset of <BlockName>.

5-3-3-26 Module Reset Control EXT1 Register: MRSTCNTEXT1

[Base + 0x0118]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	MSIF RSTB	SDIF2 RSTB	SDIF1 RSTB	SDIF0 RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	PTM2 RSTB	PTM1 RSTB	PTM0 RSTB	MTM3 RSTB	MTM2 RSTB	MTM1 RSTB	MTM0 RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit11-0: <Block Name> RSTB
Controls the reset of <Block Name>.
0: Resets <BlockName>.
1: Releases the reset of <BlockName>.

5-3-3-27 Module Reset Control EXT3 Register: MRSTCNTEXT3

[Base + 0x011C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	AudioBuf RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit25-0: <Block Name> RSTB
Controls the reset of <Block Name>.
0: Resets <BlockName>.
1: Releases the reset of <BlockName>.

5-3-3-28 Module Reset Control EXT4 Register: MRSTCNTEXT4

[Base + 0x0120]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	SDRAMC RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0: SDRAMCRSTB
Controls the reset of SDRAM Controller.
0: Resets SDRAM Controller.
1: Releases the reset of SDRAM Controller.

5-3-3-29 Module Reset Control APB Register: MRSTCNTAPB

[Base + 0x0124]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	Reserved	-	RTC RSTB	UART2 RSTB	UART1 RSTB	UART0 RSTB	I2C1 RSTB	I2C0 RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	SPI RSTB	ADC RSTB	PORT5 RSTB	PORT4 RSTB	PORT3 RSTB	PORT2 RSTB	PORT1 RSTB	PORT0 RSTB
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit13-0: <Block Name> RSTB
Controls the reset of <Block Name>
0: Resets <BlockName>.
1: Releases the reset of <BlockName>.

5-3-3-30 Port Function Switching Register n: PMDCNTn

[Base + 0x0400 - 0x0410]

Bit	31	30	29	28	27	26	25	24
Name	PnSELF1	PnSELF0	PnSELE1	PnSELE0	PnSELD1	PnSELD0	PnSELC1	PnSELC0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	PnSELB1	PnSELB0	PnSELA1	PnSELA0	PnSEL91	PnSEL90	PnSEL81	PnSEL80
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	PnSEL71	PnSEL70	PnSEL61	PnSEL60	PnSEL51	PnSEL50	PnSEL41	PnSEL40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	PnSEL31	PnSEL30	PnSEL21	PnSEL20	PnSEL11	PnSEL10	PnSEL01	PnSEL00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: PnSELm1-m0

Sets functions for combination pins. Pin function can be set by changing the corresponding PnSEL. Each pin function for PnSELm1-m0 is as shown in the table below.

(n = 0 to 4, m = F to 0)

Port 0 combination pin settings

	P0SELm1-m0 = 00 b		P0SELm1-m0 = 01 b		P0SELm1-m0 = 10 b		P0SELm1-m0 = 11 b	
	Function	I/O	Function	I/O	Function	I/O	Function	I/O
GPIO00	GPIO00	B	TCLKA0	I	-	-	BCK1	B
GPIO01	GPIO01	B	TCLKB0	I	-	-	LRCK1	B
GPIO02	GPIO02	B	TIOCB00	B	DMDIN0	I	DIN1	I
GPIO03	GPIO03	B	TIOCB01	B	DMCKO0	O	QSCS	O
GPIO04	GPIO04	B	TXD1	O	SDAT20	B	-	-
GPIO05	GPIO05	B	RXD1	I	SDAT21	B	-	-
GPIO06	GPIO06	B	NCS0	O	-	-	-	-
GPIO07	GPIO07	B	SCL0	O	-	-	-	-
GPIO08	GPIO08	B	SDA0	B	-	-	-	-
GPIO09	GPIO09	B	TIOCA00	B	SDCLK2	O	PHI0	O
GPIO0A	GPIO0A	B	TIOCA01	B	SDCMD2	B	PHI1	O
GPIO0B	GPIO0B	B	TXD2	O	TIOCA10	B	-	-
GPIO0C	GPIO0C	B	RXD2	I	TIOCA11	B	-	-
GPIO0D	GPIO0D	B	SCK1	O	-	-	-	-
GPIO0E	GPIO0E	B	SDI1(QIO0)	O(B)	-	-	-	-
GPIO0F	GPIO0F	B	SDO1(QIO1)	I(B)	-	-	-	-

* “-” indicates setting prohibited.

Port 1 combination pin settings

	P1SELm1-m0 = 00 b		P1SELm1-m0 = 01 b		P1SELm1-m0 = 10 b		P1SELm1-m0 = 11 b	
	Function	I/O	Function	I/O	Function	I/O	Function	I/O
GPIO10	GPIO10	B	NCS1	O	-	-	RXD0	I
GPIO11	GPIO11	B	SWP1(QIO2)	O(B)	-	-	-	-
GPIO12	GPIO12	B	SHOLD1(QIO3)	O(B)	-	-	-	-
GPIO13	GPIO13	B	BCK1	B	-	-	-	-
GPIO14	GPIO14	B	LRCK1	B	-	-	-	-
GPIO15	GPIO15	B	DOUT1	O	-	-	-	-
GPIO16	GPIO16	B	NLBEXA0	O	-	-	-	-
GPIO17	GPIO17	B	NRD	O	-	-	-	-
GPIO18	GPIO18	B	MCLK0	B	MCLK1	B	-	-
GPIO19	GPIO19	B	BCK0	B	DMCKO1	O	-	-
GPIO1A	GPIO1A	B	LRCK0	B	DMDIN1	I	-	-
GPIO1B	GPIO1B	B	DIN0	I	DMDIN0	I	-	-
GPIO1C	GPIO1C	B	DOUT0	O	DMCKO0	O	-	-
GPIO1D	GPIO1D	B	SCK0	B	-	-	-	-
GPIO1E	GPIO1E	B	SDI0	I	-	-	-	-
GPIO1F	GPIO1F	B	SDO0	O	-	-	-	-

* “-” indicates setting prohibited.

MCLK0 and MCLK1 of GPIO18; Master clock on the PCM0 side is MCLK0 and Master clock on the PCM1 side is MCLK1. There is no need to distinguish MCLK0 and 1 when inputting the Master clock.

Port 2 combination pin settings

	P2SELm1-m0 = 00 b		P2SELm1-m0 = 01 b		P2SELm1-m0 = 10 b		P2SELm1-m0 = 11 b	
	Function	I/O	Function	I/O	Function	I/O	Function	I/O
GPIO20	GPIO20	B	SDCD1	I	SWO	O	TDI	I
GPIO21	GPIO21	B	SDWP1	I	INS	I	TDO	O
GPIO22	GPIO22	B	SDCLK1	O	SCLK	O	-	-
GPIO23	GPIO23	B	SDCMD1	B	BS	O	-	-
GPIO24	GPIO24	B	SDAT10	B	DATA0	B	-	-
GPIO25	GPIO25	B	SDAT11	B	DATA1	B	-	-
GPIO26	GPIO26	B	SDAT12	B	DATA2	B	-	-
GPIO27	GPIO27	B	SDAT13	B	DATA3	B	-	-
GPIO28	GPIO28	B	-	-	SDWP2	I	TMS	I
GPIO29	GPIO29	B	-	-	SDCD2	I	TCK	I
GPIO2A	GPIO2A	B	-	-	SDRADDR12	O	-	-
GPIO2B	GPIO2B	B	SCL1	O	-	-	-	-
GPIO2C	GPIO2C	B	SDA1	B	-	-	-	-
GPIO2D	GPIO2D	B	DMCKO0	O	SDRADDR11	O	-	-
GPIO2E	GPIO2E	B	-	-	-	-	-	-
GPIO2F	GPIO2F	B	-	-	-	-	-	-

* “-” indicates setting prohibited.

Port 3 combination pin settings

	P3SELm1-m0 = 00 b		P3SELm1-m0 = 01 b		P3SELm1-m0 = 10 b		P3SELm1-m0 = 11 b	
	Function	I/O	Function	I/O	Function	I/O	Function	I/O
GPIO30	GPIO30	B	NWRENWRL	O	-	-	-	-
GPIO31	GPIO31	B	NHBNWRH	O	-	-	TXD0	O
GPIO32	GPIO32	B	EXA1	O	-	-	-	-
GPIO33	GPIO33	B	EXA2	O	-	-	-	-
GPIO34	GPIO34	B	EXA3	O	-	-	-	-
GPIO35	GPIO35	B	EXA4	O	-	-	-	-
GPIO36	GPIO36	B	EXA5	O	-	-	-	-
GPIO37	GPIO37	B	EXA6	O	-	-	-	-
GPIO38	GPIO38	B	EXA7	O	-	-	-	-
GPIO39	GPIO39	B	EXA8	O	-	-	-	-
GPIO3A	GPIO3A	B	EXA9	O	-	-	-	-
GPIO3B	GPIO3B	B	EXA10	O	-	-	-	-
GPIO3C	GPIO3C	B	EXA11	O	-	-	-	-
GPIO3D	GPIO3D	B	EXA12	O	-	-	-	-
GPIO3E	GPIO3E	B	EXA13	O	-	-	-	-
GPIO3F	GPIO3F	B	EXA14	O	-	-	-	-

* “-” indicates setting prohibited.

Port 4 combination pin settings

	P4SELm1-m0 = 00 b		P4SELm1-m0 = 01 b		P4SELm1-m0 = 10 b		P4SELm1-m0 = 11 b	
	Function	I/O	Function	I/O	Function	I/O	Function	I/O
GPIO40	GPIO40	B	EXA15	O	-	-	-	-
GPIO41	GPIO41	B	EXA16	O	-	-	-	-
GPIO42	GPIO42	B	EXA17	O	-	-	-	-
GPIO43	GPIO43	B	EXA18	O	-	-	-	-
GPIO44	GPIO44	B	EXA19	O	-	-	-	-
GPIO45	GPIO45	B	EXA20	O	-	-	-	-
GPIO46	GPIO46	B	EXD0	B	-	-	-	-
GPIO47	GPIO47	B	EXD1	B	-	-	-	-
GPIO48	GPIO48	B	EXD2	B	-	-	-	-
GPIO49	GPIO49	B	EXD3	B	-	-	-	-
GPIO4A	GPIO4A	B	EXD4	B	-	-	-	-
GPIO4B	GPIO4B	B	EXD5	B	-	-	-	-
GPIO4C	GPIO4C	B	EXD6	B	-	-	-	-
GPIO4D	GPIO4D	B	EXD7	B	-	-	-	-
GPIO4E	GPIO4E	B	EXD8	B	-	-	-	-
GPIO4F	GPIO4F	B	EXD9	B	-	-	-	-

* “-” indicates setting prohibited.

5-3-3-31 Port Function Switching Register 5:PMDCNT5

[Base + 0x0414]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	P5SEL91	P5SEL90	P5SEL81	P5SEL80
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	P5SEL71	P5SEL70	P5SEL61	P5SEL60	P5SEL51	P5SEL50	P5SEL41	P5SEL40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	P5SEL31	P5SEL30	P5SEL21	P5SEL20	P5SEL11	P5SEL10	P5SEL01	P5SEL00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit19-0 : P5SE9m1-m0

Sets function for combination pins. Pin function can be set by changing the corresponding P5SEL. Each pin function for the values of P5SE9m1-m0 are as listed below.

Port5 combination pin settings

	P5SELm1-m0 = 00 b		P5SELm1-m0 = 01 b		P5SELm1-m0 = 10 b		P5SELm1-m0 = 11 b	
	Function	I/O	Function	I/O	Function	I/O	Function	I/O
GPIO50	GPIO50	B	EXD10	B	-	-	-	-
GPIO51	GPIO51	B	EXD11	B	-	-	-	-
GPIO52	GPIO52	B	EXD12	B	-	-	-	-
GPIO53	GPIO53	B	EXD13	B	-	-	-	-
GPIO54	GPIO54	B	EXD14	B	-	-	-	-
GPIO55	GPIO55	B	EXD15	B	-	-	-	-
GPIO56	GPIO56	B	CTS1	I	SDAT22	B	RXD0	I
GPIO57	GPIO57	B	RTS1	O	SDAT23	B	TXD0	O
GPIO58	GPIO58	B	DMCKO1	O	-	-	SWDCLK	I
GPIO59	GPIO59	B	DMDIN1	I	-	-	SWDIO	B
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-

* “-” indicates setting prohibited.

5-3-3-32 Drive Function Switching Register n: PTDRVCNTn

[Base + 0x0430 - 0x0440]

Bit	31	30	29	28	27	26	25	24
Name	PnDRVF1	PnDRVF0	PnDRVE1	PnDRVE0	PnDRVD1	PnDRVD0	PnDRVC1	PnDRVC0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	PnDRVB1	PnDRVB0	PnDRVA1	PnDRVA0	PnDRV91	PnDRV90	PnDRV81	PnDRV80
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	PnDRV71	PnDRV70	PnDRV61	PnDRV60	PnDRV51	PnDRV50	PnDRV41	PnDRV40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	PnDRV31	PnDRV30	PnDRV21	PnDRV20	PnDRV11	PnDRV10	PnDRV01	PnDRV00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : PnDRVm1-m0

Sets drive current capability of the pins. Drive current capability can be set by changing corresponding PnDRV. Drive current capability of each pin for the values in PnDRVm1-m0 are as shown below.

(n=0 ~ 4, m=F ~ 0)

Port0 Pin drive current capability setting (Target IO voltage = 2.7 v~3.6 v)

	P0DRVm1-m0 = 00 b	P0DRVm1-m0 = 01 b	P0DRVm1-m0 = 10 b	P0DRVm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO00	1 mA	-	2 mA	4 mA
GPIO01	1 mA	-	2 mA	4 mA
GPIO02	1 mA	-	2 mA	4 mA
GPIO03	6 mA	-	8 mA	10 mA
GPIO04	6 mA	-	8 mA	10 mA
GPIO05	6 mA	-	8 mA	10 mA
GPIO06	2 mA	-	4 mA	8 mA
GPIO07	1 mA	-	2 mA	4 mA
GPIO08	1 mA	-	2 mA	4 mA
GPIO09	6 mA	-	8 mA	10 mA
GPIO0A	6 mA	-	8 mA	10 mA
GPIO0B	1 mA	-	2 mA	4 mA
GPIO0C	1 mA	-	2 mA	4 mA
GPIO0D	6 mA	-	8mA	10 mA
GPIO0E	6 mA	-	8 mA	10 mA
GPIO0F	6 mA	-	8 mA	10 mA

* P0DRV with “-” across the row indicates there is no register for it. Therefore, it will be READ only.

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

Port1 Pin drive current capability setting (Target IO voltage = 2.7 v~3.6 v)

	P1DRVm1-m0 = 00 b	P1DRVm1-m0 = 01 b	P1DRVm1-m0 = 10 b	P1DRVm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO10	2 mA	-	4 mA	8 mA
GPIO11	6 mA	-	8 mA	10 mA
GPIO12	6 mA	-	8 mA	10 mA
GPIO13	1 mA	-	2 mA	4 mA
GPIO14	1 mA	-	2 mA	4 mA
GPIO15	1 mA	-	2 mA	4 mA
GPIO16	2 mA	-	4 mA	8 mA
GPIO17	2 mA	-	4 mA	8 mA
GPIO18	1 mA	-	2 mA	4 mA
GPIO19	1 mA	-	2 mA	4 mA
GPIO1A	1 mA	-	2 mA	4 mA
GPIO1B	1 mA	-	2 mA	4 mA
GPIO1C	1 mA	-	2 mA	4 mA
GPIO1D	1 mA	-	2 mA	4 mA
GPIO1E	1 mA	-	2 mA	4 mA
GPIO1F	1 mA	-	2 mA	4 mA

* P1DRV with “-” across the row indicates there is no register for it. Therefore, it will be READ only.

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

Port2 Pin drive current capability setting (Target IO voltage = 2.7 v~3.6 v)

	P2DRVm1-m0 = 00 b	P2DRVm1-m0 = 01 b	P2DRVm1-m0 = 10 b	P2DRVm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO20	-	-	-	-
GPIO21	-	-	-	-
GPIO22	6 mA	-	8 mA	10 mA
GPIO23	6 mA	-	8 mA	10 mA
GPIO24	6 mA	-	8 mA	10 mA
GPIO25	6 mA	-	8 mA	10 mA
GPIO26	6 mA	-	8 mA	10 mA
GPIO27	6 mA	-	8 mA	10 mA
GPIO28	1 mA	-	2 mA	4 mA
GPIO29	1 mA	-	2 mA	4 mA
GPIO2A	2 mA	-	4 mA	8 mA
GPIO2B	1 mA	-	2 mA	4 mA
GPIO2C	1 mA	-	2 mA	4 mA
GPIO2D	2 mA	-	4 mA	8 mA
GPIO2E	1 mA	-	2 mA	4 mA
GPIO2F	1 mA	-	2 mA	4 mA

* P2DRV with “-” across the row indicates there is no register for it. Therefore, it will be READ only.

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

Port3 Pin drive current capability setting (Target IO voltage = 2.7 v~3.6 v)

	P3DRVm1-m0 = 00 b	P3DRVm1-m0 = 01 b	P3DRVm1-m0 = 10 b	P3DRVm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO30	2 mA	-	4 mA	8 mA
GPIO31	2 mA	-	4 mA	8 mA
GPIO32	2 mA	-	4 mA	8 mA
GPIO33	2 mA	-	4 mA	8 mA
GPIO34	2 mA	-	4 mA	8 mA
GPIO35	2 mA	-	4 mA	8 mA
GPIO36	2 mA	-	4 mA	8 mA
GPIO37	2 mA	-	4 mA	8 mA
GPIO38	2 mA	-	4 mA	8 mA
GPIO39	2 mA	-	4 mA	8 mA
GPIO3A	2 mA	-	4 mA	8 mA
GPIO3B	2 mA	-	4 mA	8 mA
GPIO3C	2 mA	-	4 mA	8 mA
GPIO3D	2 mA	-	4 mA	8 mA
GPIO3E	2 mA	-	4 mA	8 mA
GPIO3F	2 mA	-	4 mA	8 mA

* P3DRV with “-” across the row indicates there is no register for it. Therefore, it will be READ only.

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

Port4 Pin drive current capability setting (Target IO voltage = 2.7 v~3.6 v)

	P4DRVm1-m0 = 00 b	P4DRVm1-m0 = 01 b	P4DRVm1-m0 = 10 b	P4DRVm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO40	2 mA	-	4 mA	8 mA
GPIO41	2 mA	-	4 mA	8 mA
GPIO42	2 mA	-	4 mA	8 mA
GPIO43	2 mA	-	4 mA	8 mA
GPIO44	2 mA	-	4 mA	8 mA
GPIO45	2 mA	-	4 mA	8 mA
GPIO46	2 mA	-	4 mA	8 mA
GPIO47	2 mA	-	4 mA	8 mA
GPIO48	2 mA	-	4 mA	8 mA
GPIO49	2 mA	-	4 mA	8 mA
GPIO4A	2 mA	-	4 mA	8 mA
GPIO4B	2 mA	-	4 mA	8 mA
GPIO4C	2 mA	-	4 mA	8 mA
GPIO4D	2 mA	-	4 mA	8 mA
GPIO4E	2 mA	-	4 mA	8 mA
GPIO4F	2 mA	-	4 mA	8 mA

* P4DRV with “-” across the row indicates there is no register for it. Therefore, it will be READ only.

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

5-3-3-33 Drive Capability Switching Register5 : PTDRVCNT5

[Base + 0x0444]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	P5DRV91	P5DRV90	P5DRV81	P5DRV80
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	P5DRV71	P5DRV70	P5DRV61	P5DRV60	P5DRV51	P5DRV50	P5DRV41	P5DRV40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	P5DRV31	P5DRV30	P5DRV21	P5DRV20	P5DRV11	P5DRV10	P5DRV01	P5DRV00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit19-0 : P5DRVm1-m0

Sets drive current capability of pins. Drive current capability can be set by changing corresponding P5DRV. Drive current capability of each pin for values in P5DRVm1-m0 are as shown below.

(m=9~0)

Port5 Pin drive current capability setting (Target IO voltage = 2.7 v~3.6 v)

	P5DRVm1-m0 = 00 b	P5DRVm1-m0 = 01 b	P5DRVm1-m0 = 10 b	P5DRVm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO50	2 mA	-	4 mA	8 mA
GPIO51	2 mA	-	4 mA	8 mA
GPIO52	2 mA	-	4 mA	8 mA
GPIO53	2 mA	-	4 mA	8 mA
GPIO54	2 mA	-	4 mA	8 mA
GPIO55	2 mA	-	4 mA	8 mA
GPIO56	6 mA	-	8 mA	10 mA
GPIO57	6 mA	-	8 mA	10 mA
GPIO58	1 mA	-	2 mA	4 mA
GPIO59	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

* P5DRV with “-” across the row indicates there is no register for it. Therefore, it will be READ only.

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

5-3-3-34 Drive Capability Switching Register 6 : PTDRVCNT6

[Base + 0x0448]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	XTALINFO1DRV1	XTALINFO1DRV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	XTALINFO0DRV1	XTALINFO0DRV0	SDRDQMDRV1	SDRDQMDRV0	SDRCSWEDRV1	SDRCSWEDRV0	SDRCLKSDRV1	SDRCLKSDRV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	SDRCKESDRV1	SDRCKESDRV0	SDRCRASDRV1	SDRCRASDRV0	SDRBA10DRV1	SDRBA10DRV0	SDRADDR10-0DRV1	SDRADDR15-0DRV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	SDRDATA15-0DRV1	SDRDATA15-0DRV0	SDAT03-0DRV1	SDAT03-0DRV0	SDCMD0DRV1	SDCMD0DRV0	SDCLK0DRV1	SDCLK0DRV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit25-0 : *DRV1-0

Sets drive current capability of pins. Drive current capability of pins can be set by changing corresponding DRV. Drive current capability of each pin for the value of DRV1-0 are as below.

Other pin drive current capability settings

	*DRV1-0 = 00 b	*DRV1-0 = 01 b	*DRV1-0 = 10 b	*DRV1-0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
SDCLK0	6 mA	-	8 mA	10 mA
SDCMD0	6 mA	-	8 mA	10 mA
SDAT03-0	6 mA	-	8 mA	10 mA
SDRDATA15:0	2 mA	-	4 mA	8 mA
SDRADDR10:0	2 mA	-	4 mA	8 mA
SDRBA1:0	2 mA	-	4 mA	8 mA
SDRCAS,RAS	2 mA	-	4 mA	8 mA
SDRCKE	2 mA	-	4 mA	8 mA
SDRCLK	2 mA	-	4 mA	8 mA
SDRCS,WE	2 mA	-	4 mA	8 mA
SDRDQM1:0	2 mA	-	4 mA	8 mA
XTALINFO0	2 mA	-	4 mA	8 mA
XTALINFO1	2 mA	-	4 mA	8 mA
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

* “-” indicates setting prohibited. Drive capability will be half if target IO voltage is 1.7 v~1.95 v.

5-3-3-35 Resistance Control Registern : PUDCNTn

[Base + 0x044C - 0x045C]

Bit	31	30	29	28	27	26	25	24
Name	PnPUDF1	PnPUDF0	PnPUDE1	PnPUDE0	PnPUDD1	PnPUDD0	PnPUDC1	PnPUDC0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	PnPUDB1	PnPUDB0	PnPUDA1	PnPUDA0	PnPUD91	PnPUD90	PnPUD81	PnPUD80
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	PnPUD71	PnPUD70	PnPUD61	PnPUD60	PnPUD51	PnPUD50	PnPUD41	PnPUD40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	PnPUD31	PnPUD30	PnPUD21	PnPUD20	PnPUD11	PnPUD10	PnPUD01	PnPUD00
Reset	0	0*A	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0 : PnPUDm1-m0

Setting for whether to execute internal pull-up/pull-down of the pin. This can be set by making changes on the PnPUD. Pull-up/pull-down setting of each pin for the values of PnPUDm1-m0 is as shown below.

(n=0 ~ 4, m=F ~ 0) *1: Only 0x044C P0PUD30 will have an initial value of 1.

Port0 pull-up/pull-down settings

	P0PUDm1-m0 = 00 b	P0PUDm1-m0 = 01 b	P0PUDm1-m0 = 10 b	P0PUDm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO00	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO01	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO02	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO03	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO04	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO05	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO06	PU and PD both OFF	Pull-UP	-	-
GPIO07	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO08	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO09	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO0A	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO0B	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO0C	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO0D	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO0E	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO0F	PU and PD both OFF	Pull-UP	Pull-Down	-

* “-” indicates setting prohibited.

Port1 pull-up/pull-down settings

	P1PUDm1-m0 = 00 b	P1PUDm1-m0 = 01 b	P1PUDm1-m0 = 10 b	P1PUDm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO10	PU and PD both OFF	Pull-UP	-	-
GPIO11	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO12	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO13	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO14	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO15	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO16	PU and PD both OFF	-	Pull-Down	-
GPIO17	PU and PD both OFF	-	Pull-Down	-
GPIO18	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO19	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO1A	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO1B	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO1C	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO1D	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO1E	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO1F	PU and PD both OFF	Pull-UP	Pull-Down	-

* “-” indicates setting prohibited.

Port2 pull-up/pull-down settings

	P2PUDm1-m0 = 00 b	P2PUDm1-m0 = 01 b	P2PUDm1-m0 = 10 b	P2PUDm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO20	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO21	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO22	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO23	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO24	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO25	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO26	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO27	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO28	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO29	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO2A	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO2B	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO2C	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO2D	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO2E	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO2F	PU and PD both OFF	Pull-UP	Pull-Down	-

* “-” indicates setting prohibited.

Port3 pull-up/pull-down settings

	P3PUDm1-m0 = 00 b	P3PUDm1-m0 = 01 b	P3PUDm1-m0 = 10 b	P3PUDm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO30	PU and PD both OFF	-	Pull-Down	-
GPIO31	PU and PD both OFF	-	Pull-Down	-
GPIO32	PU and PD both OFF	-	Pull-Down	-
GPIO33	PU and PD both OFF	-	Pull-Down	-
GPIO34	PU and PD both OFF	-	Pull-Down	-
GPIO35	PU and PD both OFF	-	Pull-Down	-
GPIO36	PU and PD both OFF	-	Pull-Down	-
GPIO37	PU and PD both OFF	-	Pull-Down	-
GPIO38	PU and PD both OFF	-	Pull-Down	-
GPIO39	PU and PD both OFF	-	Pull-Down	-
GPIO3A	PU and PD both OFF	-	Pull-Down	-
GPIO3B	PU and PD both OFF	-	Pull-Down	-
GPIO3C	PU and PD both OFF	-	Pull-Down	-
GPIO3D	PU and PD both OFF	-	Pull-Down	-
GPIO3E	PU and PD both OFF	-	Pull-Down	-
GPIO3F	PU and PD both OFF	-	Pull-Down	-

* “-” indicates setting prohibited.

Port4 pull-up/pull-down settings

	P4PUDm1-m0 = 00 b	P4PUDm1-m0 = 01 b	P4PUDm1-m0 = 10 b	P4PUDm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO40	PU and PD both OFF	-	Pull-Down	-
GPIO41	PU and PD both OFF	-	Pull-Down	-
GPIO42	PU and PD both OFF	-	Pull-Down	-
GPIO43	PU and PD both OFF	-	Pull-Down	-
GPIO44	PU and PD both OFF	-	Pull-Down	-
GPIO45	PU and PD both OFF	-	Pull-Down	-
GPIO46	PU and PD both OFF	-	Pull-Down	-
GPIO47	PU and PD both OFF	-	Pull-Down	-
GPIO48	PU and PD both OFF	-	Pull-Down	-
GPIO49	PU and PD both OFF	-	Pull-Down	-
GPIO4A	PU and PD both OFF	-	Pull-Down	-
GPIO4B	PU and PD both OFF	-	Pull-Down	-
GPIO4C	PU and PD both OFF	-	Pull-Down	-
GPIO4D	PU and PD both OFF	-	Pull-Down	-
GPIO4E	PU and PD both OFF	-	Pull-Down	-
GPIO4F	PU and PD both OFF	-	Pull-Down	-

* “-” indicates setting prohibited.

5-3-3-36 Resistance Control Register 5 : PUDCNT5

[Base + 0x0460]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	P5PUD91	P5PUD90	P5PUD81	P5PUD80
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	P5PUD71	P5PUD70	P5PUD61	P5PUD60	P5PUD51	P5PUD50	P5PUD41	P5PUD40
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	P5PUD31	P5PUD30	P5PUD21	P5PUD20	P5PUD11	P5PUD10	P5PUD01	P5PUD00
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit19-0 : P5PUDm1-m0

Setting for whether to execute internal pull-up/pull-down of the pin. This can be set by making changes on the P5PUD. Pull-up/pull-down setting of each pin for the values of P5PUDm1-m0 is as shown below.

(m=9 ~ 0)

Port5 pull-up/pull-down settings

	P5PUDm1-m0 = 00 b	P5PUDm1-m0 = 01 b	P5PUDm1-m0 = 10 b	P5PUDm1-m0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
GPIO50	PU and PD both OFF	-	Pull-Down	-
GPIO51	PU and PD both OFF	-	Pull-Down	-
GPIO52	PU and PD both OFF	-	Pull-Down	-
GPIO53	PU and PD both OFF	-	Pull-Down	-
GPIO54	PU and PD both OFF	-	Pull-Down	-
GPIO55	PU and PD both OFF	-	Pull-Down	-
GPIO56	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO57	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO58	PU and PD both OFF	Pull-UP	Pull-Down	-
GPIO59	PU and PD both OFF	Pull-UP	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

* “-” indicates setting prohibited.

5-3-3-37 Resistance Control Setting Register 6 : PUDCNT6

[Base + 0x0464]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	XTALINFO1PUD1	XTALINFO1PUD0	XTALINFO0PUD1	XTALINFO0PUD0	BMODE1PUD1	BMODE1PUD0	BMODE0PUD1	BMODE0PUD0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	SDRDATAPUD1	SDRDATAPUD0	SDAT03-0PUD1	SDAT03-0PUD0	SDCMD0PUD1	SDCMD0PUD0	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit11-2 : *PUD1-0

Setting for whether to execute internal pull-up/pull-down of the pin. This can be set by making changes on the *PUD. Pull-up/pull-down setting of each pin for the values of *PUDm1-m0 is as shown below.

Port6 Pull-up/pull-down settings

	*PUD1-0 = 00 b	*PUD1-0 = 01 b	*PUD1-0 = 10 b	*PUD1-0 = 11 b
	Drive capability	Drive capability	Drive capability	Drive capability
SDCMD0	PU and PD both OFF	Pull-UP	Pull-Down	-
SDAT03-0	PU and PD both OFF	Pull-UP	Pull-Down	-
SDRDATA15-0	PU and PD both OFF	-	Pull-Down	-
BMODE0	PU and PD both OFF	Pull-UP	Pull-Down	-
BMODE1	PU and PD both OFF	Pull-UP	Pull-Down	-
XTALINFO0	PU and PD both OFF	Pull-UP	-	-
XTALINFO1	PU and PD both OFF	Pull-UP	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

* “-” indicates setting prohibited.

5-3-3-38 SD IF Supporting Register : SDCTL

[Base + 0x0800]

Bit	31	30	29	28	27	26	25	24
Name	COREVLT	SDCLK0EN	SD2LOW	SDCLK2LOW	SD1LOW	SDCLK1LOW	SD0LOW	SDCLK0LOW
Reset	0	0	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	SDTIMERSEL2	IODRV21	IODRV20	MMCVLT2	ACSMODE22	ACSMODE21	ACSMODE20	SDMMC2
Reset	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	SDTIMERSEL1	IODRV11	IODRV10	MMCVLT1	ACSMODE12	ACSMODE11	ACSMODE10	SDMMC1
Reset	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	SDTIMERSEL0	IODRV01	IODRV00	MMCVLT0	ACSMODE02	ACSMODE01	ACSMODE00	SDMMC0
Reset	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31 : COREVLT
Selects LOGIC core voltage (Ch0, Ch1, and Ch2 common).
0 : 1.0 V (typ)
1: 1.2 V (typ)

Bit30 : SDCLK0EN
Controls SDCLK0 pin status
0 : Output
1 : Hi-Z

Bit25/27/29 : SDxLOW
Switches Chx SDCMDx/SDATx0-3 to forced low output.
0 : SDCMDx/SDATx0-3 in low output
1: SDCMDx/SDATx0-3 in normal output

Bit24/26/28 : SDCLKxLOW
Switches Chx SDCLKx to forced low output (Change must be made when clock is stopped)
0 : SDCLKx in low output
1: SDCLKx in normal output

Bit7/15/23 : SDTIMESELx
Selects to use either Timer for Chx SD or external timer (e.g. MTM)
0 : Use external timer (Timer for SD is invalid)
1: Use timer for SD

Bit6-5/14-13/22-21 : IODRVx1-0

This selection shall be made in accordance with Chx IO cell drive capability. (SD Card is valid only in UHS1, MMC is only valid when MMCVLTx=1.8 V)
0 : 6 mA (drive type D)
1 : Prohibited
2 : 8 mA (drive type C)
3 : 10 mA (drive type B)

* Please set Bit6-5 in accordance with SDCLK0DRV1-0, SDCMD0DRV1-0, and SDATA3-0DRV1-0 (Ch. 5-3-3-34 Drive Capability Switching Register 6 : PTDRVCNT6:PTDRVCNT6_Address). Operation cannot be guaranteed if there are differences.
* Please set Bit14-13 in accordance with P2DRV21-0 through P2DRV71-0 (Ch.5-3-3-32Drive Function Switching Register n: PTDRVCNTn:Port 2 pin drive current capability setting). Operation cannot be guaranteed if there are differences.
* Please set Bit22-21 in accordance with P0DRV41-0, P0DRV51-0, P0DRV91-0, P0DRVA1-0, P5DRV61-0, and P5DRV71-0 (Ch. 5-3-3-32 Drive Function Switching Register n: PTDRVCNTn:Port 0 Pin drive current capability setting and Ch. 5-3-3-33Drive Capability Switching Register5 : PTDRVCNT5: Port 5 pin drive current capability setting). Operation cannot be guaranteed if there are differences.

Bit4/12/20 : MMCVLTx
Selects the voltage to connect MMC (eMMC) to Chx (SD Card is invalid).
0 : 3.3 V
1: 1.8 V

Bit3-1/11-9/19-17 : ACSMODEx2-0
Selects mode to access Chx
0 : Normal speed (Card free, voltage free) * Initial operation (for card identification)
1 : Normal speed
2 : High speed
3 : UHS1(SDR12, SDR25, SDR50, DDR50)
4 : MMC-DDR

Bit0/8/16 : SDMMCx
Selects media to connect to Chx
0 : SD Card
1: MMC (eMMC)

* Invalid when ACSMODEx2-0=0 is set

5-3-3-39 SD IF Supporting Register : SDCTL2

[Base + 0x0804]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	SDMSSEL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	Reserved	SDBTCKSTP	SDBTSTS	Reserved	SDBTSTART	SDBTRST
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit16 : SDBTRST
Stops Boot by resetting Ch0 eMMC Physical Boot dedicated circuit
0 : Execute Reset
1: Release Reset

Bit17 : SDBTSTART
Indicates the acquisition of Ch0 eMMC Physical boot ID code
0 : With code identification
1: No code identification

Bit19 : SDBTSTS
Indicates Data read access state to eMMC at the time of Ch0 eMMC Physical boot
0 : Accessing eMMC
1: Stop accessing eMMC

Bit20 : SDBTCKSTP
Stops SDCLK to enable SD clock change after identifying Ch0 eMMC Physical boot code. Be sure to change SD clock before transferring data to release the stop state.
0 : Do not release SDCLK stop state
1: Release SDCLK stop state

Bit24 : SDMSSEL
SD I/F Ch1 and MS I/F are used in Ch1. Need to select either one.
0 : SD I/F Ch1
1: MS I/F

5-3-3-40 DREQ Selection Register 0-3 : DRQSEL0-3

[Base + 0x0808 - 0x0814]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	DRQnSEL 4	DRQnSEL 3	DRQnSEL 2	DRQnSEL 1	DRQnSEL 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	DRQnSEL 4	DRQnSEL 3	DRQnSEL 2	DRQnSEL 1	DRQnSEL 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	DRQnSEL 4	DRQnSEL 3	DRQnSEL 2	DRQnSEL 1	DRQnSEL 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	DRQnSEL 4	DRQnSEL 3	DRQnSEL 2	DRQnSEL 1	DRQnSEL 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit28-24, 20-16, 12-8, 4-0 : DRQnSEL4-0

Selects DMAC request signal. N=0~15 and “n” represents the amount of DMAC request bits. Register values and request signals to be selected are as follows:

DMAC Request setting 0

DMAC Request	DRQnSEL4-0
Uart0 Reception	0
Uart0 Transmission	1
Uart1 Reception	2
Uart1 Transmission	3
Uart2 Reception	4
Uart2 Transmission	5
SIO Reception	6
SIO Transmission	7
AudioBuf 0	8
AudioBuf 1	9
AudioBuf 2	10
AudioBuf 3	11
AudioBuf 4	12
AudioBuf 5	13
AudioBuf 6	14
AudioBuf 7	15

DMAC Request	DRQnSEL4-0
Reserved	16
Reserved	17
Reserved	18
Reserved	19
Reserved	20
Reserved	21
USB Device CH1	22
USB Device CH2	23
Reserved	24
Reserved	25
Reserved	26
Reserved	27
Reserved	28
Reserved	29
Reserved	30
Reserved	31

5-3-3-41 CACHE Control Register : CACHE_CTL

[Base + 0x0818]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	Reserved	use_cache
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W

This is a cache control register for S-Flash (Serial flash).
Reserved

Bit1 : Reserved
don't use

Bit0 : use_cache
0 : Invalid cache (Flashes cache)
1 : Valid cache

Attention: Do not change register values while READ access is occurring for S-Flash I/F (memory, cache) area.

5-3-3-42 BMODE Pin I/O Control Register : BMODE_CNT

[Base + 0x081C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	XTAL INFO1EN	XTAL INFO0EN	-	-	BMODE1 EN	BMODE0 EN
Reset	0	0	1	1	0	0	1	1
R/W	R	R	R/W	R/W	R	R	R/W	R/W

I/O control register for XTALINFO pin
Bit5 : XTALINFO1EN
0 : Set XTALINFO1 pin as output
1: Set XTALINFO1 pin as input (Default)

Bit4 : XTALINFO0EN
0 : Set XTALINFO0 pin as output
1: Set XTALINFO0 pin as input (Default)

I/O control register for BMODE pin
Bit1 : BMODE1EN
0 : Set BMODE1 pin as output
1: Set BMODE1 pin as input (Default)

Bit0 : BMODE0EN
0 : Set BMODE0 pin as output
1: Set BMODE0 pin as input (Default)

5-3-3-43 BMODE Data Register : BMODE_DT

[Base + 0x0820]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	XTAL INFO1DT	XTAL INFO0DT	-	-	BMODE1 DT	BMODE0 DT
Reset	0	0	x	x	0	0	x	x
R/W	R	R	R/W	R/W	R	R	R/W	R/W

A data register for XTALINFO pin.

Bit5-4 : XTALINFO1-0DT

This is a register which treats WRITE and READ data of XTALINFO1 and 0. This register will hold data to be outputted to XTALINFO1 and 0 at WRITE. In READ, external pins of XTALINFO1 and 0 can be read. In reset, it will be in the Input state, so the values to be read will vary by the pin condition.

A data register for BMODE pin.

Bit1-0 : BMODE1-0DT

This is a register which treats WRITE and READ data of BMODE1 and 0. This register will hold data to be outputted to BMODE1 and 0 at WRITE. In READ, external pins of BMODE1 and 0 can be read. In reset, it will be in the Input state, so the values to be read will vary by the pin condition.

5-3-3-44 GPO Function Control Register : GPO_CNT

[Base + 0x0824]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	GPOCNT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

This register controls the use of DAMP pin as GPO.
Bit0 : GPOCNT
0 : Use LOUT and ROUT pins as DAMP output (Default)
1: Use LOUT and ROUT pins as GPO

5-3-3-45 GPO DAMP Control Register : GPO_DZCTL

[Base + 0x0828]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	GPO DSL1	GPO DSL0	-	-	-	GPO DSLEEP
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R/W

This is a control register of DAMP when selecting GPO. This is only valid in [Base + 0x0824] GPOCNT=1.

Bit5-4 : GPODSL1-0
D-class Amp MOS transistor ON/OFF control
00 b : zero time gap
01 b : :2 ns time gap
10 b : 4 ns time gap
11 b : 6 ns time gap

Attention) Time gap is a designed value.

Bit0 : GPODSLEEP
D-class Amp sleep state
0 : Sleep
1 : Normal operation

5-3-3-46 GPO Drive Capability Setting Register : GPO_DZINP

[Base + 0x082C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	DZINP17	DZINP16
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DZINP15	DZINP14	DZINP13	DZINP12	DZINP11	DZINP10	DZINP9	DZINP8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DZINP7	DZINP6	DZINP5	DZINP4	DZINP3	DZINP2	DZINP1	DZINP0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This is a drive capability setting register of DAMP when GPO is selected. Valid in [Base + 0x0824] GPOCNT=1 only.

Bit17-0 : DZINP17-0
D-class Amp drive capability setting (ON resistance).

Remarks) D-class Amp drive capability (ON resistance) can be changed in the range of 0x00000 ~ 0x3ff00. ON resistance will be smaller with a larger value. If the value is set to “0”, ON resistance will be at its maximum and D-class Amp driver output will become Hi-z.

5-3-3-47 GPO Data Output Register : GPO_DOUT

[Base + 0x0830]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	GPO LOUT	GPO ROUT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

This data register is for using DAMP pin (LOUT, ROUT) as GPO.

Bit1-0 : GPOLOUT, GPOROUT

This is a register which treats WRITE data of LOUT and ROUT pins. This register holds the data to be outputted to LOUT and ROUT at WRITE. In READ, that register value can be read. LOUT and ROUT are output-only pins, thus it does not mean they can read input-data values at READ.

5-3-3-48 USB Control Register : USBCNT

[Base + 0x0834]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	RSM CONT	APP_PRT_ OVRCUR	SYS_INTE RRUPT	VBUS VALID
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	CRYCNT SW2	CRYCNT SW1	CRYCNT SW0	CLKSEL2	CLKSEL1	CLKSEL0	ANPD	HF_SEL
Reset	0	0	1	1	1	0	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This is an USB control register.

Bit0 : HF_SEL
Switches USB host and USB device
0 : USB device
1 : USB host

Be sure to switch USB host/USB device state when USB communication is in inactive state.
“Inactive state” means…
USB device : DEVC register (Base+200h) disconnect (bit5) is “1”
USB host : PORTSC_1 register (Base+54h) PP (bit12) is “0”

Also, soft reset must be executed for the mode to be in operation after switching the state.
<Soft reset register>
USB device : CONF register (Base+0h) soft_reset (bit2)
USB host : PHY mode setting1 register (Base+4h) RESET (bit31)

* Precautions
Do not set HF_SEL to “1” (USB host operation) when VBUS is supplied from the counter host.
<VBUS supply state>
USB Device : OTGSTS register (Base+310h) vbus_vld (bit10) is “1”

An internal p_vbusvalid_i input value of USB device will be reflected to vbus_vld.

Bit1 : ANPD
Controls the power down for PHY block analog circuit
0 : No power down
1 : Power down

It is prohibited to execute power down during USB communication.

Bit4-2 : CLKSEL2-0
Selects PHY PLL reference clock frequency to be used for USB host/USB device. Select 48 MHz for the USB host. A value needs to be selected according to the reference clock frequency for the USB device.
000 : 48 MHz
001 : 24 MHz
010 : 12 MHz
011 : Prohibited
100 : Prohibited
101 : 40 MHz
110 : 20 MHz
111 : Prohibited

Bit7-5 : CRYCNTSW2-0
Selects PHY ripple counter wait time to be used for USB host/USB device. Select 48 MHz for the USB host. A value needs to be selected according to the reference clock frequency for the USB device.
000 : 48 MHz,40 MHz
001 : 24 MHz,20 MHz
010 : 12 MHz
011 ~ 111 : Prohibited

Bit8 : VBUSVALID
This is a register for the USB device to be set when detecting “valid” in VBUS. This bit will be connected to USB device’s internal p_vbusvalid_i input.
0 : No VBUS VALID detection
1 : With VBUS VALID detection

Bit9 : SYS_INTERRUPT
This is a register for USB host which will be set when the upper system detects some sort of a system error. This bit will be connected to USB host’s internal sys_interrupt_i input.
0 : No system error
1 : With system error

Bit10 : APP_PRT_OVRCUR
This is a register for USB host to be set when detecting VBUS over-current. This bit will be connected to USB host’s internal app_prt_ovrcur_i input.
0 : No over-current detection
1 : With over-current detection

Bit11 : RSMCONT
This is a register for USB device which controls USB resume interrupt by the USB bus state
0 : Prohibit USB resume interrupt
1 : Permit USB resume interrupt

5-3-3-49 USB Status Register : USBSTAT

[Base + 0x0838]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	EHCI_PRT_PWR	LINESTE1	LINESTE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

This is an USB status register.

Bit1-0 : LINESTE1-0
This is a Read-only register for USB device which can read USB bus status.
Bit1 : DM
Bit0 : DP

Bit2 : EHCI_PRT_PWR
This is a Read-only register for USB host which can read VBUS supply condition. Able to monitor ehci_prt_pwr_o output inside the USB host.
0 : Indicates VBUS supply is impossible
1 : Indicates VBUS supply is possible

5-3-3-50 S-Flash I/F Selection Register : SFIFSEL

[Base + 0x083C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	SFIFSEL3	SFIFSEL2	SFIFSEL1	SFIFSEL0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

This is an S-Flash delay adjustment setting register. Please execute appropriate setting based on the working voltage and a device to be in use.

Bit3 : SFIFSEL3

Selects either Voltage-free or Use-device-free state. When one is set, SFIFSEL2-0 is ignored and will operate regardless of the internal voltage or the I/O voltage, but the operation frequency will drop. Normally, this selection shall be made in the BOOT where the used voltage or devices are unidentified.

0 : Validate SFIFSEL2-0 and will not set voltage-free nor use-device-free

1: Validate SFIFSEL2-0 and set voltage-free and use-device-free

Bit2 : SFIFSEL2

Selects targeted S-Flash. The targeted S-Flash specification is as follows:

0 : Selects Setting mode 0 specification

1: Selects Setting mode 1 specification

Setting mode	Parameter	min	max	Unit
0	Input Setup Time	7	-	ns
	Input Hold Time	5	-	ns
	Output Valid from CLK	-	8	ns
	Output Hold Time	0	-	ns
1	Input Setup Time	5	-	ns
	Input Hold Time	5	-	ns
	Output Valid from CLK	-	7	ns
	Output Hold Time	0	-	ns

Bit1 : SFIFSEL1
Selects the internal voltage. This shall be set according to the voltage in use.
0 : Use 1.0 V (0.93 V ~ 1.1 V).
1: Use 1.2 V (1.1 V ~ 1.27 V)

Bit0 : SFIFSEL0
Selects the I/O voltage. This shall be set according to the voltage in use.
0 : Use 1.8 V (1.7 V ~ 1.95 V)
1: Use 3.3 V (2.7 V ~ 3.6 V)

Attention) Do not change register values when READ access is occurring to the S-Flash I/F(memory, cache) range.

5-3-3-51 SDRAMC I/F Selection Register : SDRAMIFSEL

[Base + 0x0840]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	reserved	SDRAMIFSEL1	SDRAMIFSEL0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit1 : SDRAMIFSEL1
Selects the internal voltage. This shall be set according to the used voltage.
0 : Use 1.0 V (0.93 V~1.1 V)
1: Use 1.2 V (1.1 V~1.27 V)

Bit0 : SDRAMIFSEL0
Selects the I/O voltage. This shall be set according to the used voltage.
0 : Use 1.8 V (1.7 V~1.95 V)
1: Use 3.3 V (2.7 V~3.6 V)

5-3-3-52 I2C Mode Selection Register : I2CMODE

[Base + 0x0844]}

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	I2CMODE 1	I2CMODE 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

This is a register to select either Open drain or Push-pull for SCL pin in I2C Ch0 and Ch1.

Bit1 : I2CMODE1
A register to select the SCL pin type for I2C Ch1.
Bit1 : Use as Push-pull
Bit0 : Use as Open drain (Default)

Bit0 : I2CMODE0
A register to select SCL type for I2C Ch0
Bit1 : Use as Push-pull
Bit0 : Use as Open drain (Default)

5-3-3-53 AudioPLL Selection Register : AUDPLLSEL

[Base + 0x0F00]}

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	AUDPLL SEL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

A register to select PLL to be used for Audio.

Bit0 : AUDPLLSEL
Selects either PLL2 (0.9~1.3 v) or PLL3 (2.7~3.6 v) for Audio PLL
Bit1 : Use PLL3 (2.7~3.6 v)
Bit0 : Use PLL2 (0.9~1.3 v)

* If the PLL is switched in this register, Bit10-8 of the AudPLL control register: AudPLLCNT (Refer to ProgrammersModel_AudioPLLv*.pdf) must also be changed.

When PLL2 is selected: AudPLLS2-0 (Bit10-8) = 101 b

When PLL3 is selected: AudPLLS2-0 (Bit10-8) = 110 b

This value may change after the sample evaluation.

5-3-3-54 Reserved

[Base + 0x0F04]}

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

5-3-3-55 Reserved

[Base + 0x0FFC]}

Bit	31	30	29	28	27	26	25	24
Name	Reserved	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

5-4 Operation

This section provides setting flow of the System controller.

5-4-1 Reset Functions

There are reset control functions for internal function blocks in SystemController. This sub-section provides described all reset functions with them of SystemController.

There are below system function resets in this LSI.

- Hardware reset(Power ON reset)
- Watch Dog Timer resets
- Lockup resets
- Software resets for function blocks (Cortex-M3 Core1, LPDSP, Peripheral)

Don’t use resets which isn’t describe in this sub-section.

The block that is initialized by the reset function is shown block number in under diagrams.

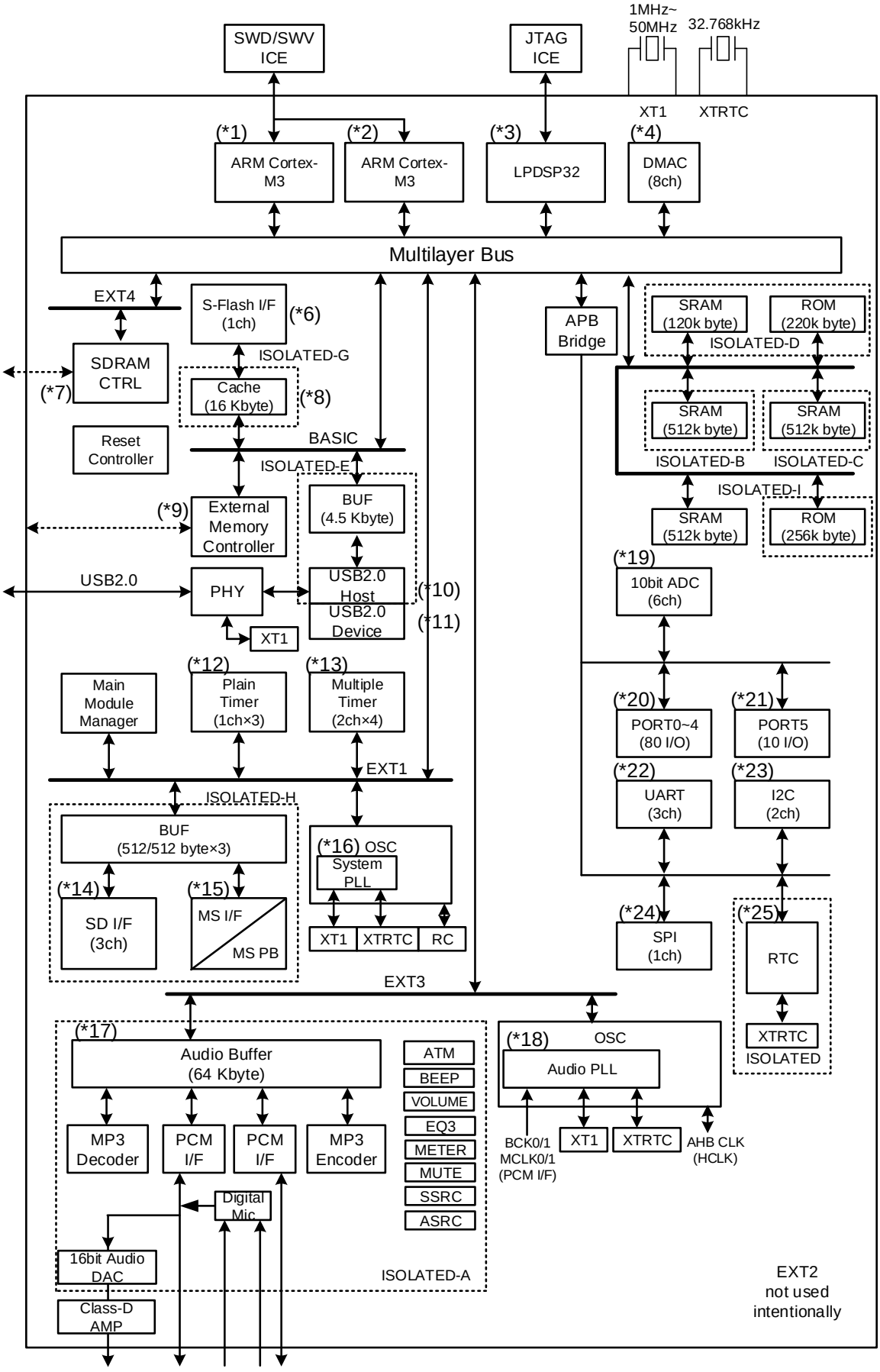


Figure 5-2 Block Diagram

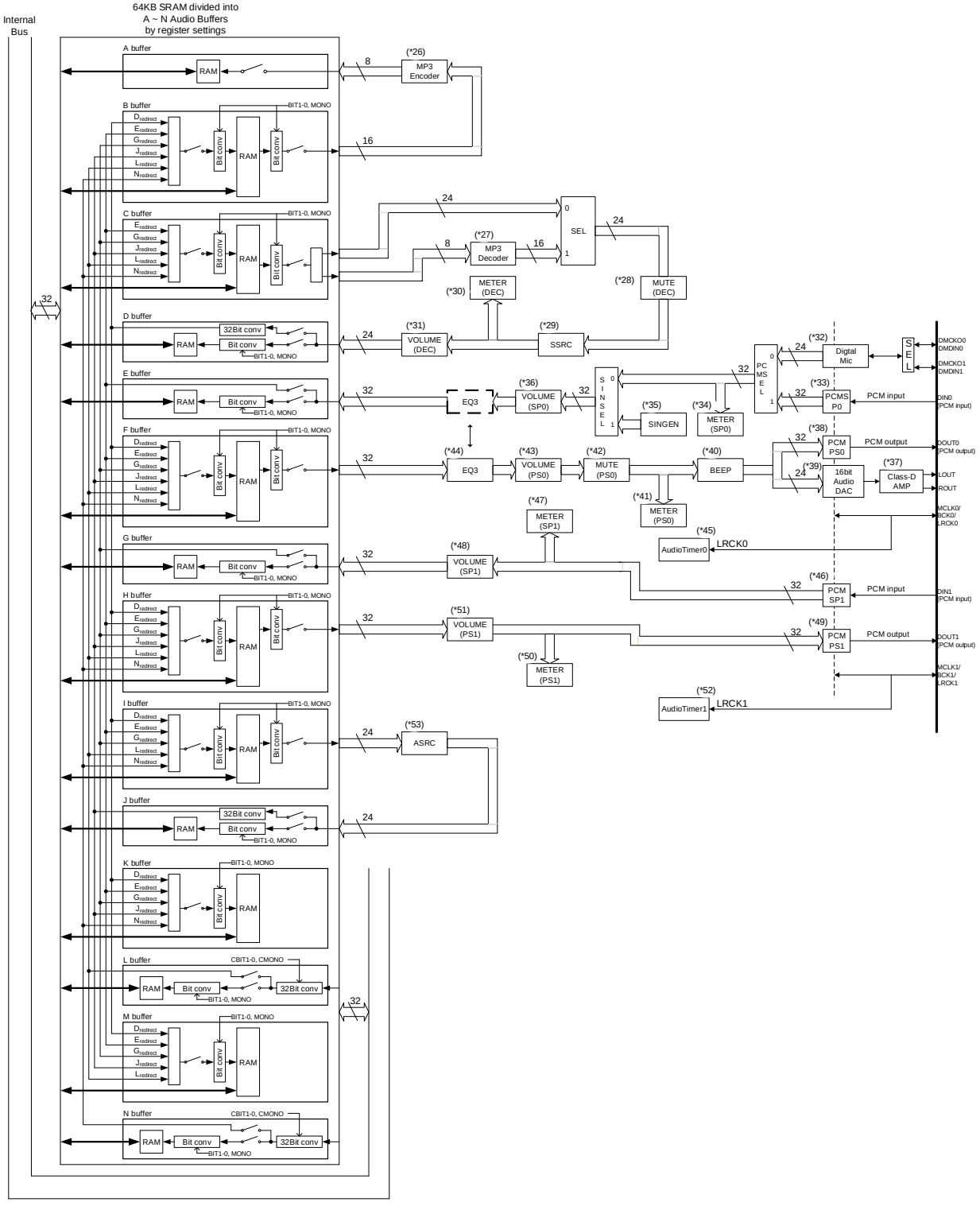


Figure 5-3 Audio Block Diagram

5-4-1-1 Hardware reset(Power ON reset)

The reset factor of hardware reset is input low level to NRES port.

The area which is initialized by hardware reset is below.

- The list of reset factor of hardware reset and the blocks which is initialized by it:

Reset factor	Blocks to be initialized
NRES Port=Low	From (*1) to (*53) (All blocks)

5-4-1-2 Watch Dog Timer resets

There are three Plain Timers in this LSI. Because there is an individual watch dog timer in these Plain Timers, in total there are three watch dog timers in this LSI. Watch Dog Timer0 can work for Cortex-M3 Core0, Watch Dog Timer1 can work for Cortex-M3 Core1 and Watch Dog Timer2 can work for LPDSP32.

The areas which are initialized by watch dog timer resets are below.

- The list of a reset factor of watch dog timer resets and the blocks which is initialized by them:

Reset factor	Blocks to be initialized
Watch Dog Timer0	From (*1) to (*53) (All blocks)
Watch Dog Timer1	(*2)Cortex-M3 Core1
Watch Dog Timer2	(*3)LPDSP32

* The connectivity information of the Watch Dog Timer resets are shown in Multi-Core Users Manual(ProgrammersModel-Multi-Core).

5-4-1-3 Lockup resets

If there is serious mistake in kernel software which work on Cortex-M3, Lockup may be caused by Cortex-M3. There is a function which some blocks are initialized by Lockup reset in this LSI.

* Please refer to [DDI0403D_arm_architecture_v7m_reference_manual.pdf] for the detail confitions of Lockup caused on archtecter of ARMv7-M.

- The list of a reset factor of Lockup resets and the blocks which is initialized by them:

Reset factor	Blocks to be initialized
Core0 Lockup	From (*1) to (*53) (All blocks)
Core1 Lockup	0x4008_003C Bit0 C1LUPC == Low:(*2)Cortex-M3 Core1
	0x4008_003C Bit0 C1LUPC == High:From (*1) to (*53) (All blocks)

* The connectivity information of the Lockup resets are shown in Multi-Core Users Manual(ProgrammersModel-Multi-Core).

5-4-1-4 Software resets for function blocks (Cortex-M3 Core1, LPDSP, Peripheral)

There is an individual software reset function which can initialize internal function blocks(From (*1) to (*53)) in this LSI. Their reset factor and the blocks to be initialized are shown under lists.

- The list of the reset factor of software resets and the blocks which is initialized by them:

Register address	Reset factor	Blocks to be initialized
Cortex-M3 Core0 0xE000ED0C Bit2	Core0 SYSRESETREQ=1	(*1) to (*53) (All blocks)
Cortex-M3 Core1 0xE000ED0C Bit2	Core1 SYSRESETREQ=1	(*2)ARM Cortex-M3 Core1
0x4008_0000 Bit1	C1RSTN=0	(*2)ARM Cortex-M3 Core1
0x4008_0000 Bit9	DSPRSTN=0	(*3)LPDSP32
0x4008_0114 Bit0	EXTMEMCRSTB=0	(*9)External Memory Controller
0x4008_0114 Bit1	SFIFRSTB=0	(*6)S-Flash I/F
0x4008_0114 Bit2	USBRSTB=0	(*10)USB2.0 Host
0x4008_0114 Bit3	IRQCNTRSTB=0	IRQ Controller
0x4008_0114 Bit4	DMACRSTB=0	(*4)DMAC
0x4008_0114 Bit8	MUTEXRSTB=0	Mutex Register Block
0x4008_0114 Bit9	DSPCMDRSTB=0	DSP Command I/F Block
0x4008_0114 Bit10	CACHERSTB=0	(*8)Cache
0x4008_0114 Bit11	USBDEVRSTB=0	(*11)USB2.0 Device
0x4008_0118 Bit0	MTM0RSTB=0	(*13)-0 Multiple Timer ch0
0x4008_0118 Bit1	MTM1RSTB=0	(*13)-1 Multiple Timer ch1
0x4008_0118 Bit2	MTM2RSTB=0	(*13)-2 Multiple Timer ch2
0x4008_0118 Bit3	MTM3RSTB=0	(*13)-3 Multiple Timer ch3
0x4008_0118 Bit4	PTM0RSTB=0	(*12)-0 Plain Timer ch0
0x4008_0118 Bit5	PTM1RSTB=0	(*12)-1 Plain Timer ch1
0x4008_0118 Bit6	PTM2RSTB=0	(*12)-2 Plain Timer ch2
0x4008_0118 Bit8	SDIF0RSTB=0	(*14)-0 SD I/F ch0
0x4008_0118 Bit9	SDIF1RSTB=0	(*14)-1 SD I/F ch1
0x4008_0118 Bit10	SDIF2RSTB=0	(*14)-2 SD I/F ch2
0x4008_0118 Bit11	MSIF0RSTB=0	(*15)MS Host & MS PB
0x4008_011C Bit0	AudioBufRSTB=0	(*17)Audio Buffer
0x4008_0120 Bit0	SDRAMCRSTB=0	(*7)SDRAM CTRL
0x4008_0124 Bit0	PORT0RSTB=0	(*20)-0 PORT0
0x4008_0124 Bit1	PORT1RSTB=0	(*20)-1 PORT1

0x4008_0124 Bit2	PORT2RSTB=0	(*20)-2 PORT2
0x4008_0124 Bit3	PORT3RSTB=0	(*20)-3 PORT3
0x4008_0124 Bit4	PORT4RSTB=0	(*20)-4 PORT4
0x4008_0124 Bit5	PORT5RSTB=0	(*20)-5 PORT5
0x4008_0124 Bit6	ADCRSTB=0	(*19)10 bit A/D
0x4008_0124 Bit7	SPIRSTB=0	(*24)SPI
0x4008_0124 Bit8	I2C0RSTB=0	(*23)-0 I2C ch0
0x4008_0124 Bit9	I2C1RSTB=0	(*23)-1 I2C ch1
0x4008_0124 Bit10	UART0RSTB=0	(*22)-0 UART ch0
0x4008_0124 Bit11	UART1RSTB=0	(*22)-1 UART ch1
0x4008_0124 Bit12	UART2RSTB=0	(*22)-2 UART ch2
0x4008_0124 Bit13	RTCRSTB=0	(*25)Only RTC APB I/F circuit
0x4006_4004 Bit0	ZARST=0	(*26)~(*53) All audio blocks except audio buffer *A
0x4006_4004 Bit1	RSTMP3E=0	(*26)MP3 Encoder *A
0x4006_4004 Bit2	RSTMP3D=0	(*27)MP3 Decoder *A
0x4006_4004 Bit3	RSTMUTEDEC=0	(*28)MUTE(DEC) *A
0x4006_4004 Bit4	RSTSSRC=0	(*29)SSRC *A
0x4006_4004 Bit5	RSTMTRDEC=0	(*30)METER(DEC) *A
0x4006_4004 Bit6	RSTVOLDEC=0	(*31)VOLUME(DEC) *A
0x4006_4004 Bit7	RSTDGMIC=0	(*32)Digital MIC *A
0x4006_4004 Bit8	RSTPCM0SP=0	(*33)PCMSP0 *A
0x4006_4004 Bit9	RSTMTRSP0=0	(*34)METER(SP0) *A
0x4006_4004 Bit10	RSTSINGEN=0	(*35)SINGEN *A
0x4006_4004 Bit11	RSTVOLSP0=0	(*36)VOLUME(SP0) *A
0x4006_4004 Bit12	RSTEQ3=0	(*44)EQ3 *A
0x4006_4004 Bit13	RSTVOLPS0=0	(*43)VOLUME(PS0) *A
0x4006_4004 Bit14	RSTMUTEPS0=0	(*42)MUTE(PS0) *A
0x4006_4004 Bit15	RSTMTRPS0=0	(*41)METER(PS0) *A
0x4006_4004 Bit16	RSTBEEP=0	(*40)BEEP *A
0x4006_4004 Bit17	RSTPCM0PS=0	(*38)PCMPS0 *A
0x4006_4004 Bit18	RSTDAC=0	(*39)DAC *A
0x4006_4004 Bit19	RSTDAMPCTL=0	(*37)Class -D AMP Control *A
0x4006_4004 Bit20	RSTATM0=0	(*45)AudioTimer0 *A
0x4006_4004 Bit21	RSTPCM1SP=0	(*46)PCMSP1 *A
0x4006_4004 Bit22	RSTMTRSP1=0	(*47)METER(SP1) *A
0x4006_4004 Bit23	RSTVOLSP1=0	(*48)VOLUME(SP1) *A
0x4006_4004 Bit24	RSTVOLPS1=0	(*51)VOLUME(PS1) *A

0x4006_4004 Bit25	RSTMTRPS1=0	(*50)METER(PS1) *A
0x4006_4004 Bit26	RSTPCM1PS=0	(*49)PCMPS1 *A
0x4006_4004 Bit27	RSTATM1=0	(*52)AudioTimer1 *A
0x4006_4004 Bit28	RSTPCKGEN=0	PCM Clock Generator *A
0x4006_4004 Bit29	RSTASRC=0	(*53)ASRC *A
0x4004_1000 Bit[1:0]	PLL1STYB=0 PLL1RESETB=0	(*16)System PLL (Reset phase compare circuit only and VCO make standby. The control register and the clock divider's setting registers are not initialized.)
0x4004_2000 Bit[1:0]	PLL2STYB=0 PLL2RESETB=0	(*18)Audio PLL (Reset phase compare circuit only and VCO make standby. The control register and the clock divider's setting registers are not initialized.)

*A:It is a reset of function circuit except setting registers. Setting registers can initialize by hardware reset.

The register of 0xE000_ED0C address is a reset of Cortex-M3 Core.

Please refer DDI0403D_arm_architecture_v7m_reference_manual.pdf for details.

The registers of 0x4008_**** address is registers of this SystemController.

Please refer under sub-sections for delatils.

5-3-3-25Module Reset Control BASIC Register: MRSTCNTBASIC

5-3-3-26Module Reset Control EXT1 Register: MRSTCNTEXT1

5-3-3-27Module Reset Control EXT3 Register: MRSTCNTEXT3

5-3-3-28Module Reset Control EXT4 Register: MRSTCNTEXT4

5-3-3-29Module Reset Control APB Register: MRSTCNTAPB

The register of 0x4006_4000 address is a register of AudioControl.

Please refer AudioControl(AUDCTRL) Users Manual(ProgrammersModel_AUDCTRL) for details.

The register of 0x4004_1000 address is a register of SystemPLL.

Please refer System PLL controller Users Manual(ProgrammersModel_SystemPLL) for details.

The register of 0x4004_2000 address is a register of AudioPLL.

Please refer Audio PLLcontroller Users Manual(ProgrammersModel_AudioPLL) for details.

5-4-2 Clock control functions

There are clock ON/OFF control functions for internal function blocks in SystemController. This sub-section provides described clock ON/OFF functions include them of audio blocks with tem of SystemController.

In this LSI clocks which are generated in OSC block. The clocks which are provided to internal function blocks can be controlled ON/OFF. The internal function blocks which are need to operate must be provided the clock for them.

Below describe with reference to the block number in next diagram.

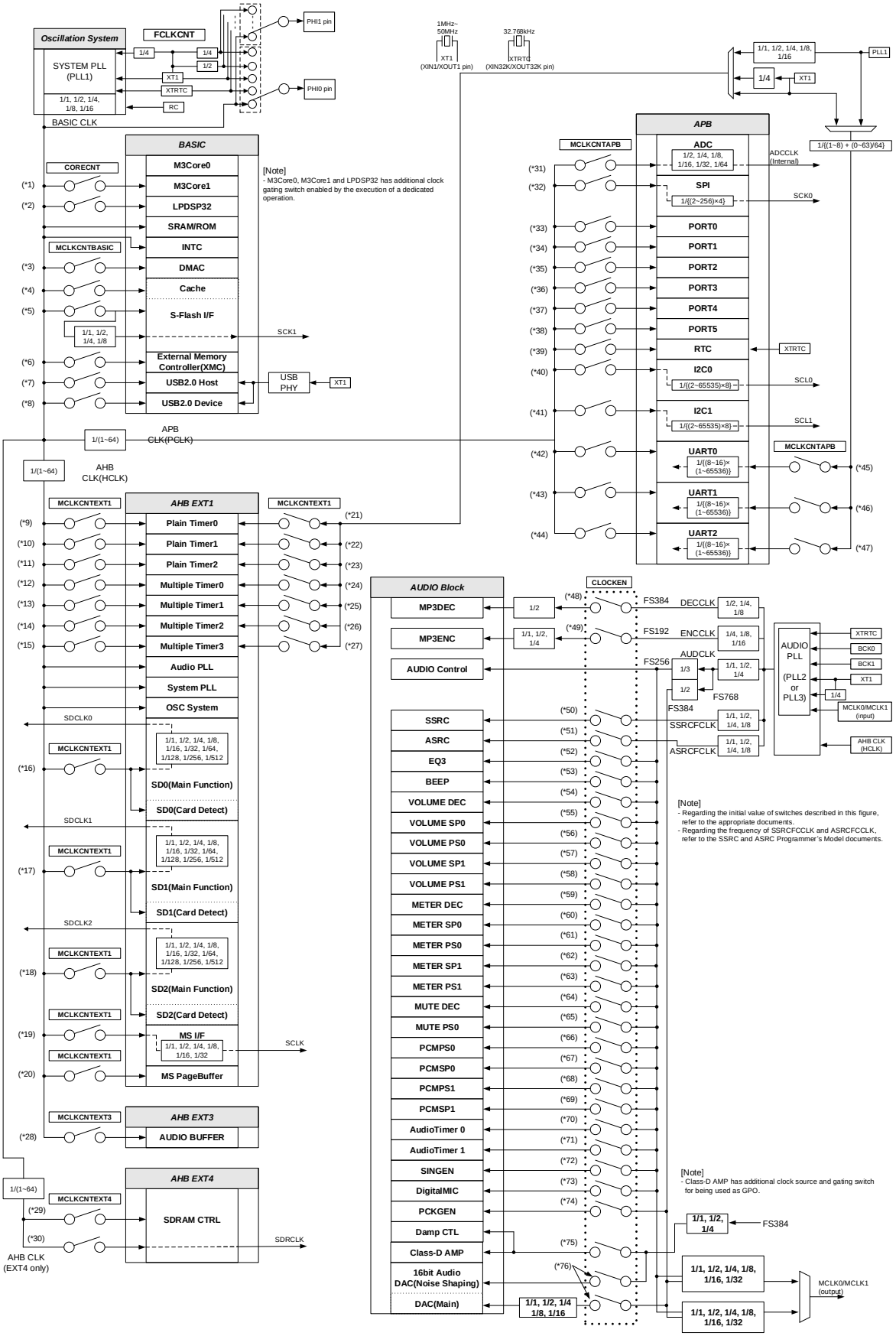


Figure 5-4 Clock System Diagram

5-4-2-1 Clock control for function blocks (Cortex-M3 Core1, LPDSP, Peripheral)

There are an individual clock ON/OFF control for (*1)~(*76) blocks in this LSI. Their clock control register and the blocks to be controlled are shown under lists. If the register is set 1 the blocks are provided clock but if it is set 0 the blocks aren't provided clock.

- The list of the clock control register and the blocks which is controlled by them:

Address	Register condition	Blocks to be controlled
0x4008_0000 Bit0	C1CLKEN=1	(*1)ARM Cortex-M3 Core1
0x4008_0000 Bit8	DSPCLKEN=1	(*2)LPDSP32
0x4008_0100 Bit0	EXTMEMCCLKEN=1	(*6)External Memory Controller
0x4008_0100 Bit1	SFIFCLKEN=1	(*5)S-Flash I/F
0x4008_0100 Bit2	USBHOSTCLKEN=1	(*7)USB2.0 Host
0x4008_0100 Bit4	DMACCLKEN=1	(*3)DMAC
0x4008_0100 Bit10	CACHECLKEN=1	(*4)Cache
0x4008_0100 Bit11	USBDEVCLKEN=1	(*8)USB2.0 Device
0x4008_0104 Bit0	MTM0CLKEN=1	(*12)Multiple Timer ch0
0x4008_0104 Bit1	MTM1CLKEN=1	(*13)Multiple Timer ch1
0x4008_0104 Bit2	MTM2CLKEN=1	(*14)Multiple Timer ch2
0x4008_0104 Bit3	MTM3CLKEN=1	(*15)Multiple Timer ch3
0x4008_0104 Bit4	PTM0CLKEN=1	(*9)Plain Timer ch0
0x4008_0104 Bit5	PTM1CLKEN=1	(*10)Plain Timer ch1
0x4008_0104 Bit6	PTM2CLKEN=1	(*11)Plain Timer ch2
0x4008_0104 Bit8	SDIF0CLKEN=1	(*16)SD I/F ch0
0x4008_0104 Bit9	SDIF1CLKEN=1	(*17)SD I/F ch1
0x4008_0104 Bit10	SDIF2CLKEN=1	(*18)SD I/F ch2
0x4008_0104 Bit11	MSHOSTCLKEN=1	(*19)MS Host
0x4008_0104 Bit12	MSPBCLKEN=1	(*20)MS PB
0x4008_0104 Bit24	MTM0CCLKEN=1	(*24)Multiple Timer ch0 for count
0x4008_0104 Bit25	MTM1CCLKEN=1	(*25)Multiple Timer ch1 for count
0x4008_0104 Bit26	MTM2CCLKEN=1	(*26)Multiple Timer ch2 for count
0x4008_0104 Bit27	MTM3CCLKEN=1	(*27)Multiple Timer ch3 for count
0x4008_0104 Bit28	PTM0CCLKEN=1	(*21)Plain Timer ch0 for count
0x4008_0104 Bit29	PTM1CCLKEN=1	(*22)Plain Timer ch1 for count
0x4008_0104 Bit30	PTM2CCLKEN=1	(*23)Plain Timer ch2 for count
0x4008_0108 Bit0	AudioBufCLKEN=1	(*28)Audio Buffer
0x4008_010C Bit0	SDRAMCCLKEN0=1	(*29)SDRAM CTRL

0x4008_010C Bit1	SDRAMCCLKEN1=1	(*30)SDRAM CTRL for I/F
0x4008_0110 Bit0	PORT0CLKEN=1	(*33)PORT0
0x4008_0110 Bit1	PORT1CLKEN=1	(*34)PORT1
0x4008_0110 Bit2	PORT2CLKEN=1	(*35)PORT2
0x4008_0110 Bit3	PORT3CLKEN=1	(*36)PORT3
0x4008_0110 Bit4	PORT4CLKEN=1	(*37)PORT4
0x4008_0110 Bit5	PORT5CLKEN=1	(*38)PORT5
0x4008_0110 Bit6	ADCCLKEN=1	(*31)10 bit A/D
0x4008_0110 Bit7	SPICLKEN=1	(*32)SPI
0x4008_0110 Bit8	I2C0CLKEN=1	(*40)I2C ch0
0x4008_0110 Bit9	I2C1CLKEN=1	(*41)I2C ch1
0x4008_0110 Bit10	UART0CLKEN=1	(*42)UART ch0
0x4008_0110 Bit11	UART1CLKEN=1	(*43)UART ch1
0x4008_0110 Bit12	UART2CLKEN=1	(*44)UART ch2
0x4008_0110 Bit13	RTCCLKEN=1	(*39)RTC APB I/F
0x4008_0110 Bit14	GPIOCLKEN=1	DAMP in use as a GPO
0x4008_0110 Bit24	UART0IFCLKEN=1	(*45)UART ch0 for
0x4008_0110 Bit25	UART1IFCLKEN=1	(*46)UART ch1 for correspondence control
0x4008_0110 Bit26	UART2IFCLKEN=1	(*47)UART ch2 for correspondence control
0x4006_4000 Bit1	FCEMP3ENC=1	(*49)MP3 Encoder *A
0x4006_4000 Bit2	FCEMP3DEC=1	(*50)MP3 Decoder *A
0x4006_4000 Bit3	FCEMUTEDEC=1	(*64)MUTE(DEC) *A
0x4006_4000 Bit4	FCESSRC=1	(*50)SSRC *A
0x4006_4000 Bit5	FCEMTRDEC=1	(*59)METER(DEC) *A
0x4006_4000 Bit6	FCEVOLDEC=1	(*54)VOLUME(DEC) *A
0x4006_4000 Bit7	FCEDGMIC=1	(*73)Digital MIC *A
0x4006_4000 Bit8	FCEPCMSP0=1	(*67)PCMSP0 *A
0x4006_4000 Bit9	FCEMTRSP0=1	(*60)METER(SP0) *A
0x4006_4000 Bit10	FCESINGEN=1	(*72)SINGEN *A
0x4006_4000 Bit11	FCEVOLSP0=1	(*55)VOLUME(SP0) *A
0x4006_4000 Bit12	FCEEQ3=1	(*52)EQ3 *A
0x4006_4000 Bit13	FCEVOLPS0=1	(*56)VOLUME(PS0) *A
0x4006_4000 Bit14	FCEMUTEPS0=1	(*65)MUTE(PS0) *A
0x4006_4000 Bit15	FCEMTRPS0=1	(*61)METER(PS0) *A
0x4006_4000 Bit16	FCEBEEP=1	(*53)BEEP *A
0x4006_4000 Bit17	FCEPCMPS0=1	(*66)PCMPS0 *A
0x4006_4000 Bit18	FCEDAC=1	(*76)DAC *A

0x4006_4000 Bit19	FCEDAMPCTL=1	(*75)Class -D AMP Control *A
0x4006_4000 Bit20	FCEATM0=1	(*70)AudioTimer0 *A
0x4006_4000 Bit21	FCEPCMSP1=1	(*69)PCMSP1 *A
0x4006_4000 Bit22	FCEMTRSP1=1	(*62)METER(SP1) *A
0x4006_4000 Bit23	FCEVOLSP1=1	(*57)VOLUME(SP1) *A
0x4006_4000 Bit24	FCEVOLPS1=1	(*58)VOLUME(PS1) *A
0x4006_4000 Bit25	FCEMTRPS1=1	(*63)METER(PS1) *A
0x4006_4000 Bit26	FCEPCMPS1=1	(*68)PCMPS1 *A
0x4006_4000 Bit27	FCEATM1=1	(*71)AudioTimer1 *A
0x4006_4000 Bit28	FCEPCKGEN=1	(*74)PCM Clock Generator *A
0x4006_4000 Bit29	FCEASRC=1	(*51)ASRC *A

The registers of 0x4008_**** address is registers of this SystemController.

Please refer under sub-sections for delatils.

5-3-3-20Module Clock Control BASIC Register: MCLKCNTBASIC

5-3-3-21Module Clock Control EXT1 Register: MCLKCNTEXT1

5-3-3-22Module Clock Control EXT3 Register: MCLKCNTEXT3

5-3-3-23Module Clock Control EXT4 Register: MCLKCNTEXT4

5-3-3-24Module Clock Control APB Register: MCLKCNTAPB

The register of 0x4006_4000 address is a register of AudioControl.

Please refer AudioControl(AUDCTRL) Users Manual(ProgrammersModel_AUDCTRL) for details.

5-4-3 The setting flow of software reset control and clock ON/OFF control

After Power ON reset is released, the following is the condition of the function blocks which can be controlled by described 5-4-1-4Software resets for function blocks (Cortex-M3 Core1, LPDSP, Peripheral).

0xE000_ED0C SYSRESETREQ is 0 (The reset is released).

0x4008_**** Registers is 0 (Reset status)

0x4006_4004 Registers is 1 (Reset status)

0x4004_1000,0x4004_2000 Registers is 0 (Reset or standby status)

Also the condition of the function blocks which can be controlled by described 5-4-2-1Clock control for function blocks (Cortex-M3 Core1, LPDSP, Peripheral) is 0 (Clocks aren't probided).

Therefor it is necessary that the blocks which must work are provided clocks for them and are released resets for them, if you want to start operation. In its case please set register along following example flow.

Example : The setting flow to release a reset and to provide a clock for Cortex-M3 Core1

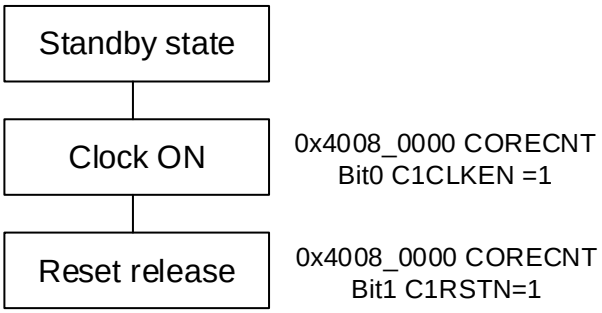


Figure 5-5

Also it is possible that the blocks which don't need to work are stoped clocks for them and are set reset for them if you want to reduce power consumption. In its case please set register along following example flow.

Example: The setting flow to set reset and to stop clock for Cortex-M3 Core1

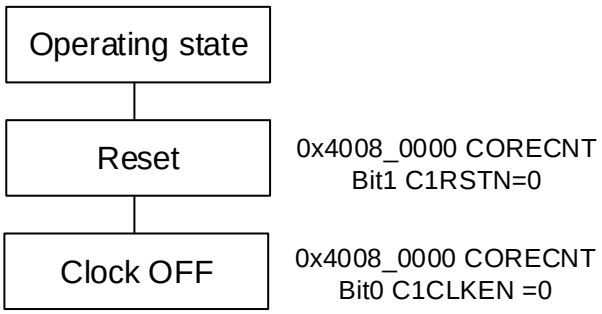


Figure 5-6

5-4-4 The setting flow of power supply control and isoration control for internal function blocks

There are power supply controls and isoration controls for internal function blocks in this LSI. If there are internal function blocks which don't need to work, power consumption of this LSI is reduced by using these functions.

There are power supply controllable 8 blocks which are described as from ISORATED-A to ISORATED-I in Figure 5-2. After Power ON reset is released, these blocks are supplied power and set non-isorate as blocks can work. Please set registers along following example flow if you want to make the block power non-supply and isorate.

Example: The setting flow to set Power OFF and isorate for ISOLATED-A

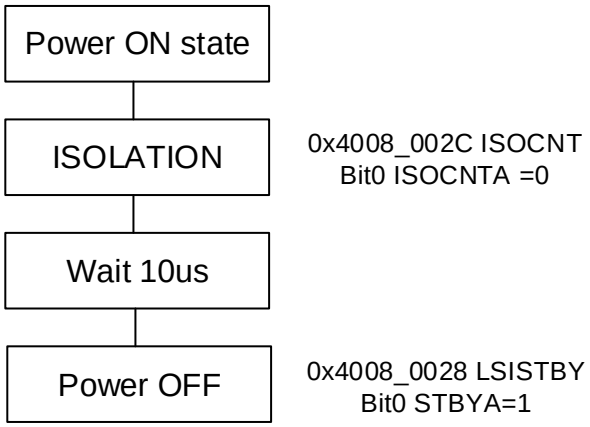


Figure 5-7

Also please set registers along following example flow if you want to make the block power supply and non-isorate.

Example: The setting flow to set Power ON and non-isorate for ISOLATED-A

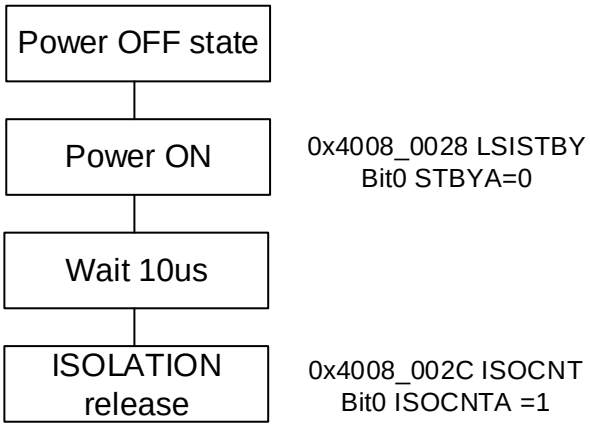


Figure 5-8

5-4-5 Operating Precautions

Please execute System controller setting by following this manual.

6 Multi-Core

6-1 Introduction

This chapter provides the outline of Multi-Core in the section below.

- The outline of Multi-Core

6-1-1 The outline of Multi-Core

Multi-Core is comprised of Cortex-M3 Core0 and Cortex-M3 Core1 and DSP(DSP name:LPDSP). A variety of information about Cortex-M3 Core0 and Cortex-M3 Core1 and LPDSP and the details about the Multi-Core constitution are described in this document.

6-2 Functional Overview

This chapter provides the outline of Multi-Core function block in the sections below.

- Multi-Core Circuit Configuration
- Cortex-M3 Core0/Cortex-M3 Core1 function
- LPDSP function

6-2-1 Multi-Core Circuit Configuration

6-2-1-1 Multi-core Reset System

Figure 6-1 shows Core reset configuration of the multi-core system used in this device.

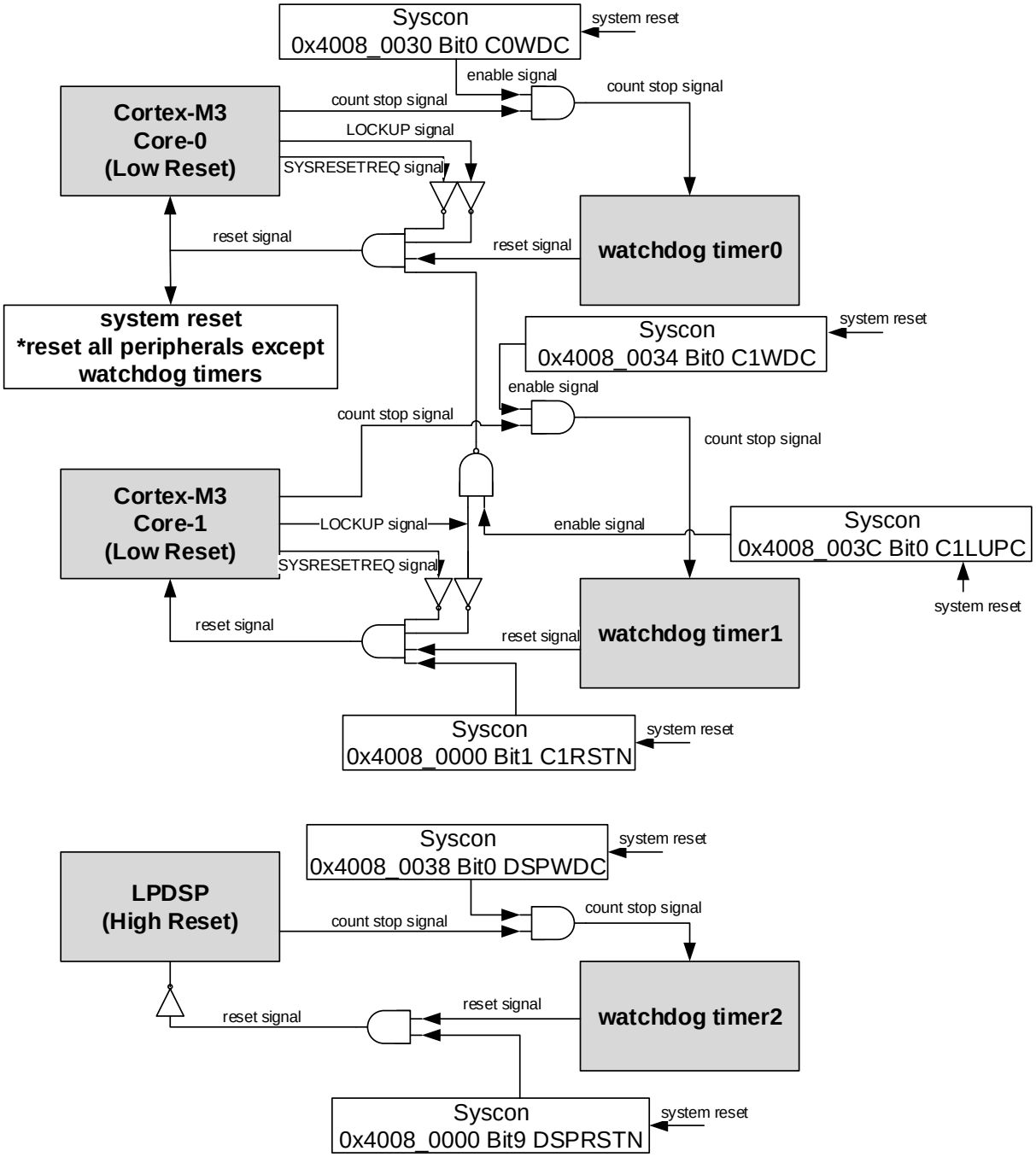


Figure 6-1

Power-on by the NRES-pin will reset all cores.

Aside from the above, the following reset factors exist per each core:

Core0: Watchdog reset by watchdog timer 0 (*1)

Reset by Core0 LOCKUP occurrence

Reset by Core1 LOCKUP occurrence (with enable control)

Core1: Watchdog reset by watchdog timer 1 (*1)

Reset by Core1 LOCKUP occurrence

Reset by 0x4008_0000 Bit1 C1RSTN register of SYSCON (*2)

LPDSP: Watchdog reset by watchdog timer 2 (*1)

Reset by 0x4008_0000 Bit9 DSPRSTN register of SYSCON (*2)

*1: Each watchdog timer can select whether to stop the watchdog count or not when the relevant core is debugged and stopped by Break. Corresponding bits are C0WDC, C1WDC and DSPWDC shown in Figure 6-1

*2: When Core0 is reset, system reset is affected to the entire device. This means SYSCON is also reset and register values are returned to the initial value. When register indicates initial value, it is reset. As a result, when Core0 is reset, Core1 and LPDSP are also reset.

6-2-2 Cortex-M3 Core0/Cortex-M3 Core1 function

6-2-2-1 Cortex-M3 configuration option set value

This table shows the configuration options for the Cortex-M3 implementation for this device.

Parameter	Set value	Comment
NUM_IRQ	65	IRQ input is set at 65 lines
LVL_WIDTH	5	Interrupt priority setting: 5 Bit
MPU_PRESENT	1	MPU is present
BB_PRESENT	1	Bit-banding is enabled
AHB_CONST_CTRL	0	Control signal at AHB-Lite bus wait is not maintained
DEBUG_LVL	3	Full debug function
TRACE_LVL	1	Standard trace
RESET_ALL_REGS	0	Resets only essential states
JTAG_PRESENT	0	SW-DP
CLKGATE_PRESENT	1	Clock gating
OBSERVATION	0	No internal state observation
WIC_PRESENT	1	WIC is implemented.
WIC_LINES	68	3+NUM_IRQ

6-2-2-2 Multiprocessor communication interface

Figure 6-2 shows the image of the multiprocessor communication interface. TXEV output of Cortex-M3 Core0 is connected to RXEV input of Cortex-M3 Core1. TXEV output of Core1 is connected to RXEV input of Core0.

Both processors can synchronize the task by entering into sleep by using WFE command, and issuing an event to release the sleep by using SEV command.

WFE command behavior is affected by the event latch in core. If the event latch is not set, it will enter into the sleep state. If the event latch is set, the latch is cleared and command execution does not enter into the sleep state and continues on. Please refer to ARM document such as Cortex-M3 WFE command document for more information about the event latch.

Image diagram of Multiprocessor Communication Interface

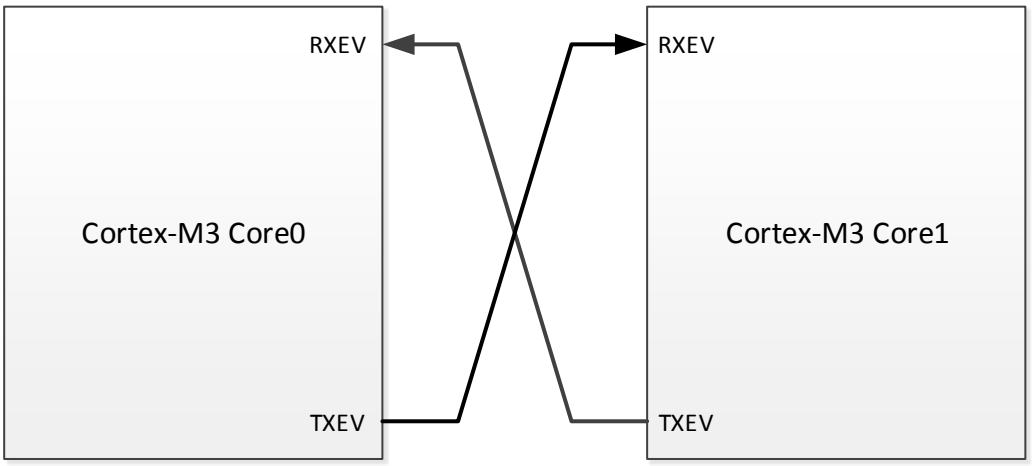


Figure 6-2

6-2-2-3 System Tick Timer

Figure 6-3 shows the image of a system tick timer. The system tick clock of 8.192 kHz is input to Cortex-M3 Core0 and Core1. The system tick timer can make a 10 ms timer by having 8.192 kHz clock as its reference.

When 32.768 kHz oscillation does not exist, select a core clock (1) as the clock source of the system tick timer with “CLKSOURCE” bit of “SysTick control and status register” in core. In this case, the external reference clock (0) cannot be used. See ARM document for more details.

“NOREF” bit (bit [31]) of “SysTick calibration value register” in core is fixed at “0” regardless of the existence of 32.768 kHz oscillation. Be sure not to use this NOREF bit when 32.768 kHz oscillation does not exist.

“SKEW” bit (bit [30]) of “SysTick calibration value register” is fixed at “0”, regardless of the accuracy of 10 ms configuration. Do not use this SKEW bit because error may occur when configuring 10 ms timer from 8.192 kHz.

“TENMS” bits (bits [23:0]) of “SysTick calibration value register” are fixed at “4Fh”. Error may occur when configuring 10 ms timer from 8.192 kHz. Therefore, in order to make 10 ms timer with these TENMS bits, set a value of TENMS bits + 2h (over 10 ms) or + 1h (under 10 ms)) in “RELOAD” bits of “SysTick reload value register” as shown below:

- Set TENMS bits + 2h in RELOAD bits (Approximately over 10 ms)

  Realizes (1 / 8.192 kHz) x ((4fh + 2h) + 1) = 10.00976 ms timer

- Set TENMS bits + 1h in RELOAD bits (Approximately under 10 ms)

  Realizes (1 / 8.192 kHz) x ((4fh + 1h) + 1) = 9.887695 ms timer

Image diagram of System Tick Timer

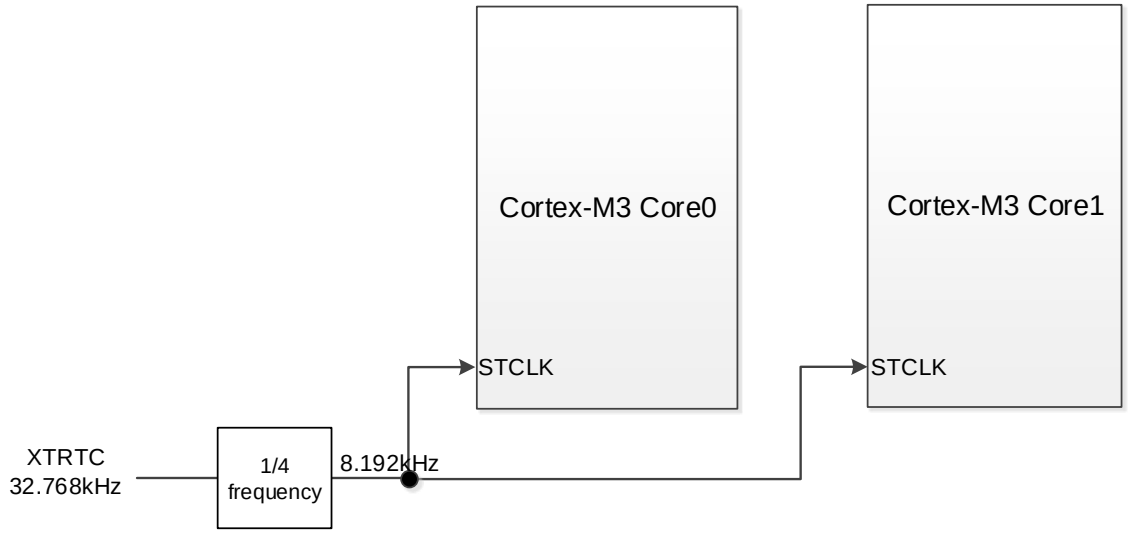


Figure 6-3

6-2-2-4 Endian

This device supports only little endian. Big endian is unsupported.

6-2-2-5 Exclusive commands

This device does not support exclusive commands (LDREX, STREX) of Cortex-M3 Core0 and Core1. There is a MUTEX hardware dedicated for the exclusive control. Please see MUTEX section for more details.

6-2-2-6 Reset output control

This device will reset by operating “SYSRESETREQ” bit of “Application interrupt and reset control register” of Cortex-M3 Core0 and Core1. However, Core0 will reset all peripherals, Core1 and LPDSP32 except watchdog timers, but Core1 will reset only itself.

6-2-2-7 SLEEP/SLEEPDEEP

This device supports the Sleep function and the Sleepdeep function of Cortex-M3 Core0 and Core1. The Sleep function can stop core clocks that can be stopped. The Sleepdeep function of this device has WIC function. The Sleepdeep function can stop the oscillation (OSC) itself, which is the clock oscillation source.

This device has Core0 and Core1 but the Sleep function has no dependency between these cores and both have an independent core operation control. On the other hand, the Sleepdeep function has dependency between the cores because it stops the oscillation. Thus, if Core1 is not used, the oscillation can be stopped when Core0 is in the Sleepdeep state. If both Core0 and Core1 are used, the oscillation can be stopped only when both cores are in the Sleepdeep state. Having one of the cores in the Sleepdeep state will not stop the oscillation. In this case, the core that is in the Sleepdeep state can only stop stop-enable clocks; just like Sleep function.

6-2-3 LPDSP function

6-2-3-1 The outline of LPDSP

LPDSP is 32-bit fixed point, dual-MAC low power original DSP optimized for audio processing.

6-3 Operation

This section provides information about Multi-Core setting flow.

6-3-1 Reset function

This section explains about the reset function of this device.

This device has the following reset functions:

- Hardware reset (Power ON reset)
- Watch Dog Timer reset
- Lockup reset
- Software reset of each block
  (Cortex-M3 Core1, LPDSP, and each Peripherals)

More explanation for each of these functions is written from the next page.

Please do not use any reset function that is not mentioned in here.

The reset range is also explained using block numbers in the figures used in this section.

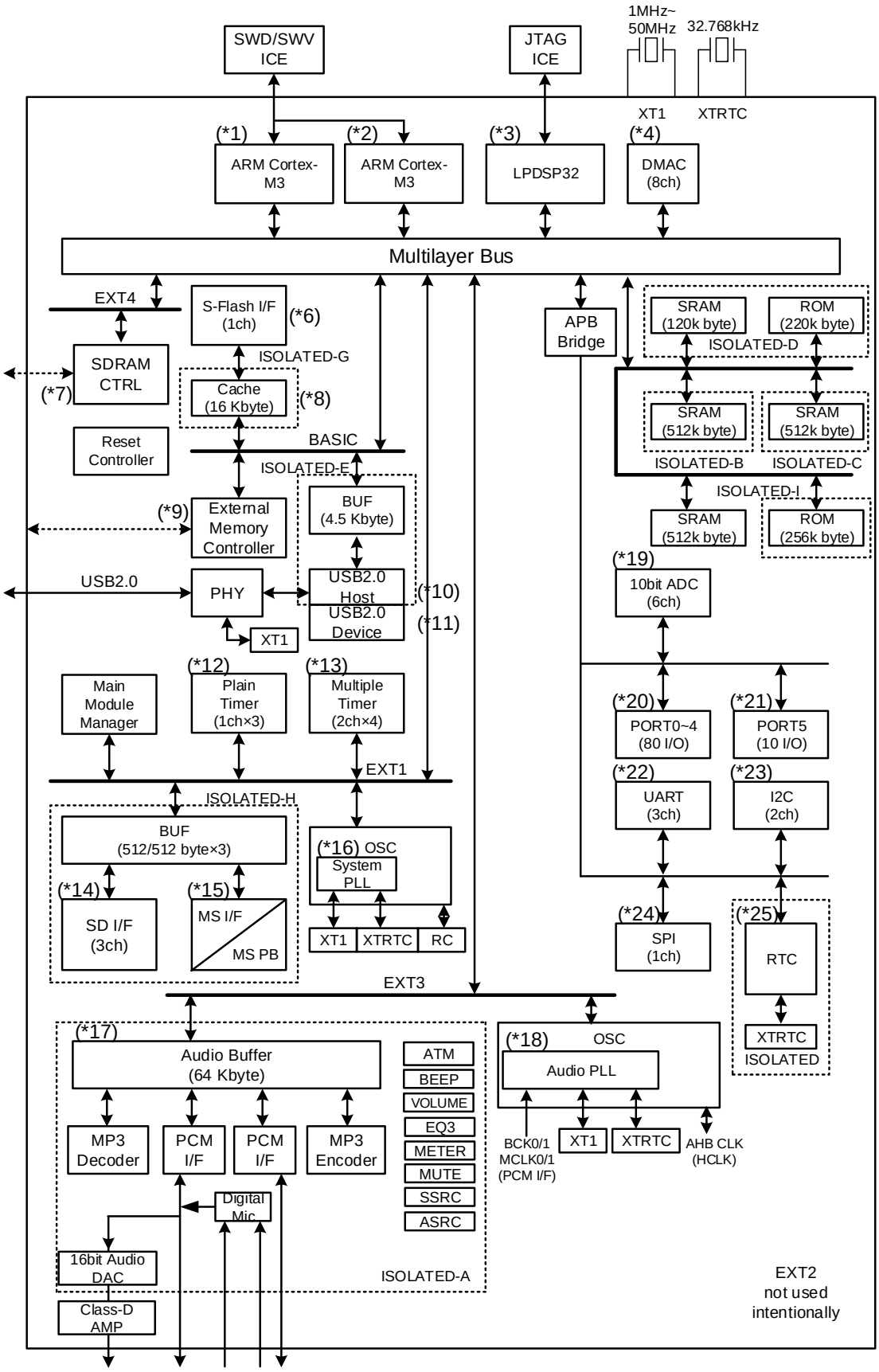


Figure 6-4 General block diagram

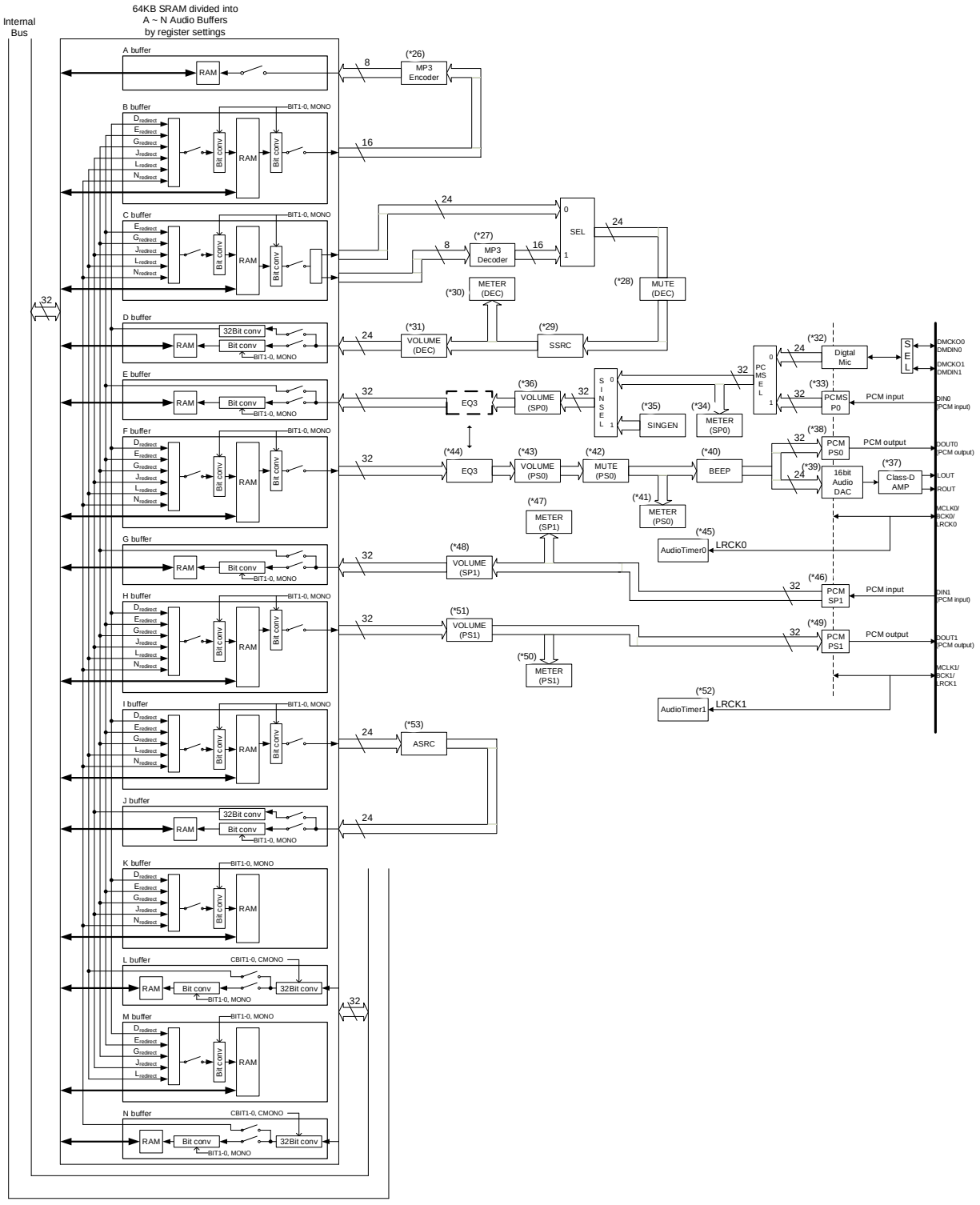


Figure 6-5 Audio block diagram

6-3-1-1 Hardware reset (Power ON reset)

Hardware reset is caused by low input to NRES pin.

Hardware reset initialization range is as below:

- Hardware reset types and the initialization range

Reset factor	Initialization range
NRES pin = Low	All blocks from (*1) to (*53)

6-3-1-2 Watch Dog Timer reset

This device has 3 channels of plain timers and each of those include a watchdog timer. Therefore, this device also has 3 channels of watchdog timers. Watch Dog Timer0 can be used for Cortex-M3 Core0, Watch Dog Timer1 can be used for Cortex-M3 Core1, and Watch Dog Timer2 can be used for LPDSP32.

Below table shows the initialization range of watchdog timer reset.

- Reset types of Watch Dog Timers and the initialization range

Reset factor	Initialization range
Watch Dog Timer0	All blocks from (*1) to (*53)
Watch Dog Timer1	(*2) Cortex-M3 Core1
Watch Dog Timer2	(*3) LPDSP32

* See 6-2-1-1Multi-core Reset System for the connection information of watchdog timer resets.

6-3-1-3 Lockup reset

Lockup may occur if Cortex-M3 kernel software has a significant error.

This device is configured to be reset with the occurrence of the lockup.

* See “ARMv7-M Architecture Reference Manual” for more details of lockup conditions in ARMv7-M architecture.

- Lockup reset types and the initialization range

Reset factor	Initialization range
Core0 Lockup	All blocks from (*1) to (*53)
Core1 Lockup	When 0x4008_003C Bit0 C1LUPC == Low: (*2) Cortex-M3 Core1
	When 0x4008_003C Bit0 C1LUPC == High: All blocks from (*1) to (*53)

* See 6-2-1-1Multi-core Reset System for the connection information about the lockup reset.

6-3-1-4 Software reset of each block (Cortex-M3 Core1, LPDSP, and each Peripherals)

This device has a software reset function which enables individual Reset and Reset release of the blocks from (*1) to (*53) in the ranges shown below:

- Types of software reset and the initialization range

Register address	Reset factor	Initialization range
0xE000ED0C Bit2 in Cortex-M3 Core0	Core0 SYSRESETREQ=1	All blocks from (*1) to (*53)
0xE000ED0C Bit2 in Cortex-M3 Core1	Core1 SYSRESETREQ=1	(*2)ARM Cortex-M3 Core1
0x4008_0000 Bit1	C1RSTN=0	(*2)ARM Cortex-M3 Core1
0x4008_0000 Bit9	DSPRSTN=0	(*3)LPDSP32
0x4008_0114 Bit0	EXTMEMCRSTB=0	(*9)External Memory Controller
0x4008_0114 Bit1	SFIFRSTB=0	(*6)S-Flash I/F
0x4008_0114 Bit2	USBRSTB=0	(*10)USB2.0 Host
0x4008_0114 Bit3	IRQCNTRSTB=0	IRQ Controller
0x4008_0114 Bit4	DMACRSTB=0	(*4)DMAC
0x4008_0114 Bit8	MUTEXRSTB=0	Mutex Register Block
0x4008_0114 Bit9	DSPCMDRSTB=0	DSP Command I/F Block
0x4008_0114 Bit10	CACHERSTB=0	(*8)Cache
0x4008_0114 Bit11	USBDEVRSTB=0	(*11)USB2.0 Device
0x4008_0118 Bit0	MTM0RSTB=0	(*13)-0 Multiple Timer ch0
0x4008_0118 Bit1	MTM1RSTB=0	(*13)-1 Multiple Timer ch1
0x4008_0118 Bit2	MTM2RSTB=0	(*13)-2 Multiple Timer ch2
0x4008_0118 Bit3	MTM3RSTB=0	(*13)-3 Multiple Timer ch3
0x4008_0118 Bit4	PTM0RSTB=0	(*12)-0 Plain Timer ch0
0x4008_0118 Bit5	PTM1RSTB=0	(*12)-1 Plain Timer ch1
0x4008_0118 Bit6	PTM2RSTB=0	(*12)-2 Plain Timer ch2
0x4008_0118 Bit8	SDIF0RSTB=0	(*14)-0 SD I/F ch0
0x4008_0118 Bit9	SDIF1RSTB=0	(*14)-1 SD I/F ch1
0x4008_0118 Bit10	SDIF2RSTB=0	(*14)-2 SD I/F ch2
0x4008_0118 Bit11	MSIF0RSTB=0	(*15)MS Host & MS PB
0x4008_011C Bit0	AudioBufRSTB=0	(*17)Audio Buffer
0x4008_0120 Bit0	SDRAMCRSTB=0	(*7)SDRAM CTRL
0x4008_0124 Bit0	PORT0RSTB=0	(*20)-0 PORT0
0x4008_0124 Bit1	PORT1RSTB=0	(*20)-1 PORT1
0x4008_0124 Bit2	PORT2RSTB=0	(*20)-2 PORT2

0x4008_0124 Bit3	PORT3RSTB=0	(*20)-3 PORT3
0x4008_0124 Bit4	PORT4RSTB=0	(*20)-4 PORT4
0x4008_0124 Bit5	PORT5RSTB=0	(*20)-5 PORT5
0x4008_0124 Bit6	ADCRSTB=0	(*19)10 bit A/D
0x4008_0124 Bit7	SPIRSTB=0	(*24)SPI
0x4008_0124 Bit8	I2C0RSTB=0	(*23)-0 I2C ch0
0x4008_0124 Bit9	I2C1RSTB=0	(*23)-1 I2C ch1
0x4008_0124 Bit10	UART0RSTB=0	(*22)-0 UART ch0
0x4008_0124 Bit11	UART1RSTB=0	(*22)-1 UART ch1
0x4008_0124 Bit12	UART2RSTB=0	(*22)-2 UART ch2
0x4008_0124 Bit13	RTCRSTB=0	(*25)RTC APB I/F circuit alone is reset.
0x4006_4004 Bit0	ZARST=0	(*26) to (*53) Whole audio block except an audio buffer *1
0x4006_4004 Bit1	RSTMP3E=0	(*26)MP3 Encoder *1
0x4006_4004 Bit2	RSTMP3D=0	(*27)MP3 Decoder *1
0x4006_4004 Bit3	RSTMUTEDEC=0	(*28)MUTE(DEC) *1
0x4006_4004 Bit4	RSTSSRC=0	(*29)SSRC *1
0x4006_4004 Bit5	RSTMTRDEC=0	(*30)METER(DEC) *1
0x4006_4004 Bit6	RSTVOLDEC=0	(*31)VOLUME(DEC) *1
0x4006_4004 Bit7	RSTDGMIC=0	(*32)Digital MIC *1
0x4006_4004 Bit8	RSTPCM0SP=0	(*33)PCMSP0 *1
0x4006_4004 Bit9	RSTMTRSP0=0	(*34)METER(SP0) *1
0x4006_4004 Bit10	RSTSINGEN=0	(*35)SINGEN *1
0x4006_4004 Bit11	RSTVOLSP0=0	(*36)VOLUME(SP0) *1
0x4006_4004 Bit12	RSTEQ3=0	(*44)EQ3 *1
0x4006_4004 Bit13	RSTVOLPS0=0	(*43)VOLUME(PS0) *1
0x4006_4004 Bit14	RSTMUTEPS0=0	(*42)MUTE(PS0) *1
0x4006_4004 Bit15	RSTMTRPS0=0	(*41)METER(PS0) *1
0x4006_4004 Bit16	RSTBEEP=0	(*40)BEEP *1
0x4006_4004 Bit17	RSTPCM0PS=0	(*38)PCMPS0 *1
0x4006_4004 Bit18	RSTDAC=0	(*39)DAC *1
0x4006_4004 Bit19	RSTDAMPCTL=0	(*37)Class -D AMP Control *1
0x4006_4004 Bit20	RSTATM0=0	(*45)AudioTimer0 *1
0x4006_4004 Bit21	RSTPCM1SP=0	(*46)PCMSP1 *1
0x4006_4004 Bit22	RSTMTRSP1=0	(*47)METER(SP1) *1
0x4006_4004 Bit23	RSTVOLSP1=0	(*48)VOLUME(SP1) *1
0x4006_4004 Bit24	RSTVOLPS1=0	(*51)VOLUME(PS1) *1

0x4006_4004 Bit25	RSTMTRPS1=0	(*50)METER(PS1) *1
0x4006_4004 Bit26	RSTPCM1PS=0	(*49)PCMPS1 *1
0x4006_4004 Bit27	RSTATM1=0	(*52)AudioTimer1 *1
0x4006_4004 Bit28	RSTPCKGEN=0	PCM Clock Generator *1
0x4006_4004 Bit29	RSTASRC=0	(*53)ASRC *1
0x4004_1000 Bit[1:0]	PLL1STYB=0 PLL1RESETB=0	(*16)System PLL phase-comparison circuit and VCO standby only. Frequency dividing setting register is not reset.
0x4004_2000 Bit[1:0]	PLL2STYB=0 PLL2RESETB=0	(*18)Audio PLL phase-comparison circuit and VCO standby only. Frequency dividing setting register is not reset.

*1: This is a Function block reset excluding a setting register. Only hardware reset can initialize the setting registers.

6-3-2 Clock Function

This section explains about the clock function of this device.

Clocks generated in the OSC have a function to control the supply state for each block.

Clock supply is necessary for the blocks to be operative.

The block numbers in the figure below are used for the further explanation.

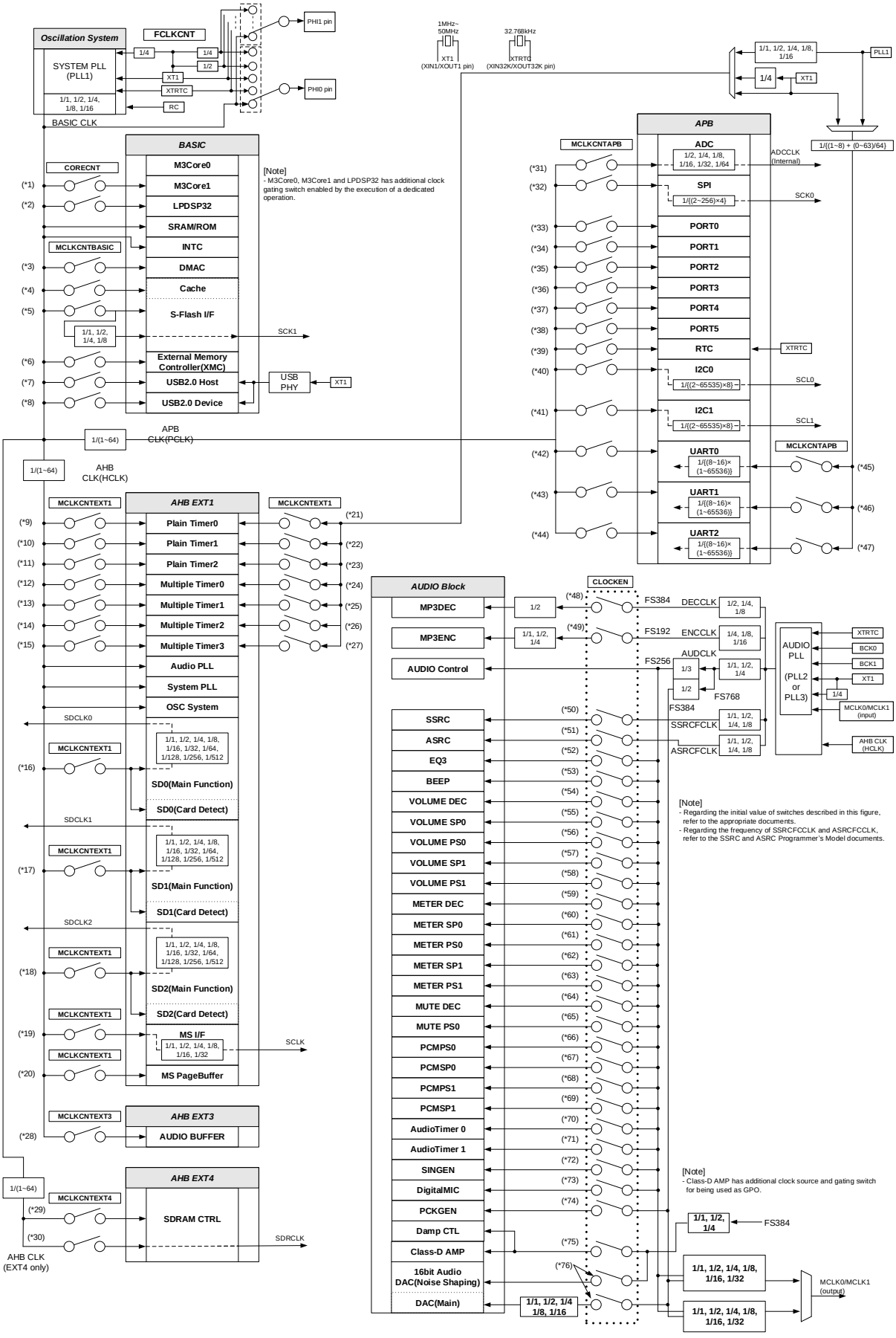


Figure 6-6 Clock system diagram

6-3-2-1 Clock in each block (Cortex-M3 Core1, LPDSP, and each Peripheral)

This device has a function to control the clock ON/OFF individually for the blocks (*1) to (*76). The below table shows the clock control types and the target block range. Setting to “1” supplies clock. Setting to “0” stops clock supply.

- Clock types and the target range

Register address	Clock supply	Target block range
0x4008_0000 Bit0	C1CLKEN=1	(*1)ARM Cortex-M3 Core1
0x4008_0000 Bit8	DSPCLKEN=1	(*2)LPDSP32
0x4008_0100 Bit0	EXTMEMCCLKEN=1	(*6)External Memory Controller
0x4008_0100 Bit1	SFIFCLKEN=1	(*5)S-Flash I/F
0x4008_0100 Bit2	USBHOSTCLKEN=1	(*7)USB2.0 Host
0x4008_0100 Bit4	DMACCLKEN=1	(*3)DMAC
0x4008_0100 Bit10	CACHECLKEN=1	(*4)Cache
0x4008_0100 Bit11	USBDEVCLKEN=1	(*8)USB2.0 Device
0x4008_0104 Bit0	MTM0CLKEN=1	(*12)Multiple Timer ch0
0x4008_0104 Bit1	MTM1CLKEN=1	(*13)Multiple Timer ch1
0x4008_0104 Bit2	MTM2CLKEN=1	(*14)Multiple Timer ch2
0x4008_0104 Bit3	MTM3CLKEN=1	(*15)Multiple Timer ch3
0x4008_0104 Bit4	PTM0CLKEN=1	(*9)Plain Timer ch0
0x4008_0104 Bit5	PTM1CLKEN=1	(*10)Plain Timer ch1
0x4008_0104 Bit6	PTM2CLKEN=1	(*11)Plain Timer ch2
0x4008_0104 Bit8	SDIF0CLKEN=1	(*16)SD I/F ch0
0x4008_0104 Bit9	SDIF1CLKEN=1	(*17)SD I/F ch1
0x4008_0104 Bit10	SDIF2CLKEN=1	(*18)SD I/F ch2
0x4008_0104 Bit11	MSHOSTCLKEN=1	(*19)MS Host
0x4008_0104 Bit12	MSPBCLKEN=1	(*20)MS PB
0x4008_0104 Bit24	MTM0CCLKEN=1	(*24)Multiple Timer ch0 for timer count
0x4008_0104 Bit25	MTM1CCLKEN=1	(*25)Multiple Timer ch1 for timer count
0x4008_0104 Bit26	MTM2CCLKEN=1	(*26)Multiple Timer ch2 for timer count
0x4008_0104 Bit27	MTM3CCLKEN=1	(*27)Multiple Timer ch3 for timer count
0x4008_0104 Bit28	PTM0CCLKEN=1	(*21)Plain Timer ch0 for timer count
0x4008_0104 Bit29	PTM1CCLKEN=1	(*22)Plain Timer ch1 for timer count
0x4008_0104 Bit30	PTM2CCLKEN=1	(*23)Plain Timer ch2 for timer count
0x4008_0108 Bit0	AudioBufCLKEN=1	(*28)Audio Buffer
0x4008_010C Bit0	SDRAMCCLKEN0=1	(*29)SDRAM CTRL
0x4008_010C Bit1	SDRAMCCLKEN1=1	(*30)SDRAM CTRL for I/F

0x4008_0110 Bit0	PORT0CLKEN=1	(*33)PORT0
0x4008_0110 Bit1	PORT1CLKEN=1	(*34)PORT1
0x4008_0110 Bit2	PORT2CLKEN=1	(*35)PORT2
0x4008_0110 Bit3	PORT3CLKEN=1	(*36)PORT3
0x4008_0110 Bit4	PORT4CLKEN=1	(*37)PORT4
0x4008_0110 Bit5	PORT5CLKEN=1	(*38)PORT5
0x4008_0110 Bit6	ADCCLKEN=1	(*31)10 bit A/D
0x4008_0110 Bit7	SPICLKEN=1	(*32)SPI
0x4008_0110 Bit8	I2C0CLKEN=1	(*40)I2C ch0
0x4008_0110 Bit9	I2C1CLKEN=1	(*41)I2C ch1
0x4008_0110 Bit10	UART0CLKEN=1	(*42)UART ch0
0x4008_0110 Bit11	UART1CLKEN=1	(*43)UART ch1
0x4008_0110 Bit12	UART2CLKEN=1	(*44)UART ch2
0x4008_0110 Bit13	RTCCLKEN=1	(*39)RTC APB I/F circuit-only clock
0x4008_0110 Bit14	GPIOCLKEN=1	Clock when DAMP pin is used as GPO.
0x4008_0110 Bit24	UART0IFCLKEN=1	(*46)UART ch0 for communication control
0x4008_0110 Bit25	UART1IFCLKEN=1	(*46)UART ch1 for communication control
0x4008_0110 Bit26	UART2IFCLKEN=1	(*47)UART ch2 for communication control
0x4006_4000 Bit1	FCEMP3ENC=1	(*49)MP3 Encoder *1
0x4006_4000 Bit2	FCEMP3DEC=1	(*50)MP3 Decoder *1
0x4006_4000 Bit3	FCEMUTEDEC=1	(*64)MUTE(DEC) *1
0x4006_4000 Bit4	FCESSRC=1	(*50)SSRC *1
0x4006_4000 Bit5	FCEMTRDEC=1	(*59)METER(DEC) *1
0x4006_4000 Bit6	FCEVOLDEC=1	(*54)VOLUME(DEC) *1
0x4006_4000 Bit7	FCEDGMIC=1	(*73)Digital MIC *1
0x4006_4000 Bit8	FCEPCMSP0=1	(*67)PCMSP0 *1
0x4006_4000 Bit9	FCEMTRSP0=1	(*60)METER(SP0) *1
0x4006_4000 Bit10	FCESINGEN=1	(*72)SINGEN *1
0x4006_4000 Bit11	FCEVOLSP0=1	(*55)VOLUME(SP0) *1
0x4006_4000 Bit12	FCEEQ3=1	(*52)EQ3 *1
0x4006_4000 Bit13	FCEVOLPS0=1	(*56)VOLUME(PS0) *1
0x4006_4000 Bit14	FCEMUTEPS0=1	(*65)MUTE(PS0) *1
0x4006_4000 Bit15	FCEMTRPS0=1	(*61)METER(PS0) *1
0x4006_4000 Bit16	FCEBEEP=1	(*53)BEEP *1
0x4006_4000 Bit17	FCEPCMPS0=1	(*66)PCMPS0 *1
0x4006_4000 Bit18	FCEDAC=1	(*76)DAC *1
0x4006_4000 Bit19	FCEDAMPCTL=1	(*75)Class-D AMP Control *1

0x4006_4000 Bit20	FCEATM0=1	(*70)AudioTimer0 *1
0x4006_4000 Bit21	FCEPCMSP1=1	(*69)PCMSP1 *1
0x4006_4000 Bit22	FCEMTRSP1=1	(*62)METER(SP1) *1
0x4006_4000 Bit23	FCEVOLSP1=1	(*57)VOLUME(SP1) *1
0x4006_4000 Bit24	FCEVOLPS1=1	(*58)VOLUME(PS1) *1
0x4006_4000 Bit25	FCEMTRPS1=1	(*63)METER(PS1) *1
0x4006_4000 Bit26	FCEPCMPS1=1	(*68)PCMPS1 *1
0x4006_4000 Bit27	FCEATM1=1	(*71)AudioTimer1 *1
0x4006_4000 Bit28	FCEPCKGEN=1	(*74)PCM Clock Generator *1
0x4006_4000 Bit29	FCEASRC=1	(*51)ASRC *1

6-3-3 LPDSP Program download and Start sequence

The following outline shows the sequence from program download to its activation for LPDSP.

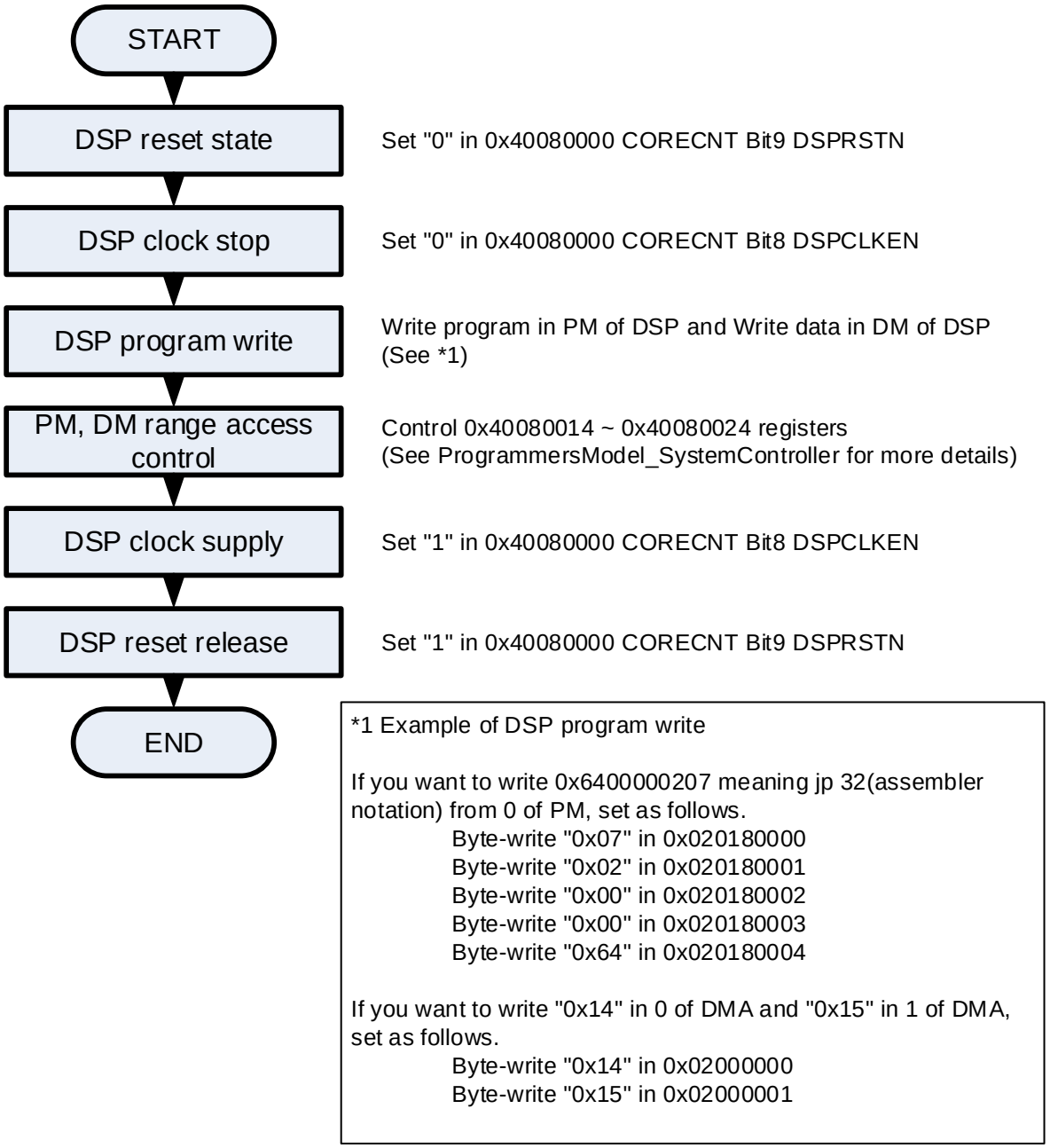


Figure 6-7 LPDSP program download and startup sequence

7 CORE_REG

7-1 Introduction

This chapter describes the outline of CORE_REG in the section below.

- The Outline of CORE_REG

7-1-1 The Outline of CORE_REG

This document describes a group of registers that each core has in the multicore configuration.

7-2 Functional Overview

This chapter describes the outline of CORE_REG function blocks in the section below.

- CORE_REG Circuit Configuration

7-2-1 CORE_REG Circuit Configuration

CORE_REG circuit configuration is as shown in Figure 7-1

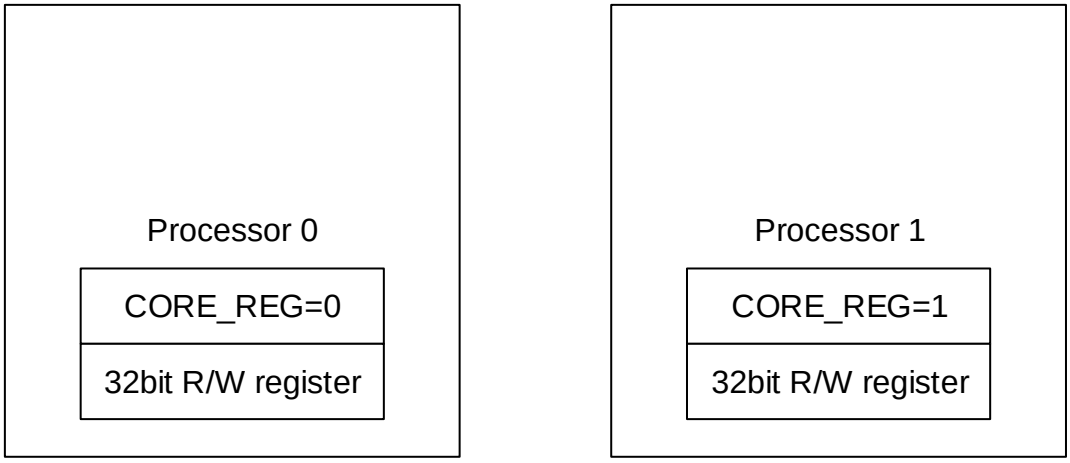


Figure 7-1

7-3 Programmer's Model

This chapter contains CORE_REG register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the Programmer's model
- Summary of CORE_REG registers
- Register Descriptions

7-3-1 About the Programmer's model

CORE_REG base address is not fixed and varies by the system implementation.

Please see the *System specifications* for more information about base address.

Offset is fixed for all registers.

7-3-2 Summary of CORE_REG registers

CORE_REG is controlled by the registers in Table 7-1 below.

Remarks: Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF" only. Do not access any other address.

Table 7-1 CORE_REG registers

Offset	Abbrev.	Attribute	Contents	Initial value
0x00	CORE_ID	R	CPU number register	0x0000_0000
0x04	CORE_RWREG	R/W	Read/Write register	0x0000_0000

7-3-3 Register Descriptions

7-3-3-1 CPU number register:CORE_ID

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	ID
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit0: ID
Indicates a number of CPU in the multicore configuration
0: Processor is CPU0.
1: Processor is CPU1.

7-3-3-2 Read/Write register: CORE_RWREG

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	RWREG31	RWREG30	RWREG29	RWREG28	RWREG27	RWREG26	RWREG25	RWREG24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	RWREG23	RWREG22	RWREG21	RWREG20	RWREG19	RWREG18	RWREG17	RWREG16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	RWREG15	RWREG14	RWREG13	RWREG12	RWREG11	RWREG10	RWREG9	RWREG8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	RWREG7	RWREG6	RWREG5	RWREG4	RWREG3	RWREG2	RWREG1	RWREG0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: RWREG31-0
This is a 32-bit read/write register

7-4 Operation

The remainder of this page is intentionally left blank.

8 MUTEX_REG

8-1 Introduction

This chapter provides the brief summary of MUTEX_REG in the sections below:

- Brief Summary of MUTEX_REG

8-1-1 Brief Summary of MUTEX_REG

This is the manual of the registers prepared to mediate the access to the common resource in the multitask system.

8-2 Functional Overview

This chapter provides overview of functional blocks used for the MUTEX_REG configuration in the sections below:

- MUTEX_REG Circuit Configuration

8-2-1 MUTEX_REG Circuit Configuration

Below shows MUTEX_REG circuit configuration.
This figure shows the appearance when the access to the common resource has been permitted to task 1 among two or more tasks.

This figure shows the state that the access to the common resource has been permitted to task1 among two or more tasks. At this time, the access is prohibited to tasks except task1.

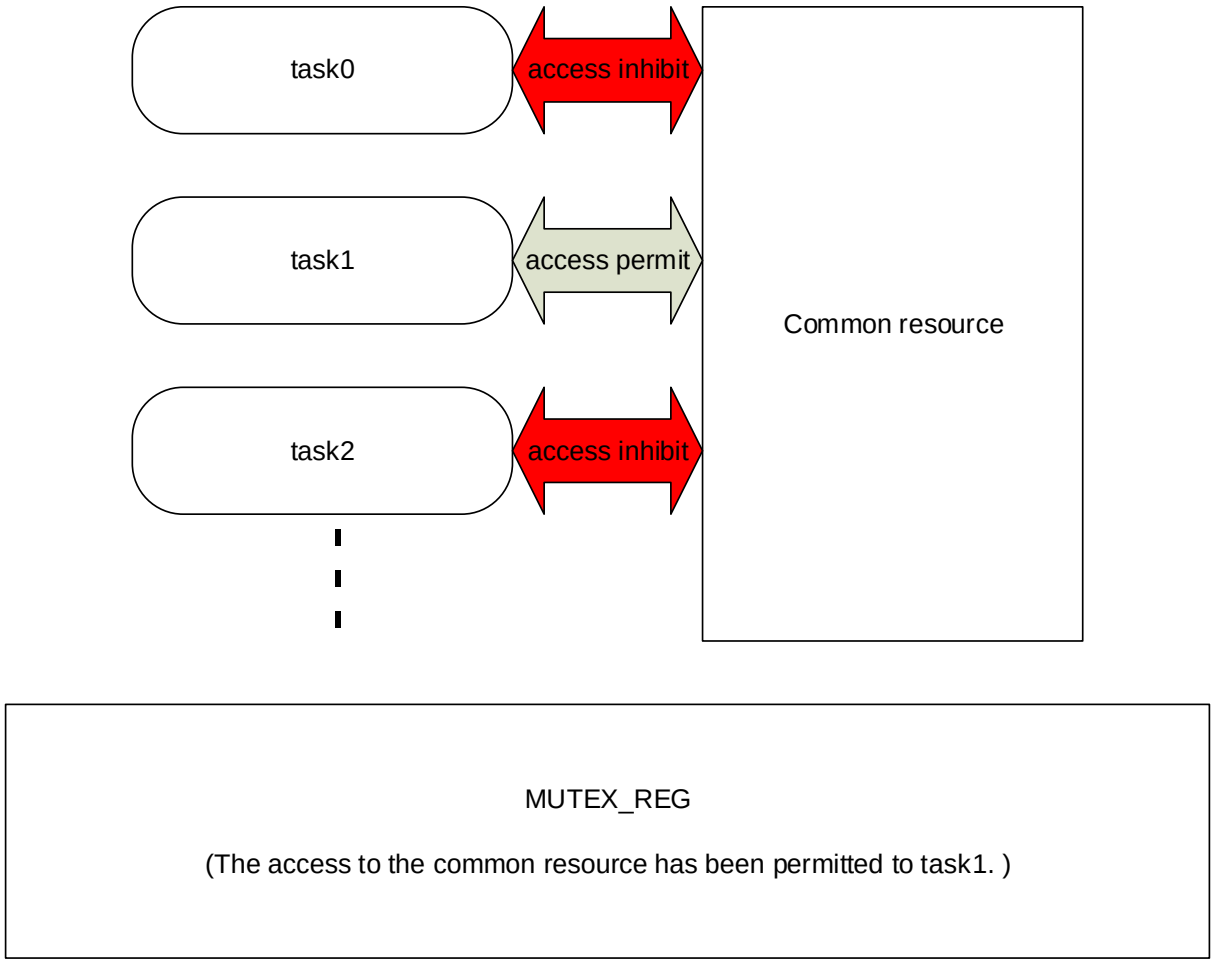


Figure 8-1

8-3 Programmer's Model

This chapter contains MUTEX_REG register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of MUTEX_REG registers
- Register Descriptions

8-3-1 About the programmer's model

MUTEX_REG base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

8-3-2 Summary of MUTEX_REG registers

MUTEX_REG is controlled by the registers shown in Table 8-1 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 8-1 MUTEX_REG registers

OFFSET	Abbreviation	Attribute	Register Names	Default value
0x00	MUTEX0	R/W	Common resource0 mediation register	0x0000_0000
0x04	MUTEX1	R/W	Common resource1 mediation register	0x0000_0000
0x08	MUTEX2	R/W	Common resource2 mediation register	0x0000_0000
0x0C	MUTEX3	R/W	Common resource3 mediation register	0x0000_0000
0x10	MUTEX4	R/W	Common resource4 mediation register	0x0000_0000
0x14	MUTEX5	R/W	Common resource5 mediation register	0x0000_0000
0x18	MUTEX6	R/W	Common resource6 mediation register	0x0000_0000
0x1C	MUTEX7	R/W	Common resource7 mediation register	0x0000_0000
0x20	MUTEX8	R/W	Common resource8 mediation register	0x0000_0000
0x24	MUTEX9	R/W	Common resource9 mediation register	0x0000_0000
0x28	MUTEX10	R/W	Common resource10 mediation register	0x0000_0000
0x2C	MUTEX11	R/W	Common resource11 mediation register	0x0000_0000
0x30	MUTEX12	R/W	Common resource12 mediation register	0x0000_0000
0x34	MUTEX13	R/W	Common resource13 mediation register	0x0000_0000
0x38	MUTEX14	R/W	Common resource14 mediation register	0x0000_0000
0x3C	MUTEX15	R/W	Common resource15 mediation register	0x0000_0000

8-3-3 Register Descriptions

8-3-3-1 Common resource xx mediation register:MUTEXxx

[Base + 0x0000~0x003C]

Bit	31	30	29	28	27	26	25	24
Name	CORE15	CORE14	CORE13	CORE12	CORE11	CORE10	CORE9	CORE8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	CORE7	CORE6	CORE5	CORE4	CORE3	CORE2	CORE1	CORE0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	VALUE15	VALUE14	VALUE13	VALUE12	VALUE11	VALUE10	VALUE9	VALUE8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	VALUE7	VALUE6	VALUE5	VALUE4	VALUE3	VALUE2	VALUE1	VALUE0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-16 : CORE15-0
Bit15-0 : VALUE15-0

When the VALUE field is "0x0000", MUTEXxx can be used (In a word, it is unlocked). MUTEXxx cannot be used besides (In a word, it is locked).
MUTEXxx can always be reading. The processor (Or, any master) can confirm current state by reading MUTEXxx.
MUTEXxx can only be written in a specific condition. Writing can operate only when one side or both of the following conditions are approved and MUTEXxx is changed.
Condition 1: The VALUE field of MUTEXxx is "0x0000".
Condition 2: The CORE field of MUTEXxx matches the CORE field of the write data.
The processor tries the acquisition of MUTEXxx by writing own ID in the CORE field and non-zero in VALUE. Next, the processor confirms the CORE field, and checks whether to have succeeded in acquisition.

Note) xx indicates 0-15.
Note) 1-byte write access and 2-byte write access are invalid.

8-4 Operation

This section provides information about MUTEX_REG operation.

8-4-1 Common resource access lock

(*1) Writes own ID in the CORE field and non-zero in VALUE field.

(*2) Check whether own ID is written in the CORE field. If it matches the own ID, the access to the common resource is possible. If it doesn't match the own ID, returns to (*1) because it is an access inhibit.

8-4-2 Common resource access unlock

(*1) Writes the value stored in CORE field in the CORE field and "0" in VALUE field.

9 DSP_CMDIF

9-1 Introduction

This chapter provides the outline of DSP Command IF in the section below.

- The Outline of DSP Command IF

9-1-1 The Outline of DSP Command IF

DSP Command IF is a module used for command notification from HOST0 (HOST1) to DSP and command response from DSP to HOST0 (HOST1).

Command notification can generate an interrupt to DSP.

Command response can generate an interrupt to HOST0 (HOST1).

9-2 Functional Overview

This chapter provides overview of functional blocks used for the DSP Command IF configuration in the section below.

- DSP Command IF Circuit Configuration

9-2-1 DSP Command IF Circuit Configuration

The circuit configuration of DSP Command IF is as shown below.

DSP, HOST0 and HOST1 will access each register via System bus.

Access to each register will enable Data transfer from HOST0 (HOST1) to DSP, Interrupt control for DSP, Data transfer from DSP to HOST0 (HOST1), and Interrupt control for HOST0 (HOST1).

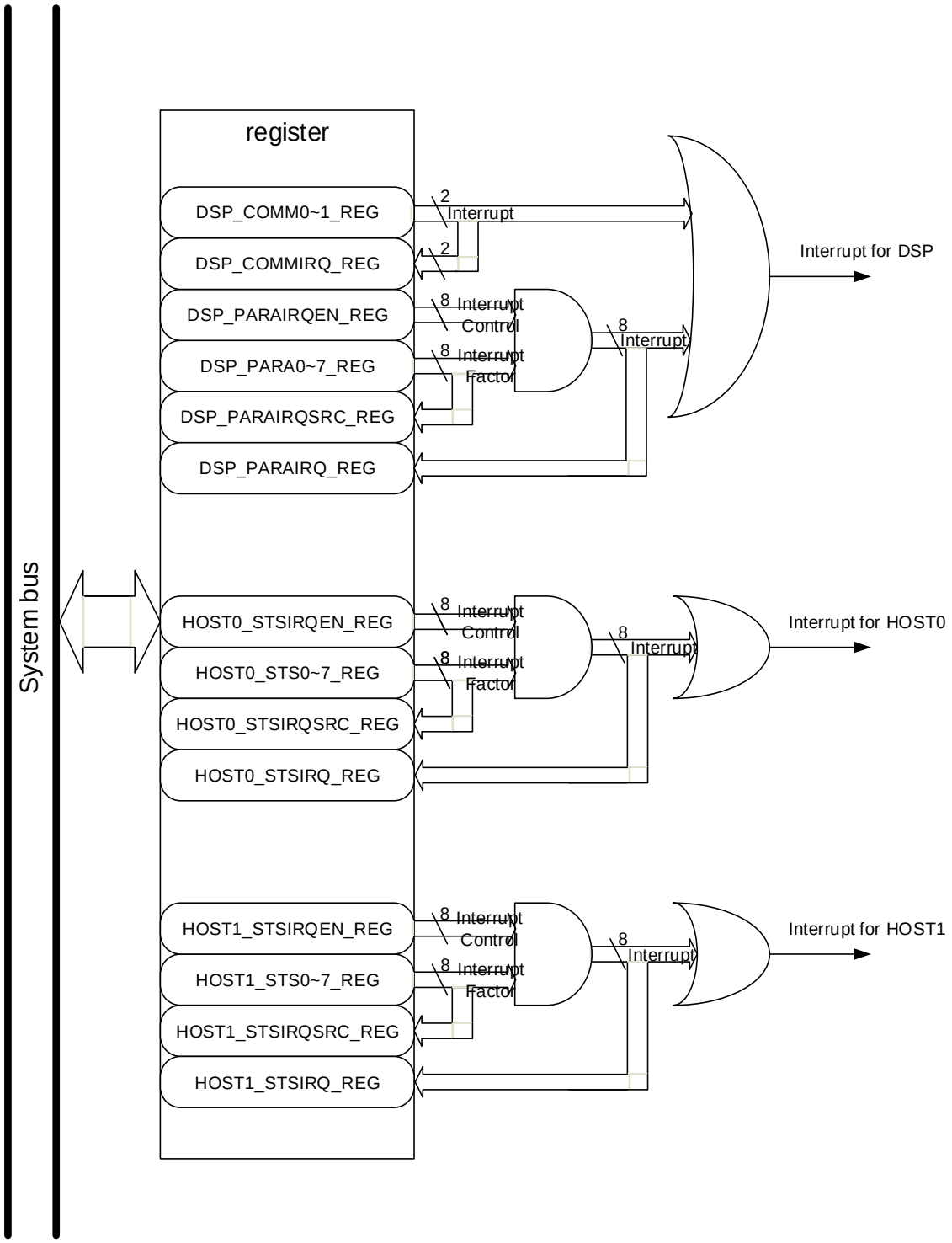


Figure 9-1

9-3 Programmer's Model

This chapter provides DSP Command IF register information and other information necessary for the register setup using microcontrollers in the sections below.

- About the Programmer's Model
- Summary of DSP Command IF Registers
- Register Descriptions

9-3-1 About the Programmer's Model

DSP Command IF base address is not fixed and varies by the system implementation.

Please refer to the *System specifications* for more information about the base address.

Offset is fixed for all registers.

9-3-2 Summary of DSP Command IF Registers

DSP Command IF is controlled by the registers in Table 9-1.

Remark: Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 9-1 DSP Command IF registers

Offset	Abbrev.	Attribute	Register name	Initial value
0x00	DSP_COMM0_REG	R/W	DSP Command 0 setting register	0x0000_0000
0x04	DSP_COMM1_REG	R/W	DSP Command 1 setting register	0x0000_0000
0x08	DSP_PARA0_REG	R/W	DSP Parameter 0 setting register	0x0000_0000
0x0C	DSP_PARA1_REG	R/W	DSP Parameter 1 setting register	0x0000_0000
0x10	DSP_PARA2_REG	R/W	DSP Parameter 2 setting register	0x0000_0000
0x14	DSP_PARA3_REG	R/W	DSP Parameter 3 setting register	0x0000_0000
0x18	DSP_PARA4_REG	R/W	DSP Parameter 4 setting register	0x0000_0000
0x1C	DSP_PARA5_REG	R/W	DSP Parameter 5 setting register	0x0000_0000
0x20	DSP_PARA6_REG	R/W	DSP Parameter 6 setting register	0x0000_0000
0x24	DSP_PARA7_REG	R/W	DSP Parameter 7 setting register	0x0000_0000
0x28	HOST0_STS0_REG	R/W	HOST0 Status 0 setting register	0x0000_0000
0x2C	HOST0_ STS 1_REG	R/W	HOST0 Status 1 setting register	0x0000_0000
0x30	HOST0_ STS 2_REG	R/W	HOST0 Status 2 setting register	0x0000_0000
0x34	HOST0_ STS 3_REG	R/W	HOST0 Status 3 setting register	0x0000_0000
0x38	HOST0_STS4_REG	R/W	HOST0 Status 4 setting register	0x0000_0000
0x3C	HOST0_STS5_REG	R/W	HOST0 Status 5 setting register	0x0000_0000
0x40	HOST0 _STS6_REG	R/W	HOST0 Status 6 setting register	0x0000_0000
0x44	HOST0 _STS7_REG	R/W	HOST0 Status 7 setting register	0x0000_0000
0x48	HOST1_STS0_REG	R/W	HOST1 Status 0 setting register	0x0000_0000
0x4C	HOST1_STS1_REG	R/W	HOST1 Status 1 setting register	0x0000_0000
0x50	HOST1_STS2_REG	R/W	HOST1 Status 2 setting register	0x0000_0000
0x54	HOST1_STS3_REG	R/W	HOST1 Status 3 setting register	0x0000_0000
0x58	HOST1_STS4_REG	R/W	HOST1 Status 4 setting register	0x0000_0000
0x5C	HOST1_STS5_REG	R/W	HOST1 Status 5 setting register	0x0000_0000
0x60	HOST1 _STS6_REG	R/W	HOST1 Status 6 setting register	0x0000_0000
0x64	HOST1 _STS7_REG	R/W	HOST1 Status 7 setting register	0x0000_0000
0x68	DSP_PARAIRQEN_REG	R/W	DSP Parameter interrupt control register	0x0000_0000
0x6C	HOST0_STSIRQEN_REG	R/W	HOST0 Status interrupt control register	0x0000_0000
0x70	HOST1_STSIRQEN_REG	R/W	HOST1 Status interrupt control register	0x0000_0000
0x74	DSP_PARAIRQSRC_REG	R	DSP Parameter interrupt factor confirmation register	0x0000_0000
0x78	HOST0_STSIRQSRC_REG	R	HOST0 Status interrupt factor confirmation register	0x0000_0000
0x7C	HOST1_STSIRQSRC_REG	R	HOST1 Status interrupt factor confirmation register	0x0000_0000
0x80	DSP_COMMIRQ_REG	R	DSP Command interrupt confirmation register	0x0000_0000
0x84	DSP_PARAIRQ_REG	R	DSP Parameter interrupt confirmation register	0x0000_0000
0x88	HOST0_STSIRQ_REG	R	HOST0 Status interrupt confirmation register	0x0000_0000
0x8C	HOST1_STSIRQ_REG	R	HOST1 Status interrupt confirmation register	0x0000_0000

9-3-3 Register Descriptions

9-3-3-1 DSP Command n Setting Register: DSP_COMMn_REG

[Base + 0x00 - 0x04]

Bit	31	30	29	28	27	26	25	24
Name	DSP_COMMn_31	DSP_COMMn_30	DSP_COMMn_29	DSP_COMMn_28	DSP_COMMn_27	DSP_COMMn_26	DSP_COMMn_25	DSP_COMMn_24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSP_COMMn_23	DSP_COMMn_22	DSP_COMMn_21	DSP_COMMn_20	DSP_COMMn_19	DSP_COMMn_18	DSP_COMMn_17	DSP_COMMn_16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSP_COMMn_15	DSP_COMMn_14	DSP_COMMn_13	DSP_COMMn_12	DSP_COMMn_11	DSP_COMMn_10	DSP_COMMn_9	DSP_COMMn_8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSP_COMMn_7	DSP_COMMn_6	DSP_COMMn_5	DSP_COMMn_4	DSP_COMMn_3	DSP_COMMn_2	DSP_COMMn_1	DSP_COMMn_0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: DSP_COMMn_31-0
Sets a command to be sent from HOST0 (HOST1) to DSP.

Accessing this register will set/clear the interrupt for DSP.
Write-access to DSP_COMMn will generate an interrupt by DSP_COMMn.
Read-access will clear an interrupt by DSP_COMMn.

DSP_COMMn interrupt state can be confirmed by the DSP Command Interrupt Confirmation Register: DSP_COMM.

Note: “n” indicates 0 ~ 1

9-3-3-2 DSP Parameter n Setting Register : DSP_PARAn_REG

[Base + 0x08 - 0x24]

Bit	31	30	29	28	27	26	25	24
Name	DSP_PARAn_31	DSP_PARAn_30	DSP_PARAn_29	DSP_PARAn_28	DSP_PARAn_27	DSP_PARAn_26	DSP_PARAn_25	DSP_PARAn_24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	DSP_PARAn_23	DSP_PARAn_22	DSP_PARAn_21	DSP_PARAn_20	DSP_PARAn_19	DSP_PARAn_18	DSP_PARAn_17	DSP_PARAn_16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DSP_PARAn_15	DSP_PARAn_14	DSP_PARAn_13	DSP_PARAn_12	DSP_PARAn_11	DSP_PARAn_10	DSP_PARAn_9	DSP_PARAn_8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DSP_PARAn_7	DSP_PARAn_6	DSP_PARAn_5	DSP_PARAn_4	DSP_PARAn_3	DSP_PARAn_2	DSP_PARAn_1	DSP_PARAn_0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: DSP_PARAn_31-0
Sets data to be sent from HOST0 (HOST1) to DSP.

Accessing this register will set/clear DSP interrupt factor.
Write-access to DSP_PARAn will set “1” for DSP_PARAn interrupt factor.
Read-acces will clear DSP_PARAn interrupt factor to “0”.

DSP_PARAn interrupt factor state can be confirmed with DSP Parameter Interrupt Factor Confirmation Register:
DSP_PARAIRQSRC_REG.

Note: “n” indicates 0 ~ 7

9-3-3-3 HOST0 Status n Setting Register : HOST0_STSn_REG

[Base + 0x28 - 0x44]

Bit	31	30	29	28	27	26	25	24
Name	HOST0_STSn_31	HOST0_STSn_30	HOST0_STSn_29	HOST0_STSn_28	HOST0_STSn_27	HOST0_STSn_26	HOST0_STSn_25	HOST0_STSn_24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	HOST0_STSn_23	HOST0_STSn_22	HOST0_STSn_21	HOST0_STSn_20	HOST0_STSn_19	HOST0_STSn_18	HOST0_STSn_17	HOST0_STSn_16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	HOST0_STSn_15	HOST0_STSn_14	HOST0_STSn_13	HOST0_STSn_12	HOST0_STSn_11	HOST0_STSn_10	HOST0_STSn_9	HOST0_STSn_8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	HOST0_STSn_7	HOST0_STSn_6	HOST0_STSn_5	HOST0_STSn_4	HOST0_STSn_3	HOST0_STSn_2	HOST0_STSn_1	HOST0_STSn_0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: HOST0_STSn_31-0
Sets data to be sent from DSP to HOST0.

Accessing this register will set/clear the HOST0 interrupt factor.
Write-access to HOST0_STSn will set “1” to the interrupt factor by HOST0_STSn.
Read-access will clear the interrupt factor by HOST0_STSn to “0”.

HOST0_STSn interrupt factor state can be confirmed with HOST0 Status Interrupt Factor Confirmation Register:
HOST0_STSIRQSRC_REG.

Note: “n” indicates 0 ~ 7

9-3-3-4 HOST1 Status n Setting Register: HOST1_STSn_REG

[Base + 0x48 - 0x64]

Bit	31	30	29	28	27	26	25	24
Name	HOST1_STSn_31	HOST1_STSn_30	HOST1_STSn_29	HOST1_STSn_28	HOST1_STSn_27	HOST1_STSn_26	HOST1_STSn_25	HOST1_STSn_24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	HOST1_STSn_23	HOST1_STSn_22	HOST1_STSn_21	HOST1_STSn_20	HOST1_STSn_19	HOST1_STSn_18	HOST1_STSn_17	HOST1_STSn_16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	HOST1_STSn_15	HOST1_STSn_14	HOST1_STSn_13	HOST1_STSn_12	HOST1_STSn_11	HOST1_STSn_10	HOST1_STSn_9	HOST1_STSn_8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	HOST1_STSn_7	HOST1_STSn_6	HOST1_STSn_5	HOST1_STSn_4	HOST1_STSn_3	HOST1_STSn_2	HOST1_STSn_1	HOST1_STSn_0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: HOST1_STSn_31-0
Sets data to be sent from DSP to HOST1.

Accessing this register will set/clear the HOST1 interrupt factor.
Write-access to HOST1_STSn will set “1” to the interrupt factor by HOST1_STSn.
Read-access will clear the interrupt factor by HOST1_STSn to “0”.

HOST1_STSn interrupt factor state can be confirmed with HOST1 Status Interrupt Factor Confirmation Register:
HOST1_STSIRQSRC_REG.

Note: “n” indicates 0 ~ 7

9-3-3-5 DSP Parameter Interrupt Control Register: DSP_PARAIRQEN_REG

[Base + 0x68]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSP_PAR AIRQEN7	DSP_PAR AIRQEN6	DSP_PAR AIRQEN5	DSP_PAR AIRQEN4	DSP_PAR AIRQEN3	DSP_PAR AIRQEN2	DSP_PAR AIRQEN1	DSP_PAR AIRQEN0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bitn: DSP_PARAIRQENn

Controls parameter interrupt to DSP.

0: No interrupt occurrence to DSP by DSP parameter interrupt factor DSP_PARAIRQSRCn

1: Interrupt will occur when DSP parameter interrupt factor DSP_PARAIRQSRCn is set to “1”

Note: “n” indicates 0 ~ 7

9-3-3-6 HOST0 Status Interrupt Control Register: HOST0_STSIRQEN_REG

[Base + 0x6C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	HOST0_STSIRQEN7	HOST0_STSIRQEN6	HOST0_STSIRQEN5	HOST0_STSIRQEN4	HOST0_STSIRQEN3	HOST0_STSIRQEN2	HOST0_STSIRQEN1	HOST0_STSIRQEN0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bitn: HOST0_STSIRQENn

Controls a status interrupt for HOST0.

0: No interrupt occurrence to HOST0 by its status interrupt factor HOST0_STSIRQSRCn

1: Interrupt will occur to HOST0 when HOST0 status interrupt factorHOST0_STSIRQSRCn is set to “1”

Note: “n” indicates 0 ~ 7

9-3-3-7 HOST1 Status Interrupt Control Register: HOST1_STSIRQEN_REG

[Base + 0x70]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	HOST1_STSIRQEN7	HOST1_STSIRQEN6	HOST1_STSIRQEN5	HOST1_STSIRQEN4	HOST1_STSIRQEN3	HOST1_STSIRQEN2	HOST1_STSIRQEN1	HOST1_STSIRQEN0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bitn: HOST1_STSIRQENn

Controls a status interrupt for HOST1.

0: No interrupt occurrence to HOST1 by its status interrupt factor HOST1_STSIRQSRCn

1: Interrupt will occur to HOST1 when HOST1 status interrupt factorHOST1_STSIRQSRCn is set to “1”

Note: “n” indicates 0 ~ 7

9-3-3-8 DSP Parameter Interrupt Factor Confirmation Register: DSP_PARAIRQSRC_REG

[Base + 0x74]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSP_PARAIRQRC7	DSP_PARAIRQRC6	DSP_PARAIRQRC5	DSP_PARAIRQRC4	DSP_PARAIRQRC3	DSP_PARAIRQRC2	DSP_PARAIRQRC1	DSP_PARAIRQRC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: DSP_PARAIRQSRCn
Able to confirm DSP parameter interrupt factors.
Write-access to DSP_PARAn will set “1” for DSP_PARAIRQRCn.
Read-access to DSP_PARAn will set “0” for DSP_PARAIRQRCn.

Note: “n” indicates 0 ~ 7

9-3-3-9 HOST0 Status Interrupt Factor Confirmation Register: HOST0_STSIRQSRC_REG

[Base + 0x78]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	HOST0_STSIRQSRC7	HOST0_STSIRQSRC6	HOST0_STSIRQSRC5	HOST0_STSIRQSRC4	HOST0_STSIRQSRC3	HOST0_STSIRQSRC2	HOST0_STSIRQSRC1	HOST0_STSIRQSRC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: HOST0_STSIRQSRCn
Able to confirm HOST0 status interrupt factors.
Write-access to HOST0_STSn will set “1” for HOST0_STSIRQSRCn.
Read-access to HOST0_STSn will set “0“ for HOST0_STSIRQSRCn.

Note: “n” indicates 0 ~ 7

9-3-3-10 HOST1 Status Interrupt Factor Confirmation Register: HOST1_STSIRQSRC_REG

[Base + 0x7C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	HOST1_STSIRQSRC7	HOST1_STSIRQSRC6	HOST1_STSIRQSRC5	HOST1_STSIRQSRC4	HOST1_STSIRQSRC3	HOST1_STSIRQSRC2	HOST1_STSIRQSRC1	HOST1_STSIRQSRC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: HOST1_STSIRQSRCn
Able to confirm HOST1 status interrupt factors.
Write-access to HOST1_STSn will set “1” for HOST1_STSIRQSRCn.
Read-access to HOST1_STSn will set “0“ for HOST1_STSIRQSRCn.

Note: “n” indicates 0 ~ 7

9-3-3-11 DSP Command Interrupt Confirmation Register: DSP_COMMIRQ_REG

[Base + 0x80]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	DSP_COMMIRQ1	DSP_COMMIRQ0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: DSP_COMMIRQn
Able to confirm interrupt generated by write-access to DSP_COMMn.
0: No interrupt generated
1: Interrupt is generated

Note: “n” indicates 0 ~ 7

9-3-3-12 DSP Parameter Interrupt Confirmation Register: DSP_PARAIRQ_REG

[Base + 0x84]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSP_PAR AIRQ 7	DSP_PAR AIRQ 6	DSP_PAR AIRQ 5	DSP_PAR AIRQ 4	DSP_PAR AIRQ 3	DSP_PAR AIRQ 2	DSP_PAR AIRQ 1	DSP_PAR AIRQ 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: DSP_PARAIRQn

Able to confirm an interrupt by DSP parameter interrupt factor DSP_PARAIRQSRCn.

0: No interrupt generated

1: Interrupt is generated

When both DSP_PARAIRQSRC0 and DSP_PARAIRQEN0 are set to “1”, DSP_PARAIRQ0 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC1 and DSP_PARAIRQEN1 are set to “1”, DSP_PARAIRQ1 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC2 and DSP_PARAIRQEN2 are set to “1”, DSP_PARAIRQ2 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC3 and DSP_PARAIRQEN3 are set to “1”, DSP_PARAIRQ3 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC4 and DSP_PARAIRQEN3 are set to “1”, DSP_PARAIRQ4 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC5 and DSP_PARAIRQEN3 are set to “1”, DSP_PARAIRQ5 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC6 and DSP_PARAIRQEN3 are set to “1”, DSP_PARAIRQ6 will be”1” and an interrupt for DSP will be generated.
When both DSP_PARAIRQSRC7 and DSP_PARAIRQEN3 are set to “1”, DSP_PARAIRQ7 will be”1” and an interrupt for DSP will be generated.

Note: “n” indicates 0 ~ 7

9-3-3-13 HOST0 Status Interrupt Confirmation Register: HOST_STSIRQ_REG

[Base + 0x88]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	HOST0_STSIRQ 7	HOST0_STSIRQ 6	HOST0_STSIRQ 5	HOST0_STSIRQ 4	HOST0_STSIRQ 3	HOST0_STSIRQ 2	HOST0_STSIRQ 1	HOST0_STSIRQ 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: HOST0_STSIRQn

Able to confirm an interrupt by HOST0 status interrupt factor HOST0_STSIRQSRCn.
0: No interrupt generated
1: Interrupt is generated

When both HOST0_STSIRQSRC0 and HOST0_STSIRQEN0 are “1,” HOST0_STSIRQ0 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC1 and HOST0_STSIRQEN1 are “1,” HOST0_STSIRQ1 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC2 and HOST0_STSIRQEN2 are “1,” HOST0_STSIRQ2 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC3 and HOST0_STSIRQEN3 are “1,” HOST0_STSIRQ3 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC4 and HOST0_STSIRQEN4 are “1,” HOST0_STSIRQ4 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC5 and HOST0_STSIRQEN5 are “1,” HOST0_STSIRQ5 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC6 and HOST0_STSIRQEN6 are “1,” HOST0_STSIRQ6 becomes “1” and HOST0 is interrupted.
When both HOST0_STSIRQSRC7 and HOST0_STSIRQEN7 are “1,” HOST0_STSIRQ7 becomes “1” and HOST0 is interrupted.

Note: “n” indicates 0 ~ 7

9-3-3-14 HOST1 Status Interrupt Confirmation Register: HOST_STSIRQ_REG

[Base + 0x8C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	HOST1_STSIRQ 7	HOST1_STSIRQ 6	HOST1_STSIRQ 5	HOST1_STSIRQ 4	HOST1_STSIRQ 3	HOST1_STSIRQ 2	HOST1_STSIRQ 1	HOST1_STSIRQ 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bitn: HOST1_STSIRQn

Able to confirm an interrupt by HOST1 status interrupt factor HOST1_STSIRQSRCn.
0: No interrupt generated
1: Interrupt is generated

When both HOST1_STSIRQSRC0 and HOST1_STSIRQEN0 are “1,” HOST1_STSIRQ0 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC1 and HOST1_STSIRQEN1 are “1,” HOST1_STSIRQ1 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC2 and HOST1_STSIRQEN2 are “1,” HOST1_STSIRQ2 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC3 and HOST1_STSIRQEN3 are “1,” HOST1_STSIRQ3 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC4 and HOST1_STSIRQEN4 are “1,” HOST1_STSIRQ4 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC5 and HOST1_STSIRQEN5 are “1,” HOST1_STSIRQ5 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC6 and HOST1_STSIRQEN6 are “1,” HOST1_STSIRQ6 becomes “1” and HOST1 is interrupted.
When both HOST1_STSIRQSRC7 and HOST1_STSIRQEN7 are “1,” HOST1_STSIRQ7 becomes “1” and HOST1 is interrupted.

Note: “n” indicates 0 ~ 7

9-4 Operation

This section describes some examples of DSP Command IF setting flow.

9-4-1 Ex: Command notification from HOST0 to DSP and the response

(*1) HOST0 sets a parameter to be sent to DSP in DSP_PARA0–3_REG.

(*2) HOST0 sets a command to be sent to DSP in DSP_COMM0_REG.

Write-accessing DSP_COMM0_REG generates an interrupt to DSP.

(*3) DSP is interrupted and checks that DSP_COMMIRQ0 is “1.”

(*4) DSP reads DSP_PARA0–3_REG and DSP_COMM0_REG to start responding.

(*5) DSP sets a parameter in HOST0_STS0–7_REG to be sent to HOST0.

(*6) DSP controls HOST0_STSIRQEN_REG to notify HOST0 of an interrupt by HOST0_STS0–7_REG.

(*7) HOST0 is interrupted and reads HOST0_STS0–7_REG to check DSP response content.

9-4-2 Ex: Command notification from HOST1 to DSP and the response

(*1) HOST1 sets a parameter to be sent to DSP in DSP_PARA4–7_REG.

(*2) HOST1 sets a command to be sent to DSP in DSP_COMM1_REG.

Write-accessing DSP_COMM1_REG generates an interrupt to DSP.

(*3) DSP is interrupted and checks that DSP_COMMIRQ1 is “1.”

(*4) DSP reads DSP_PARA4–7_REG and DSP_COMM1_REG to start responding.

(*5) DSP sets a parameter in HOST1_STS0–7_REG to be sent to HOST1.

(*6) DSP controls HOST1_STSIRQEN_REG to notify HOST1 of an interrupt by HOST1_STS0–7_REG.

(*7) HOST1 is interrupted and reads HOST1_STS0–7_REG to check DSP response content.

10 PORT

10-1 Introduction

This chapter provides the outline of PORTs in the section below.

- The Outline of PORTs

10-1-1 The Outline of PORTs

PORTn is 16-bit input/output port which controls 16 pins.

Each of the PORTn-controlled 16 pins is equipped with CMOS type output driver that outputs data from a pin. Each pin is directly connected to a Schmitt-type input buffer. Through this input buffer, electric potential at a pin is always transferred to each function block. Each pin has a built-in pull-up/pull-down resistor that can be enabled or disabled by the program.
See the specification, ProgrammersModel_SystemController for pull-up/pull-down resistor setting and port function switching setting.

PORTn control can enable or disable an output driver. Enabling an output driver allows the pin to output data. Disabling an output driver allows the pin to be used only for input.

Table 10-1 Signal used with PORTn

Signal	Input/Output	Function
PORTn Clock	Input	Clock for PORTn
SYSTEM Bus	Bus	For PORTn control

10-2 Functional Overview

This chapter describes the overview of functional blocks used for the PORTs configuration in the section below.

- PORTs Circuit Configuration

10-2-1 PORTs Circuit Configuration

PORTs circuit configuration is as shown in Figure 10-1

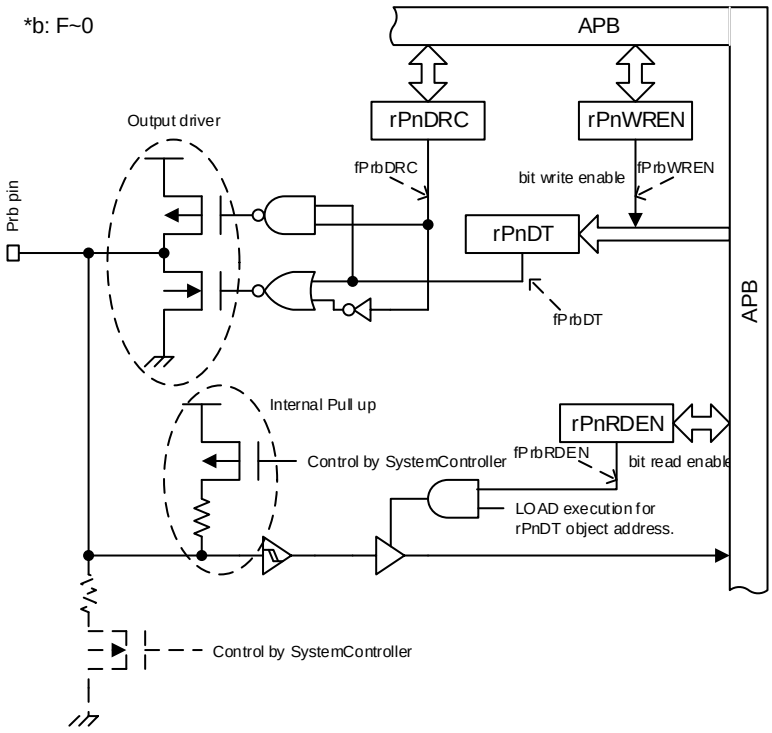


Figure 10-1

10-3 Programmer’s Model

This chapter describes PORTs registers and setting via microcomputer in the sections below.

- About the Programmer’s model
- Summary of PORTs registers
- Register Descriptions

10-3-1 About the Programmer’s model

PORTs base address is not fixed and varies by the system implementation.

Please refer to the *System specification* for more information about the base address.

Each register's offset is fixed.

10-3-2 Summary of PORTs registers

PORTs are controlled by the registers listed in Table 10-2.

Remarks: Accessible addresses are the ones listed in between the range of “Base+0x0000” to “Base+0xFFFF”. Do not access any other address.

Table 10-2 PORTs registers

Offset	Abbrev.	Attribute	Register name	Initial value
0x00	rPnDRC	R/W	PORTn input/output control register	0x0000_0000
0x04	rPnDT	R/W	PORTn data register	0x0000_XXXX
0x08	rPnFRH	R/W	PORTn data-hold register	0x0000_0000
0x10	rPnWREN	R/W	PORTn data write enable register	0x0000_FFFF
0x14	rPnRDEN	R/W	PORTn data read enable register	0x0000_FFFF

10-3-3 Register Descriptions

10-3-3-1 PORTn Input/Output Control Register: rPnDRC

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fPnFDRC	fPnEDRC	fPnDDRC	fPnCDRC	fPnBDRC	fPnADRC	fPn9DRC	fPn8DRC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fPn7DRC	fPn6DRC	fPn5DRC	fPn4DRC	fPn3DRC	fPn2DRC	fPn1DRC	fPn0DRC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0: fPnFDRC-fPn0DRC
Port “n” input/output control
0: Set Port “n” to input side.
1: Set Port “n” to output side.

Attention: Ports that are not in use shall have the following settings;

When controlling input → Enable pull-up/pull-down resistor and open the input

When controlling output → open the output.

10-3-3-2 PORTn Data Register: rPnDT

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fPnFDT	fPnEDT	fPnDDT	fPnCDT	fPnBDT	fPnADT	fPn9DT	fPn8DT
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fPn7DT	fPn6DT	fPn5DT	fPn4DT	fPn3DT	fPn2DT	fPn1DT	fPn0DT
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0: fPnFDT-fPn0DT

This register manages write/read data in port “n”. This register holds the data to be output to port “n” when writing data and can read a value of external pins of port “n” when reading data.

When the register is reset, it will be in the input state, so the read-out value will vary by the port status.

10-3-3-3 PORTn Data-hold Register: rPnFRH

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fPnFFRH	fPnEFRH	fPnDFRH	fPnCFRH	fPnBFRH	fPnAFRH	fPn9FRH	fPn8FRH
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fPn7FRH	fPn6FRH	fPn5FRH	fPn4FRH	fPn3FRH	fPn2FRH	fPn1FRH	fPn0FRH
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0: fPnFFRH-fPn0FRH
This register temporarily holds data.

Remarks: This is a buffer register that can write and read independently of rPnDRC and rPnDT.

10-3-3-4 PORTn Data-write Enable Register: rPnWREN

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fPnF WREN	fPnE WREN	fPnD WREN	fPnC WREN	fPnB WREN	fPnA WREN	fPn9 WREN	fPn8 WREN
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fPn7 WREN	fPn6 WREN	fPn5 WREN	fPn4 WREN	fPn3 WREN	fPn2 WREN	fPn1 WREN	fPn0 WREN
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0: fPnFWREN-fPn0WREN
This register is used for mask setup for rPnDT register write. Mask can be set for each bit.
0: Writing data to rPnDT is prohibited (Masked)
1: Writing data to rPnDT is permitted (Not masked)

10-3-3-5 PORTn Data-read Enable Register: rPnRDEN

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fPnF RDEN	fPnE RDEN	fPnD RDEN	fPnC RDEN	fPnB RDEN	fPnA RDEN	fPn9 RDEN	fPn8 RDEN
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fPn7 RDEN	fPn6 RDEN	fPn5 RDEN	fPn4 RDEN	fPn3 RDEN	fPn2 RDEN	fPn1 RDEN	fPn0 RDEN
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0: fPnFRDEN-fPn0RDEN
This register is used for mask setup for rPnDT register read. Mask can be set for each bit.
0: “0” can be read as rPnDT (Masked).
1: Port status can be read as rPnDT (Not masked).

10-4 Operation

This section provides information about PORTs setting flow.

10-4-1 PORTn Pin Assignment

The relation between ports and register bit positions is as shown below:

Table 10-3 Pins assigned to PORTn

Port name	Bit position
PnF	PORTn:15
PnE	PORTn:14
PnD	PORTn:13
PnC	PORTn:12
PnB	PORTn:11
PnA	PORTn:10
Pn9	PORTn:9
Pn8	PORTn:8
Pn7	PORTn:7
Pn6	PORTn:6
Pn5	PORTn:5
Pn4	PORTn:4
Pn3	PORTn:3
Pn2	PORTn:2
Pn1	PORTn:1
Pn0	PORTn:0

10-4-2 Precautions for use

“0” is always read out from bits 31 to 16. WRITE is disabled.

11 PTM

11-1 Introduction

This chapter provides the brief summary of PTMs in the sections below:

- Brief Summary of PTMs

11-1-1 Brief Summary of PTMs

PTMs is composed of Basic Timer : BTMn and Watch Dog Timer : WTMn, and each of these is an independent function.

Basic Timer : BTMn (hereinafter referred to as the BTMn) counts constant time and it is also possible to generate the basic timer interruption.

Watch Dog Timer : WTMn (hereinafter referred to as the WTMn) counts constant time and it is also possible to generate the watch dog timer interruption or the system reset.

11-1-1-1 Brief Summary of BTMn

BTMn is composed of the reload counter of 16 bit as a nucleus.
It is possible to generate the BTMn interruption by the overflow of the counter.

Table 11-1 Signal used with BTMn

Signal	I/O	Function
BTMn Clock	Input	Clock for register access of BTMn
PTMn Clock	Input	Clock for count of BTMn
PTMn Standby	Input	For standby control of PTMs including BTMn
SYSTEM Bus	Bus	For control of BTMn

11-1-1-2 Brief Summary of WTMn

WTMn is composed of the counter (WTM counter) of 16 bit as a nucleus.

It is possible to generate the watch dog timer interruption or the system reset by the overflow of the WTM counter. Setting the operation mode of WTMn by program can decide that which event of these is used.

WTMn has various functions. The functions of WTMn are as follows.

- Setting of operation mode of watch dog timer

WTMn has two kinds of modes, one is the watchdog interruption mode that generates the interruption by the overflow and the other is the watchdog reset mode that generates the system reset by the overflow. They can be set by the program.

- Setting of overflow time of watch dog timer

Time until overflowing can be set by the program.

- External output function of watchdog timer reset

The reset signal of the system reset by WTMn can be output as nRESO signal (internal signal of LSI).

- Plain timer standby function

WTMn is a part of PTMs, therefore, WTMn is controlled to be the power saving mode by the plain timer standby function.

WTMn can correspond to various applications by using these functions.

Table 11-2 Signal used with WTMn

Signal	I/O	Function
WTMn Clock	Input	Clock for register access of WTMn
PTMn Clock	Input	Clock for count of WTMn
PTMn Standby	Input	For standby control of PTMs including WTMn
nRESO	Output	Watchdog reset signal
SYSTEM Bus	Bus	For control of WTMn

11-2 Functional Overview

This chapter provides overview of functional blocks used for the PTMs configuration in the sections below:

- BTMn Circuit Configuration
- WTMn Circuit Configuration

11-2-1 BTMn Circuit Configuration

BTMn circuit configuration is shown in a Figure 11-1

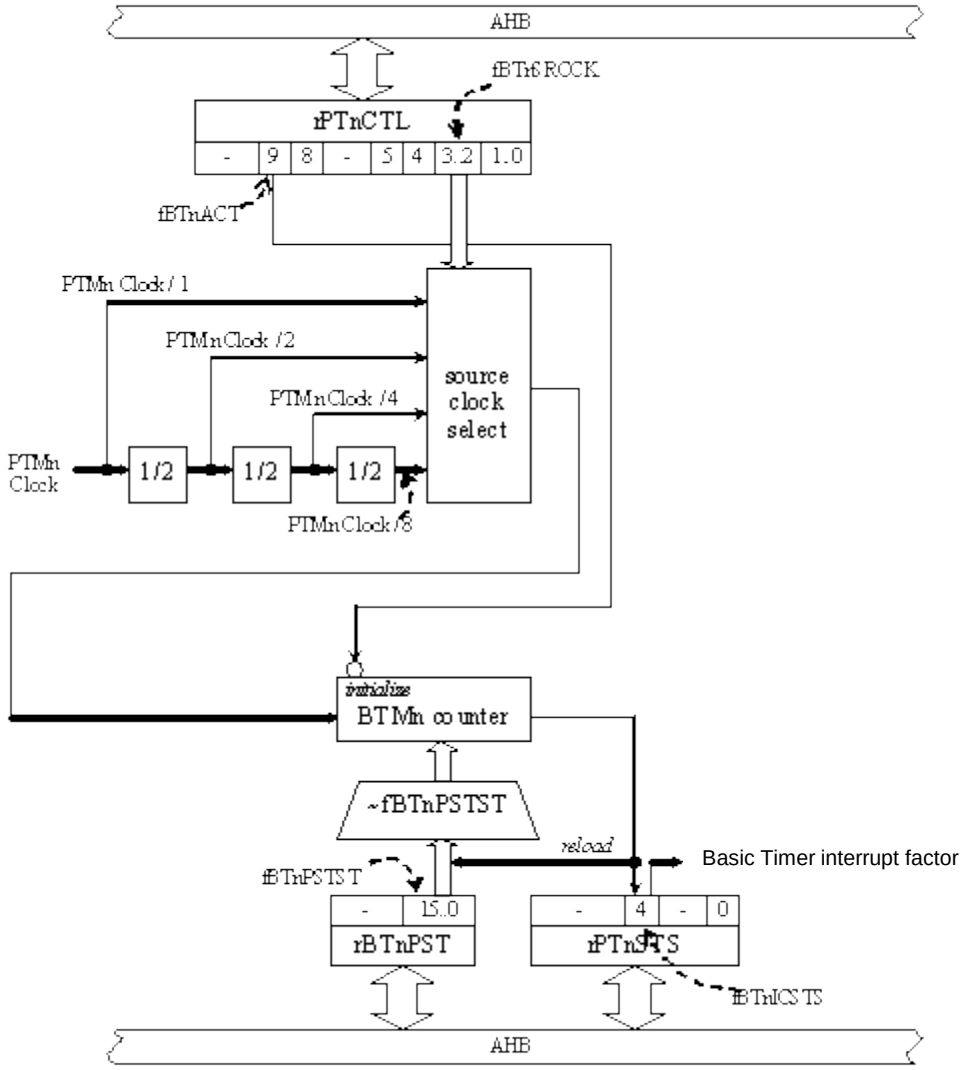


Figure 11-1 Basic Timer : BTMn Configuration

11-2-2 WTMn Circuit Configuration

WTMn circuit configuration is shown in a Figure 11-2

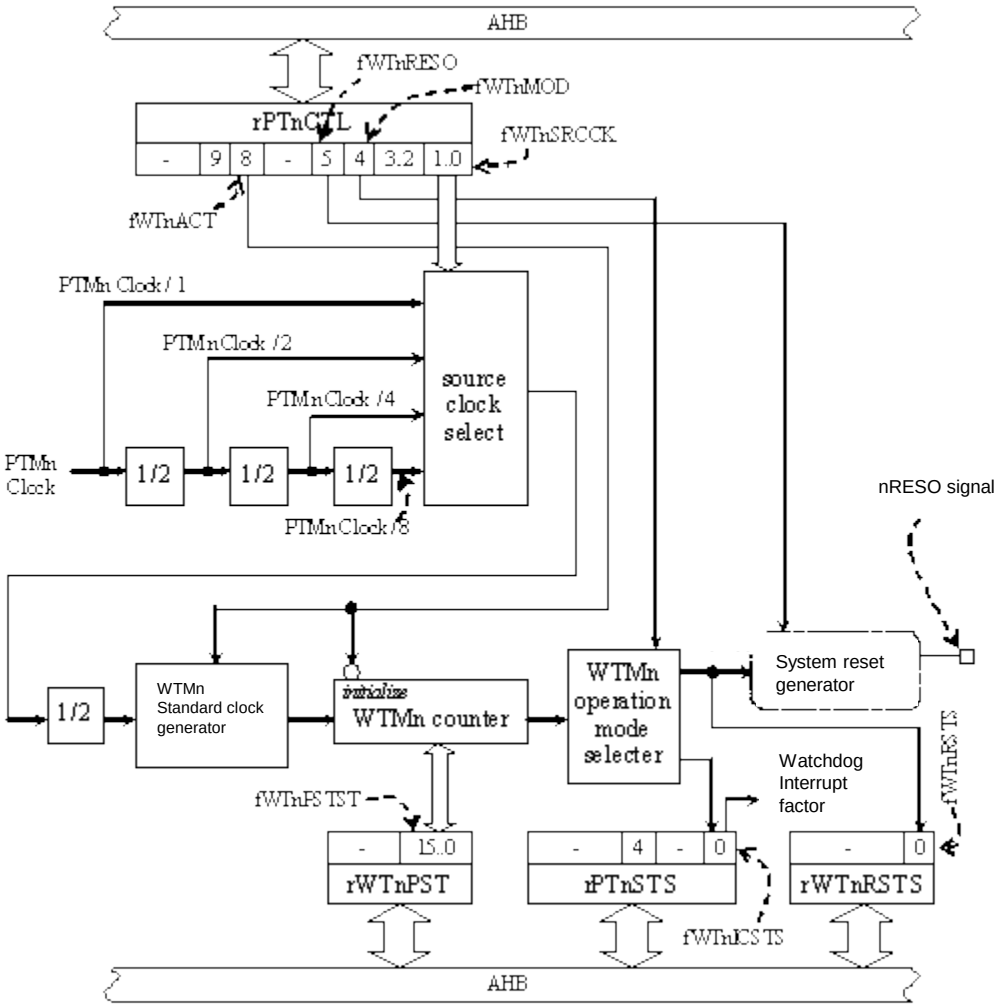


Figure 11-2 Watch Dog Timer : WTMn Configuration

11-3 Programmer's Model

This chapter contains PTMs register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of PTMs registers
- Register Descriptions

11-3-1 About the programmer's model

PTMs base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

11-3-2 Summary of PTMs registers

PTMs is controlled by the registers shown in Table 11-3 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 11-3 PTMs registers

Offset	Abbreviation	Attribute	Register Name	Initial value
0x00	rPTnCTL	R/W	PTMsn control register	0x0000_0000
0x04	rPTnSTS	R/W	PTMsn status register	0x0000_0000
0x08	rWTnBCG	R/W	WTMn standard clock generator register	0x0000_0000
0x0C	rWTnPST	R/W	WTMn preset register	0x0000_0000
0x10	rWTnRSTS	R/W	WTMn system reset status register	0x0000_0000
0x14	rBTnPST	R/W	BTMn preset register	0x0000_0000

11-3-3 Register Descriptions

11-3-3-1 PTMsn control register : rPTnCTL

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fBTnACT	fWTnACT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	fWTn RESO	fWTnMOD	fBTn SRCCK1	fBTn SRCCK0	fWTn SRCCK1	fWTn SRCCK0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit9 : fBTnACT
It controls starting or stopping of the operation of BTMn.
0:BTMn is stopped.
1:BTMn is started.

Remarks) The value of BTMn preset register is reloaded to BTMn counter when BTMn is stoped.

Bit8 : fWTnACT
It controls starting or stopping of operation of WTMn.
0:WTMn is stopped.
1:WTMn is started.

Remarks) The counter is initialized by 0 when WTMn is stopped.

Bit5 : fWTnRESO
Select whether outputting the system reset signal by WTMn to outside or not.
0: Outputting the system set signal by WTMn is prohibited.
1: Outputting the system set signal by WTMn is permitted.

Attention) When system reset is generated by the watchdog reset function, fWTnRESO is not initialized.

Bit4 : fWTnMOD
Select the use mode of WTMn.
0: Watchdog interrupt mode
1: Watchdog reset mode

Remarks) In watchdog reset mode, the system reset is generated by the overflow of WTMn.
In watchdog interrupt mode, the watchdog timer interruption factor is generated by the overflow of WTMn.

Bit3-2 : fBTnSRCCK1-0
Select the clock source of BTMn.
00 b: PTMn Clock / 1 as the source clock of BTMn is selected.
01 b: PTMn Clock / 2 as the source clock of BTMn is selected.
10 b: PTMn Clock / 4 as the source clock of BTMn is selected.
11 b: PTMn Clock / 8 as the source clock of BTMn is selected.

Bit1-0 : fWTnSRCCK1-0
Select the clock source of WTMn.
00 b: PTMn Clock / 1 as the source clock of WTMn is selected.
01 b: PTMn Clock / 2 as the source clock of WTMn is selected.
10 b: PTMn Clock / 4 as the source clock of WTMn is selected.
11 b: PTMn Clock / 8 as the source clock of WTMn is selected.

Remarks) The overflow time of WTMn is decided by fWTnSRCCK, setting of WTMn preset register and WTMn standard clock generator register, and WTMn Clock.

11-3-3-2 PTMsn status register : rPTnSTS

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	fBTnOVF	-	-	-	fWTnICSTS
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R/W

Bit4 : fBTnOVF

It shows whether BTMn overflowed or not.

At Read

0: BTMn doesn't overflow.

1: BTMn overflowed.

At Write

0: fBTnOVF is not influenced.

1: fBTnOVF is cleared.

Remarks) This fBTnOVF is transmitted to the interruption control part of the interruption function as basic timer interruption factor.

Bit0 : fWTnICSTS

It shows whether the watchdog timer interruption condition consisted or not.
At Read
0: The watchdog timer interruption condition is failure.
1: The watchdog timer interruption condition consisted.

At Write
0: FWTnICSTS is not influenced.
1: FWTnICSTS is cleared.

Attention) The overflow of WTMn generated in the watchdog interruption mode is assumed as the completion of the watchdog timer interruption condition. Therefore, even if the overflow is generated at the watchdog reset mode, this flag is not set.

Remarks) This fWTnICSTS is transmitted to the interruption control part of the interruption function as watchdog timer interruption factor.

11-3-3-3 WTMn standard clock generator register : rWTnBCG

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fWTnBCGST7	fWTnBCGST6	fWTnBCGST5	fWTnBCGST4	fWTnBCGST3	fWTnBCGST2	fWTnBCGST1	fWTnBCGST0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0 : fWTnBCGST7-0

The cycle of the reference clock of WTMn is set.

The overflow time of WTMn is decided by fWTnBCGST, setting of WTMn preset register and fWTnSRCCK of PTMsn control register, and PTMn Clock.

11-3-3-4 WTMn preset register : rWTnPST

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fWTnPST ST15	fWTnPST ST14	fWTnPST ST13	fWTnPST ST12	fWTnPST ST11	fWTnPST ST10	fWTnPST ST9	fWTnPST ST8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fWTnPST ST7	fWTnPST ST6	fWTnPST ST5	fWTnPST ST4	fWTnPST ST3	fWTnPST ST2	fWTnPST ST1	fWTnPST ST0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fWTnPSTST15-0

It is access data to the WTMn counter.

At Read

The count value of the WTMn counter is read at the read point.

At Write

The value of fWTnPSTST is written into the WTMn counter. However, the value of fWTnPSTST is not reflected to the WTMn counter even if rWTnPST register is written when WTMn has stopped.

Remarks) The overflow time of WTMn is decided by fWTnPSTST, setting of fWTnSRCCK of PTMsn control register and WTMn standard clock generator register, and PTMn Clock.

The WTMn counter does the count up operation.

When a big value is written in this register, time by reaching the overflow shortens.

Because interrupt might be immediately generated if 0xFFFF is written, this operation is prohibited.

11-3-3-5 WTMn system reset status register : rWTnRSTS

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	fWTn RSTS
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0 : fWTnRSTS

It shows whether the system reset condition by WTMn consisted or not.

At Read

0: The system reset condition by WTMn is unapproval.

1: The system reset condition by WTMn consisted.

At Write

0: FWTnRSTS is not influenced.

1: FWTnRSTS is cleared.

Attention) The overflow of WTMn generated in the watchdog reset mode is assumed as the completion of the system reset condition. Therefore, this flag is not set even if the overflow is generated at the watchdog interruption mode, and system reset by the overflow is not generated either.

Remarks) When the system is reset by WTMn, the reset release is done as the value of this fWTnRSTS is 1. When system reset is generated by factors other than WTMn, fWTnRSTS is initialized to 0. Therefore, the confirmation whether system reset by WTMn was generated becomes possible by the monitor of this fWTnRSTS.

11-3-3-6 BTMn preset register : rBTnPST

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fBTnPST ST15	fBTnPST ST14	fBTnPST ST13	fBTnPST ST12	fBTnPST ST11	fBTnPST ST10	fBTnPST ST9	fBTnPST ST8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fBTnPST ST7	fBTnPST ST6	fBTnPST ST5	fBTnPST ST4	fBTnPST ST3	fBTnPST ST2	fBTnPST ST1	fBTnPST ST0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fBTnPSTST15-0

It is the reload value to the BTMn counter.

When the overflow of the BTMn counter is generated and BTMn stops, the value of this fBTnPSTST is reload to the BTMn counter.

The BTMn counter does the count up operation. On the other hand, This fBTnPSTST is reload to the BTMn counter after fBTnPSTST is converted to one's complement (~fBTnPSTST) of the value of fBTnPSTST. Therefore, the value set to fBTnPSTST comes to be treated as count number of BTMn source clocks.

Remarks) The overflow time of BTMn is decided by fBTnPSTST, setting of fBTnSRCCK of PTMsn control register, and PTMn Clock.

11-4 Operation

This section provides information about PTMn operation.

11-4-1 Function of basic timer

Basic timer: BTMn has the basic timer source clock selection, the basic timer interruption function, and the standby function as functions of it.

11-4-1-1 Basic timer source clock selection

In BTMn, the source clock supplied to 16 bits counter for the time count can be selected from among four kinds. The clock source is selected by fBTnSRCCK of the PTMsn control register.

11-4-1-2 Basic timer interruption function

It is an interruption function by the overflow of BTMn.

The overflow time of BTMn is decided by setting of fBTnSRCCK of PTMsn control register and fBTnPSTST of BTMn preset register.

Overflow time of BTMn [sec] =

$$(1 / \text{PTMn Clock[Hz]}) \times 2^{\text{fBTnSRCCK}} \times (\text{fBTnPSTST} + 1)$$

fBTnOVF of the PTMsn status register is transmitted to the interruption control part as basic timer interruption factor.

When BTMn overflows, fBTnOVF becomes 1, and this is transmitted to the interruption control part as BTMn interruption factor.

Even if the interruption which treats fBTnOVF as an interruption factor is generated, this fBTnOVF is never automatically cleared. Please clear fBTnOVF by the program.

11-4-1-3 Basic timer standby function

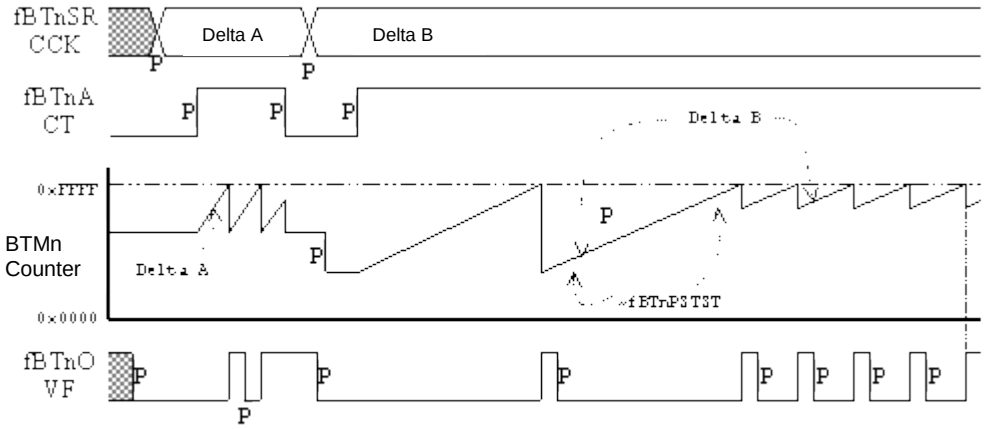
BTMn corresponds to software or hardware standby by the power save function.

If the demand of software or hardware standby to BTMn from the power save function was generated, BTMn rushes into the state of the standby at once.

Moreover, if the demand of the release of the standby state was generated by the power save function, the standby is released. Please initialize the setting of BTMn by the program after releasing the standby.

11-4-2 Operation of basic timer

The operation of BTMn is shown in the following.



- P : It is the timing controlled by the program.
- BTMn Counter : It is the value of BTMn Counter.

Figure 11-3 Operation of basic timer

11-4-3 Function of watchdog timer

Watch dog timer: WTMn has the watchdog timer operation mode selection, the setting of the watchdog overflow time, the watchdog timer reset output function, and the watchdog timer standby function as functions of it.

11-4-3-1 Watchdog timer operation mode selection

Watch dog timer: WTMn has two operational modes of the watchdog interruption mode and the watchdog reset mode.

- Watchdog interruption mode

It is a mode which generates the interruption by the overflow of WTMn.

The watchdog interruption mode is set by fWTnMOD of the PTMsn control register.

When WTMn overflows, fWTnICSTS of the PTMsn status register becomes 1, and this is transmitted to the interruption control part as a watchdog timer interruption factor.

Even if the interruption which treats fWTnICSTS as an interruption factor is generated, this fWTnICSTS is never automatically cleared. Please clear fWTnICSTS by the program.

- Watchdog reset mode

It is a mode which generates the system reset by the overflow of WTMn.

The watchdog reset mode is set by fWTnMOD of the PTMsn control register.

When WTMn overflows, System reset is generated at the same time as fWTnRSTS of the WTMn system reset status register is becoming 1.

FWTnRSTS is not initialized at system reset which originates in the overflow of WTMn.

11-4-3-2 Setting of the watchdog overflow time

The overflow time of WTMn is decided by setting of WTMn preset register, WTMn standard clock generator register and fWTnSRCCK of PTMsn control register.

Overflow time of WTMn [sec] =

$$(1 / \text{PTMn Clock[Hz]}) \times 2^{\text{fWTnSRCCK}} \times 2 \times (256 - \text{fWTnBCGST}) \times (65536 - \text{fWTnPSTST})$$

11-4-3-3 Watchdog timer reset output function

When the system reset is generated in the watchdog reset mode, this reset signal can be output as nRESO signal.

This function can be used by setting fWTnRESO of the PTMsn control register to 1.

11-4-3-4 Watchdog timer standby function

WTMn corresponds to the software or hardware standby by the power save function.

If the demand of the software or hardware standby to WTMn from the power save function was generated, WTMn rushes into the state of the standby at once.

Moreover, if the demand of the release of the standby state was generated by the power save function, the standby is released. Please initialize the setting of WTMn by the program after releasing the standby.

11-4-4 Operation of watchdog timer

When FWTnMOD of the PTMsn control register is set, the use mode of watchdog is decided.

Please set fWTnRESO of the PTMsn control register when you use the external output function of watchdog timer reset in the watchdog reset mode.

Please set the overflow time of WTMn with the WTMn preset register, the WTMn standard clock generator register, and fWTnSRCCK of the PTMsn control register.

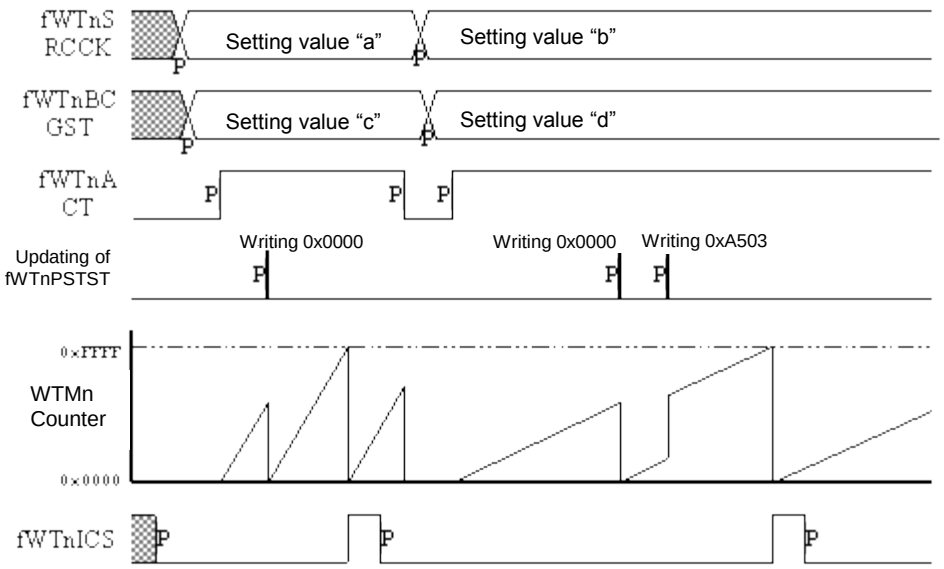
Afterwards, the operation of WTMn begins when fWTnACT of the PTMsn control register is set to 1.
When the given value is written in fWTnPSTST of the WTMn preset register before WTMn overflows while it is operating, the WTMn counter is preset to this value, and continues the count from its value.

The following operation is as follows depending on fWTnMOD of the PTMsn control register.

11-4-4-1 Operation of watchdog interrupt mode

fWTnICSTS of the PTMsn status register becomes 1 when the WTMn counter overflows without updating the WTMn preset register while WTMn is operating, and it shows that WTMn overflowed.

FWTnICSTS is transmitted to the interruption control part as WTMn interruption factor.



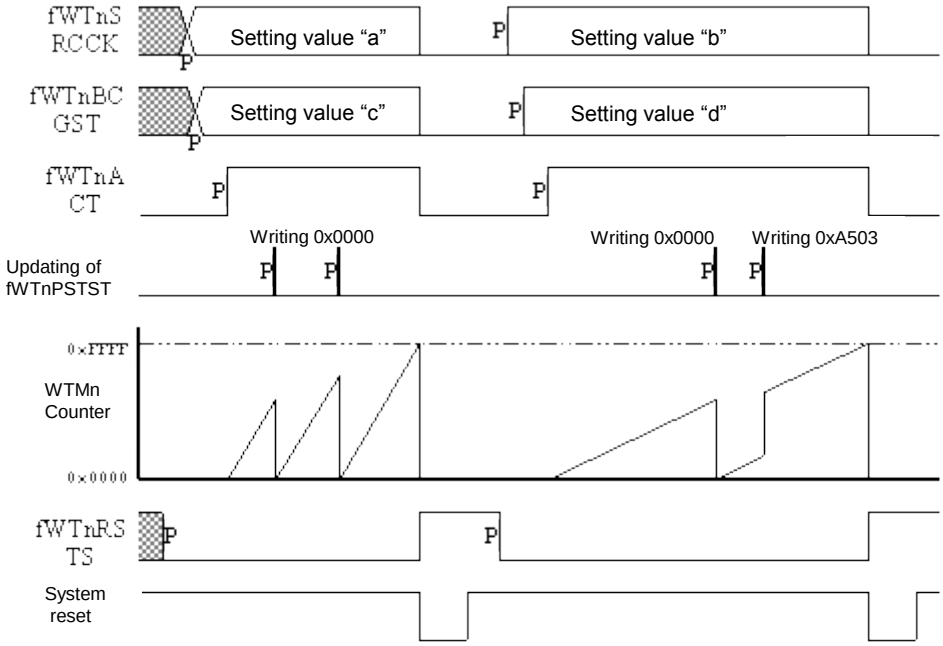
- P : It is the timing controlled by the program.
- WTMn Counter : It is the value of WTMn Counter.

Figure 11-4 Operation of WTMn : watchdog interrupt mode

11-4-4-2 Operation of watchdog reset mode

FWTnRSTS of the WTMn system reset status register becomes 1 when the WTMn counter overflows without updating the WTMn preset register while WTMn is operating, and WTMn generates system reset. Moreover, this reset signal is output as nRESO signal when the external output function of watchdog timer reset was effective.

Only fWTnRSTS of the WTMn system reset status register is maintained as 1 though the system is initialized by this system reset. Therefore, whether a reset exception based on the system reset by the overflow of WTMn was generated or not can be judged by monitoring this fWTnRSTS in the following program after accepting the reset exception.



- P : It is the timing controlled by the program.
- WTMn Counter : It is the value of WTMn Counter.
- System reset : It is low active.

Figure 11-5 Operation of WTMn : watchdog reset mode

11-4-5 Directions

The remainder of this page is intentionally left blank.

12 MTM

12-1 Introduction

This chapter provides the brief summary of MTMs in the sections below:

- Brief Summary of MTMs

12-1-1 Brief Summary of MTMs

Multiple timer: MTMsn has various functions of the measurement of the external input pulse, the output of the count situation, and the PWM output etc as well as the time count.

MTMsn is composed of the timer of two channels of MTMn-Ch0 and MTMn-Ch1 as a nuckeus of 16 bit counter, it is possible for each channel to function independently or to function cooperatively.

In MTMsn, it is possible to output the signal to the terminal or to input the signal from the terminal according to the usage of it.

Table 12-1 Signal used with MTMs

Signal	I/O	Function
TCLKnA	Input	Counter clock source to MTMsn-Ch0/Ch1
TCLKnB	Input	Counter clock source to MTMsn-Ch0/Ch1
SNTINFn0A	Output	Output signal in sentinel inform function of MTMn-Ch0
PWMn0		PWM output signal in PWM function of MTMn-Ch0
PLRn0A	Input	Input signal in pulse length reader function of MTMn-Ch0
SNTINFn0B	Output	Output signal in sentinel inform function of MTMn-Ch0
PLRn0B	Input	Input signal in pulse length reader function of MTMn-Ch0
SNTINFn1A	Output	Output signal in sentinel inform function of MTMn-Ch1
PWMn1		PWM output signal in PWM function of MTMn-Ch1
PLRn1A	Input	Input signal in pulse length reader function of MTMn-Ch1
SNTINFn1B	Output	Output signal in sentinel inform function of MTMn-Ch1
PLRn1B	Input	Input signal in pulse length reader function of MTMn-Ch1

12-1-1-1 Feature of MTMs

- Counter source clock control function

MTMn-Ch0 and MTMn-Ch1The can select a source clock of the counter respectively. The internal clock using pre-scaler and 4 kinds of external clocks are prepared as a counter clock source which can be used.

- Counter synchronous function

The synchronous operation between MTMn-Ch0 and MTMn-Ch1 is possible.

- Pulse length reader function

The measurement of the pulse width or the cycle of the external input signal is possible.
This is an input capture function.

- Sentinel inform function

Outputting the count situation of the timer outside is possible.
This is an output compare function.

- PWM output function

Outputting the PWM signal is possible.

- Multiple timer interruption

Each of MTMn-Ch0 and MTMn-Ch1 has an individual interruption factor, and this interruption factor is transmitted to the interruption control part.

- Multiple timer standby function

It corresponds to the software or hardware standby by the power save function.

12-2 Functional Overview

This chapter provides overview of functional blocks used for the MTMs configuration in the sections below:

- MTMs Circuit Configuration

12-2-1 MTMs Circuit Configuration

MTMs circuit configuration is shown in a Figure 12-1

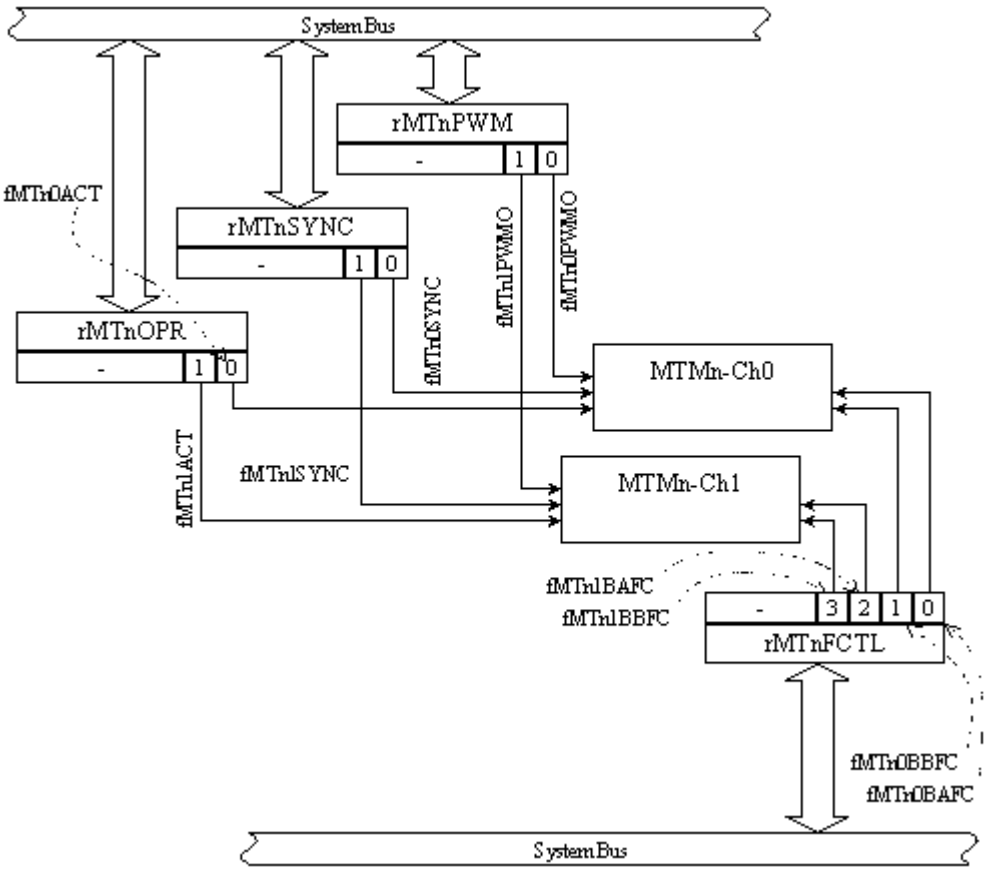


Figure 12-1 Multiple Timers : MTMs Configuration

12-3 Programmer's Model

This chapter contains MTMs register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of MTMs registers
- Register Descriptions

12-3-1 About the programmer's model

MTMs base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

12-3-2 Summary of MTMs registers

MTMs is controlled by the registers shown in Table 12-2 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 12-2 MTMs registers

Offset	Abbreviation	Attribute	Content	Initial value
0x00	rMTnOPR	R/W	MTMsn operation control register	0x0000_0000
0x04	rMTnSYNC	R/W	MTMsn synchronous setting register	0x0000_0000
0x08	rMTnPWM	R/W	MTMsnPWM output control register	0x0000_0000
0x0C	rMTnFCTL	R/W	MTMsn function control register	0x0000_0000
0x40	rMTn0CTL	R/W	MTMn-Ch0 control register	0x0000_0000
0x44	rMTn0IOCL	R/W	MTMn-Ch0 IO control register	0x0000_0000
0x4C	rMTn0STS	R/W	MTMn-Ch0 status register	0x0000_0000
0x50	rMTn0CNT	R/W	MTMn-Ch0 counter	0x0000_0000
0x54	rMTn0A	R/W	MTMn-Ch0A register	0x0000_FFFF
0x58	rMTn0B	R/W	MTMn-Ch0B register	0x0000_FFFF
0x5C	rMTn0SOL	R/W	MTMn-Ch0 output logic setting register	0x0000_0000
0x60	rMTn0BA	R/W	MTMn-Ch0 buffer A	0x0000_FFFF
0x64	rMTn0BB	R/W	MTMn-Ch0 buffer B	0x0000_FFFF
0x68	rMTn0PSCL	R/W	MTMn-Ch0 pre-scaler setting register	0x0000_00FF
0x6C	rMTn0TIER	R/W	MTMn-Ch0 interrupt control register	0x0000_0000
0x70	rMTn0TISR	R	MTMn-Ch0 interrupt status register	0x0000_0000
0x80	rMTn1CTL	R/W	MTMn-Ch1 control register	0x0000_0000
0x84	rMTn1IOCL	R/W	MTMn-Ch1 IO control register	0x0000_0000
0x8C	rMTn1STS	R/W	MTMn-Ch1 status register	0x0000_0000
0x90	rMTn1CNT	R/W	MTMn-Ch1 counter	0x0000_0000
0x94	rMTn1A	R/W	MTMn-Ch1A register	0x0000_FFFF
0x98	rMTn1B	R/W	MTMn-Ch1B register	0x0000_FFFF
0x9C	rMTn1SOL	R/W	MTMn-Ch1 output logic setting register	0x0000_0000
0xA0	rMTn1BA	R/W	MTMn-Ch1 buffer A	0x0000_FFFF
0xA4	rMTn1BB	R/W	MTMn-Ch1 buffer B	0x0000_FFFF
0xA8	rMTn1PSCL	R/W	MTMn-Ch1 pre-scaler setting register	0x0000_00FF
0xAC	rMTn1TIER	R/W	MTMn-Ch1 interrupt control register	0x0000_0000
0xB0	rMTn1TISR	R/W	MTMn-Ch1 interrupt status register	0x0000_0000

12-3-3 Register Descriptions

12-3-3-1 MTMsn operation control register : rMTnOPR

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	fMTn1 ACT	fMTn0 ACT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : fMTn1ACT(MTMn-Ch1 Activate Flag)
It controls operation of whether to start MTMn-Ch1 or not
0: MTMn-Ch1 stops.
1: MTMn-Ch1 starts.

Attention) When MTMn-Ch1 is started by fMTn1ACT, it is necessary to set the registers for general control from 0x00 to 0x0C and for MTMn-Ch1 from 0x80 to 0xB0 in advance.

Bit0 : fMTn0ACT(MTMn-Ch0 Activate Flag)
It controls operation of whether to start MTMn-Ch0 or not
0: MTMn-Ch0 stops.
1: MTMn-Ch0 starts.

Attention) When MTMn-Ch0 is started by fMTn0ACT, it is necessary to set the registers for general control from 0x00 to 0x0C and for MTMn-Ch0 from 0x40 to 0x70 in advance.

12-3-3-2 MTMsn synchronous setting register : rMTnSYNC

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	fMTn1 SYNC	fMTn0 SYNC
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : fMTn1SYNC(MTMn-Ch1 Synchronize Flag)
It controls whether MTMn-Ch1 operates as synchronous function or not.
0: MTMn-Ch1 becomes a synchronous non-object (independent timer) in the counter synchronous function.
1: MTMn-Ch1 becomes a synchronous object in the counter synchronous function.

Bit0 : fMTn0SYNC(MTMn-Ch0 Synchronize Flag)
It controls whether MTMn-Ch0 operates as synchronous function or not.
0: MTMn-Ch0 becomes a synchronous non-object (independent timer) in the counter synchronous function.
1: MTMn-Ch0 becomes a synchronous object in the counter synchronous function.

Note): You must set this register as follows because there are 2 counters only for Ch0 and Ch1 substantially.

In case of synchronizing both Ch0 and Ch1: {fMTn1SYNC, fMTn0SYNC} = 11 b.
In case of synchronizing neither Ch0 nor Ch1: { fMTn1SYNC, fMTn0SYNC} = 00 b.

The setting of {FMTn1SYNC, fMTn0SYNC} = 01 b or 10 b is prohibited.

12-3-3-3 MTMsn PWM output control register : rMTnPWM

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	fMTn1 PWMO	fMTn0 PWMO
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : fMTn1PWMO

It controls whether to use the PWM output signal in standard PWM mode of MTMn-Ch1 or not.

0: The PWM output signal in standard PWM mode of MTMn-Ch1 is invalid.

1: The PWM output signal in standard PWM mode of MTMn-Ch1 is effective.

Bit0 : fMTn0PWMO

It controls whether to use the PWM output signal in standard PWM mode of MTMn-Ch0 or not.

0: The PWM output signal in standard PWM mode of MTMn-Ch0 is invalid.

1: The PWM output signal in standard PWM mode of MTMn-Ch0 is effective.

12-3-3-4 MTMsn function control register : rMTnFCTL

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	fMTn1 BBFC	fMTn1 BAFC	fMTn0 BBFC	fMTn0 BAFC
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit3 : fMTn1BBFC(rMTn1BB Function Setting Flag)
It controls the buffer structure shared in both MTMn-Ch1 reader B block of the counter reader function and MTMn-Ch1 sentinel B block of the counter sentinel function in MTMn-Ch1.

0: The buffer structure of reader B block and sentinel B block of MTMn-Ch1 is set to single buffer.
1: The buffer structure of reader B block and sentinel B block of MTMn-Ch1 is set to double buffer.

Remarks) If fMTn1BBFC is 1, the buffer structure of MTMn-Ch1 reader B block and MTMn-Ch1 sentinel B block becomes a double buffer structure composed of MTMn-Ch1 B register and MTMn-Ch1 buffer B.
If fMTn1BBFC is 0, the buffer structure of MTMn-Ch1 reader B block and MTMn-Ch1 sentinel B block becomes a single buffer structure composed of MTMn-Ch1 B register.

Bit2 : fMTn1BAFC(rMTn1BA Function Setting Flag)
It controls the buffer structure shared in both MTMn-Ch1 reader A block of the counter reader function and MTMn-Ch1 sentinel A block of the counter sentinel function in MTMn-Ch1.

0: The buffer structure of reader A block and sentinel A block of MTMn-Ch1 is set to single buffer.
1: The buffer structure of reader A block and sentinel A block of MTMn-Ch1 is set to double buffer.

Remarks) If fMTn1BAFC is 1, the buffer structure of MTMn-Ch1 reader A block and MTMn-Ch1 sentinel A block becomes a double buffer structure composed of MTMn-Ch1 A register and MTMn-Ch1 buffer A.
If fMTn1BAFC is 0, the buffer structure of MTMn-Ch1 reader A block and MTMn-Ch1 sentinel A block becomes a single buffer structure composed of MTMn-Ch1 A register.

Bit1 : fMTn0BBFC(rMTn0BB Function Setting Flag)
It controls the buffer structure shared in both MTMn-Ch0 reader B block of the counter reader function and MTMn-Ch0 sentinel B block of the counter sentinel function in MTMn-Ch0.

0: The buffer structure of reader B block and sentinel B block of MTMn-Ch0 is set to single buffer.
1: The buffer structure of reader B block and sentinel B block of MTMn-Ch0 is set to double buffer.

Remarks) If fMTn0BBFC is 1, the buffer structure of MTMn-Ch0 reader B block and MTMn-Ch0 sentinel B block becomes a double buffer structure composed of MTMn-Ch0 B register and MTMn-Ch0 buffer B.
If fMTn0BBFC is 0, the buffer structure of MTMn-Ch0 reader B block and MTMn-Ch0 sentinel B block becomes a single buffer structure composed of MTMn-Ch0 B register.

Bit0 : fMTn0BAFC(rMTn0BA Function Setting Flag)
It controls the buffer structure shared in both MTMn-Ch0 reader A block of the counter reader function and MTMn-Ch0 sentinel A block of the counter sentinel function in MTMn-Ch0.

0: The buffer structure of reader A block and sentinel A block of MTMn-Ch0 is set to single buffer.
1: The buffer structure of reader A block and sentinel A block of MTMn-Ch0 is set to double buffer.

Remarks) If fMTn0BAFC is 1, the buffer structure of MTMn-Ch0 reader A block and MTMn-Ch0 sentinel A block becomes a double buffer structure composed of MTMn-Ch0 A register and MTMn-Ch0 buffer A.
If fMTn0BAFC is 0, the buffer structure of MTMn-Ch0 reader A block and MTMn-Ch0 sentinel A block becomes a single buffer structure composed of MTMn-Ch0 A register.

12-3-3-5 MTMn-Ch0 control register : rMTn0CTL

[Base + 0x0040]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 CLST1	fMTn0 CLST0	fMTn0 EXEG1	fMTn0 EXEG0	fMTn0 SCCK3	fMTn0 SCCK2	fMTn0 SCCK1	fMTn0 SCCK0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-6 : fMTn0CLST1-0(MTMn-Ch0 Counter Clear Condition Setting)
The condition of clearing the counter of MTMn-Ch0 is set.
00 b: The counter of MTMn-Ch0 is not cleared.
01 b: The counter of MTMn-Ch0 is cleared by the event regarding MTMn-Ch0 reader A block and MTMn-Ch0 sentinel A block.
10 b: The counter of MTMn-Ch0 is cleared by the event regarding MTMn-Ch0 reader B block and MTMn-Ch0 sentinel B block.
11 b: The counter of MTMn-Ch0 is cleared when the counter in other MTMsn which is a synchronous object is cleared.

Bit5-4 : fMTn0EXEG1-0(MTMn-Ch0 External Clock Edge Select)
When the external clock has been selected as a counter source clock of MTMn-Ch0, the clock edge of the counter is selected.
00 b: The counter of MTMn-Ch0 is counted with the rising edge of the counter source clock.
01 b: The counter of MTMn-Ch0 is counted with the falling edge of the counter source clock.
10 b/11 b: The counter of MTMn-Ch0 is counted with the both edges of the counter source clock.

Bit3-0 : fMTn0SCCK(MTMn-Ch0 Counter Source Clock Select)
The source clock of the counter of MTMn-Ch0 is selected.
0000 b: internal clock. It is internal clock supplied from the pre-scaler of MTMn-Ch0.
1000 b: external clock. It is TCLKnA signal supplied from outside.
1001 b: external clock. It is TCLKnB signal supplied from outside.

Attention): Setting other than the above is prohibited.

12-3-3-6 MTMn-Ch0 IO control register : rMTn0IOCL

[Base + 0x0044]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	fMTn0 BFNC2	fMTn0 BFNC1	fMTn0 BFNC0	-	fMTn0 AFNC2	fMTn0 AFNC1	fMTn0 AFNC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit6-4 : fMTn0BFNC2-0(MTMn-Ch0 Block-B Function Setting)
The usage of MTMn-Ch0 reader B block of the counter reader function in MTMn-Ch0 or the usage of MTMn-Ch0 sentinel B block of the counter sentinel function in MTMn-Ch0 is set.

000 b: SNTn0BC0 event is generated when the value of MTMn-Ch0B register and the value of counter match up, but SNTINFn0B signal is not generated.
001 b: SNTn0BC0 event is generated when the value of MTMn-Ch0B register and the value of counter match up, and SNTINFn0B signal is set to 0.
010 b: SNTn0BC0 event is generated when the value of MTMn-Ch0B register and the value of counter match up, and SNTINFn0B signal is set to 1.
011 b: SNTn0BC0 event is generated when the value of MTMn-Ch0B register and the value of counter match up, and SNTINFn0B signal is reversed.
100 b: RDn0B event is generated at the rising edge of PLRn0B signal, and the counter value is transmitted to MTMn-Ch0B register.
101 b: RDn0B event is generated at the falling edge of PLRn0B signal, and the counter value is transmitted to MTMn-Ch0B register.
110 b/111 b: RDn0B event is generated at the rising edge or falling edge of PLRn0B signal, and the counter value is transmitted to MTMn-Ch0B register.

Remarks) When fMTn0BFNC is set to the value from 000 b to 011 b, the use of MTMn-Ch0 sentinel B block becomes possible, and the use of MTMn-Ch0 reader B block becomes impossible. When fMTn0BFNC is set to the value from 100 b to 111 b, the use of MTMn-Ch0 reader B block becomes possible, and the use of MTMn-Ch0 sentinel B block becomes impossible.

Bit2-0 : fMTn0AFNC2-0(MTMn-Ch0 Block-A Function Setting)
The usage of MTMn-Ch0 reader A block of the counter reader function in MTMn-Ch0 or the usage of MTMn-Ch0 sentinel A block of the counter sentinel function in MTMn-Ch0 is set.

000 b: SNTn0AC0 event is generated when the value of MTMn-Ch0A register and the value of counter match up, but SNTINFn0A signal is not generated.
001 b: SNTn0AC0 event is generated when the value of MTMn-Ch0A register and the value of counter match up, and SNTINFn0A signal is set to 0.
010 b: SNTn0AC0 event is generated when the value of MTMn-Ch0A register and the value of counter match up, and SNTINFn0A signal is set to 1.
011 b: SNTn0AC0 event is generated when the value of MTMn-Ch0A register and the value of counter match up, and SNTINFn0A signal is reversed.
100 b: RDn0A event is generated at the rising edge of PLRn0A signal, and the counter value is transmitted to MTMn-Ch0A register.
101 b: RDn0A event is generated at the falling edge of PLRn0A signal, and the counter value is transmitted to MTMn-Ch0A register.
110 b/111 b: RDn0A event is generated at the rising edge or falling edge of PLRn0A signal, and the counter value is transmitted to MTMn-Ch0A register.

Remarks) When fMTn0AFNC is set to the value from 000 b to 011 b, the use of MTMn-Ch0 sentinel A block becomes possible, and the use of MTMn-Ch0 reader A block becomes impossible. When fMTn0AFNC is set to the value from 100 b to 111 b, the use of MTMn-Ch0 reader A block becomes possible, and the use of MTMn-Ch0 sentinel A block becomes impossible.

12-3-3-7 MTMn-Ch0 status register : rMTn0STS

[Base + 0x004C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	fMTn0 OVF	fMTn0 BEVT	fMTn0 AEVT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2 : fMTn0OVF(MTMn-Ch0 Overflow Flag)
It shows whether the counter of MTMn-Ch0 overflowed or not.
At Read
0: The counter of MTMn-Ch0 has not overflowed.
1: The counter of MTMn-Ch0 overflowed.

At Write
0: fMTn0OVF is not influenced.
1: fMTn0OVF is cleared.

Remarks): This fMTn0OVF is transmitted to the interrupt control part of the interrupt function as MTMn-Ch0 counter interrupt factor.

Bit1 : fMTn0BEVT(MTMn-Ch0 Block-B Event Flag)
It shows whether the RDn0B event caused by MTMn-Ch0 reader B block or the SNTn0BC0 event caused by MTMn-Ch0 sentinel B block was generated or not.
At Read
0: The event regarding MTMn-Ch0 reader B block or MTMn-Ch0 sentinel B block is not generated in MTMn-Ch0.
1: The event regarding MTMn-Ch0 reader B block or MTMn-Ch0 sentinel B block is generated in MTMn-Ch0.

At Write
0: fMTn0BEVT is not influenced.
1: fMTn0BEVT is cleared.

Remarks): This fMTn0BEVT is transmitted to the interrupt control part of the interrupt function as MTMn0B interrupt factor.

Bit0 : fMTn0AEVT(MTMn-Ch0 Block-A Event Flag)

It shows whether the RDn0A event caused by MTMn-Ch0 reader A block or the SNTn0AC0 event caused by MTMn-Ch0 sentinel A block was generated or not.

At Read

0: The event regarding MTMn-Ch0 reader A block or MTMn-Ch0 sentinel A block is not generated in MTMn-Ch0.

1: The event regarding MTMn-Ch0 reader A block or MTMn-Ch0 sentinel A block is generated in MTMn-Ch0.

At Write

0: fMTn0AEVT is not influenced.

1: fMTn0AEVT is cleared.

Remarks): This fMTn0AEVT is transmitted to the interrupt control part of the interrupt function as MTMn0A interrupt factor.

Attention): Above-mentioned Bit2-0 is Read/Clear bit. Writing directly to this bit by the program is not permitted though reading to the bit is free. Only the instruction of the bit clearness is permitted to the program.

Please store 1 to the bit position in the register corresponding to the Read/Clear bit that wants to clear when you want to clear the Read/Clear bit. As a result, the Read/Clear bit becomes 0.

On the other hand, even if 0 is stored to the bit position in the register corresponding to the Read/Clear bit, the data in the corresponding Read/Clear bit is not updated.

12-3-3-8 MTMn-Ch0 counter : rMTn0CNT

[Base + 0x0050]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn0 CNDT15	fMTn0 CNDT14	fMTn0 CNDT13	fMTn0 CNDT12	fMTn0 CNDT11	fMTn0 CNDT10	fMTn0 CNDT9	fMTn0 CNDT8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 CNDT7	fMTn0 CNDT6	fMTn0 CNDT5	fMTn0 CNDT4	fMTn0 CNDT3	fMTn0 CNDT2	fMTn0 CNDT1	fMTn0 CNDT0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn0CNDT(MTMn-Ch0 Counter Data)
It shows a count value of the MTMn-Ch0 counter. Setting a value to the MTMn-Ch0 counter is also possible.

12-3-3-9 MTMn-Ch0A register : rMTn0A

[Base + 0x0054]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn0 ADT15	fMTn0 ADT14	fMTn0 ADT13	fMTn0 ADT12	fMTn0 ADT11	fMTn0 ADT10	fMTn0 ADT9	fMTn0 ADT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 ADT7	fMTn0 ADT6	fMTn0 ADT5	fMTn0 ADT4	fMTn0 ADT3	fMTn0 ADT2	fMTn0 ADT1	fMTn0 ADT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn0ADT(MTMn-Ch0 A Register Data)
It is used in MTMn-Ch0 reader A block of the counter reader function and MTMn-Ch0 sentinel A block of the counter sentinel function in MTMn-Ch0.

12-3-3-10 MTMn-Ch0B register : rMTn0B

[Base + 0x0058]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn0 BDT15	fMTn0 BDT14	fMTn0 BDT13	fMTn0 BDT12	fMTn0 BDT11	fMTn0 BDT10	fMTn0 BDT9	fMTn0 BDT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 BDT7	fMTn0 BDT6	fMTn0 BDT5	fMTn0 BDT4	fMTn0 BDT3	fMTn0 BDT2	fMTn0 BDT1	fMTn0 BDT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn0BDT(MTMn-Ch0 B Register Data)
It is used in MTMn-Ch0 reader B block of the counter reader function and MTMn-Ch0 sentinel B block of the counter sentinel function in MTMn-Ch0.

12-3-3-11 MTMn-Ch0 output logic setting register : rMTn0SOL

[Base + 0x005C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	fMTn0 BOL	fMTn0 AOL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : fMTn0BOL(MTMn-Ch0 Sentinel-B Output Logic Setting Flag)
It shows the level of the SNTINFn0B signal in the MTMn-Ch0 sentinel inform function. It is also possible to control the SNTINFn0B signal directly by updating this fMTn0BOL with the program.
0: set the level of SNTINFn0B signal to low
1: set the level of SNTINFn0B signal to high

Remarks): This fMTn0BOL can be used for the setting of the initial state of the SNTINFn0B signal in the sentinel inform function.

Bit0 : fMTn0AOL(MTMn-Ch0 Sentinel-A Output Logic Setting Flag)
It shows the level of the SNTINFn0A signal in the MTMn-Ch0 sentinel inform function or the PWMn0 signal in standard PWM mode. It is also possible to control the SNTINFn0A signal or the PWMn0 signal directly by updating this fMTn0AOL with the program.

0: set the level of SNTINFn0A signal or PWMn0 signal to low.
1: set the level of SNTINFn0A signal or PWMn0 signal to high.

Remarks): This fMTn0AOL can be used for the setting of the initial state of the SNTINFn0A signal in the sentinel inform function or the PWMn0 signal in the standard PWM mode.

12-3-3-12 MTMn-Ch0 buffer A : rMTn0BA

[Base + 0x0060]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn0 BADT15	fMTn0 BADT14	fMTn0 BADT13	fMTn0 BADT12	fMTn0 BADT11	fMTn0 BADT10	fMTn0 BADT9	fMTn0 BADT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 BADT7	fMTn0 BADT6	fMTn0 BADT5	fMTn0 BADT4	fMTn0 BADT3	fMTn0 BADT2	fMTn0 BADT1	fMTn0 BADT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn0BADT(MTMn-Ch0 Buffer A Data)

It is used in composing the double buffer structure in MTMn-Ch0 reader A block of the counter reader function and MTMn-Ch0 sentinel A block of the counter sentinel function in MTMn-Ch0.

12-3-3-13 MTMn-Ch0 buffer B : rMTn0BB

[Base + 0x0064]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn0 BBDT15	fMTn0 BBDT14	fMTn0 BBDT13	fMTn0 BBDT12	fMTn0 BBDT11	fMTn0 BBDT10	fMTn0 BBDT9	fMTn0 BBDT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 BBDT7	fMTn0 BBDT6	fMTn0 BBDT5	fMTn0 BBDT4	fMTn0 BBDT3	fMTn0 BBDT2	fMTn0 BBDT1	fMTn0 BBDT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn0BBDT(MTMn-Ch0 Buffer B Data)

It is used in composing the double buffer structure in MTMn-Ch0 reader B block of the counter reader function and MTMn-Ch0 sentinel B block of the counter sentinel function in MTMn-Ch0.

12-3-3-14 MTMn-Ch0 pre-scaler setting register : rMTn0PSCL

[Base + 0x0068]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fMTn0 PSCL7	fMTn0 PSCL6	fMTn0 PSCL5	fMTn0 PSCL4	fMTn0 PSCL3	fMTn0 PSCL2	fMTn0 PSCL1	fMTn0 PSCL0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0 : fMTn0PSCL(MTMn-Ch0 Prescaler Preset Data)
It is a set value to MTMn-Ch0 pre-scaler. The relation to the clock which MTMn-Ch0 pre-scaler generates and a setting value of fMTn0PSCL becomes the following.

Clock supplied to MTMn-Ch0[Hz] = XT1 Clock[Hz] ÷ (fMTn0PSCL + 1)

Note) XT1 is a clock generated by the oscillation amplifier.

Attention): There is a limitation in a minimum setting value of fMTn0PSCL register.
Minimum setting value: FMTn0PSCL=0x0002.

12-3-3-15 MTMn-Ch0 interrupt control register : rMTn0TIER

[Base + 0x006C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	fMTn0 OVE	fMTn0 IMEB	fMTn0 IMEA
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2 : fMTn0OVE

It controls the generation of the interrupt request by the overflow of the MTMn-Ch0 counter.

0: The use of the overflow interrupt of MTMn-Ch0 is prohibited.

1: The use of the overflow interrupt of MTMn-Ch0 is permitted.

Bit1 : fMTn0IMEB

It countrols the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch0B register.

0: The use of input-capture / compare-match interrupt B of MTMn-Ch0 is prohibited.

1: The use of input capture / compare-match interrupt B of MTMn-Ch0 is permitted.

Bit0 : fMTn0IMEA

It controls the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch0A register.

0: The use of input-capture / compare-match interrupt A of MTMn-Ch0 is prohibited.

1: The use of input-capture / comapre-match interrupt A of MTMn-Ch0 is permitted.

12-3-3-16 MTMn-Ch0 interrupt status register : rMTn0TISR

[Base + 0x0070]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	fMTn0 OVS	fMTn0 IMSB	fMTn0 IMSA
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit2 : fMTn0OVS

It shows the generation of the interrupt request by the overflow of the MTMn-Ch0 counter.

0: the overflow interrupt request of MTMn-Ch0 is not generated.

1: the overflow interrupt request of MTMn-Ch0 was generated.

Bit1 : fMTn0IMSB

It shows the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch0B register.

0: input-capture / compare-match interrupt B request of MTMn-Ch0 is not generated.

1: input-capture / compare-match interrupt B request of MTMn-Ch0 was generated.

Bit0 : fMTn0IMSA

It shows the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch0A register.

0: input-capture / compare-match interrupt A request of MTMn-Ch0 is not generated.

1: input-capture / compare-match interrupt A request of MTMn-Ch0 was generated.

12-3-3-17 MTMn-Ch1 control register : rMTn1CTL

[Base + 0x0080]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 CLST1	fMTn1 CLST0	fMTn1 EXEG1	fMTn1 EXEG0	fMTn1 SCCK3	fMTn1 SCCK2	fMTn1 SCCK1	fMTn1 SCCK0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-6 : fMTn1CLST1-0(MTMn-Ch1 Counter Clear Condition Setting)
The condition of clearing the counter of MTMn-Ch1 is set.
00 b: The counter of MTMn-Ch1 is not cleared.
01 b: The counter of MTMn-Ch1 is cleared by the event regarding MTMn-Ch1 reader A block and MTMn-Ch1 sentinel A block.
10 b: The counter of MTMn-Ch1 is cleared by the event regarding MTMn-Ch1 reader B block and MTMn-Ch1 sentinel B block.
11 b: The counter of MTMn-Ch1 is cleared when the counter in other MTMsn which is a synchronous object is cleared.

Bit5-4 : fMTn1EXEG1-0(MTMn-Ch1 External Clock Edge Select)
When the external clock has been selected as a counter source clock of MTMn-Ch1, the clock edge of the counter is selected.
00 b: The counter of MTMn-Ch1 is counted with the rising edge of the counter source clock.
01 b: The counter of MTMn-Ch1 is counted with the falling edge of the counter source clock.
10 b/11 b: The counter of MTMn-Ch1 is counted with the both edges of the counter source clock.

Bit3-0 : fMTn1SCCK(MTMn-Ch1 Counter Source Clock Select)
The source clock of the counter of MTMn-Ch1 is selected.
0000 b: internal clock. It is internal clock supplied from the pre-scaler of MTMn-Ch1.
1000 b: external clock. It is TCLKnA signal supplied from outside.
1001 b: external clock. It is TCLKnB signal supplied from outside.

Attention): Setting other than the above is prohibited.

12-3-3-18 MTMn-Ch1 IO control register : rMTn1IOCL

[Base + 0x0084]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	fMTn1 BFNC2	fMTn1 BFNC1	fMTn1 BFNC0	-	fMTn1 AFNC2	fMTn1 AFNC1	fMTn1 AFNC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit6-4 : fMTn1BFNC2-0(MTMn-Ch1 Block-B Function Setting)
The usage of MTMn-Ch1 reader B block of the counter reader function in MTMn-Ch1 or the usage of MTMn-Ch1 sentinel B block of the counter sentinel function in MTMn-Ch1 is set.

000 b: SNTn1BC1 event is generated when the value of MTMn-Ch1B register and the value of counter match up, but SNTINFn1B signal is not generated.
001 b: SNTn1BC1 event is generated when the value of MTMn-Ch1B register and the value of counter match up, and SNTINFn1B signal is set to 0.
010 b: SNTn1BC1 event is generated when the value of MTMn-Ch1B register and the value of counter match up, and SNTINFn1B signal is set to 1.
011 b: SNTn1BC1 event is generated when the value of MTMn-Ch1B register and the value of counter match up, and SNTINFn1B signal is reversed.
100 b: RDn1B event is generated at the rising edge of PLRn1B signal, and the counter value is transmitted to MTMn-Ch1B register.
101 b: RDn1B event is generated at the falling edge of PLRn1B signal, and the counter value is transmitted to MTMn-Ch1B register.
110 b/111 b: RDn1B event is generated at the rising edge or falling edge of PLRn1B signal, and the counter value is transmitted to MTMn-Ch1B register.

Remarks) When fMTn1BFNC is set to the value from 000 b to 011 b, the use of MTMn-Ch1 sentinel B block becomes possible, and the use of MTMn-Ch1 reader B block becomes impossible. When fMTn1BFNC is set to the value from 100 b to 111 b, the use of MTMn-Ch1 reader B block becomes possible, and the use of MTMn-Ch1 sentinel B block becomes impossible.

Bit2-0 : fMTn1AFNC2-0(MTMn-Ch1 Block-A Function Setting)
The usage of MTMn-Ch1 reader A block of the counter reader function in MTMn-Ch1 or the usage of MTMn-Ch1 sentinel A block of the counter sentinel function in MTMn-Ch1 is set.

000 b: SNTn1AC1 event is generated when the value of MTMn-Ch1A register and the value of counter match up, but SNTINFn1A signal is not generated.
001 b: SNTn1AC1 event is generated when the value of MTMn-Ch1A register and the value of counter match up, and SNTINFn1A signal is set to 0.
010 b: SNTn1AC1 event is generated when the value of MTMn-Ch1A register and the value of counter match up, and SNTINFn1A signal is set to 1.
011 b: SNTn1AC1 event is generated when the value of MTMn-Ch1A register and the value of counter match up, and SNTINFn1A signal is reversed.
100 b: RDn1A event is generated at the rising edge of PLRn1A signal, and the counter value is transmitted to MTMn-Ch1A register.
101 b: RDn1A event is generated at the falling edge of PLRn1A signal, and the counter value is transmitted to MTMn-Ch1A register.
110 b/111 b: RDn1A event is generated at the rising edge or falling edge of PLRn1A signal, and the counter value is transmitted to MTMn-Ch1A register.

Remarks) When fMTn1AFNC is set to the value from 000 b to 011 b, the use of MTMn-Ch1 sentinel A block becomes possible, and the use of MTMn-Ch1 reader A block becomes impossible. When fMTn1AFNC is set to the value from 100 b to 111 b, the use of MTMn-Ch1 reader A block becomes possible, and the use of MTMn-Ch1 sentinel A block becomes impossible.

12-3-3-19 MTMn-Ch1 status register : rMTn1STS

[Base + 0x008C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	fMTn1 OVF	fMTn1 BEVT	fMTn1 AEVT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2 : fMTn1OVF(MTMn-Ch1 Overflow Flag)
It shows whether the counter of MTMn-Ch1 overflowed or not.
At Read
0: The counter of MTMn-Ch1 has not overflowed.
1: The counter of MTMn-Ch1 overflowed.

At Write
0: fMTn1OVF is not influenced.
1: fMTn1OVF is cleared.

Remarks): This fMTn1OVF is transmitted to the interrupt control part of the interrupt function as MTMn-Ch1 counter interrupt factor.

Bit1 : fMTn1BEVT(MTMn-Ch1 Block-B Event Flag)
It shows whether the RDn1B event caused by MTMn-Ch1 reader B block or the SNTn1BC1 event caused by MTMn-Ch1 sentinel B block was generated or not.
At Read
0: The event regarding MTMn-Ch1 reader B block or MTMn-Ch1 sentinel B block is not generated in MTMn-Ch1.
1: The event regarding MTMn-Ch1 reader B block or MTMn-Ch1 sentinel B block is generated in MTMn-Ch1.

At Write
0: fMTn1BEVT is not influenced.
1: fMTn1BEVT is cleared.

Remarks): This fMTn1BEVT is transmitted to the interrupt control part of the interrupt function as MTMn1B interrupt factor.

Bit0 : fMTn1AEVT(MTMn-Ch1 Block-Event Flag)

It shows whether the RDn1A event caused by MTMn-Ch1 reader A block or the SNTn1AC1 event caused by MTMn-Ch1 sentinel A block was generated or not.

At Read

0: The event regarding MTMn-Ch1 reader A block or MTMn-Ch1 sentinel A block is not generated in MTMn-Ch1.

1: The event regarding MTMn-Ch1 reader A block or MTMn-Ch1 sentinel A block is generated in MTMn-Ch1.

At Write

0: fMTn1AEVT is not influenced.

1: fMTn1AEVT is cleared.

Remarks): This fMTn1AEVT is transmitted to the interrupt control part of the interrupt function as MTMn1A interrupt factor.

Attention): Above-mentioned Bit2-0 is Read/Clear bit. Writing directly to this bit by the program is not permitted though reading to the bit is free. Only the instruction of the bit clearness is permitted to the program.

Please store 1 to the bit position in the register corresponding to the Read/Clear bit that wants to clear when you want to clear the Read/Clear bit. As a result, the Read/Clear bit becomes 0.

On the other hand, even if 0 is stored to the bit position in the register corresponding to the Read/Clear bit, the data in the corresponding Read/Clear bit is not updated.

12-3-3-20 MTMn-Ch1 counter : rMTn1CNT

[Base + 0x0090]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn1 CNDT15	fMTn1 CNDT14	fMTn1 CNDT13	fMTn1 CNDT12	fMTn1 CNDT11	fMTn1 CNDT10	fMTn1 CNDT9	fMTn1 CNDT8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 CNDT7	fMTn1 CNDT6	fMTn1 CNDT5	fMTn1 CNDT4	fMTn1 CNDT3	fMTn1 CNDT2	fMTn1 CNDT1	fMTn1 CNDT0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn1CNDT(MTMn-Ch1 Counter Data)
It shows a count value of the MTMn-Ch1 counter. Setting a value to the MTMn-Ch1 counter is also possible.

12-3-3-21 MTMn-Ch1A register : rMTn1A

[Base + 0x0094]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn1 ADT15	fMTn1 ADT14	fMTn1 ADT13	fMTn1 ADT12	fMTn1 ADT11	fMTn1 ADT10	fMTn1 ADT9	fMTn1 ADT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 ADT7	fMTn1 ADT6	fMTn1 ADT5	fMTn1 ADT4	fMTn1 ADT3	fMTn1 ADT2	fMTn1 ADT1	fMTn1 ADT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn1ADT(MTMn-Ch1 A Register Data)
It is used in MTMn-Ch1 reader A block of the counter reader function and MTMn-Ch1 sentinel A block of the counter sentinel function in MTMn-Ch1.

12-3-3-22 MTMn-Ch1B register : rMTn1B

[Base + 0x0098]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn1 BDT15	fMTn1 BDT14	fMTn1 BDT13	fMTn1 BDT12	fMTn1 BDT11	fMTn1 BDT10	fMTn1 BDT9	fMTn1 BDT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 BDT7	fMTn1 BDT6	fMTn1 BDT5	fMTn1 BDT4	fMTn1 BDT3	fMTn1 BDT2	fMTn1 BDT1	fMTn1 BDT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn1BDT(MTMn-Ch1 B Register Data)
It is used in MTMn-Ch1 reader B block of the counter reader function and MTMn-Ch1 sentinel B block of the counter sentinel function in MTMn-Ch1.

12-3-3-23 MTMn-Ch1 output logic setting register : rMTn1SOL

[Base + 0x009C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	fMTn1 BOL	fMTn1 AOL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : fMTn1BOL(MTMn-Ch1 Sentinel-B Output Logic Setting Flag)
It shows the level of the SNTINFn1B signal in the MTMn-Ch1 sentinel inform function. It is also possible to control the SNTINFn1B signal directly by updating this fMTn1BOL with the program.
0: set the level of SNTINFn1B signal to low
1: set the level of SNTINFn1B signal to high

Remarks): This fMTn1BOL can be used for the setting of the initial state of the SNTINFn1B signal in the sentinel inform function.

Bit0 : fMTn1AOL(MTMn-Ch1 Sentinel-A Output Logic Setting Flag)
It shows the level of the SNTINFn1A signal in the MTMn-Ch1 sentinel inform function or the PWMn1 signal in standard PWM mode. It is also possible to control the SNTINFn1A signal or the PWMn1 signal directly by updating this fMTn1AOL with the program.

0: set the level of SNTINFn1A signal or PWMn1 signal to low.
1: set the level of SNTINFn1A signal or PWMn1 signal to high.

Remarks): This fMTn1AOL can be used for the setting of the initial state of the SNTINFn1A signal in the sentinel inform function or the PWMn1 signal in the standard PWM mode.

12-3-3-24 MTMn-Ch1 buffer A : rMTn1BA

[Base + 0x00A0]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn1 BADT15	fMTn1 BADT14	fMTn1 BADT13	fMTn1 BADT12	fMTn1 BADT11	fMTn1 BADT10	fMTn1 BADT9	fMTn1 BADT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 BADT7	fMTn1 BADT6	fMTn1 BADT5	fMTn1 BADT4	fMTn1 BADT3	fMTn1 BADT2	fMTn1 BADT1	fMTn1 BADT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn1BADT(MTMn-Ch1 Buffer A Data)

It is used in composing the double buffer structure in MTMn-Ch1 reader A block of the counter reader function and MTMn-Ch1 sentinel A block of the counter sentinel function in MTMn-Ch1.

12-3-3-25 MTMn-Ch1 buffer B : rMTn1BB

[Base + 0x00A4]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	fMTn1 BBDT15	fMTn1 BBDT14	fMTn1 BBDT13	fMTn1 BBDT12	fMTn1 BBDT11	fMTn1 BBDT10	fMTn1 BBDT9	fMTn1 BBDT8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 BBDT7	fMTn1 BBDT6	fMTn1 BBDT5	fMTn1 BBDT4	fMTn1 BBDT3	fMTn1 BBDT2	fMTn1 BBDT1	fMTn1 BBDT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : fMTn1BBDT(MTMn-Ch1 Buffer B Data)

It is used in composing the double buffer structure in MTMn-Ch1 reader B block of the counter reader function and MTMn-Ch1 sentinel B block of the counter sentinel function in MTMn-Ch1.

12-3-3-26 MTMn-Ch1 pre-scaler setting register : rMTn1PSCL

[Base + 0x00A8]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fMTn1 PSCL7	fMTn1 PSCL6	fMTn1 PSCL5	fMTn1 PSCL4	fMTn1 PSCL3	fMTn1 PSCL2	fMTn1 PSCL1	fMTn1 PSCL0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0 : fMTn1PSCL(MTMn-Ch1 Prescaler Preset Data)
It is a set value to MTMn-Ch1 pre-scaler. The relation to the clock which MTMn-Ch1 pre-scaler generates and a setting value of fMTn1PSCL becomes the following.

Clock supplied to MTMn-Ch1[Hz] = XT1 Clock[Hz] ÷ (fMTn1PSCL + 1)

Note) XT1 is a clock generated by the oscillation amplifier.

Attention): There is a limitation in a minimum setting value of fMTn1PSCL register.
Minimum setting value: FMTn1PSCL=0x0002.

12-3-3-27 MTMn-Ch1 interrupt control register : rMTn1TIER

[Base + 0x00AC]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	fMTn1 OVE	fMTn1 IMEB	fMTn1 IMEA
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2 : fMTn1OVE

It controls the generation of the interrupt request by the overflow of the MTMn-Ch1 counter.

0: The use of the overflow interrupt of MTMn-Ch1 is prohibited.

1: The use of the overflow interrupt of MTMn-Ch1 is permitted.

Bit1 : fMTn1IMEB

It countrols the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch1B register.

0: The use of input-capture / compare-match interrupt B of MTMn-Ch1 is prohibited.

1: The use of input capture / compare-match interrupt B of MTMn-Ch1 is permitted.

Bit0 : fMTn1IMEA

It controls the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch1A register.

0: The use of input-capture / compare-match interrupt A of MTMn-Ch1 is prohibited.

1: The use of input-capture / comapre-match interrupt A of MTMn-Ch1 is permitted.

12-3-3-28 MTMn-Ch1 interrupt status register : rMTn1TISR

[Base + 0x00B0]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	fMTn1 OVS	fMTn1 IMSB	fMTn1 IMSA
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit2 : fMTn1OVS

It shows the generation of the interrupt request by the overflow of the MTMn-Ch1 counter.

0: the overflow interrupt request of MTMn-Ch1 is not generated.

1: the overflow interrupt request of MTMn-Ch1 was generated.

Bit1 : fMTn1IMSB

It shows the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch1B register.

0: input-capture / compare-match interrupt B request of MTMn-Ch1 is not generated.

1: input-capture / compare-match interrupt B request of MTMn-Ch1 was generated.

Bit0 : fMTn1IMSA

It shows the generation of the interrupt request by input-capture / compare-match of the MTMn-Ch1A register.

0: input-capture / compare-match interrupt A request of MTMn-Ch1 is not generated.

1: input-capture / compare-match interrupt A request of MTMn-Ch1 was generated.

12-4 Operation

This section provides information about MTMs operation.

12-4-1 Counter source clock control function

Counter source clock control function selects the source clock of the counter and generates the demand signal for count.

12-4-1-1 Selection of counter source clock

As for the clock which can be used as a counter source clock in MTMsn, the internal clock supplied by pre-scaler and 4 kinds of external clocks are prepared. Which of these clock sources is supplied to the counter of MTMn-Chc as a counter source clock is decided by setting fMTncSCCK of MTMn-Chc control register.

Only when either of external clocks is chosen as a counter clock source, the count condition can be set.

As for the count condition, three conditions of the rising edge, the falling edge and both edges of the counter source clock are prepared, and the count condition of the counter can be selected by setting fMTncEXEG of MTMn-Chc control register.

Note) c: 0-1

12-4-1-2 Generation of demand signal for count and count operation

The timing chart of the MTMn-Chc counter is shown in the following.

- In case of selecting the internal clock to the counter source clock

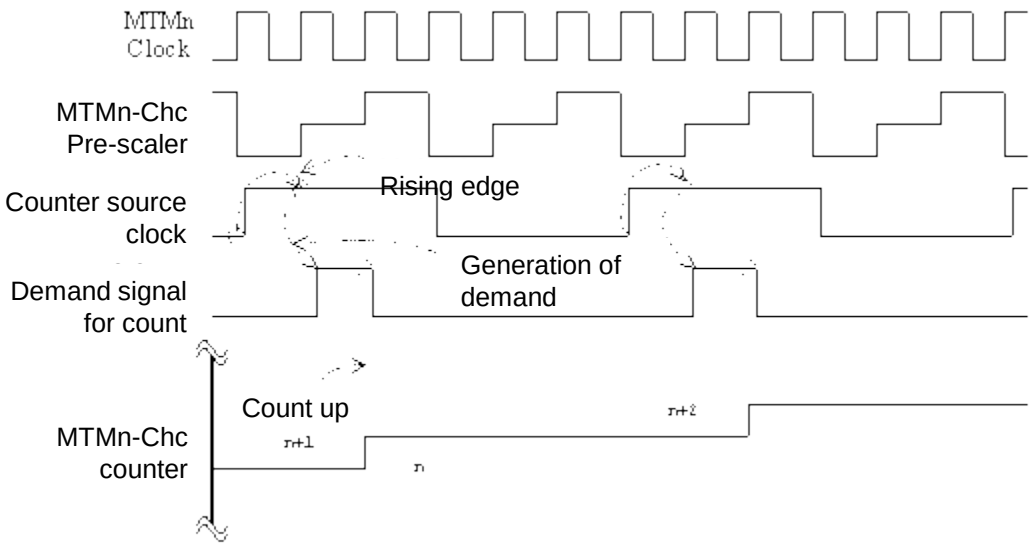


Figure 12-2

- In case of selecting the external clock to the counter source clock

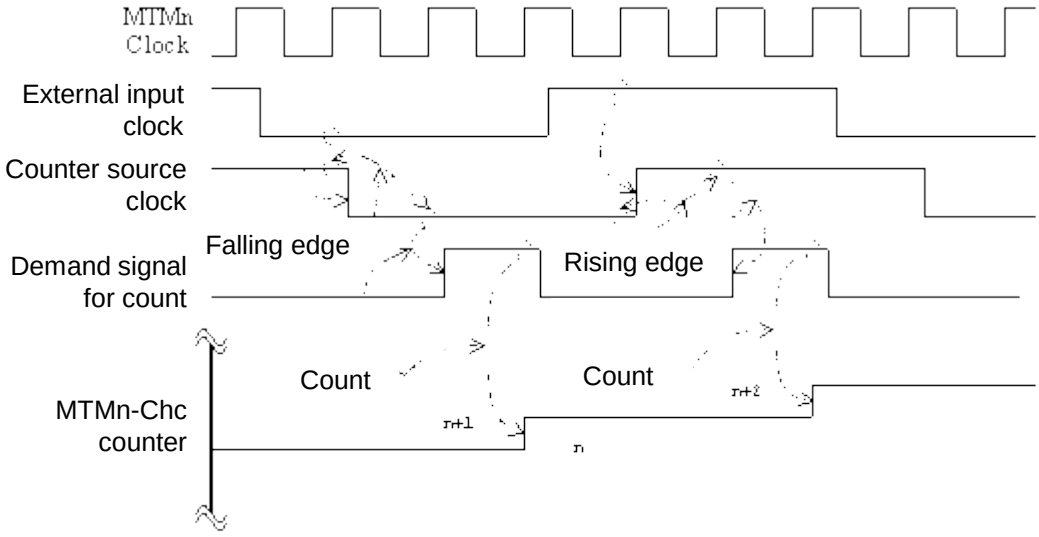


Figure 12-3

These are figures to use both edges as a count edge. The pulse width of the external input clock needs 1.5 systems clock or more in a single edge of the rising or falling, and it needs 2.5 systems clock or more in both edges. In the pulse width less than this, it is likely not to be counted correctly.

Note) c: 0-1

12-4-2 Counter synchronous function

In MTMsn, synchronization in the count operation between MTMn-Ch0 and MTMn-Ch1 can be achieved by a counter synchronous function. Two kinds of functions of a synchronous counter preset and a synchronous counter clearness to synchronize the counter are provided.

12-4-2-1 Synchronous counter preset

The preset timing to the counters is synchronized between timers selected to a synchronous object of the counter synchronous function in MTMn-Ch0 - MTMn-Ch1.

When the preset operation is done to the counter of either timer selected to a synchronous object, the same value is also preset to the counter of the other timer of the synchronous object at the same time. The preset to the counter can be done by the store to fMTncCNDT of the MTMn-Chc counter.

MTMsn synchronous setting register and MTMn-Chc control register are used to set an arbitrary timer to an object of the synchronous counter preset. Arbitrary timer MTMn-Chc is selected as a target of a synchronous counter preset by setting an arbitrary timer to the object of the counter synchronous function with fMTncSYNC of MTMsn synchronous setting register, and prohibitting the use of the synchronous counter clearness function (FMTncCLST is concretely made to be excluding 11 b) with fMTncCLST of MTMn-Chc control register.

12-4-2-2 Synchronous counter clearness

The clearness timing to the counters is synchronized between timers selected to a synchronous object of the counter synchronous function in MTMn-Ch0 - MTMn-Ch1.

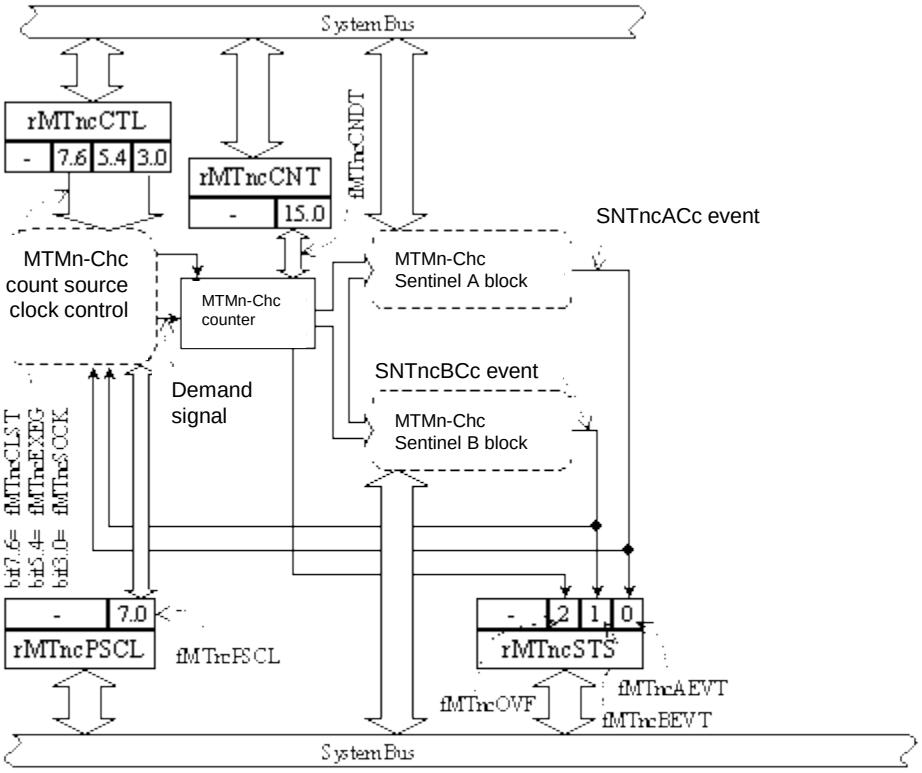
When clearness to the counter of either timer selected to a synchronous object is generated, the counter of the other timer of the synchronous object is also cleared at the same time.
A clear factor of the counter caused synchronization by synchronous counter clearness function is either SNTncXCc event which MTMn-Chc sentinel X block generates or RDncX event which MTMn-Chc reader X block generates. Both updating to count value =0 as the count result of the counter and writing 0 to the MTMn-Chc counter are outside of the synchronous object of the synchronous counter clearness function.

MTMsn synchronous setting register and MTMn-Chc control register are used to set an arbitrary timer to an object of the synchronous counter clearness. Arbitrary timer MTMn-Chc is selected as a target of a synchronous counter clearness by setting an arbitrary timer to the object of the counter synchronous function with fMTncSYNC of MTMsn synchronous setting register, and permitting the use of the synchronous counter clearness function (FMTncCLST is concretely made to be 11 b) with fMTncCLST of MTMn-Chc control register.

Note) c:0-1

12-4-3 Standard timer operation

When using MTMns as a standard timer thich measures prescribed time and makes a constant period, it is possible to implement the timer with the following composition.



*MTMn-Chc sentinel A block : rMTncA and rMTncBA are included in the block.
*MTMn-Chc sentinel B block : rMTncB and rMTncBB are included in the block.
*c: 0-1

Figure 12-4 MTMn-Chc composition at standard timer

Moreover, sentinel block becomes the following buffer structure by setting the MTMsn function control register.

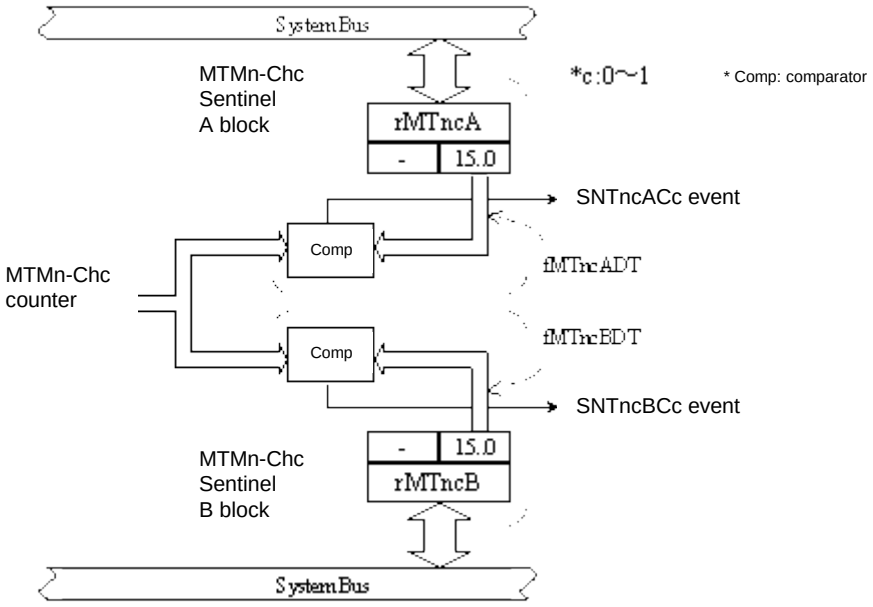


Figure 12-5 Single buffer sentinel structure

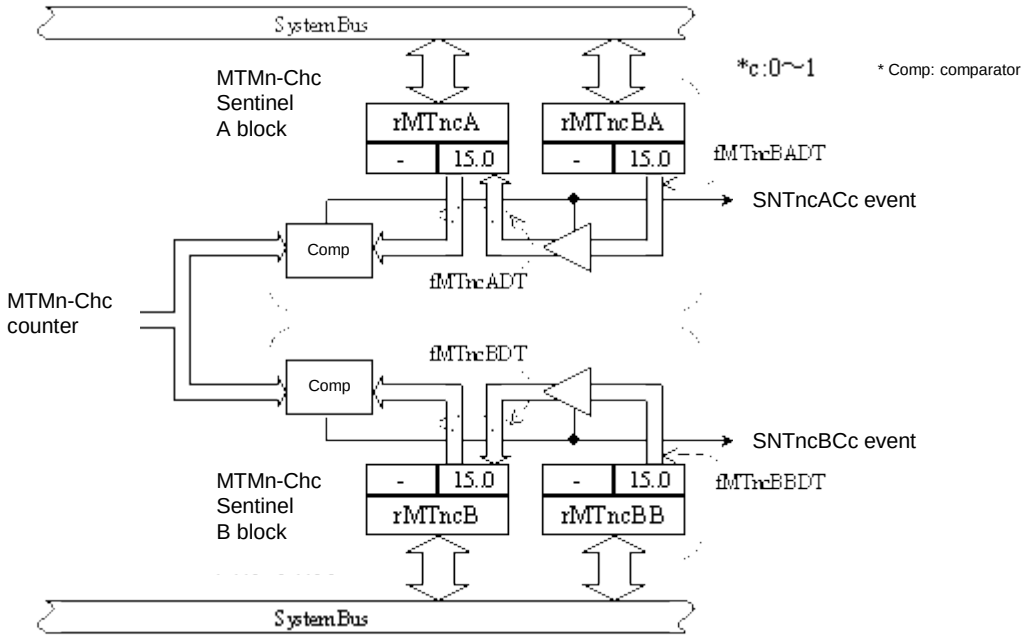
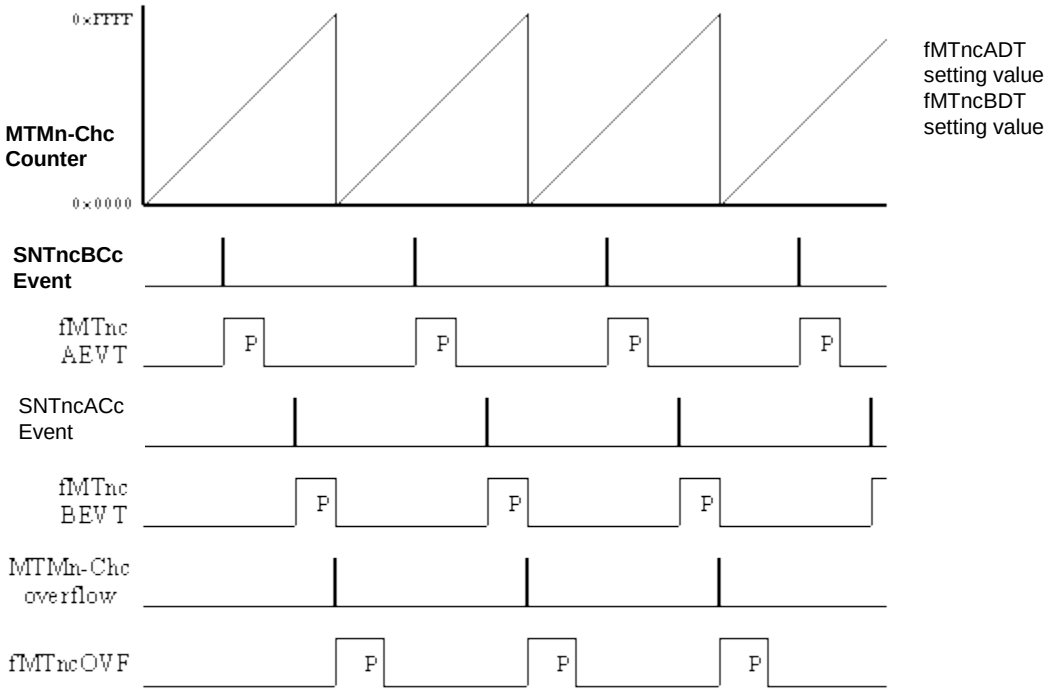


Figure 12-6 Double buffer sentinel structure

12-4-3-1 Free running

MTMn-Chc becomes a free running operation when MTMn-Chc is set to the state of the standard timer operation, in addition fMTncCLST of MTMn-Chc control register is set to 0 and it prohibits the clearness of the counter of MTMn-Chc.

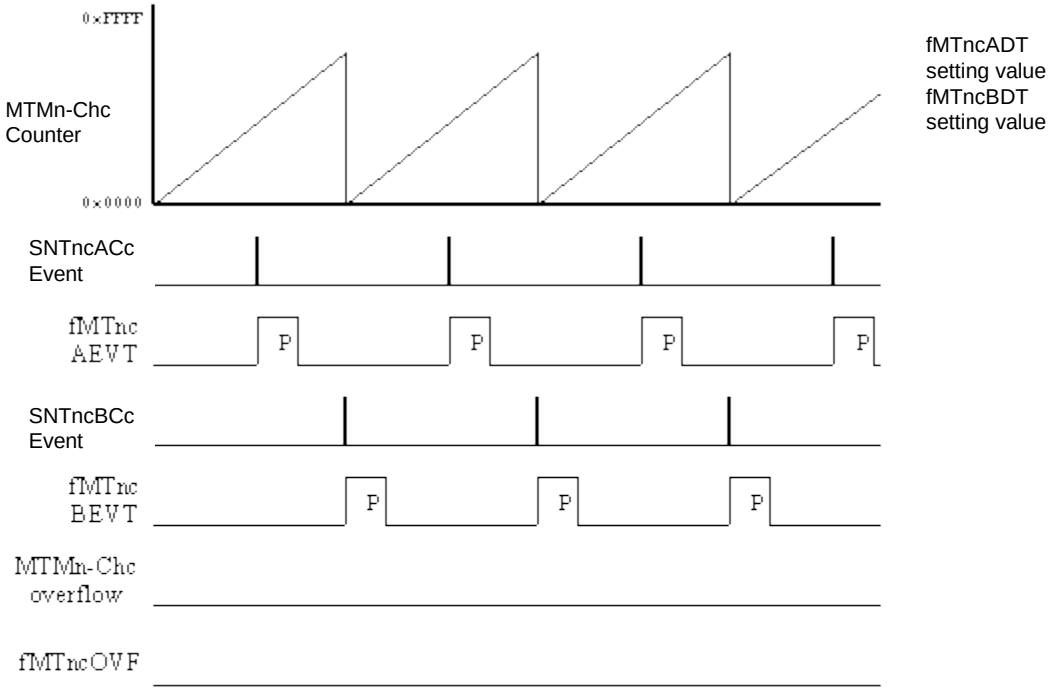


*c: 0-1
*P: The timing is controlled by program.
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=0 --- it doesn't generate PWMnc signal.
fMTncCLST=0 ----- it doesn't clear MTMn-Chc counter.
fMTncAFNC=0 ---- it sets the sentinel A block to be valid.
fMTncBFNC=0 ---- it sets the sentinel B block to be valid.

Figure 12-7 MTMn-Chc free running

12-4-3-2 Cycle timer

MTMn-Chc becomes possible to make a prescribed cycle when MTMn-Chc is set to the state of the standard timer operation, in addition fMTncCLST of MTMn-Chc control register is set to clear the counter of MTMn-Chc with generating SNTncACc event or SNTncBCc event.



*c: 0-1
*P: The timing is controlled by program.
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=0 --- it doesn't generate PWMnc signal.
fMTncCLST=1 ----- it clears MTMn-Chc counter with SNTncACc event.
fMTncAFNC=0 ---- it sets the sentinel A block to be valid.
fMTncBFNC=0 ---- it sets the sentinel B block to be valid.
fMTncADT=cycle – it sets the cycle made by the counter of MTMn-Chc.

Figure 12-8 MTMn-Chc cycle timer

12-4-4 Pulse length reader function

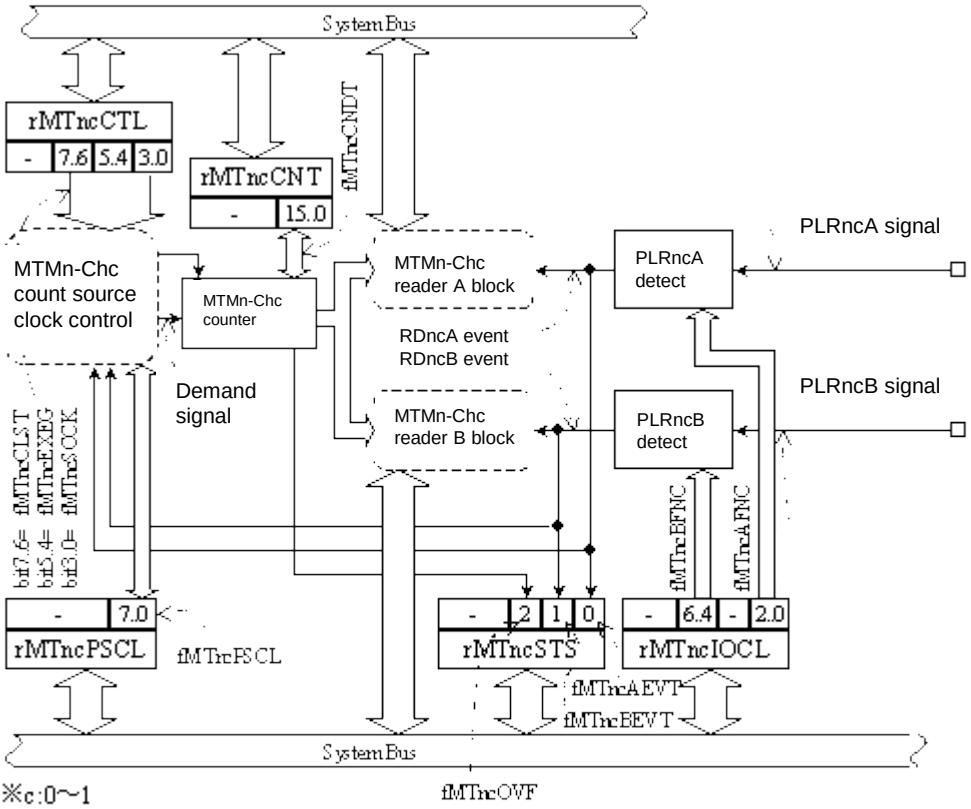
Pulse length reader function: PLRn enables pulse width and cycle of the signal input from the outside to be measured easily by using the counter reader function.

An independent pulse length reader function is prepared in MTMn-Ch0 - MTMn-Ch1 respectively, and each of them is divided into A block and B block further as for these functions. On the other hand, the target signal for the measurement with the pulse length reader function is a following signal of each channel.

Table 12-3 Composition list of pulse length reader function

MTMs	Function	Function according to block	Event	Signal for measurement
MTMn-Ch1	MTMn-Ch1 pulse length reader function: PLRn1	MTMn-Ch1 pulse length reader function B: PLRn1B	RDn1B event	PLRn1B signal
		MTMn-Ch1 pulse length reader function A: PLRn1A	RDn1A event	PLRn1A signal
MTMn-Ch0	MTMn-Ch0 pulse length reader function: PLRn0	MTMn-Ch0 pulse length reader function B: PLRn0B	RDn0B event	PLRn0B signal
		MTMn-Ch0 pulse length reader function A: PLRn0A	RDn0A event	PLRn0A signal

The composition of the pulse length reader function is the following.



*MTMn-Chc reader A block : rMTncA and rMTncBA are included in the block.
*MTMn-Chc reader B block : rMTncB and rMTncBB are included in the block.

Figure 12-9 Conposition of pulse length reader function

Moreover, the reader block becomes the following buffer structure by setting the MTMsn function control register.

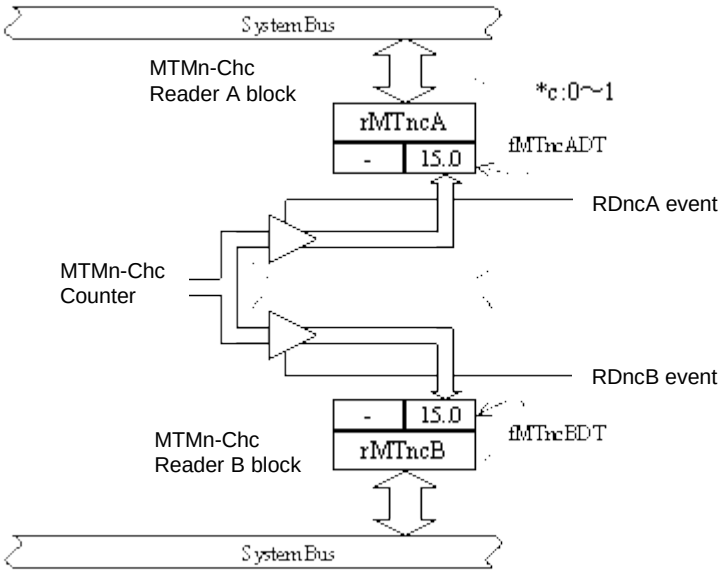


Figure 12-10 Single buffer reader sturucture

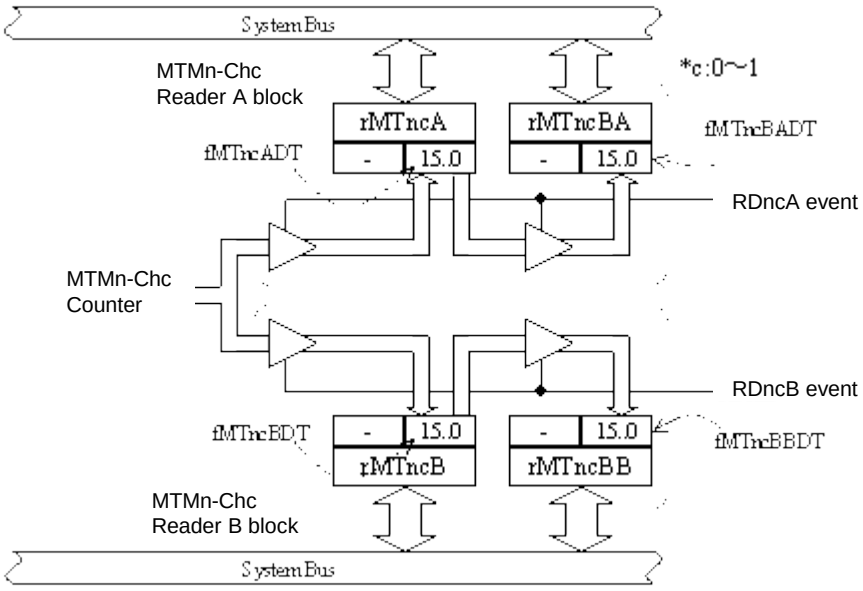


Figure 12-11 Double buffer reader structure

12-4-4-1 Pulse measurement

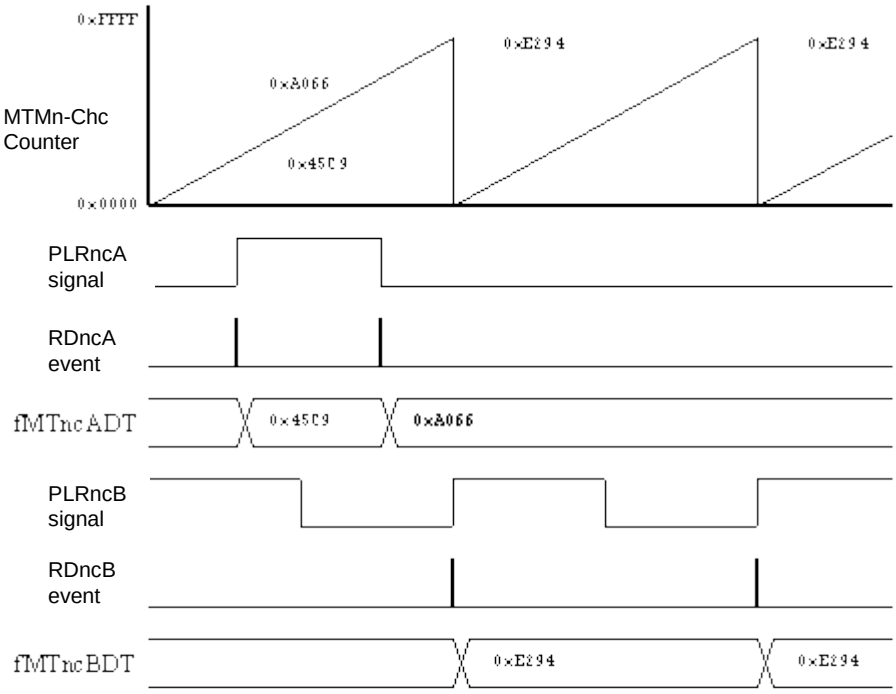
4 kinds of PLRncX signals (c:0 or 1, X:A or B) for the measurement are supplied from the outside of the module, and these are measured in each pulse length reader function corresponding to them.

In each pulse length reader function, setting a condition as the read trigger to the signal for measurement object is possible. When the PLRncX signal meets this condition as the read trigger, an RDncX event corresponding to the PLRncX signal is generated, and the count value of the MTMn-Chc counter at that time is updated to MTMn-Chc reader X block as a read data. The read trigger condition is set with fMTncXFNC of MTMn-Chc IO control register. There are three kinds of conditions which can be set as a read trigger the following.

- Rising edge of PLRncX signal
- Falling edge of PLRncX signal
- Both edge of PLRncX signal

These read trigger conditions can be individually set to all PLRncX. Therefore, even if the read trigger condition needed in the application is different respectively, it is possible to measure the pulse length.

Note): c:0-1, X:A or B



*c: 0-1
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=0 --- it doesn't generate PWMnc signal.
fMTncCLST=2 ----- it clears MTMn-Chc counter with RDncB event.
fMTncAFNC=6 ---- it sets the reader A block to be valid and generates RDncA event by detecting the both edges of PLRncA signal.
fMTncBFNC=4 ---- it sets the reader B block to be valid and generates RDncB event by detecting the rising edge of PLRncB signal.

Figure 12-12 Example of measuring pulse

12-4-5 Sentinel inform function

In sentinel inform function: SNTINFn, it is possible to output the signal outside which shows that the count value of the timer have reached the comparison value set in advance to use the counter sentinel function.

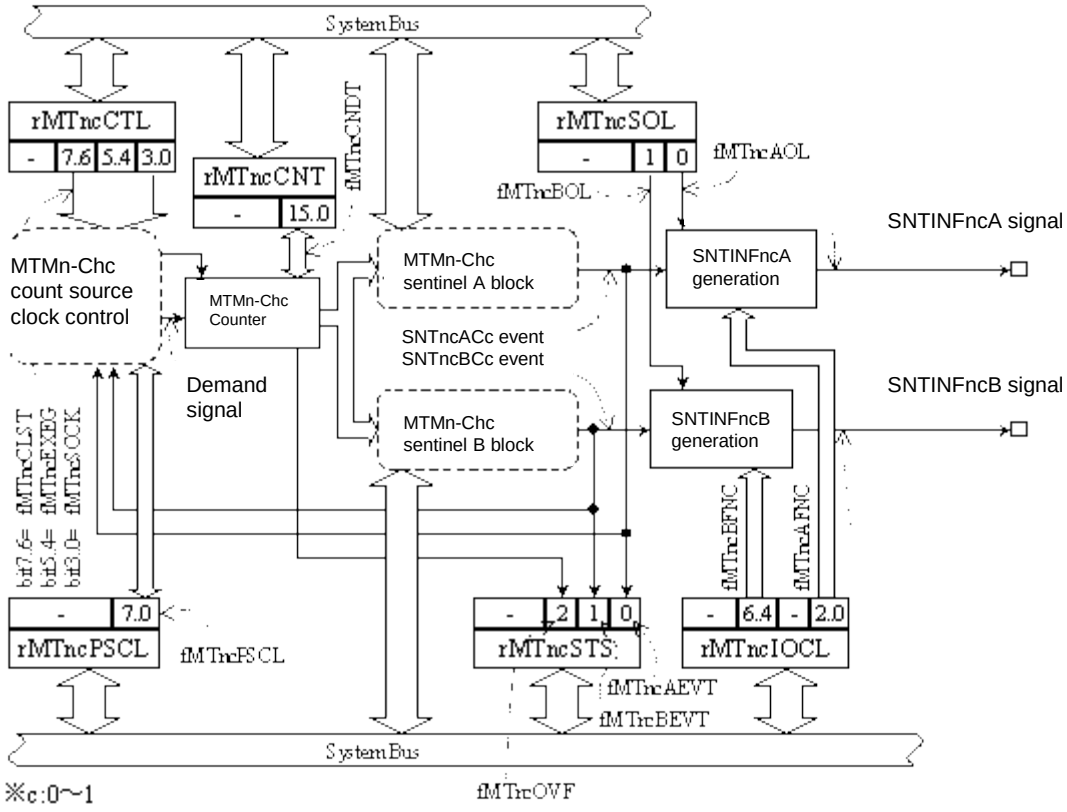
Moreover, it generates the sentinel inform signal by using MTMn-Chc sentinel A block or MTMn-Chc sentinel B block in MTMn-Chc.

Note): c: 0-1

Table 12-4 Composition list of sentinel inform function

MTMs	Function	Function according to block	Event	Generated signal
MTMn-Ch1	MTMn-Ch1 sentinel inform function: SNTINFn1	MTMn-Ch1 sentinel inform function B: SNTINFn1B	SNTn1BC1 event	SNTINFn1B signal
		MTMn-Ch1 sentinel inform function A: SNTINFn1A	SNTn1AC1 event	SNTINFn1A signal
MTMn-Ch0	MTMn-Ch0 sentinel inform function: SNTINFn0	MTMn-Ch0 sentinel inform function B: SNTINFn0B	SNTn0BC0 event	SNTINFn0B signal
		MTMn-Ch0 sentinel inform function A: SNTINFn0A	SNTn0AC0 event	SNTINFn0A signal

The composition of the sentinel inform function becomes the following.



*MTMn-Chc sentinel A block : rMTncA and rMTncBA are included in the block.
*MTMn-Chc sentinel B block : rMTncB and rMTncBB are included in the block.

Figure 12-13 Conposition of sentinel inform function

Moreover, sentinel block becomes the buffer structure of Figure 12-5, Figure 12-6 by setting the MTMsn function control register.

12-4-5-1 Sentinel inform signal generation

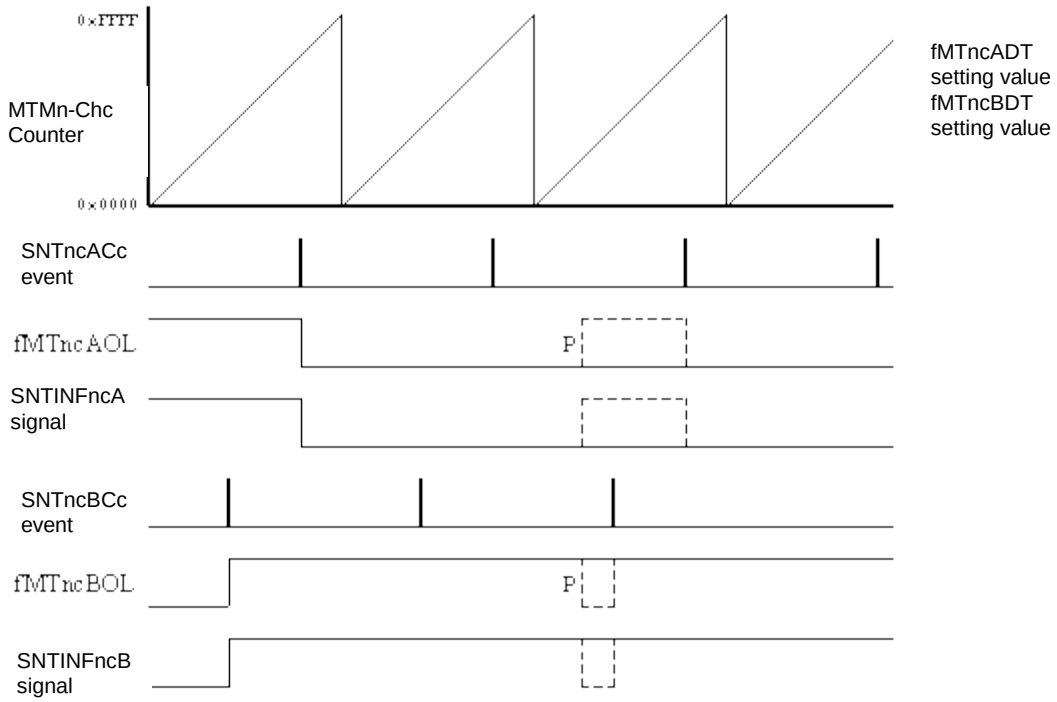
In MTMsn, the counter sentinel function of MTMn-Chc sentinel A block and MTMn-Chc sentinel B block is prepared for each of MTMn-Ch0 and MTMn-Ch1. These sentinel blocks generate each sentinel event: SNTncXCc(c:0 or 1, X:A or B) according to the operation situation. In the sentinel inform function, sentinel inform signal: SNTINFncX depending on this sentinel event is generated. It is possible to output this SNTINFncX signal outside of the module.

The sentinel event is generated when fMTncCNDT of MTMn-Chc counter corresponds with the comparison value set to MTMn-ChcX register.

It is possible to generate a sentinel inform signal based on this sentinel event. However, it is necessary to set the value to enable the generation of the sentinel inform signal to the registers for MTMn control to generate the sentinel inform signal. The registers which the setting is necessary for is fMTncXFNC of MTMn-Chc IO control register and fMTncXOL of the MTMn-Chc output logic setting register.

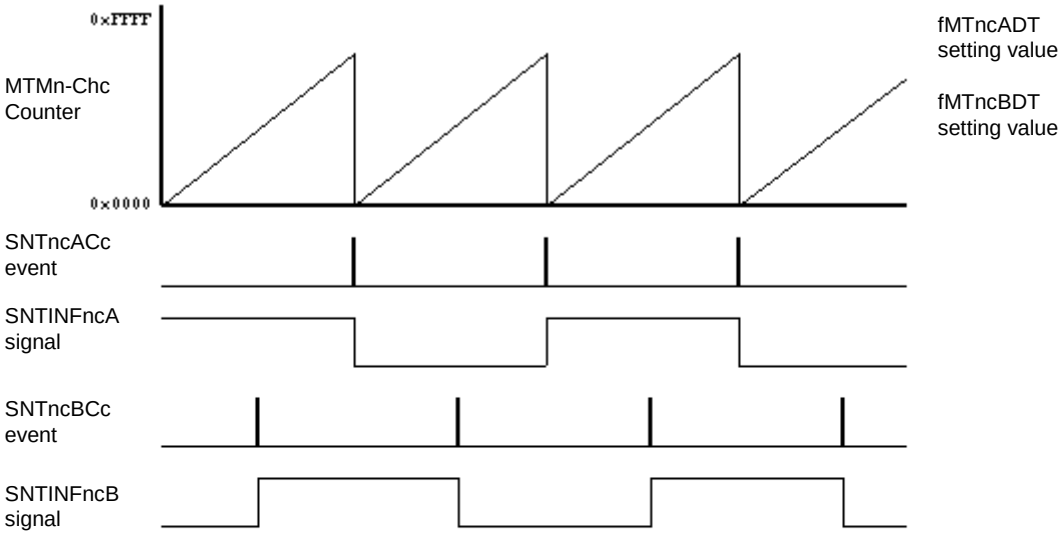
Two kinds of sentinel inform signal generation of a fixed level and a level reversing(toggle) is shown in the following as examples of generating the sentinel inform signal.

Note): c:0-1, X: A or B



*c: 0-1
*P: The timing is controlled by program.
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=0 --- it doesn't generate PWMnc signal.
fMTncCLST=0 ----- it doesn't clear MTMn-Chc counter.
fMTncAFNC=1 ---- it sets the sentinel A block to be valid and sets SNTINFncA signal to 0 based on SNTncACc event.
fMTncBFNC=2 ---- it sets the sentinel B block to be valid and sets SNTINFncB signal to 1 based on SNTncBCc event.

Figure 12-14 Generation of sentinel inform signal as a fixed level



*c: 0-1
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=0 --- it doesn't generate PWMnc signal.
fMTncCLST=1----- it clears MTMn-Chc counter with SNTncACc event.
fMTncAFNC=3 ---- it sets the sentinel A block to be valid and reverses SNTINFncA signal based on SNTncACc event.
fMTncBFNC=3 ---- it sets the sentinel B block to be valid and reverses SNTINFncB signal based on SNTncBCc event.

Figure 12-15 Generation of sentinel inform signal as a level reversing(toggle)

12-4-6 PWM output function

MTMsn is equipped with a PWM output function. This PWM output function can correspond to various usages on applications because it is prepared in each channel of MTMn-Ch0-MTMn-Ch1 respectively independently.

The PWM output function is equipped with standard PWM mode, and each of MTMn-Ch0 and MTMn-Ch1 corresponds to this mode.

The PWM output function has been achieved by making MTMn-Chc sentinel A block and MTMn-Chc sentinel B block in the counter sentinel function operate with a pair. Therefore, to use the PWM output function, you should enable the counter sentinel function of a channel (MTMn-Ch0 or MTMn-Ch1) that becomes an object to be effective, and set the timer for the PWM output function to operate.

MTMn-Chc sentinel A block and MTMn-Chc sentinel B block are used in a pair to generate PWM output signal (PWMnc). In MTMn-Chc sentinel A block, SNTncACc event sets PWM output signal (PWMnc) to “1” and SNTncBCc event sets PWMnc signal to “0”. When the standard PWM mode is enabled, SNTINFncA signal of sentinel inform function is not generated. However, SNTINFncB signal is generated according to SNTINFnc. Therefore, even when the standard PWM mode is enabled, SNTINFncB signal can be used.

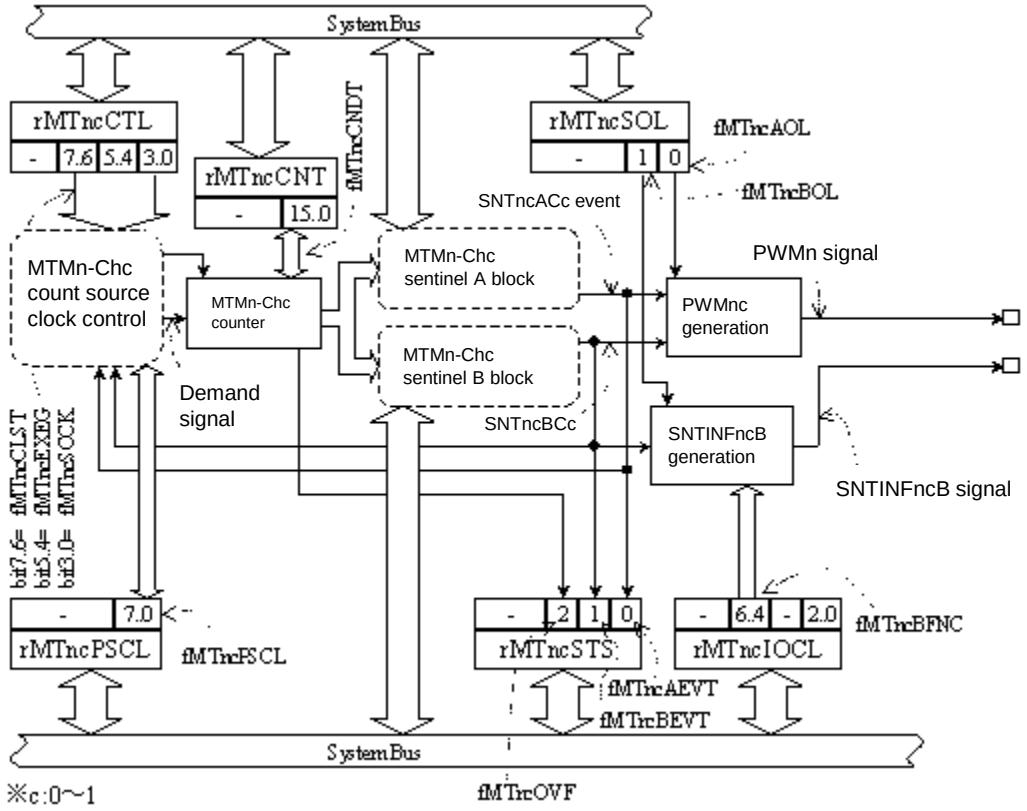
Table 12-5 Standard PWM mode configuration list

The oblique-line pattern means that it can be used as the sentinel inform function.

MTMs	Function	Event	Signals generated
MTMn-Ch1	MTMn-Ch1 PWM output function: PWMn1	SNTn1BC1 event	SNTINFn1B signal
		SNTn1AC1 event	PWMn1 signal
MTMn-Ch0	MTMn-Ch0 PWM output function: PWMn0	SNTn0BC0 event	SNTINF0B signal
		SNTn0AC0 event	PWMn0 signal

The following is the configuration of the standard PWM mode.

Note) c: 0 to 1



*MTMn-Chc sentinel A block : rMTncA and rMTncBA are included in the block.
*MTMn-Chc sentinel B block : rMTncB and rMTncBB are included in the block.

Figure 12-16 The configuration of the standard PWM mode

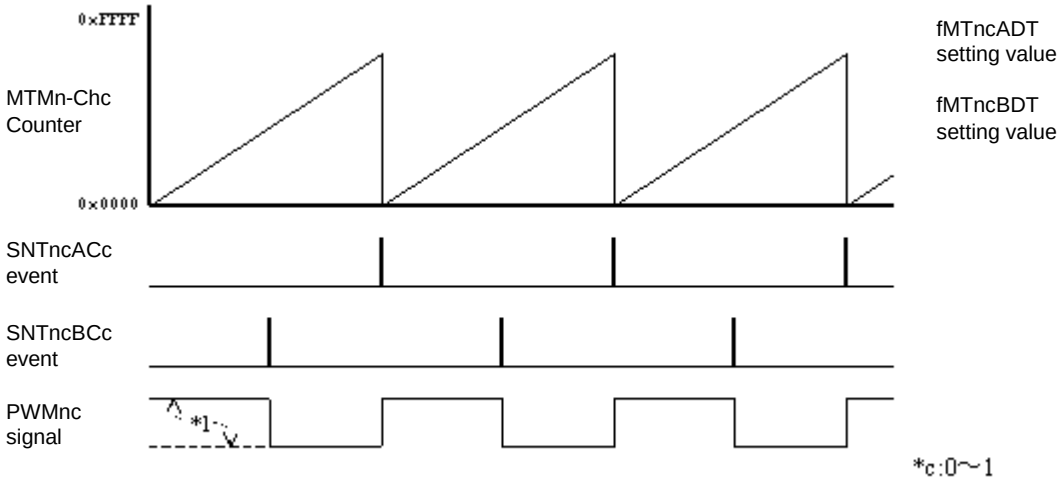
The sentinel block buffer structures set by MTMsn function control register are shown in Figure 12-5 and Figure 12-6.

12-4-6-1 PWM signal generation

The following are examples of PWM signal generation in the standard PWM mode.

- Example: PWMnc signal generation in the standard PWM mode

The following is an example in case that PWM signal cycle is set by fMTncADT of MTMn-ChcA register and the duty is set by fMTncBDT of MTMn-ChcB register. For fMTncADT to decide PWM signal cycle, set “1” in fMTncCLST of MTMn-Chc control register so that SNTncACc event clears MTMn-Chc counter.



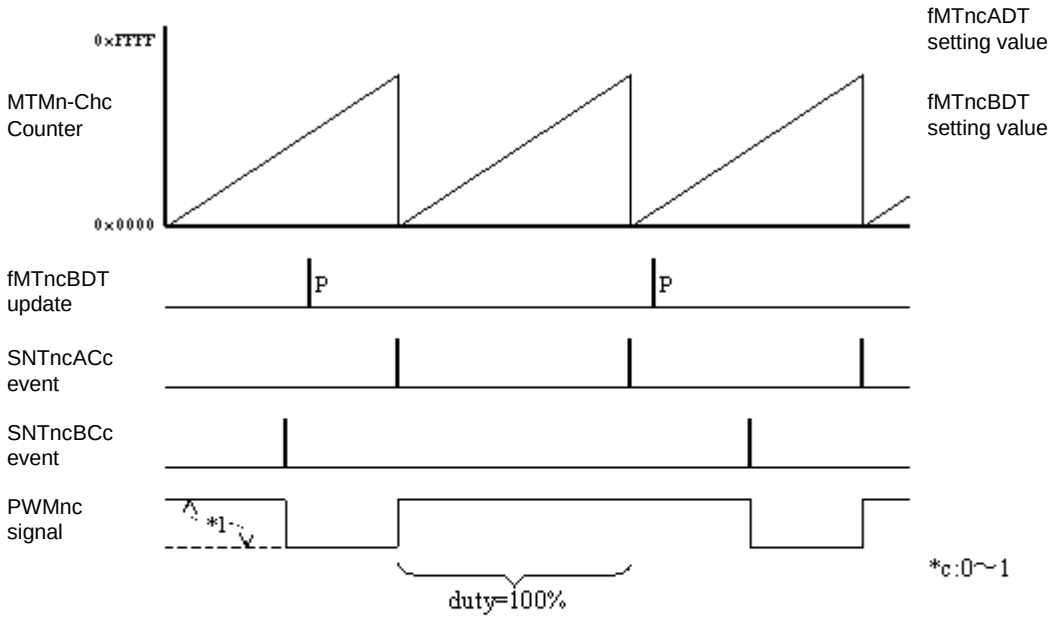
*c: 0-1
*1: It is solid line when fMTncAOL is set to 1,
it is broken line when fMTncAOL is set to 0.
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=1 --- it generates PWMnc signal.
fMTncCLST=1----- it clears MTMn-Chc counter with SNTncACc event.
fMTncBFNC=0-3 -- it sets the sentinel B block to be valid.
fMTncAOL ---- refer to *1

Figure 12-17 Example of PWMnc signal generation in the standard PWM mode

- Example: PWMnc signal generation with duty = 100% in the standard PWM mode

In order to generate PWMnc signal that includes duty = 100%, set MTMn-Chc so that PWM signal cycle is decided by fMTncADT of MTMn-ChcA register and the duty is decided by fMTncBDT of MTMn-ChcB register. For fMTncADT to decide PWM signal cycle, set “1” in fMTncCLST of MTMn-Chc control register so that SNTncACc event clears MTMn-Chc counter.

In order to generate PWM signal of duty = 100% in a cycle decided by fMTncADT, after SNTncBCc event occurs, update fMTncBDT by program to satisfy the relationship: fMTncADT < fMTncBDT. This operation generates PWM signal of duty = 100% in the next cycle.



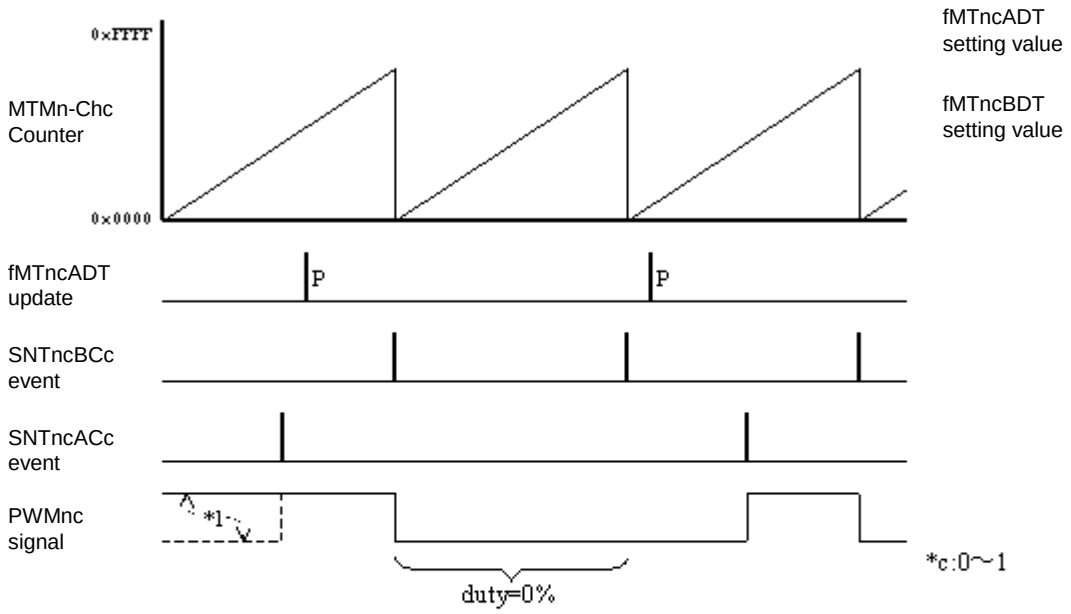
*c: 0-1
*1: It is solid line when fMTncAOL is set to 1,
it is broken line when fMTncAOL is set to 0.
*P: The timing is controlled by program.
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=1 --- it generates PWMnc signal.
fMTncCLST=1----- it clears MTMn-Chc counter with SNTncACc event.
fMTncBFNC=0-3 -- it sets the sentinel B block to be valid.
fMTncAOL ---- refer to *1

Figure 12-18 Example of PWMnc signal generation with duty = 100% in the standard PWM mode

Example: PWMnc signal generation with duty = 0% in the standard PWM mode

In order to generate PWMnc signal that includes duty = 0%, set MTMn-Chc so that PWM signal cycle is decided by fMTncBDT of MTMn-ChcB register and the duty is decided by fMTncADT of MTMn-ChcA register. For fMTncBDT to decide PWM signal cycle, set “2” in fMTncCLST of MTMn-Chc control register so that SNTncBCc event clears MTMn-Chc counter.

In order to generate PWM signal of duty = 0% in a cycle decided by fMTncBDT, after SNTncACc event occurs, update fMTncADT by program to satisfy the relationship: fMTncBDT < fMTncADT. This operation generates PWM signal of duty = 0% in the next cycle.



*c: 0-1
*1: It is solid line when fMTncAOL is set to 1,
it is broken line when fMTncAOL is set to 0.
*P: The timing is controlled by program.
*The setting to execute the timing chart above is as follows.
fMTncACT=1 ------- it operates MTMn-Chc.
fMTncPWMO=1 --- it generates PWMnc signal.
fMTncCLST=2----- it clears MTMn-Chc counter with SNTncBCc event.
fMTncBFNC=0-3 -- it sets the sentinel B block to be valid.
fMTncAOL ---- refer to *1

Figure 12-19 Example of PWMnc signal generation with duty = 0% in the standard mode

12-4-7 Multiple timers interrupt

MTMsn supports the interrupt function according to the operation state. MTMsn consists of 2 channels. Interrupt factors are prepared for each channel.

12-4-7-1 Multiple timer interrupt cause generation

- Generation of MTMn-Chc counter overflow event

When MTMn-Chc counter overflows, MTMn-Chc counter overflow event is issued. This event sets fMTncOVF of MTMn-Chc status register to “1” and that is communicated to the interrupt control as MTMn-Chc interrupt factor.

Even when MTMn-Chc counter overflow event causes MTMn-Chc interrupt, fMTncOVF is not automatically cleared. Program clears fMTncOVF.

- Generation of RDncB event and SNTncBCc event

When, in MTMn-Chc, the counter read function generates RDncB event or the counter sentinel function generates SNTncBCc event, fMTncBEVT of MTMn-Chc status register becomes “1”. fMTncBEVT is communicated to the interrupt control as MTMn-Chc interrupt factor.

Even when RDncB event or SNTncBCc event causes MTMn-Chc interrupt, fMTncBEVT is not automatically cleared. Program clears fMTncBEVT.

- Generation of RDncA event and SNTncACc event

When, in MTMn-Chc, the counter read function generates RDncA event or the counter sentinel function generates SNTncACc event, fMTncAEVT of MTMn-Chc status register becomes “1”. fMTncAEVT is communicated to the interrupt control as MTMn-Chc interrupt factor.

Even when RDncA event or SNTncACc event causes MTMn-Chc interrupt, fMTncAEVT is not automatically cleared. Program clears fMTncAEVT.

Note) c: 1-0

12-4-8 Multiple timers standby function

MTMsn supports software standby and hardware standby by the power save function.

When the power save function generates software standby or hardware standby to MTMsn, it immediately transitions to the standby state.

The power save function also generates a request to cancel the standby state. After cancelling standby, initialize MTMsn setting by program.

12-4-9 Precautions for use

The external input clock pulse length needs 1.5 or more system clocks for single edge of rising or falling and 2.5 or more system clocks for both edges.

When a pulse length is shorter than the above, counting may not be correctly performed.

13 RTC

13-1 Introduction

This chapter provides the outline of RTC in the section below.

- The outline of RTC

13-1-1 The outline of RTC

RTC offers a clock with hours, minutes, and seconds as well as a calendar with years, months, dates and days of a week. RTC can also control an alarm.

13-1-1-1 Features of RTC

- Clock function with alarm
- A calendar that can be used until the year of 9999
- Summer time can be set.
- Automatic correction of leap years and different month lengths
- 3 types of interrupt outputs
  - The alarm interrupt
  - The periodic interrupt
  - The update completion interrupt

Attention: VDET pin needs to be controlled according to RTC operating power voltage state.

Normally, when the operating voltage is lower than the operation ensuring voltage, set VDET = Low and reset RTC.

(See the datasheet for the reset conditions.)

NRES pin does not reset RTC.

13-2 Functional Overview

This chapter provides the outline of RTC function block in the section below.

- RTC circuit configuration
- RTC Sequence

13-2-1 RTC circuit configuration

The RTC circuit configuration is as shown in

Figure 13-1 and

Figure 13-2 In TQFP version, RTCMODE is internally fixed.

13-2-1-1 General RTC mode (when RTCMODE = 1)

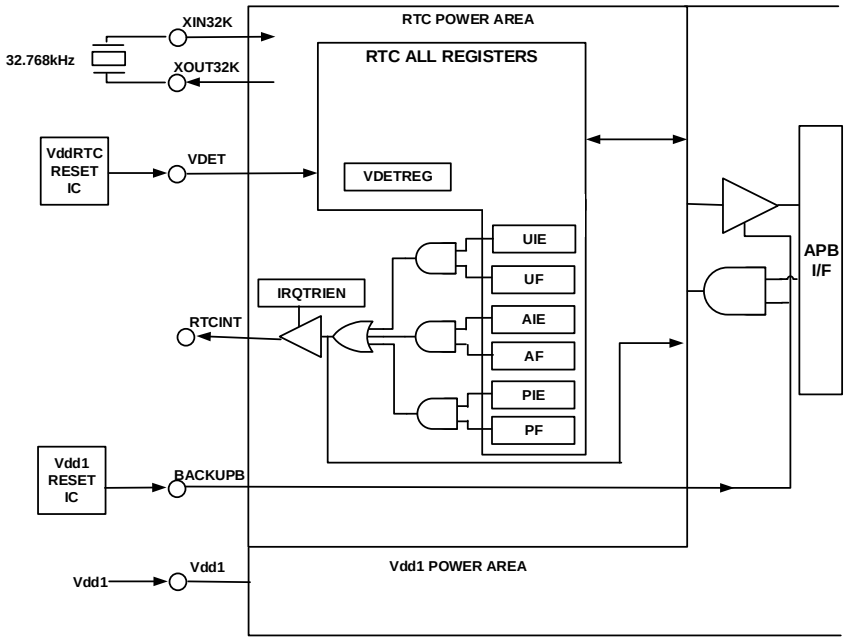


Figure 13-1 Real time clock configuration

13-2-1-2 Key Int RTC mode (when RTCMODE = 0)

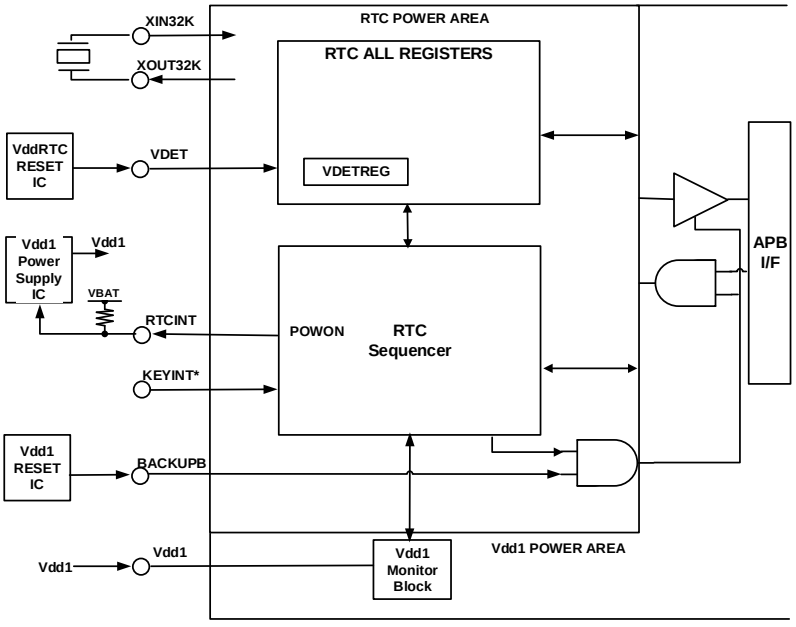


Figure 13-2 Real time clock configuration

13-2-2 RTC Sequence

The following explains RTC power supply and pin control sequence. In TQFP version, RTCMODE is internally fixed.

13-2-2-1 General RTC mode (when RTCMODE = 1)

To operate only RTC when the device is powered off, it is necessary to detect voltage decreases of Vdd1 and Vdd2, set BACKUPB = Low, and control signals (ISOLATOR control) between VddRTC domain and Vdd1 domain.

When RTC power, VddRTC is turned off (RTC operation stops), it is necessary to detect VddRTC voltage decrease and set VDET = Low.

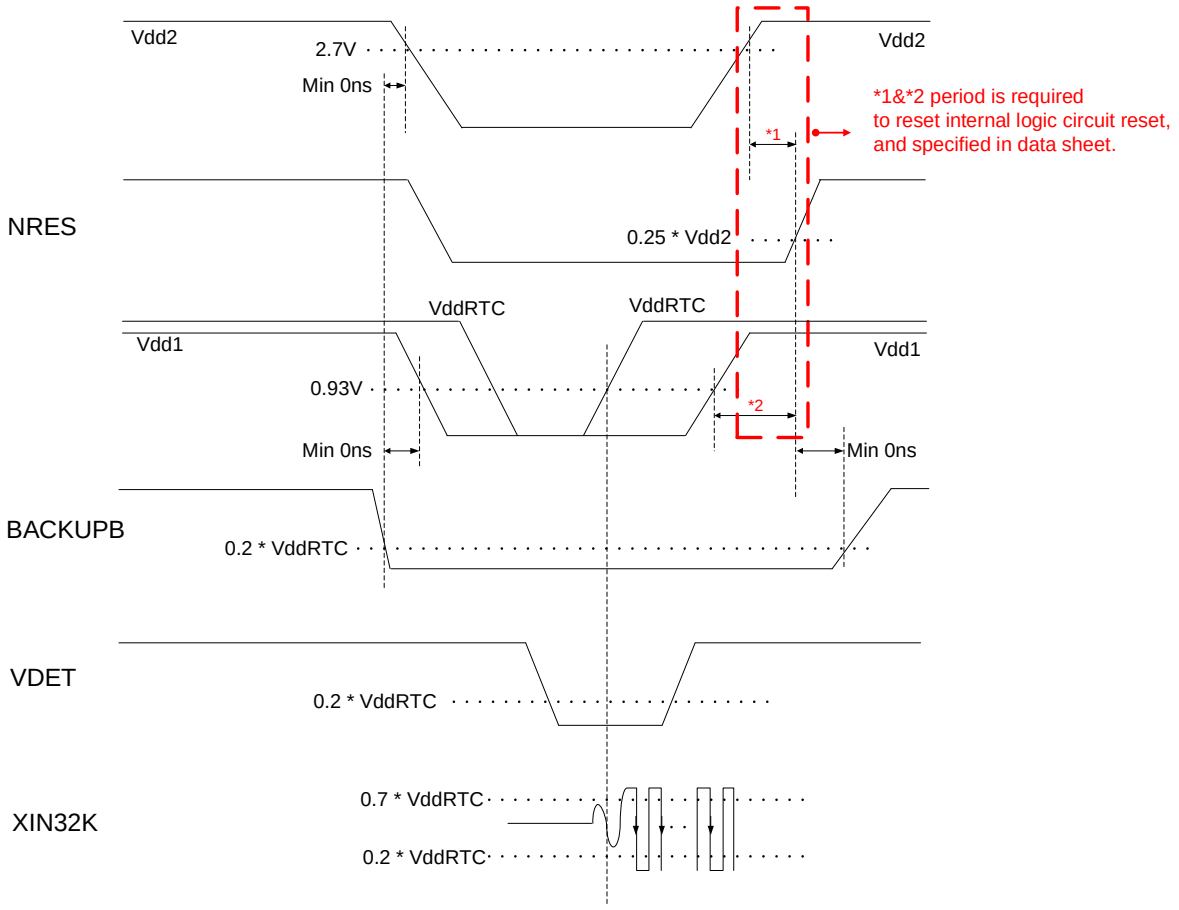


Figure 13-3

13-2-2-2 Key Int RTC mode (when RTCMODE = 0)

The sequencer in RTC receives master commands from Cortex-M3 and controls BACKUPB (ISOLATOR). KEYINT pin input and internal RTCINT perform returning. BACKUPB pin can perform the external control and cope with unexpected power-off. When RTC power, VddRTC is powered off (RTC operation stops), it is necessary to detect VddRTC voltage decrease and set VDET = Low.

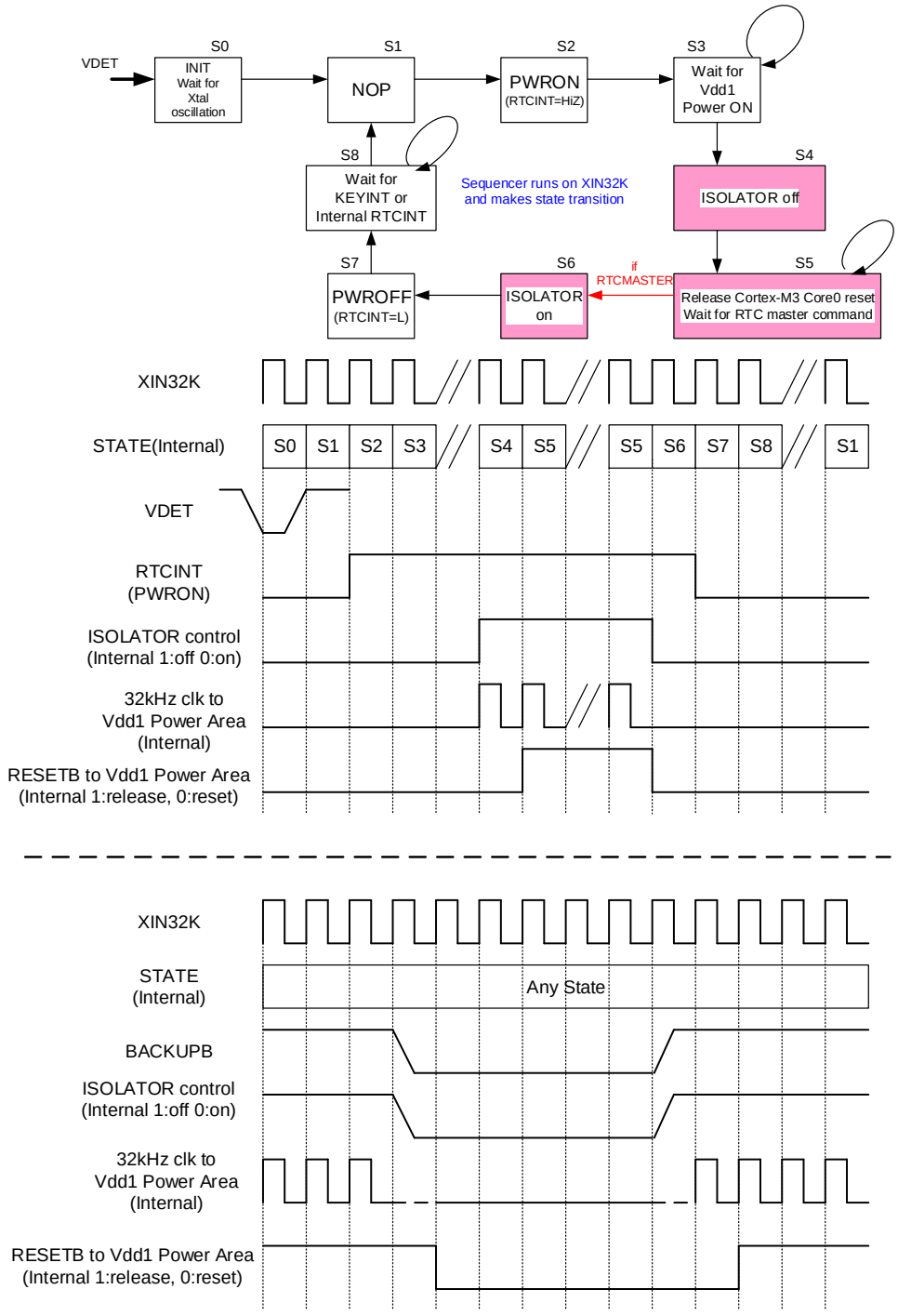


Figure 13-4

13-3 Programmer’s Model

This chapter provides RTC registers and setting via microcontroller in the sections below.

- About the programmer’s model
- Summary of RTC registers
- Register Descriptions

13-3-1 About the programmer’s model

RTC base address is not fixed, which depends on system implementation. See the system specification for base address.

Each register's offset is fixed.

13-3-2 Summary of RTC registers

RTC is controlled by the registers in “Table 13-1”.

Remark: Do not access any unspecified address from Base+0x0000 to Base+0xFFFF.

Table 13-1 RTC registers

Offset	Abbreviation	Attribute	Register name	Initial value
0x00	SEC	R/W	RTC second register	0x0000_0000
0x04	MIN	R/W	RTC minute register	0x0000_0000
0x08	HOUR	R/W	RTC hour register	0x0000_0000
0x0C	WEEK	R/W	RTC day of week register	0x0000_0000
0x10	DAY	R/W	RTC day register	0x0000_0000
0x14	MONTH	R/W	RTC month register	0x0000_0000
0x18	YEAR	R/W	RTC year (lower 2 digits) register	0x0000_0000
0x1C	CENT	R/W	RTC year (upper 2 digits) register	0x0000_0000
0x20	SECALM	R/W	RTC second alarm register	0x0000_0000
0x24	MINALM	R/W	RTC minute alarm register	0x0000_0000
0x28	HOURALM	R/W	RTC hour alarm register	0x0000_0000
0x2C	WEEKALM	R/W	RTC day of week alarm register	0x0000_0000
0x30	DAYALM	R/W	RTC day alarm register	0x0000_0000
0x34	MONTHALM	R/W	RTC month alarm register	0x0000_0000
0x38	PRSEL	R/W	RTC periodic interrupt register	0x0000_0000
0x3C	RTCINT	R/W	RTC interrupt control register	0x0000_0081
0x40	DST	R/W	RTC Day light saving time control register	0x0000_0000
0x44	RTCSTAT	R/W	RTC status register	0x0000_0080
0x4C	PTN32BIT0	R/W	RTC 32-bit pattern register 0	0x0000_00XX
0x50	PTN32BIT1	R/W	RTC 32-bit pattern register 1	0x0000_00XX
0x54	PTN32BIT2	R/W	RTC 32-bit pattern register 2	0x0000_00XX
0x58	PTN32BIT3	R/W	RTC 32-bit pattern register 3	0x0000_00XX
0x5C	VPTN32BIT0	R/W	RTC VDET 32-bit pattern register 0	0x0000_00XX
0x60	VPTN32BIT1	R/W	RTC VDET 32-bit pattern register 1	0x0000_00XX
0x64	VPTN32BIT2	R/W	RTC VDET 32-bit pattern register 2	0x0000_00XX
0x68	VPTN32BIT3	R/W	RTC VDET 32-bit pattern register 3	0x0000_00XX
0x6C	VDET	R/W	RTC voltage decrease detection register	0x0000_000X
0x70	RTCINTCNT	R/W	RTCINT control register	0x0000_0000
0x90	RTCMSTR	W	RTC master register	0x0000_0000
0x94	KEYPOL1	R/W	KEY interrupt polarity register 1	0x0000_0000
0x9C	KEYINTEN1	R/W	KEY interrupt enable register 1	0x0000_0000
0xA4	KEYINTMON1	R/W	KEY interrupt monitor register 1	0x0000_0000
0xAC	KEYMON1	R	KEY state monitor register 1	0x0000_00XX
0xB4	STATE	R	Sequencer state monitor register	0x0000_0000
0xD0	KEYINTPD	R/W	KEY pin pull-down control register	0x0000_0007
0xE0	XIN32KSTS	R	Oscillation stability wait status register	0x0000_0000

Note: Bit 7 of RTCSTAT can be only written. Therefore, “0” is always read.

13-3-3 Register Descriptions

13-3-3-1 RTC second register: SEC

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit5-0: SEC5-0
Sets seconds of clock in binary format.

Remark: (Example) 59 seconds → Binary 0x3B

13-3-3-2 RTC minute register: MIN

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit5-0: MIN5-0
Sets minutes of clock in binary format.

Remark: (Example) 45 minutes → Binary 0x2D

13-3-3-3 RTC hour register: HOUR

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	HOUR4	HOUR3	HOUR2	HOUR1	HOUR0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit4-0: HOUR4-0
Sets hours of clock in binary format.

Remark: (Example) 11 p.m. → Binary 0x17

13-3-3-4 RTC day of week register: WEEK

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	WEEK2	WEEK1	WEEK0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2-0: WEEK2-0
Sets a day of week of clock.

Remark: (Example) Sunday is 1. Saturday is 7.

13-3-3-5 RTC day register: DAY

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	DAY4	DAY3	DAY2	DAY1	DAY0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit4-0: DAY4-0
Sets a date of clock in binary format.

Remark: (Example) Date 31 → Binary 0x1F

13-3-3-6 RTC month register: MONTH

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	MONTH3	MONTH2	MONTH1	MONTH0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit3-0: MONTH3-0
Sets a month of clock in binary format.

Remark: (Example) December → Binary 0x0C

13-3-3-7 RTC year (lower 2 digits) register: YEAR

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	YEAR6	YEAR5	YEAR4	YEAR3	YEAR2	YEAR1	YEAR0
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit6-0: YEAR6-0
Sets the lower 2 digits of a year of clock in binary format.

Remark: (Example) Year 2123 → Binary 0x17

13-3-3-8 RTC year (upper 2 digits) register: CENT

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	CENT6	CENT5	CENT4	CENT3	CENT2	CENT1	CENT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit6-0: CENT6-0
Sets the upper 2 digits of a year of clock in binary format.

Remark: (Example) Year 2123 → Binary 0x15

13-3-3-9 RTC second alarm register: SECALM

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	SECALM5	SECALM4	SECALM3	SECALM2	SECALM1	SECALM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit5-0: SECALM5-0
Sets seconds of alarm time in binary format.

Remark: When seconds of time is not designated, set 0x3F.

13-3-3-10 RTC minute alarm register: MINALM

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	MINALM5	MINALM4	MINALM3	MINALM2	MINALM1	MINALM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit5-0: MINALM5-0
Sets minutes of alarm time in binary format.

Remark: When minutes of time is not designated, set 0x3F.

13-3-3-11 RTC hour alarm register: HOURALM

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	HOUR ALM4	HOUR ALM3	HOUR ALM2	HOUR ALM1	HOUR ALM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit4-0: HOURALM4-0
Sets hours of alarm time in binary format.

Remark: When hours of time is not designated, set 0x1F.

13-3-3-12 RTC day of week alarm register: WEEKALM

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	WEEK ALM2	WEEK ALM1	WEEK ALM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2-0: WEEKALM2-0
Sets a day of week of alarm time in binary format.

Remark: When a day of week is not designated, set 0x00.

13-3-3-13 RTC day alarm register: DAYALM

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	DAYALM5	DAYALM4	DAYALM3	DAYALM2	DAYALM1	DAYALM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit5-0: DAYALM5-0
Sets a date of alarm time in binary format.

Remark: When a date is not designated, set 0x3F.

13-3-3-14 RTC month alarm register: MONTHALM

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	MONTH ALM3	MONTH ALM2	MONTH ALM1	MONTH ALM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit3-0: MONTHALM3-0
Sets a month of alarm time in binary format.

Remark: When a month is not designated, set 0x0F.

13-3-3-15 RTC periodic interrupt register: PRSEL

[Base + 0x0038]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	PRSEL3	PRSEL2	PRSEL1	PRSEL0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit7-4: PRSEL3-0
Sets a period of periodic interrupt.
0: 61.034 us
1: 244.141 us
2: 976.562 us
3: 3.9063 ms
4: 15.625 ms
5: 62.5 ms
6: 250 ms
7: 500 ms
8: 1 sec
9: 1 min

* Do not set 10 to 15.
The periodic period shall be changed when periodic interrupt factor is cleared (PE = 1).
The first interrupt after the period is changed may be earlier than the expected period.

13-3-3-16 RTC interrupt control register: RTCINT

[Base + 0x003C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	SET	PIE	AIE	UIE	-	-	-	Reserved
Reset	1	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Bit7: SET
Stops update cycle. This is used to set data in the time and calendar area.
0: Update cycles 1 time every 1 second.
1: Stops update cycle.

Bit6: PIE
Enables periodic interrupt.
0: Disables periodic interrupt.
1: Enables periodic interrupt according to the period set by PRSEL register.

Bit5: AIE
Enables alarm interrupt output.
0: Disables alarm interrupt.
1: Enables alarm interrupt when alarm conditions are met.

Bit4: UIE
Enables update-cycle-ended-interrupt output.
0: Disables update-cycle-ended interrupt.
1: Enables update-cycle-ended interrupt.

Bit0: Reserved
Fixed at the initial value “1”.

13-3-3-17 RTC Day light saving time control register: DST

[Base + 0x0040]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DSTE	SOUTH	-	EFE	EMON	SFE	SMON1	SMON0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit7: DSTE
Sets DST.
0: Normal.
1: It is updated from AM 1:59:59 to AM 3:00:00 on Sunday (the first or the last) set by SFE in a month set by SMON. It is updated from AM 1:59:59 to AM 1:00:00 on Sunday (the first or the last) set by EFE in a month set by EMON.

Bit6: SOUTH
Reverses the designated dates of DST start and end.
0: EFE and EMON designate the end date, and SFE and SMON designate the start date.
1: EFE and EMON designate the start date, and SFE and SMON designate the end date.

Bit4: EFE
Designates DST end date.
0: The first Sunday
1: The last Sunday

Bit3: EMON
Designates DST end month.
0: October
1: November

Bit2: SFE
Designates DST start date.
0: The first Sunday
1: The last Sunday

Bit1-0: SMON1-0

Designates DST (Day light Saving Time) start month.

00: March

01: April

10: May

11: Setting prohibited.

13-3-3-18 RTC status register: RTCSTAT

[Base + 0x0044]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	PCLR	PF	AF	UF	-	-	-	UIP
Reset	1	0	0	0	0	0	0	0
R/W	W	R/W	R/W	R/W	R	R	R	R

Bit7: PCLR
Clears interrupt. Writing “0” clears flags of PF, AF and UF.
To enable flags, write “1”.

Attention: The initial value is “1”, but “1” cannot be read because this register is write-only. So, “0” is always read out.

Bit6: PF
At read
Periodical interrupt factor
0: Normal
1: PESEL-designated period passes.

At write
Clears periodical interrupt factor.
0: Does not clear interrupt.
1: Clears interrupt.

Bit5: AF
At read
Alarm interrupt factor
0: Normal
1: Alarm conditions are met.

At write
Clears alarm interrupt factor.
0: Does not clear interrupt.
1: Clears interrupt.

Bit4: UF
At read
Update-cycle-ended-interrupt factor
0: Normal
1: Update cycle end

At write
Clears update-cycle-ended-interrupt factor.
0: Does not clear interrupt.
1: Clears interrupt.

Bit0: UIP
Update cycle status
0: Update cycle does not start for at least 30 us.
1: Update cycle is in progress.

* When Bit0 is written, “0” shall be written.

13-3-3-19 RTC 32-bit pattern register 0: PTN32BIT0

[Base + 0x004C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	PTN7	PTN6	PTN5	PTN4	PTN3	PTN2	PTN1	PTN0
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: PTN7-0
32-bit pattern register

13-3-3-20 RTC 32-bit pattern register 1: PTN32BIT1

[Base + 0x0050]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	PTN15	PTN14	PTN13	PTN12	PTN11	PTN10	PTN9	PTN8
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: PTN15-8
32-bit pattern register

13-3-3-21 RTC 32-bit pattern register 2: PTN32BIT2

[Base + 0x0054]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	PTN23	PTN22	PTN21	PTN20	PTN19	PTN18	PTN17	PTN16
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: PTN23-16
32-bit pattern register

13-3-3-22 RTC 32-bit pattern register 3: PTN32BIT3

[Base + 0x0058]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	PTN31	PTN30	PTN29	PTN28	PTN27	PTN26	PTN25	PTN24
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: PTN31-24
32-bit pattern register

13-3-3-23 RTC VDET 32-bit pattern register 0: VPTN32BIT0

[Base + 0x005C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	VPTN7	VPTN6	VPTN5	VPTN4	VPTN3	VPTN2	VPTN1	VPTN0
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: VPTN7-0
32-bit pattern register to control VDET

13-3-3-24 RTC VDET 32-bit pattern register 1: VPTN32BIT1

[Base + 0x0060]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	VPTN15	VPTN14	VPTN13	VPTN12	VPTN11	VPTN10	VPTN9	VPTN8
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: VPTN15-8
32-bit pattern register to control VDET

13-3-3-25 RTC VDET 32-bit pattern register 2: VPTN32BIT2

[Base + 0x0064]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	VPTN23	VPTN22	VPTN21	VPTN20	VPTN19	VPTN18	VPTN17	VPTN16
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: VPTN23-16
32-bit pattern register to control VDET

13-3-3-26 RTC VDET 32-bit pattern register 3: VPTN32BIT3

[Base + 0x0068]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	VPTN31	VPTN30	VPTN29	VPTN28	VPTN27	VPTN26	VPTN25	VPTN24
Reset	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0: VPTN31-24
32-bit pattern register to control VDET

13-3-3-27 RTC voltage decrease detection register: VDET

[Base + 0x006C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	VDET
Reset	0	0	0	0	0	0	0	X
R/W	R	R	R	R	R	R	R	R/W

Bit0: VDET

At read

Detects and latches VDET pin input falling. Low active.

Remark: Detects even the initial reset to be low.

At write

0: Monitors VDET pin input falling.

1: Resets this register to return to high (VDET not detected).

Attention: After writing “1”, write “0” again to start monitoring.

The register value change is detected by RTC XTAL clock of 32.768 KHz. Therefore, after reset, at least 32.768 KHz*2 shall pass before setting “0”.

13-3-3-28 RTCINT control register: RTCINTCNT

[Base + 0x0070]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	IRQTRIEN2	IRQTRIEN1	IRQTRIEN0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2: IRQTRIEN2 (Effective only in Key Int RTC mode)
0: Outputs an interrupt to RTC sequencer.
RTCINT, which is a condition for S8 → S1 in the state transition diagram in 13-2-2-2 Key Int RTC mode (when RTCMODE = 0), is output to RTC sequencer.
1: Does not output an interrupt to RTC sequencer.
RTCINT, which is a condition for S8 → S1 in the state transition diagram in 13-2-2-2 Key Int RTC mode (when RTCMODE = 0), is not output to RTC sequencer.

Bit1: IRQTRIEN1
0: Outputs RTCINT interrupt to Cortex-M3 Core0, Core1.
1: Does not output RTCINT interrupt to Cortex-M3 Core0, Core1.
* The actual interrupt needs the interrupt setting of Cotex-M3 Core0, Core1.

Bit0: IRQTRIEN0
0: Outputs an interrupt to RTCINT pin.
Low is output when an interrupt occurs. At other times, Hiz is output.
1: RTCINT pin is Hiz output.

13-3-3-29 RTC master register: RTCMSTR

[Base + 0x0090]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	RTC_ MSTR
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Bit0: RTC_MSTR
1: Powers off the device except RTC.

Remark: This register is used only to power off the device except RTC.

Writing “1” in this bit makes RTC turn off the isolator that separates power source.

Key interrupt or timer interrupt turns on the isolator.

When read, “0” can be read.

Attention: Writing “1” turns off RTC isolator. After that, RTC cannot be accessed until RTC generates an interrupt (key interrupt or timer interrupt).

13-3-3-30 KEY interrupt polarity register 1: KEYPOL1

[Base + 0x0094]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	POL1_2	POL1_1	POL1_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2-0: POL1_2-0

Selects whether to detect rising or falling of KEYINT0-2 interrupt.

0: Detects a falling edge.

1: Detects a rising edge.

13-3-3-31 KEY interrupt enable register 1:KEYINTEN1

[Base + 0x009C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	EN1_2	EN1_1	EN1_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2-0: EN1_2-0
Enables KEYINT0-2 interrupt.
0: Disables interrupt.
1: Enables interrupt.

13-3-3-32 KEY interrupt monitor register 1: KEYINTMON1

[Base + 0x00A4]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	IMON1_2	IMON1_1	IMON1_0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2-0: IMON1_2-0
Monitors KEYINT0-2 interrupt state.
At read
0: No interrupt factor occurs.
1: Interrupt factor occurs.

At write
0: Unnecessary to write.
1: Writing clears.

13-3-3-33 KEY state monitor register 1: KEYMON1

[Base + 0x00AC]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	MON1_2	MON1_1	MON1_0
Reset	0	0	0	0	0	X	X	X
R/W	R	R	R	R	R	R	R	R

Bit2-0: MON1_2-0
Monitors KEYINT0-2 state.
Can read the values input in KEYINT0-2 directly.

13-3-3-34 Sequencer state monitor register: STATE

[Base + 0x00B4]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	STATE3	STATE2	STATE1	STATE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit3-0: STATE3_0
Monitors the sequencer state.
Can read the state value specified in the hardware manual directly.

13-3-3-35 KEY pin pull-down control register: KEYINTPD

[Base + 0x00D0]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	PD2	PD1	PD0
Reset	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit2-0: PD2-0
Controls the pull-down of KEYINT2-0 pin.
0: Pull-down OFF
1: Pull-down ON

13-3-3-36 Oscillation stability wait status register: XIN32KSTS

[Base + 0x00E0]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	RTCMODE	XIN32KSTS
Reset	0	0	0	0	0	0	x	0
R/W	R	R	R	R	R	R	R	R

Bit1: RTCMODE
Indicates RTC mode.
0: Key Int RTC mode
1: General RTC mode

Bit0: XIN32KSTS
Indicates the oscillation state of RTC oscillation amp.
0: Oscillation amp stability wait is not completed.
1: Oscillation amp stability wait is completed.

13-4 Operation

This section provides RTC setting flow.

13-4-1 Explanation about operations

13-4-1-1 Time, calendar and alarm

To set time, calendar and alarm registers, stop the update cycle by setting SET bit to “1” while UIP flag is “0”. When UIP flag is “0”, the update cycles does not start for at least 30 μs. Therefore, check that UIP flag is 0, and within 30 μs, set SET bit to “1” to stop the update cycle.

Next, set the values in time, calendar and alarm registers in binary format and then set SET bit to “0” to start the update. To acquire the elapsed time, make sure that an update-ended interrupt has occurred after update start. Therefore, after setting, it takes 1 or more seconds at most to acquire the elapsed time. The following table shows data ranges of time, calendar and alarm registers.

Table 13-2 Data ranges of registers

Register	Data range (decimal)	Data range (hexadecimal)
Second, Second alarm	0-59	00-3B
Minute, Minute alarm	0-59	00-3B
Hour, Hour alarm	0-23	00-17
Day of week, Day of week alarm	1-7	01-07
Date, Date alarm	1-31	01-1F
Month, Month alarm	1-12	01-0C
Year (lower 2 digits), Year alarm	0-99	00-63
Year (upper 2 digits), Year alarm	19-20	13-14

13-4-1-2 Update cycle

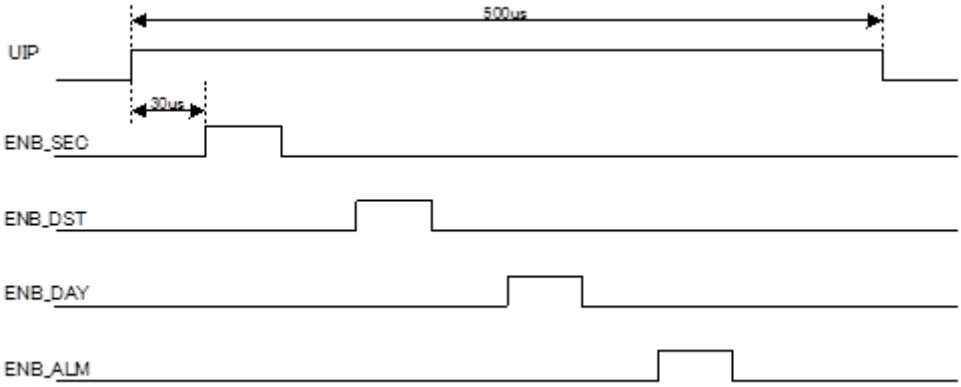


Figure 13-5 Update cycle timing of UIP and process blocks

SET bit “0” performs update cycle 1 time every 1 second. The figure above shows the relationship between UIP (update cycle start signal) and the update cycle timing signals of process blocks (ENB_XXX).

The update cycle (UIP = 1) starts and asserts signals in the following order to perform each process: a signal enabling clock second update operation (ENB_SEC), a signal enabling DST control operation (ENB_DST), a signal enabling calendar date update operation (ENB_DAY) and a signal enabling alarm control operation (ENB_ALM).

SET bit “1” stops the update cycle. The update cycle takes approximately 500 μs. Not during the update cycle, the processor can access time, calendar and alarm registers. There are two ways to avoid accesses during the update cycle.

- Using UIP bit: The update cycle starts 30 μs after UIP bit becomes “1”. Therefore, when UIP bit is “0”, access can be performed for at least 30 μs.
- Using update-ended interrupt: After interrupt occurrence, the update cycle does not start for approximately 999.5 ms. During the time, access can be performed.

13-4-1-3 Alarm

An alarm interrupt occurs when a time or a time and date set by the alarm registers (seconds, minutes, hours, day of week, date and month) matches a time or a time and date indicated by time and calendar registers (seconds, minutes, hours, day of week, date and month). The following are some examples of alarm setting in binary mode.

- Interrupt every 1 second

MONTHALM = 0xFF,DAYALM = 0xFF,WEEKALM = 0x00
HOURALM = 0xFF,MINALM = 0xFF,SECALM = 0xFF

- Interrupt at 17 seconds every minute

MONTHALM = 0xFF,DAYALM = 0xFF,WEEKALM = 0x00
HOURALM = 0xFF,MINALM = 0xFF,SECALM = 0x11

- Interrupt at 32 minutes 45 seconds every hour

MONTHALM = 0xFF,DAYALM = 0xFF,WEEKALM = 0x00
HOURALM = 0xFF,MINALM = 0x20,SECALM = 0x2D

- Interrupt at 19 hours, 29 minutes 30 seconds every day

MONTHALM = 0xFF,DAYALM = 0xFF,WEEKALM = 0x00
HOURALM = 0x13,MINALM = 0x1D,SECALM = 0x1E

- Interrupt at 20 hours 14 minutes 52 seconds every Tuesday

MONTHALM = 0xFF,DAYALM = 0xFF,WEEKALM = 0x03
HOURALM = 0x14,MINALM = 0x0E,SECALM = 0x34

- Interrupt at 10 hours 03 minutes 28 seconds on date 25 every month

MONTHALM = 0xFF,DAYALM = 0x19,WEEKALM = 0x00
HOURALM = 0x0A,MINALM = 0x03,SECALM = 0x1C

- Interrupt at 10 hours 03 minutes 28 seconds on June 18, Saturday

MONTHALM = 0x06,DAYALM = 0x12,WEEKALM = 0x07
HOURALM = 0x0A,MINALM = 0x03,SECALM = 0x1C

13-4-1-4 Interrupt

Three types of independent interrupt requests can be generated. An alarm interrupt occurs when a time and date matches the setting of alarm registers. A periodic interrupt occurs in a period from μs to s. The update-ended interrupt occurs at the end of an update cycle.

Interrupts are enabled by setting “1” in the interrupt enable bits: PIE, AIE and UIE. Interrupt occurrence sets any of PF, AF and UF bits to “1”. These bits are set independently of the interrupt enable bits.

13-4-1-5 DST function

DST (Day light Saving Time) function automatically changes setting into summer time where the clock is advanced by 1 hour only during summer, when the sun rises early. The days when DST is used vary according to countries and regions. Seasons in the northern hemisphere are opposite to those in the southern hemisphere. The real time clock has a function to cope with these. SMON register can designate DST start month and SFE bit can designate DST start day of week. EMON bit and EFE bit can set DST end month and day of week. SOUTH bit can reverse the designated dates of DST start and end to support the countries and regions in the southern hemisphere.

When setting DSTE bit to “1” turns on DST function:
On the designated day when DST begins, it operates as shown below.
→ → → AM 1 hour 59 minutes 59 seconds → AM 3 hours 00 minutes 00 seconds → → →
After that, it operates normally until the designated day of DST end.

On the designated day when DST ends, it operates as shown below.
→ → → AM 1 hour 59 minutes 59 seconds → AM 1 hour 00 minutes 00 seconds → → →
After that, it operates normally.

Note: The alarm does not work when it is set from AM 2 hours 00 minutes 00 seconds to AM 3 hours 00 minutes 00 seconds on the day when DST begins. By contrast, the alarm works twice when it is set from AM 1 hour 00 minutes 00 seconds to AM 1 hour 59 minutes 59 seconds on the day when DST ends.
If this function does not support the actual regional summer time, control the setting by software.

13-4-1-6 Leap year automatic correction

This real time clock automatically corrects leap years. The following is the rules of judging whether it is a leap year or not.

- A year that is a multiple of 4 is a leap year.
- A year that is a common multiple of 4 and 100 is not a leap year.
- A year that is a common multiple of 100 and 400 is a leap year.

13-4-1-7 Backup function

The real time clock circuit block has a different power source from that of other circuits. Therefore, after asserting BACKUPB signal, other circuits can be powered off to save LSI consumption power. In Key Int RTC mode, the sequencer in RTC can control the power source of other circuits.

13-4-1-8 RTC operating power-off judgement

RTC 32-bit pattern registers: PTN32BIT3–0 judge whether RTC operating power is turned off.

Set a certain value (There is no recommended value.) and compare the setting value with the value at a later time when needed so that power supply state can be judged as below.
From the time of setting to the time of comparing,

- the comparison value is matched. … RTC operating power has continued being normally supplied.
- the comparison value is not matched. … RTC operating power has been turned off before.

13-4-1-9 RTC VDET pin detection method control

VDET pin detects low level to reset RTC circuit. At power-on, the detection is asynchronous. However, after VDET pin = high level, setting values of RTC VDET 32-bit pattern registers: VPTN32BIT3–0 as shown below works the chattering prevention circuit of 32.768 KHz × 8 period. (It is not reset unless VDET = Low of 32.768 KHz × 8 is detected.) After reset, VPTN32BIT3–0 all become the initial values: 0x00 and the detection becomes asynchronous.

Initial values: All are 0x00.
(*1) 0x5C: 0xA7
(*2) 0x60: 0x58
(*3) 0x64: 0x3C
(*4) 0x68: 0xF1
(*1) to (*4) are in no particular order.

* This function is disabled in Key Int RTC mode.

13-4-2 Control flow

13-4-2-1 Clock and calendar

The figure shows the control flow to set initial values of the clock and calendar and to read out the update data every second interrupt. It is assumed that the system displays the read-out data on a device outside LSI. First, set UIE bit to “1” to enable an update-ended interrupt request signal of time and calendar data. Meanwhile, the control shifts to an interrupt routine when every-second update ends. To adjust time or date, set SET bit to “1” to stop update cycle and set desired values in registers. Then, set SET bit to “0” to restart update cycle and to end the interrupt process. To display the time, read out the update value from each register and transfer the data to the display to end the interrupt process.

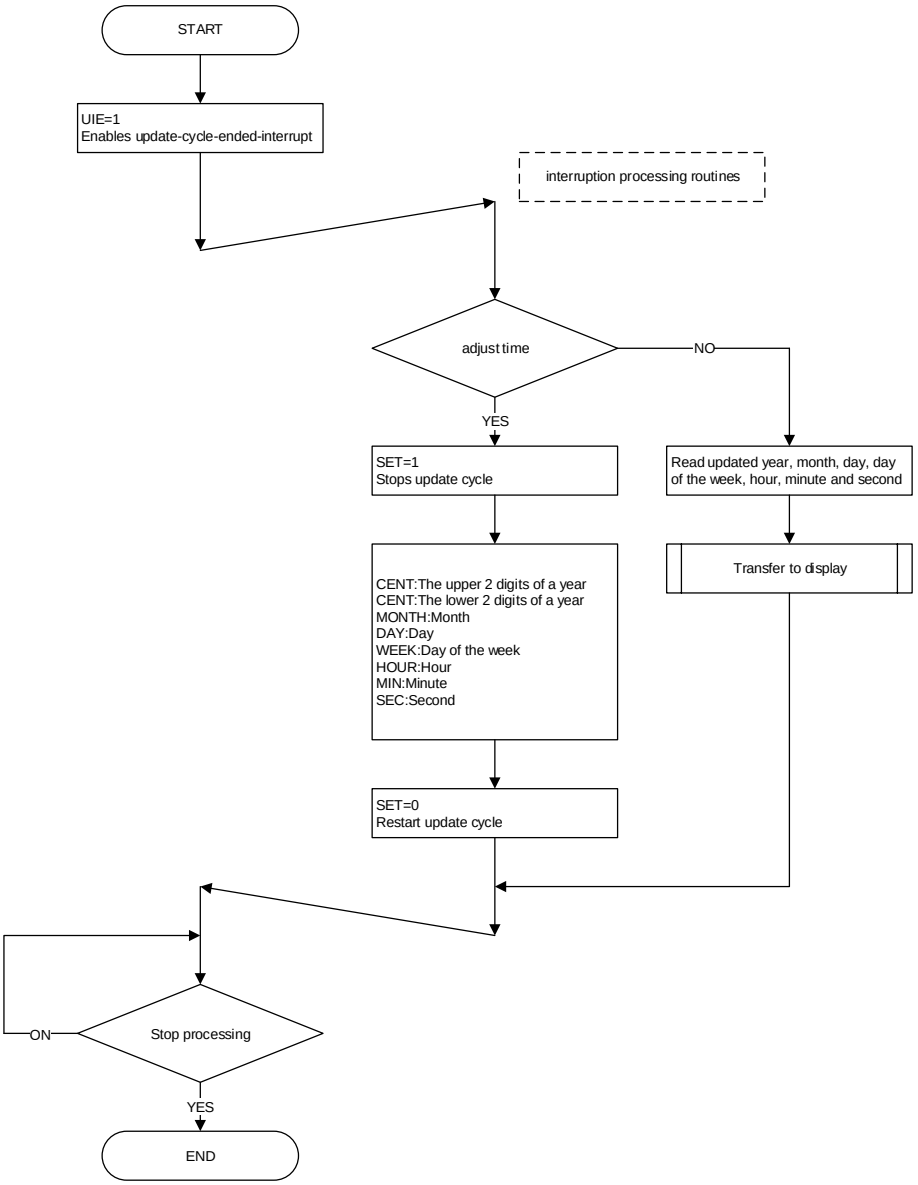


Figure 13-6 Control flow

13-4-2-2 DST automatic correction

The following is the flow to perform DST automatic correction.

Designate the northern hemisphere or the southern hemisphere with SOUTH bit.

Set DST start day by combining SMON register and SFE bit.
(When SOUTH bit is set to “1”, it becomes DST end day.)

Set DST end day by combining EMON register and EFE bit.
(When SOUTH bit is set to “1”, it becomes DST start day.)

Set DSTE bit to “1” to enable DST automatic correction.

13-4-3 Precautions for use

VDET pin needs to be controlled according to RTC operating power voltage state. Normally, when the operating voltage is lower than the operation ensuring voltage, set VDET = Low and reset RTC.
(See the datasheet for the reset conditions.)
NRES pin does not reset RTC.

14 SPI

14-1 Introduction

This chapter provides the brief summary of SPI in the sections below.

- Brief Summary of SPI

14-1-1 Brief Summary of SPI

This document is the explanation about SPI.

SPI is one - ccurre clock synchronous serial interface of full duplex transfer with four communication lines.

FIFO with 16 depths is available for transmission and reception (16 Byte during 8 bit transfer, 32 Byte during 16 bit transfer).

14-1-1-1 Feature of SPI

- CS selection

Transfer function by using chip select terminal is possible.

- Clock selection

Transfer clock can be set to the internal clock(It is possible to change clock rate) or the external clock.

- Clock polarity selection

The polarity of SCK can be changed to High or Low during non-communicating.

- Data length selection

The data length can be set to 8 bits or 16 bits.

- Bit length selection

The bit length can be set to one frame (8 bits/16 bits) transfer or continuous transfer.

- LSB/MSB first bit selection

The first bit of the data can be set to LSB first or MSB first.

- Error detection

The error detection is possible.

FIFO.disable Overrun error, receive register overwrite error

FIFO enable Receive FIFO underrun error and receive FIFO overwrite error

Transmit FIFO underrun error and transmit FIFO overwrite error

- DMA request

The DMA request of transmission and reception is available.

- FIFO use selection

The transmission and reception by using FIFO of 16 depths is available.

Table 14-1 SPI terminal table

Pin Name	Direction	Description
SDI	I	serial data input SIO input of receive data.
SDO	O	Cereal data output SIO output of transmit data.
SCK	I/O	Serial clock SIO input and output of clock.
CSN	I/O	Chip selection SIO input and output of chip selection.

Attention) The chip select function is not available in this model.

14-2 Functional Overview

This chapter describes the outline of the functional blocks of SPI.

The following section is included.

- SPI circuit configuration

14-2-1 SPI circuit configuration

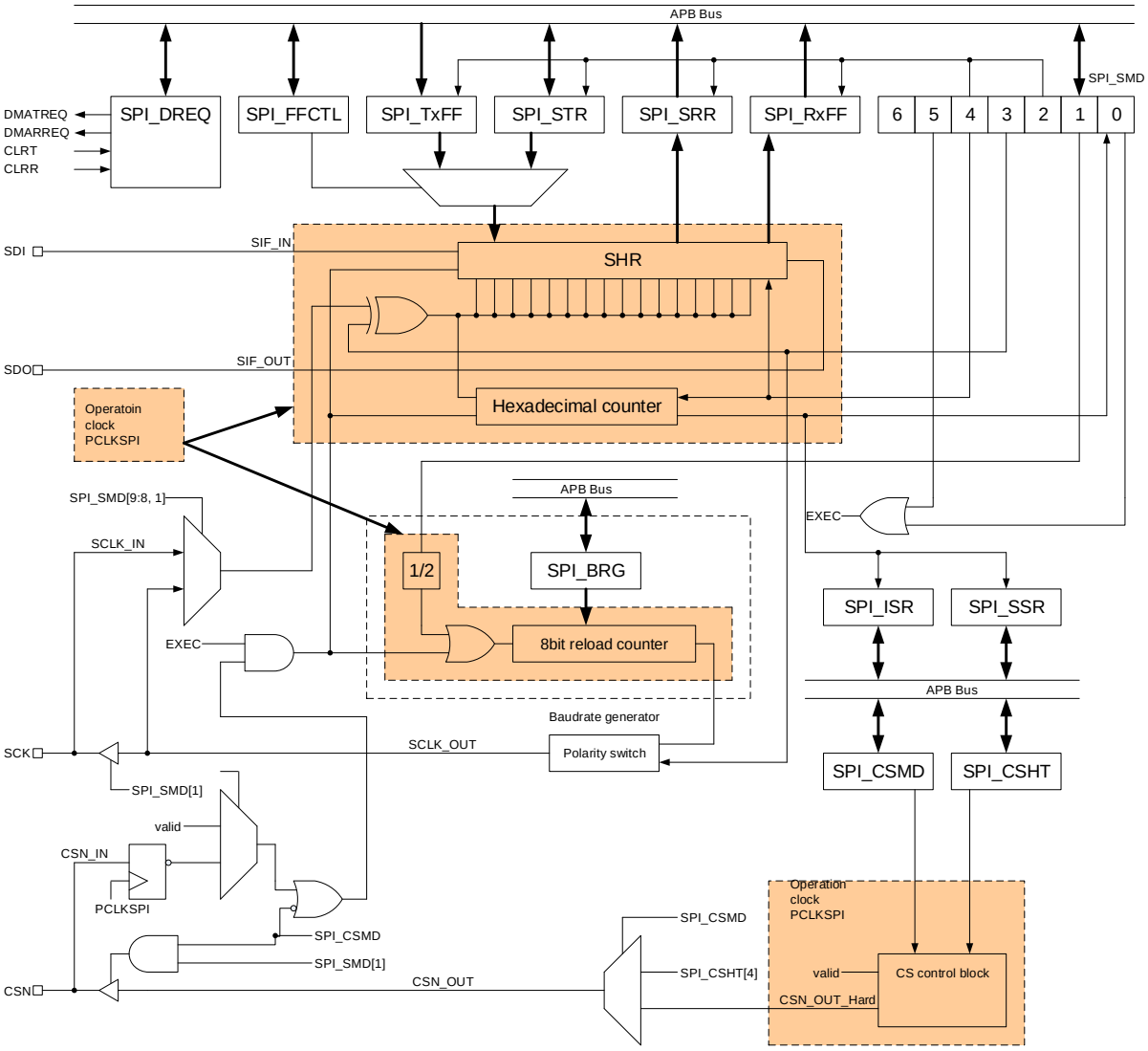


Figure 14-1 SPI circuit block diagram

14-3 Programmer's Model

As for this chapter, a register of SPI and necessary information when setting it from the microcomputer etc. are included.

The following sections are included.

- About the programmer's model
- Summary of SPI registers
- Register Descriptions

14-3-1 About the programmer's model

SPI base address is not fixed. It depends on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

14-3-2 Summary of SPI registers

SPI is controlled by the register shown in Table 14-2 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 14-2 SPI registers

Address	Abbreviation	Attribute	Content	Initial value
0x000	SPI_STR	R/W	SPI transmission register	0x0000_0000
0x004	SPI_SRR	R	SPI reception register	0x0000_0000
0x008	SPI_SMD	R/W	SPI mode register	0x0000_0000
0x00C	SPI_SSR	R	SPI status register	0x0000_0000
0x010	SPI_BRG	R/W	SPI baud rate generator	0x0000_0000
0x014	SPI_ISR	R/W	SPI interrupt factor register	0x0000_0000
0x018	SPI_DREQ	R/W	SPI DMA request generation register	0x0000_0000
0x01C	SPI_TxFF	R/W	SPI transmission FIFO register	0x0000_0000
0x020	SPI_RxFF	R/W	SPI reception FIFO register	0x0000_0000
0x024	SPI_FFCTL	R/W	SPI FIFO control register	0x0000_0000
0x028	SPI_MSK	R/W	SPI interrupt mask register	0x0000_0000
0x02C	SPI_INT	R	SPI interrupt status register	0x0000_0000
0x030	SPI_CSHT	R/W	SPI chip selection setup/ Holding time set register	0x0000_0010
0x034	SPI_CSMD	R/W	SPI chip selection mode register	0x0000_0000
0x100	Reserved	R/W	-	0x0000_0000
0x104	Reserved	R	-	0x0000_0000
0x108	Reserved	R/W	-	0x0000_0000
0x10C	Reserved	R/W	-	0x0000_0000
0x110	Reserved	R	-	0x0000_0000

14-3-3 Register Descriptions

14-3-3-1 SPI transmission register:SPI_STR

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	STR15	STR14	STR13	STR12	STR11	STR10	STR9	STR8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	STR7	STR6	STR5	STR4	STR3	STR2	STR1	STR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0 : STR15-0
Register of transmission data of SPI (8 bits/16 bits).
At Read
Read data from transmission data register.
At Write
Write data to transmission data register.

Remarks) Write transmission data to transmission data register, and set SSTR or WTR of SPI_SMD register to start transmission, transmission data is reloaded to shift register

Attention) When the following transmission data is not set to transmission register while continuous transmission, the same transmission data is reloaded. In 8 bits transmission, 0 should be written as higher 8 bits data.

14-3-3-2 SPI reception register:SPI_SRR

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	SRR15	SRR14	SRR13	SRR12	SRR11	SRR10	SRR9	SRR8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	SRR7	SRR6	SRR5	SRR4	SRR3	SRR2	SRR1	SRR0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit15-0 : SRR15-0
Register of reception data of SPI (8 bits/16 bits).
Read data from reception data register.

Attention) When one frame (8 bits/16 bits) transfer completes, data in shift register is - ccurred- ed to reception data register. In 8 bits transfer, 0 should be written as higher 8 bits data.

14-3-3-3 SPI mode register:SPI_SMD

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	DMS1	DMS0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	REGCLR	WTR	CHL	PO	LM	BGE	SSTR
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9-8 :DMS1-0
The mode of dividing is selected when internal clock is used.
00 b:Baud rate setting mode.
01 b:Divide by 2
10 b:Divide by 4
11 b:Reserved

Remarks) As for baud rate setting mode, the value of dividing is decided depending on SPI_BRG register. As for the divide by 2 mode, baud rate value is fixed to 2(2Tcyc) regardless of the value of the SPI_BRG register. As for the divide by 4 mode, baud rate value is fixed to 4(4Tcyc) regardless of the value of the SPI_BRG register.

Attention) This bit becomes effective only when BGE(bit1) is “1”,. The value 11 b is prohibited.

Note) Tcyc is the cycle time of function clock.

Bit6 :REGCLR
The data clear of transmission, reception, and shift register.
0:Hold data of transmission/reception/shift register/transmission FIFO/reception FIFO.
1:Clear data of transmission/reception/shift register/transmission FIFO/reception FIFO.

Remarks) Clear function by this register clears to 0 all data in FIFO, and clears all pointers which control the position of data in FIFO and all pointers return to initial state.

Attention) Set “0” to the register to transfer data. No valid data is transferred while the register is set to “1”.

Bit5 :WTR
Control continuous transfer.
0:Continuous transfer operation stop.
1:Continuous transfer operation start.

Remarks) Continuous transfer starts when this register is set to “1”, and data in transmission register (transmission FIFO when FIFO is used) is reload to shift register. Regarding transmission data of 2nd frame and after 2nd frame, data is reload when the transfer of previous frame finishes.

Attention) Transmission stops when transmission FIFO becomes empty when FIFO is used. Set “0” to WTR to stop continuous transmission when FIFO is not used because transmission doesn’t stop automatically when FIFO is not used.
Followings are the - ccurred when set “0” to WTR during continuous transfer.

- When internal clock is selected

When FIFO is used: After the completion of one frame transfer ongoing, continuous transfer stops.

When FIFO is unused: After the completion of one frame transfer ongoing, continuous transfer stops.

- When external clock is selected

When FIFO is used: After the completion of one frame transfer ongoing, continuous transfer stops.

When FIFO is unused: At the timing of setting “0” to WTR, continuous transfer stops. When transfer starts again after stop, remaining data is not considered. The transmission is performed by using the data in transmission register when transfer starts again.

Continuous transfer is performed when both of SSTR and WTR are set.

Bit4 :CHL
The transfer bit length.
0:8 bit transfer.
1:16 bit transfer.

Attention) Do not change this register value while transfer.

Bit3 :PO
The polarity of SCK.
0: Output data synchronous with negative edge, get input data at positive edge and high when non-communicating
1: Output data synchronous with positive edge, get input data at negative edge and Low when non-communicating

Attention) Do not change this register value while transfer.

Bit2 :LM
The first bit of transfer data.
0:LSB first
1:MSB first

Attention) Do not change this register value while transfer.

Bit1 :BGE
Switch of external clock and internal clock, and control of baud rate generator.
0:Stop baud rate generator, and use external clock
1:Start baud rate generator, and use internal clock

Attention) Do not change this register value while transfer.

There is a constraint to the period of SCLK, when BGE is set 0.

External clock period ≥ 8 *Tcyc

Tcyc: Period of function clock.

Bit0 :SSTR
Control of one frame transfer.
0:1 frame transfer stop
1:1 frame transfer start

Remarks) 1 frame transfer starts when set “1” to this register, and the data of transmission register (transmission FIFO when FIFO is used) is reloaded to shift register.

Attention) When one frame transfer finishes, this register is cleared to “0” automatically.

Followings are the - ccurred when set “0” to SSTR during a frame transfer ongoing.

- When internal clock is selected

When FIFO is used : After the completion of one frame transfer ongoing, transfer stops.

When FIFO is unused : After the completion of one frame transfer ongoing, transfer stops.

- When external clock is selected

When FIFO is used : After the completion of one frame transfer ongoing, transfer stops.

When FIFO is unused: At the timing of setting “0” to SSTR, transfer stops. When transfer starts again after stop, remaining data is not considered. The transmission is performed by using the data in transmission register when transfer starts again.

Continuous transfer is performed when both of SSTR and WTR are set.

Before setting “1” to this register, set the registers DMS, CHL, PO, LM and BGE in order to determine the usage mode.

14-3-3-4 SPI status register:SPI_SSR

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	TFF	SHRF	RRF
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit2 : TFF

The state of data of transmission register.
0:Data doesn't exist in transmission register
1:Data exists in transmission register

Remarks) This register is set when data is set to transmission register, and cleared automatically when transmission data is reload to shift register. Moreover, it is cleared when REGCLR of SPI_SMD register is set to "1".

Attention) If the following data is - ccurred- ed to transmission register while this register is "1", the previous data which is not reloaded yet is overwritten.

Bit1 : SHRF

The state of data of shift register.
0:Data doesn't exist in shift register.
1:Data exists in shift register.

Remarks) This register is set when data is set to shift register, and it is cleared automatically when one frame(8 bits/16 bits) transmission finishes. Moreover, it is cleared when REGCLR of SPI_SMD register is set to "1".

Bit0 : RRF

The state of data of the reception register.
0:Data doesn't exist in reception register.
1:Data exists in reception register.

Remarks) This register is set when one frame(8 bits/16 bits) reception finishes, and it is cleared automatically when data is read from reception register. Moreover, it is cleared when REGCLR of SPI_SMD register is set to "1".

14-3-3-5 SPI baud rate generator register:SPI_BRG

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0 :BRG7-0

Set baud rate of transfer clock generated in the baud rate generation part when internal clock is selected as communication clock.

Regarding setting of the baud rate, refer to 14-4-1 Baud rate generator.

Attention) The - ccurred- value range of BRG is 0x00-0xFE. Setting 0xFF is prohibited. Do not change baud rate value while transfer.

14-3-3-6 SPI interrupt factor register:SPI_ISR

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	CS_END	BURST_ END	RxORE	TxORE	RxOWE	TxOWE	RxFULL
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	TxFULL	RxEMP	TxEMP	RxWLM	TxWLM	ROWE	OVE	SPIF
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit14 : CS_END
The completion of CS.
At Read
0:CS not complete.
1:CS complete.

Remarks) This register should be used if transfer using CS is performed by internal clock. This register becomes “1” when transition of CS happens from Low to High.

Attention) CS function isn’t available for this model.

At Write
0:No effect on CS_END.
1:Clear CS_END.

Bit13 :BURST_END
The completion of continuous transfer.
At Read
0:Continuous transfer not complete.
1:Continuous transfer complete.

Remarks) This register is used for continuous transfer.

At Write
0:No effect on BURST_END.
1:Clear BURST_END.

Bit12 :RxORE
Reception FIFO underflow error.
This indicates data read happens when reception FIFO is empty.
At Read
0:No error.
1:Error - ccurred.

At Write
0:No effect on RxORE.
1:Clear RxORE.

Attention) It can be cleared by REGCLR of the SPI_SMD register and RxFF_CLR of SPI_FFCTL register, too.

Bit11 :TxORE
Transmission FIFO underflow error.
This indicates data read happens when transmission FIFO is empty.
At Read
0:No error.
1:Error - ccurred.

At Write
0:No effect on TxORE.
1:Clear TxORE.

Attention) It can be cleared by REGCLR of the SPI_SMD register and TxFF_CLR of SPI_FFCTL register, too.

Bit10 :RxOWE
Reception FIFO overwrite error.
This indicates data write to shift register happens when reception FIFO is full.
At Read
0:No error.
1:Error - ccurred.

At Write
0:No effect on RxOWE.
1:Clear RxOWE.

Attention) It can be cleared by REGCLR of the SPI_SMD register and RxFF_CLR of SPI_FFCTL register, too.

Bit9 :TxOWE
Transmission FIFO overwrite error.
This indicates write data to transmission FIFO happens when transmission FIFO is full.
At Read
0:No error.
1:Error - ccurred.

At Write
0:No effect on TxOWE.
1:Clear TxOWE.

Attention) It can be cleared by REGCLR of the SPI_SMD register and TxFF_CLR of SPI_FFCTL register, too.

Bit8 :RxFULL
Reception FIFO data full.
This indicates reception FIFO is full.
At Read
0:Not full with data.
1:Full with data.

At Write
0:No effect on RxFULL.
1:Clear RxFULL.

Attention) It can be cleared by reading receive data. Moreover, it can be cleared by REGCLR of the SPI_SMD register and RxFF_CLR of SPI_FFCTL register, too.

Bit7 :TxFULL
Transmission FIFO data full.
This indicates transmission FIFO is full.
At Read
0:Not full with data.
1:Full with data.

At Write
0:No effect on TxFULL.
1:Cleare TxFULL.

Attention) It can be cleared when transmission data is reload to shift register. Moreover, it can be cleared by REGCLR of the SPI_SMD register and TxFF_CLR of SPI_FFCTL register, too.

Bit6 :RxEMP
Reception FIFO data empty.
This indicates whether there is data in reception FIFO or not.
At Read
0:No receive data (empty).
1:Receive data exists.

At Write
0:No effect on RxEMP.
1:Clear RxEMP.

Attention) It can be cleared when valid receive data is written to shit register. Moreover, it can be cleared by REGCLR of the SPI_SMD register and RxFF_CLR of SPI_FFCTL register, too.

Bit5 :TxEMP
Transmission FIFO data empty.
This indicates whether there is data in transmission FIFO or not.
At Read
0:No transmission data (empty).
1:Transmission data exists.

At Write
0:No effect on TxEMP.
1:Clear TxEMP.

Attention) It can be cleared when valid transmit data is written to transmission FIFO. Moreover, it can be cleared by REGCLR of the SPI_SMD register and TxFF_CLR of SPI_FFCTL register, too.

Bit4 :RxWLM
Reception FIFO water level match.
This indicates the occupied number of depths in reception FIFO reaches water level.
At Read
0: The occupied number of depths in reception FIFO is less than reception FIFO water level.
1: The occupied number of depths in reception FIFO is equal or more than reception FIFO water level.

Remarks) It is set when the occupied number of depths in reception FIFO reaches reception FIFO water level at data tramsfer from shift register to reception FIFO.

Attention) When RxFF_set_LV2-0 of SPI_FFCTL register is set to Reserved, this register is never set.

At Write
0:No effect on RxWLM.
1:Clear RxWLM.

Attention) It is cleared if the occupied number of depths in reception FIFO becomes less than water level specified when receive data is read . Moreover, it can be cleared by REGCLR of the SPI_SMD register and RxFF_CLR of SPI_FFCTL register, too.

Bit3 : TxWLM
Transmission FIFO water level match.
This indicates the occupied number of depths in transmissions FIFO reaches water level.
At Read
0: The occupied number of depths in transmissions FIFO exceeds transmissions FIFO water level.
1: The occupied number of depths in transmissions FIFO is equal or less than transmission FIFO water level.

Remarks) It is set if the occupied number of depths in transmission FIFO reaches water level specified by decreasing when data is reloaded from transmission FIFO to shift register.

Attention) When TxFF_set_LV2-0 of the SPI_FFCTL register is set to Reserved, this register is never set. Moreover, this register is “0” while reset, and becomes “1” after reset release.

At Write
0:No effect on TxWLM.
1:Clear TxWLM.

Attention) It is cleared if the occupied number of depths in transmission FIFO exceeds water level specified by increasing when data is written to transmission FIFO. Moreover, it is cleared by REGCLR of SPI_SMD register and TxFF_CLR of SPI_FFCTL register, too.

Bit2 :ROWE
Reception register overwrite error.
This indicates overwrite to reception register happens when reception FIFO is not used.
At Read
0:No error.
1:Error - ccurred.

Remarks) This register shows that the following data is - ccurred- ed to SPI_SRR while the previous data exists in SPI_SRR when reception FIFO is not used, and the previous data is overwritten.

Attention) Data is transmitted to SPI_SRR and this register is set at the completion of transfer of one frame while only transmission is performed.

At Write
0:No effect on ROWE.
1:Clear ROWE.

Attention) It is cleared by REGCLR of SPI_SMD register.

Bit1 :OVE
Overrun error.
This indicates the number of serial clocks is more than expected when FIFO(transmission, reception) is not used.
At Read
0:No error.
1:Error - ccurred.

At Write
0:No effect on OVE.
1:Clear OVE.

Attention) It is cleared by REGCLR of SPI_SMD register.
Please mask the interrupt by setting of the interrupt mask register(SPI_MSK) when FIFO(transmission, reception) is used.
.

Bit0 :SPIF
SPI transfer end flag.
This indicates one frame transfer completes when FIFO(transmission, reception) is not used.
At Read
0:Not complete.
1:Complete.

At Write
0:No effect on SPIF.

1:Clear SPIF.

Attention) It is cleared by REGCLR of SPI_SMD register.

Please mask the interrupt by setting of the interrupt mask register(SPI_MSK) when continuous transmission is used.

Please mask the interrupt by setting of the interrupt mask register(SPI_MSK) when FIFO(transmission, reception) is used.

14-3-3-7 SPI DMA request generation register:SPI_DREQ

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	DREQ1	DREQ0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : DREQ1
Control transmission side DMA request.
0:Transmission DMA request disable.
1:Transmission DMA request enable.

Bit0 : DREQ0
Control reception side DMA request.
0:Reception DMA request disable
1:Reception DMA request enable.

14-3-3-8 SPI transmission FIFO register:SPI_TxFF

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	TxFF DATA15	TxFF DATA14	TxFF DATA13	TxFF DATA12	TxFF DATA11	TxFF DATA10	TxFF DATA9	TxFF DATA8
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Bit	7	6	5	4	3	2	1	0
Name	TxFF DATA7	TxFF DATA6	TxFF DATA5	TxFF DATA4	TxFF DATA3	TxFF DATA2	TxFF DATA1	TxFF DATA0
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit15-0 : TxFFDATA15-0
Write data to transmission FIFO register.

Remarks) This register is data register written to 16 depths FIFO storing transmit data that will be reloaded to shift register. The sequence of reload to shift register is the same as the write sequence to this register. The write data is not - ccurred- ed when transmission FIFO is full.

Attention) When data is reloaded to shift register while there is no transmission data, the data is not valid data. It is cleared by TxFF_CLR of SPI_FFCTL and REGCLR of SPI_SMD register.

14-3-3-9 SPI reception FIFO register:SPI_RxFF

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	RxFF DATA15	RxFF DATA14	RxFF DATA13	RxFF DATA12	RxFF DATA11	RxFF DATA10	RxFF DATA9	RxFF DATA8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	RxFF DATA7	RxFF DATA6	RxFF DATA5	RxFF DATA4	RxFF DATA3	RxFF DATA2	RxFF DATA1	RxFF DATA0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit15-0 : RxFFDATA15-0
Read data from reception FIFO register.

Remarks) This register is data register read from 16 depths FIFO storing receive data which will be transferred from shift register. The sequence of read data is the same as the written sequence to shift register. The data in shift register is - ccurred- ed to reception FIFO when - ccurred- of one frame completes. The data from shift register is not - ccurred- ed when reception FIFO is full.

Attention) Read data from this register is not valid while there is no reception data. It is cleared by RxFF_CLR of SPI_FFCTL and REGCLR of SPI_SMD register.

14-3-3-10 SPI FIFO control register:SPI_FFCTL

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	RxFF_ set_LV2	RxFF_ set_LV1	RxFF_ set_LV0	-	-	RxFF_ CLR	RxFF_ EN
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	TxFF_ set_LV2	TxFF_ set_LV1	TxFF_ set_LV0	-	-	TxFF_ CLR	TxFF_ EN
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Bit14-12 :RxFF_set_LV2-0
Reception FIFO water level.
000 b:Water Level2
001 b:Water Level4
010 b:Water Level8
011 b:Water Level12
100 b:Water Level14
101 b-111 b:Reserved

Remarks) Water Level is the number of valid data in FIFO. It is a condition to generate DMA request and interrupt when FIFO is used.

Attention) DMA reception request is activated when the number of data in FIFO is equal to or more than this register value. If the transfer of the number of the value specified is performed and the condition is not met, DMA reception request is negated.

Bit9 :RxFF_CLR
Reception FIFO clear.
0:No effect on data in reception FIFO.
1:Clear data in reception FIFO.

Attention) Reception FIFO is cleared by REGCLR of SIO_SMD register.

Bit8 :RxFF_EN
Control reception FIFO enable.
0:Reception FIFO disable(receive data is transferred to SPI_SRR register).
1:Reception FIFO enable(receive data is transferred to reception FIFO).

Bit6-4 : TxFF_set_LV2-0
Transmission FIFO water level.
000 b:Water Level2
001 b:Water Level4
010 b:Water Level8
011 b:Water Level12
100 b:Water Level14
101 b-111 b:Reserved

Remarks) Water Level is the number of valid data in FIFO. It is a condition to generate DMA request and interrupt when FIFO is used.

Attention) DMA transmission request is activated when the number of data in FIFO is equal or less than this register value. When the number of data in FIFO becomes more than this register value by writing data, DMA transmission request is negated. The setting of water level 14 is not allowed to use DMA transmission request.

Bit1 : TxFF_CLR
Transmission FIFO clear.
0:No effect on data in transmission FIFO.
1:Clear data in transmission FIFO.

Attention) Transmission FIFO is cleared by REGCLR of the SIO_SMD register.

Bit0 : TxFF_EN
Transmission FIFO is enable.
0:Transmission FIFO disable(The transmission data is - ccurred- ed to SPI_STR register).
1:Transmission FIFO enable(The transmission data is - ccurred- ed to transmission FIFO).

Attention) Do not change each register while DMA transfer is ongoing.

14-3-3-11 SPI interrupt mask register:SPI_MSK

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	M_ CS_END	M_BURST _ END	M_RxORE	M_TxORE	M_ RxOWE	M_ TxOWE	M_ RxFULL
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	M_ TxFULL	M_ RxEMP	M_ TxEMP	M_ RxWLM	M_ TxWLM	M_ROWE	M_OVE	M_SPIF
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit14 :M_CS_END
Interrupt mask register of CS completion.
0:CS completion interrupt disable.
1:CS completion interrupt enable.

Attention) CS function isn't available for this model.

Bit13 :M_BURST_END
Interrupt mask register of continuous transfer completion.
0:Continuous transfer completion interrupt disable.
1:Continuous transfer completion interrupt enable.

Bit12 :M_RxORE
Interrupt mask register of reception FIFO overread error.
0:Error interrupt disable.
1:Error interrupt enable.

Bit11 :M_TxORE
Interrupt mask register of transmission FIFO overread error.
0:Error interrupt disable.
1:Error interrupt enable.

Bit10 :M_RxOWE
Interrupt mask register of reception FIFO overwrite error.
0:Error interrupt disable.
1:Error interrupt enable.

Bit9 :M_TxOWE
Interrupt mask register of transmission FIFO overwrite error.
0:Error interrupt disable.
1:Error interrupt enable.

Bit8 :M_RxFULL
Interrupt mask register of reception FIFO data full.
0:Reception FIFO data full interrupt disable.
1:Reception FIFO data full interrupt enable.

Bit7 :M_TxFULL
Interrupt mask register of transmission FIFO data full.
0:Transmission FIFO data full interrupt disable.
1:Transmission FIFO data full interrupt enable.

Bit6 :M_RxEMP
Interrupt mask register of reception FIFO data empty.
0:Reception FIFO data empty interrupt disable.
1:Reception FIFO data empty interrupt enable.

Bit5 :M_TxEMP
Interrupt mask register of transmission FIFO data empty.
0:Transmission FIFO data empty interrupt disable.
1:Transmission FIFO data empty interrupt enable.

Bit4 :M_RxWLM
Interrupt mask register of reception FIFO water level match.
0:Reception FIFO water level match interrupt disable.
1:Reception FIFO water level match interrupt enable.

Bit3 :M_TxWLM
Interrupt mask register of transmission FIFO water level match.
0:Transmission FIFO water level match interrupt disable.
1:Transmission FIFO water level match interrupt enable.

Bit2 :M_ROWE
Interrupt mask register of reception register overwrite error.
0:Error interrupt disable.
1:Error interrupt enable.

Bit1 :M_OVE
Interrupt mask register of overrun error.
0:Error interrupt disable.
1:Error interrupt enable.

Bit0 :M_SPIF
Interrupt mask register of SPI transfer complete.
0:Transfer end interrupt disable.
1:Transfer end interrupt enable.

14-3-3-12 SPI interrupt status register:SPI_INT

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	I_ CS_END	I_BURST_ END	I_RxORE	I_TxORE	I_ RxOWE	I_TxOWE	I_RxFULL
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	I_TxFULL	I_RxEMP	I_TxEMP	I_RxWLM	I_TxWLM	I_ROWE	I_OVE	I_SPIF
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit14 :I_CS_END
Status of CS completion interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Attention) CS function isn't available for this model.

Bit13 :I_BURST_END
Status of continuous transfer completion interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit12 :I_RxORE
Status of Reception FIFO over read error interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit11 :I_TxORE
Status of Transmission FIFO over read error interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit10 :I_RxOWE
Status of Reception FIFO overwrite error interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit9 :I_TxOWE
Status of Transmission FIFO overwrite error interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit8 :I_RxFULL
Status of Reception FIFO data full interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit7 :I_TxFULL
Status of Transmission FIFO data full interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit6 :I_RxEMP
Status of Reception FIFO data empty interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit5 :I_TxEMP
Status of Transmission FIFO data empty interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit4 :I_RxWLM
Status of Reception FIFO water level match interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit3 :I_TxWLM
Status of Transmission FIFO water level match interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit2 :I_ROWE
Status of The reception register overwrite error interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit1 :I_OVE
Status of The - ccurre error interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

Bit0 :I_SPIF
Status of The SPI transfer end interrupt request.
0:Interrupt request has not - ccurred.
1:Interrupt request has - ccurred.

14-3-3-13 SPI chip selection setup/hold time set register:SPI_CSHT

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	CSThd3	CSThd2	CSThd1	CSThd0	CSTsu3	CSTsu2	CSTsu1	CSTsu0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	CST_REG	-	-	CSThd_EN	CSTsu_EN
Reset	0	0	0	1	0	0	0	0
R/W	R	R	R	R/W	R	R	R/W	R/W

Remarks) When the external clock is selected, this register is not used.

Attention) Chip selection function isn't available for this model.

Bit15-12 :CSThd3-0
Chip select hold wait time.
0000 b:PCLKSPI 1 cycle wait
0001 b:PCLKSPI 2 cycle wait
0010 b:PCLKSPI 3 cycle wait
0011 b:PCLKSPI 4 cycle wait
0100 b:PCLKSPI 5 cycle wait
0101 b:PCLKSPI 6 cycle wait
0110 b:PCLKSPI 7 cycle wait
0111 b:PCLKSPI 8 cycle wait
1000 b:PCLKSPI 9 cycle wait
1001 b:PCLKSPI 10 cycle wait
1010 b:PCLKSPI 11 cycle wait
1011 b:PCLKSPI 12 cycle wait
1100 b:PCLKSPI 13 cycle wait
1101 b:PCLKSPI 14 cycle wait
1110 b:PCLKSPI 15 cycle wait
1111 b:PCLKSPI 16 cycle wait

Remarks) Hold time is PCLKSPI wait time plus hold time base. Hold time base depends on DMS(mode of divide) of SPI_SMD register.

Bit11-8 :CSTsu3-0
Chip selection setup wait time.
0000 b:PCLKSPI 1 cycle wait
0001 b:PCLKSPI 2 cycle wait
0010 b:PCLKSPI 3 cycle wait
0011 b:PCLKSPI 4 cycle wait
0100 b:PCLKSPI 5 cycle wait
0101 b:PCLKSPI 6 cycle wait
0110 b:PCLKSPI 7 cycle wait
0111 b:PCLKSPI 8 cycle wait
1000 b:PCLKSPI 9 cycle wait
1001 b:PCLKSPI 10 cycle wait
1010 b:PCLKSPI 11 cycle wait
1011 b:PCLKSPI 12 cycle wait
1100 b:PCLKSPI 13 cycle wait
1101 b:PCLKSPI 14 cycle wait
1110 b:PCLKSPI 15 cycle wait
1111 b:PCLKSPI 16 cycle wait

Remarks) Setup time is PCLKSPI wait time plus setup time base. Setup time base depends on DMS(mode of divide) of SPI_SMD register.

Bit4 :CSN_REG
Chip select register control.
Control terminal CSN output.
0:Terminal CSN state 0
1:Terminal CSN state 1

Remarks) This register is used when BGE of SPI_SMD register is set to internal clock, and CSN_EN of SPI_CSMD register is set to enable.

Bit1 : CSThd_EN
Control chip select hold wait.
0: Disable chip select hold wait
1: Enable chip select hold wait.

Remarks) This register is used for hard flow control of CSN.

Attention) Hold time is set to hold time base when CSThd is not used.

Bit0 :CSTsu_EN
Control chip select setup wait.
0:Disable chip select setup wait.
1:Enable chip select setup wait.

Remarks) This register is used for hard flow control of CSN.

Attention) Setup time is set to setuo time base when when CSTsu is not used.

14-3-3-14 SPI chip selection mode register:SPI_CSMD

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	CSN_SEL	CSN_EN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Remarks) When external clock is selected, this register is not used.

Attention) Chip selection function isn't available for this model

Bit1 :CSN_SEL
Control CSN select.
0:Hard flow control of CSN.
1:S oft register control of CSN.

Remarks) This. Register selects one of two methods of CSN terminal output control when internal clock is used, one is hard flow control and the other is soft register control.

Bit0 :CSN_EN
Control CSN enable.
0: CSN disable(CS function is not used).
1: CSN enable(CS function is used).

Remarks) This register sets terminal CSN function. SPI can communicate by using CSN when CS function is used. The communication is possible regardless of the state of terminal CSN when CS function is not used.

14-4 Operation

In this chapter, control flow of SPI is described.

14-4-1 Baud rate generator

Baud rate generator generates transfer clock cycle according to SPI baud rate generator register when internal clock is selected.

Transfer clock cycle is calculated by the following formula.

However, when DMS of SPI mode register is set to 01 b or 10 b, it is fixed.

Tscyc = 4 × (256 – BRG) × Tcyc

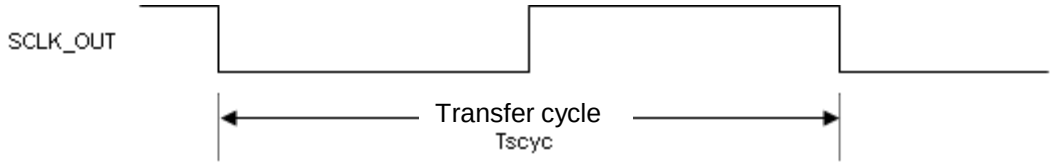


Figure 14-2 Cycle of transfer clock

Note) Tscyc is cycle of transfer clock, BRG is BRG7-0 of SPI_BRG register, Tcyc is cycle of function clock.

14-4-2 DMA transfer

Continuous DMA transfer is shown as the following in case FIFO is used/unused.

14-4-2-1 Continuous DMA transfer by FIFO unused

- Continuous transmission DMA transfer

When DREQ1 of SPI_DREQ register is set to Transmission DMA request enable, transmission DMA request is activated if TFF of SPI_SSR register is “0”(there is no transmission data). The request signal is negated by a clear signal from DMA controller which receives the request.

- Continuous reception DMA transfer

When DREQ0 of SPI_DREQ register is set to reception DMA request enable, reception DMA request is activated if RFF of SPI_SSR register is “1”(there is a receive data).. The request signal is negated by a clear signal from DMA controller which receives the request.

14-4-2-2 Continuous DMA transfer by FIFO use

- Continuous transmission DMA transfer

After setting enable to TxFF_EN of SPI_FFCTL register and water level to TxFF_set_LV2-0, and DREQ1 of SPI_DREQ register is set to transmission DMA requestenable, if the number of data in transmission FIFO is equal or less than water level, transmission DMA request is activated. The request signal is negated by a clear signal from DMA controller which receives the request.

- Continuous reception DMA transfer

After setting enable to RxFF_EN of SPI_FFCTL register and water level to RxFF_set_LV2-0, and DREQ0 of SPI_DREQ register is set to reception DMA request enable, if the number of data in reception FIFO is equal or more than water level, reception DMA request is activated. The request signal is negated by a clear signal from DMA controller which receives the request.

14-4-3 Chip selection setup/holding time

When internal clock is selected, setup/holding time of chip select signal can be set.

14-4-3-1 Chip select hold time

Hold time is decided by holding time base and hold wait time.

However, when CSThd_EN of SPI_CSHT register is “0”(disable), it is decided only by hold time base.

Hold time base is decided as follows by DMS of SPI_SMD register.

Bit9-8:DMS1-0

00 b:Baud rate setting mode = PCLKSPI 2 cycle

01 b:Divide by 2 = PCLKSPI 1 cycle

10 b:Divide by 4 = PCLKSPI 1 cycle

Hold time = hold time base + CSThd3-0 of SPI_CSHT register

14-4-3-2 Chip selection setup time

Setup time is decided by setup time base and setup wait time.

However, when CSTsu_EN of SPI_CSHT register is “0” (disable), it is decided only by setup time base.

Setup time base is decided as follows by DMS of SPI_SMD register.

Bit9-8:DMS1-0

00 b:Baud rate setting mode = Half cycle of serial clock (depend on SPI_BRG register)

01 b:Divide by 2 = Half cycle of serial clock (PCLKSPI 1 cycle)

10 b:Divide by 4 = Half cycle of serial clock (PCLKSPI 2 cycle)

Setup time = setup time base + CSTsu3-0 of SPI_CSHT register

14-4-4 Polarity of SCLK

SCLK polarity can be changed by the setting of SIO_SDM bit3 PO.

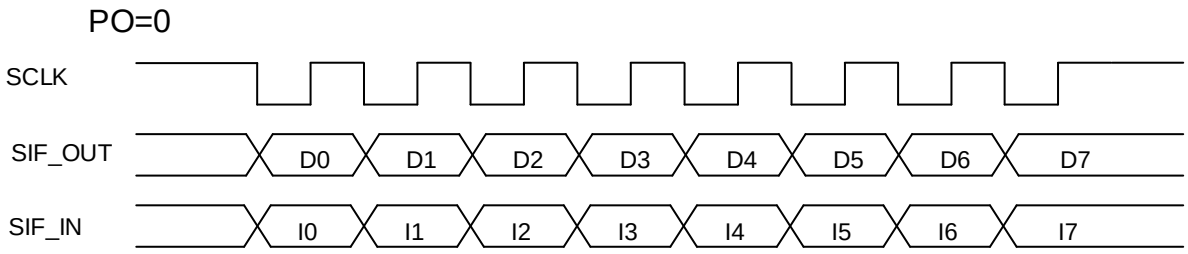


Figure 14-3

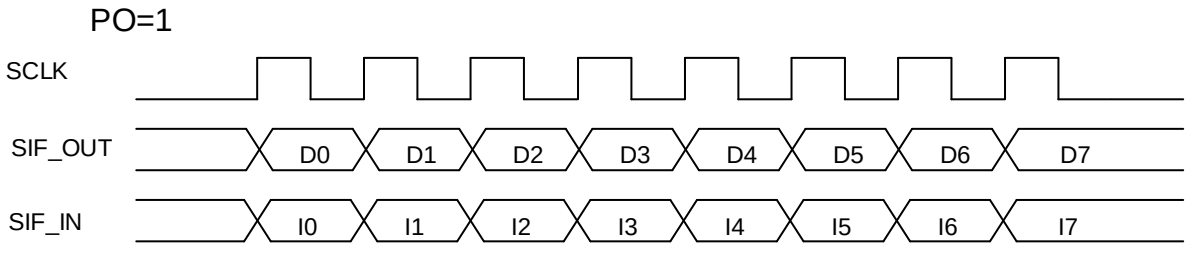


Figure 14-4

14-4-5 Timing chart

Timing figure of SPI is shown in the following.

14-4-5-1 CS disable

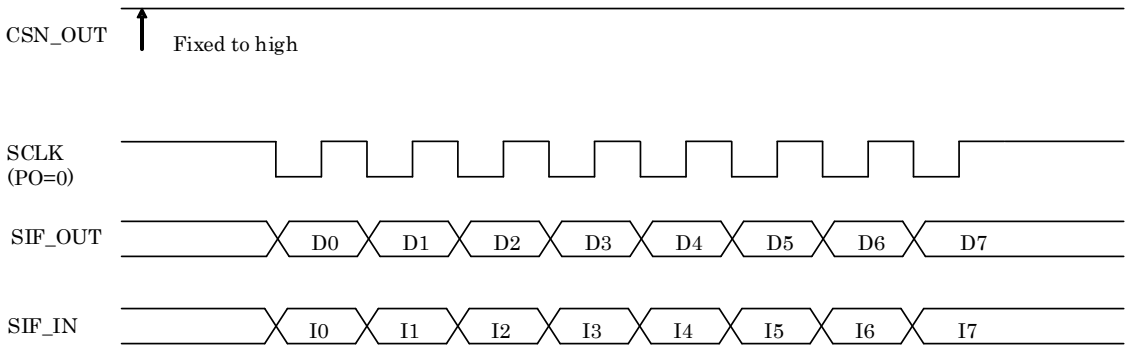


Figure 14-5 CSN .disable

14-4-5-2 CS enable, hard flow control and one frame transfer

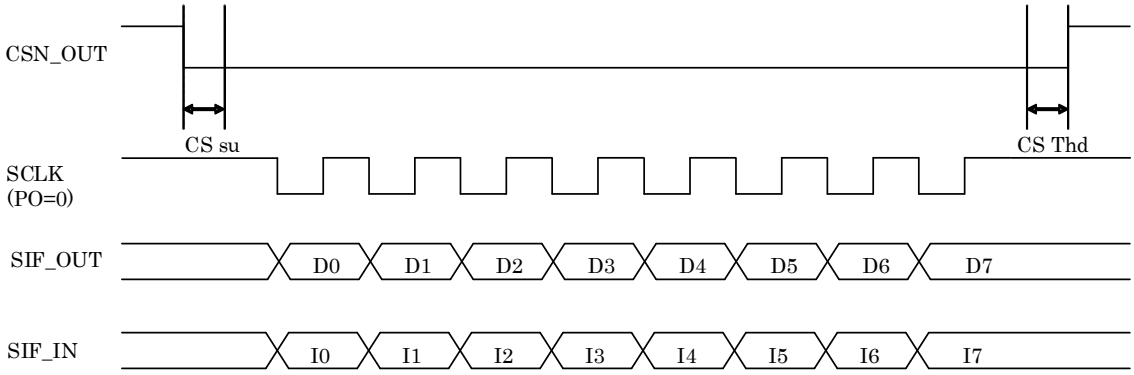


Figure 14-6 CSN enable, hard flow control, and one frame transfer

14-4-5-3 CS enable, hard flow control and continuous transfer

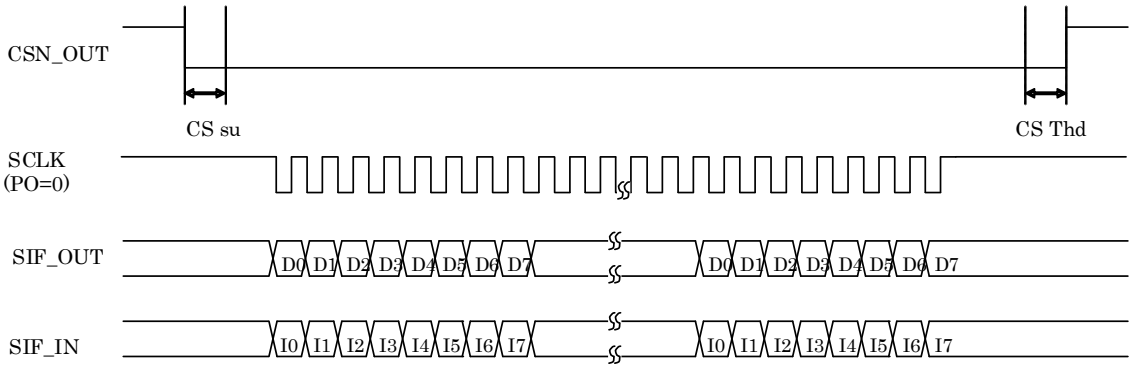


Figure 14-7 CSN enable, hard flow control and continuous transfer

14-4-5-4 CS enable, soft register control and one frame transfer

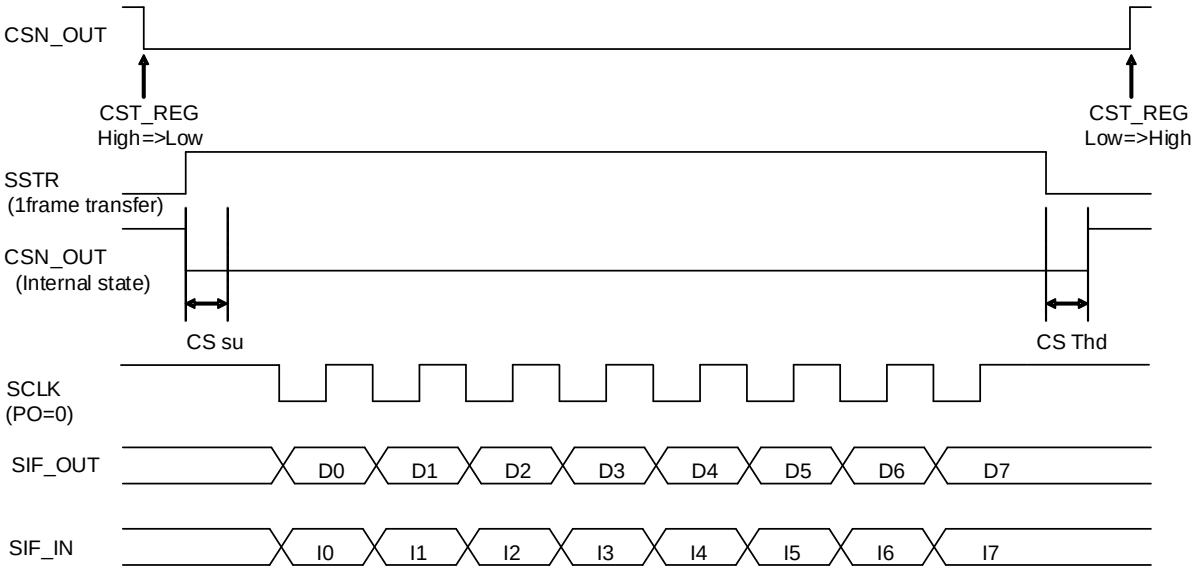


Figure 14-8 CSN enable, soft register control, and one frame transfer

14-4-5-5 CS enable, soft register control and continuous transfer

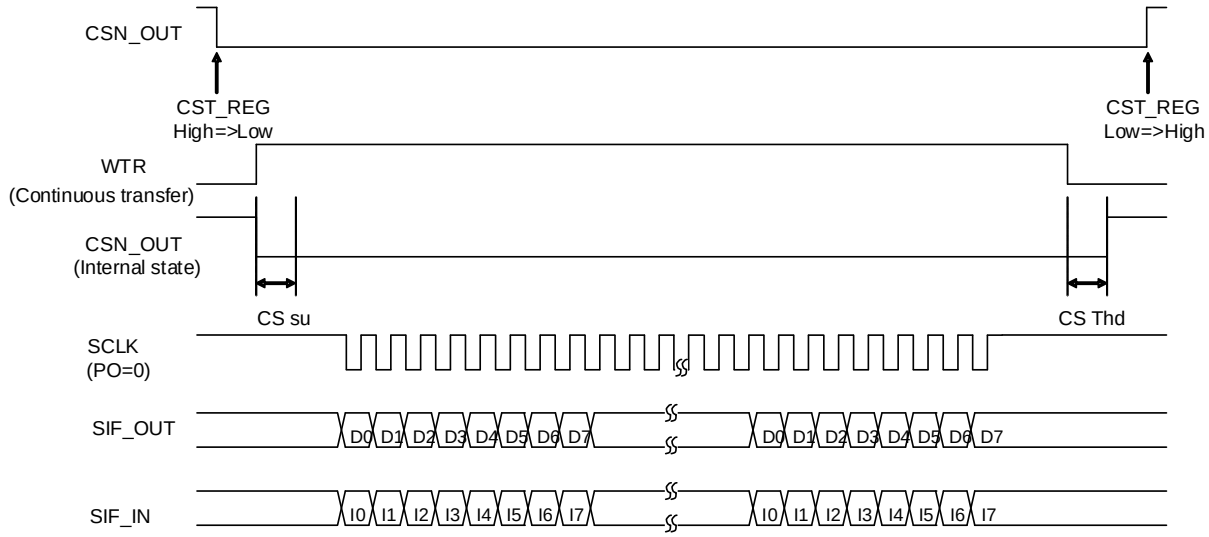


Figure 14-9 CSN enable, soft register control and continuous transfer

14-4-6 Directions

There is a constraint to the period of SCLK, when an external clock is used.

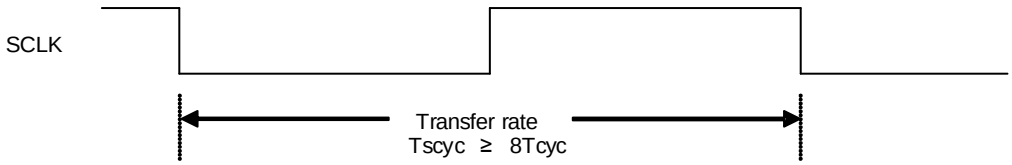


Figure 14-10

Please keep “External clock period(Transfer rate) ≥ 8 *Tcyc”

Tscyc: Period of external clock.
Tcyc: Period of function clock.

15 UART

15-1 Introduction

This chapter provides the brief summary of UARTn in the sections below:

- Brief Summary of UARTn

15-1-1 Brief Summary of UARTn

This is a start-stop synchronous communication type LSI which has internal full duplex communication UART. There are 24 different communication types.

Here are the overviews of the features:

15-1-1-1 Data Length Select Function

There are two types of data length to choose from for Transmission and reception of data.

- 7 bit transfer
- 8 bit transfer

15-1-1-2 Transmission Stop bit Select Function

There are two types of Stop bits to choose from for transmission:

- 1 bit stop
- 2 bit stop

15-1-1-3 Parity Setup Function

There are three types of parity setup to choose from:

- Even parity
- Odd parity
- No parity

15-1-1-4 Auto Break Detection Function

Break is detected when the LOW level is continuously entered into Start bit to Stop bit of Reception register data.

15-1-1-5 Polarity Select Function

This function can reverse Transmission and Reception data polarities as well as Reception clear (CTS_N) and Transmission request (RTS_N) signal polarities.

15-1-1-6 Full-duplex double buffer function / 16-steps FIFO function

Full-duplex double buffer enables continuous data transmission/reception by using 16-steps FIFO.

15-1-1-7 Error detection function

There are three different ways to detect an error:

- Parity error
- Reception register overwrite error
- Framing error

15-1-1-8 Initialization

Reset signal will initialize the setup as follows:

- Transmission/reception data : 0x00
- Data length : 7 bit
- Parity bit : None
- Stop bit : 1 bit
- Transmission/reception disallowed
- No error
- Polarity function : Not selected
- CTS/RTS function not selected
- Buffer select (FIFO (FIFO : Disabled)
- No error interrupt factor
- Error interrupt factor mask condition
- DMA request non-output

Table 15-1 Signal used with UARTn

Signal	I/O	Function
TXDn	Output	Asynchronous serial communication output
RXDn	Input	Asynchronous serial communication input
CTSn	Input	Hardware flow control signal input
RTSn	Output	Hardware flow control signal output

15-2 Functional Overview

This chapter provides overview of functional blocks used for the UARTn configuration in the sections below:

- UARTn Circuit Configuration

15-2-1 UARTn Circuit Configuration

UARTn circuit configuration is shown in a Figure 15-1

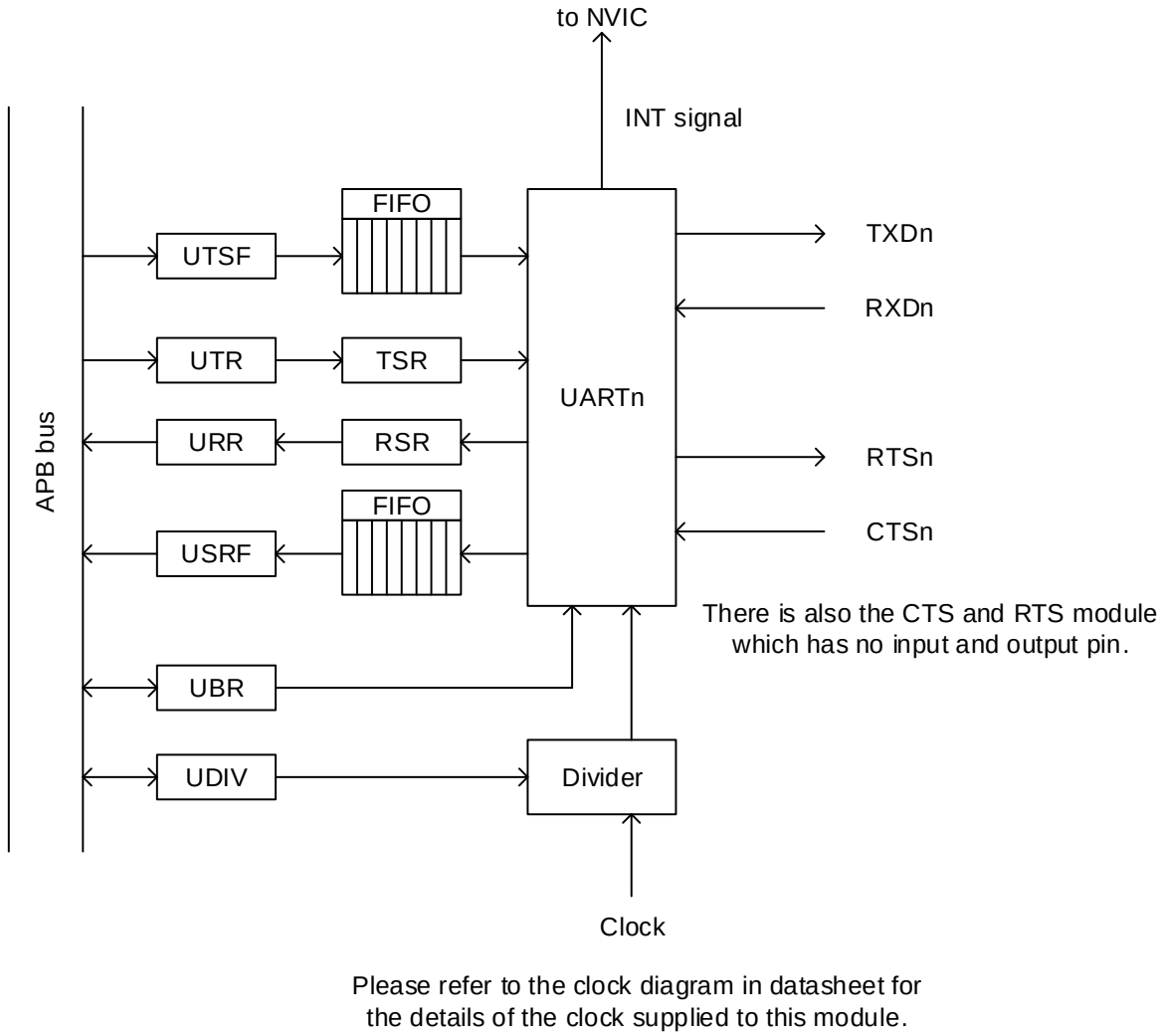


Figure 15-1 UARTn Configuration

15-3 Programmer's Model

This chapter contains UARTn register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of UART registers
- Register Descriptions

15-3-1 About the programmer's model

UARTn base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

15-3-2 Summary of UART registers

UART is controlled by the registers shown in Table 15-2 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 15-2 UARTn registers

OFFSET	abbreviation	Attribute	Register Names	Default value
0x00	UTR	R/W	UARTn Transmission register	0x0000_0000
0x04	URR	R/W	UARTn Reception register	0x0000_0000
0x08	UMD	R/W	UARTn Mode register	0x0000_0000
0x0C	UCM	R/W	UARTn Command register	0x0000_0000
0x10	USR	R/W	UARTn Status register	0x0000_2600
0x14	UBR	R/W	UARTn Baud rate register	0x0000_0000
0x18	UISR	R/W	UARTn Interrupt status register	0x0000_0400
0x1C	UDIV	R/W	UARTn Division setup register	0x0000_000F
0x20	UIEN	R/W	UARTn Interrupt enable register	0x0000_0000
0x24	U INT	R/W	UARTn Interrupt request register	0x0000_0000
0x28	USTF	W	UARTn Transmission FIFO register	0x0000_0000
0x2C	USRF	R	UARTn Reception FIFO register	0x0000_0000
0x30	USFC	R/W	UARTn FIFO Control register	0x0000_0000
0x34	UDMA	R/W	UARTn DMA Control register	0x0000_0000
0x38	USFS	R	UARTn FIFO Status register	0x0000_0000

15-3-3 Register Descriptions

15-3-3-1 UARTn Transmission register : UTR

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	UTR7	UTR6	UTR5	UTR4	UTR3	UTR2	UTR1	UTR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0 : UTR7-0

UARTn Transmission Register.

Data to be send by the UARTn is stored here.

Its Double buffer structure enables continuous transmission by setting a data into UTR (UART Transmission Register) after transferring a data into TSR (Transmission Sift Register).

A READ command to bits 31 ~8 will always return '0'. WRITE command is invalid.

15-3-3-2 UARTn Reception Register : URR

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	URR7	URR6	URR5	URR4	URR3	URR2	URR1	URR0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit7-0 : URR7-0

UARTn Reception Register.

It stores data received by the UARTn.

When serial data reception for 1 frame is complete, received data is transferred from RSR (Reception Shift Register) to URR (UARTn Reception Register).

A READ command to bits 31 ~8 will always return '0'. WRITE command is invalid.

15-3-3-3 UARTn Mode register : UMD

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	CTSEN	RTSEN	PO	STL	PS1	PS0	CL
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 6 : CTSEN

Select the CTS functions

0 : CTS function Disable

1 : CTS function Enable

- When CTS function Disable is selected, the reception operation is active regardless of the status of CTS_N input pin.
- When CTS function Enable is selected, the reception operation is controlled by the status of the CTS_N input pin.

If it is at LOW level (HIGH level when the polarity is inverted): Transmission enable

If it is at HIGH level (LOW level when the polarity is inverted): Transmission disable

Even if the data transmission status becomes inactive in middle of the process, it will complete transmitting 1 frame worth of data.

If a user wants to stop transmission in middle of the process, put CTS_N pin in the transmission disable status before reaching the middle of Stop bit transmission (1/2 bit period).

Bit 5 : RTSEN

Select the RTS Function

0 : RTS function Disable

1 : RTS function Enable

- This flag determines whether to output Transmission request signal to the RTS_N pin or not.
- When RTS function disable is selected, transmission operation is active regardless of the condition of RTS_N output pin.

RTS_N output pin is fixed to HIGH level (LOW level when the polarity is inverted).

- when RTS function enable is selected, transmission operation is controlled by the status of the RTS_N output pin.

RTS_N output pin changes by the transmission buffer condition.

At LOW level (HIGH level when the polarity is inverted) : Receivable (Reception buffer is empty)

At HIGH level (LOW level when the polarity is inverted) : Not receivable (Reception buffer has data)

Bit 4 : PO

Select the Polarity control

0 : Start bit = 0, Transmission data and Stop bit = 1 (at normal state)

1 : Start bit = 1, Transmission data (Inverted transfer data) and Stop bit = 0 (When Inverted)

- This flag selects the polarity switching of Start bit, Transmission data, Stop bit, or CTS/RTS.
- When the polarity is switched, the error detection polarity is also inverted.

Bit 3 : STL

Select the number of Stop bits

0 : 1 bit

1 : 2 bits

- This flag selects either 1 bit or 2 bits for the Stop bit at data transmission/reception.

Bit2-1 : PS1-0

Select the Parity bit

00 b : No parity

01 b : Odd parity

10 b : Even parity

11 b : Setting prohibited

- This flag selects “No parity”, “Odd parity”, “Even parity”, or at the time of data transmission/reception.
- When “No parity” is selected, parity add check will not be performed for both data transmission and reception.

Bit0 : CL

Select the number of bits for data length.

0 : 7 bits

1 : 8 bits

- This flag selects either 7 bits or 8 bits for the length of the data.
- When “0” is selected, the lower 7 bits of USTR (UARTn transmission register) is transferred at data transmission, and the most significant 1 bit of URR (UARTn reception register) is fixed to 0 when receiving data.

A READ command to bits 31 ~ 7 will always return “0”. WRITE command is invalid.

15-3-3-4 UARTn Command Register : UCM

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	REGTCLR	REGRCLR	SBRK	TE	RE
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit 4 : REGTCLR

Transmission Register Clear flag.

0 : Transmission register data in status quo

1 : Transmission register data is cleared

- This flag selects whether or not to clear data in UTR (UART transmission register) and TSR (Transmission shift register).
- Be sure to reset “0” for REGTCLR flag when transferring data. Data cannot be transferred if “1” is set in this flag.
- Do not set “1” to REGTCLR flag in the middle of data transferring process. The transfer data will be destroyed if “1” is set in middle of the process.

Bit 3 : REGRCLR

Reception Register Clear flag.

0 : Reception register data in status quo

1 : Reception register data is cleared

- This flag selects whether to clear the data in URR (UART reception register) and RSR (Reception shift register).
- Be sure to reset “0” for REGRCLR flag when transferring data. Data cannot be transferred when “1” is set.
- Do not set “1” for REGRCLR flag in the middle of data transferring process. The transfer data will be destroyed if the REGRCLR flag is set in the middle of the process.

Bit 2 : SBRK

Break Transmission Set flag

0 : Do not generate break transmission

1 : Break transmission is generated

- This flag controls the asynchronous break status transmission.
- In the Break transmission, TXD pin will output LOW level (HIGH level if the polarity is “1”) regardless of the data. Break transmission function is enabled even under transmission disabled status.

Bit 1 : TE

Transmission enable flag

0 : Transmission Disable

1 : Transmission Enable

- This flag controls Disable/Enable status of the data transfer.
- When transmission is disabled, even if a data is written into UTR (UART transmission register), the data will not be transferred into the Transmission shift register and will be stored in the UTR. Once the transmission becomes enabled, that stored data in UTR will be transferred to Transmission shift register and start data transmission process. In Transmission disabled status, TXD pin is at the HIGH level (LOW level if the polarity is “1”).

Bit 0 : RE

Reception enable flag.

0 : Reception Disable

1 : Reception Enable

- This flag controls Disable/Enable status of the data reception.
- In the reception disabled status, Start bit is not detected and Reception shift register is inoperative.

A READ command to bits 31 ~ 5 will always return “0”. WRITE command is invalid.

15-3-3-5 UART Status Register : USR

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	RxWLM	RxEMP	RxFULL	-	TxWLM	TxEMP	TxFULL
Reset	0	0	1	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	BKD	RRF	TFF
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit 14 : RxWLM

Reception FIFO water level matching flag.

0 : The numbers of Reception FIFO data is less than the Reception FIFO water level.

1 : The numbers of Reception FIFO data is more than the Reception FIFO water level

- This flag indicates that the numbers of Reception FIFO data are reaching the water level that is set for each bit.
- It is set to “0” when the numbers of Reception FIFO data becomes less than the set water level. It will be set to “1” by the Reception data register clear.

Bit 13 : RxEMP

Reception FIFO empty status flag

0 : Reception FIFO is not empty

1 : Reception FIFO is empty

- This is a flag to show whether the Reception FIFO is empty or not.
- It is cleared when the Reception FIFO becomes non-empty. It will be set to “1” by the Transmission data register clear.

Bit 12 : RxFULL

Reception FIFO data full status flag.

0 : Reception FIFO is not in “Data full” status

1 : Reception FIFO is in “Data full” status

- This is a flag to show whether the Reception FIFO is in “Data full” status or not.
- It is cleared when the Reception FIFO status become “not data full”. Transmission data register clear will clear this flag.

Bit 10 : TxWLM

Transmission FIFO water level matching flag.

0 : The numbers of Transmission FIFO data is at the Transmission FIFO water level or more.

1 : The numbers of Transmission FIFO data is at the Transmission FIFO water level or less

- This flag indicates that the numbers of Reception FIFO data are reaching a set water level.
- It is cleared to “0” when the numbers of Transmission FIFO data is exceeding the set water level which is set.
  It will be set to “1” by the Transmission data register clear.

Bit 9 : TxEMP

Transmission FIFO empty status flag

0 : Transmission FIFO is not empty

1 : Transmission FIFO is empty

- This is a flag to show whether the Transmission FIFO is empty or not.
- It is cleared when the Transmission FIFO becomes non-empty.
  It will be set to “1” by the Transmission data register clear.

Bit 8 : TxFULL

Transmission FIFO data full status flag.

0 : Transmission FIFO is not in “Data full” status

1 : Transmission FIFO is in “Data full” status

- This is a flag to show whether the Transmission FIFO is in “not Data full”.
- It is cleared when the Transmission FIFO status become “not data full”.
  It will be cleared by the Transmission data register clear.

Bit 2 : BKD

Break reception.

0 : No Break reception

1 : With Break reception

- This flag indicates the break reception status at the time of data reception.
- When the polarity control flag is set to “0”: Break reception is detected when LOW level is entered continuously from the Start bit to the Stop bit and this flag will be set to “1”.
- When the polarity control flag is set to “1” : Break reception is detected when HIGH level is entered continuously from the Start bit to the Stop bit and this flag will be set to “1”.

Bit1 : RRF

UART reception register full flag.

0 : No valid data in UART reception register

1 : Valid data in UART reception register is available

- This flag indicates the Reception register status.
- UART RRF (Reception register full flag) is set when a frame of data reception is completed, and it is cleared when URR (UART reception register) data is read out. Setting REGRCLR to “1” will also clear the UART RRF.

Bit 0 : TFF

UART transmission register full flag.

0 : No valid data for UART transmission register

1 : Valid data for UART transmission register is available

- This flag indicates the Transmission register status.
- At the time of transmission, UART TFF (Transmission register full flag) will be set by setting a data to UTR (UART transmission register). It will be cleared by the data transfer to Reception shift register. UART TFF will also be cleared by setting “1” to REGTCLR (Transmission register clear flag) of the UMC (UART command register). If “1” is set to UART TFF and a data is transferred to UTR, the data that have not been yet transmitted will be destroyed.

A READ command to bits 31 ~ 15, 11, and 7 ~ 3 will always return “0”. WRITE command is invalid

15-3-3-6 UART Baud rate Register : UBR

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	UBR15	UBR14	UBR13	UBR12	UBR11	UBR10	UBR9	UBR8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	UBR7	UBR6	UBR5	UBR4	UBR3	UBR2	UBR1	UBR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15-0 : UBR15-0

Baud rate register.

This is a register used for Baud rate generator setup.

The following table indicates a gap between bit rates used as a transfer spec and the actual transfer rate.

Transfer rate conversion

Dedicated baud rate setting

Operation frequency	20 MHz			40 MHz		
Bit rate (bps)	Set point (Decimal)	Transfer rate (bps)	Error (%)	Set point (Decimal)	Transfer rate (bps)	Error (%)
1200	64494	1199.6	-0.03	63453	1200.2	0.02
2400	65015	2399.2	-0.03	64494	2399.2	-0.03
4800	65276	4807.7	0.16	65015	4798.5	-0.03
9600	65406	9615.4	0.16	65276	9615.4	0.16
19200	65471	19230.8	0.16	65406	19230.8	0.16
38400	65503	37878.8	-1.36	65471	38461.5	0.16
57600	65514	56818.2	-1.36	65493	58139.5	0.94
115200	65525	113636.4	-1.36	65514	113636.4	-1.36
230400	65531	250000	8.51	65525	227272.7	-1.36

Calculation method

Set point = 65536 – (operation frequency / (Division set point + 1) × Bit rate))

Transfer rate = Operation frequency / {(Division set point + 1) × (65536 – Baud rate set point)}

Error = (Transfer rate – Bit rate) / Bit rate × 100

A READ command to bits 31 ~ 16 will always return “0”. WRITE command is invalid.

15-3-3-7 UART Interrupt Status Register : UISR

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	RxWLM_IS	RxUFE_IS	RxOWE_IS	-	TxWLM_IS	-	TxOWE_IS
Reset	0	0	0	0	0	1	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	BUSERR	-	-	FE	ROWE	PE	UARTTF	UARTRF
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit 14 : RxWLM_IS
Reception FIFO water level matching interrupt factor flag.
0 : Numbers of Reception FIFO data is less than the Reception FIFO water level
1 : Numbers of Reception FIFO data is more than the Reception FIFO water level
- This flag indicates whether the numbers of Reception FIFO data is reaching the water level or not.

Attention) Reception FIFO water level matching interrupt factor flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 13 : RxUFE_IS
Reception FIFO underflow error flag.
0 : No error
1 : Error occurred
- This flag indicates that the Reception FIFO data was read at the empty state.

Attention) Reception FIFO underflow error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 12 : RxOWE_IS
Reception FIFO overwrite error flag.
0 : No error
1 : Error occurred
- This flag indicates that Reception FIFO was written at the data full state.

Attention) Reception FIFO overwrite error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 10 : TxWLM_IS
Transmission FIFO water level matching interrupt factor flag
0 : Numbers of Transmission FIFO data is exceeding the Transmission FIFO water level
1 : Numbers of Transmission FIFO data is at the Transmission FIFO water level or less.
- This flag indicates that the numbers of Transmission FIFO data is at the set water level or less.

Transmission FIFO water level matching interrupt factor flag.
0 : Numbers of Transmission FIFO data is more than the Transmission FIFO water level
1 : Numbers of Transmission FIFO data is less than the Transmission FIFO water level
- This flag indicates the numbers of Transmission FIFO data is at the set water level or less.

Attention) Transmission FIFO water level matching interrupt factor flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 8 : TxOWE_IS
Transmission FIFO overwrite error flag.
0 : No error
1 : Error occurred
- This flag indicates that a WRITE command was used for Transmission FIFO in “data full” state.

Attention) Transmission FIFO overwrite error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 7 : BUSERR
Bus access error flag.
0 : No error
1 : Error occurred
- This flag indicates that the Bus access did not complete (When Function clock is stopped).

Attention) Bus access error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 4 : FE
Framing error flag.
0 : No error
1 : Error occurred
- This flag indicates that the HIGH level (LOW level if the polarity is “1”) was not detected in the timing where the Stop bit was supposed to be existing.

Attention) Fleming error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 3 : ROWE
Reception register overwrite error flag.
0 : No error
1 : Error occurred
- This flag indicates that a next data is been overwritten before a reception data is read by a CPU.

Attention) Reception register overwrite error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 2 : PE
Parity error flag
0 : No error
1 : Error occurred
- This flag indicates the Parity error occurrence at the time of data reception when the parity is valid.

Attention) Parity error flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 1 : UARTTF
Transmission complete interrupt factor flag.
0 : No interrupt factor
1 : With interrupt factor
- This flag indicates the existing of Transmission complete interrupt factor for a frame.

Attention) Interrupt factor flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

Bit 0 : UARTRF
Reception complete interrupt factor flag.
0 : No interrupt factor
1 : With interrupt factor
- This flag indicates the existing of Reception complete interrupt factor for a frame.

Attention) Interrupt factor flag is not cleared automatically. Writing “1” in the program will clear this flag. A value of the flag is retained by writing “0” to this flag.

A READ command to bits 31 ~ 15, 11, and 9.6 ~ 5 will always return “0”. WRITE command is invalid.

15-3-3-8 UART Divider Setting Register : UDIV

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	UDIV3	UDIV2	UDIV1	UDIV0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit 3-0 : UDIV3-0

UART Divider setting register.

This is a divider setting register used for clocks supplied to UARTn.

Divider setting value				Divider value
UDIV3	UDIV2	UDIV1	UDIV0	
1	1	1	1	1/16 (Default)
1	1	1	0	1/15
1	1	0	1	1/14
1	1	0	0	1/13
1	0	1	1	1/12
1	0	1	0	1/11
1	0	0	1	1/10
1	0	0	0	1/9
0	1	1	1	1/8
0	1	1	0	Setting prohibited
0	1	0	1	Setting prohibited
0	1	0	0	Setting prohibited
0	0	1	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	0	0	Setting prohibited

Below table shows a gap between the standard transfer bit rate and the actual transfer rate.

Transfer rate switching
Dedicated baud rate setting

(Example calculation using 1/16 bit rate)

Operation frequency	20 MHz			40 MHz		
Bit rate (bps)	Set point (Decimal)	Transfer rate (bps)	Error (%)	Set point (Decimal)	Transfer rate (bps)	Error (%)
1200	64494	1199.6	-0.03	63453	1200.2	0.02
2400	65015	2399.2	-0.03	64494	2399.2	-0.03
4800	65276	4807.7	0.16	65015	4798.5	-0.03
9600	65406	9615.4	0.16	65276	9615.4	0.16
19200	65471	19230.8	0.16	65406	19230.8	0.16
38400	65503	37878.8	-1.36	65471	38461.5	0.16
57600	65514	56818.2	-1.36	65493	58139.5	0.94
115200	65525	113636.4	-1.36	65514	113636.4	-1.36
230400	65531	250000	8.51	65525	227272.7	-1.36

Calculation method
Set point = 65536 – (operation frequency / (Division set point + 1) × Bit rate))
Transfer rate = Operation frequency / {(Division set point + 1) × (65536 – Baud rate set point)}
Error = (Transfer rate – Bit rate) / Bit rate × 100

A READ command to bits 31 ~ 4 will always return “0”. WRITE command is invalid.

15-3-3-9 UART Interrupt Enable Register : UIEN

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	RxWLM_ IEN	RxUFE_ IEN	RxOWE_ IEN	-	TxWLM_ IEN	-	TxOWE_ IEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	BUSERR_ IEN	-	-	FE_IEN	ROWE_ IEN	PE_IEN	UARTTF_ IEN	UARTRF_ IEN
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit 14 : RxWLM_IEN
Reception FIFO water level matching interrupt factor enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Reception FIFO water level matching interrupt factor enable/disable status.

Bit 13 : RxUFE_IEN
Reception FIFO underflow error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Reception FIFO underflow error enable/disable status.

Bit 12 : RxOWE_IEN
Reception FIFO overwrite error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Reception FIFO overwrite error enable/disable status.

Bit 10 : TxWLM_IEN
Transmission FIFO water level matching interrupt factor enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Transmission FIFO water level matching interrupt factor enable/disable status.

Bit 8 : TxOWE_IEN
Transmission FIFO overwrite error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Transmission FIFO overwrite error enable/disable status.

Bit 7 : BUSERR_IEN
Bus access error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Bus access error enable/disable status.

Bit 4 : FE_IEN
Flaming error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Flaming error enable/disable status.

Bit 3 : ROWE_IEN
Reception register overwrite error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Reception register overwrite error enable/disable status.

Bit 2 : PE_IEN
Parity error enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Parity error enable/disable status.

Bit 1 : UARTTF_IEN
Transmission complete interrupt factor enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Transmission complete interrupt factor for a frame.

Bit 0 : UARTRF_IEN
Reception complete interrupt factor enable flag.
0 : Interrupt Disable
1 : Interrupt Enable
- This flag is used to select Reception complete interrupt factor for a frame.

A READ command to bits 31 ~ 15, 11, 9, and 6 ~ 5 will always return “0”. WRITE command is invalid.

15-3-3-10 UART Interrupt Request Register : UINT

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	RxWLM_ INT	RxUFE_ INT	RxOWE_ INT	-	TxWLM_ INT	-	TxOWE_ INT
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	BUSERR_ INT	-	-	FE_INT	ROWE_ INT	PE_INT	UARTTF_ INT	UARTRF_ INT
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit 14 : RxWLM_INT
Reception FIFO water level matching interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Reception FIFO water level matching interrupt request.

Bit 13 : RxUFE_INT
Reception FIFO underflow error interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Reception FIFO underflow error interrupt request.

Bit 12 : RxOWE_INT
Reception FIFO overwrite error interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Reception FIFO overwrite error interrupt request.

Bit 10 : TxWLM_INT
Transmission FIFO water level matching interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Transmission FIFO water level matching interrupt request.

Bit 8 : TxOWE_INT
Transmission FIFO overwrite error interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Transmission FIFO overwrite error interrupt request.

Bit 7 : BUSERR_INT
Bus access error interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Bus access error interrupt request.

Bit 4 : FE_INT
Flaming error interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Flaming error interrupt request.

Bit 3 : ROWE_INT
Reception register overwrite error interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Reception register overwrite error interrupt request.

Bit 2 : PE_INT
Parity error interrupt request flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates Parity error interrupt request.

Bit 1 : UARTTF_INT
Transmission complete interrupt flag
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates the Transmission complete interrupt request of a frame.

Bit 0 : UARTRF_INT
Reception complete interrupt flag.
0 : No Interrupt request
1 : With Interrupt request
- This flag indicates the Reception complete interrupt request of a frame.

A READ command to bits 31 ~ 15, 11, 9, and 6 ~ 5 will always return “0”. WRITE command is invalid.

15-3-3-11 UART Transmission FIFO Register : USTF

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	TxFF DATA7	TxFF DATA6	TxFF DATA5	TxFF DATA4	TxFF DATA3	TxFF DATA2	TxFF DATA1	TxFF DATA0
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bits 7-0 : TxFFDATA7-0

Transmission FIFO data.
This bit retains UART transmission FIFO data (7 bit / 8 bit).
When 7 bit is set as the data length, data in Bit7 is not transferred.

This register adds data to Transmission FIFO (8 bit×16) which holds UART transmission data.
The following bit allocation shows an allocation of 1-step bit out of 16-steps. A common address is used for writingdata into Transmission FIFO.
Writing data to this address will also add data to Transmission FIFO. No data can be extracted from the Transmission FIFO.
FIFO (Data, Read/Write pointer, amount of data) initialization can be done by Transmission data register clear.

In Transmission data register clear state, no data is written into this register.
In Transmission FIFO data full state, data will not be added even when you try to write a data to this register. It will cause an Overwrite error.
Read data to this register is not valid. Read value is fixed to 0x00.
In case of the Transmission FIFO disable (Transmission buffer enable), you cannot write data to this register.

A READ command to bits 31 ~ 8 will always return “0” . WRITE command is invalid.

15-3-3-12 UART Reception FIFO Register : USRF

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	RxFF DATA7	RxFF DATA6	RxFF DATA5	RxFF DATA4	RxFF DATA3	RxFF DATA2	RxFF DATA1	RxFF DATA0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit7-0 : RxFFDATA7-0

Reception FIFO data.

This bit indicates extracted reception data from UART reception FIFO.

When the data length is set to 7 bit, “0” will be set to Bit7.

This register is used to extract FIFO data (8 bit×16) which holds UART reception data.

The following bit allocation shows an allocation of 1-step out of 16-steps. At the end of receiving for a frame, data in the Reception shift register will be added into the Reception FIFO.

FIFO (Data, Read/Write pointer, amount of data) initialization can be done by Reception data register clear.

Data in this register cannot be read in the Reception data register clear state.

In Reception FIFO data full state, received data from Reception shift register will not be added into the Reception FIFO. It will cause an Overwrite error.

When the Reception FIFO is empty, the read data will not be valid and will cause an Underflow error. Read value is fixed to 0x00.

In the Reception FIFO disable state, UART reception data will not be transferred into the Reception FIFO. It will be transferred into Reception buffer.

In the UART reception disable state, there will be no data transfer from RXD pin to the Reception shift register. Data cannot be read as well. Read value is fixed to 0x00.

A READ command to bits 31 ~ 8 will always return “0”. WRITE command is invalid.

15-3-3-13 UART FIFO Control Register : USFC

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	RxFF_END
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	TxFF_EN	TxFF_set_ LV2	TxFF_set_ LV1	TxFF_set_ LV0	RxFF_EN	RxFF_set_ LV2	RxFF_set_ LV1	RxFF_set_ LV0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 8 : RxFF_END

FIFO operation selection after the Reception FIFO continuous transfer completion.
0 : Emit all FIFO data after completing reception continuous transfer. (default)
1 : Will operate by following FIFO setting even after Reception continuous transfer is complete
This bit is used for selecting FIFO operation after completing reception continuous transfer.
If Start-bit is not detected for a period of 1 bit after detecting a Stop-bit, this bit will also be used to select FIFO data read request operation (DMARREQ signal, RxWLM status, RxWLM interrupt generation) activity by temporarily setting “1” to the Reception FIFO water level. It will go back to its normal operation once the Start-bit is detected.

Bit 7 : TxFF_EN

Transmission FIFO enable flag
0 : Transmission FIFO Disable / Transmission buffer Enable (default)
1 : Transmission FIFO Enable / Transmission buffer Disable
This bit is used to select either Transmission FIFO or Transmission buffer for use.
Data will not be transferred from the Transmission shift register for disabled Transmission FIFO/Transmission buffer.

Bit 6-4 : TxFF_set_LV

Transmission FIFO water level.
000 b : Transmission FIFO data is less than × 2 (default)
001 b : Transmission FIFO data is less than × 4
010 b : Transmission FIFO data is less than × 8
011 b : Transmission FIFO data is less than × 12
100 b : Transmission FIFO data is less than × 14
101 b : Transmission FIFO data is less than × 8 (reserved)
110 b : Transmission FIFO data is less than × 8 (reserved)
111 b : Transmission FIFO data is less than × 8 (reserved)

This bit is used to set numbers of FIFO data which the Transmission FIFO water level matching event flag will be set.

Bit 3 : RxFF_EN

Reception FIFO enable flag.
0 : Reception FIFO Disable / Reception buffer Enable (default)
1 : Reception FIFO Enable / Reception buffer Disable

This bit is used to select either Reception FIFO or Reception buffer for use.
Data will not be transferred from the Reception shift register for disabled Reception FIFO/ Reception buffer.

Bit 2-0 : RxFF_set_LV

Reception FIFO water level.
000 b : Reception FIFO data is more than × 2 (default)
001 b : Reception FIFO data is more than × 4
010 b : Reception FIFO data is more than × 8
011 b : Reception FIFO data is more than × 12
100 b : Reception FIFO data is more than × 14
101 b : Reception FIFO data is more than × 8 (reserved)
110 b : Reception FIFO data is more than × 8 (reserved)
111 b : Reception FIFO data is more than × 1

This bit is used to set numbers of FIFO data which the Reception FIFO water level matching event flag will be set.

A READ command to bits 31 ~ 9 will always return “0” . WRITE command is invalid.

15-3-3-14 UART DMA Control Register : UDMA

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	DMATREQEN	DMARREQEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit 1 : DMATREQEN

DMA Transmission request output enable flag.

0 : DMA transmission request output Disable (default)

1 : DMA transmission request output Enable

○ Continuous transmission DMA request

(1) DMA request which uses Transmission buffer

When Transmission buffer is enabled and there is no valid data in the Transmission buffer, it will assert a DMA transmission request signal. DMA transmission request signal is negated by either having a valid data in the Transmission buffer or receiving CLEAR signal from DMA with request.

(2) DMA transmission request using Transmission FIFO

In Transmission FIFO enable state, when the amount of FIFO data is at the set level of Transmission FIFO water level of less, the DMA transmission request signal will be asserted. DMA transmission request signal is negated when the amount of data is exceeding the set point or with the Clear signal from DMA which received a DMA request.

Bit 0 : DMARREQEN

DMA Reception request output enable flag.

0 : DMA reception request output Disable (default)

1 : DMA reception request output Enable

○ Continuous reception DMA request

(1) DMA reception request using Reception buffer

In Reception buffer enable state, it will assert DMA reception request signal when there is a valid data in the Reception buffer. DMA reception request signal will be negated either by having no valid data in the Reception buffer or having no Clear signal from DMAC where DMA reception request was received.

(2) DMA reception request using Reception FIFO
In the Reception FIFO enable state, DMA reception request signal is asserted when the numbers of FIFO data is at the Reception FIFO water level set point or higher. DMA reception request signal will be negated if the amount of FIFO data becomes less than the set point of Reception FIFO water level or by the Clear signal from DMA which received DMA request; when the same amount of data as the water level set data is extracted.
When data is extracted at the same amount as the water level set data, it will stay in “assert” even if the numbers of FIFO data reaches the Reception FIFO water level set point or less.
If water level set amount of data is extracted and the numbers of FIFO data stays at the Reception FIFO water level set point or more, it will also remain as “assert”, and will extract the same amount of data (as the water level set data) again.

A READ command to bits 31 ~ 2 will always return “0”. WRITE command is invalid.

15-3-3-15 UART FIFO Status Register : USFS

[Base + 0x0038]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	TxFF_LV4	TxFF_LV3	TxFF_LV2	TxFF_LV1	TxFF_LV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	RxFF_LV4	RxFF_LV3	RxFF_LV2	RxFF_LV1	RxFF_LV0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit 12-8 : TxFF_LV
Transmission FIFO data volume status.
These bits indicates the numbers of valid data inside of the Transmission FIFO (0 ~ 16).
For example, if the READ value is 12, writing to USTF register 4 more times will fill up the FIFO.

Bit 4-0 : RxFF_LV
Reception FIFO data volume status.
These bits indicates the number of valid data inside of the Reception FIFO (0 ~ 16).
For example, if the READ value is 8, FIFO will become empty by reading out USRF register 8 times.

A READ command to bits 31~16, 15~13, and 7~5 will always return “0”.
WRITE command is invalid.

15-4 Operation

This section provides information about UARTn operation.

15-4-1 Start-stop synchronous operation mode

15-4-1-1 When Polarity control is set to “0”

Start-stop synchronization data format

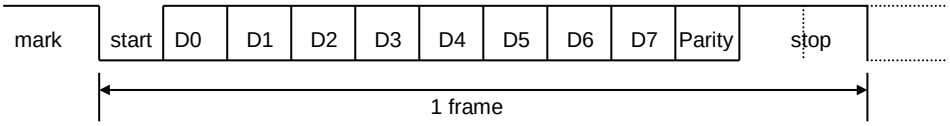


Figure 15-2

- Mark (HIGH level)

  TxD pin is at HIGH level (in marking state) when data is not being transferred.

- Start bit (LOW level)

  LOW level of the 1 bit length which indicates the start of a frame.

- Data bit (D0 ~ D7)

  This is used as an actual data for transmission and reception. It is set as either 7 bit or 8 bit.

- Parity bit

  When parity is permitted, Parity bit is added or checked to create odd parity (having odd numbers of bit with 1’s) and even parity (having even numbers of bit with 1’s) by considering both Data bit and Parity bit.

  When there is no parity, parity bit is not added or checked. Stop-bit will fill that space.

- Stop bit

  This indicates the end of a frame. This also indicates 1 bit or 2 bit length is “1”.

Communication format can be set to 24 different types in UMD (UART mode register). The following diagram shows 12 types of communication format when “0” is set for the Polarity control.

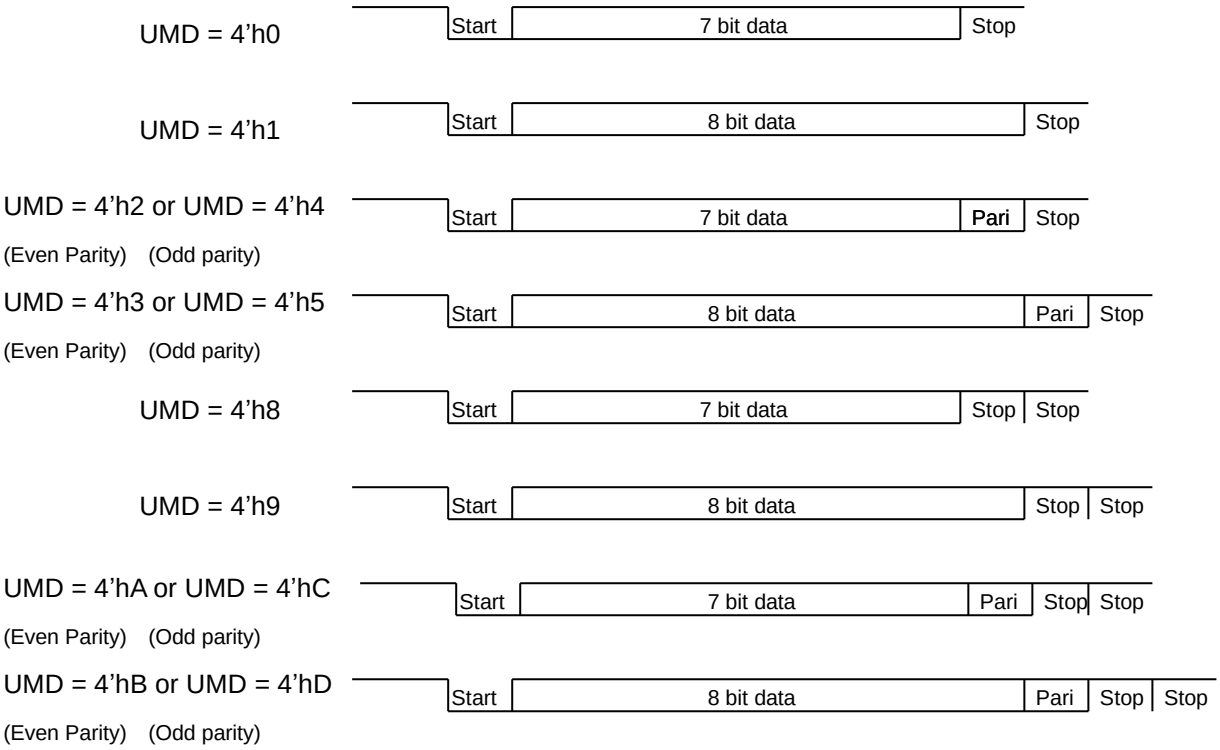


Figure 15-3

Attention) Odd parity and Even parity can be set to the areas where the parity setup is enabled.
UMD value in the diagram shows bit 3 ~ bit 0 setting in the UART mode register.

.

15-4-1-2 When Polarity control is set to “1”

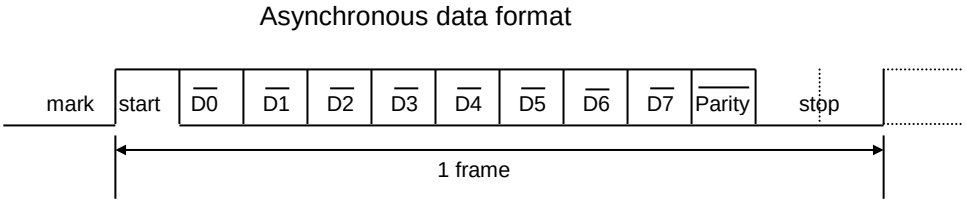


Figure 15-4

- Mark (LOW level)

  TxD pin is at LOW level (in marking state) when data is not being transferred.

- Start bit (HIGH level)

  HIGH level of the 1 bit length which indicates the start of a frame.

- Data bit (D0 ~ D7)

  This is used as an actual data for transmission and reception. It is set as either 7 bit or 8 bit.

- Parity bit

  When parity is permitted, Parity bit is added or checked to create odd parity (having odd numbers of bit with 0’s) and even parity (having even numbers of bit with 0’s) by considering both Data bit and Parity bit.

  When there is no parity, parity bit is not added or checked. Stop-bit will fill that space.

- Stop bit

  This indicates the end of a frame. This also indicates 1 bit or 2 bit length is “0”.

Communication format can be set to 24 different types in UMD (UART mode register). The following diagram shows 12 types of communication format when “1” is set for the Polarity control.

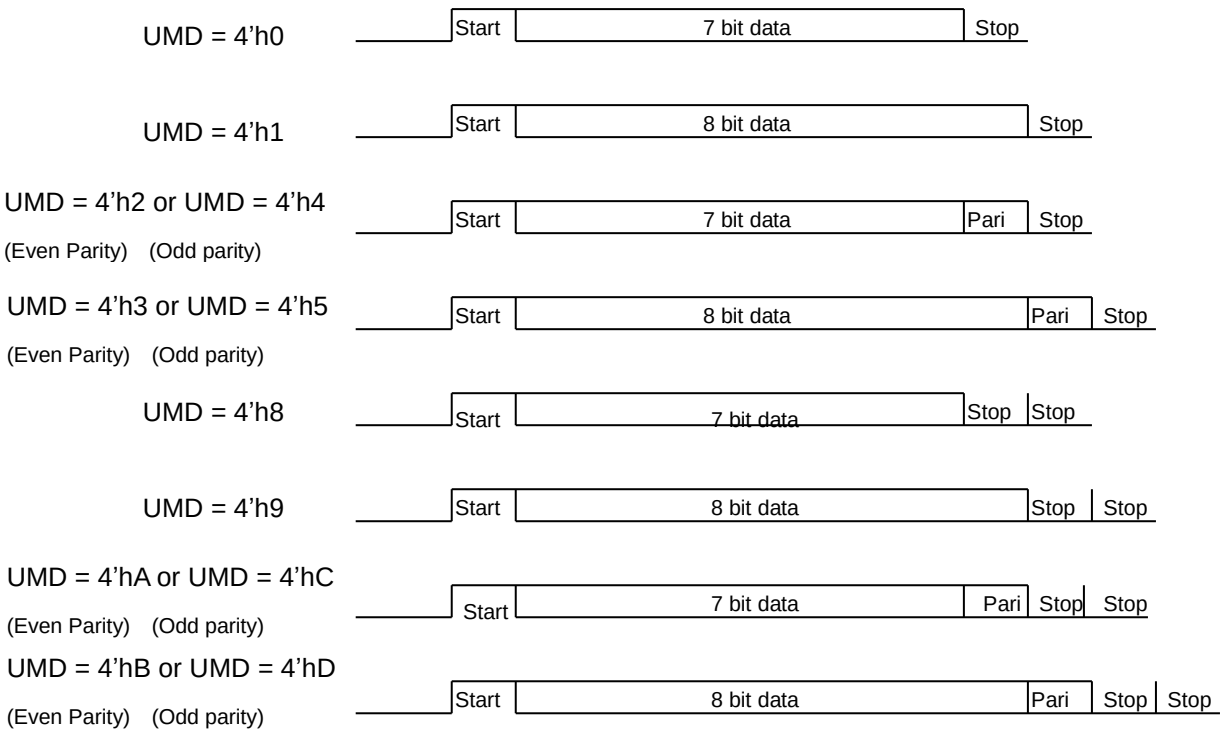


Figure 15-5

Attention) Odd parity and Even parity can be set to the areas where the parity setup is enabled.

UMD value in the diagram shows bit 3 ~ bit 0 setting in the UART mode register.

15-4-2 Transmission/Reception Timing

Determine a communication format by selecting a number of bit / Parity bit / a number of Stop bit / Polarity control status in a frame of the UMD (UART mode register). For data transmission, writing a data into UTR (UART transmission register) after setting TE (Transmission enable flag) of the UCM (UART command register) will start the data transmission operation. For data reception, detecting a Start bit of RXD pin after setting RE (Reception enable flag) of the UCM (UART command register) will start the data reception operation. Those transmission/reception timing are explained more from the next section. At this point, if the UMD (UART mode register) PO (Polarity control selection bit) has “0”, Start bit is set to LOW and Stop bit is set to HIGH. But if the UMD (UART mode register) PO (Polarity control selection bit) has “1”, Start bit is set to HIGH and Stop bit is set to LOW.

Figure 15-6 shows the transmission timing of PO=0, and Figure 15-7 shows the same timing when PO=1. Figure 15-8 shows the reception timing of PO=0 and Figure 15-9 shows the same timing when PO=1.

In this section, communication format is set as follows:

Data length = 8 bit

Parity = With Parity

Stop bit = 2 bit

CTS/RTS function = Not selected.

15-4-2-1 Transmission Operation

In order to start the transmission operation, a data has to be written into UTR (UART transmission register) after setting TE (Transmission enable flag) of the UCM (UART command register) <(*1) shown in the figure below>. Once it is complete, the TFF (Transmission register full flag) of the USR (UART status register) is set <(*2) in the figure>. When UTR data is transferred to a Transmission shift register, TFF will be cleared <(*3) in the figure>. Then, once transmission is completed for a frame, UARTTF (Transmission complete interrupt factor flag) will be set for UISR (UART interrupt status register) <(*4) in the figure>.

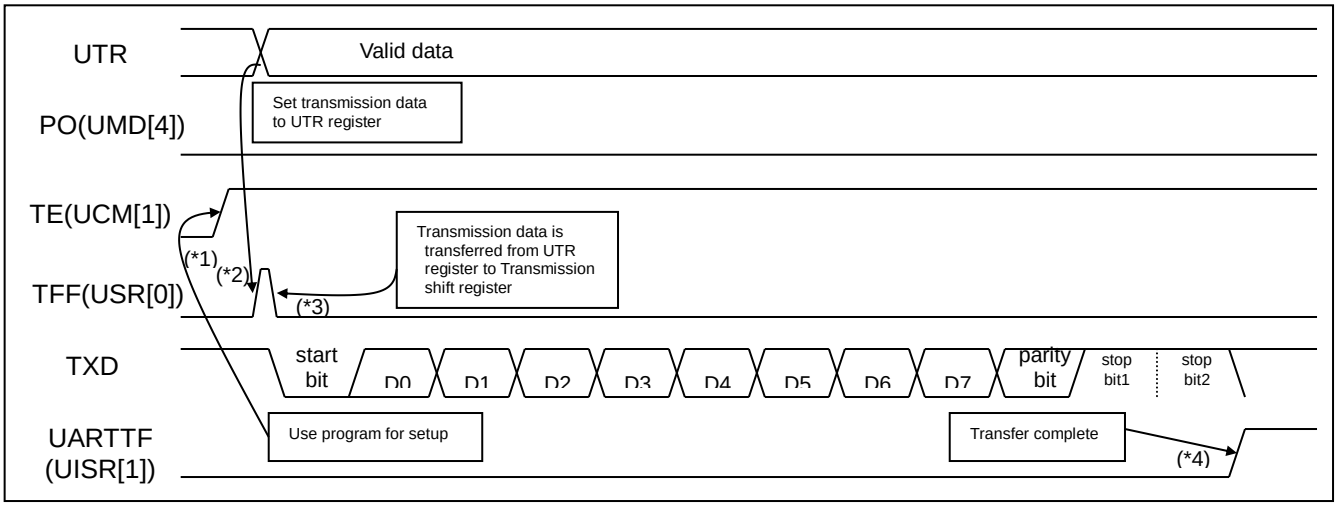


Figure 15-6 Transmission transfer at LOW Polarity control

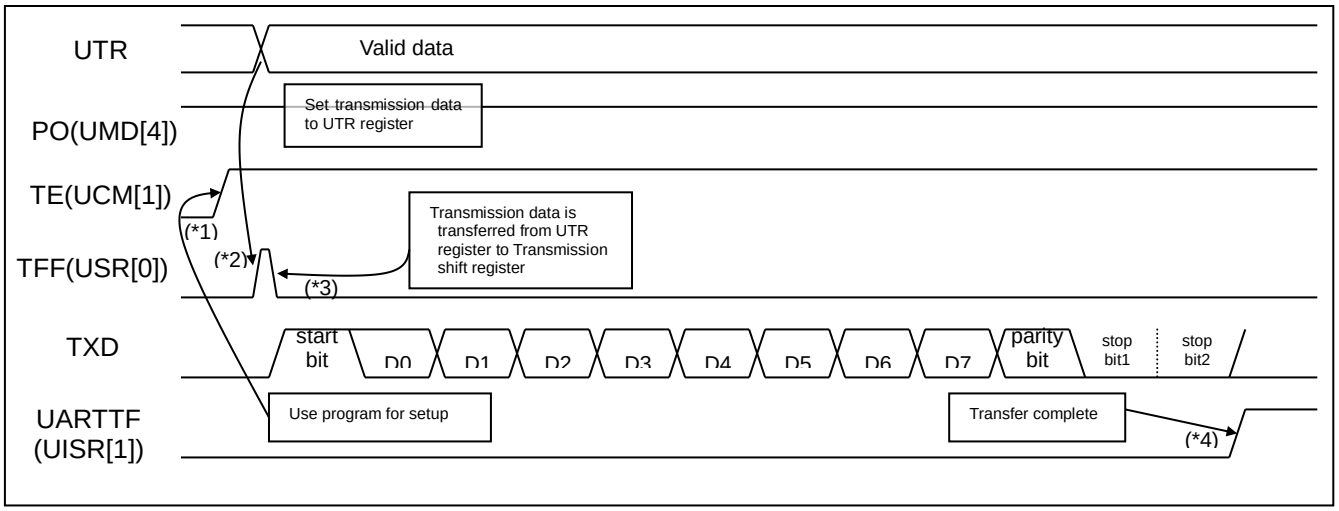


Figure 15-7 Transmission transfer at HIGH Polarity control

15-4-2-2 Reception Operation

Reception operation will begin when LOW level is inputted from RXD pin after UCM (UART command register) RE (Reception enable flag) is being set and Start bit is being detected <(*1) in the figure below>. Then, when data reception for a frame is complete, Reception shift register data is transferred to URR (UART reception register). RRF (Reception register full flag) of the USR (UART status register) <(*2) in the figure> and UARTRF (Reception complete interrupt factor flag) of UISR (UART interrupt status register) <(*3) in the figure> are also being set. RFF will be cleared when UART reception register data is read <(*4) in the figure>.

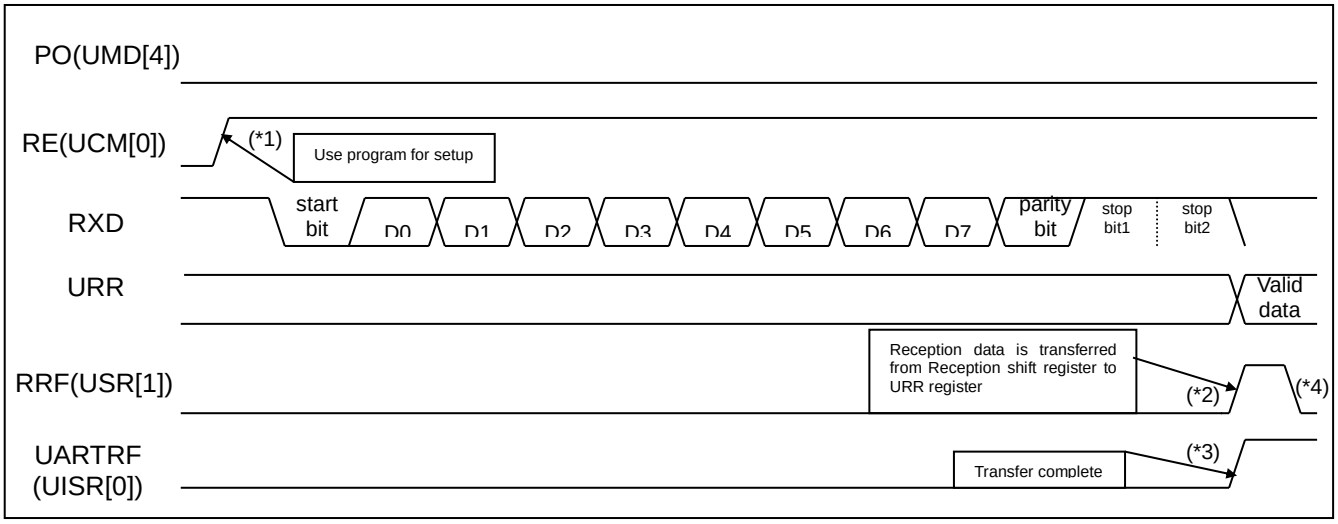


Figure 15-8 Reception transfer at LOW Polarity control

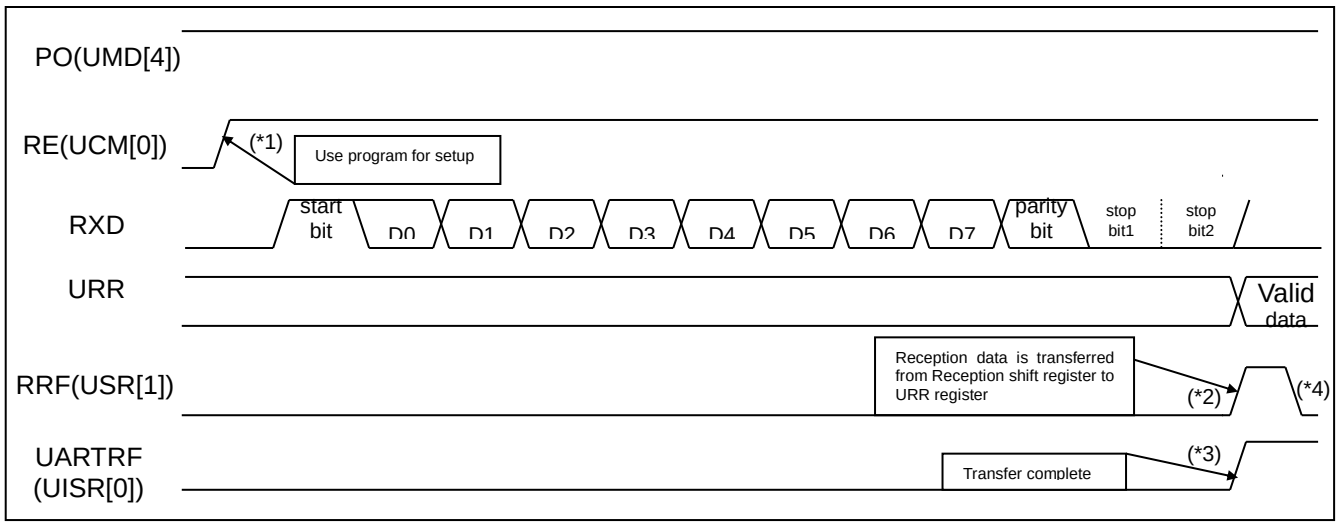


Figure 15-9 Reception transfer at HIGH Polarity control

15-4-3 Continuous Transmission Timing

This UART has a full-duplex double buffer structure which enables continuous transfer.
Two different timing patterns for continuous transfer are explained in the next two sections.
Figure 15-10 shows a timing where a transmission data is written into UTR (UART transmission register) before the previous data transmission is complete. Figure 15-11 shows timing where a transmission data is written into UTR after its previous data is transmitted completely. The communication format condition for these cases is; Data length=8 bit, with Parity, Stop bit=1 bit, and Polarity control selection bit=0.

15-4-3-1 When writing a transmission data before completing the previous transmission

To start transmission operation, write a data into UTR (UART transmission register) after setting UCM (UART command register) TE (Transmission enable flag) <(*1) in the figure below>. This will set TFF (Transmission register flag) <(*2) in the figure>. TFF will be cleared when the UTR data is transferred to Transmission shift register <(*3) in the figure>. TFF will be set when transmission data is written into UTR during the first transmission process <(*4) in the figure>. When the transmission is complete for a frame, the next transmission data will be transferred from UTR to a transmission shift register, and TFF will be cleared <(*5) in the figure>.

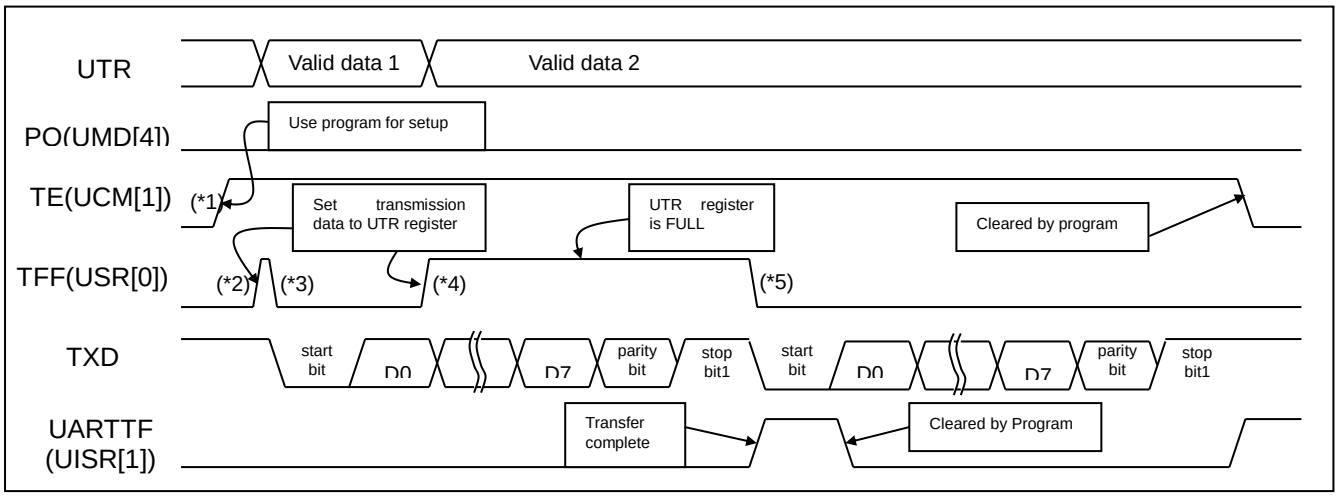


Figure 15-10 When data is written into UART register before the completion of previous transmission process.

15-4-3-2 When a transmission data is written into UTR after its previous data is transmitted completely.

The condition is same as what is explained in Figure 15-10, up to (*3). In this continuous transmission operation, UARTTF (Transmission complete interrupt factor flag) will be set once the transmission is complete for the first 1 frame <(*4) in the figure below>, then the next transmission data will be written into the UTR. When this happens, TFF will be reset <(*5) in the figure>. TFF will be cleared once the UTR data is transferred to the transmission shift register and data transfer will restart again <(*6) in the figure>

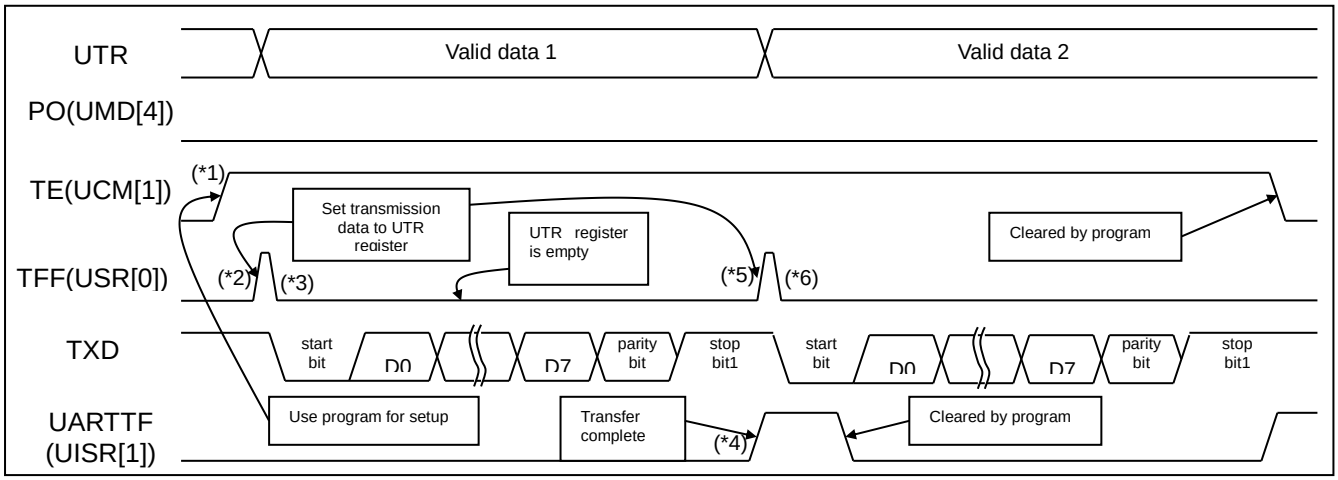


Figure 15-11 Transmission transfer of a frame

15-4-4 Break Transmission and Reception Detection

This UART can detect Break transmission and Break reception.

When executing Break transmission, setting “1” to SBRK (Break Transmission Control) will make TXD pin output a LOW level. The time of a frame can be measured by transmitting a dummy data.

When executing Break reception detection, BKD (Break reception) can be used for the judgement. When this bit is set to “1”, it indicates that the LOW level of the reception data is continuing for more than the reception period of a frame (when the Polarity select bit is set to “0”). Figure 19-9 shows a case of executing break transmission of a frame, and Figure 15-13 shows a case of detecting a break reception. The communication format used here are; Data length=8 bit, With Parity, Stop bit=2, and Polarity control selection bit=0.

15-4-4-1 Break Transmission of a frame1

To execute Break transmission of a frame, set TE (Transmission enable flag) and then set SBRK (Break transmission set flag) when TFF (UART transmission register full flag) is “0”. ((*1) in the figure below). Writing a dummy data into UTR (UART transmission register) after the first step will set TFF ((*2) in the figure). Once UTR data is transferred to a Transmission shift register, TFF will be cleared ((*3) in the figure). While polling UARTTF (Transmission complete interrupt factor flag) and when UARTTF is being set ((*4) in the figure), using software to clear SBRK ((*5) in the figure) will execute the break transmission of a frame.

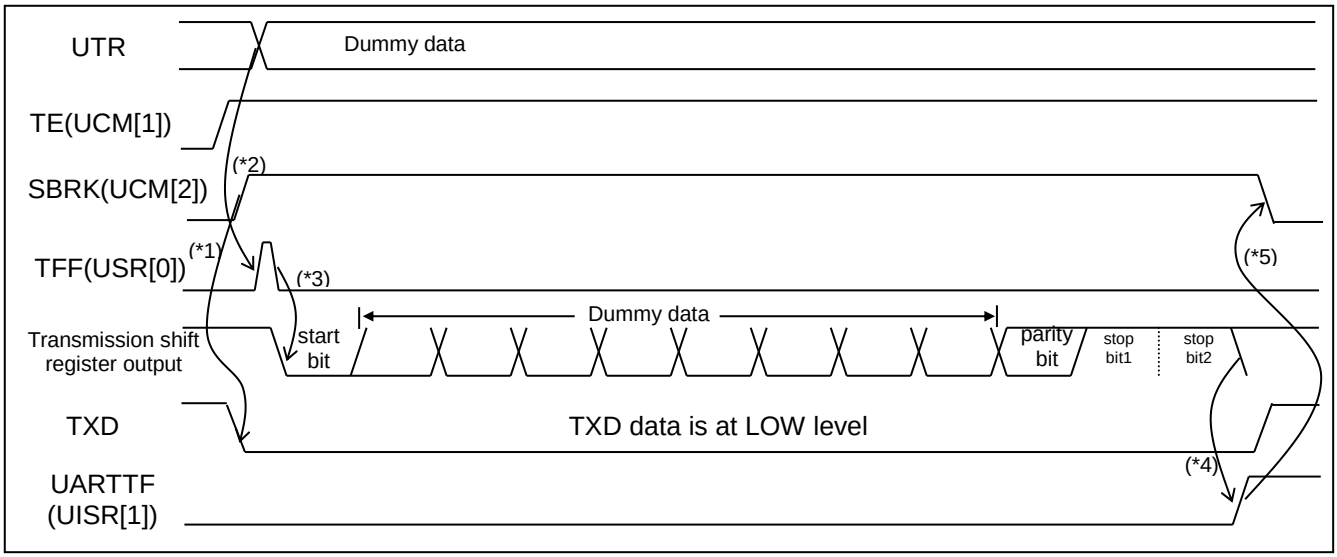


Figure 15-12 Break transmission at LOW polarity control

15-4-4-2 Break Reception Detection

When Polarity control selection bit is set to “0”, if a LOW level data of a frame is received in the Reception shift register, BKD (Break reception) is set at the data transfer timing from the Reception shift register to URR (UART reception register) ((*1) in the figure below). RRF is will be set at this point. FE (frame error) will not be set ((*2) in the figure). FF data will be stored in the URR in this case <(*3) in the figure>.

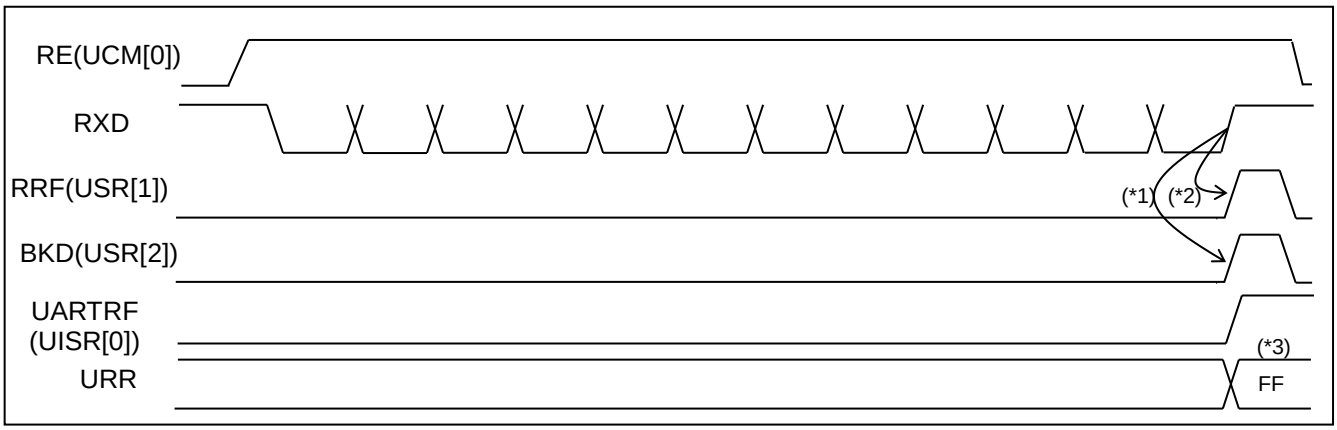


Figure 15-13 Break reception detection at LOW polarity control

15-4-5 Reception Register Overwrite Error Detection Function

After completing data reception, the Reception register overwrite error will be detected if the current data is transferred to URR (UART reception register) with having the previous reception data in the USR (UART status register); meaning, when Reception register full flag is set to “1”. Figure 15-14 shows the ROWE (Reception register overwrite error flag) set timing. The communication format used here are; Data length=8 bit, with Parity, and Stop bit=1 bit.

15-4-5-1 Reception Register Overwrite Error Occurrence Timing

In the reception operation, a reception data is transferred from the Reception shift register to URR ((*1) in the figure below) when the first data reception process is complete. RFF (Reception register full flag) will be set at this point <(*2) in the figure>. In the state of having stored data in the URR (when RFF is set to “1”), the reception register overwrite error will be set when the next data reception is complete and the reception shift register data is transferred to the URR <(*3), (*4), (*5) in the figure>. The data to be stored into this URR will not the current data but the following/next reception data.

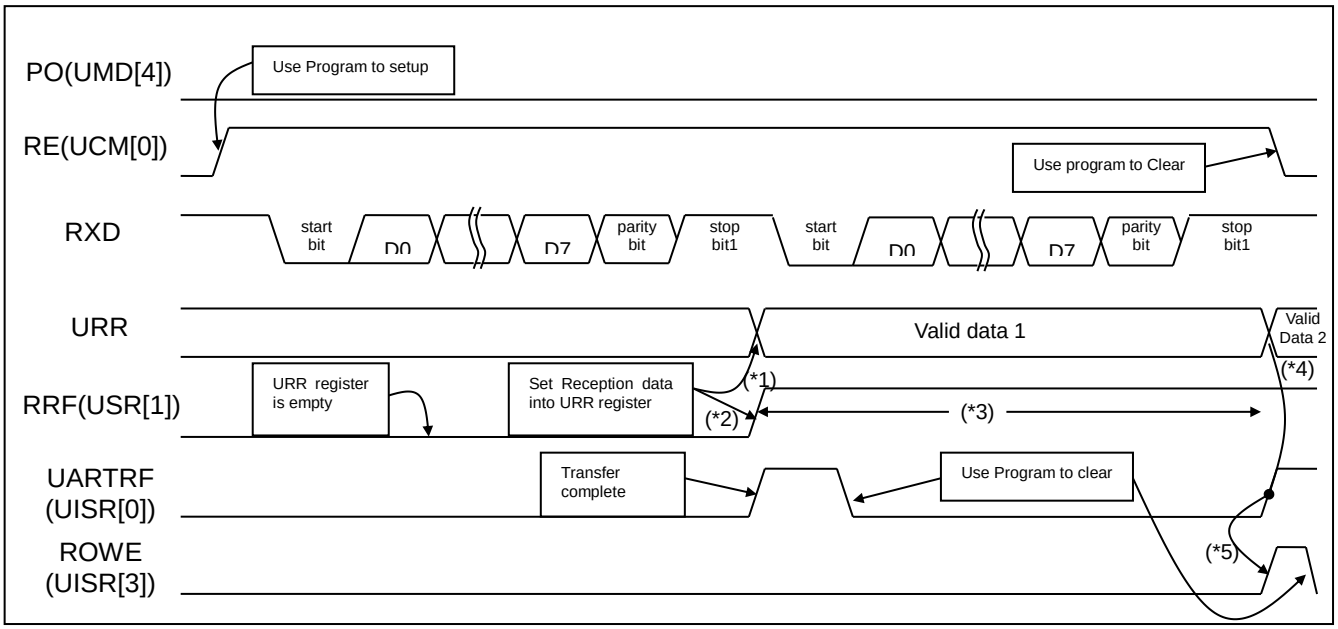


Figure 15-14 Reception register overwrite error detection

15-4-6 Parity Error Detection Function

Parity error in the reception operation is detected when Odd/Even parity is selected in the UMD (UART mode register). Parity is not detected when “No parity” is selected.
Figure 15-15 shows a case of detecting a parity error at PO (Polarity control selection bit)=0 in the UMD. Figure 15-16 shows a case of detecting a parity error at PO=1. The following communication format is; Data length=8 bit, with Odd parity, and Stop bit=2 bit.

15-4-6-1 Parity Error Detection at PO=0 (When Odd parity is selected)

In UMD, if PO=0 and an ODD parity is selected in the Parity selection bit, the PE (parity error) will be set if there are even number of 1s in between Data bit and the Parity. PE will be set when the data reception is complete for a frame <(*1) in the figure below>. RFF (Reception register full flag) will also be set at this time <(*2) in the figure>.

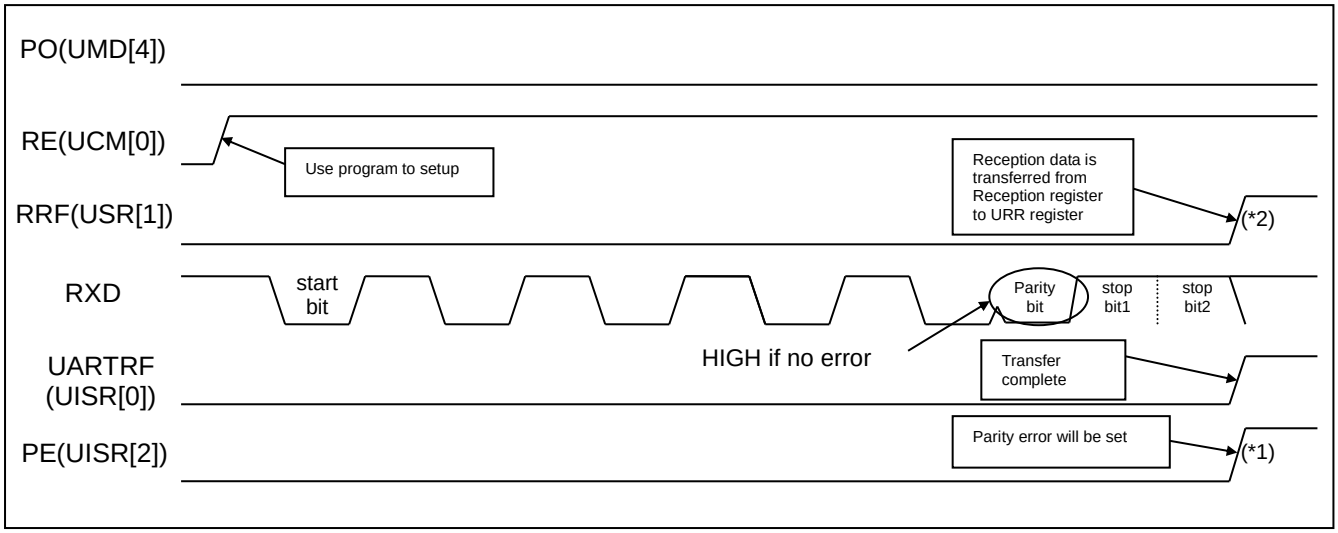


Figure 15-15 Parity error detection at reception transfer (with normal data polarity)

15-4-6-2 Parity Error Detection at PO=1 (When Odd parity is selected)

In UMD, if PO=1 and an ODD parity is selected in the Parity selection bit, the PE (parity error) will be set if there are even number of 0s in between Data bit and the Parity. PE will be set when the data reception is complete for a frame <(*1) in the figure below>. RFF will also be set at this time <(*2) in the figure>.

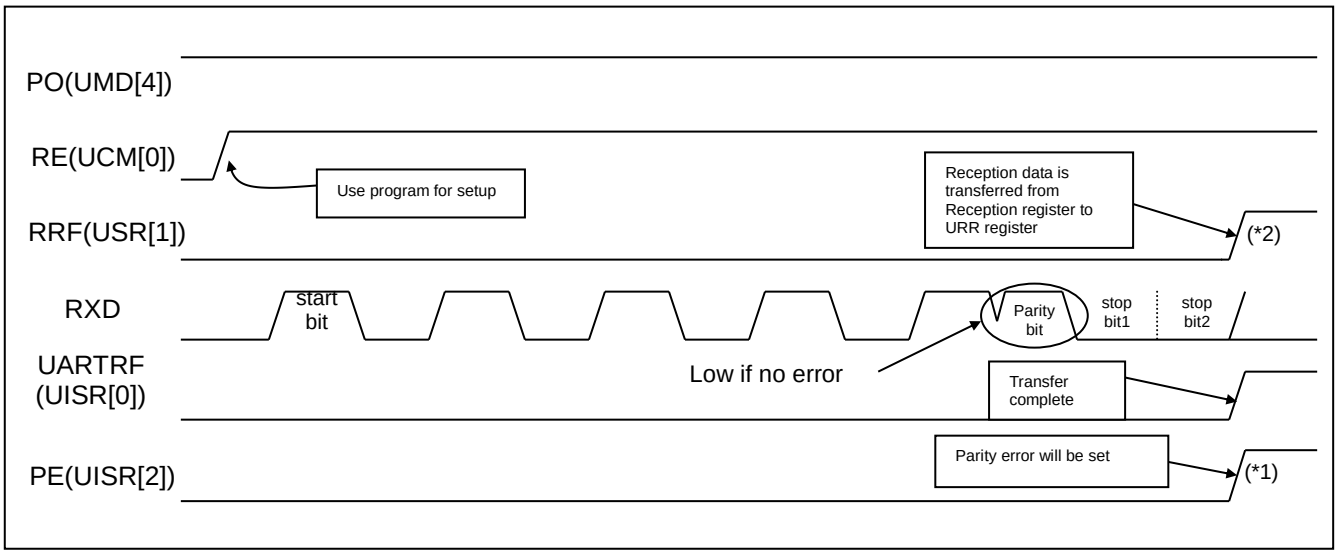


Figure 15-16 Parity error detection at reception transfer (with data polarity inversion)

15-4-7 Framing Error Detection Function

In a data reception, FE (framing error) is detected when a Stop bit is unable to detect at the proper timing. Also, The Stop bit polarity is “1” when PO=0 and the Stop bit polarity is “0” when PO=1 in the UMD.

Figure 15-17 shows a case of detecting the framing error when PO=0 in the UMD. Figure 15-18 shows a case of detecting the framing error when PO=1 in the UMD.

The communication format used here is : Data length=8 bit, with Odd parity, and Stop bit=2.

15-4-7-1 Framing Error Detection at PO=0

In a data reception, FE will be set if a Stop bit is not detected at its proper timing within the reception data of a frame, and it will be set at the time of completing data reception of a frame. <(*1) in the figure below>. RFF will also be set at this time <(*2) in the figure>.

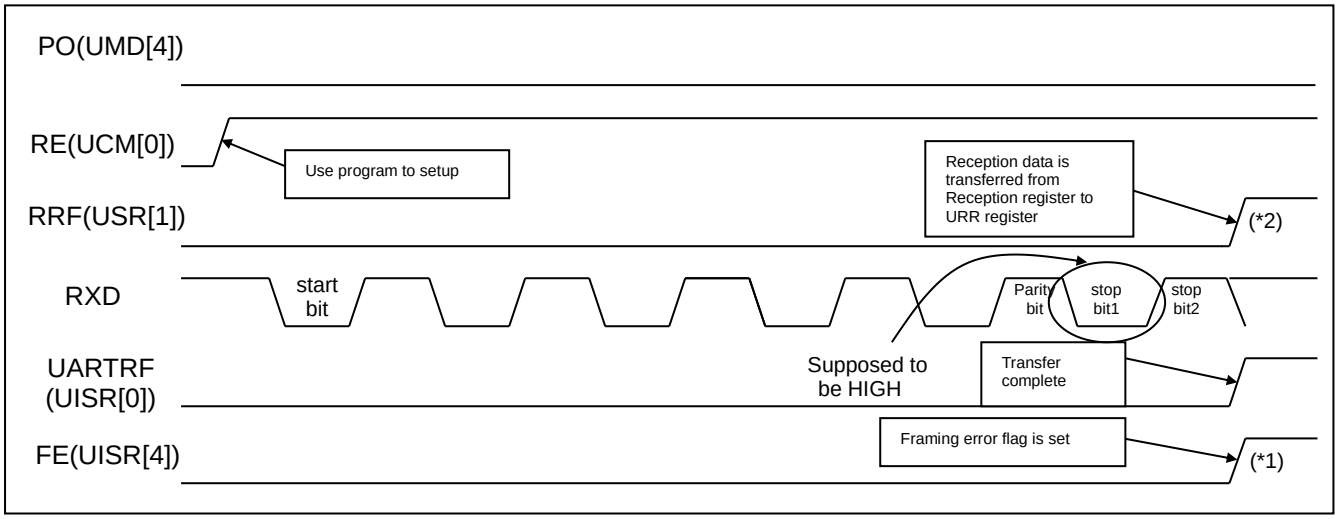


Figure 15-17 Framing error detection at data reception (with normal data polarity)

15-4-7-2 Framing Error Detection at PO=1

In a data reception, FE will be set if a Stop bit is not detected at its proper timing within the reception data of a frame, and it will be set at the time of completing data reception of a frame. <(*1) in the figure below>. RFF will also be set at this time <(*2) in the figure>.

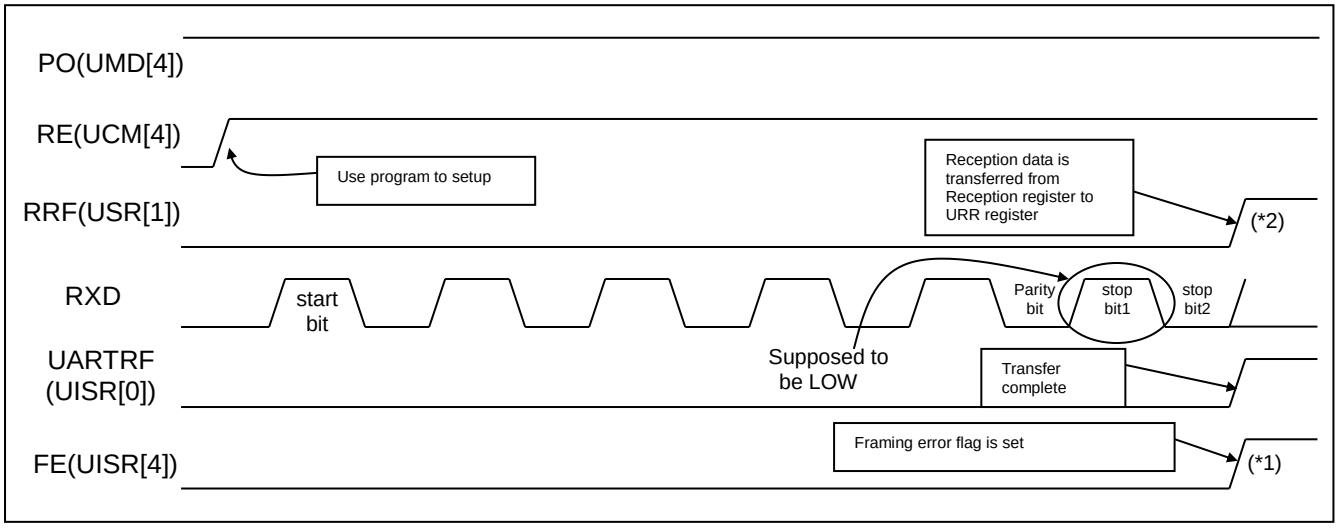


Figure 15-18 Framing error detection at data reception (with data polarity inversion)

15-4-8 Transmission Timing at Active CTS Function

When CTS function is active, transmission operation can be controlled by the input status of CTS_N pin.
CTS_N pin at LOW level (HIGH with polarity inversion): Status is active and data is transmittable.
CTS_N pin at HIGH level (LOW with polarity inversion): Status is inactive and data is not transmittable.

Two timing patterns with active CTS_N pin at the beginning of the data transmission are shown in the following figures:

Figure 15-19: Transmission transfer at LOW polarity control
Figure 15-20: Transmission transfer at HIGH polarity control

Two timing patterns with inactive CTS_N pin at the beginning of the data transmission are shown in the following figures:

Figure 15-21: Transmission transfer at LOW polarity control
Figure 15-22: Transmission transfer at HIGH polarity control

Two operation timing patterns at continuous transmission using CTS function are shown in the following figures:

Figure 15-24: Transmission transfer at LOW polarity control
Figure 15-25: Transmission transfer at HIGH polarity control

Communication format used for these is: Data length=8 bit, with parity, Stop bit=1 bit, PO=0 or 1, and CTS function selection bit=1.

15-4-8-1 Active CTS_N pin at the beginning of Data transmission

To begin the transmission operation, write a data into UTR (UART transmission register) after setting UCM TE (Transmission enable flag) <(*1) in the figure below>. When this action is taken, USR (UART status register) TFF (Transmission register full flag) will be set <(*2) in the figure>. TFF will be cleared when the UTR data is transferred into the Transmission shift register <(*3) in the figure>. Once the data transmission is complete for a frame, UARTTF (Transmission complete interrupt factor flag) of the UISR (UART interrupt status register) will be set <(*4) in the figure>.

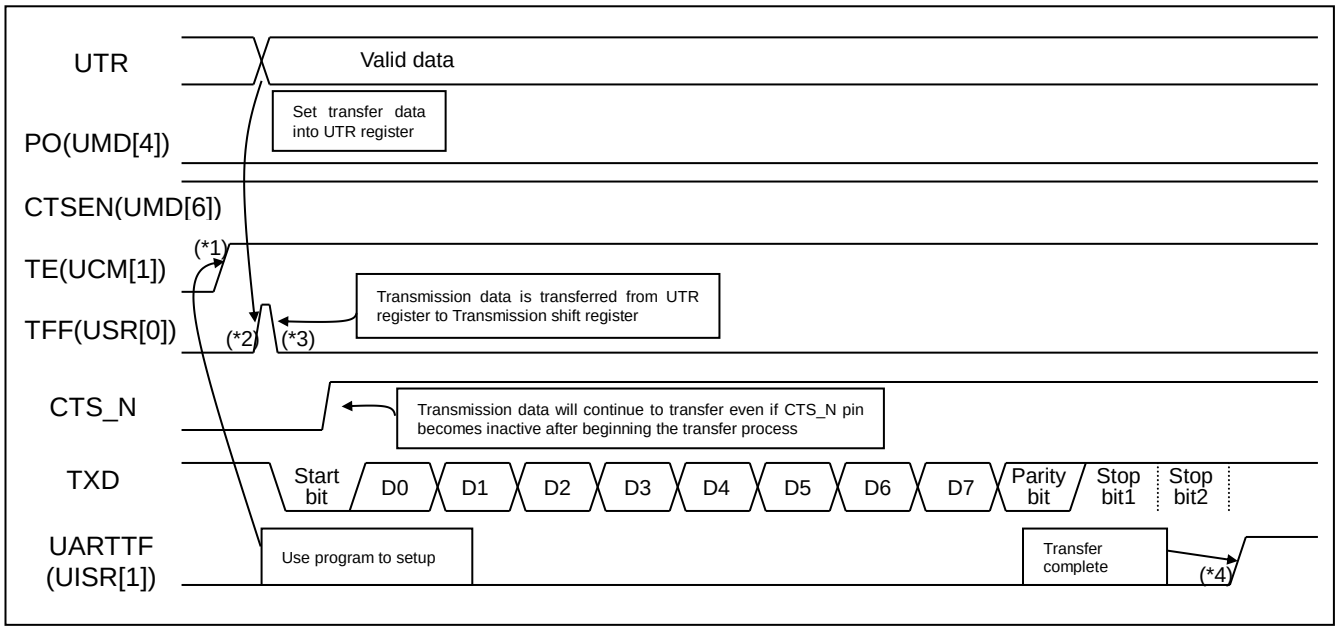


Figure 15-19 Transmission transfer at LOW polarity control

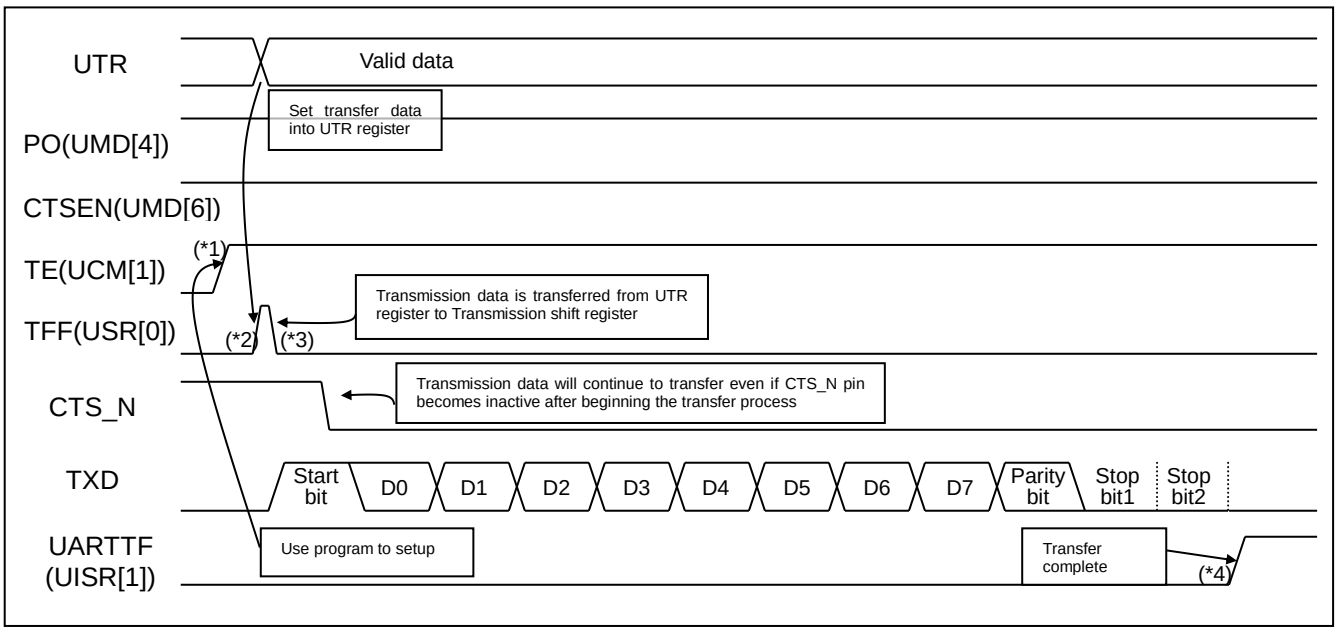


Figure 15-20 Transmission transfer at HIGH polarity control

15-4-8-2 Inactive CTS_N pin at the beginning of Data transmission

To begin the transmission operation, write a data into UTR (UART transmission register) after setting UCM TE (Transmission enable flag) <(*1) in the figure below>. When this action is taken, USR (UART status register) TFF (Transmission register full flag) will be set <(*2) in the figure>. The UART transmission operation will not start when CTS_N pin is inactive <(*3) in the figure>, Thus, when CTS_N pin status becomes active, TFF will be cleared when the UTR data is transferred into the Transmission shift register <(*4) in the figure>. Once the data transmission is complete for a frame, UARTTF (Transmission complete interrupt factor flag) of the UISR (UART interrupt status register) will be set <(*5) in the figure>.

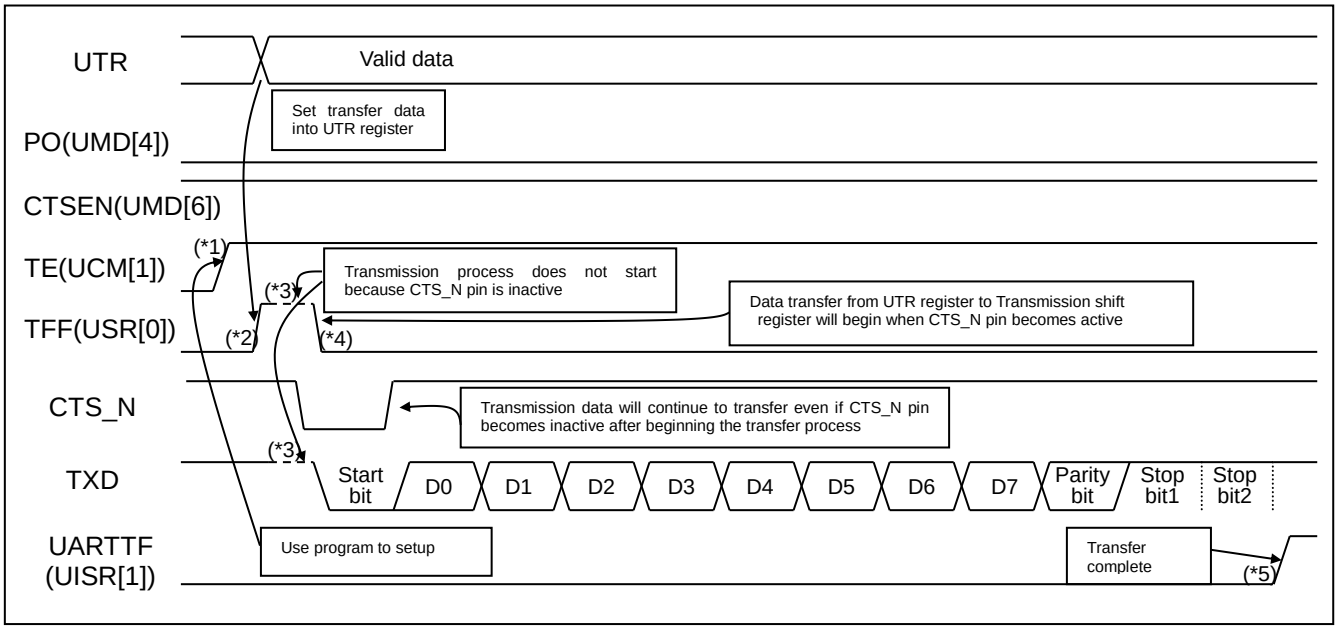


Figure 15-21 Transmission transfer at LOW polarity control

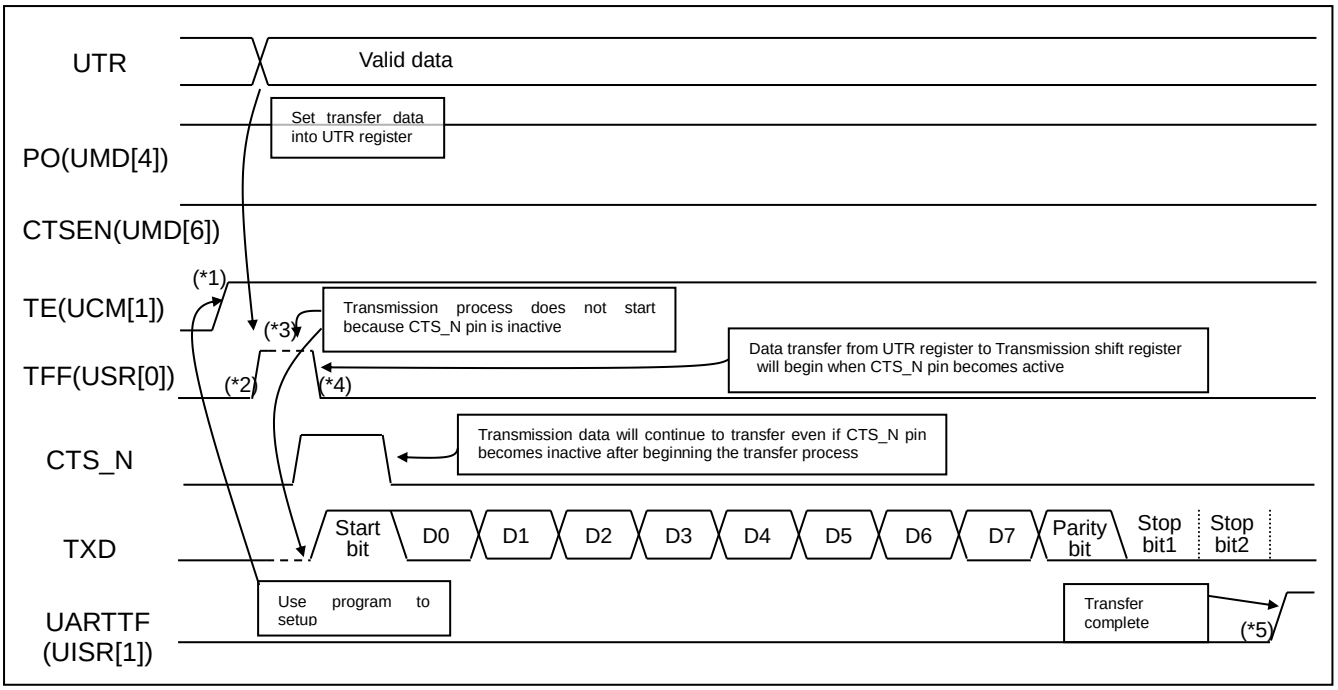


Figure 15-22 Transmission transfer at HIGH polarity control

15-4-8-3 Operation example at Continuous transmission using CTS function

This section provides an operation example of the continuous transmission at CTS_N pin in an active state at the beginning of the transmission which becomes inactive in the middle of the process.

To begin transmission operation, write a data into UTR after setting UCM (UART command register) TE (Transmission enable flag) <(*1) in the figure below>. When this action is taken, the TFF (Transmission register flag) will be set <(*2) in the figure>. When CTS_N pin is active, TFF will be cleared when UTR data is transferred into the Transmission shift register <(*3) in the figure>. Writing the following/next transmission data into UTR during the first transmission process will set a TFF <(*4) in the figure>. When CTS_N pin becomes inactive before completing the data transfer of a frame (before the timing of Tsetupcts at the Stop bit), the transmission process will be interrupted until CTS_N pin becomes active <(*5) in the figure>. Once the CTS_N pin becomes active again, the next transmission data will be transferred from UTR to Transmission shift register, TFF will be cleared, and the next transmission operation will begin <(*6) in the figure>.

It is necessary to set CTS_N pin to be inactive before Tsetupcts time or more at least from the end of the stop bit in the current one byte transfer in order to cancel the next one byte transfer by using CTS_N pin. Regarding Tsetupcts, please refer the datasheet.

Figure 15-23

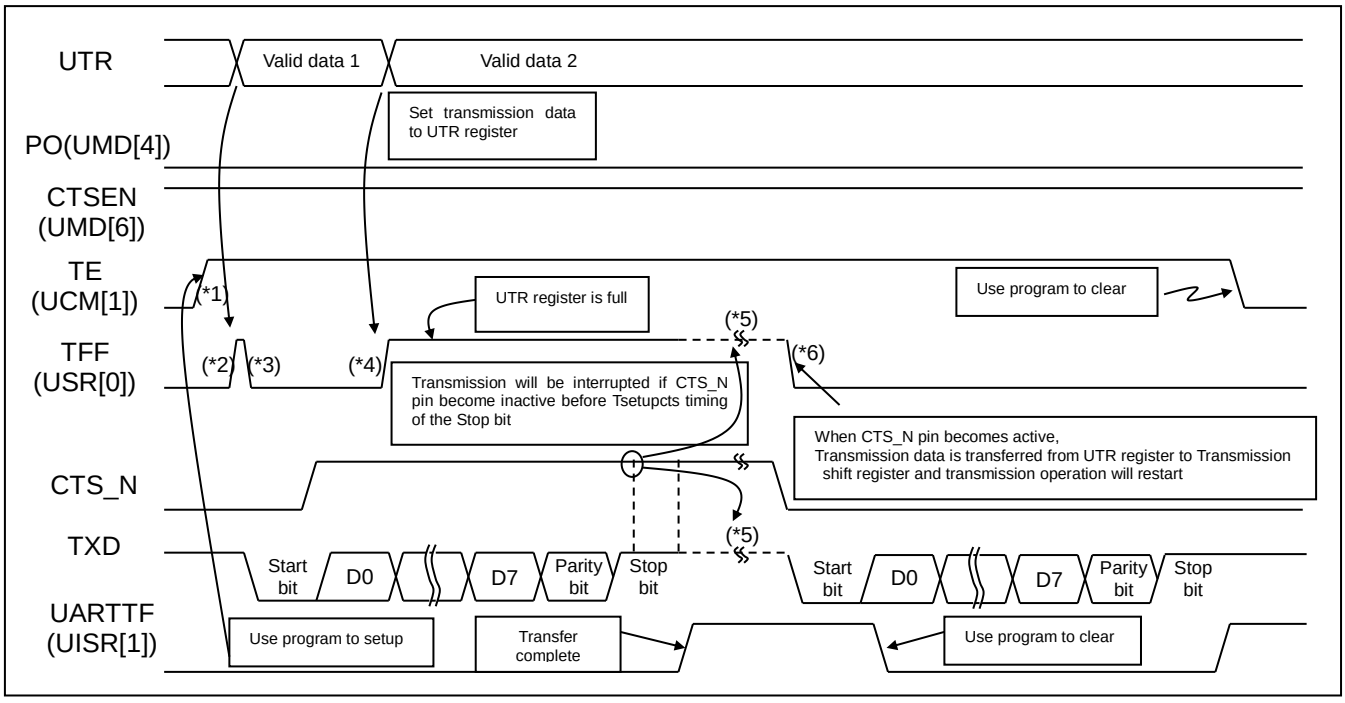


Figure 15-24 Example of Continuous transmission using CTS function (at LOW polarity control)

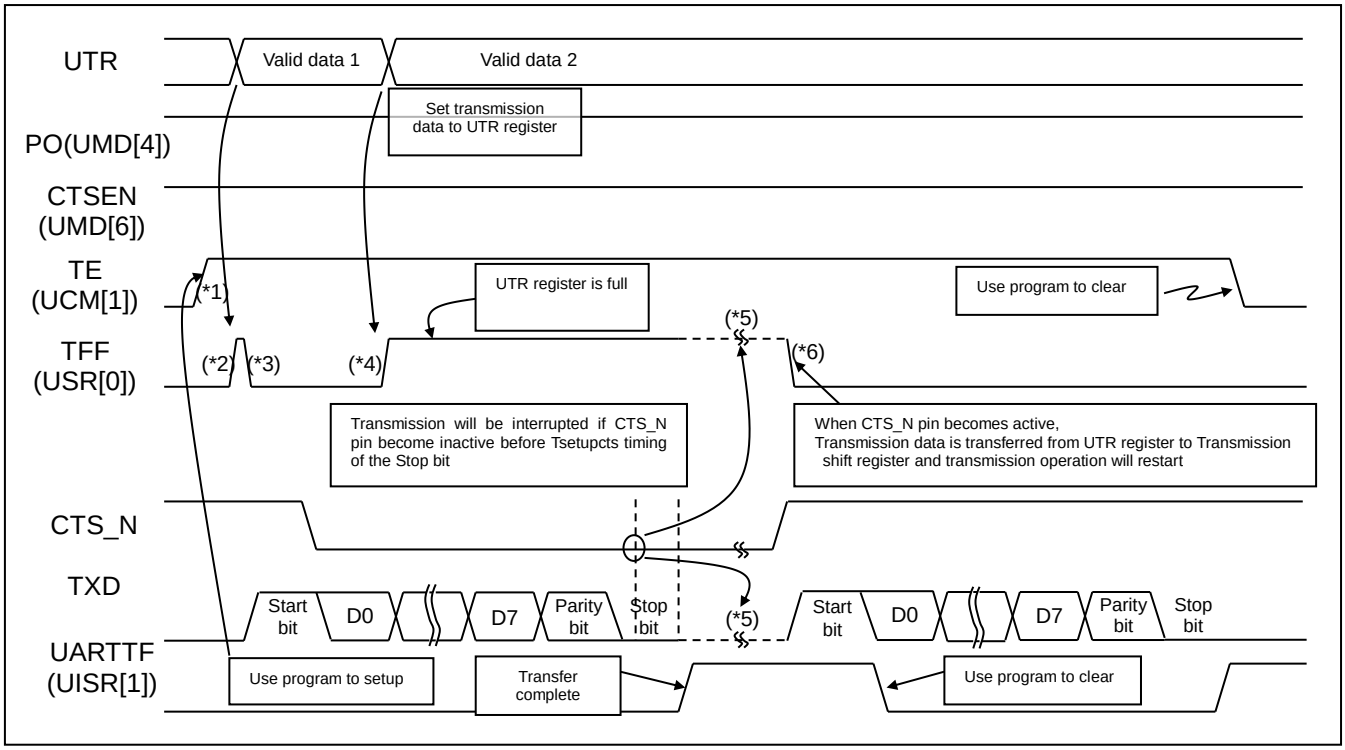


Figure 15-25 Example of Continuous transmission using CTS function (at HIGH polarity control)

Attention: Regarding behavior when Tsetupcts is not satisfied

Figure 15-26 shows that the next one byte data will be transmitted at the time of having prepared for transmission of it under the condition that Tsetupcts is not satisfied. In addition, in case of HIGH polarity control, the polarity of Both CTS_N pin and TXD pin is reversed.

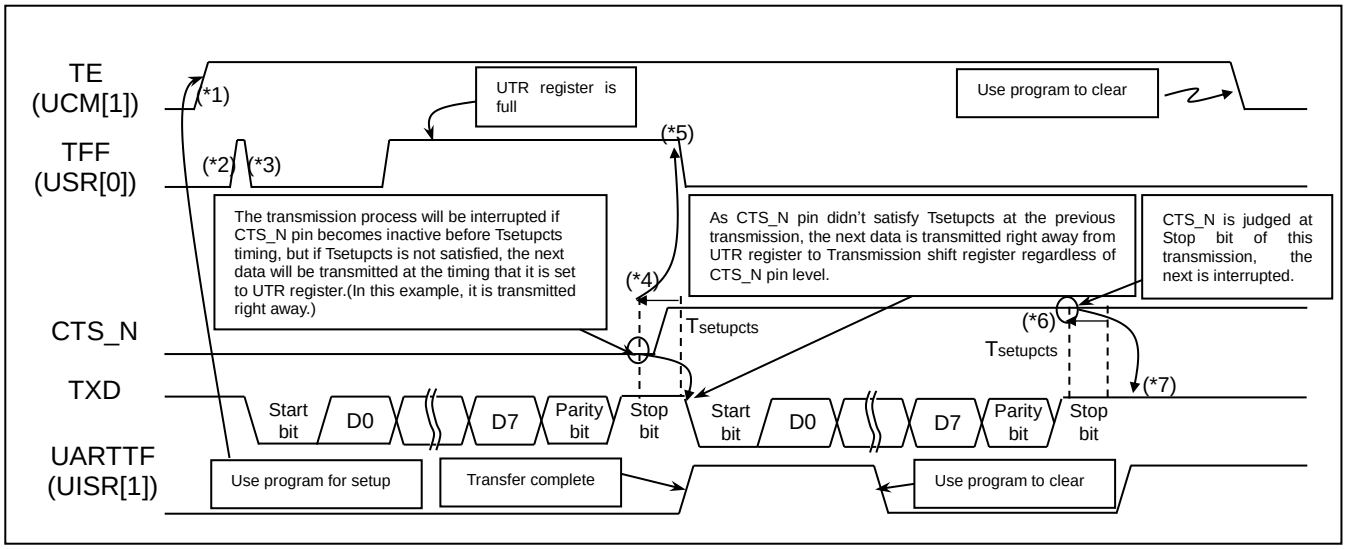


Figure 15-26 Example1 of transmission in CTS_N not satisfying Tsetupcts (at LOW polarity control)

In case that preparing for the next data is delayed, similar transmission will happen as well (Figure 15-27). Valid data2 is transmitted as next data as soon as it is set. In addition, in case of HIGH polarity control, the polarity of Both CTS_N pin and TXD pin is reversed.

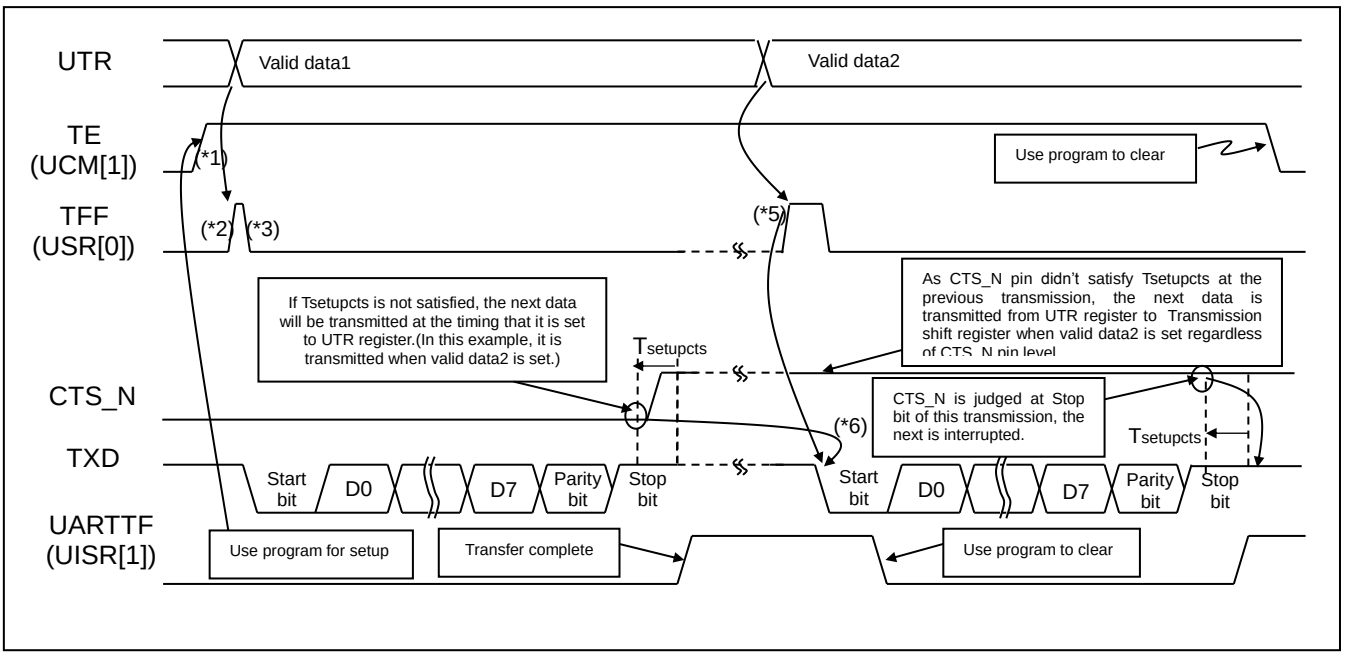


Figure 15-27 Example2 of transmission in CTS_N not satisfying Tsetupcts (at LOW polarity control)

While CTS_N pin is inactive after the current one byte has been transmitted, if TE(UCM[1]) is cleared to 0 once, it is set to 1 again, and the next one byte data is set to UTR register, the data for next can be transmitted by controlling CTS_N pin.

Figure 15-28 shows that the data for next can be transmitted by controlling CTS_N pin in clearing and setting TE bit under the condition of not satisfying Tsetupcts timing. In addition, in case of HIGH polarity control, the polarity of Both CTS_N pin and TXD pin is reversed.

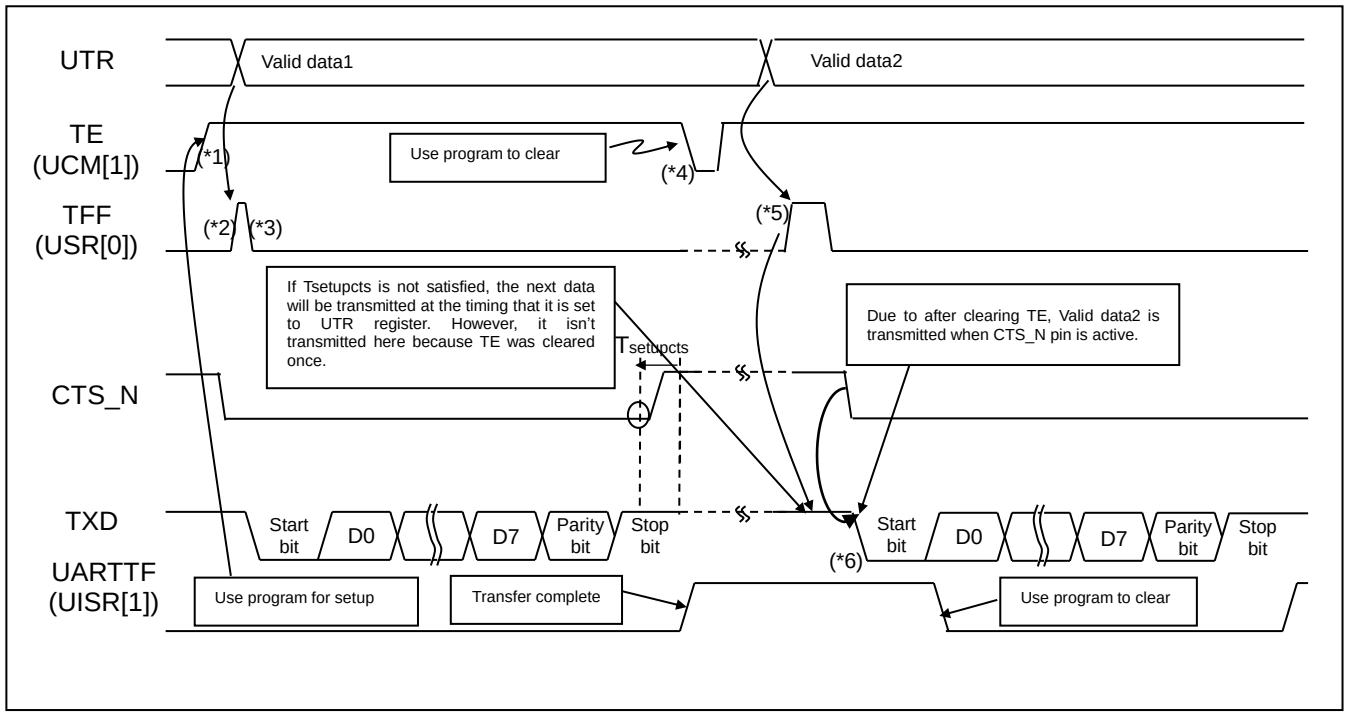


Figure 15-28 Example3 of transmission in CTS_N not satisfying Tsetupcts (at LOW polarity control)

15-4-9 Reception timing at Active RTS Function

When RTS function is active, reception operation can be controlled by controlling the RTS_N pin output

RTS_N pin at LOW level (HIGH with polarity inversion): Status is active and data is receivable.

RTS_N pin at HIGH level (LOW with polarity inversion): Status is inactive and data is not receivable.

Two reception transfer timing patterns with active RTS function are as follows:

Figure 15-29: Reception transfer at LOW polarity control

Figure 15-30: Reception transfer at HIGH polarity control

Communication format used for these is: Data length=8 bit, with parity, Stop bit=1 bit, PO=0 or 1, and RTS function selection bit=1.

15-4-9-1 Reception operation using RTS Function

Reception operation will begin when LOW level is inputted from RXD pin after setting UCM (UART command register) RE (Reception enable flag) and Start bit is being detected <(*1) in the figure below>. RTS_N pin status will change to inactive at 1/2 bit before the final data reception. Once the reception process is complete for a frame, the reception shift register data will be transferred to URR (UART reception register), and USR (UART status register) RRF (Reception register full flag) and UISR (UART interrupt status register) UARTTF (Reception complete interrupt factor flag) will also be set <(*3) and (*4) in the figure>. Reading the URR data will clear RFF and RTS_N pin will become active <(*5) in the figure>.

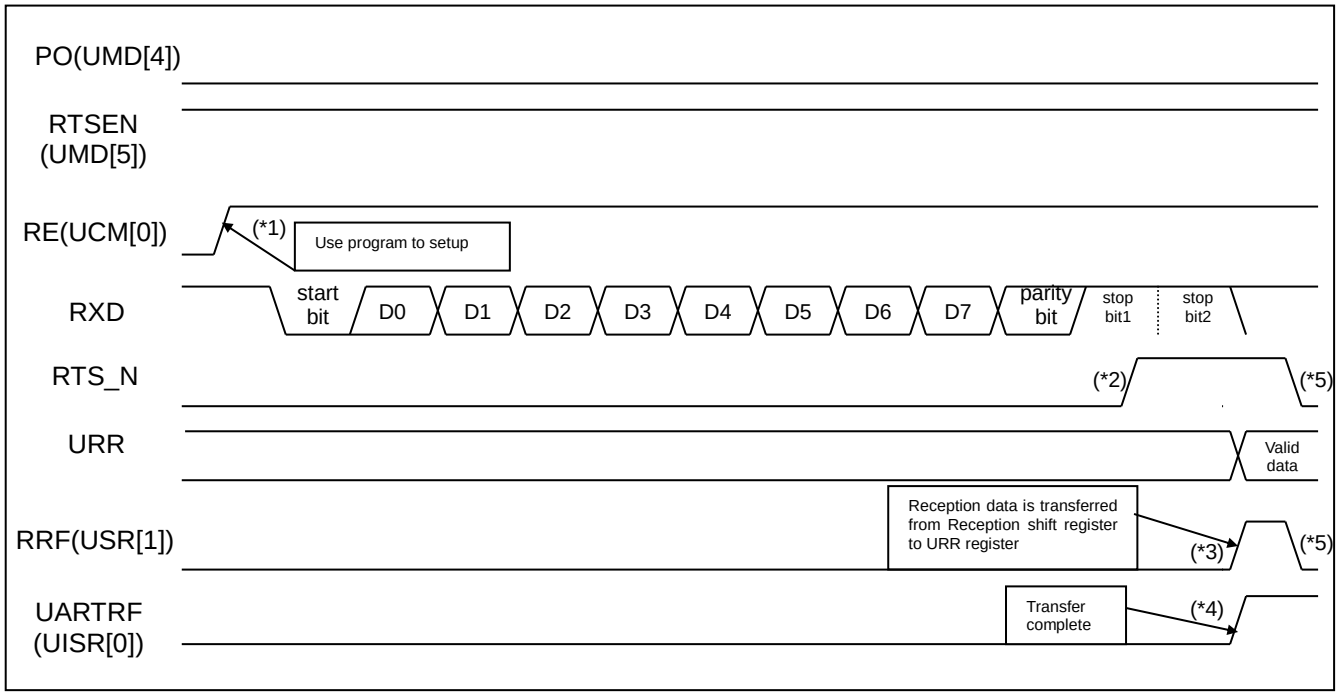


Figure 15-29 Reception transfer at LOW polarity control

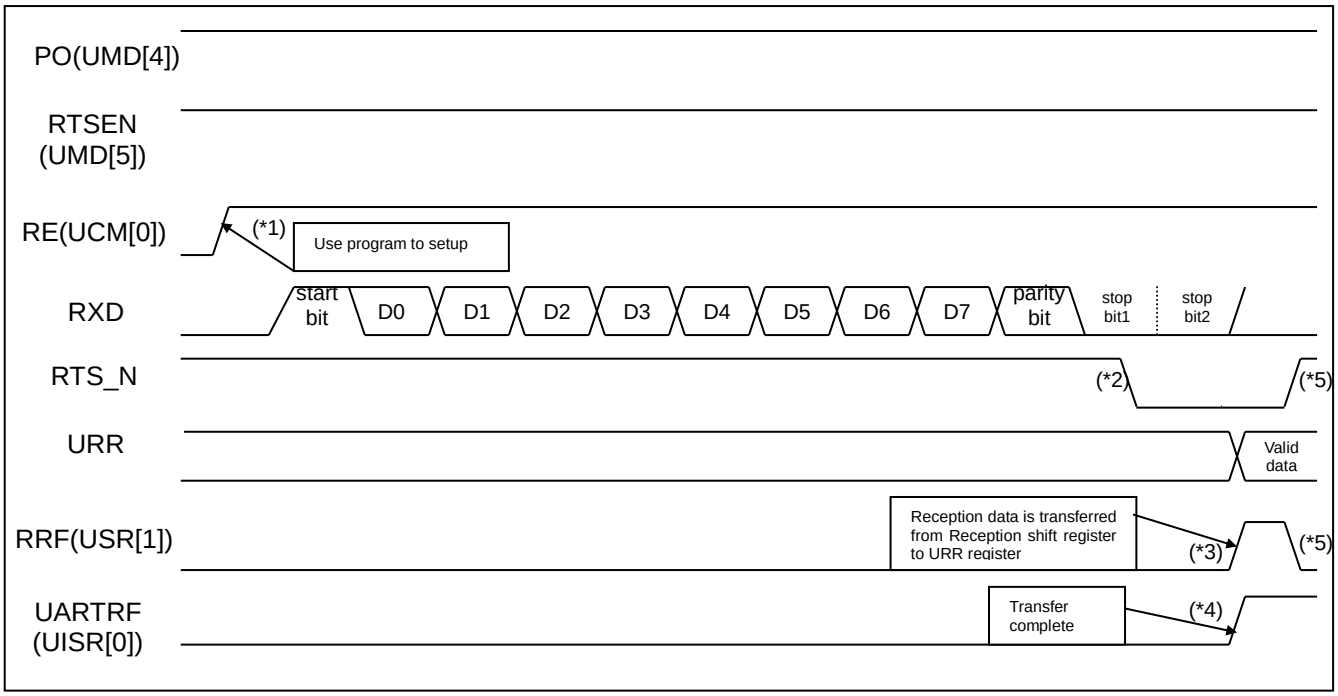


Figure 15-30 Reception transfer at HIGH polarity control

15-4-10 Asynchronous Serial Transmission Reception

15-4-10-1 Transmission data transfer control method

1. TXD pin is set at HIGH level (LOW level when polarity is “1”) in transmission disable state (excluding the break transmission timing).
2. Data length, Parity type, Stop bit, polarity control, and CTS function need to be selected in the UMD register.
3. In the UCM register, set “1” to the Transmission register clear flag to initialize Transmission register and Transmission shift register. Then, clear this flag by setting “0” before setting “1” to the Transmission enable flag.
4. Set divider values in UDIV register.
5. Set baud rate values in UBRG register.
6. Clear a flag in UISR register.
7. Confirm that the transmission register full flag is empty in USR register.
8. Set “1” to the Transmission enable flag in UCM register.
9. When transmission data is written into UTR register, that data will be transferred to a Shift register. In there, Start bit, Parity bit, and Stop bit are added to the data and it will come out from TXD pin as a serial data.

   When CTS function is valid, the transmission operation will begin when CTS_N pin is set to LOW level (HIGH level when the polarity is “1”).

   After data transmission is started, even if the CTS_N pin status changes to HIGH level (LOW level when the polarity is “1”), its transmission operation will continue until the data transmission is complete for a frame.
10. When a frame data is transmitted, Transmission transfer complete interrupt factor flag will be set.

Attention)

- Must make sure the UCM register’s Transmission enable flag is set in order to transmit a data to the UTR register.
- To execute continuous transmission, confirm that the transfer register full flag of USR register is empty before writing the next/following data to the transmission register.

15-4-10-2 Reception data Transfer control method

1. RXD pin must be set to HIGH level (LOW level if the polarity is “1”) when not receiving any data.
2. Data length, Parity type, Stop bit, Polarity control, and RTS function are selected in the UMD register.
3. Reception register and Reception shift register are initialized by setting “1” to the Reception register clear flag in the UCM register. Then, clear this flag by setting “0” before setting “1” to the Reception enable flag.
4. Set divider values in UDIV register.
5. Set baud rate values in UBRG register.
6. Clear flag in UISR register.
7. Set a port. See *Attention 3* below for more information.
8. Set “1” to Reception enable flag in UCM register.
9. When LOW level (or HIGH level if the polarity is “1”) is detected in the RXD pin, in order to confirm the validity of this Start bit, RXD input level will be sampled at a position that is 1/2 bit from the falling edge (rising edge if the polarity is “1”). If LOW level (HIGH level if the polarity is “1”) is detected at this point, it is considered as a valid Start bit and will begin the reception process. If HIGH level (LOW level if the polarity is “1”) is being detected, it will not consider as a valid Start bit and will wait for the LOW level input (HIGH level if the polarity is “1”).
10. When the Start bit is detected, Data bit, Parity bit, and Stop bit will be sampled for data reception.
11. When one frame worth of data is received, that data is transferred from a Shift register to URR, and USR register’s Reception register full flag will be set to “1”. Reception transfer complete interrupt factor flag will also be set.

    When RTS function is selected, RTS_N output pin status changes to HIGH level (LOW level when the polarity is “1”) right before the completion of the reception process (1/2 bit before the Stop bit reception) and data reception will be disabled. To enable the reception process, read the URR register and empty out the Reception buffer.
12. When USR register’s Reception register full flag is set to “1” and the next/following data is transferred to the URR, UISR register’s Reception register overwrite error flag will be set to “1”.
13. If the parity is valid, it will check the parity state and if an error is detected, the Parity error flag of UISR register will be set to “1”.
14. When HIGH level (LOW level if the polarity is “1”) is detected at a timing of where a Stop bit is supposed to be located, it will wait for the start of the next/following data. However, if the HIGH level (LOW level if the polarity is “1”) is not detected, UISR register’s Framing error flag will be set to “1”.

15-4-10-3 FIFO Control Method at Transmission transfer

1. Follow Steps 1~6 of section 4-10-1 to setup a register.
2. Set Transmission FIFO water level and FIFO enable in the USFC register.
3. In case of using DMA, use USTF as a destination address. If it is hardware controlled setup the status to use DMACTREQ.
4. Set “1” to Transmission enable flag in the UCM register.
5. In case of using DMA, the transmission data is written into USTF register by DMA operation because DMACTREQ will be generated. When DMA is not used, data will be written by the program. Data will be transferred to a Shift register; Start bit, Parity bit, and Stop bit will be added, and the data will be transferred from TXD pin as a serial data.
6. Use DMACTREQ, TxWLM transmission FIFO water level match flag, and TxWLM_INT transmission FIFO water level matching interrupt factor to write transmission data to reach the FIFO water level. When using TxWLM_INT for the first time, the Factor flag is set because FIFO is empty. Interrupt output will occur from the moment where the TxWLM_IEN interrupt permission is validated.
7. When a data of a frame is transmitted, Transmission transfer complete interrupt factor flag will be set.
8. TxFF_LV status register can provide information such as the amount of data in FIFO to the overflow level and the numbers of remaining transfer frame.
   For example, if the register value is 12, data can be write into FIFO 4 more times.

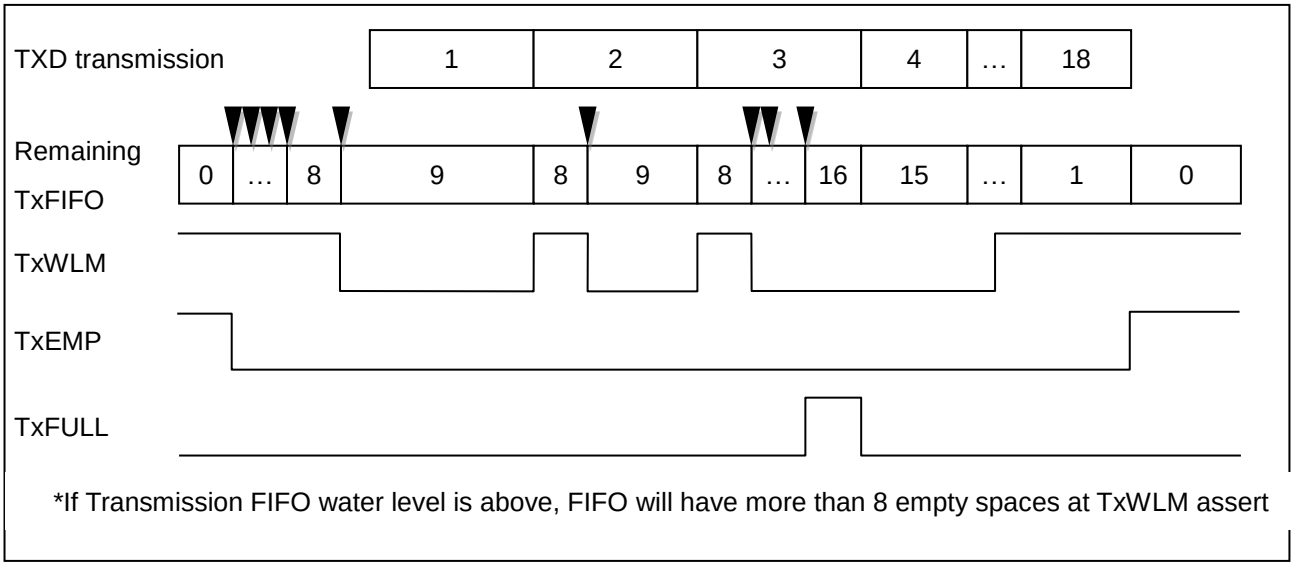


Figure 15-31 Operation example of Transmission FIFO (In case the water level is more than 8)

15-4-10-4 FIFO Control method at Reception data transfer

1. Follow Steps 1~6 of section 4-10-2 to setup a register.
2. Set Reception FIFO water level and FIFO enable in USFC register.
3. When using DMA USRF shall be used for the source address and DMACRREQ shall be used for the hardware control.
4. Set “1” to the Reception enable flag in the UCM register.
5. After detecting a Start bit on RXD pin, Data bit, Parity bit, and Stop bit will be sampled and data will be received.
6. Once one frame worth of data is received, the data will be transferred from Shift register to FIFO and Reception transfer complete interrupt factor flag will be set.
7. DMACRREQ,RxWLM reception FIFO water level matching flag, RxWLM_INT reception FIFO water level matching interrupt factor can be used for detecting whether the reception data is accumulated to reach the FIFO water level or not.

   When using DMA, reception data is read from the USRF register by DMA operation with DMACRREQ occurrence. When not using DMA, the reception data is read out by a program.
8. When the amount of reception data and completion is determined, after RxFF_END reception FIFO continuous transfer, FIFO operation select bit is switched to “1”. The last data in FIFO is read out by a program or with DMA software DREQ function.
9. When the amount of reception data and completion is not determined, RxFF_END is switched to “0”and at the time of reception interrupt, the remaining data in FIFO is read out by using DMACRREQ, RxWLM reception FIFO water level matching flag, and RxWLM_INT reception FIFO water level matching interrupt factor.
10. To read out the reception data without storing it, RxFF_set_LV reception FIFO water level must be at [111] (Reception FIFO data has to be 1 or more. Reading interval can be 16 frame period at the maximum.
11. RxFF_LV status register can provide information about the amount of current FIFO data.

    If a register value is 8, FIFO will be empty after reading the data 8 times.

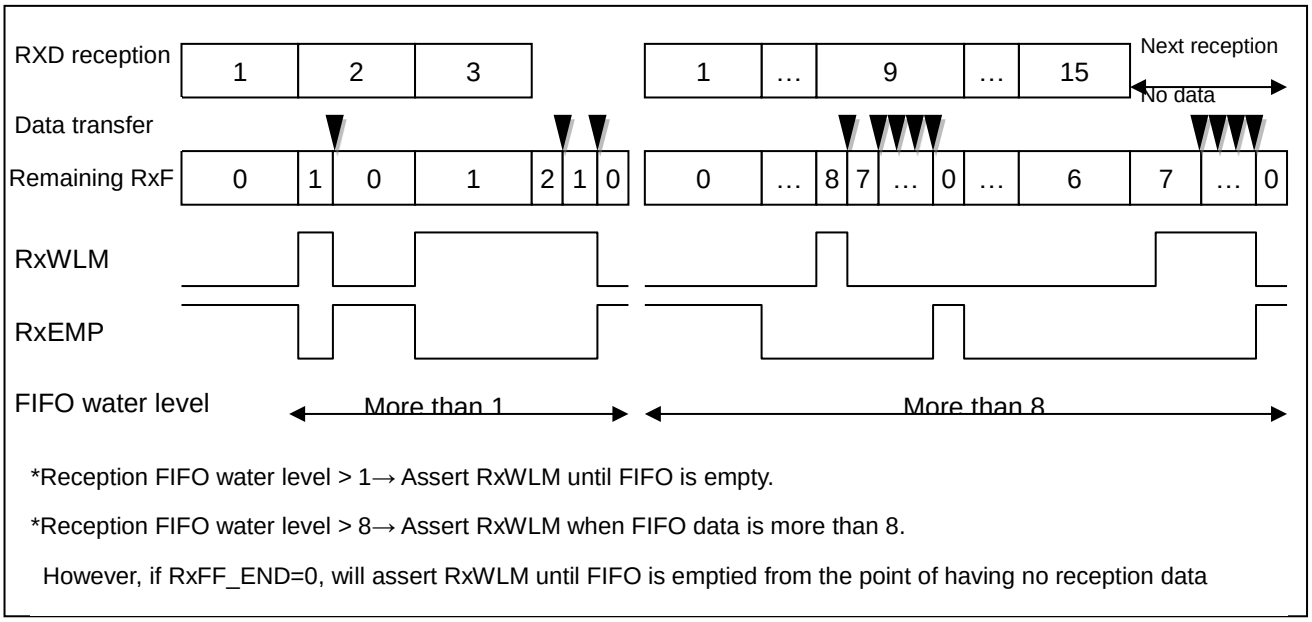


Figure 15-32 Operation example of Reception FIFO (In case the water level is more than 1 or 8)

16 I2C

16-1 Introduction

This chapter provides the outline of I2C in the section below.

- The Outline of I2C

16-1-1 The Outline of I2C

I2C interface: I2C is a block that conforms to I2C bus interface standards.

I2C bus is a serial bus protocol that is Philips standard for data transfer between ICs, which uses two open-drain serial lines (SDA and SCL). SDA is a serial data line that transfers data in series. SCL is a serial clock line that synchronizes data transfer between ICs. Each signal which I2C uses as an interface with the external is defined as SCL or SDA.

Table 16-1 Signal used with I2C

Signal	Input/Output	Function
SCL	Input/Output	I2C communication serial clock
SDA	Input/Output	I2C communication serial data

16-1-1-1 I2C Characteristics

- Supports Philips I2C bus interface.
- Single master function only (no arbitration function)
- Fast mode and Standard mode selectable.
- 9-bit data transfer (8-bit data and 1-bit acknowledge) only

16-2 Functional Overview

This chapter provides the outline of functional blocks used for the I2C configuration in the section below.

- I2C Circuit Configuration

16-2-1 I2C Circuit Configuration

I2C circuit configuration is as shown below.

16-3 Programmer's Model

This chapter contains I2C registers information and other information necessary for the register setup using microcontrollers in the sections below.

- About the Programmer's model
- Summary of I2C Registers
- Register Descriptions

16-3-1 About the Programmer's model

I2C base address is not fixed and it will depend on the system implementation. Please refer to the *System specifications* for more information about the base address.

Offset is fixed for all registers.

16-3-2 Summary of I2C Registers

I2C is controlled by the registers listed in Table 16-2.

Remarks: Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 16-2 I2C registers

Offset	Abbrev.	Attribute	Register name	Initial value
0x00	I2CCTL	R/W	I2C control register	0x0000_0003
0x04	I2CSTR	R/W	I2C status register	0x0000_0003
0x08	I2CTXD	W	I2C transmission data register	0x0000_0000
0x0C	I2CRXD	R	I2C reception data register	0x0000_0000
0x10	I2CCKS	R/W	I2C clock setting register	0x0000_FFFF

16-3-3 Register Descriptions

16-3-3-1 I2C Control Register: I2CCTL

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	FMODE	-	-	BTRIG	-	-	-	SRST
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	TRX	ST	ACK	IREQEN	SCLR	BC	SCLD	SDAD
Reset	0	0	0	0	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15: FMODE
Mode setting (Fast mode and Standard mode).
0: Standard mode
1: Fast mode

Bit12: BTRIG
It is a start trigger for 1 byte transmission/reception. When this bit is "1," 1 byte transmission or reception will be executed according to TRX setting.
0: No action.
1: Start transmission/reception of 1 byte data.

Remarks: This bit will automatically become “0” after the setup.

Bit8: SRST
Initializes all the registers (Synchronous reset). This will be the same state as the power-on reset state.
0: No initialization.
1: Execute initialization.

Remarks: This bit will automatically become “0” after the setup.

Bit7: TRX
Transmitter/Receiver setting. When address is sent, the set direction bit automatically sets a value.
0: Receiver
1: Transmitter

Bit6: ST
Generates start/stop condition
0: Generates the Stop condition
1: Generates the start condition

Bit5: ACK
Sets the acknowledge bit. It is effective when setting reception.
0: Output "1" to SDA at the acknowledge timing (NACK)
1: Output "0" to SDA at the acknowledge timing (ACK)

Attention: No change shall be made during communication.

Bit4: IREQEN
Sets an interrupt to the external
0: No interrupt output
1: Interrupt output

Bit3: SCLR
Clears the internal state. This is used for restart.
0: Do not clear the internal state.
1: Clears the internal state.

Remarks: This bit will automatically become “0” after the setup.

Bit2: BC (Bus Control)
This is an I2C forced control setting bit. When this bit is set to“1”, the values of SCLD and SDAD are directly sent onto the bus.
0: Executes normal operation
1: Controls the bus forcibly.

Bit1: SCLD
When BC = 1, this value is outputted to SCL line.
0: Outputs “0”
1: Outputs “1”

Bit0: SDAD
When BC = 1, this value is outputted to SDA line.
0: Outputs “0”
1: Outputs “1”

16-3-3-2 I2C Status Register: I2CSTR

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	BBSY	IREQ	ACKD	-	-	-	SCLB	SDAB
Reset	0	0	0	0	0	0	1	1
R/W	R	R/W	R	R	R	R	R	R

Bit7: BBSY (Bus Busy)
Bus communication status flag. The value will change by detecting start/stop condition.
0: I2C bus is in the open state
1: I2C bus is in the communication state

Bit6: IREQ
Interrupt status flag. This will change independently of IREQEN setting. Writing “0” will clear this bit.
0: No interrupt
1: With Interrupt

Remarks: It is automatically cleared when “1” is written into BTRIG of I2C control register.

Bit5: ACKD (ACK Detect)
Acknowledge signal detection status flag. This indicates the Acknowledge signal detection status of slave device and itself.
0: No acknowledge signal is detected.
1: Acknowledge signal is detected.

Bit1: SCLB
SCL pin level monitor on I2C bus
0: “L” level
1: “H” level

Bit0: SDAB
SDA pin level monitor on I2C bus
0: “L” level
1: “H” level

16-3-3-3 I2C Transmission Data Register: I2CTXD

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit7-0: TXD7-0
I2C bus transmission data register

Attention: Do not write data during the transmission.

16-3-3-4 I2C Reception Data Register: I2CRXD

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit7-0: RXD7-0
I2C bus reception data register

Attention: Do not read out data during the reception

16-3-3-5 I2C Clock Setting Register: I2CCKS

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	CKS15	CKS14	CKS13	CKS12	CKS11	CKS10	CKS9	CKS8
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	CKS7	CKS6	CKS5	CKS4	CKS3	CKS2	CKS1	CKS0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit15-0: CKS15-0

This is a register to set I2C bus SCL frequency. The system clock and SCL are related as shown below.

fSCL = (fSYS/CKS)/8

Note:

fSCL: SCL frequency

fSYS: System clock frequency

CKS: CKS setting value

Attention: Do not set “1” or “0” to CKS. It is also prohibited to change the setting during the communication.

16-4 Operation

This section describes I2C setting flow.

16-4-1 SCL and SDA Timing

The timing of SCL and SDA is as shown below.

16-4-1-1 Standard mode

SCL pin is sampled at a frequency 8 times higher than SCL frequency. When the sum of SCL extended “L” period by the Slave and the rising time does not exceed 1/8 fSCL, SCL “H” period cannot be extended. Therefore, in Standard mode, the minimum of 3/8 fSCL is ensured for SCL “H” period. Note that the specified minimum value 4.0 μs cannot be ensured at 93 Kbps or more.

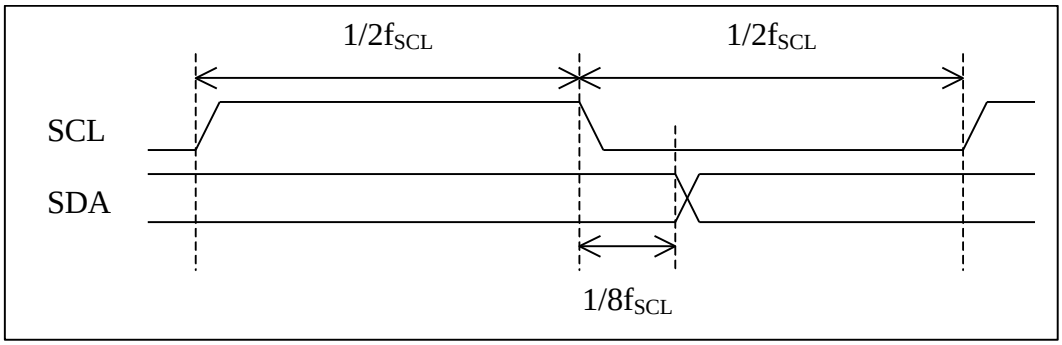


Figure 16-1 Standard Mode

16-4-1-2 Fast mode

Note that when SCL frequency is 390 KHz or more in the Fast mode, the specified minimum value of SCL “L” period may not be ensured even if SCL falling time is within the specification. For example, when SCL frequency is at 400 KHz and SCL falling time is 300 ns, SCL “L” period will be:

1/(400×103)×(5/8)−300×10-9 = 1.262×10-6

And the result will be 1.262 μs. The specified minimum value of SCL “L” period is 1.3 μs. So, this does not meet the specification.

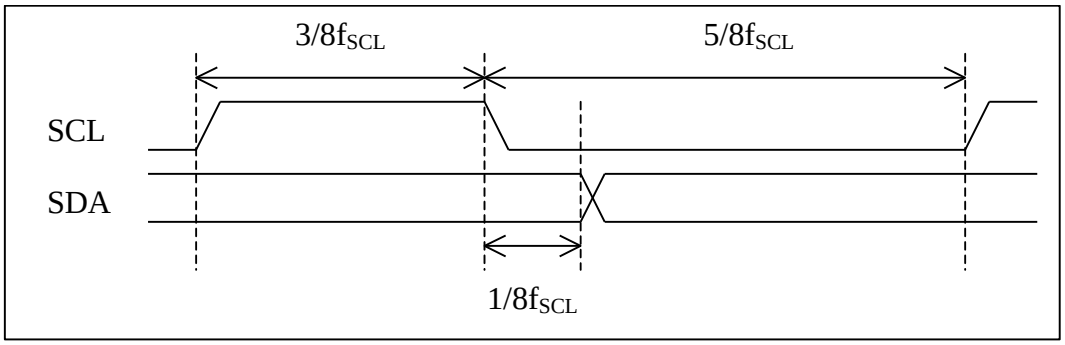


Figure 16-2 Fast Mode

16-4-2 Operating Sequence

This module starts sequence by the register settings. These sequence operations and register setting are as shown below.

Table 16-3

Operation	TRX	ST	SCLR	BTRIG	Note
Start condition + Address generation	1	0→1	-	-	
Stop condition generation	1	1→0	-	-	
1 byte send	1	1	-	0→1	
1 byte receive	0	1	-	0→1	
State clear	-	-	0→1	-	For restart

After these sequence operations are complete, IREQ of I2C status register will always become “1.” During the sequence operation, another sequence start is not accepted. So, always use IREQ polling or interrupt to complete a sequence before starting the next sequence.

There is no need to set the bits simultaneously. However, in the table above, “*→*” indicates a bit that triggers internal operation. Therefore, to control bits separately, a bit marked “*→*” shall be set last.

16-4-3 Control Flow

1: Initialization

1.1 Set I2C clock setting register in accordance with the transfer speed.

2: Generation of start condition and slave address

2.1 Write a slave address and direction bit to be sent to I2C transmission data register.

2.2 Make sure that I2C status register has BBSY = 0 (bus open state).

2.3 When “1” is written into TRX and ST of I2C control register, start condition and a slave address written in I2C transmission data register are sent to I2C bus. After that, at the 9th-clock fall of SCL pin, I2C status register, IREQ becomes “1” and an interrupt request occurs. At this time, SCL pin will remain as “L.” In synchronization with interrupt request occurrence, TRX automatically changes according to the transmission direction bit.

After detecting an interrupt, check the acknowledge return by reading out I2C status register, ACKD. If ACKD is set to “0”, acknowledge is not returned so generate stop condition as described below to finish the communication. If ACKD is set to “1”, acknowledge is returned and 1 byte data transfer continues.

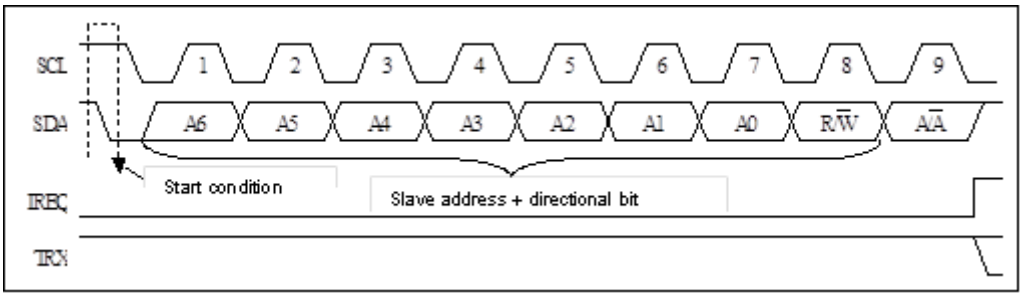


Figure 16-3

3: 1 byte data transmission

3.1: In case of Transmission

Write transmission data in I2C transmission data register. Then, write “1” in BTRIG of I2C control register. When “1” is written in BTRIG, IREQ of I2C status register is automatically cleared and clock from SCL pin and the content of I2C transmission data register from SDA pin are sent as data. After completing the data transmission, IREQ becomes “1” and an interrupt request occurs. At this time, SCL pin remains “L.” If multiple byte transmission is necessary, it will repeat the process of acknowledge signal confirmation by ACKD of I2C status register, the next transmission data write, and writing “1” to BTRIG; this procedure will occur at the end of each 1 byte data transmission. After sending the last byte, communication will complete by following the Stop condition generation process as described below.

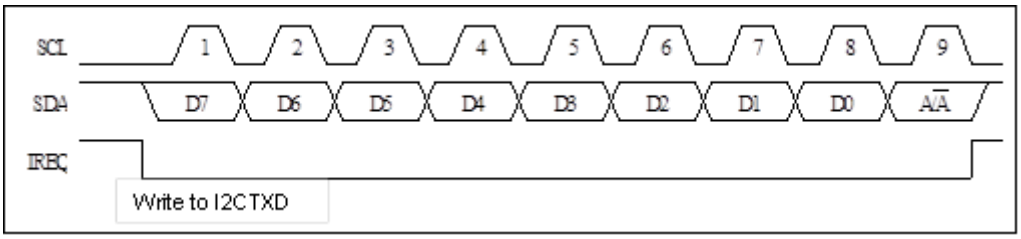


Figure 16-4

3.2 In case of reception

To return acknowledge, set ACK of I2C control register to “1”. Then, to start receiving 1 byte data, write “1” to BTRIG of I2C control register. When “1” is written in BTRIG, IREQ of I2C status register is automatically cleared, a clock is sent from SCL pin and “L” is output to SDA pin at the acknowledge timing. The received data is imported to I2C reception data register. To receive multiple data, read out data from I2C reception data register and write “1” in BTRIG each time per 1 byte reception.

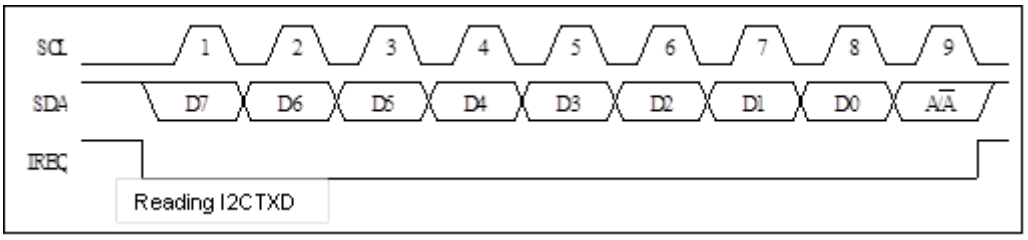


Figure 16-5

To finish data transmission for the transmitter, set ACK to “0” before writing “1” in BTRIG for starting to receive the last byte. This allows outputting “H” to SDA pin at the timing of the last-received-byte acknowledge to inform the transmitter about the completion of data transmission.

After the last-byte receiving finishes, the communication will complete by following the Stop condition generation process after reading out the last reception data.

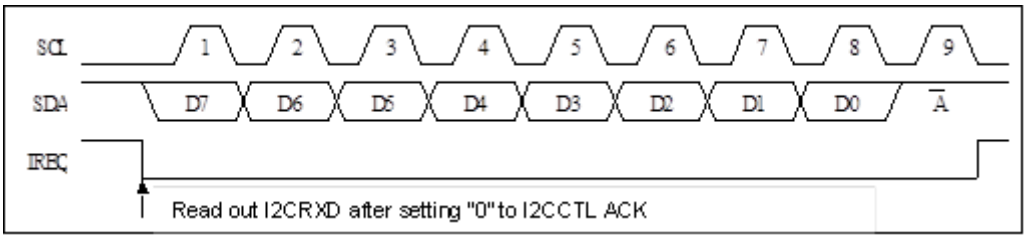


Figure 16-6

4: Generation of Stop condition

4.1 To clear the last transmission/reception interrupt, write “0” in IREQ of I2C status register.

4.2 When I2C status register, BBSY = 1, write “1” in TRX and “0” in ST of I2C control register to generate the stop condition on I2C bus.

4.3 Write “0” in IREQ of I2C status register to clear it.

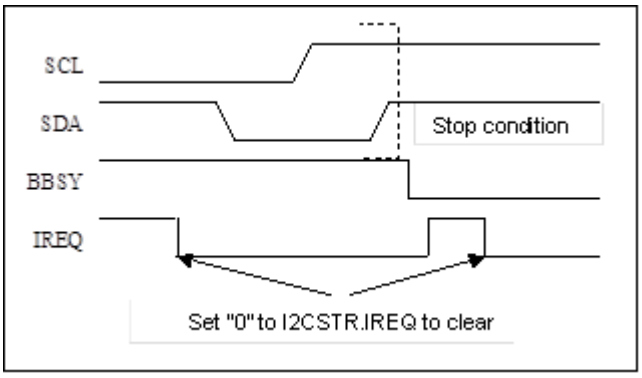


Figure 16-7

5: Restart flow

5.1 To clear the last transmission/reception interrupt, write “0” in IREQ of I2C status register.

5.2 When I2C status register, BBSY = 1, write “1” in SCLR of I2C control register to clear the internal state.

5.3 Wait for an interrupt to occur and make sure SCL pin is opened.

5.4 Write “0” in IREQ of I2C status register to clear it.

5.5 Write “0” in TRX and ST of I2C control register to initialize the bits.

5.6 Be sure to confirm that the bus is in the same state as FREE by following the above steps, and then generate the Start condition and slave address by following Step 2 mentioned above.

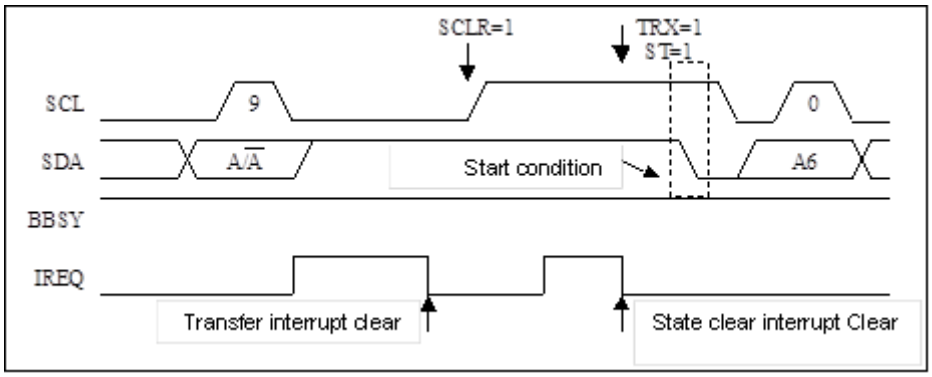


Figure 16-8

16-4-4 Precautions for use

Please pay attention to the above mentioned procedures for I2C setup.

17 S-Flash

17-1 Introduction

This chapter provides the brief summary of S-Flash in the sections below:

- Brief Summary of S-Flash

17-1-1 Brief Summary of S-Flash

This document is Serial Flash Controller(S-Flash) document that controls the serial flash device. Several kinds of modes are prepared to correspond to SPIMode and ParallelMode of some manufacturer.

Table 17-1 Signal used with S-Flash

signal	I/O	function
SF_nCS	O	chip select to serial flash device
SF_SCK	O	clock to serial flash device
SF_SDOUT	I/O	reception data from serial flash device
SF_nWP	I/O	write protection to serial flash device
SF_nHOLD	I/O	hold to serial flash device
SF_SDIN	I/O	transmission data to serial flash device

17-1-1-1 Feature of S-Flash

- Transfer mode

SPI Mode: It corresponds to SPI Mode0.

Parallel Mode: It corresponds to ParallelMode of some manufacturers.

- FIFO function

It provides with transmission FIFO and reception FIFO.

Transmission FIFO size is 8 bytes.Reception FIFO size is 8 bytes.

The data stored in transmission FIFO is transmitted to the serial flash device in stored order.

The data that has been transmitted from the serial flash device is stored in reception FIFO in transferred order.

Transmission and reception of data to the serial flash device is achieved by the access to this transmission FIFO and reception FIFO.

- Accelerator function

The area where serial flash device is arranged can be read from the system bus by using the accelarator function when the area for the serial flash device is arranged on the memory map.

17-2 Functional Overview

This chapter provides overview of functional blocks used for the S-Flash configuration in the sections below:

- S-Flash Circuit Configuration

17-2-1 S-Flash Circuit Configuration

S-Flash circuit configuration is shown in a Figure 17-1

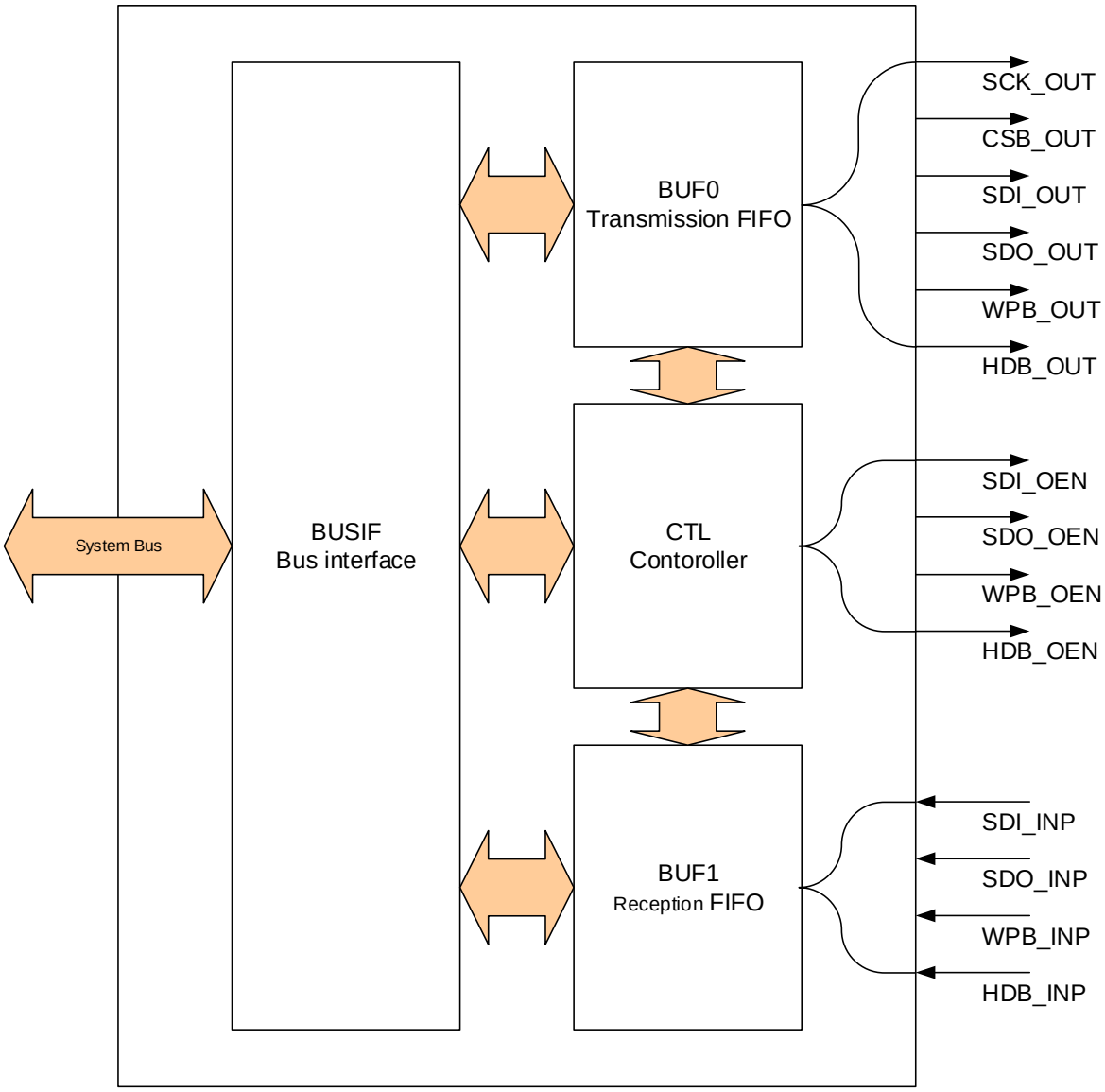


Figure 17-1 S-Flash Configuration

17-3 Programmer's Model

This chapter contains S-Flash register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of S-Flash registers
- Register Descriptions

17-3-1 About the programmer's model

S-Flash base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

17-3-2 Summary of S-Flash registers

S-Flash is controlled by the registers shown in Table 17-2 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 17-2 S-Flash registers

OFFSET	Abbreviation	Attribute	Register Names	Default value
0x00	SF_SIZE	R/W	Transfer size setting register	0x0000_0000
0x04	SF_CTL	R/W	Transfer control register	0x0000_0000
0x08	SF_MODE	R/W	Transfer mode register	0x0000_0000
0x0C	SF_DUMMY	R/W	Dummy data setting register	0x0000_0000
0x10	SF_FIFO_CLR	R/W	FIFO clear register	0x0000_0000
0x14	SF_STATUS	R/W	Status register	0x0000_0103
0x18	SF_IRQEN	R/W	Interrupt enable register	0x0000_0000
0x1C	SF_FIFO_NUM	R	FIFO data volume register	0x0000_0000
0x20	SF_SRSTB	R/W	Soft reset register	0x0000_0001
0x24	SF_PHASE_SET	R/W	Phase setting register	0x0000_0000
0x28	SF_BUS	R/W	Bus accelarator register	0x0000_0000
0x2C	SF_TIMING	R/W	Timing adjustment register	0x0000_0000
0x30	SF_T_FIFO	W	Transmission FIFO access register	0x0000_0000
0x34	SF_R_FIFO	R	Reception FIFO access register	0x0000_0000

17-3-3 Register Descriptions

17-3-3-1 Transfer size setting register:SF_SIZE

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	NO_READ	UL_SIZE3	UL_SIZE2	UL_SIZE1	UL_SIZE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	T_SIZE19	T_SIZE18	T_SIZE17	T_SIZE16
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	T_SIZE15	T_SIZE14	T_SIZE13	T_SIZE12	T_SIZE11	T_SIZE10	T_SIZE9	T_SIZE8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	T_SIZE7	T_SIZE6	T_SIZE5	T_SIZE4	T_SIZE3	T_SIZE2	T_SIZE1	T_SIZE0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit28 : NO_READ

It is used to set whether to store reception data from serial flash device in reception FIFO.
0: The reception data is stored in reception FIFO.
1: The reception data isn't stored in reception FIFO.

Attention) Don't change the setting when [17-3-3-2Transfer control register:SF_CTL].ACT is "1".

Bit27-24 : UL_SIZE

It is used to set numbers of annulled reception data while transferring once(The period when SF_nCS change as follow. H→L→H).
When "5" is set, the first 5 bytes data is annulled and the reception data from 6th byte is stored in reception FIFO.
When "0" is set, reception data from 1st byte is stored to reception FIFO.

Attention) Set [17-3-3-1Transfer size setting register:SF_SIZE].T_SIZE or less.
Attention) Don't change the setting when [17-3-3-2Transfer control register:SF_CTL].ACT is "1".

Bit19-0 : T_SIZE

It is used to set transfer data size in bytes while transferring once(The period when SF_nCS change as follow. H→L→H).

Attention) Don't change the setting when [17-3-3-2Transfer control register:SF_CTL].ACT is "1".

17-3-3-2 Transfer control register:SF_CTL

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	ACT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit11-0 : ACT

It is used to control transfer.Transfer is begun by setting "1".
It automatically returns to 0 when [17-3-3-1Transfer size setting register:SF_SIZE].tsize bytes data is transmitted.

0:transfer completion.
1:transfer.

Attention) At the start of transfer, reception FIFO is automatically cleared.
Attention) Even if "0" is set, it is invalid.

17-3-3-3 Transfer mode register:SF_MODE

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	R_MODE	T_MODE
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	MODE2	MODE1	MODE0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit9 : R_MODE
It is used to control read access to reception FIFO.
For details, refer to [17-3-3-14Reception FIFO access register:SF_R_FIFO].

Bit8 : T_MODE
It is used to control write access to transmission FIFO.
For details, refer to [17-3-3-13Transmission FIFO access register:SF_T_FIFO].

Bit2-0 : MODE
It is used to set transfer mode.
0: Refer to Figure 17-3.
1: Refer to Figure 17-4.
2: Refer to Figure 17-5.
3: Refer to Figure 17-6
4: Refer to Figure 17-7.
5: Refer to Figure 17-8
6: Refer to Figure 17-9
7: Refer to [17-3-3-10Phase setting register:SF_PHASE_SET].

Attention) Please note that the collision of bidirectional pin doesn't occur.

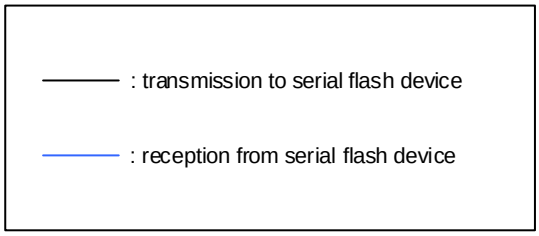


Figure 17-2

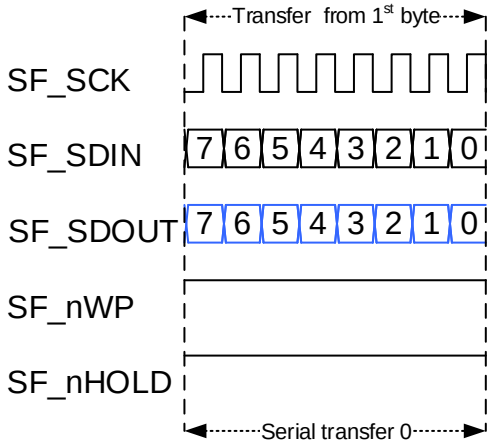


Figure 17-3 transfer mode 0

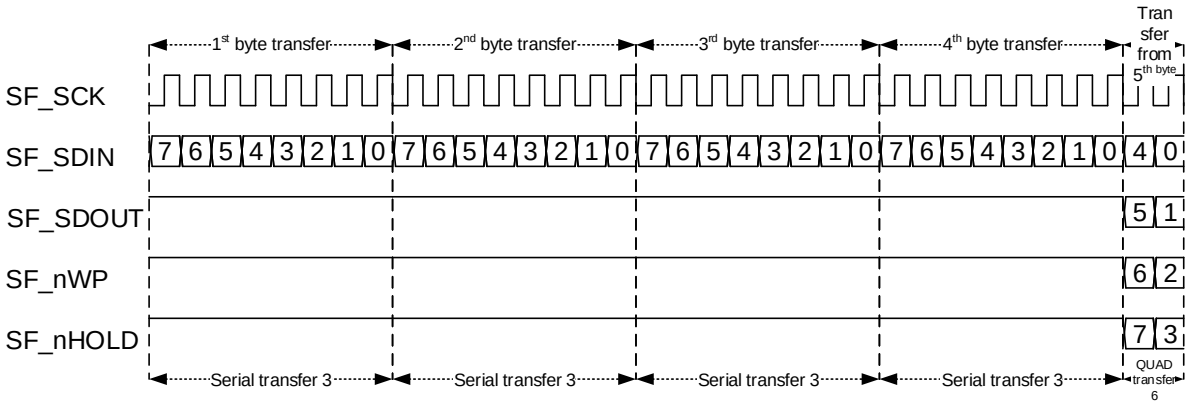


Figure 17-4 transfer mode 1

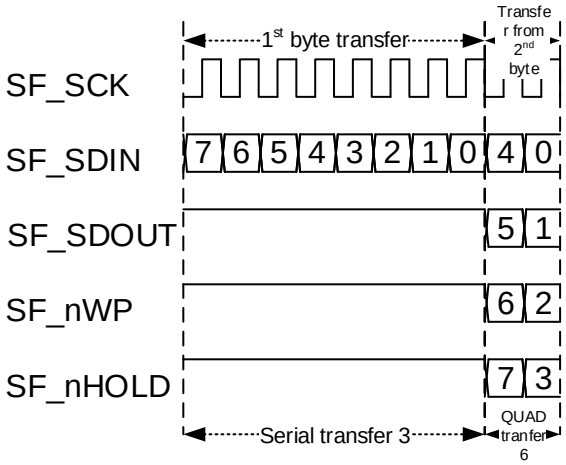


Figure 17-5 transfer mode 2

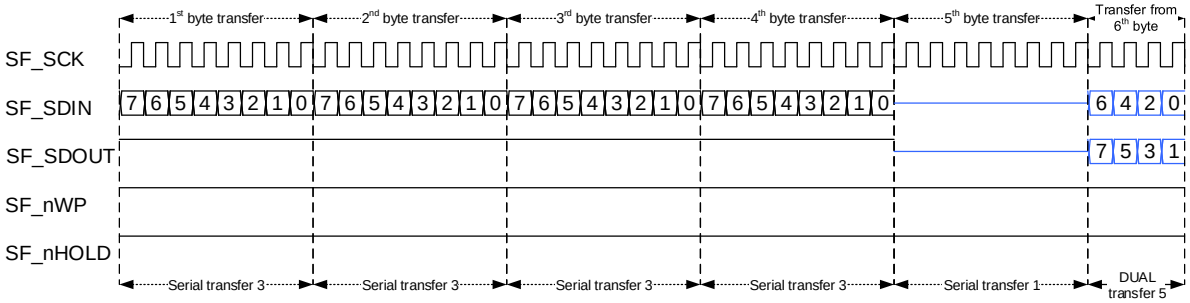


Figure 17-6 transfer mode 3

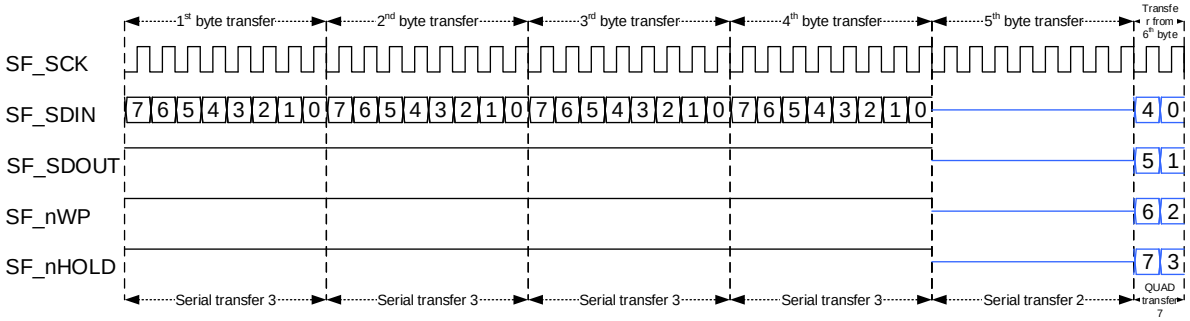


Figure 17-7 transfer mode 4

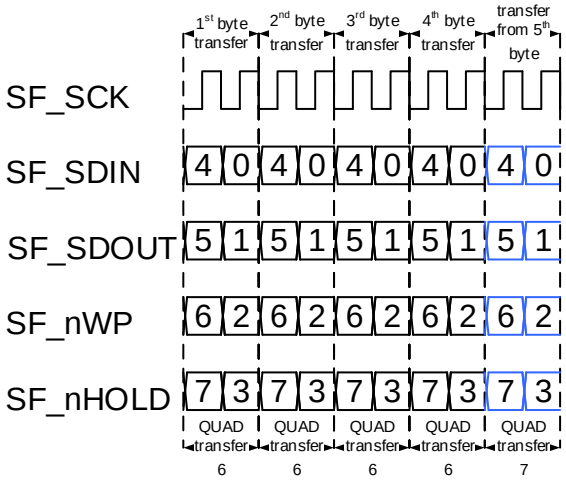


Figure 17-8 transfer mode 5

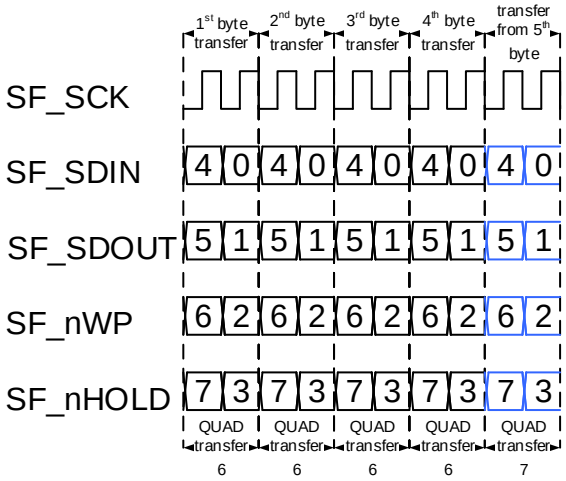


Figure 17-9 transfer mode 6

17-3-3-4 Dummy data setting register:SF_DUMMY

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	DUMMY
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0 : DUMMY

It is the function to receive data from the serial flash device without setting the dummy data in transmission FIFO.

When [17-3-3-2Transfer control register:SF_CTL].ACT is "1" & transmission FIFO is empty & reception FIFO isn't full, clock supply is repeated until transfer is completed.

0:clock isn't supplyed without setting the dummy data in transmission FIFO.

1:clock is supplied without setting the dummy data in transmission FIFO.

17-3-3-5 FIFO clear register:SF_FIFO_CLR

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	R_CLR	T_CLR
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1 : R_CLR

The data volume of reception FIFO and reception FIFO is cleared.

0:Clear isn't performed.

1:Clear is performed.It automatically returns to 0 when clear process is completed.

Attention) It is prohibited to clear while transfer.

Bit0 : T_CLR

The data volume of transmission FIFO and transmission FIFO is cleared.

0: Clear isn't performed.

1: Clear is performed.It automatically returns to 0 when clear process is completed.

Attention) It is prohibited to clear while transfer.

17-3-3-6 Status register:SF_STATUS

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	FIN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	R_U_ERR	R_O_ERR	-	R_FULL	R_HALF	R_EMP
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	T_O_ERR	-	T_FULL	T_HALF	T_EMP
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R

Bit16 : FIN
It indicates transfer completion status.
0:Transfer isn't completed.
1:Transfer was completed.

Attention) FIN doesn't automatically return to "0". It is necessary to set "0" to return it to "0".

Bit13: R_U_ERR
It indicates that the underflow of reception FIFO occurred.
0:underflow doesn't occur.
1:underflow occurred.

Attention) when it is 1, perform hard reset or soft reset.

Bit12 : R_O_ERR
It indicates that the overflow of reception FIFO occurred.
0:overflow doesn't occur.
1:overflow occurred.

Attention) when it is 1, perform hard reset or soft reset.

Bit10 : R_FULL
It indicates that the data volume of reception FIFO is 8 bytes.
0: It is less than 8 bytes.
1: It is 8 bytes.

Bit9 : R_HALF
It indicates that the data volume of reception FIFO is 4 bytes or more.
0: It is less than 4 bytes.
1: It is 4 bytes or more.

Bit8 : R_EMP
It indicates that the data volume of reception FIFO is 0 byte.
0:It is 1 byte or more.
1:It is 0 byte.

Bit4: T_O_ERR
It indicates that the overflow of transmission FIFO occurred.
0:overflow doesn't occur.
1:overflow occurred.

Attention) when it is 1, perform hard reset or a soft reset.

Bit2 : T_FULL
It indicates that the data volume of transmission FIFO is 8 bytes.
0: It is less than 8 bytes.
1: It is 8 bytes.

Bit1 : T_HALF
It indicates that the data volume of transmission FIFO is less than 5 bytes.
0: It is 5 bytes or more.
1: It is less than 5 bytes.

Bit0 : T_EMP
It indicates that the data volume of transmission FIFO is 0 byte.
0: It is 1 byte or more.
1: It is 0 byte.

17-3-3-7 Interrupt enable register:SF_IRQEN

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	FIN_EN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	R_U_ERR_EN	R_O_ERR_EN	-	R_FULL_EN	R_HALF_EN	R_EMP_EN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	T_O_ERR_EN	-	T_FULL_EN	T_HALF_EN	T_EMP_EN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit13 : FIN_EN
Interrupt by FIN is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit13: R_U_ERR_EN
Interrupt by R_U_ERR is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit12 : R_O_ERR_EN
Interrupt by R_O_ERR is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit4: T_O_ERR_EN
Interrupt by T_O_ERR is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit9 : R_OVER_EN
Interrupt by R_OVER is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit8 : R_FULL_EN
Interrupt by R_FULL is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit1 : T_UNDER_EN
Interrupt by T_UNDER is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

Bit0 : T_EMP_EN
Interrupt by T_EMP is controlled.
0:Interrupt is prohibited.
1:Interrupt is permitted.

17-3-3-8 FIFO data volume register:SF_FIFO_NUM

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	R_NUM3	R_NUM2	R_NUM1	R_NUM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	T_NUM3	T_NUM2	T_NUM1	T_NUM0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit11-8 : R_NUM
It indicates data volume of reception FIFO in bytes.

Bit3-0 : T_NUM
It indicates data volume of transmission FIFO in bytes.

17-3-3-9 Soft reset register:SF_SRSTB

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	SRSTB
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Bit0 : SRSTB
It used to control soft reset.
0: Soft reset.
1: Soft reset release.

Attention) Registers that are described in this manual are not reset by SRSTB.

17-3-3-10 Phase setting register:SF_PHASE_SET

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	PHASE52	PHASE51	PHASE50	-	PHASE42	PHASE41	PHASE40
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	PHASE32	PHASE31	PHASE30	-	PHASE22	PHASE21	PHASE20
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	PHASE12	PHASE11	PHASE10	-	PHASE02	PHASE01	PHASE00
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

When 7 is set to [17-3-3-3Transfer mode register:SF_MODE].MODE, It used to control each byte transdfer.

Bit22-20 : PHASE5
It used to control transfer from 6th byte. Note)

Bit18-16 : PHASE4
It used to control 5th byte transfer. Note)

Bit14-12 : PHASE3
It used to control 4th byte transfer. Note)

Bit10-8 : PHASE2
It used to control 3rd byte transfer. Note)

Bit6-4 : PHASE1
It used to control 2nd byte transfer. Note)

Bit2-0 : PHASE0
It used to control 1st byte transfer. Note)

Note) Byte transfer is shown below.

0: Refer to Figure 17-10 (serial transfer 0).
1: Refer to Figure 17-10 (serial transfer 1).
2: Refer to Figure 17-10 (serial transfer 2).
3: Refer to Figure 17-10 (serial transfer 3).
4: Refer to Figure 17-10 (DUAL transfer 4).
5: Refer to Figure 17-10 (DUAL transfer 5).
6: Refer to Figure 17-10 (QUAD transfer 6).
7: Refer to Figure 17-10 (QUAD transfer 7).

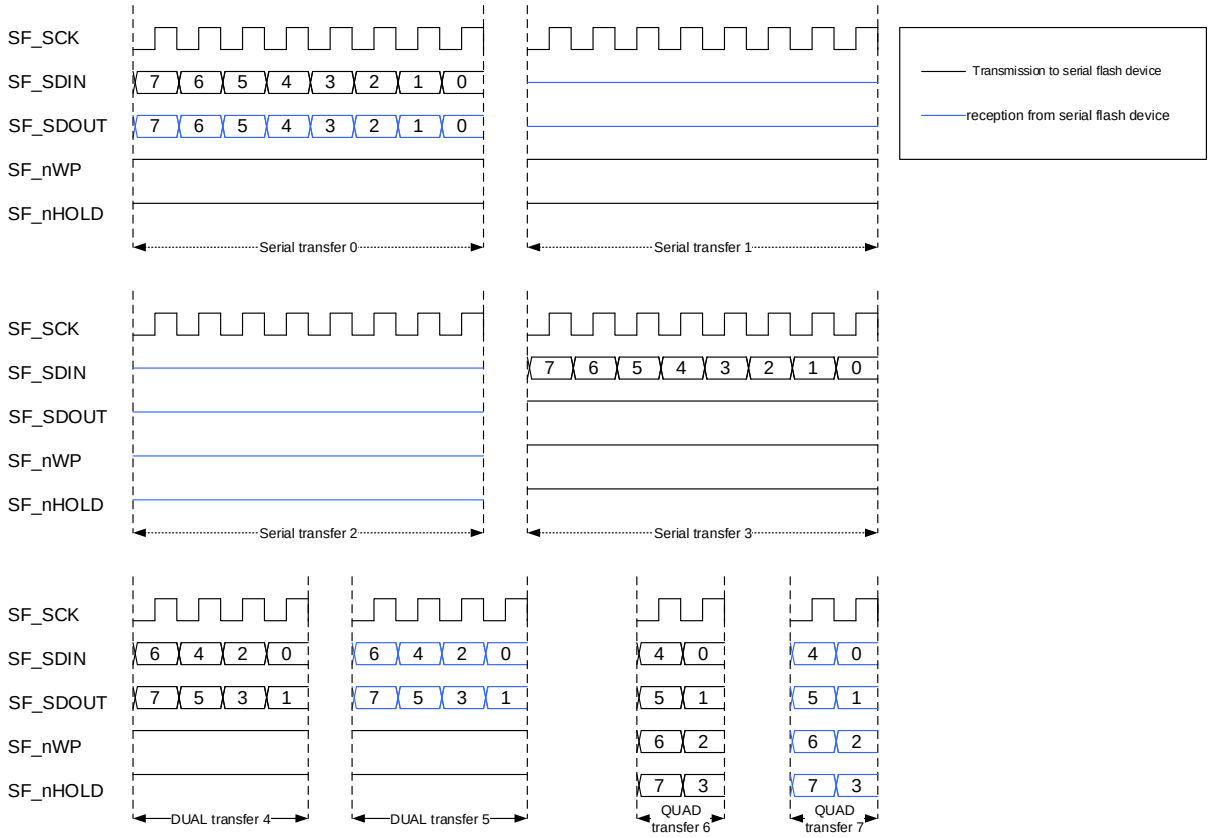


Figure 17-10 regarding byte transfer

Figure 17-11 shows byte transfer when "0" is set in Bit2-0, "6" is set in Bit6-4, "6" is set in Bit10-8, "6" is set in Bit14-12, "6" is set in Bit18-16, and "7" is set to Bit22-20.

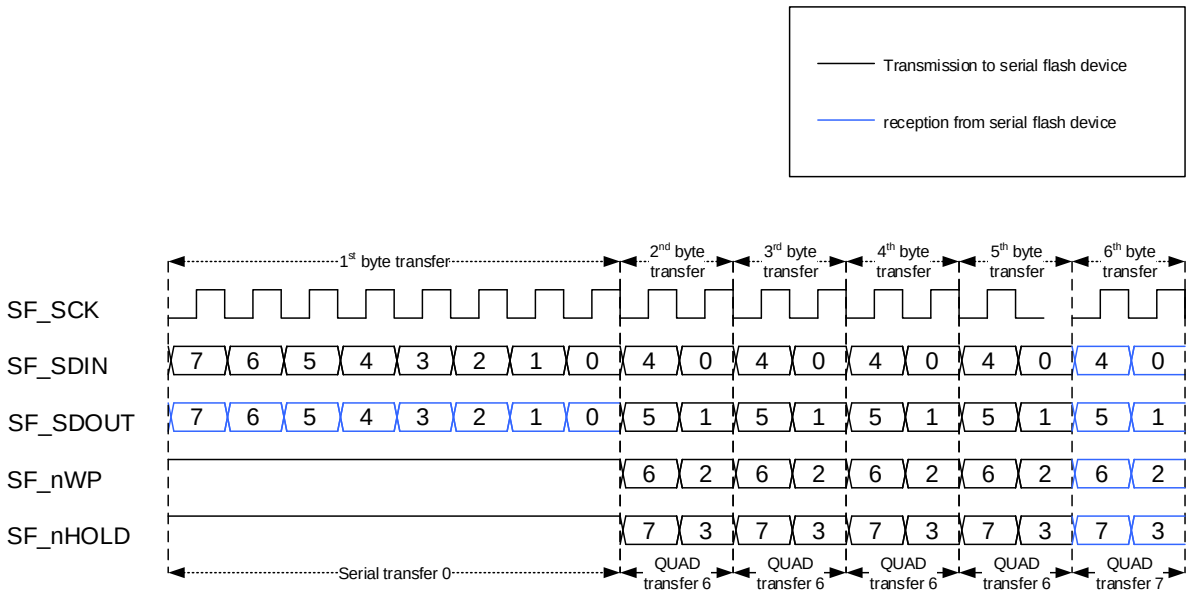


Figure 17-11 example of setting each byte transfer

17-3-3-11 Bus accelarator register:SF_BUS

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	BUSACT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	BUSMODE2	BUSMODE 1	BUSMODE 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	LOOKAHEAD	BUSEN
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R/W

It used to control accelarator function between system bus and serial flash device.

It is neccesary to be arranged the area for the serial flash device on the memory map.

The area where serial flash device is arranged can be read from the system bus by using the accelarator function.

Accelatator function details is shown by using Figure 17-12.

In this case, the area for the serial flash device is arranged at 0x1000~0x13FF on the memory map.

By read access from the system bus at 0x1000~0x13FF on the memory map, accelarator reply on the following procedures.

(*1) Accelarator recognizes read access and read address from host.

(*2) Accelarator issues read command to serial flash device.

(*3) Accelarator issues adress to serial flash device based on read address acquired in (*1).

(*4) Accelarator acquires read data from Serial flash device.

(*5) Accelarator returns read data to host.

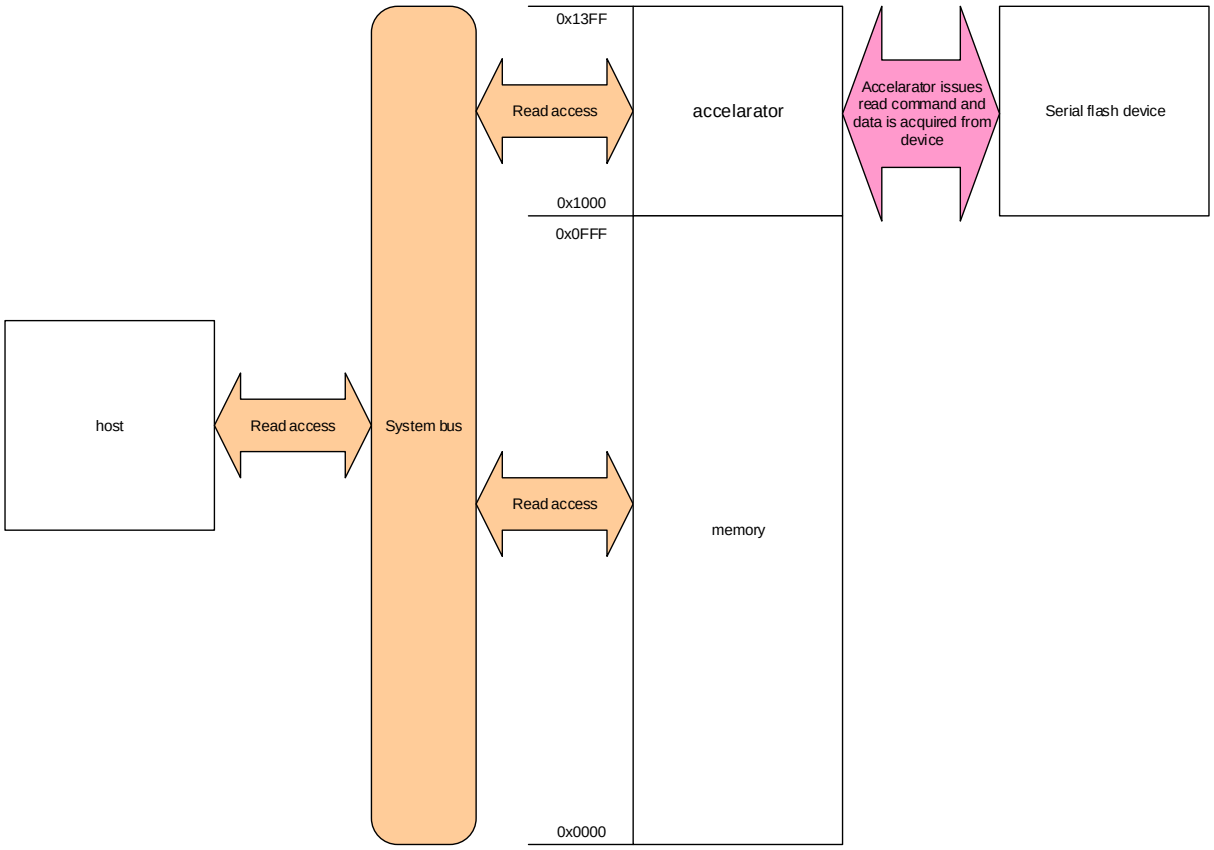


Figure 17-12 regarding accelatator function

Bit24 : BUSACT
It indicates accelarator funciton state.
0:accelarator function is invalid state.
1:accelarator function is valid state.

Bit10-8 : BUSMODE
It used to select read command that accelarator issues.
0:0x03.
1: 0x 0B.
2: 0x 3B.
3: 0x EB.
4: 0x E7.
5: 0x EB(MODE=0xa5).
6: 0x E7(MODE=0xa5).

Attention) Select read command supported by serial flash device.
Attention) Read command might not be accepted by the state of serial flash device. In this case, perform necessary setting to serial flash device, before validing accelarator funciton.
Attention) It might not correspond to read command to odd address by serial flash device. In this case, if host access to odd address, don't use command that doesn't support read access to odd adderss.

Bit1 : LOOKAHEAD
write acess
0: Read-ahead funciton is valid.
1: Read-ahead funciton is invalid.

Read access
Always read 0.

Read-ahead funciton is valid:
Even if read access isn't performed from host to accelarator, accelerator performs read accesses sequentially to serial flash device and store read data in reception FIFO.
If access to accelerator is non-sequential read, reception FIFO is cleared.
If access to accelerator is sequential read, accelerator return data in reception FIFO.When reception FIFO is full, accelerator stop read access to serial flash device.However, accelerator keeps chip select for serial flash device to ON state for next sequential read.
Read-ahead funciton is invalid:
Accelerator performs read access according read access from host. When read access from host to accelerator is not performed, accelerator change chip select for serial flash device from ON state to OFF state.

Bit0 : BUSEN
0:accelarator function is invalid.
1:accelarator function is valid.

Attention) When [17-3-3-2Transfer control register:SF_CTL].ACT is "1", Don't change BUSEN from "0" to "1".

Attention) When BUSEN is "1", access to serial flash device by using register is invalid. Please do not access registers excluding [17-3-3-11Bus accelarator register:SF_BUS] described in this specifications.When BUSEN is "1", don't change BUSMODE.

Attention) When BUSACT is "0", Accelaretor return "0" to access from host.

Attention) Time is required until the control by BUSEN is reflected in BUSACT.

BUSACT changes to "1" while read access to this register once after setting BUSEN to "1".

BUSACT changes to "0" while read access to this register three times after setting BUSEN to "0".

Attention) When accelarator function is switched, reception FIFO and transmission FIFO are cleared.

17-3-3-12 Timing adjustment register:SF_TIMING

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	CS2CLK
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	CS2	CS1	CS0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Bit8 : CS2CLK

It used to adjust period from falling edge of SF_nCS to rising edge of SF_SCLK.

0: After the half cycle of CLK from falling edge of SF_nCS, rising edge of SF_SCK is generated.

1: After the 1.5 cycle of CLK from falling edge of SF_nCS, rising edge of SF_SCK is generated.

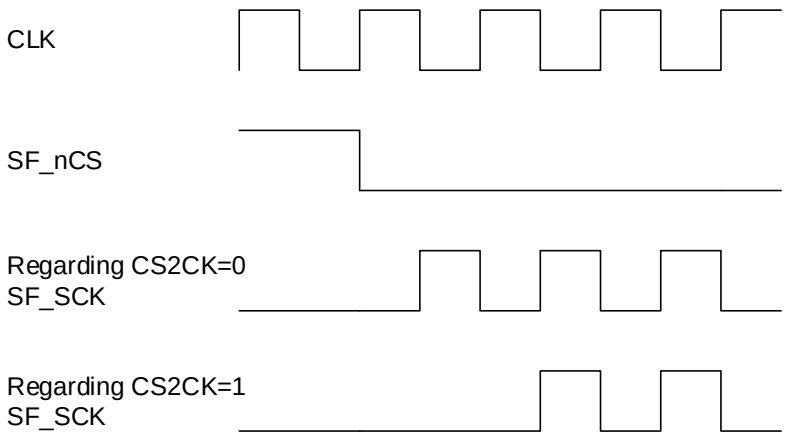


Figure 17-13

Bit2-0 : CS

It used to ensure the period from rising edge of SF_nCS to falling edge of SF_nCS.

0:During 1 cycle of CLK, It is guaranteed that SF_nCS is H.

1:During 2 cycle of CLK, It is guaranteed that SF_nCS is H.

2:During 3 cycle of CLK, It is guaranteed that SF_nCS is H.

3:During 4 cycle of CLK, It is guaranteed that SF_nCS is H.

4:During 5 cycle of CLK, It is guaranteed that SF_nCS is H.

5:During 6 cycle of CLK, It is guaranteed that SF_nCS is H.

6:During 7 cycle of CLK, It is guaranteed that SF_nCS is H.

7:During 8 cycle of CLK, It is guaranteed that SF_nCS is H.

Remarks) CLK is function clock for S-Flash function block.

17-3-3-13 Transmission FIFO access register:SF_T_FIFO

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	T_FIFO31	T_FIFO30	T_FIFO29	T_FIFO28	T_FIFO27	T_FIFO26	T_FIFO25	T_FIFO24
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Bit	23	22	21	20	19	18	17	16
Name	T_FIFO23	T_FIFO22	T_FIFO21	T_FIFO20	T_FIFO19	T_FIFO18	T_FIFO17	T_FIFO16
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Bit	15	14	13	12	11	10	9	8
Name	T_FIFO15	T_FIFO14	T_FIFO13	T_FIFO12	T_FIFO11	T_FIFO10	T_FIFO9	T_FIFO8
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Bit	7	6	5	4	3	2	1	0
Name	T_FIFO7	T_FIFO6	T_FIFO5	T_FIFO4	T_FIFO3	T_FIFO2	T_FIFO1	T_FIFO0
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit31-0 : T_FIFO

It used to write access to transmission FIFO.

When [17-3-3-2Transfer control register:SF_CTL].ACT is "1", the data in transmission FIFO are sequentially transmitted to serial flash device. Data volume of transmission FIFO is decremented by 1 whenever 1 byte data is transferred.

By byte write access to T_FIFO7-0, write data is stored in transmission FIFO. The data volume is incremented by 1.

When T_MODE is 0.

By 2 bytes(half word) write access to T_FIFO15-0, write data is stored in order of T_FIFO15-8 and T_FIFO7-0. Data volume of transmission FIFO is incremented by 2.

By 4 bytes(word) write access to T_FIFO31-0, write data is stored in order of T_FIFO31-24, T_FIFO23-16, T_FIFO15-8 and T_FIFO7-0. Data volume of transmission FIFO is incremented by 4.

When T_MODE is 1.

By 2 bytes(half word) write access to T_FIFO15-0, write data is stored in order T_FIFO7-0 of and T_FIFO15-8. Data volume of transmission FIFO is incremented by 2.

By 4 bytes(word) write access to T_FIFO31-0, write data is stored in order of T_FIFO7-0, T_FIFO15-8, T_FIFO23-16 and T_FIFO31-24. Data volume of transmission FIFO is incremented by 4.

Remarks) Direction of I/O pin becomes input only by setting [17-3-3-3Transfer mode register:SF_MODE].MODE(example:When "3" is set in MODE, direction of I/O pin becomes input only for transfer from 5th byte).

In this case, data in transmission FIFO isn't reached to serial flash device.But Data volume of transmission FIFO is decremented by 1 whenever 1 byte data is transferred.

Attention) Transmission FIFO size is 8 bytes. Please don't write access when transmission FIFO is full.

17-3-3-14 Reception FIFO access register:SF_R_FIFO

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	R_FIFO31	R_FIFO30	R_FIFO29	R_FIFO28	R_FIFO27	R_FIFO26	R_FIFO25	R_FIFO24
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	R_FIFO23	R_FIFO22	R_FIFO21	R_FIFO20	R_FIFO19	R_FIFO18	R_FIFO17	R_FIFO16
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	R_FIFO15	R_FIFO14	R_FIFO13	R_FIFO12	R_FIFO11	R_FIFO10	R_FIFO9	R_FIFO8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	R_FIFO7	R_FIFO6	R_FIFO5	R_FIFO4	R_FIFO3	R_FIFO2	R_FIFO1	R_FIFO0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-0 : R_FIFO

It used to read access to reception FIFO.

Reception data is stored in reception FIFO. Data volume of reception FIFO is incremented by 1 whenever 1 byte data is transferred.

By byte read access to R_FIFO7-0, data in reception FIFO is displayed. And the data volume is decremented by 1.

When R_MODE is 0.

By 2 bytes(half word) read access to R_FIFO15-0, data in reception FIFO is displayed in order of R_FIFO15-8 and R_FIFO7-0. Data volume of reception FIFO is decremented by 2.

By 4 bytes(word) read access to T_FIFO31-0, data in reception FIFO is displayed in order of R_FIFO31-24, R_FIFO23-16, R_FIFO15-8 and R_FIFO7-0. Data volume of reception FIFO is decremented by 4.

When R_MODE is 1.

By 2 bytes(half word) read access to R_FIFO15-0, data in reception FIFO is displayed in order R_FIFO7-0 of and R_FIFO15-8. Data volume of reception FIFO is decremented by 2.

By 4 bytes(word) read access to R_FIFO31-0, data in reception FIFO is displayed in order of R_FIFO7-0, R_FIFO15-8, R_FIFO23-16 and R_FIFO31-24. Data volume of reception FIFO is decremented by 4.

Attention) When there is no data in reception FIFO,don't read access to reception FIFO.
Attention) Reception FIFO size is 8 bytes. When reception FIFO is full, don't transfer.

17-4 Operation

This section provides information about S_Flash operation.

17-4-1 Read transfer procedure

It is indicated 30 bytes read transfer procedure by using read command(08h).

1st byte transfer	2nd byte transfer	3rd byte transfer	4th byte transfer	5th byte transfer	transfer from 6th byte
08h	A23-A16	A15-A8	A7-A0	Invalid	read data

Example that doesn't use DUMMY

(*1) Set "5" in UL_SIZE,"35" in T_SIZE, and "0" in MODE.

(*2) Set "{0x08,A23-A0}" in T_FIFO.

(*3) Set "0" in DUMMY.

(*4) Set "1" in ACT.

(*5) Set 31 bytes dummy data in T_FIFO and read access 30 bytes data to R_FIFO while checking the data volume of transmission FIFO and reception FIFO.

(*6) If FIN is changed "1", set "0" in FIN.

Example that use DUMMY

(*1) Set "5" in UL_SIZE,"35" in T_SIZE, and "0" in MODE.

(*2) Set "{0x08,A23-A0}" in T_FIFO.

(*3) Set "1" in DUMMY.

(*4) Set "1" in ACT.

(*5) Read access 30 bytes data to R_FIFO while checking the data volume of reception FIFO.

(*6) If FIN is changed "1", set "0" in FIN.

17-4-2 Write transfer procedure

It is indicated 30 bytes write transfer procedure by using write command(02h).

1st byte transfer	2nd byte transfer	3rd byte transfer	4th byte transfer	transfer from 5th byte
02h	A23-A16	A15-A8	A7-A0	write data

(*1) set "1" in NO_READ, set "34" in T_SIZE, set "0" in MODE.

(*2) Set "{0x02,A23-A0}" in T_FIFO.

(*3) Set "0" in DUMMY.

(*4) Set "1" in ACT.

(*5) Set 30 bytes data in T_FIFO. Note the data volume of transmission FIFO.

(*6) If FIN is changed "1", set "0" in FIN.

18 ADC

18-1 Introduction

This chapter provides the brief summary of ADC in the sections below:

- Brief Summary of ADC Block
- ADC Features

18-1-1 Brief Summary of ADC Block

This ADC block has an AD converter (ADC) which can support up to 6 ch input. AD conversion engine has 10 bits (with 8 bit accuracy) successive comparison converter. It also has Sample hold circuit which enables application system designing without using any sample hold circuit outside the LSI.

It also has AD channel control function which enables up to 6 ch analog input control in ADC only with the register setup. There are two different methods to use a channel: Conversion method with an input source fixed to a single channel, and another conversion method using multiple channels in succession for AD conversion.

One of the characteristics of this ADC block is having 4 different AD conversion modes. The most suitable mode for an application can be selected by the program.

ADC has an ADC interrupt. In an AD converter, a flag which indicates the completion of AD conversion is communicated to an interrupt control block as an A/D converter interrupt factor. Therefore, it can generate ADC interrupt which is caused by the completion of the AD conversion.

Software-standby and Hardware-standby of the Power saving function are also supported. Thus, the standby control is enabled by this Power saving function.

Table 18-1 Signal used with ADC

Signal	I/O	Function
ADC Clock	Input	Clock for ADC. System clock is supplied
ADC Standby	Input	For ADC standby control
System Bus	Bus	For ADC control
AN5	Input	Input signal for ADC-Ch5
AN4	Input	Input signal for ADC-Ch4
AN3	Input	Input signal for ADC-Ch3
AN2	Input	Input signal for ADC-Ch2
AN1	Input	Input signal for ADC-Ch1
AN0	Input	Input signal for ADC-Ch0

18-1-2 ADC Features

- Single mode

Executes single channel AD conversion once

- Continuous single mode

Repeats single channel AD conversion

- Scan mode

Executes AD conversion for all channels that are targeted for AD conversion only once

- Continuous scan mode

Repeats AD conversion for all channels that are targeted for AD conversion

18-2 Functional Overview

This chapter provides the outline of functional blocks used for the ADC configuration in a section below:

- ADC Circuit Configuration

18-2-1 ADC Circuit Configuration

ADC circuit configuration is as shown in Figure 18-1 below:

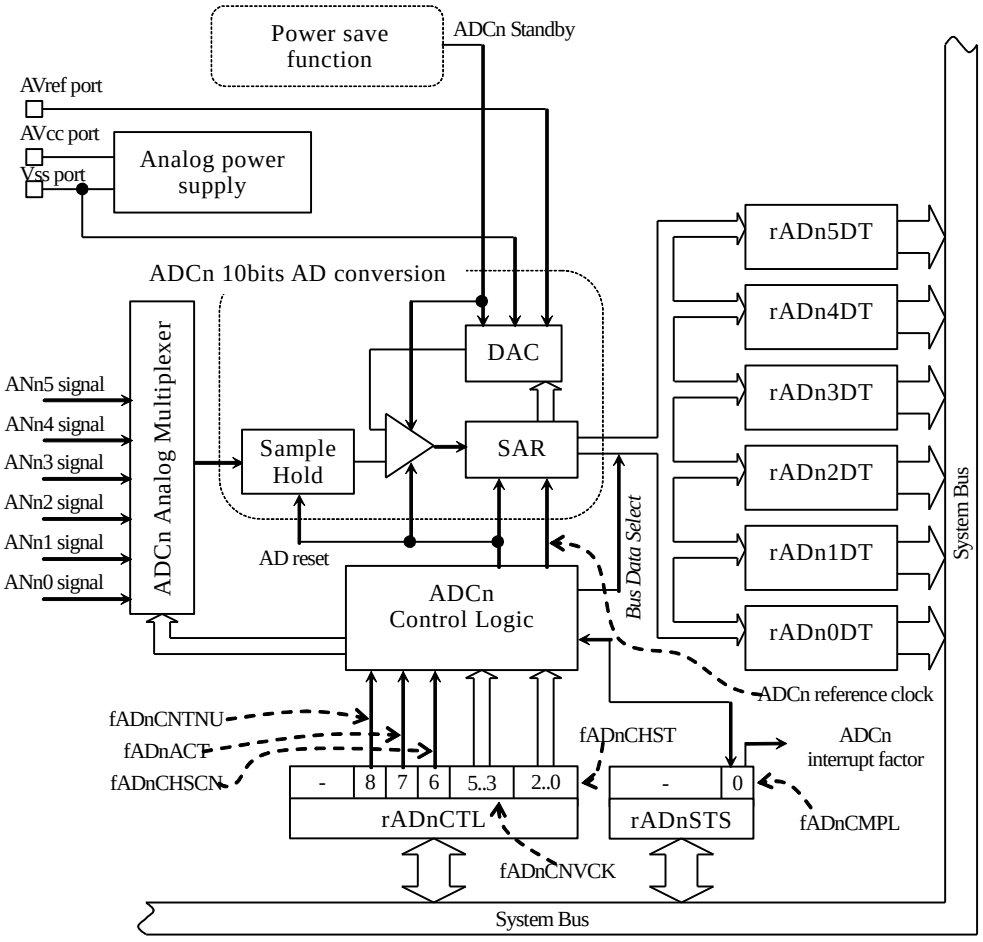


Figure 18-1 ADC Configuration

18-3 Programmer's Model

This chapter provides necessary information for the setup using ADC registers along with microcontrollers in the sections below:

- About the programmer's model
- Summary of ADC registers
- Register Descriptions

18-3-1 About the programmer's model

ADC block base address is not fixed and varies by the system implementation.

Please refer to the *System specification* for more information about the base address.

Offset is fixed for all registers.

18-3-2 Summary of ADC registers

ADC block is controlled by the registers mentioned in Table 18-2

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 18-2 ADC registers

Offset	Abbreviation	Attribute	Register name	Initial value
0x00	rAD0DT	R/W	ADC-Ch0 data register	0x0000_0000
004	rAD1DT	R/W	ADC-Ch1 data register	0x0000_0000
0x08	rAD2DT	R/W	ADC-Ch2 data register	0x0000_0000
0x0C	rAD3DT	R/W	ADC-Ch3 data register	0x0000_0000
0x10	rAD4DT	R/W	ADC-Ch4 data register	0x0000_0000
0x14	rAD5DT	R/W	ADC-Ch5 data register	0x0000_0000
0x28	rADCTL	R/W	ADC control register	0x0000_0000
0x2C	rADSTS	R/W	ADC status register	0x0000_0000
0x30	rADCSMPL	R/W	ADC sample control register	0x0000_0000
0x34	rADCSTBY	R/W	ADC standby register	0x0000_0000

18-3-3 Register Descriptions

18-3-3-1 ADC-Ch0 Data Register : rAD0DT

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD0DT9	fAD0DT8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fAD0DT7	fAD0DT6	fAD0DT5	fAD0DT4	fAD0DT3	fAD0DT2	fAD0DT1	fAD0DT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : fAD0DT9-0
AD conversion result at ADC channel 0.

Attention) There will be some unevenness of accuracy found in the lower 2 bits of a 10 bit data.

18-3-3-2 ADC-Ch1 Data Register : rAD1DT

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD1DT9	fAD1DT8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fAD1DT7	fAD1DT6	fAD1DT5	fAD1DT4	fAD1DT3	fAD1DT2	fAD1DT1	fAD1DT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : fAD1DT9-0
AD conversion result at ADC channel 1.

Attention) There will be some unevenness of accuracy found in the lower 2 bits of a 10 bit data.

18-3-3-3 ADC-Ch2 Data Register : rAD2DT

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD2DT9	fAD2DT8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fAD2DT7	fAD2DT6	fAD2DT5	fAD2DT4	fAD2DT3	fAD2DT2	fAD2DT1	fAD2DT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : fAD2DT9-0
AD conversion result at ADC channel 2.

Attention) There will be some unevenness of accuracy found in the lower 2 bits of a 10 bit data.

18-3-3-4 ADC-Ch3 Data Register : rAD3DT

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD3DT9	fAD3DT8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fAD3DT7	fAD3DT6	fAD3DT5	fAD3DT4	fAD3DT3	fAD3DT2	fAD3DT1	fAD3DT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : fAD3DT9-0
AD conversion result at ADC channel 3.

Attention) There will be some unevenness of accuracy found in the lower 2 bits of a 10 bit data.

18-3-3-5 ADC-Ch4 Data Register : rAD4DT

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD4DT9	fAD4DT8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fAD4DT7	fAD4DT6	fAD4DT5	fAD4DT4	fAD4DT3	fAD4DT2	fAD4DT1	fAD4DT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : fAD4DT9-0
AD conversion result at ADC channel 4.

Attention) There will be some unevenness of accuracy found in the lower 2 bits of a 10 bit data.

18-3-3-6 ADC-Ch5 Data Register : rAD5DT

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD5DT9	fAD5DT8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fAD5DT7	fAD5DT6	fAD5DT5	fAD5DT4	fAD5DT3	fAD5DT2	fAD5DT1	fAD5DT0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit9-0 : fAD5DT9-0
AD conversion result at ADC channel 5.

Attention) There will be some unevenness of accuracy found in the lower 2 bits of a 10 bit data.

18-3-3-7 ADC Control Register : rADCTL

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	fAD CNTNU	fADACT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fAD CHSCN	fAD CNVCK2	fAD CNVCK1	fAD CNVCK0	fAD CHST3	fAD CHST2	fAD CHST1	fAD CHST0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit9 : fADCNTNU (ADC Continuous Conversion Flag)
Sets ADC conversion continuity in the AD conversion.
0: Discontinuous AD conversion
1: Continuous AD conversion

Remarks) if “1” is selected, it will continue conversion. Which means even if the conversion object channel’s AD conversion is complete, ADC will not stop and repeat the conversion.
If “0” is selected, it will be discontinuous AD conversion. In this case, once conversion target channel’s AD conversion is complete, ADC will stop.
ADC’s AD conversion operation mode shall be set in a combination with fADCHSCN.

Bit8 : fADCACT (ADC Activate Flag)
Controls the AD conversion operation of an ADC
0: AD conversion stop
1: AD conversion start and convert

Remarks) by setting “1” to fADCACT by the program, ADC’s AD conversion will start and will maintain “1” during AD conversion until “0” is set by the program.
Once AD conversion is complete, it will change to “0”. However, AD conversion will continue until “0” is set for fADACT by the program if it is in the continuous conversion mode.
ADC will stop immediately by setting “0” to fADACT by the program.

Bit7 : fADCHSCN (ADC Channel Scan Flag)
Sets Channel / Scan for ADC’s AD conversion
0: AD conversion is executed only for the specific channels

1: AD conversion is executed by scanning each channel

Remarks) Channel scan is executed when “1” is set. In this case, AD conversion is executed for all object channels that are set by fADCHST.

When “0” is set, channel scan will not be executed.

ADC’s AD conversion operation mode shall be set in a combination with fADCNTNU.

Bit6-4 : fADCNVCK (ADC Converter Base Clock Select Flag)
Selects ADC reference clock
ADC will execute AD conversion by the ADC conversion reference clock (12+fADCSMPL+1) cycle time determined by this fADCNVCK. (Setting value of fADCSMPL: 18-3-3-9 ADC Sample Control Register : rADCSMPL).

* When making changes in this register, be sure to have “0” in fADCACT. It is prohibited to make changes when fADCACT is set to “1”.

0: ADC Clock [Hz] / 2
1: ADC Clock [Hz] / 4
2: ADC Clock [Hz] / 8
3: ADC Clock [Hz] / 16
4: ADC Clock [Hz] / 32
5: ADC Clock [Hz] / 64
6-7: Setting prohibited

Bit3-0 : fADCHST3-0 (ADC Channel Setting)
Sets AD conversion target channels in ADC
Either in a Single mode or Continuous single mode (fADCHSCN=0)
0: Channel 0
1: Channel 1
2: Channel 2
3: Channel 3
4: Channel 4
5: Channel 5

Either in a Scan mode or Continuous scan mode (fADCHSCN=1)
0: Channel 0
1: Channel 0 ~ 1
2: Channel 0 ~ 2
3: Channel 0 ~ 3
4: Channel 0 ~ 4
5: Channel 0 ~ 5

18-3-3-8 ADC Status Register : rADSTS

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	fADCMPL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit0 : fADCMPL (ADC Conversion Completion Flag)
Indicates the completion of AD conversion of the ADC
At READ
0: AD conversion in progress in ADC
1: Completed a series of AD conversion in ADC

Remarks) In the Single mode or Continuous single mode, it will become “1” when the AD conversion is complete for conversion target channels. In the Scan mode or Continuous scan mode, it will become “1” when AD conversion is complete for all scan conversion target channels.

At WRITE
0: fADCMPL is not impacted
1: fADCMPL is cleared

Remarks) this fADCMPL is communicated to the Interrupt control block of the Interrupt function as an ADC interrupt factor.

Attention 1) even if an interrupt is generated with the cause of this fADCMPL, it will not be cleared automatically. It will not be cleared automatically by reading out the fADCMPL as well.

Please execute fADCMPL clear from the program.

Attention 2) fADCMPL has ADC initialization function along with AD conversion completion monitoring function. Thus, if this fADCMPL is cleared by the program in the middle of AD conversion process, the series of AD conversion in progress may derive unexpected conversion result.

Do not execute fADCMPL clear with a program during AD conversion (when fADACT is “1”).

18-3-3-9 ADC Sample Control Register : rADCSMPL

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	fADC SMPL7	fADC SMPL6	fADC SMPL5	fADC SMPL4	fADC SMPL3	fADC SMPL2	fADC SMPL1	fADC SMPL0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit7-0 : fADCSMPL

Sample time setting.

Sample time = ADC reference clock cyle [T] × (fADCSMPL + 2.5)

Sample time designation is determined by driver output impedance for the analog input pin.

It is necessary to meet the requirement of “Sampling time > tA” at this point as well as satisfying Sample hold time. Please refer to the datasheet for more information about the sample hold time.

In case of adding Decoupling capacitor to a VR pin.

Rimp	tA
1	117
3.3	118
10	118
33	119
100	124
330	141
1000	199
3300	433
10000	1150
33000	3600
100000	10785

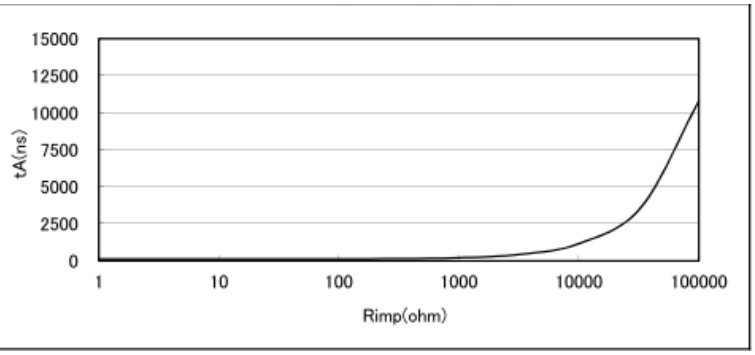


Figure 18-2

Approximation formula for this is

tA = 7.62*14.0*(1100+Rimp)/1000

In case of not adding Decoupling capacitor to a VR pin (or not having any VR pins).

Rimp	tA
1	332
3.3	335
10	332
33	335
100	336
330	346
1000	380
3300	540
10000	1183
33000	3600
100000	10622

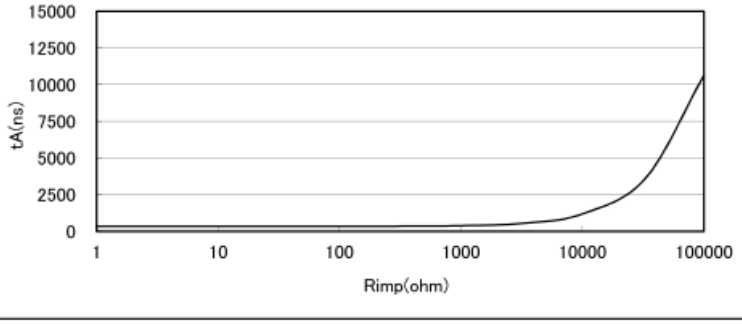


Figure 18-3

Approximation formula for this is

tA = 7.62*13.5*(3260+Rimp)/1000

Rimp = Output impedance of a driver for the analog input pin

tA = Analog input acquiring timing

Ex:

When ADC Clock = 160 MHz, Set 5 for fADCNVCK to have 2.5 MHz for AD conversion reference clock. (AD conversion reference clock must be 5 MHz or less when there is no Decoupling capacitor attached to the VR pin) In this case, the cycle becomes approximately 400 ns.

When Rimp = 100 kΩ, it will require 10622 ns as the sampling time based on the table above. Therefore, it is safe to set 25 for the fADCSMPL.

* Be sure to satisfy the sample hold time as well.

18-3-3-10 ADC Standby Register : rADCSTBY

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	fADCMPL
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

bit 0 : ADCMPL

A standby control bit for a single ADC

0: Normal

1: Standby

Attention) AD conversion block is the only portion to be in the standby setting. Control logic circuit setting of AD conversion does not change to the standby along with it.

18-4 Operation

This chapter provides information about ADC setting flow.

18-4-1 ADC Function Description

18-4-1-1 ADC Channel Control

ADC can be used as 6 channels in total; From Channel 5 ~ Channel0.

Actual AD conversion engine is one, but the channels selected as ADC conversion source channels in Analog multiplexer block are treated as ADC AD conversion objects. Correspondence with each channels and signals are as follows:

Table 18-3 : ADC channel and conversion object signals

Channel	Signal name
ADC Channel 5	AN5
ADC Channel 4	AN4
ADC Channel 3	AN3
ADC Channel 2	AN2
ADC Channel 1	AN1
ADC Channel 0	AN0

Channel setting for this Analog multiplexer block is executed with fADCHST of the ADC control register.

fADCHST and channel settings are as follows:

Table 18-4 : fADCHST and Object channels

fADCHST	Single mode and Continuous single mode (fADCHSCN=0)	Scan mode and Continuous scan mode (fADCHSCN=1)
5	Channel 5	Channel 0 ~ 5
4	Channel 4	Channel 0 ~ 4
3	Channel 3	Channel 0 ~ 3
2	Channel 2	Channel 0 ~ 2
1	Channel 1	Channel 0 ~ 1
0	Channel 0	Channel 0

18-4-1-2 ADC Conversion Time Setting

AD conversion time can be selected using fADCNVCK in the ADC control register.

- fADCNVCK: 5

$$\text{ADC time [sec]} \quad \frac{1}{\text{ADC Clock [Hz]}} \times 2^6 \times 16$$

- fADCNVCK: 4

$$\text{ADC time [sec]} \quad \frac{1}{\text{ADC Clock [Hz]}} \times 2^5 \times 16$$

- fADCNVCK: 3

$$\text{ADC time [sec]} \quad \frac{1}{\text{ADC Clock [Hz]}} \times 2^4 \times 16$$

- fADCNVCK: 2

$$\text{ADC time [sec]} \quad \frac{1}{\text{ADC Clock [Hz]}} \times 2^3 \times 16$$

- fADCNVCK: 1

$$\text{ADC time [sec]} \quad \frac{1}{\text{ADC Clock [Hz]}} \times 2^2 \times 16$$

- fADCNVCK: 0

$$\text{ADC time [sec]} \quad \frac{1}{\text{ADC Clock [Hz]}} \times 2^1 \times 16$$

ADC has a sample hold circuit to be used for the input signal. Please refer to the Datasheet for more information about the hold time in this sample hold circuit.

Please implement a countermeasure externally if this sample hold time is not enough for the application systems due to the system clock configuration.

18-4-1-3 ADC Conversion Mode

ADC has four different AD conversion modes to choose from for the application as follows:

- Single mode

Executes AD conversion only once at a channel which is set at ADC controller register fADCHST. ADC controller register fADACT becomes “0” at the same time as the completion of AD conversion and ADC will stop. Furthermore, ADC status register fADCMPL becomes “1” to monitor the completion of AD conversion. This fADCMPL is communicated to the Interrupt control block as an ADC interrupt factor.

- Continuous single mode

Repeats AD conversion in a channel which is set at ADC control register fADCHST.

Along with one-time AD conversion, ADC status register fADCMPL becomes “1” and monitors the completion of AD conversion. This fADCMPL is communicated to an Interrupt control block as an ADC interrupt factor.

However, ADC control register fADACT does not become “0” and ADC will continue to start AD conversion over and over.

AD conversion can stop by setting “0” to fADACT by the program.

- Scan mode

Executes AD conversion only once for all channels that are AD conversion objects by ADC control register fADCHST

The conversion will start from Channel 0, per each channel.

At the time of completing the AD conversion for the last channel, ADC control register fADACT becomes “0” and ADC will stop. Furthermore, ADC status register fADCMPL will become “1” and monitors the completion of AD conversion. This fADCMPL is communicated to the Interrupt control block as an ADC interrupt factor.

- Continuous scan mode

Repeats AD conversion of all channels that are targeted for ADC conversion in ADC control register fADCHST.

The conversion will start from Channel 0, per each channel.

At the time of completing the AD conversion for the last channel, ADC status register fADCMPL becomes “1” and monitors the completion of AD conversion. This fADCMPL is communicated to the Interrupt control block as an ADC interrupt factor.

However, the ADC control register fADACT does not become “0” and ADC will start repeating AD conversion for all AD conversion target channels.

AD conversion will stop by setting “0” to fADACT by the program.

AD conversion mode selection can be made by fADCNTNU and fADCHSCN settings in ADC control register as shown below:

Table 18-5: Conversion mode and fADCNTNU & fADCHSCN setup

ADC AD conversion mode	fADCNTNU	fADCHSCN
Single mode	0	0
Continuous single mode	1	0
Scan mode	0	1
Continuous scan mode	1	1

18-4-1-4 ADC Interrupt Function

Interrupt function which considers AD conversion completion as the ADC interrupt factor is supported. ADC status register fADCMPL is communicated to the Interrupt control block as an ADC interrupt factor.

This fADCMPL cannot be cleared automatically even if an interrupt is generated by having this fADCMPL as the interrupt factor. This cannot be cleared automatically by reading out fADCMPL. fADCMPL clear must be executed via program.

Condition to have “1” for fADCMPL will vary by the AD conversion mode:

- Single mode

fADCMPL will become “1” at the time of AD conversion completion in a channel which is set by ADC control register fADCHST.

- Continuous single mode

fADCMPL will become “1” when AD conversion is completed even once in a channel which is set by ADC control register fADCHST (But AD conversion will continue).

- Scan mode

fADCMPL will become “1” when all AD conversion of the AD conversion target channels are complete in ADC control register fADCHST.

- Continuous scan mode

fADCMPL will become “1” when all AD conversion of the AD conversion target channels are generally completed in ADC control register fADCHST (Bu AD conversion will continue)

18-4-1-5 ADC Standby Function

ADC corresponds to Software standby and Hardware standby by the Power save function.
Once either of the standby requests is generated in the Power save function, it will immediately plunge into the standby mode.

The standby state will be released when Standby state release request is generated by the Power save function. Please initialize the ADC by using the software immediately after releasing this standby state.

A certain amount of time will be required from the beginning of release process as ADC lockup time at the time of releasing the standby state. Please refer to the Datasheet for more information regarding the lockup time.

18-4-2 AD Converter Operation

18-4-2-1 Single mode operation

Be sure to clear ADC status register fADCMPL before activating the ADC. This process initializes the ADC.

Next, set ADC channel control and ADC conversion time, set ADC's AD conversion mode to the Single mode. To utilize ADC interrupt, set it to ADC interrupt enable state from the Interrupt control block.

AD conversion will begin once all of the above settings are complete and ADC control register fADACT is set to "1". After this, in order to maintain the AD conversion accuracy, ADC control register and ADC status register shall not be updated without any discretion.

Pin potential sampling by Sample hold circuit will start from the fifth cycle phase after beginning of the AD conversion, and the potential will be on hold after [2.5+ fADCSMPL] cycles. AD conversion by 10 bits conversion engine is executed at the 12 cycle period after the hold.

The AD conversion value is obtained in the last cycle phase and this AD conversion value will be stored into ADC*DT which corresponds to AD conversion target channels. Also, at the completion of this final cycle phase, ADC status register fADCMPL will become "1" and monitor the completion of AD conversion. At the same time, ADC control register fADACT will become "0" and ADC will stop.

fADCMPL is communicated to the Interrupt function as the ADC interrupt factor. If it is set to ADC interrupt enable status by the Interrupt control block, ADC interrupt will occur from the point where fADCMPL becomes "1".

Figure 18-4 shows the operation in the Single mode.

Note) 0 ~ 5 can fit into "*"

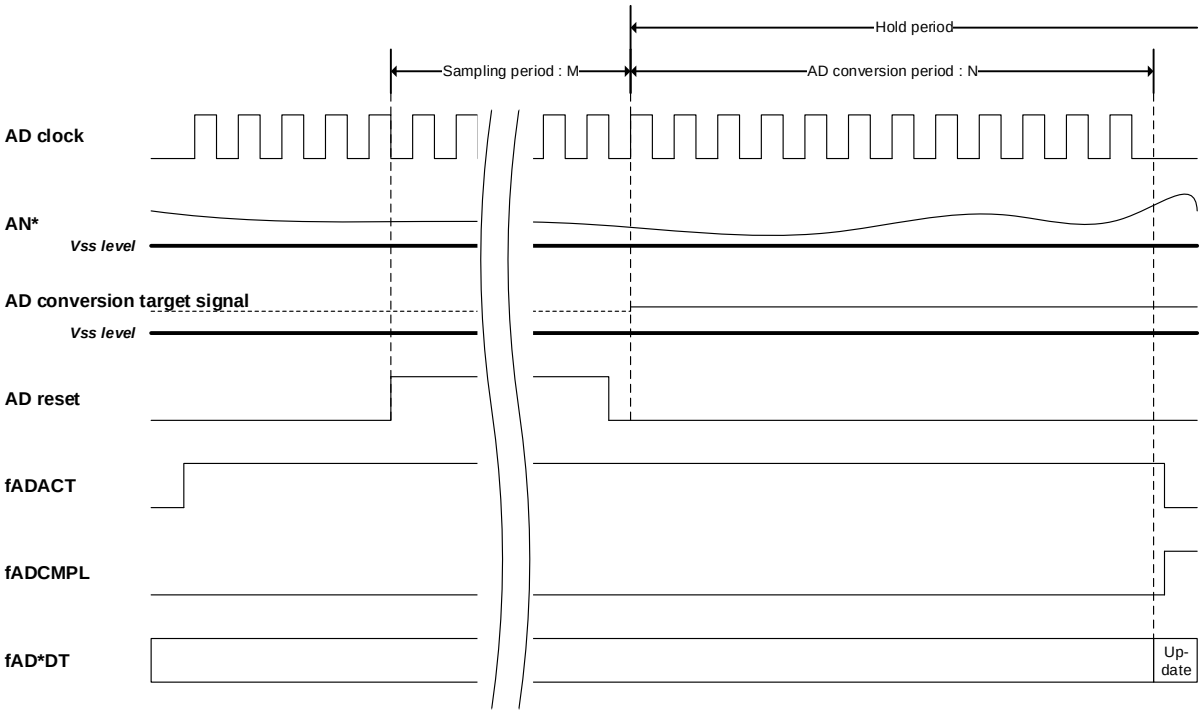


Figure 18-4 AD conversion operation in Single mode

18-4-2-2 Continuous Single Mode Operation

- From the beginning to continuation of AD conversion in the ADC continuous single mode

At first, clear ADC status register fADCMPL before activating the ADC. This process initializes the ADC.

Next, set ADC channel control and ADC conversion time, and set ADC's AD conversion mode to Continuous single mode. To utilize the ADC interrupt, set ADC interrupt enable state by the Interrupt control block.

AD conversion will start by setting ADC control register fADACT to "1" after completing the above settings. After this, in order to maintain the AD conversion accuracy, ADC control register and ADC status register shall not be updated without any discretion.

Pin potential sampling by Sample hold circuit will start from the fifth cycle phase after beginning of the AD conversion, and the potential will be on hold after [2.5+ fADCSMPL] cycles. AD conversion by 10 bits conversion engine is executed at the 12 cycle period after the hold.

The AD conversion value is obtained in the following cycle (first cycle of the next AD conversion) phase and this AD conversion value will be stored into ADC*DT which corresponds to AD conversion target channels. Also, at the completion of this final cycle phase, ADC status register fADCMPL will become "1" and monitor the completion of one AD conversion. ADC control register fADACT is remained as "1" an ADC continues the operation.

ADC will treat this phase as first cycle of a new AD conversion and begin new AD conversion.

fADCMPL is communicated to the Interrupt function as the ADC interrupt factor. If it is set to ADC interrupt enable status by the Interrupt control block, ADC interrupt will occur from the point where fADCMPL becomes "1".

Beginning of the AD conversion through the continuation in the Continuous single mode is shown in Figure 18-5

Note) 0 ~ 5 can fit into "*"

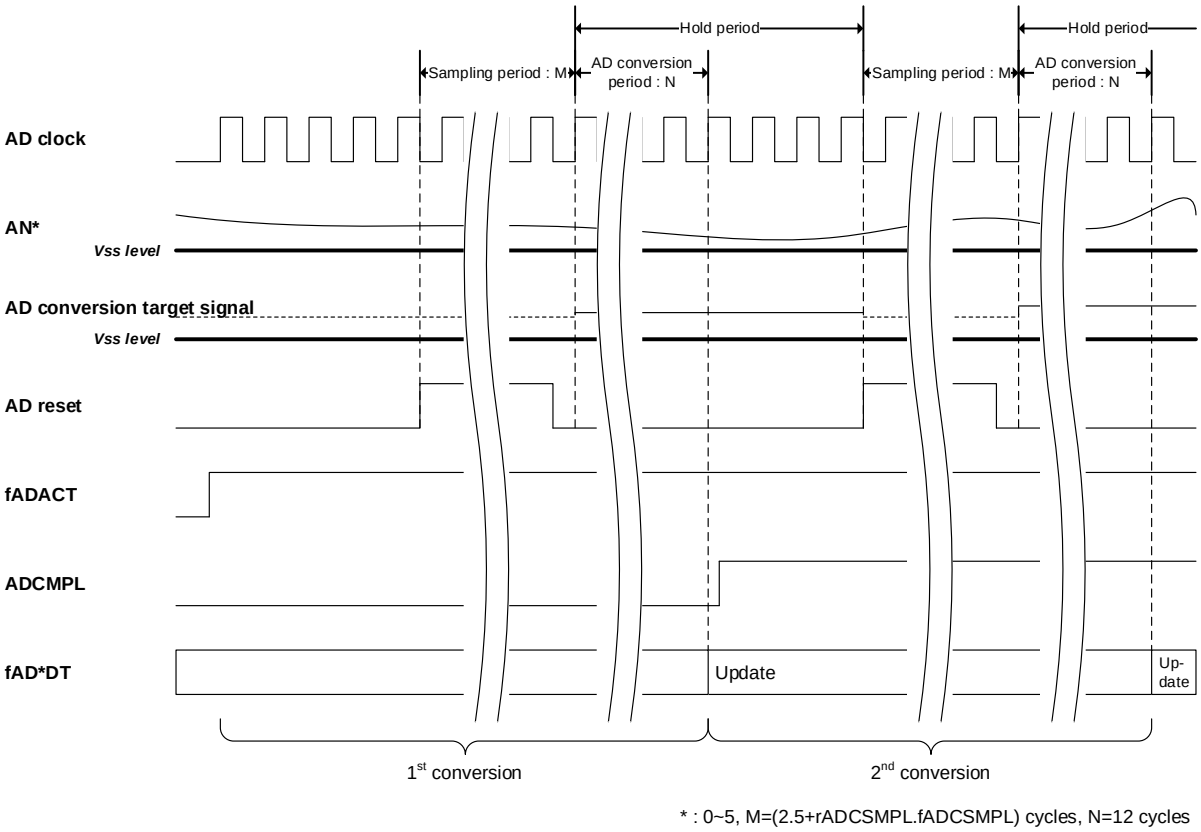


Figure 18-5 AD conversion operation in Continuous single mode

- AD conversion stop in ADC continuous single mode

As a method of stopping the AD conversion in the Continuous single mode, there is a way to forcibly set “0” to ADC control register fADACT by the program.

As shown in Figure 18-6, When “0” is set to ADC control register fADACT at the timing of (*1) by the program, ADC will immediately stop at the next phase.

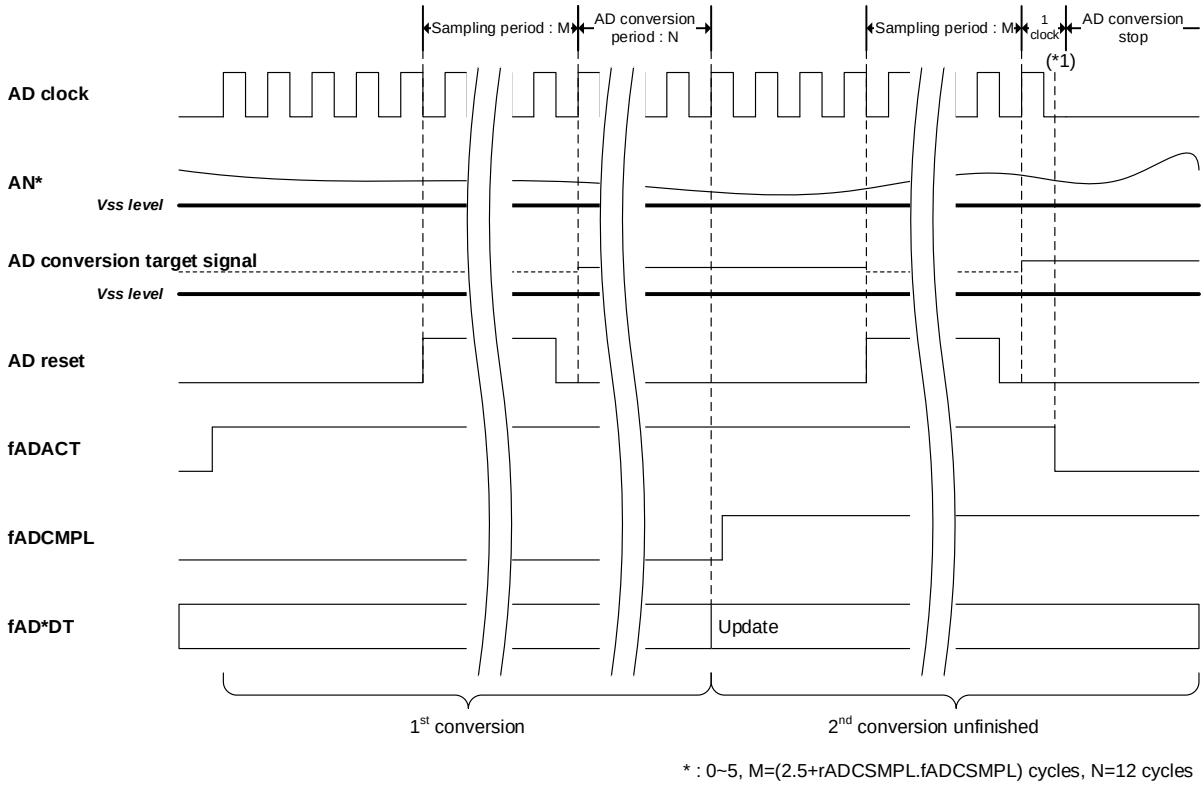


Figure 18-6 conversion stop operation in Continuous single mode

18-4-2-3 Scan Mode Operation

First, clear ADC status register rADCMPL before activating the ADC. This process initializes the ADC.

Next, set ADC channel control and ADC conversion time, and set AD conversion into the Scan mode. To utilize the ADC interrupt, set ADC interrupt enable state by the Interrupt control block.

AD conversion will start by setting ADC control register fADACT to “1” after completing the above settings. After this, in order to maintain the AD conversion accuracy, ADC control register and ADC status register shall not be updated without any discretion.

In Scan mode, first AD conversion target channel is Channel 0 and will continue to Channel 1, Channel 2 and so on.

Pin potential sampling by Sample hold circuit will start from the fifth cycle phase after beginning of the AD conversion, and the potential will be on hold after [2.5+ fADCSMPL] cycles. AD conversion by 10 bits conversion engine is executed at the 12 cycle period after the hold.

The AD conversion value is obtained in the following cycle (first cycle of the next AD conversion or the last cycle) phase and this AD conversion value will be stored into ADC*DT which corresponds to AD conversion target channels.

If AD conversion target channel is the last channel which is set by the Channel control function, this phase becomes the last cycle, and at the time of this phase completion, ADC status register fADCMPL will become “1” and monitor the completion of this sequential AD conversion. ADC control register fADACT will become “0” and ADC will stop.

fADCMPL will be communicated to the Interrupt function as ADC interrupt factor. If the status is set as ADC interrupt enable status by the Interrupt control block, ADC interrupt will occur from the point where fADCMPL becomes “1”.

Scan mode operation is shown in Figure 18-7.

Note) 0 ~ 5 can fit into “*”

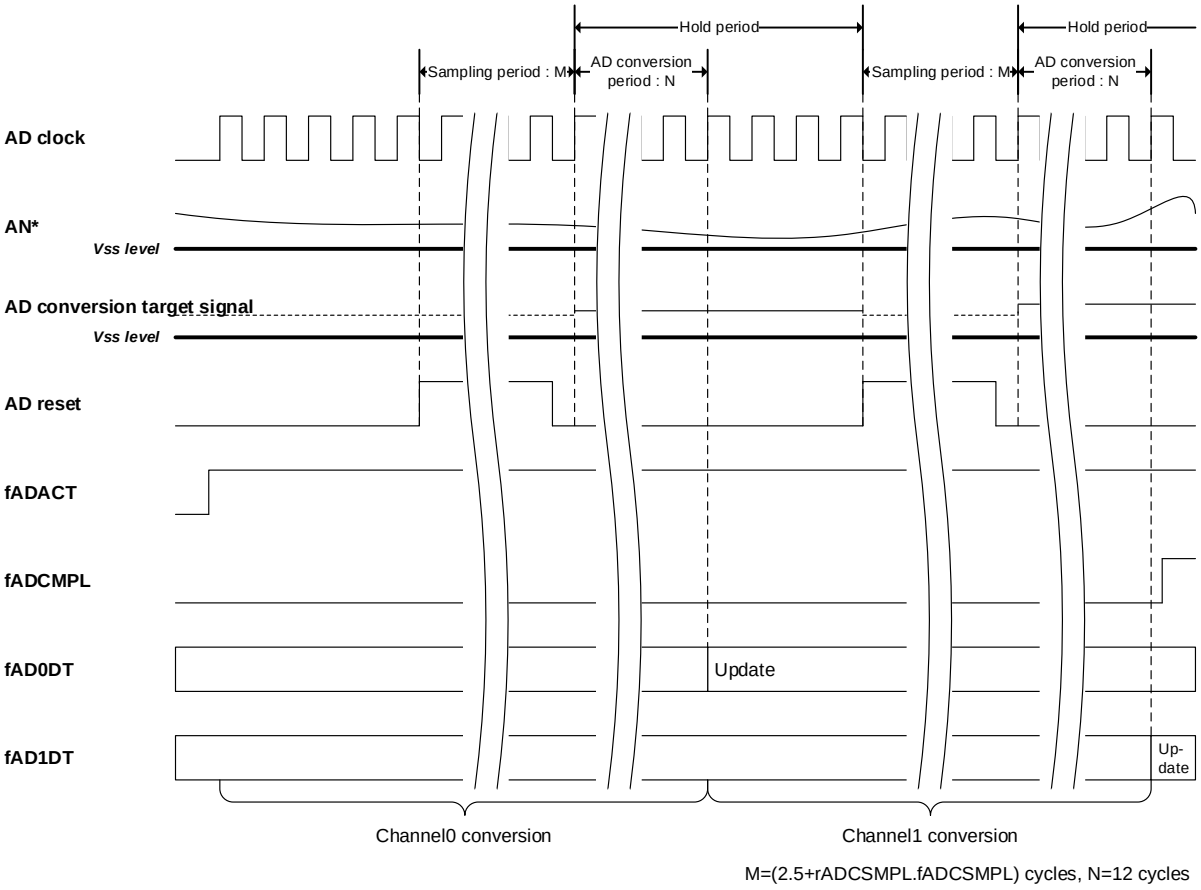


Figure 18-7 AD conversion operation in Scan mode (Channels 0~1)

18-4-2-4 Continuous Scan Mode Operation

- From the beginning to the continuation of AD conversion in ADC Continuous scan mode

At first, clear ADC status register fADCMPL before activating the ADC. This process initializes the ADC.

Next, set ADC channel control and ADC conversion time, and set ADC's AD conversion mode to Continuous scan mode. To utilize the ADC interrupt, set ADC interrupt enable state by the Interrupt control block.

AD conversion will start by setting ADC control register fADACT to "1" after completing the above settings. After this, in order to maintain the AD conversion accuracy, ADC control register and ADC status register shall not be updated without any discretion.

In Scan mode, first AD conversion target channel is Channel 0 and will continue to Channel 1, Channel 2 and so on.

Pin potential sampling by Sample hold circuit will start from the fifth cycle phase after beginning of the AD conversion, and the potential will be on hold after [2.5+ fADCSMPL] cycles. AD conversion by 10 bits conversion engine is executed at the 12 cycle period after the hold.

The AD conversion value is obtained in the following cycle (first cycle of the next AD conversion) phase and this AD conversion value will be stored into ADC*DT which corresponds to AD conversion target channels.

If the AD conversion target channel is the last channel which was set by the Channel control function, ADC status register fADCMPL becomes "1" at the end of the first cycle phase and monitors the completion of this sequential AD conversion. However, in this case, ADC control register fADACT remains as "1" and ADC will continue its operation.

fADCMPL will be communicated to the Interrupt function as ADC interrupt factor. If the status is set as ADC interrupt enable status by the Interrupt control block, ADC interrupt will occur from the point where fADCMPL becomes "1".

After these preparation period mentioned above, ADC will start AD conversion for a new channel from the first cycle of a phase.

Note) 0 ~ 5 can fit into "*"

Operation flow from the beginning to the continuation of AD conversion in Continuous scan mode is shown in Figure 18-8.

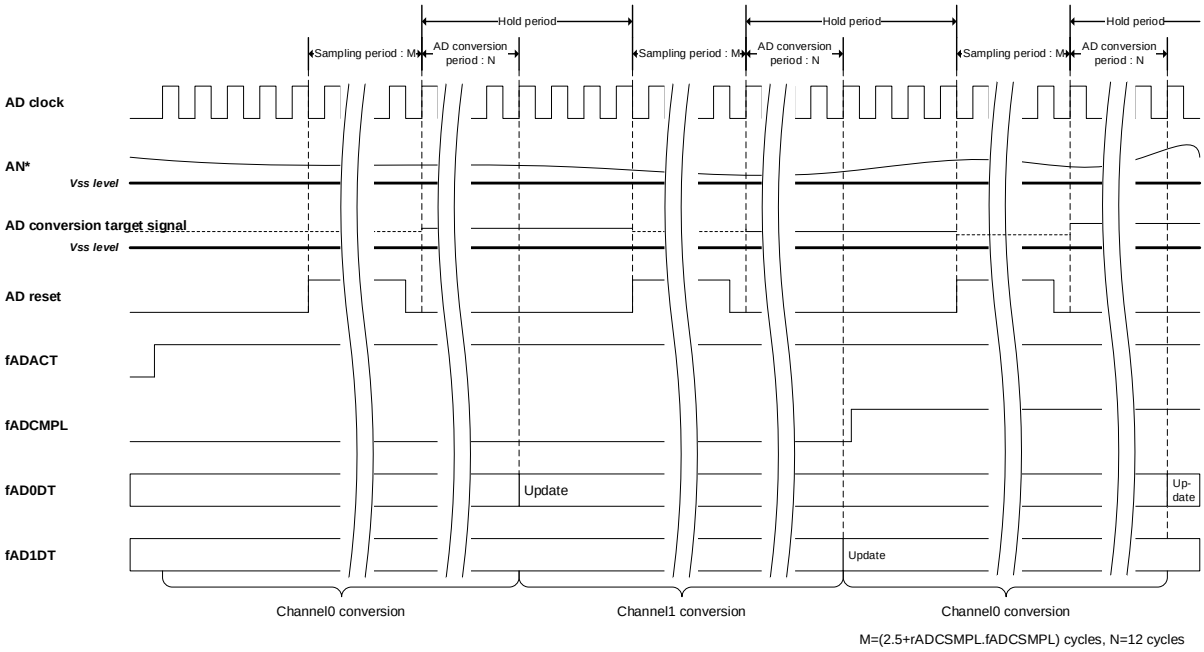


Figure 18-8 AD conversion operation in Continuous scan mode (From Channel 0 ~ 1)

- AD conversion stop in ADC continuous scan mode

As a method of stopping the AD conversion in the Continuous scan mode, there is a way to forcibly set “0” to ADC control register fADACT by the program.

As shown in
Figure 18-9, When “0” is set to ADC control register fADACT at the timing of (*1) by the program, ADC will immediately stop at the next phase.

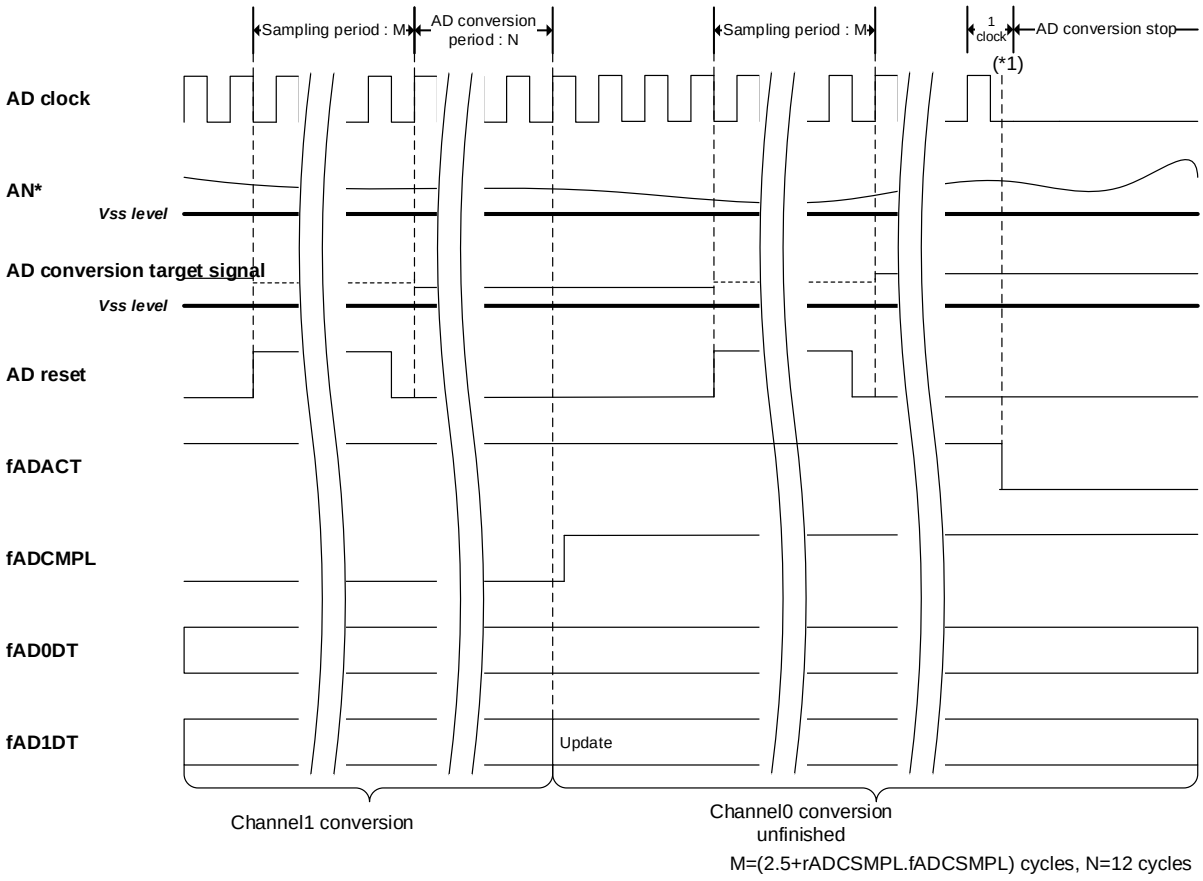


Figure 18-9 AD conversion stop operation in Continuous scan mode (Channels 0 ~ 2)

18-4-3 Operating Precautions

ADC status register fADCMPL has a function to monitor AD conversion completion along with ADC initialization. Therefore, if this fADCMPL is cleared during the AD conversion by the program, the sequence of AD conversion in process may arrive at an unexpected conversion result.

Be sure not to execute fADCMPL clear by the program during AD conversion process (When ADC control register fADACT is in “1”).

ADC has a Sample hold circuit for Input signals.

Please implement a countermeasure externally if this sample hold time is not enough for the application systems due to the system clock configuration.

19 XMC

19-1 Introduction

This chapter provides the brief summary of XMC in the sections below:

- Brief Summary of XMC

19-1-1 Brief Summary of XMC

This model has XMC with address bus 21 bits and data bus 16 bits.

The external space that can be accessed in this XMC is two kinds of areas of External-0 Area(External-0 Alias Area) and External-1 Area in the memory system of this model. Independent chip select signals (NCS0,NCS1) are allocated for these two areas, XMC generates the chip select signals for the access corresponding to the area.(Please refer to Table 19-1)

Table 19-1 Each area and Chip select signal

Area	Chip select signal
External-1	NCS1
External-0	NCS0

Moreover, the access with the best cycle for the each device allocated to each area becomes possible because an independent wait cycle and various strobe signals to each area can be set. In addition, it is also possible to correspond to the weight signal from the outside.

Table 19-2 introduces the signal used in XMC.

Table 19-2 Signal used with XMC

Signal name	I/O	Function
NCS1	Output	Strobe signal that shows that External-1 Area is object of access.
NCS0	Output	Strobe signal that shows that External-0 Area is object of access.
NWRENWRL	Output	Signal of using combinedly of following NWRE and NWRL. NWRE: Write strobe signal when data writing is done to external space. NWRL: Write strobe signal when data writing is done to external space and shows that lower byte (EXD7-EXD0) of the data bus is effective. Whether this signal functions as NWRE or NWRL is selected automatically in each area by setting the XMC data access control.
NHBNWRH	Output	Signal of using combinedly of following NWRH and NHB. NHB: Strobe signal that shows that higher byte (EXD15-EXD8) of the data bus is an access object. NWRH: Write strobe signal when data writing is done to external space and shows that higher byte (EXD15-EXD8) of the data bus is effective. Whether this signal functions as NHB or NWRH is selected automatically in each area by setting the XMC data access control.
NRD	Output	Read strobe signal when data from external space is read.
EXD15~0	I/O	External data bus.
EXA20~1	Output	External address bus.
NLBEXA0	Output	Signal of using combinedly of following EXA0 and NLB. NLB: Strobe signal that shows that lower byte (EXD7-EXD0) of the data bus is an access object. EXA0: External address bus 0. Whether this signal functions as NLB or EXA0 is selected automatically in each area by setting the XMC data access control.

19-2 Functional Overview

This chapter provides overview of functional blocks used for the XMC configuration in the sections below:

- XMC Circuit Configuration

19-2-1 XMC Circuit Configuration

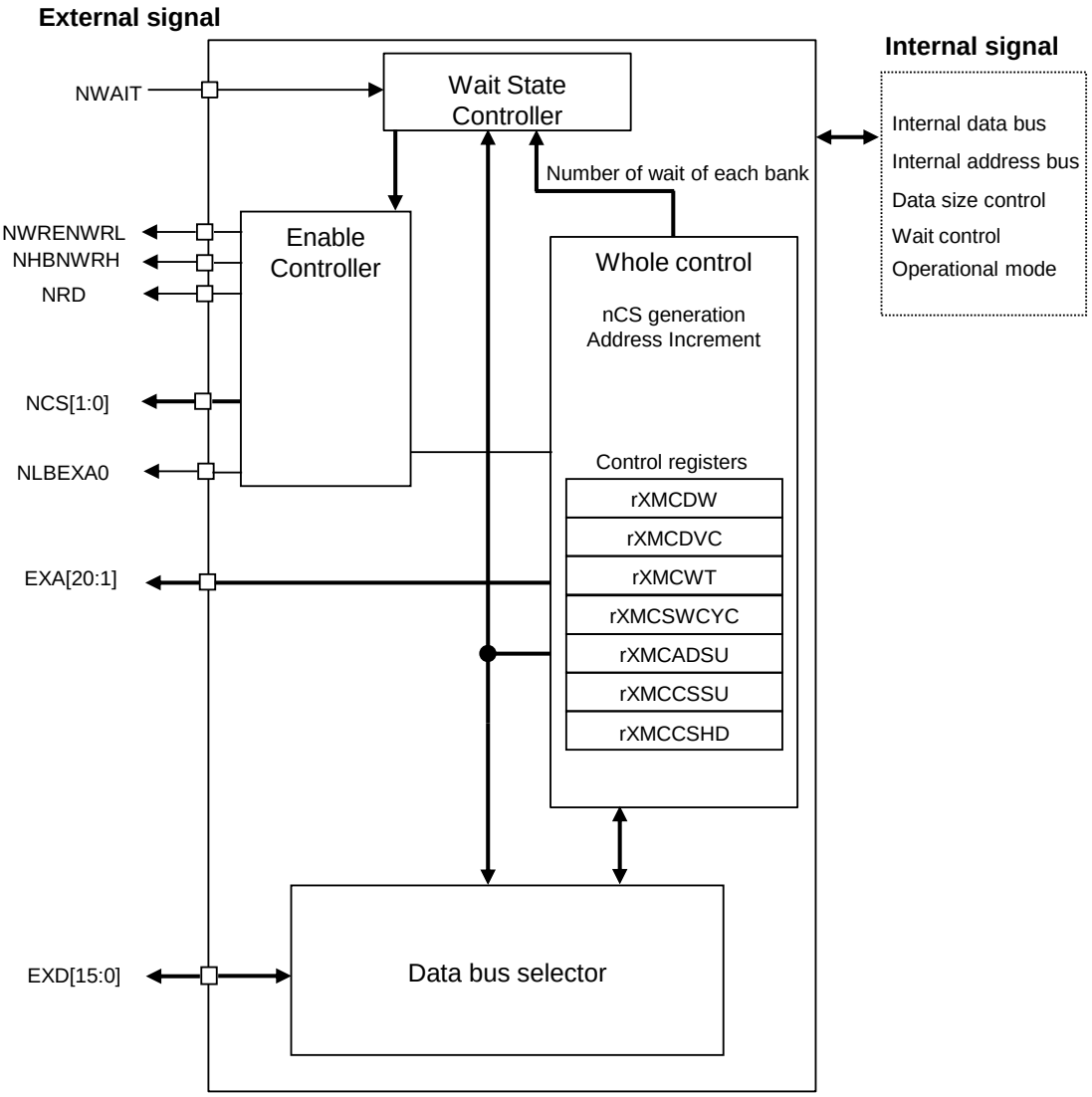


Figure 19-1

Note) : This model doesn't correspond to external signal NWAIT.

19-3 Programmer's Model

This chapter contains XMC register information and other information necessary for the register setup using microcontrollers in the sections below:

- About the programmer's model
- Summary of XMC registers
- Register Descriptions

19-3-1 About the programmer's model

XMC base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

19-3-2 Summary of XMC registers

UART is controlled by the registers shown in Table 19-3 below.

Remarks) Accessible addresses are the ones listed in between the range of "Base+0x0000" to "Base+0xFFFF". Do not access any other address.

Table 19-3 XMC registers

offset	abbreviation	attribute	content	initial value
0x0000	rXMCDW	R/W	XMC data width setting register	0x00000000
0x0004	rXMCDVC	R/W	XMC device mode setting register	0x00000000
0x0008	rXMCWT	R/W	XMC wait setting register	0x0000FFFF
0x0010	rXMCSWCYC	R/W	XMC area change cycle control register	0x00001111
0x0014	rXMCADSU	R/W	XMC address setup control register	0x00001111
0x0018	rXMCCSSU	R/W	XMC chip select setup control register	0x00001111
0x001C	rXMCCSHD	R/W	XMC chip select hold control register	0x00001111

19-3-3 Register Descriptions

19-3-3-1 XMC data width setting register : rXMCDW

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-		
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	fXMC3DW	fXMC2DW	fXMC1DW	fXMC0DW
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R

Bit3 : fXMC3DW : XMC External-3 Area data width setting

Width of the data bus of XMC's External-3 Area is set. About an initial value and the access attribute, refer to Table 19-4.
0 : the width of the data bus of External-3 Area becomes 16 bit.
1 : the width of the data bus of External-3 Area becomes 8 bit.

The function of external data bus EXD15-EXD0 terminal is different according to the setting of fXMC3DW, fXMC2DW, fXMC1DW, and fXMC0DW.

If fXMC3DW, fXMC2DW, fXMC1DW, and fXMC0DW everything is set to 1 (all of External-3 Area - External-0 Area are set to data bus 8 bit), EXD15-EXD8 of an external data bus becomes invalid on little endian and EXD7-EXD0 of an external data bus becomes invalid on big endian.
When either of one is set to 0(either of External-3 Area - Externla-0 Area is set to data bus 16 bit), all external data buses (EXD15-EXD0) become effective.

Table 19-4 Initial value and access attribute (n:3-0) of fXMCnDW

	External 16 bit	
	Initial value	Access attribute
fXMC3DW	0	Read/Write
fXMC2DW	0	Read/Write
fXMC1DW	0	Read/Write
fXMC0DW	0	Read Only

Bit2 : fXMC2DW : XMC External-2 Area data width setting
Except object of setting becomes External-2 Area, it is the same as “fXMC3DW : XMC External-3 Area data width setting” above.

Bit1 : fXMC1DW : XMC External-1 Area data width setting
Except object of setting becomes External-1 Area, it is the same as “fXMC3DW : XMC External-3 Area data width setting” above.

Bit0 : fXMC0DW : XMC External-0 Area data width setting
Except object of setting becomes External-0 Area, it is the same as “fXMC3DW : XMC External-3 Area data width setting” above. However, fXMC0DW becomes an read only bit, and the width of the data bus becomes 16 bit fixation

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2DW and fXMC3DW by an initial value.

19-3-3-2 XMC device mode setting register : rXMCDVC

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	fXMC3DVC	fXMC2DVC	fXMC1DVC	fXMC0DVC
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R

Bit3 : fXMC3DVC : XMC External-3 Area device mode setting

Access method to External-3 Area is set when the width of the data bus of XMC is set to 16 bits. When External-3 Area is accessed, various strobe signals which XMC generates are different by setting fXMC3DVC. Table 19-5 shows relation between setting value and strobe signal.
There is no influence on XMC by the setting to fXMC3DVC when the width of the data bus is set to 8 bits.

Table 19-5 Setting of fXMCnDVC and kind f strobe

fXMCnDVC	Write strobe signal	Read strobe signal	Byte select signal	Type of memory device
1	$\overline{WRH}$, $\overline{WRL}$	$\overline{RD}$	N.A.	Two of 8 bit memory device are used.
0	$\overline{WRE}$	$\overline{RD}$	$\overline{HB}$, $\overline{LB}$	16 bit memory device is used.

*n: 3~0

Bit2 : fXMC2DVC : XMC External-2 Area device mode setting
Except object of setting becomes External-2 Area, it is the same as “fXMC3DVC : XMC External-3 Area device mode setting” above.

Bit1 : fXMC1DVC : XMC External-1 Area device mode setting
Except object of setting becomes External-1 Area, it is the same as “fXMC3DVC : XMC External-3 Area device mode setting” above.

Bit0 : fXMC0DVC : XMC External-0 Area mode setting

Except object of setting becomes External-0 Area, it is the same as “fXMC3DVC : XMC External-3 Area device mode setting” above. However, the access attribute becomes as follows.

This bit becomes Read Only, and becomes the mode of 16 bit device fixation because External-0 Area functions as External 16 bit in this model.

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2DVC and fXMC3DVC by an initial value.

19-3-3-3 XMC wait setting register : rXMCWT

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	fXMC3WT7	fXMC3WT6	fXMC3WT5	fXMC3WT4	fXMC2WT7	fXMC2WT6	fXMC2WT5	fXMC2WT4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	fXMC1WT7	fXMC1WT6	fXMC1WT5	fXMC1WT4	fXMC0WT7	fXMC0WT6	fXMC0WT5	fXMC0WT4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	fXMC3WT3	fXMC3WT2	fXMC3WT1	fXMC3WT0	fXMC2WT3	fXMC2WT2	fXMC2WT1	fXMC2WT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fXMC1WT3	fXMC1WT2	fXMC1WT1	fXMC1WT0	fXMC0WT3	fXMC0WT2	fXMC0WT1	fXMC0WT0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Wait cycle inserted when it accesses an external area is set.

The setting is independently done to each External-0-3 Area.

Bit[31:28] ,[15:12] : fXMC3WT : XMC External-3 Area wait setting

The wait cycle to External-3 Area is set when accessing it. A set value and the wait cycle number are shown in Table 19-6. However please consider “Attension) : Restriction on setting wait cycle” before setting.

Bit[27:24] ,[11:8] : fXMC2WT : XMC External-2 Area wait setting

The wait cycle to External-2 Area is set when accessing it. A set value and the wait cycle number are shown in Table 19-6. However please consider “Attension) : Restriction on setting wait cycle” before setting.

Bit[23:20] ,[7:4] : fXMC1WT : XMC External-1 Area wait setting

The wait cycle to External-1 Area is set when accessing it. A set value and the wait cycle number are shown in Table 19-6. However please consider “Attension) : Restriction on setting wait cycle” before setting.

Bit[19:16] ,[3:0] : fXMC0WT : XMC External-0 Area wait setting

The wait cycle to External-0 Area is set when accessing it. A set value and the wait cycle number are shown in Table 19-6. However please consider “Attension) : Restriction on setting wait cycle” before setting.

Table 19-6 Set value and wait cycle of fXMCnWT

fXMCnWT(n:3-0)	Wait cycle
255	255 wait cycles
254	254 wait cycles
253	253 wait cycles
252	252 wait cycles
251	251 wait cycles
250	250 wait cycles
249	249 wait cycles
:	:
:	:
6	6 wait cycles
5	5 wait cycles
4	4 wait cycles
3	3 wait cycles
2	2 wait cycles
1	1 wait cycle
0	0 wait cycle

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2WT and fXMC3WT by an initial value.

19-3-3-4 XMC area change cycle control register : rXMCSWCYC

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	fXMC3SW5	fXMC3SW4	-	-	fXMC2SW5	fXMC2SW4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	fXMC1SW5	fXMC1SW4	-	-	fXMC0SW5	fXMC0SW4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	fXMC3SW3	fXMC3SW2	fXMC3SW1	fXMC3SW0	fXMC2SW3	fXMC2SW2	fXMC2SW1	fXMC2SW0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fXMC1SW3	fXMC1SW2	fXMC1SW1	fXMC1SW0	fXMC0SW3	fXMC0SW2	fXMC0SW1	fXMC0SW0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When the area to be accessed changes continuously, area change cycle inserted in addition to address setup cycle to the CS signal is set. Only once at the access that steps over the area, the fXMCnSW setting value of the previous area of the switch is added to the fXMCnADSU setting value of the following area.

The setting is independently done to each External-0-3 Area. please consider “Attention) : About the setting of fXMCnSW and fXMCnADSU” before setting.

Bit[29:28] ,[15:12] : fXMC3SW : XMC External-3 Area change cycle setting
The area change cycle to External-3 Area is set.
A set value and area change cycle number are shown in Table 19-7.

Bit[25:24] ,[11: 8] : fXMC2SW : XMC External-2 Area change cycle setting
The area change cycle to External-2 Area is set.
A set value and area change cycle number are shown in Table 19-7.

Bit[21:20] ,[7:4] : fXMC1SW : XMC External-1 Area change cycle setting
The area change cycle to External-1 Area is set.
A set value and area change cycle number are shown in Table 19-7.

Bit[17:16] ,[3:0] : fXMC0SW : XMC External-0 Area change cycle setting
The area change cycle to External-0 Area is set.
A set value and area change cycle number are shown in Table 19-7.

Table 19-7 Set value and area change cycle of fXMCnSW

fXMCnSW(n:3-0)	Area change cycle
63	63 cycles
62	62 cycles
61	61 cycles
:	:
:	:
2	2 cycles
1	1 cycle
0	0 cycle

Attention) : About the setting of fXMCnSW and fXMCnADSU

In this model, XMC higher byte select signal: NHB and XMC higher byte write signal NWRH are a terminal of using combinedly.

Therefore, when the area which functions in 8 bit device mode and the area which functions in 16 bit device mode are accessed continuously, please set the area change cycle control register (fXMCnSW) or address setup control register (fXMCnADSU) of both the areas to 1 or more to avoid misidentification by the NHB signal and the NWRH signal when the area is changed.

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2SW and fXMC3SW by an initial value.

19-3-3-5 XMC address setup control register : rXMCADSU

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	fXMC3ADSU5	fXMC3ADSU4	-	-	fXMC2ADSU5	fXMC2ADSU4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	fXMC1ADSU5	fXMC1ADSU4	-	-	fXMC0ADSU5	fXMC0ADSU4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	fXMC3ADSU3	fXMC3ADSU 2	fXMC3ADSU1	fXMC3ADSU0	fXMC2ADSU 3	fXMC2ADSU2	fXMC2ADSU 1	fXMC2ADSU0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fXMC1ADSU3	fXMC1ADSU2	fXMC1ADSU1	fXMC1ADSU0	fXMC0ADSU 3	fXMC0ADSU2	fXMC0ADSU1	fXMC0ADSU0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address setup cycle to the chip select signal is set.

please consider “Attention) : About the setting of fXMCnSW and fXMCnADSU” before setting.

Bit[29:28], [15:12] : fXMC3ADS : XMC External-3 Area address setup cycle setting
The address setup cycle to External-3 Area is set.
A set value and address setup cycle number are shown in Table 19-8.

Bit[25:24] ,[11:8] : fXMC2ADSU : XMC External-2 Area address setup cycle setting
The address setup cycle to External-2 Area is set.
A set value and address setup cycle number are shown in Table 19-8.

Bit[21:20] ,[7:4] : fXMC1ADSU : XMC External-1 Area address setup cycle setting
The address setup cycle to External-1 Area is set.
A set value and address setup cycle number are shown in Table 19-8.

Bit[17:16] ,[3:0] : fXMC0ADSU : XMC External-0 Area address setup cycle setting

The address setup cycle to External-0 Area is set.
A set value and address setup cycle number are shown in Table 19-8.

Table 19-8 Set value and address setup cycle of fXMCnADSU

fXMCnADSU (n:3-0)	Address setup cycle
63	63 cycles
62	62 cycles
61	61 cycles
:	:
:	:
2	2 cycles
1	1 cycle
0	0 cycle

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2ADSU and fXMC3ADSU by an initial value.

19-3-3-6 XMC chip select setup control register : rXMCCSSU

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	fXMC3CSSU5	fXMC3CSSU4	-	-	fXMC2CSSU5	fXMC2CSSU4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	fXMC1CSSU5	fXMC1CSSU4	-	-	fXMC0CSSU5	fXMC0CSSU4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	fXMC3CSSU3	fXMC3CSSU2	fXMC3CSSU1	fXMC3CSSU0	fXMC2CSSU3	fXMC2CSSU2	fXMC2CSSU1	fXMC2CSSU0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fXMC1CSSU3	fXMC1CSSU2	fXMC1CSSU1	fXMC1CSSU0	fXMC0CSSU3	fXMC0CSSU2	fXMC0CSSU1	fXMC0CSSU0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Chip select setup cycle of the chip select signal is set.

Bit[29:28] ,[15:12] : fXMC3CSSU : XMC External-3 Area chip select setup setting
The chip select setup cycle to External-3 Area is set.
A set value and chip select setup cycle number are shown in Table 19-9.

Bit[25:24] ,[11:8] : fXMC2CSSU : XMC External-2 Area chip select setup setting
The chip select setup cycle to External-2 Area is set.
A set value and chip select setup cycle number are shown in Table 19-9.

Bit[21:20] [7:4] : fXMC1CSSU : XMC External-1 Area chip select setup setting
The chip select setup cycle to External-1 Area is set.
A set value and chip select setup cycle number are shown in Table 19-9.

Bit[17:16] ,[3:0] : fXMC0CSSU : XMC External-0 Area chip select setup setting

The chip select setup cycle to External-0 Area is set.
A set value and chip select setup cycle number are shown in Table 19-9.

Table 19-9 Set value and chip select setup cycle of fXMCnCSSU

fXMCnCSSU (n:3-0)	Chip select setup cycle
63	63 cycles
62	62 cycles
61	61 cycles
:	:
:	:
2	2 cycles
1	1 cycle
0	0 cycle

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2CSSU and fXMC3CSSU by an initial value.

19-3-3-7 XMC chip select hold control register : rXMCCSHD

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	fXMC3CSHD5	fXMC3CSHD4	-	-	fXMC2CSHD5	fXMC2CSHD4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	fXMC1CSHD5	fXMC1CSHD4	-	-	fXMC0CSHD5	fXMC0CSHD4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	fXMC3CSHD3	fXMC3CSHD2	fXMC3CSHD1	fXMC3CSHD0	fXMC2CSHD3	fXMC2CSHD2	fXMC2CSHD1	fXMC2CSHD0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	fXMC1CSHD3	fXMC1CSHD2	fXMC1CSHD1	fXMC1CSHD0	fXMC0CSHD3	fXMC0CSHD2	fXMC0CSHD1	fXMC0CSHD0
Reset	0	0	0	1	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Chip select hold cycle of the chip select signal is set.

Bit[29:28] ,[15:12] : fXMC3CSHD : XMC External-3 Area chip select hold setting
The chip select hold cycle to External-3 Area is set.
A set value and chip select hold cycle number are shown in Table 19-10.

Bit[25:24] ,[11:8] : fXMC2CSHD : XMC External-2 Area chip select hold setting
The chip select hold cycle to External-2 Area is set.
A set value and chip select hold cycle number are shown in Table 19-10.

Bit[21:20] ,[7:4] : fXMC1CSHD : XMC External-1 Area chip select hold setting
The chip select hold cycle to External-1 Area is set.
A set value and chip select hold cycle number are shown in Table 19-10.

Bit[17:16] ,[3:0] : fXMC0CSHD : XMC External-0 Area chip select hold setting

The chip select hold cycle to External-0 Area is set.
A set value and chip select hold cycle number are shown in Table 19-10.

Table 19-10 Set value and chip select hold cycle of fXMCnCSHD

fXMCnCSHD (n:3-0)	Chip select hold cycle
63	63 cycles
62	62 cycles
61	61 cycles
:	:
:	:
2	2 cycles
1	1 cycle
0	0 cycle

Note) : This model corresponds to only 0 and 1 of XMC External Area. Please do not set the XMC External-2 and -3, and use fXMC2CSHD and fXMC3CSHD by an initial value.

19-4 Operation

This section provides information about XMC operation.

19-4-1 Function explanation of XMC

XMC aims to achieve a smooth access with external devices, and has the following function.

XMC data access control

XMC bus access timing control

XMC bus access direction monitor

Here, it introduces these functions.

19-4-1-1 XMC data access control

XMC data access control function decides width of the data bus of XMC and generates the strobe signal.

XMC has an external data bus of 16 bits. It is possible to select the external data bus of 8 bit or 16 bit with setting XMC data width setting register : rXMCDW.

Moreover, when the width of the external data bus is 16 bit bus, access control according to the bus size of connected device can be done by setting device mode. The device mode is XMC External-n Area device mode : fXMCnDVC(n:1-0) in XMC device mode setting register : rXMCDVC. An independent setting is possible by fXMCnDVC(n:1-0) in each area.

In XMC data access control function, various strobe signals based on the setting of both the width of the data bus and the device mode are generated, and they access the external data bus.

The setting of each register and the data access method are summarized in Table 19-11.

Table 19-11 XMC data access control (n:1-0)

Endian format	fXMCnDW	fXMCnDVC	Data bus that becomes effective	Write strobe signal	Read strobe signal	Byte select signal
Little Endian	1	1	EXD7~EXD0	NWRL	NRD	N.A.
		0				
	0	1	EXD15~EXD0	NWRH, NWRL	NRD	N.A.
		0		NWRE	NRD	NHB, NLB

* n: It is decided by the controlled area.

19-4-1-1-1 Device mode

When the width of the data bus of the area to be accessed by XMC is set to 16 bit, the device mode becomes effective.

The device mode is divided into two kinds of 8 bit device mode and 16 bit device mode.

<8 bit device mode>

This is a mode to be used when device of 8 bit data bus specification is chiefly connected to the external data bus of XMC 16 bit bus.

The device mode becomes 8 bit device mode when XMC External-n Area data width setting : FXMCnDW in XMC data width setting register : rXMCDW is set to 0 and XMC External-n Area device mode setting : fXMCnDVC in XMC device mode setting register : rXMCDVC is set to 1.

Two kinds of write strobe signals are generated as NWRH and NWRL. NWRH functions as a write strobe signal for the device connected to higher byte 8 bit (EXD15-EXD8) of the external data bus. NWRL functions as a write strobe signal for the device connected to lower byte 8 bit (EXD7-EXD0) of the external data bus. Please refer to “Figure 19-9” about details of the access.

<16 bit device mode>

This is a mode to be used when device of 16 bit data bus specification is chiefly connected to the external data bus of XMC 16 bit bus.

The device mode becomes 16 bit device mode when XMC External-n Area data width setting : FXMCnDW in XMC data width setting register : rXMCDW is set to 0 and XMC External-n Area device mode setting : fXMCnDVC in XMC device mode setting register : rXMCDVC is set to 0.

One kind of write strobe signals is generated as NWRE. Moreover, two kinds of byte select signals are generated as NHB and NLB to show the accessed bits of the external data bus. NHB functions as a byte select signal when higher byte 8 bit (EXD15-EXD8) of the external data bus is accessed. NLB functions as a byte select signal when lower byte 8 bit (EXD7-EXD0) of the external data bus is accessed. Please refer to “ ” about details of the access.

19-4-1-2 XMC bus access timing control

XMC can control access timing in detail when the bus is being accessed. The access timing which can be adjusted is as follows.

Basic access timing control

Wait cycle control

Area change cycle control

These are controlled independently to each area of XMC. Therefore, the access with appropriate cycles can be done for the device connected in each area.

19-4-1-2-1 Basic access timing control

XMC can set basic timing when the bus is being accessed. The item that can be set is as follows.

Address setup cycle control

Chip select setup cycle control

Setting of access mode

Chip select hold cycle control

Regarding Setting of access mode, it becomes a common control to all the area of XMC.

19-4-1-2-1-1Address setup cycle control

Setup time of the XMC address bus can be secured before the chip select signal is asserted.

The setup time is set independently in each area with XMC External-n Area address setup setting : fXMCnADSU in XMC address setup control register : rXMCADSU.

Please refer to “Table 19-8” in detail of the setting.

19-4-1-2-1-2Chip select setup cycle control

Setup time of the chip select signal can be secured for the period from asserting the chip select signal to beginning read cycle or write cycle or wait cycle.

The setup time is set independently in each area with XMC External-n Area chip select setup setting : fXMCnCSSU in XMC chip select setup control register : rXMCCSSU.

Please refer to “Table 19-9” in detail of the setting.

19-4-1-2-1-3Setting of access mode

XMC has long writing mode and short writing mode as access mode.

The access mode is a common control to all the area of XMC and it is decided according to ACCMD signal provided from the outside of XMC.

The relation of between ACCMD signal and access mode is summarized in Table 19-12.

Table 19-12 ACCMD signal and access mode

ACCMD signal	Access mode	Wait cycle that becomes indispensable	Number of effective programmable wait
high	Long writing mode	1 cycle	1 cycle or more
low	Short writing mode	2 cycle	2 cycle or more

This model is fixed to ACCMD=High (Long writing mode) in LSI.

<Long writing mode>

The write strobe signal and the data bus are asserted in the first wait cycle inserted on accessing bus, and they are negated in the write cycle. The indispensable wait cycle number is 1 cycle. Therefore, the effective programmable wait is 1 cycle or more, and the external wait is valid when the programmable wait is 0 cycle or more.

<Short writing mode>

The write strobe signal and the data bus are asserted in the next cycle of the first wait cycle inserted on accessing bus, and they are negated at the beginning of the write cycle. The indispensable wait cycle number is 2 cycles. Therefore, the effective programmable wait is 2 cycles or more, and the external wait is valid when the programmable wait is 1 cycle or more.

19-4-1-2-1-4Chip select hold cycle control

Hold time of the chip select signal can be secured for the period from ending read cycle or write cycle to negating the chip select signal.

The hold time is set independently in each area with XMC External-n Area chip select hold setting : fXMCnCSHD in XMC chip select hold control register : rXMCCSHD.

Please refer to “Table 19-10” in detail of the setting.

19-4-1-2-2 Wait cycle control

19-4-1-2-2-1Programmable wait

Wait cycles added to read cycle or write cycle are set by the program. The setting of wait cycles is respectively independently done to the areas (External-0 Area-External-1 Area) supported by XMC.

The wait cycle number is set in the XMC wait setting register : rXMCWT. The relation between a set value and the wait cycle number of rXMCWT is as follows.

However, please consider “Attension) : Restriction on setting wait cycle” before setting.

fXMCnWT(n:1-0)	Wait cycle
255	255 wait cycles
254	254 wait cycles
253	253 wait cycles
252	252 wait cycles
251	251 wait cycles
250	250 wait cycles
249	249 wait cycles
:	:
:	:
6	6 wait cycles
5	5 wait cycles
4	4 wait cycles
3	3 wait cycles
2	2 wait cycles
1	1 wait cycle
0	0 wait cycle

The control of external wait is added to this programmable wait, and the wait cycle inserted in XMC is actually decided.

Attension) : Restriction on setting wait cycle

Please consider below when the wait cycle is set.

The wait cycle which becomes indispensable is decided by “19-4-1-2-1-3Setting of access mode” on the write accessing. Therefore, the indispensable wait cycle number will be inserted by the automatic operation even if the wait cycle number set by the program is less than the indispensable wait cycle number on the write accessing. Please refer to “Table 19-12” about the indispensable wait cycle number on the write accessing.

19-4-1-2-2-2External wait

The wait signal NWAIT is supplied from the outside of XMC, and it inserts a further wait cycle into the wait cycle set by the programmable wait.

This model is not mounting this function.

19-4-1-2-3 Area change cycle control

The area change cycle can be secured in addition to the address setup cycle to the CS signal when the area to be accessed is consecutive and it changes.

The area change cycle is set independently in each area with XMC External-n Area area change cycle setting : fXMCnSW in XMC area change cycle control register : rXMCSWCYC.

Please refer to “Table 19-7” in detail of the setting.

19-4-1-3 XMC bus access direction monitor

This is a function to monitor the direction of accessing when XMC accesses the bus and the XMC bus access direction signal that indicates the direction of accessing is generated.

When the data writing to the external bus is done with XMC, the XMC bus access direction signal becomes high for the period when the chip select signal is active (low). Moreover, when the data reading from the external bus is done with XMC, the XMC bus access direction signal becomes low for the period when the chip select signal is active (low).

The XMC bus access direction signal becomes irregular for the period which the chip select signal is inactive (high) in. Please refer to the XMC bus access direction signal for the period when the chip select signal is active (low) when you use this XMC bus access direction monitor function.

This model is not mounting this function.

19-4-2 XMC’s operation

19-4-2-1 Accessing with 8 bit bus

This introduces details of the access when the memory device which has the data bus of 8 bit width is connected to this model.

This shows the example of connection to 8 bit bus in Figure 19-2

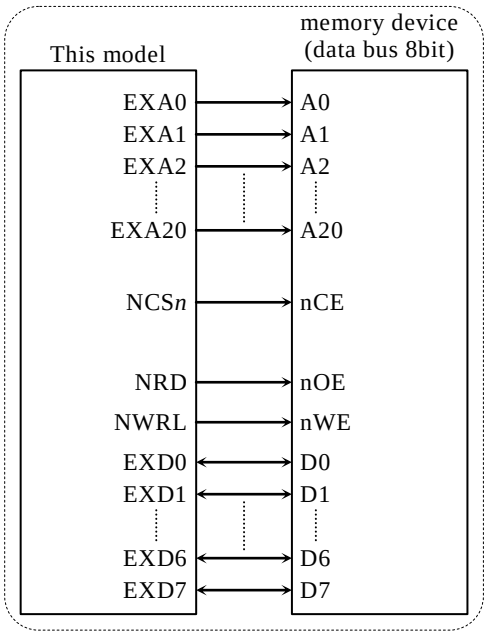
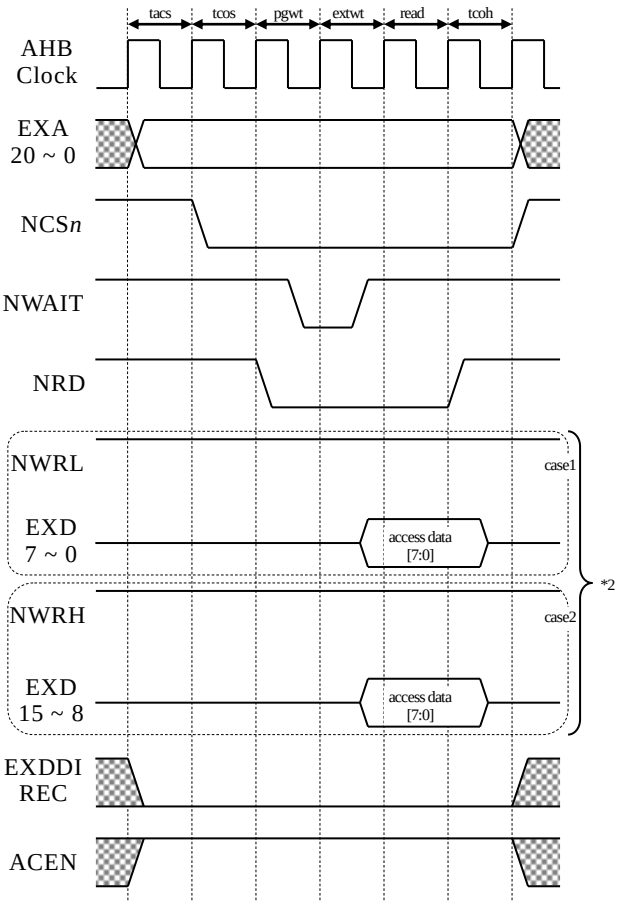


Figure 19-2 Example of connection to 8 bit bus

19-4-2-2 Byte accessing with 8 bit bus

Byte Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

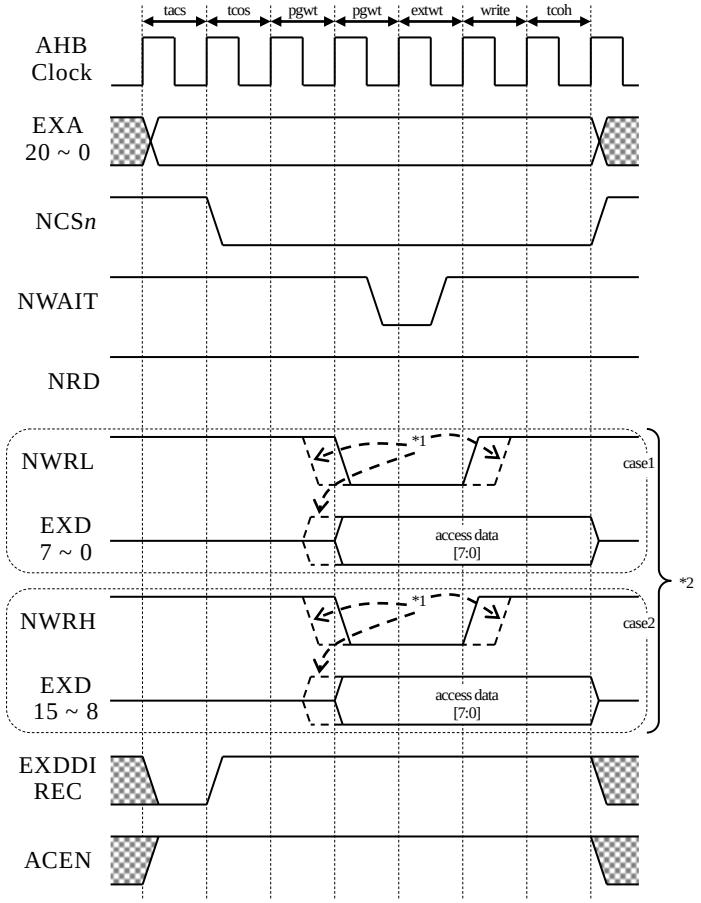
n: 3-0 (Area number of External-*n* Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-3 Byte read accessing with 8 bit bus

Byte Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

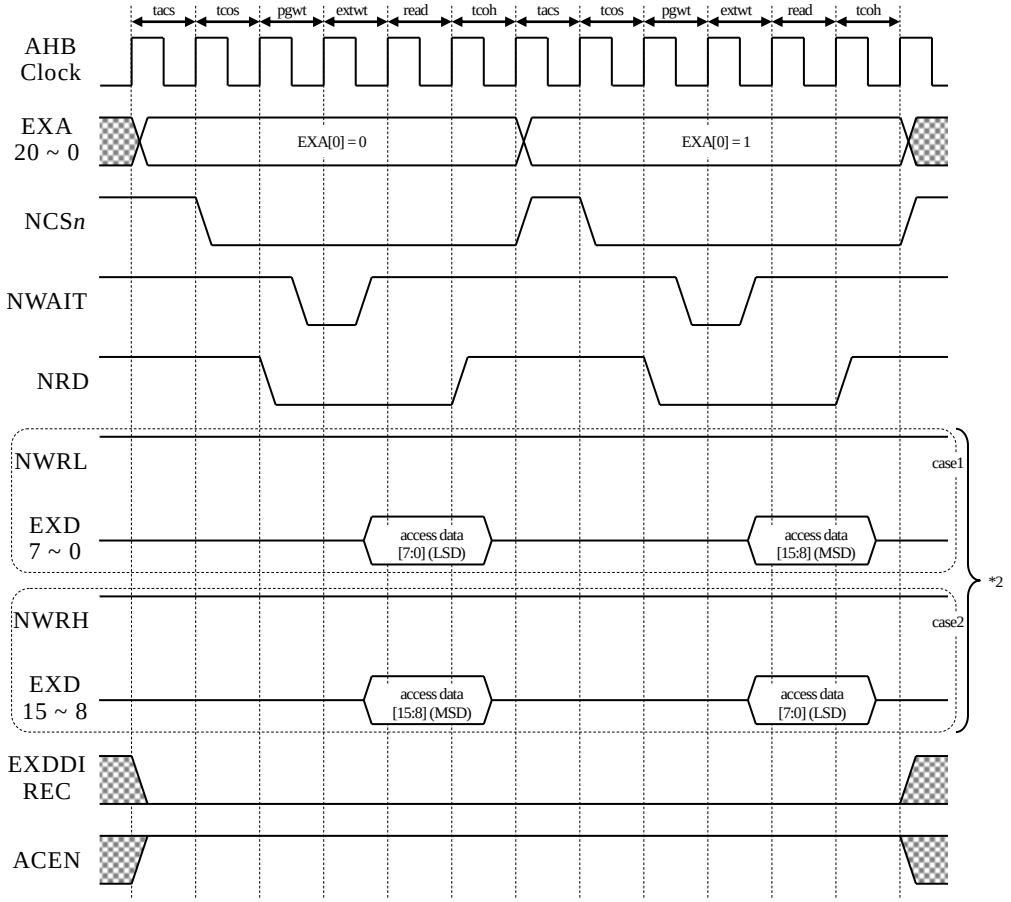
*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-4 Bye write accessing with 8 bit bus

19-4-2-3 Half word accessing with 8 bit bus

Half word Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

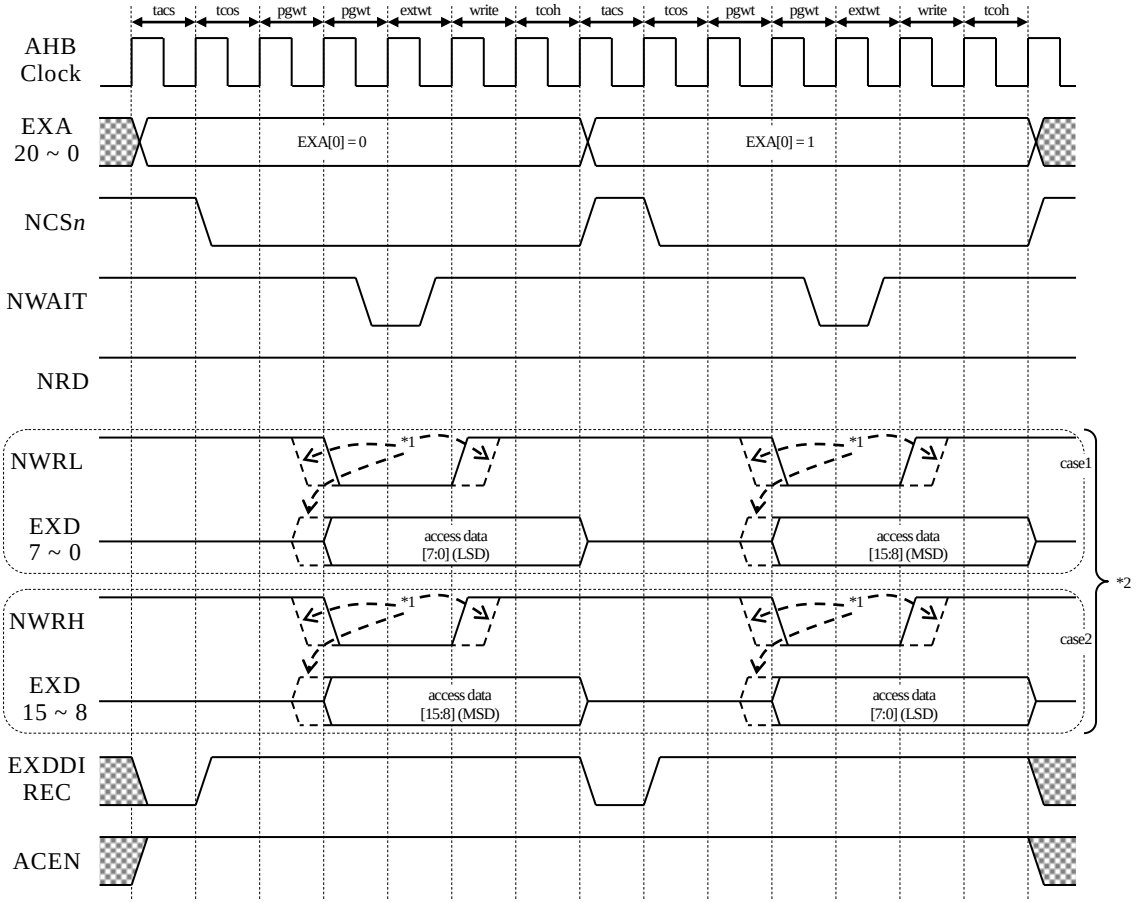
n: 3-0 (Area number of External-n Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-5 Half word read accessing with 8 bit bus

Half word Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

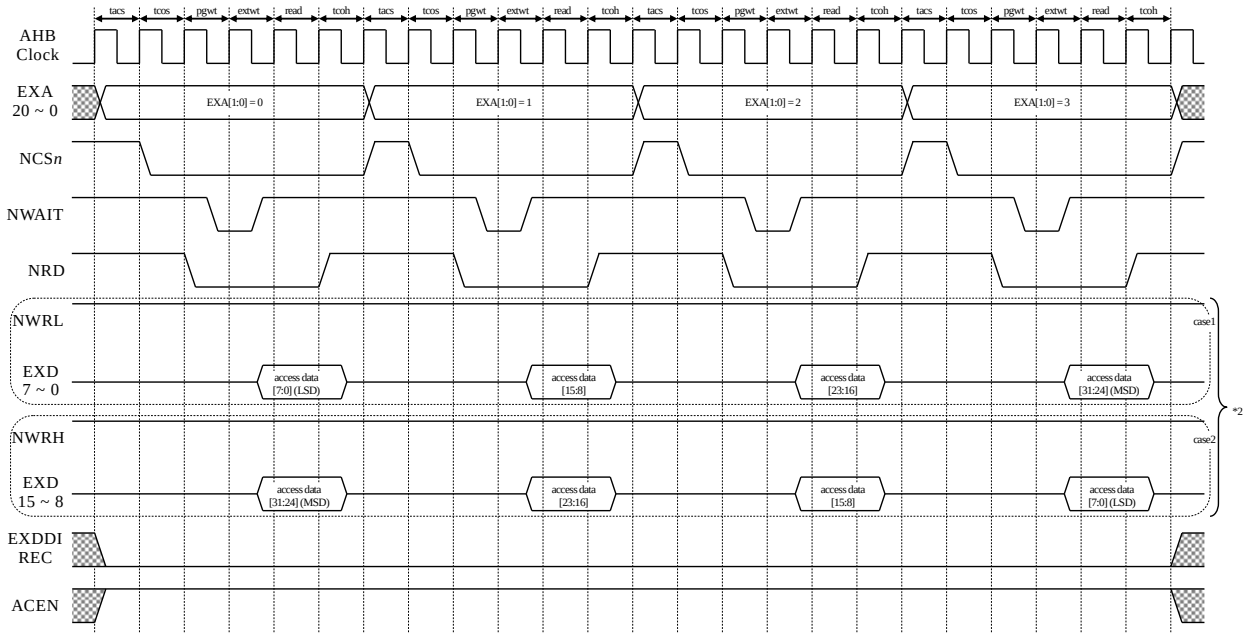
*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-6 Half word write accessing with 8 bit bus

19-4-2-4 Word accessing with 8 bit bus

Word Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

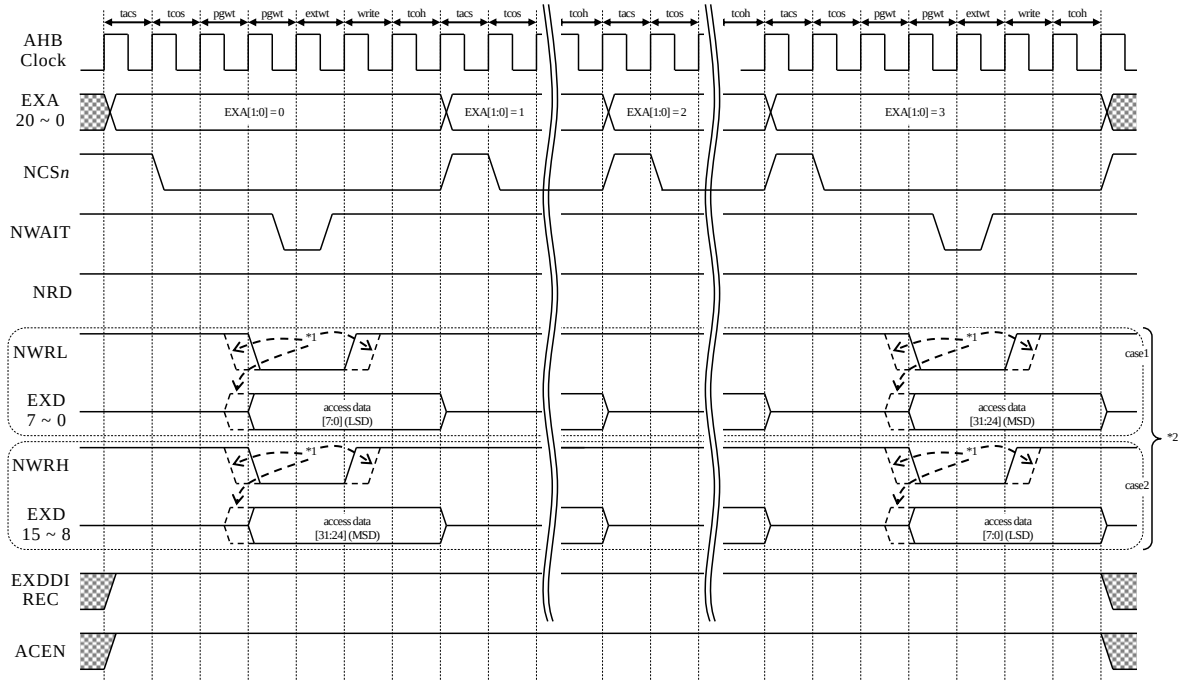
n: 3-0 (Area number of External-n Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-7 Word read accessing with 8 bit bus

Word Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-8 Word write accessing with 8 bit bus

19-4-2-5 Accessing with 16 bit bus at 8 bit device mode

This introduces details of the access when the memory devices which have the data bus of 8 bit width are connected to the 16 bit data bus of this model.

This shows the example of 8 bit device mode connection to 16 bit bus in Figure 19-9

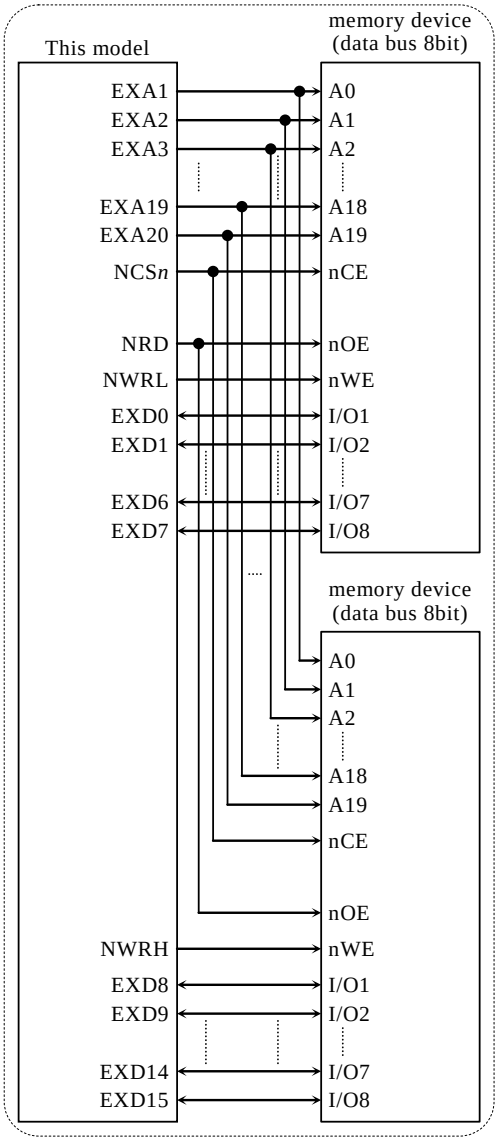
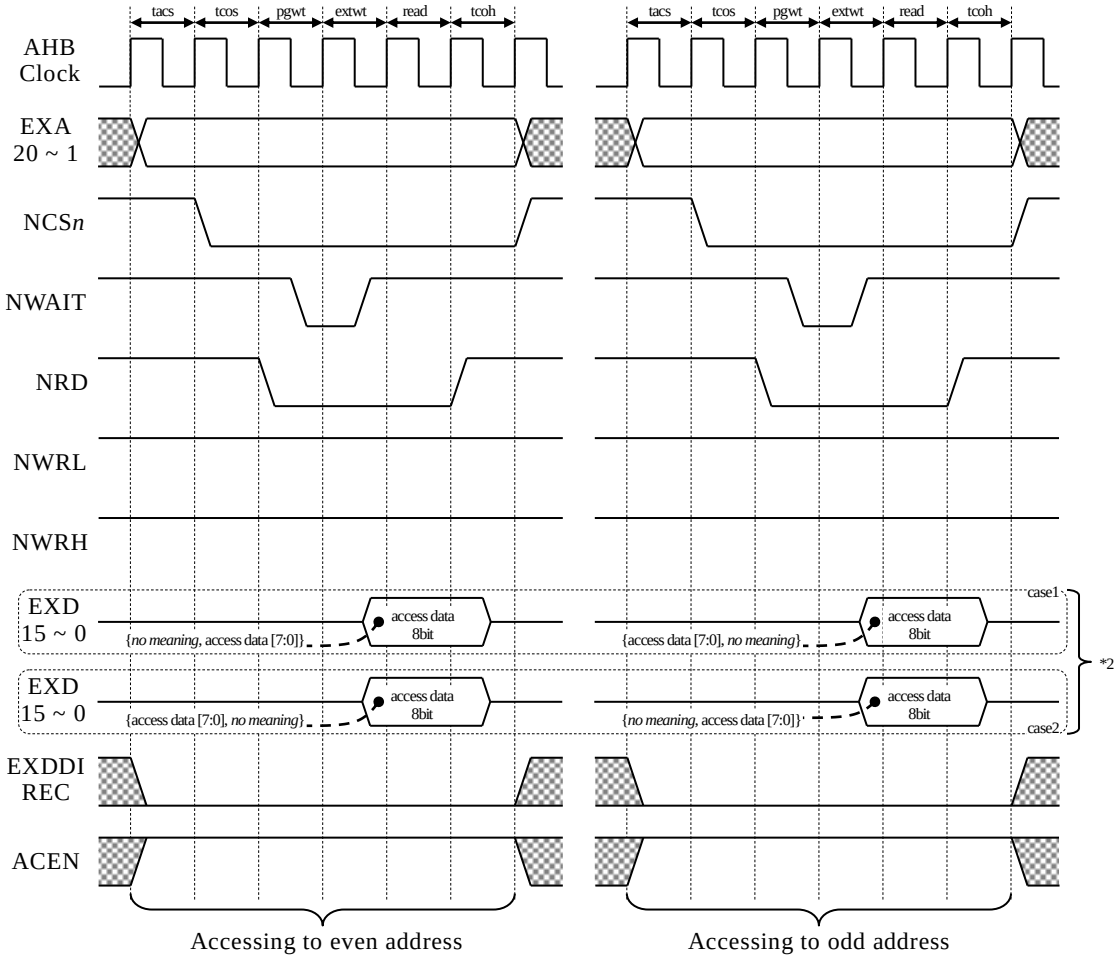


Figure 19-9 Example of 8 bit device mode connection to 16 bit data bus

19-4-2-6 Byte accessing with 16 bit bus at 8 bit device mode

Byte Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

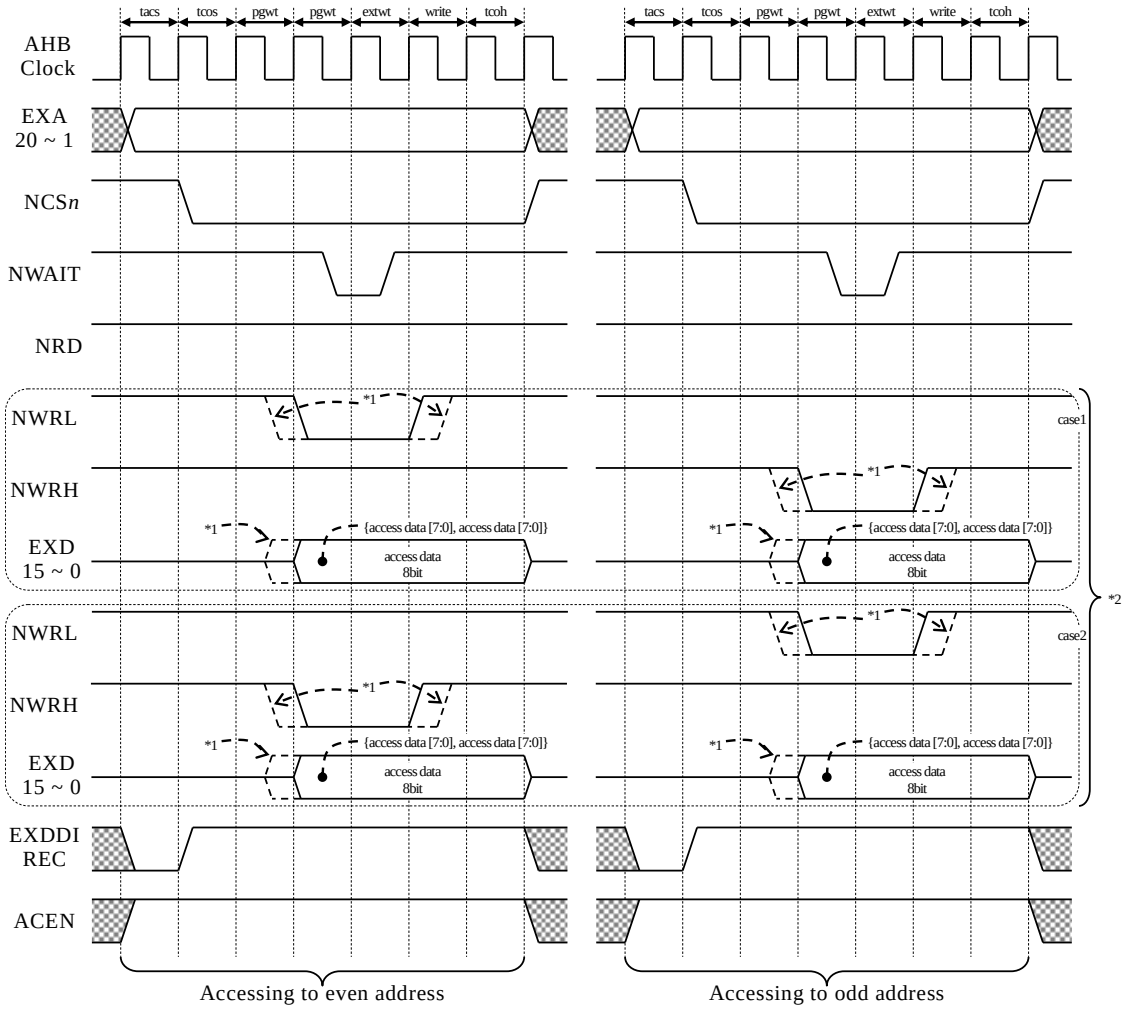
n: 3-0 (Area number of External-*n* Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-10 Byte read accessing with 16 bit bus at 8 bit device mode

Byte Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

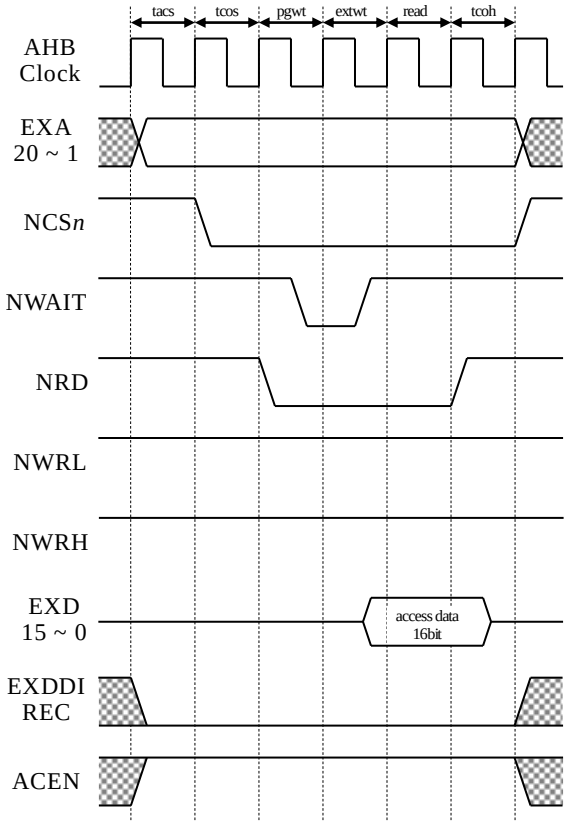
*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-11 Byte write accessing with 16 bit bus at 8 bit device mode

19-4-2-7 Half word accessing with 16 bit bus at 8 bit device mode

Half Word Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

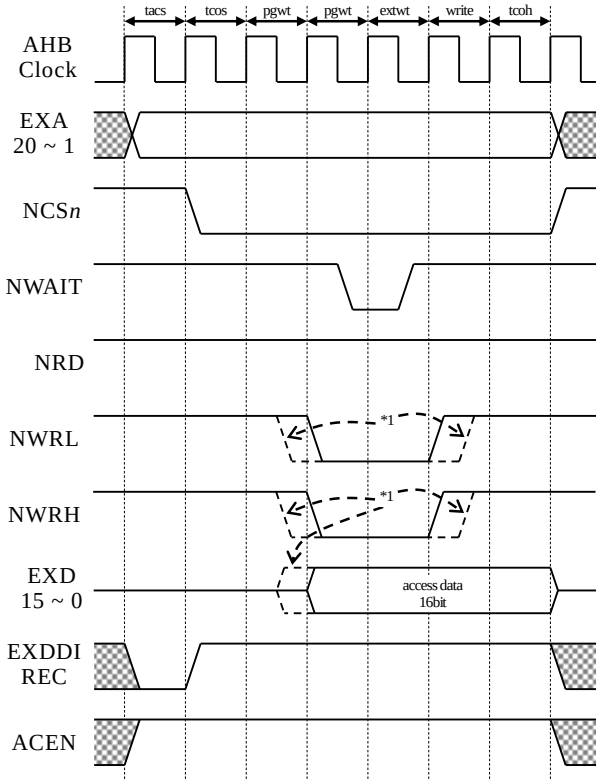
tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-*n* Area)

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-12 Half word read accessing with 16 bit bus at 8 bit device mode

Half Word Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-n Area)

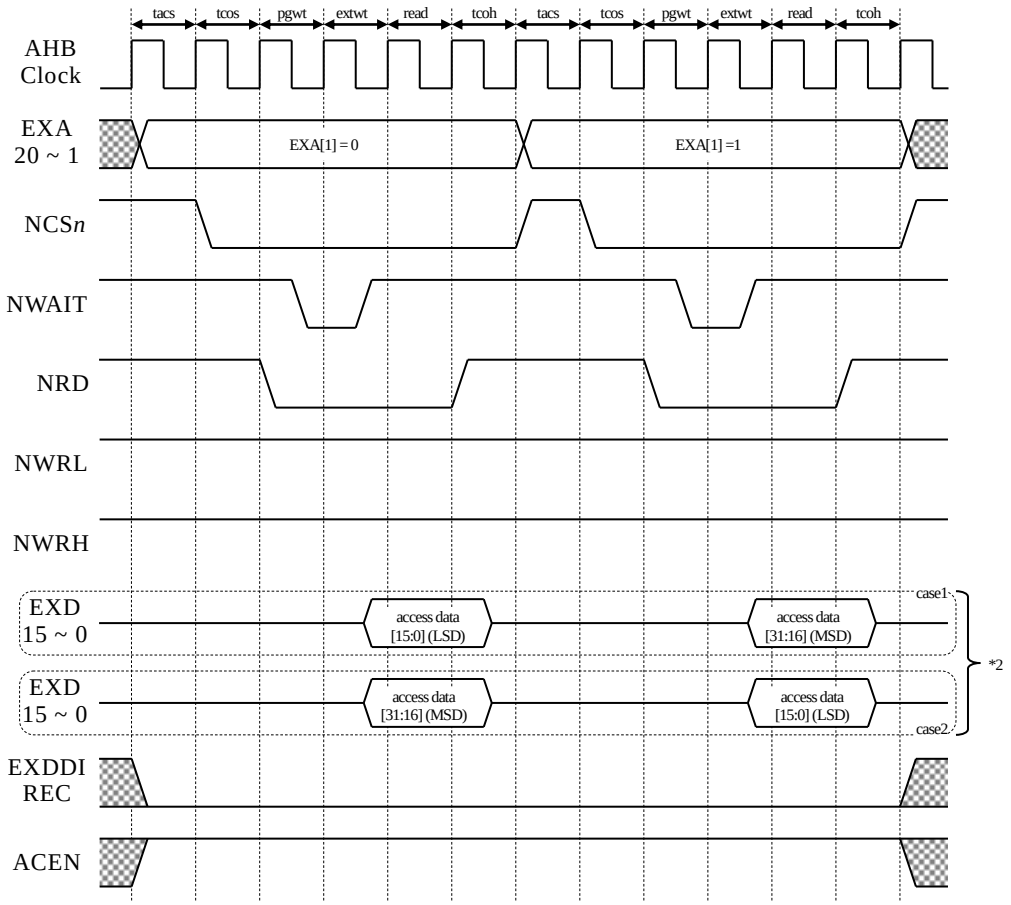
*1: It is solid line at short writing mode, it is broken line at long writing mode.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-13 Half word write accessing with 16 bit bus at 8 bit device mode

19-4-2-8 Word accessing with 16 bit bus at 8 bit device mode

Word Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

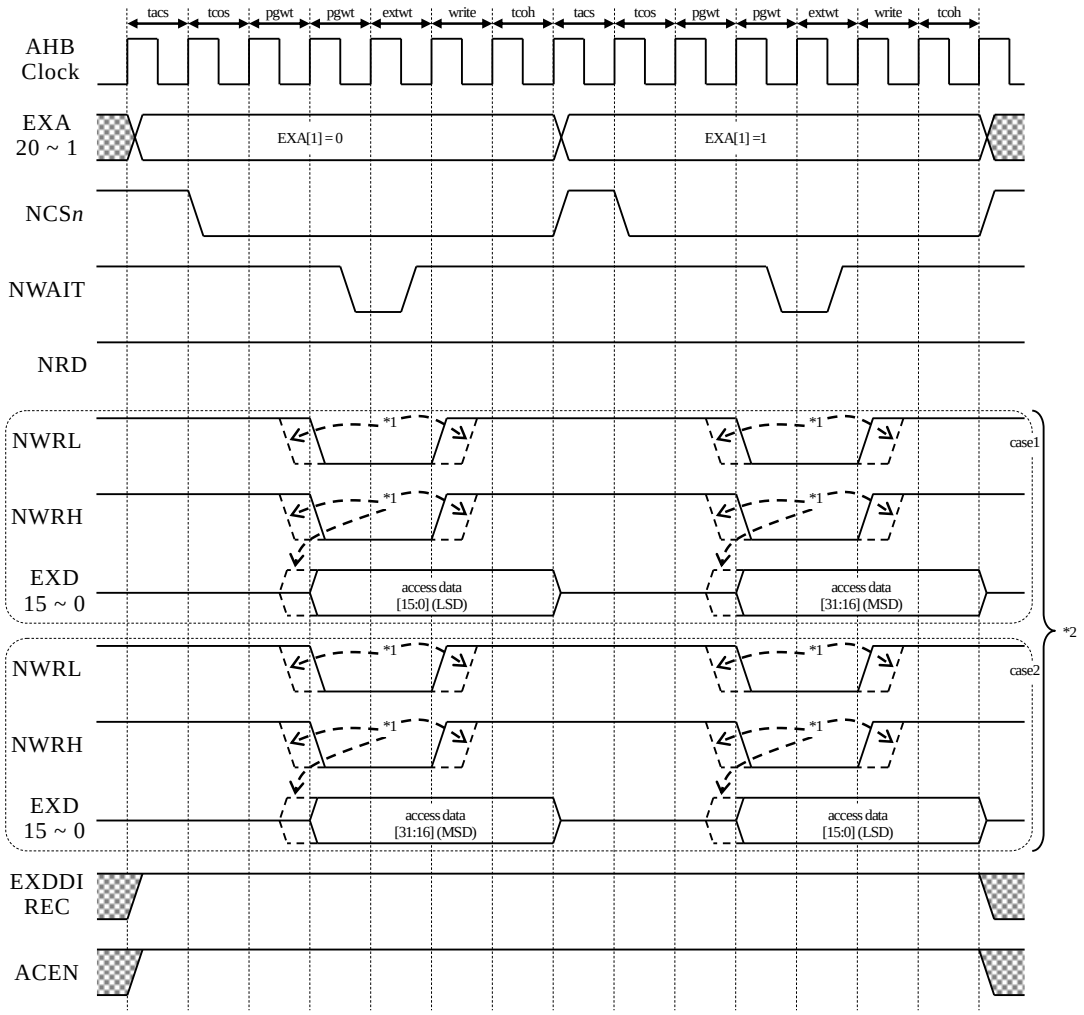
n: 3-0 (Area number of External-*n* Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-14 Word read accessing with 16 bit bus at 8 bit device mode

Word Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-15 Word write accessing with 16 bit bus at 8 bit device mode

19-4-2-9 Accessing with 16 bit bus at 16 bit device mode

This introduces details of the access when the memory device which has the data bus of 16 bit width is connected to the 16 bit data bus of this model.

This shows the example of 16 bit device mode connection to 16 bit bus in Figure 19-16.

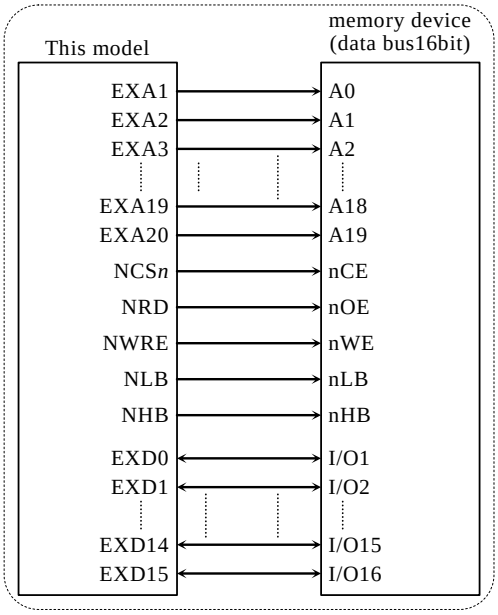
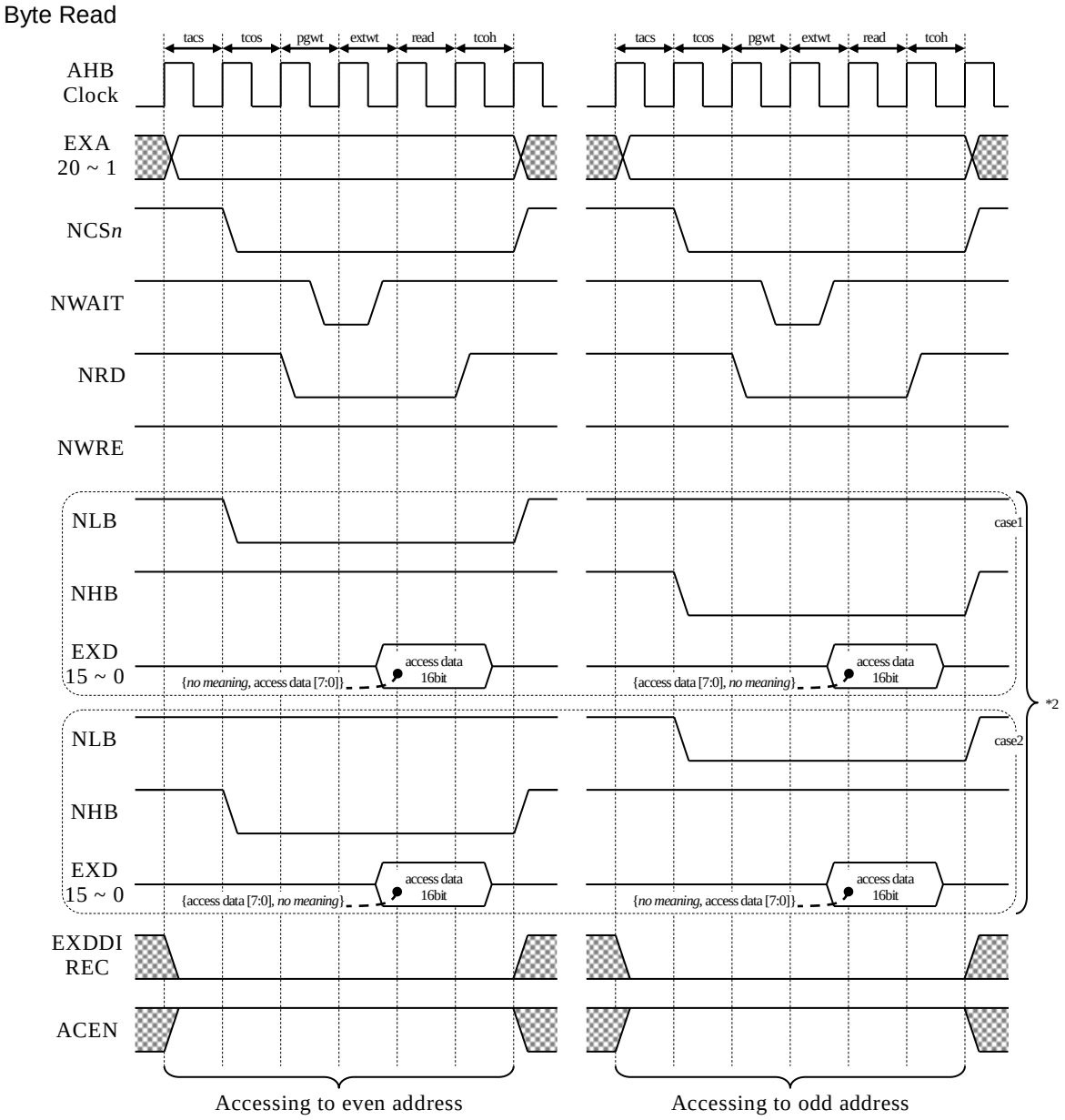


Figure 19-16 Example of 16 bit device mode connection to 16 bit data bus

19-4-2-10 Byte accessing with 16 bit bus at 16 bit device mode



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

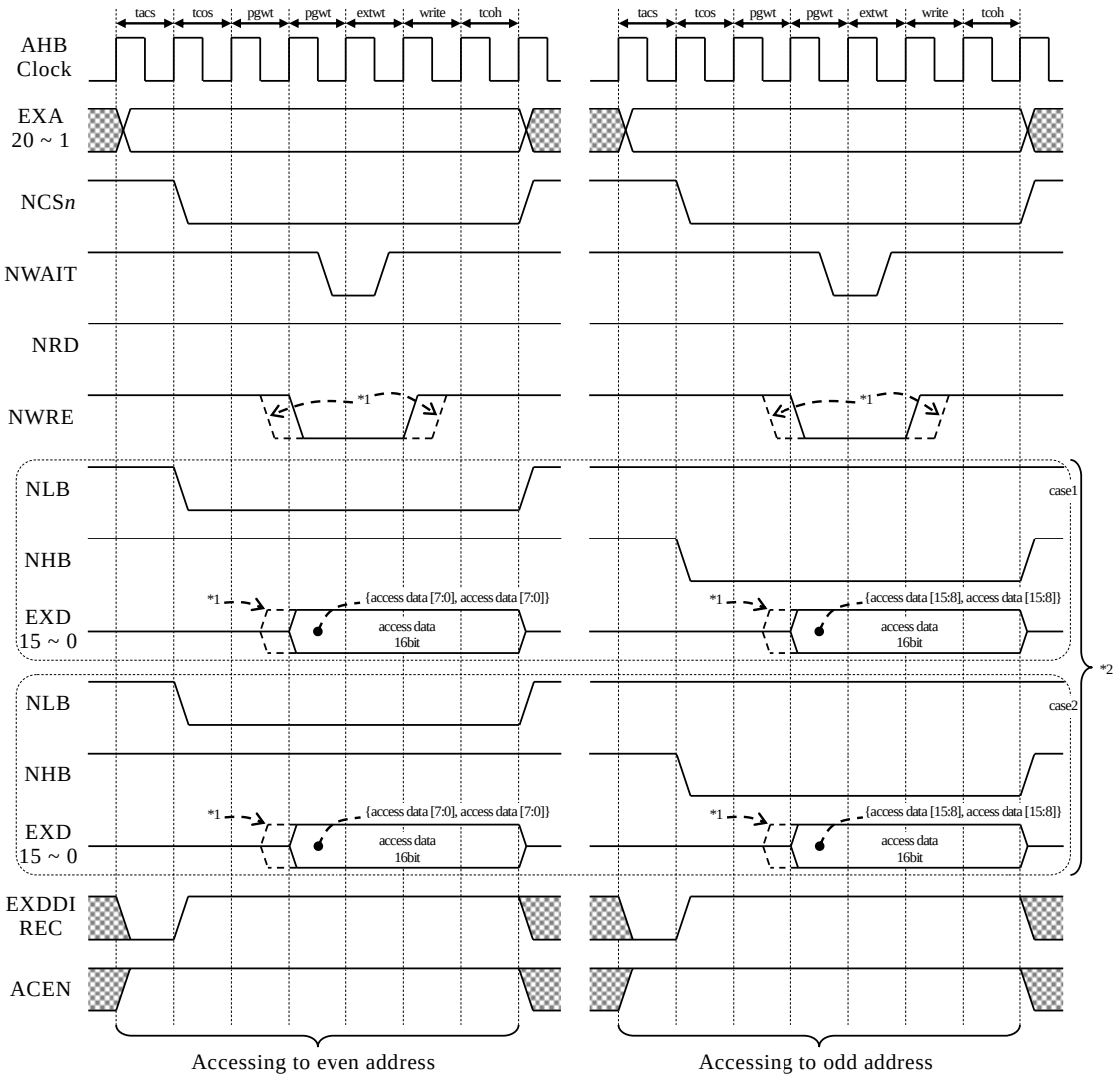
n: 3-0 (Area number of External-*n* Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-17 Byte read accessing with 16 bit bus at 16 bit device mode

Byte Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

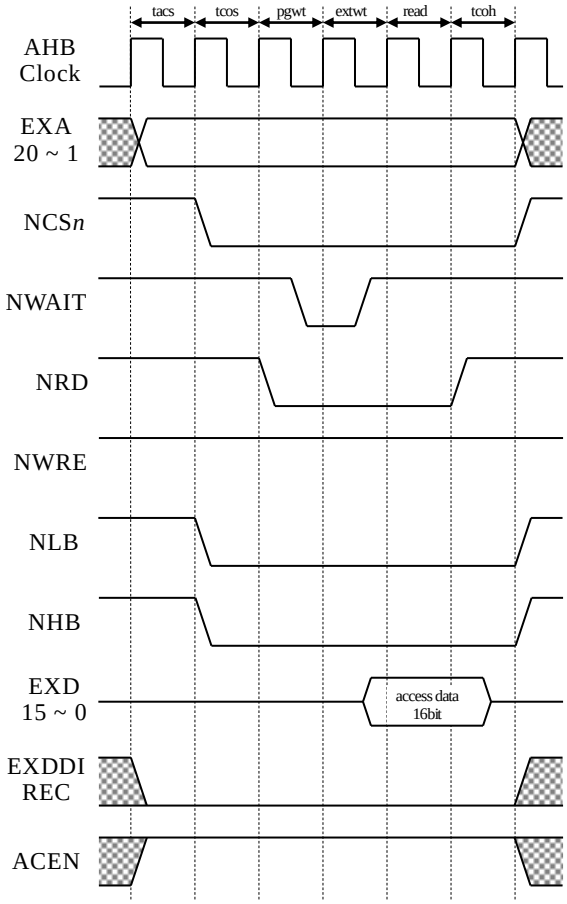
*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-18 Byte write accessing with 16 bit bus at 16 bit device mode

19-4-2-11 Half word accessing with 16 bit bus at 16 bit device mode

Half Word Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

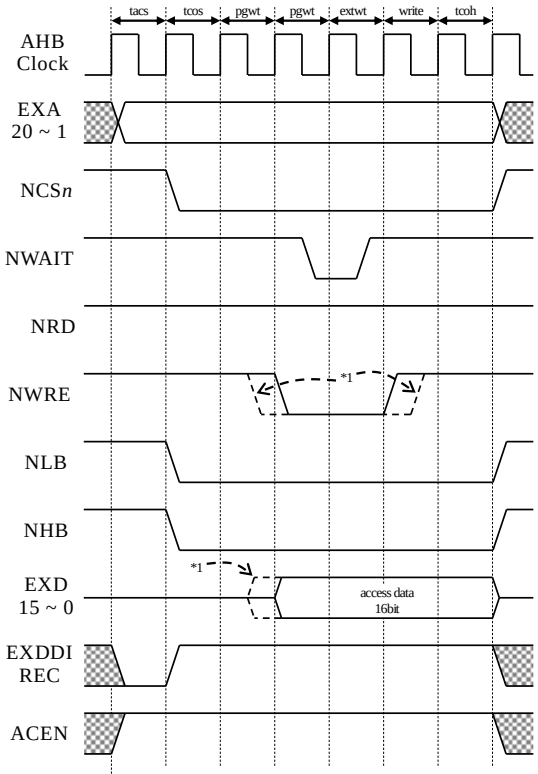
tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-*n* Area)

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-19 Half word read accessing with 16 bit bus at 16 bit device mode

Half Word Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

n: 3-0 (Area number of External-n Area)

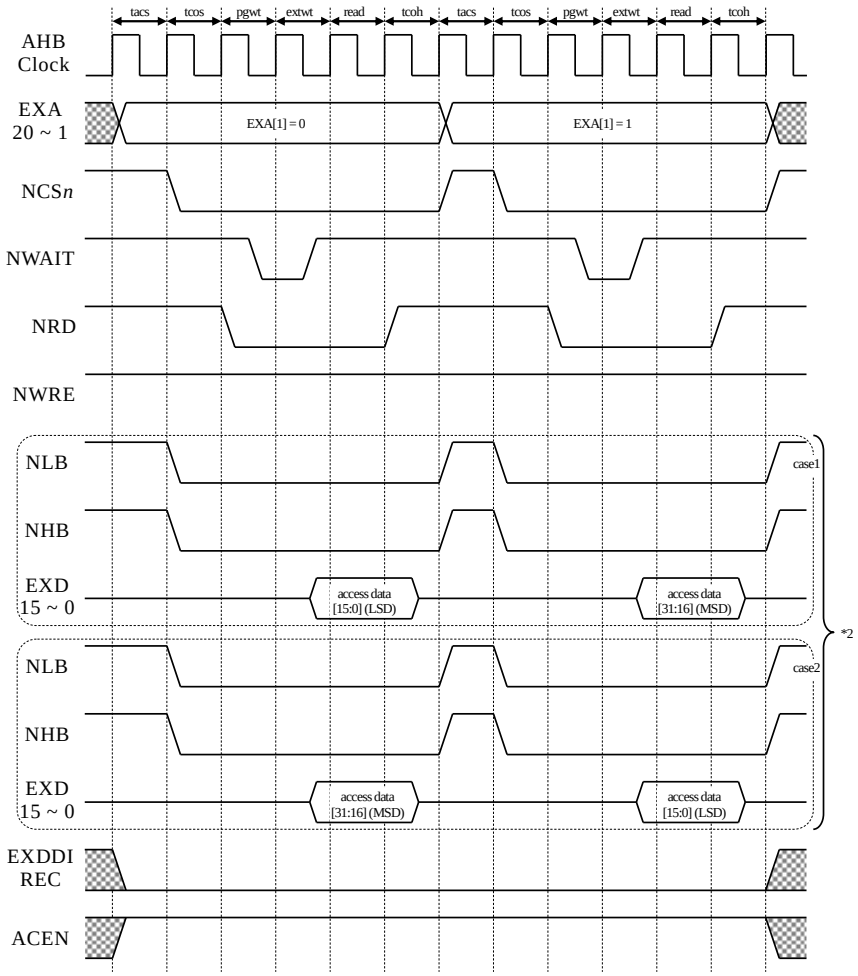
*1: It is solid line at short writing mode, it is broken line at long writing mode.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-20 Half word write accessing with 16 bit bus at 16 bit device mode

19-4-2-12 Word accessing with 16 bit bus at 16 bit device mode

Word Read



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle. It can be set to 255-0 wait cycle by setting fXMCnWT. This example supposes it is 1 wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

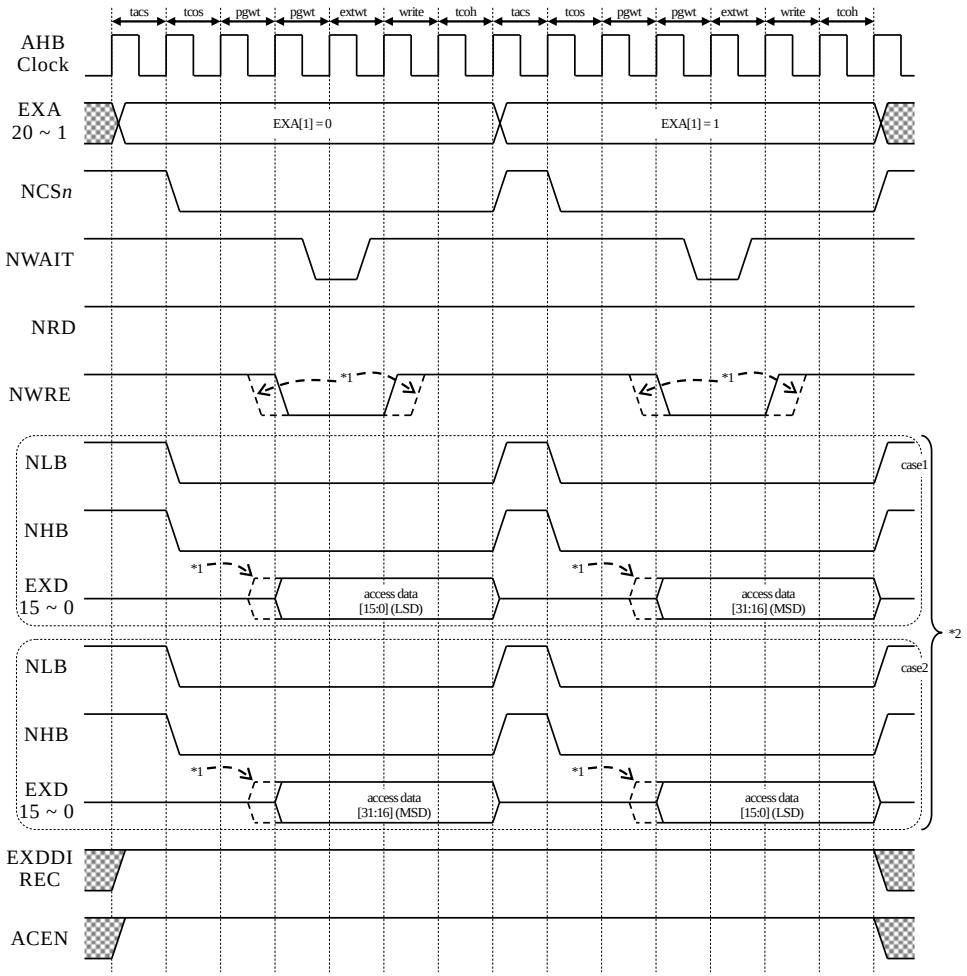
n: 3-0 (Area number of External-*n* Area)

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-21 Word read accessing with 16 bit bus at 16 bit device mode

Word Write



tacs: It is address setup cycle. It can be set to 63-0 cycle by setting fXMCnADSU. This example supposes it is 1cycle.

tcos: It is chip select setup cycle. It can be set to 63-0 cycle by setting fXMCnCSSU. This example supposes it is 1cycle.

pgwt: It is programmable wait cycle which is decided by setting fXMCnWT. Available wait cycle number is different by access mode. It can be set to 255-2 wait cycle at short writing mode, and set to 255-1 wait cycle at long writing mode. This example supposes it is 2wait cycle.

extwt: It is external wait cycle. It is decided by NWAIT signal supplied from outside.

tcoh: It is chip select hold cycle. It can be set to 63-0 cycle by setting fXMCnCSHD. This example supposes it is 1cycle.

MSD: Most Significant Digit LSD: Least Significant Digit

n: 3-0 (Area number of External-n Area)

*1: It is solid line at short writing mode, it is broken line at long writing mode.

*2: It is case1 if endian format is Little Endian, it is case2 if endian format is Big Endian.

Attention): Extwt is equal to 0 because this model is not mounting NWAIT function. And endian format is Little Endian fixation. EXDDIREC and ACEN are internal signals.

Figure 19-22 Word write accessing with 16 bit bus at 16 bit device mode

20 SDRAMC

20-1 Introduction

This chapter provides the brief summary of SDRAMC in the sections below:

- Brief Summary of SDRAMC
- Feature of SDRAMC

20-1-1 Brief Summary of SDRAMC

SDRAMC controls an external SDR-SDRAM memory (It is written at the following as SDRAM).

20-1-2 Feature of SDRAMC

- High speed by 4 burst access to SDRAM.
- Automatic initialization control with register control
- Power off control of SDRAM with register control
- Self-refresh control of SDRAM with register control
- On/Off control of auto refresh of SDRAM
- Support 8 bits and 16 bits data bus SDRAM.
- Support low power consumption type SDRAM(Mobile or LP etc.)

20-2 Functional Overview

This chapter provides overview of functional blocks used for the SDRAMC configuration in the sections below.

The following section is included.

- SDRAMC circuit configuration

20-2-1 SDRAMC circuit configuration

SDRAMC circuit configuration is shown in a Figure 20-1

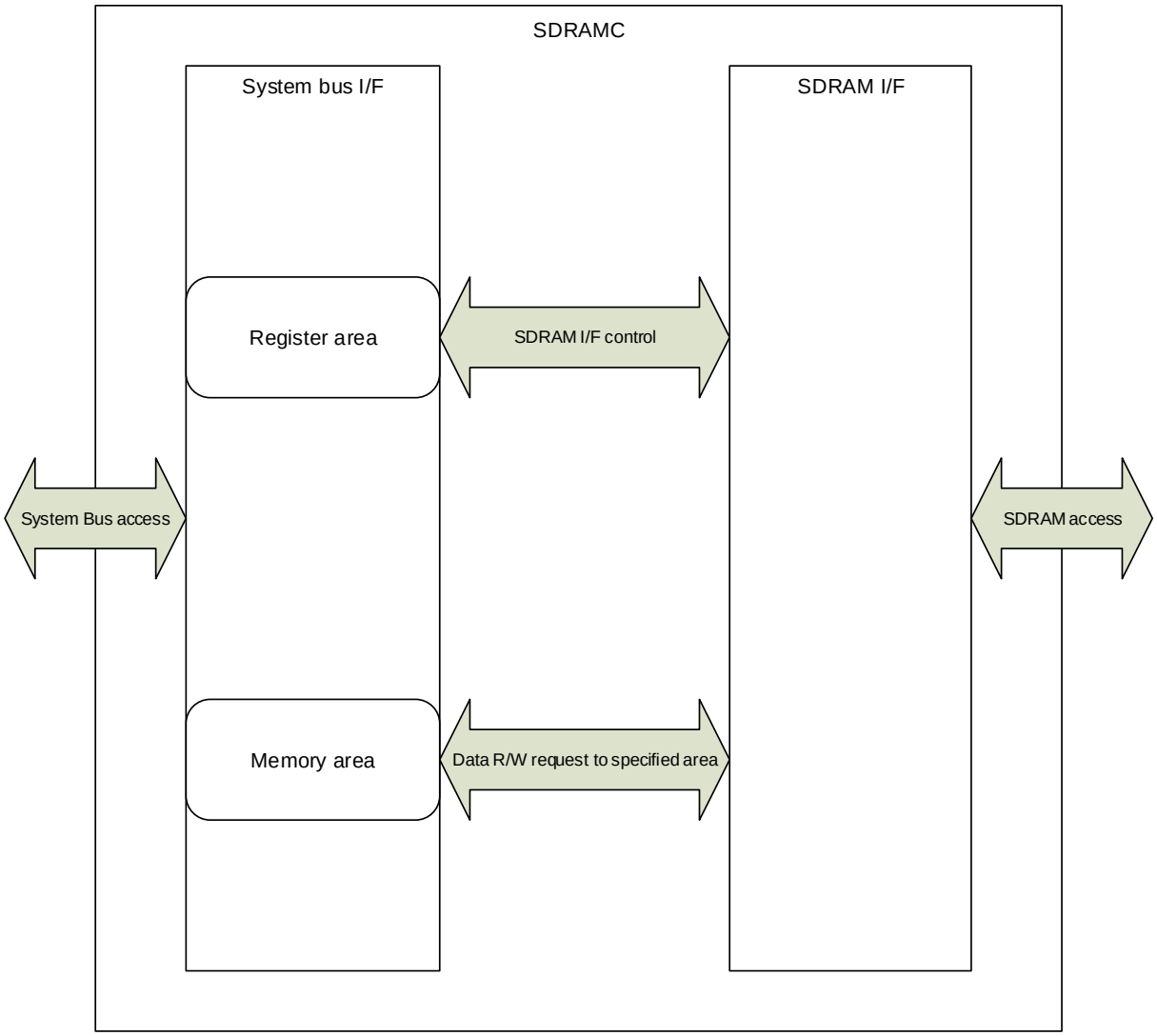


Figure 20-1

20-3 Programmer's Model

This chapter contains SDRAMC register information and other information necessary for the register setup using microcontrollers in the sections below:

The following section is included.

- About the programmer's model
- Summary of SDRAMC registers
- Register Descriptions

20-3-1 About the programmer's model

SDRAMC base address is not fixed and it will depend on the system implementation. Please refer to System specifications for more information about the base address.

Offset is fixed for all registers.

20-3-2 Summary of SDRAMC registers

SDRAMC is controlled by the registers shown in Table 20-1 below.

Remarks) Only word access is permitted. Byte and half word access are prohibited. And, accessible addresses are the ones listed in between the range of “Base+0x0000” to “Base+0xFFFF”. Do not access any other address.

Table 20-1 SDRAMC registers

Offset	Abbreviation	Attribute	Content	Initial value
0x00	AREFCTL	R/W	Auto refresh control register	0x0000_0000
0x04	SDRINIT	R/W	Initialization setting register	0x0000_0000
0x08	SDRSTBY	R/W	Standby setting register	0x0000_0303
0x0C	SDRMODE	R/W	Mode setting register	0x0000_0032
0x10	SDREXTMODE	R/W	Extended mode setting register	0x0002_0000
0x1C	MODECTL	R/W	Mode control register	0x0000_0000
0x20	CMDTIMGA	R/W	Command timing A setting register	0x1F0F_1F3F
0x24	CMDTIMGB	R/W	Command timing B setting register	0x0F0F_0F0F
0x28	CMDTIMGC	R/W	Command timing C setting register	0x0FFF_FF0F
0x2C	CMDTIMGD	R/W	Command timing D setting register	0x0707_0F1F
0x30	SDRTYPE	R/W	Type setting register	0x0000_0001
0x44	STSCTL	R/W	Status setting register	0x0000_0000

20-3-3 Register Descriptions

20-3-3-1 Auto refresh control register: AREFCTL

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	Reserved
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	AREFENB0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit8:Reserved
Please always set 0.

Bit0:AREFENB0
Auto refresh permission setting of SDRAM
0:auto refresh stop.
1:auto refresh start.
Attention) Please set after confirming the initialization operation completion (by using SDRINIT.INITDONE).

20-3-3-2 Initialization setting register: SDRINIT

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	WTIME7	WTIME6	WTIME5	WTIME4	WTIME3	WTIME2	WTIME1	WTIME0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	AREFNUM	-	-	-	PREWAIT
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	INITSET / INITDONE
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit31-24:WTIME7-0
Precharge command output wait time when initializing.
Time =256× HCLK cycle time × WTIME
Note) HCLK above and mentioned afterward means CPU I/F clock.

Bit12:AREFNUM
The number of auto refresh command output when initializing.
0:2 times
1:8 times

Bit8:PREWAIT
Precharge command output wait time to start output when initializing.
0:No wait
1:Wait till PREWAIT=0 to start precharge command output

Bit0 (At Write): INITSET.
Trigger to execute initialization of SDRAM (one shot).
0:No operation
1:Execute initialization of SDRAM

Bit0 (At Read): INITDONE.
Notification flag of initialization completion of SDRAM.
It is possible to confirm the initialization completion by reading after writing 1.
0:Initialization completed
1:SDRAM is being initialized.

20-3-3-3 Standby setting register: SDRSTBY

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	SMODE2 / -	SMODE1 / Reserved	SMODE0 / DPDN0:
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	Reserved: / Reserved:	Reserved: / PWDN0
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	Reserved	STBYSET / SREF0
Reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W

Bit18-16 (At Write): SMODE2-0.
Standby mode setting of SDRAM.
000:Self-refresh start
001:Self-refresh stop
010:Power down start
011:Power down stop
100:Deep power down start
101:Deep power down stop

Attention) For the setting to 100 b or 101 b, it is necessary to be selected that SDRTYPE register is 1 (Low power consumption type SDRAM (Mobile or LP etc.)). It is prohibited to set the values as 100 b and 101 b in the case of SDRTYPE register is 0 (General SDRAM).

Bit18 (At Read) :-.
always 0.

Bit17 (At Read): Reserved.
don't use

Bit16 (At Read): DPDN0.
Show whether SDRAM is in deep power down mode or not.
0:Not deep power down mode
1:Deep power down mode

Bit9-8 (At Write): Reserved.
don't use. Always 00 b should be written.

Bit9 (At Read): Reserved.

don't use

Bit8 (At Read): PWDN0.
Show whether SDRAM is in power down mode or not.
0:Not power down mode
1:Power down mode

Bit1:Reserved
don't use. Always 0 b should be written.

Bit0 (At Write): STBYSET.
Standby setting trigger of SDRAM.
0: No operation
1:Trigger to start standby and stop standby of SDRAM. (one shot)

Bit0 (At Read): SREF0.
Shown whether SDRAM is in self-refresh or not.
0:Not self-refresh.
1:Self-refresh.

Attention) Initial value of PWDN0 and SREF0 register is 1. This means initialization of SDRAM is not executed. After initialization, these values become 0.

20-3-3-4 Mode setting register: SDRMODE

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	Reserved	MODE12	MODE11	MODE10	MODE9	MODE8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
Reset	0	0	1	1	0	0	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit13:Reserved

don't use. Always 0 b should be written.

Bit12-0:MODE12-0

These values are outputed to address terminals when mode register for SDRAM is set.
Bit MODE12-0 corresponds to terminal SDRADDR12-0.
Set appropriate value according to SDRAM spec. Follow the restrictions below.

MODE bit	Explanation
12-10	Optional setting. Set 000 b usually. SDRAMC has no available function even though it is possible to set the values other than 000 b.
9	Write mode. 0 b: Burst write 1 b: prohibited
8-7	Optional setting. Set 00 b usually. SDRAMC has no available function even though it is possible to set the values other than 00 b.
6-4	CAS latency. Please set it according to SDRAM spec. 010 b:CAS latency 2 011 b:CAS latency 3 It is prohibited to set the values other than those mentioned above.
3	Burst type of SDRAM. Only 0 b is allowed to set. 0 b: Sequential burst
2-0	Burst length of SDRAM. Only 010 b is allowed to set. 010 b: burst length is 4

20-3-3-5 Extended mode setting register: SDREXTMODE

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	BANK1	BANK0
Reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	Resered	EMODE12	EMODE11	EMODE10	EMODE9	EMODE8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	EMODE7	EMODE6	EMODE5	EMODE4	EMODE3	EMODE2	EMODE1	EMODE0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit17-16:BANK1-0

These values are outputted to BANK terminals during setting extended mode register of low power type SDRAM.

Set 10 b normally for SDRAM whose number of BANK is 4.

Bit13:Reserved

don't use.0 should be written always.

Bit12-0:EMODE12-0

These values are outputed to address terminals when extended mode register of low power type SDRAM is set.

Bit EMODE12-0 corresponds to terminal SDRADDR12-0.

Set appropriate value according to low power type SDRAM spec.

Attention) In order to set this register, it is necessary to be selected that SDRTYPE register s 1 (low power consumption type SDRAM (Mobile or LP etc.)). It is prohibited to set any value in this register in the case of SDRTYPE register is 0 (General SDRAM).

20-3-3-6 Mode control register: MODECTL

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	EXTMODE / -
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	Resered	Resered
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	MODESET / -
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit16 (At Write): EXTMODE.
Select mode register to be set.
0:Normal mode register
1:Extended mode register

Bit16 (At Read) :-.
It is always 0.

Bit9-8 :Reserved
don't use. 0 should be written always.

Bit0 (At Write): MODESET.
Trigger of mode register setting after SDRAM is initialized.
0:No operation
1:Set the content of mode register to SDRAM (one shot).

Bit0 (At Read) :-.
It is always 0.

Attention) In order to set Extended mode register of SDRAM, it is necessary to be selected that SDRTYPE register is 1 (low power consumption type SDRAM (Mobile or LP etc.)). In the case of SDRTYPE is 0 (General SDRAM), Normal mode register is set even if Bit16: EXTMODE register is set to 1.

20-3-3-7 Command timing A setting register: CMDTIMGA

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	tRAS4	tRAS3	tRAS2	tRAS1	tRAS0
Reset	0	0	0	1	1	1	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	tRCD3	tRCD2	tRCD1	tRCD0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	tRC4	tRC3	tRC2	tRC1	tRC0
Reset	0	0	0	1	1	1	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	tRFC5	tRFC4	tRFC3	tRFC2	tRFC1	tRFC0
Reset	0	0	1	1	1	1	1	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit28-24:tRAS4-0
Set ACTIVE to PRECHARGE command time as follows.
Time = HCLK cycle time × (tRAS+1)

Bit19-16:tRCD3-0
Set ACTIVE command to WRITE or READ command time as follows.
Time = HCLK cycle time × (tRCD+1)
Attention) Set 1 or more to tRCD.

Bit12-8:tRC4-0
Set ACTIVE to ACTIVE or AUTO refresh command time as follows.
Time = HCLK cycle time × (tRC+1)

Bit5-0:tRFC5-0
Set AUTO refresh to ACTIVE or AUTO refresh command time as follows.
Time = HCLK cycle time × (tRFC+1)

20-3-3-8 Command timing B setting register: CMDTIMGB

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	tRP3	tRP2	tRP1	tRP0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	tMRD3	tMRD2	tMRD1	tMRD0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	tRTW3	tRTW2	tRTW1	tRTW0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	tWTR3	tWTR2	tWTR1	tWTR0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit27-24:tRP3-0
Set PRECHARGE time as follows.
Time = HCLK cycle time × (tRP+1)
Attention) Set 1 or more to tRP.

Bit19-16:tMRD3-0
Set MODE register setting command cycle time as follows.
Time = HCLK cycle time × (tMRD+1)
Attention) Set 1 or more to tMRD.

Bit11-8:tRTW3-0
Set READ to WRITE command interval time as follows.
Time = HCLK cycle time × (tRTW+1)

Bit3-0:tWTR3-0
Set WRITE to READ command interval time as follows.
Time = HCLK cycle time × (tWTR-3) + HCLK cycle time × 3
Attention) Set 3 or more to tWTR.

20-3-3-9 Command timing C setting register: CMDTIMGC

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	tDPL3	tDPL2	tDPL1	tDPL0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	tXSNR7	tXSNR6	tXSNR5	tXSNR4	tXSNR3	tXSNR2	tXSNR1	tXSNR0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	tXSRD7	tXSRD6	tXSRD5	tXSRD4	tXSRD3	tXSRD2	tXSRD1	tXSRD0
Reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	tRTP3	tRTP2	tRTP1	tRTP0
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit27-24:tDPL3-0
Set WRITE recovery time as follows.
Time = HCLK cycle time × (tDPL-3) + HCLK cycle time × 3
Attention) Set 3 or more to tDPL.

Bit23-16:tXSNR7-0
Set command interval time from the end of SELF refresh mode as follows.
Time = HCLK cycle time × (tXSNR+1)
Attention) Set 0x2 or more to tXSNR.

Bit15-8:Reserved
Attention) Set 0x2.

Bit3-0:tRTP3-0
Set READ to PRECHARGE command interval timeas follows.
Time = HCLK cycle time × (tRTP+1)
Attention) Set 1 or more to tRTP.

20-3-3-10 Command timing D setting register: CMDTIMGD

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	tREF11	tREF10	tREF9	tREF8	tREF7	tREF6	tREF5	tREF4
Reset	0	0	0	0	0	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	tREF3	tREF2	tREF1	tREF0	-	tXP2	tXP1	tXP0
Reset	0	0	0	0	0	1	1	1
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0	0	0	1	1	1	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit31-20:tREF11-0

Set refresh interval time as follows.
Time = HCLK cycle time × (tREF)
Attention) Set 4 or more to tREF.

Bit18-16:tXP2-0

Set command interval time after the end of power down mode as follows.
Time = HCLK cycle time × (tXP)
Attention) Set 0x1 or more to tXP.

Bit11-8:tWAC3-0

Set the write access time (from the command execution of internal WRITE to the notification of completion of writre access, WrACK) as follows.
Time = HCLK cycle time × (tWAC)
* Set 0x3 because this SDRAM controller is fixed on 4 burst access.

Bit4-0:tRAC4-0Reserved

Set the read access time (from the command execution of internal READ to the notification of completion of read access, RdACK) as follows.
Time = HCLK cycle time × (tRAC)
* Because this SDRAM controller is fixed on 4 burst access and the time from the command of internal READ to RdACK is fixed, set the value which is caliculated as follows.
Setteing value = CAS Latency of SDRAM + 6

20-3-3-11 Type setting register: SDRTYPE

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	MSDRAM
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	RTYPE3	RTYPE2	RTYPE1	RTYPE0
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit8:MSDRAM

Setting of SDRAM of low power consumption type.

0: General SDRAM

1: SDRAM of low power consumption type (Mobile or LP etc.)

Attention) Set this register before the initialization of SDRAM.
The initialization of SDRAM can be executed by the setting 1 to INITDONE in SDRINIT register.

Bit3-0:RTYPE3-0

Type of SDRAM.

RTYPE3-0				SDR TYPE				
				Capacity [Bit]	Data Width	BANK Width	ROW Width	COL Width
0	0	0	0	64 M	8	2	12	9
0	0	0	1	64 M	16	2	12	8
0	0	1	0	128 M	8	2	12	10
0	0	1	1	128 M	16	2	12	9
0	1	0	0	256 M	8	2	13	10
0	1	0	1	256 M	16	2	13	9
Setting other value is prohibited				-	-	-	-	-

The relation of address of SDRAMC(HADDR) and address of SDRAM(BANK,ROW,COLUMN) is as follows.

In case of 8 bits data width : HADDR[MAX-1:0] = {BANK,ROW,COLUMN}

In case of 16 bits data width : HADDR[MAX-1:1] = {BANK,ROW,COLUMN}

Note) HADDR above is CPU I/F address of width 25 bit.

20-3-3-12 Status setting register: STSCTL

[Base + 0x0044]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	Reserved	STBY
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Bit1:Reserved
don't use. 0 should be written always

Bit0:STBY
Show whether the access to SDRAM occurs or not during standby.
0:Access doesn’t occur during standby.
1:Access occurs during standby. Cleared to 0 by writing 1.

20-4 Operation

In this chapter, describe the operation flow of SDRAMC.

20-4-1 About the initialization operation

In general, SDRAM must be initialized after the stability period following power on. Due to no method to detect the timing of power on of SDRAM, SDRAMC need initialization by user side operation after the stability period(described in the device specifications of SDRAM) following power on of SDRAM. Please initialize only one time after power on of SDRAM(but, except for after exit from deep power down mode).

The initialization operation is as follows.

SDRAMC performs the following initialization by setting the SDRINIT.INITSET register to “1". The SDRINIT.INITDONE remains “1" during initialization.

(1) Issue NOP command : NOP is processed.

(2) Wait : If SDRINIT.PREWAIT register is set to “1", issue of following precharge command waits for the period. After waiting for the period determined by SDRINIT.WTIME register, SDRAMC goes to next step. If SDRINIT.PREWAIT register is set to “0" from the beginning, SDRAMC goes to next step without wait.

(3) Issue of precharge command to all banks : precharge of all banks is processed.

(4) Issue of auto refresh command : SDRAM issues continuous auto refresh command 2 times if SDRINIT.AREFNUM register is “0”, and 8 times if SDRINIT.AREFNUM register is “1”.

(5) Issue of mode register setting command : Set mode register.

<< In case of SDRTYPE.MSDRAM = “1” >>

Just after the previous step, SDRAMC issues extended mode register setting command.

(6) If the SDRINIT.INITDONE register becomes “0", initialization is completed.

In addition to initialization operation above, several register settings are needed to use SDRAM.

And, it is necessary to set SDRTYPE.MSDRAM and AREFCTL.AREFENB0 according to the timing shown in the following.

Set SDRTYPE.MSDRAM register before initialization operation.

Set AREFCTL.AREFENB0 register after initialization operation.

20-4-2 Setting method of mode register

After initialization, setting of mode register in SDRAM is always possible (However, except for while SDRAM is in standby mode).

<< method of setting of mode register >>

(1) Set the required value to SDRMODE.MODE register to be written to SDRAM.

(2) Write “1” to MODECTL.MODESET register and “0” to MODECTL.EXTMODE register at the same time.

Attention) There are restrictions regarding SDRMODE.MODE. Refer to 20-3-3-4Mode setting register: SDRMODE in detail.

20-4-3 Setting method of extended mode register (SDRTYPE.MSDRAM register is “1").

Extended mode register in SDRAM can be set by the similar method to set mode register.

<< setting method of extended mode register >>

(1) Set the required value to SDREXTMODE.EMODE register to be written to SDRAM.

(2) Write “1” to MODECTL.MODESET register and “1” to MODECTL.EXTMODE register at the same time.

20-4-4 Control method of auto refresh

To maintain data, SDRAM needs regular refresh operation. SDRAMC can process regular refresh operation automatically by the following operation.

<< method to start auto refresh >>

(1) Set refresh interval time to CMDTIMGD.tREF register to meet refresh interval time described in device specification of SDRAM.

(2) Refresh operation is executed regularly with the interval time set to CMDTIMGD.tREF after setting “1" to AREFCTL.AREFENB0 register.

<< method to stop auto refresh >>

(1) Auto refresh operation stops by setting “0" to AREFCTL.AREFENB0 register. Note that the data stored is not guaranteed after stopping auto refresh operation. Moreover, auto refresh operation stops automaticaly while SDRAM is in standby mode (Refer to the following clause).

20-4-5 Setting method of SDRAM standby mode

Standby mode is the mode that power consumption is reduced by putting SDRAM which is not used for data storage into stanby mode (stop).

There are three kinds of standby mode that can be set from SDRAMC.

1. Self-refresh mode
   Suppress power consumption with keeping stored data
2. (precharge) power down mode
   Suppress power consumption more than Self-refresh mode. However, stored data is not kept without refresh operation.
3. Deep power down mode
   Supress power consumption more than power down mode. However, stored data is not kept.

<< Enter into standby mode >>

(1) Set start of deep power down/power down/self-refresh by SDRSTBY.SMODE register.
   << only in case of deep power down >>
   Set “0" to AREFCTL.AREFENB0 register to stop auto refresh.
(2) Set “1” to SDRSTBY.STBYSET register after (1) or at the same time as (1).

<< Exit from standby mode >>

(1) Set stop of deep power down/power down/self-refresh by SDRSTBY.SMODE register.
(2) Set “1” to SDRSTBY.STBYSET register after (1) or at the same time as (1).
   << only in case of deep power down >>
   Do initialization of SDRAM again, and after that, restart auto refresh by setting “1” to AREFCTL.AREFENB0.

Attention) Write and read access, and set mode register to SDRAM are prohibited during standby mode. STSCTL.STBY becomes "1" if such accesses occur.

20-4-6 About the setting of waiting time between SDRAM commands

It is necessary to input the various commands to SDR-SDRAM with kept the timing rule defined each by the device specifications of SDR-SDRAM. This controller can coordinate the timing depending on SDRAM device by using four registers, SDRTIMGA, SDRTIMGB, SDRTIMGC and SDRTIMGDA. But the waiting time is set to the number of HCLK cycle supplied to this controller. Please note that it is impossible to adjust the delay time by ns unit depending on the specification of SDRAM device.

HCLK supplied to this controller is the devided clock setting by EXT4DIV5-0 in PERICLKDIV register from the BASIC clock source selecting and divide setting by OSCCNT and PLLREFCNT written in “ProgrammersModel_OSC_v*.**.doc”. On the other hand, HCLK and SDRCLK which is used for the communication to SDRAM device are same clock, therefore it is equivalent to the waiting time is depend on the period of SDRCLK.

Below figure shows the relationship between the setting of number of HCLK cycle and the parameter which is set in the register of SDRTIMGA, SDRTIMGB, SDRTIMGC and SDRTIMGDA.

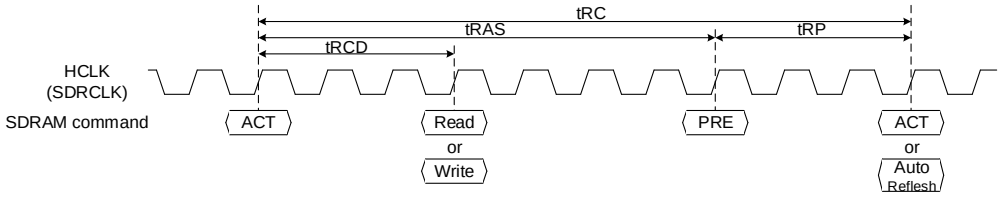


Figure 20-2

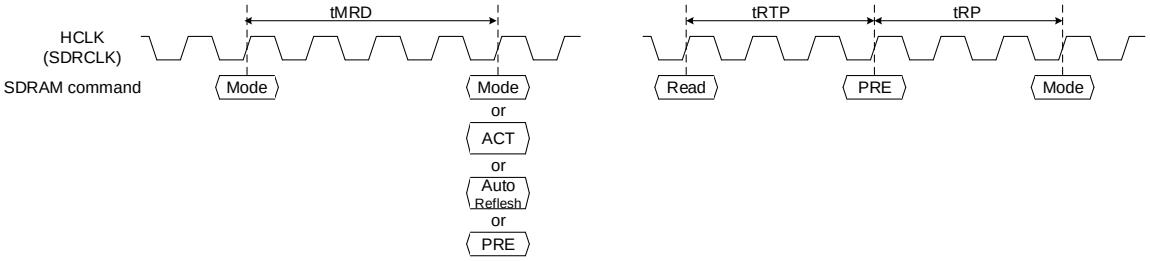


Figure 20-3

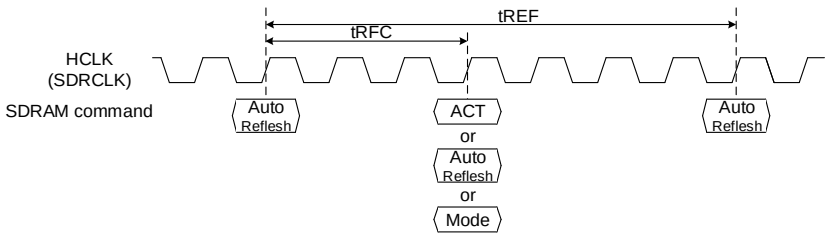


Figure 20-4

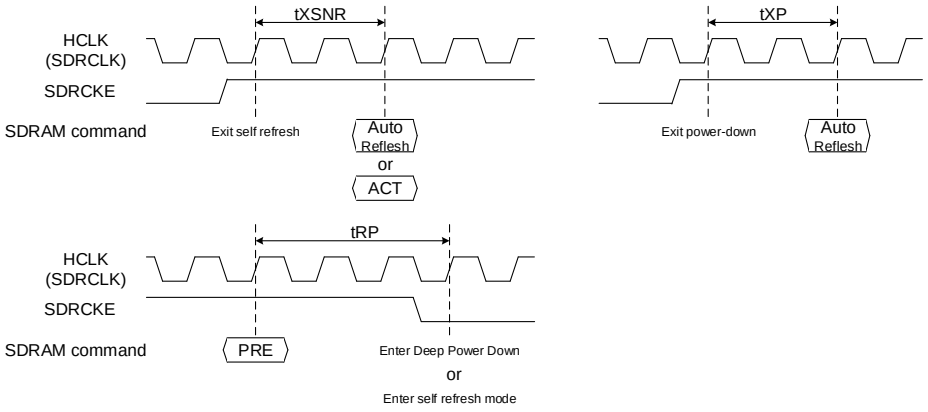


Figure 20-5

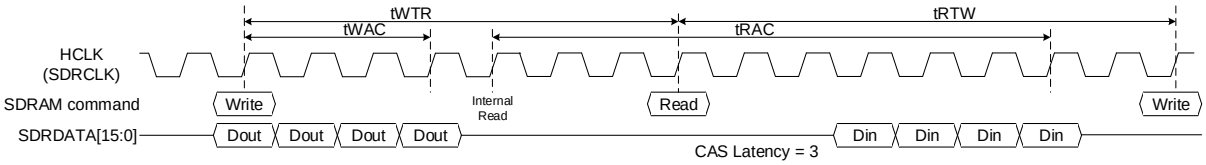


Figure 20-6

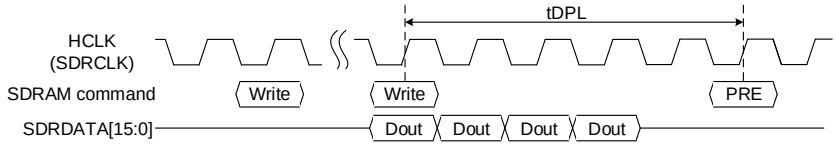


Figure 20-7

* Please note that it is necessary to be set tWLR and tDPL by (the waiting time for SDRAM device) + (HCLK period x 3).

20-4-7 About write and read access to SDRAM

After initilization, following the required register setting, read and write access to SDRAM are performed by the access to SDRAM Memory Area based on the bus control information (bus width, state of transfer, write and read, etc.) and mode register information(CAS latency) and SDRTYPE.RTYPE register, etc.

21 USB20DC

21-1 Introduction

This chapter provides the outline of USB20DC in the section below.

- The outline of USB20DC

21-1-1 The outline of USB20DC

1. The outline of macro

This macro is a USB 2.0 controller with a built-in PHY and supports USB 2.0 HS/FS/LS (Host only).
Attention: This product cannot use the built-in host function in this macro.
Use another built-in host macro compliant with EHCI.

This macro consists of two blocks: PHY block compliant with UTMI and LINK block handling USB protocol.

This macro has a CPU interface and a DMA interface as the system interfaces.

This macro performs 2 basic operations below.

(*1) The endpoint FIFO in LINK block receives data via PHY block from USB. Then, the data is output to CPU or DMA interface.
(*2) CPU or DMA interface writes transmission data in the endpoint FIFO in LINK block. Then, the data is transmitted via PHY to USB.

2. Features

2.1. Overall function

- With PHY block/LINK block

2.2. Host function (Not supported by this product.)

2.3. Device function

- Supports up to 16 high/full-speed endpoints.
  (Low speed is not supported.)
- Each endpoint buffer size (1 to 1024 Byte) and the number of buffer planes (1 to 2) can be selected.
- The maximum total buffer size is 32 KByte and the maximum number of buffers is 32.
  (RAM of this product is 4.5 KByte in total.)
- USB standard commands are supported by applications.

2.4. I/F

- With two AHB slave interfaces (CPU-I/F and DMA-I/F) compliant with AHB bus (AMBA Rev2.0)
- Endpoint buffer can be referred to as AHB address space.
- DMA I/F automatically converts AHB address to the endpoint buffer RAM address.
- DMA I/F can transfer to an external DMAC via up to 2 channels (2 endpoints).
- DMA I/F supports burst transfer of 0 wait and 1 wait.
- This macro is connected to an external 2-port RAM as a device endpoint buffer.

3. Macro pins

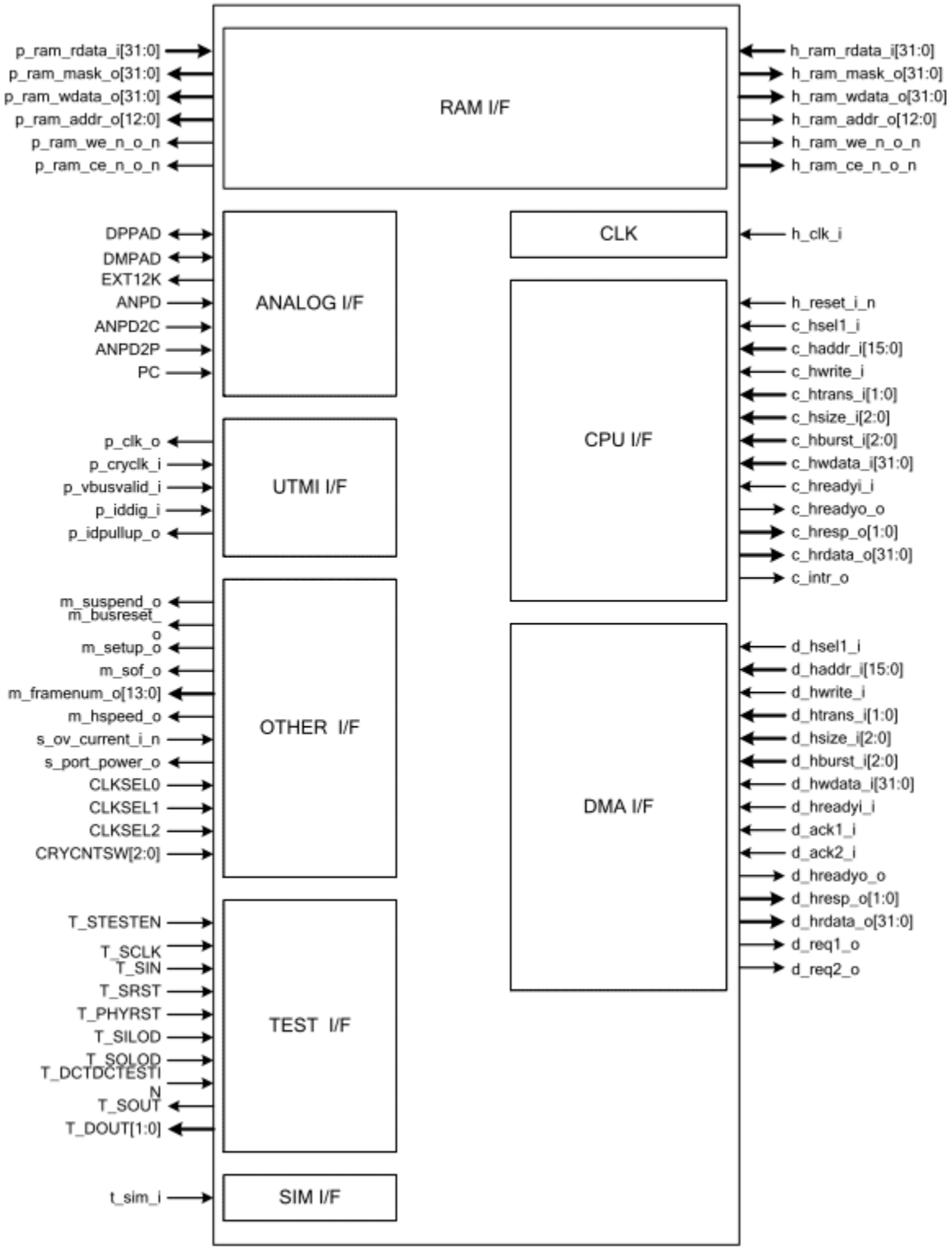


Figure 21-1

Attention: When needed, functions are explained with these macro-pin names, but not all macro pins are explained.

The table below shows where main macro pins used in this specification are connected to.

- Pin connection

Macro pins	Connection destination
h_clk_i	Bus clock, BASIC_CLK (See the datasheet.)
p_cryclk_i	XT1 oscillation (See the datasheet.) Oscillation stops when USB is suspended and can automatically restart at the time of resume. This can reduce power consumption during the suspend state.
h_reset_i_n	Internal signal that is reset by hard reset and MRSTCNTBASIC:USBDEV RSTB register. MRSTCNTBASIC: USBDEV RSTB register (See System controller specification.)
c_addr_i d_haddr_i	AHB address bus
d_req*_o	DMA controller
d_ack*_i	DMA controller
c_intr_o	Interrupt controller
m_suspend_o	Not used (open)
s_port_power_o	Not used (open)
s_ov_current_i_n	Input fixed at 1’b0
p_vbusvalid_i	USBCNT:VBUSVALID register (See the System Controller specification)
p_suspend_o_n	PHY SUSPEND (Suspend)
p_opmode[1:0]	PHY OPMODE (OP mode)
p_clki	PHY CLOCK (Operation clock)
p_reset_i	PHY RESET
p_idpullup_o	Not used (open)
p_dppulldown_o	PHY D+ pull-down enable
p_dmpulldown_o	PHY D- pull-down enable
ANPD	USBCNT:ANPD register (See System controller specification.)

Also see System controller specification for other registers necessary to control USB20DC.

5.2. CPU/DMA interface signal

CPU/DMAC is interfaced by I/F signals compliant with AMBA-AHB slave. CPU-I/F and DMAC-I/F each have an independent port and they can operate at the same time unless the internal resource contention occurs. (See 6.3.2 Access contention for the internal resource contention.)

See the document below for the detailed timing of AHB.

- AMBA Specification (Rev 2.0) AHB

CPU interface and DMA interface have the same specifications of functions and timing except that they have different transfer-type management in AHB address space.
See 6.3 CPU/DMAC interface function specification for the different transfer-type management.

DMAC interface has 2 channels of DMA transfer request signals (d_req1_o, d_req2_o) and the acknowledge signals (d_ack1_i, d_ack2_i). AHB pin uses 2 channels in common to time-share 1 Ch and 2 Ch. The timings of request signals and acknowledge signals differ according to the DMA transfer mode (demand transfer mode or block transfer mode). See 7.6.1 DMAC1, DMAC2 for DMA transfer mode setting.

* Attention: Also see the specification of DMA controller for DMA transfer that can be used by this product.

Interrupt request signals are notified as one signal “c_intr_o”, which collects various types of interrupt causes. This signal continues being asserted unless all event detection flags of unmasked interrupt causes are cleared.
See Chapter 7. Register (7.2 Device, 7.6 DMAC and 7.8 RAM areas) for the detailed interrupt factors.

5.8. Clock and reset

5.8.1. Clock

This macro has 3 clocks: PHY-side logic operation clock (p_clk_o pin), PHY reference clock (p_cryclk_i pin) and application-side (APP) logic operation clock (h_clk_i pin).

h_clk_i is used as the bus clock of CPU-I/F and DMA-I/F (AHB bus).

* Attention: The clock frequency shall be set as below.

65 MHz ≤ h_clk_i ≤ BASIC_CLK maximum operation frequency (See the datasheet)

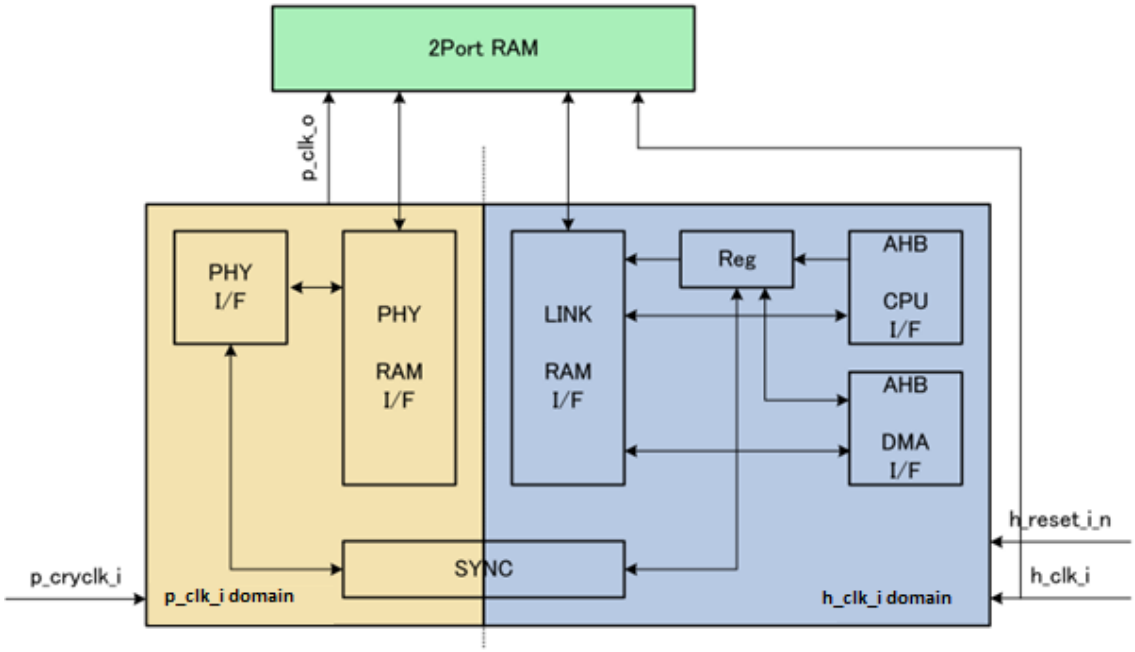


Figure 21-2 Clock domain and reset

5.8.2. Reset

Hard reset

The system reset is connected to h_reset_i_n pin for hard reset. Resetting initializes the internal registers, FSM. Note that external RAM register area is not initialized. Stable p_cryclk_i shall be input by the time of reset cancellation.

Soft reset

Setting soft_reset bit of CONF register to “1” performs resetting. Similar to a hard reset, the internal registers and FSM will be initialized by this. Note that external RAM register area is not initialized. Stable p_cryclk_i shall be input by the time of reset cancellation.

The reset sequence is as shown below.

Reset sequence

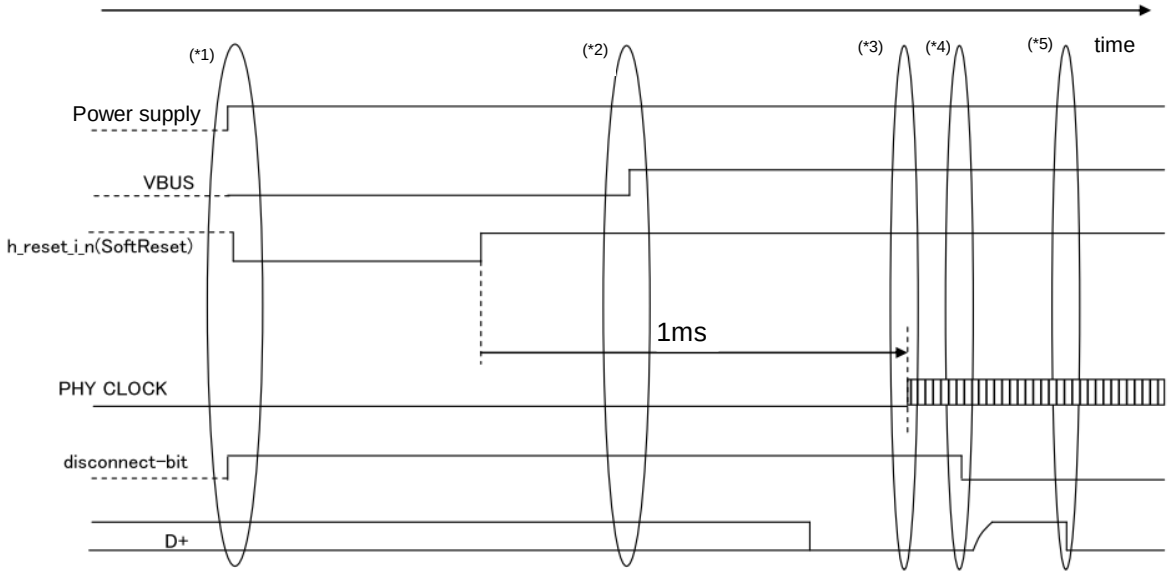


Figure 21-3

(*1) Power supply starts.
Power-on-Reset sets DEVC.disconnect to the initial value “1”.
* At the time of reset, stable frequency shall be input to p_cryclk_i.
(*2) VBUS detection confirms that USB is connected.
(*3) After h_reset_i_n is de-asserted, 1 ms passes and then PHY CLOCK starts to be supplied.
(*4) Firmware writes “0” in DEVC.disconnect, the pull-up resistor is connected, and USB host recognizes the connection.
(*5) USB host executes a USB reset.

- At (*4), disconnect cannot be cancelled until p_vbusvalid_i = 1 is set at the time of VBUS detection and that is reflected as OTGSTS:vbus_vld = 1 in the macro. The reflection time varies depending

on whether before or after h_reset_i_n assertion cancellation.

Before reset cancellation: p_vbusvalid_i change is immediately reflected in the macro.

After reset cancellation: p_vbusvalid_i change takes 30 ms to be reflected in the macro.

- Before h_reset_i_n assertion, set the system controller: HF_SEL = 0 (device), ANPD = 0 (PHY power-down cancellation), CLKSEL[2:0] and CRYCNTSW[1:0].

  (See System controller specification.)
- h_reset_i_n assertion period shall be 200 us or more, and during assertion, 4 or more cycles of p_cryclk_i shall be input.

5.8.3. Suspend (Device Mode)

The following is the relation between the signals affecting suspend operation and the suspend state.

Suspend and disconnect control

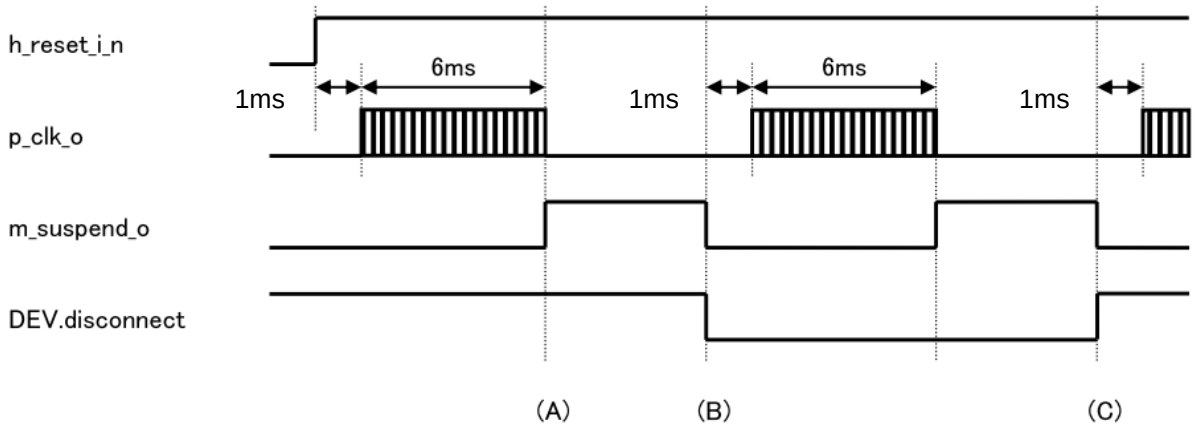


Figure 21-4

After h_reset_i_n is cancelled, if disconnect is not cancelled, USB20DC transitions to the suspend state.

(A) Suspend in the disconnect state
After reset is cancelled, if disconnect is not cancelled for 6 ms or more, this macro transitions to the suspend state and stops PHY clock.

(B) Connect in the suspend state
When disconnect is cancelled, 1 ms passes and then PHY clock restarts. When communication is not established after that, 6 ms pass and then it transitions to the suspend state.

(C) Disconnect in the suspend state
When there is disconnect in the suspend state, 1 ms passes and then PHY clock restarts. When disconnect is not cancelled, 6 ms pass and then it transitions to the suspend state (* the same state as (A)).

Suspend and VBUS change

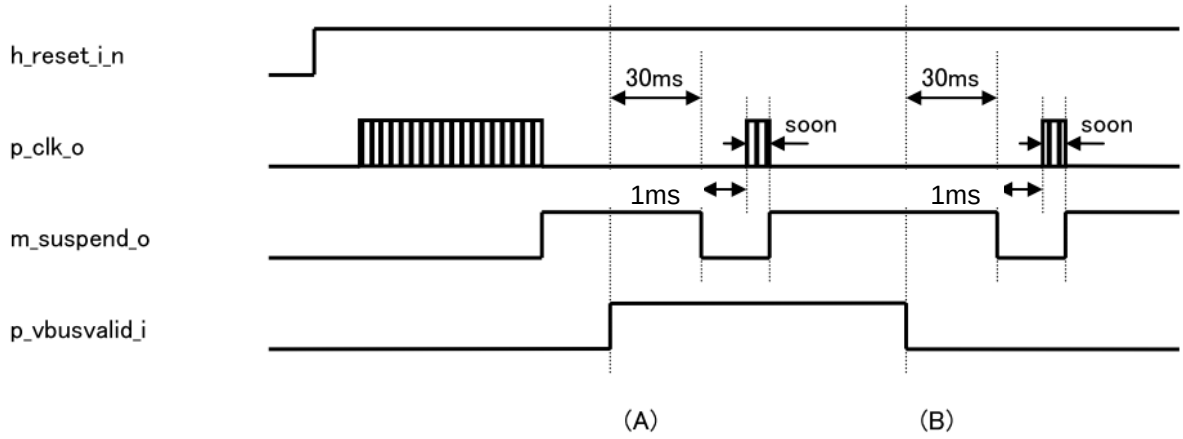


Figure 21-5

(A) VBUS change (rise) in the suspend state

When VBUS starts to be supplied in the suspend state, approximately 30 ms later, the suspend state is cancelled. Then, 1 ms passes and PHY clock restarts, but this macro immediately gets into the suspend state to stop PHY clock.

(B) VBUS change (fall) in the suspend state

When VBUS supply is stopped in the suspend state, approximately 30 ms later, the suspend state is cancelled. Then, 1 ms passes and PHY clock restarts, but this macro immediately transitions to the suspend state to stop PHY clock.

The specifications (A) and (B) are unique to USB20DC.

Suspend and USB BUS change (noise)

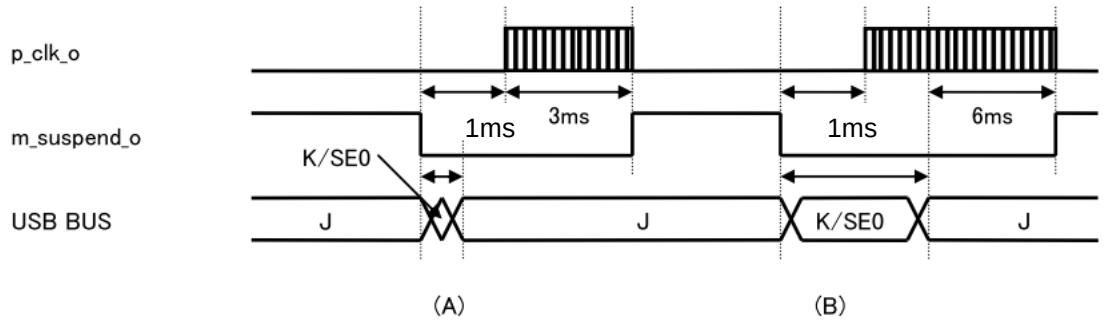


Figure 21-6

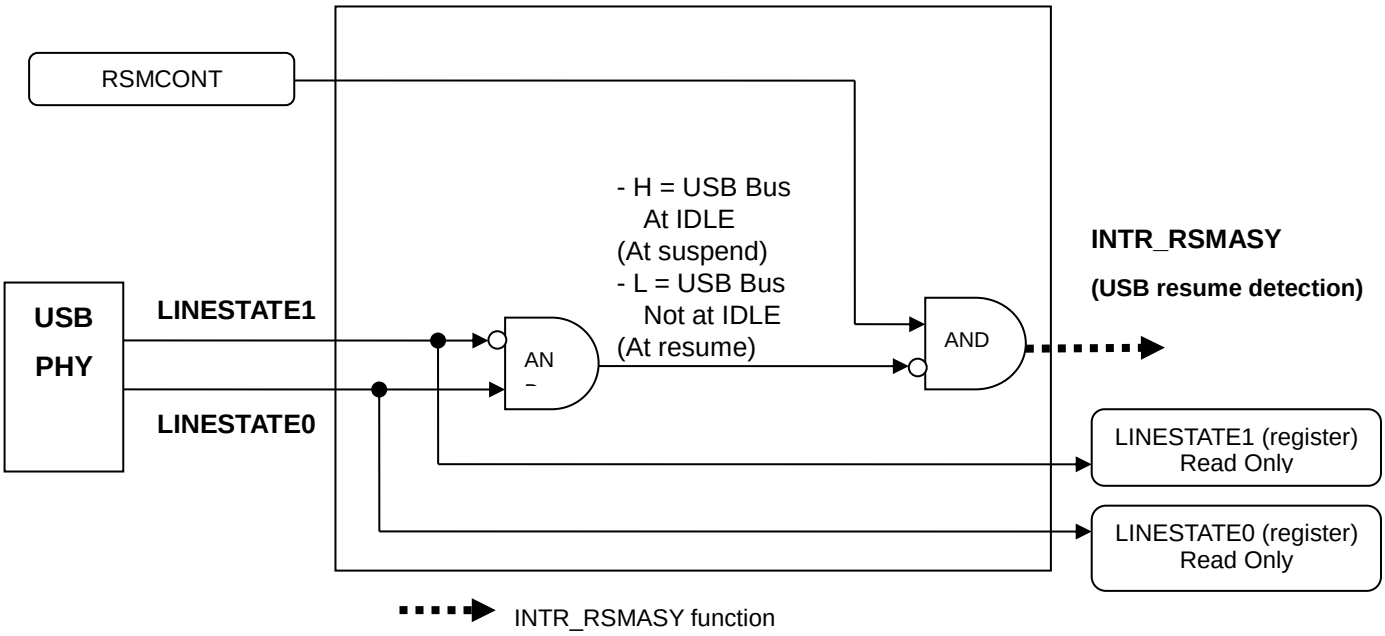
(A) USB bus change (1 ms or less noise) in the suspend state

When, in the suspend state, a period during which USB bus changes from J state to another state (noise) is 1 ms or less by the time of PHY clock restart, 1 ms passes and then PHY clock restarts, but 3 ms later, it transitions to the suspend state.

(B) USB bus change (noise more than 1 ms) in the suspend state

When, in the suspend state, a period during which USB bus changes from J state to another state (noise) is more than 1 ms, which exceeds the time of PHY clock restart, 1 ms passes and then PHY clock restarts. The noise subsides and it transitions to J state again. 6 ms later, it transitions to the suspend state.

XTAL (p_cryclk_i) oscillation stop at suspend and the oscillation start by resume



USB bus signals: LINESTATE0,1 output H to INTR_RSMASY (signals asynchronous with the internal clock) so that USB resume state can be detected to generate interrupt signal even if the internal clock is stopped.

c_intr_o and INTR_RSMASY in the timing chart below are merged to be output and then connected to the interrupt controller.

Figure 21-7

Additionally,

- RSMCONT (register) is connected to USBCNT: RSMCONT register.
- LINESTATE1 (register) is connected to USBSTAT: LINESTE1 register.
- LINESTATE0 (register) is connected to USBSTAT: LINESTE0 register.

See System controller specification for more information about these registers.

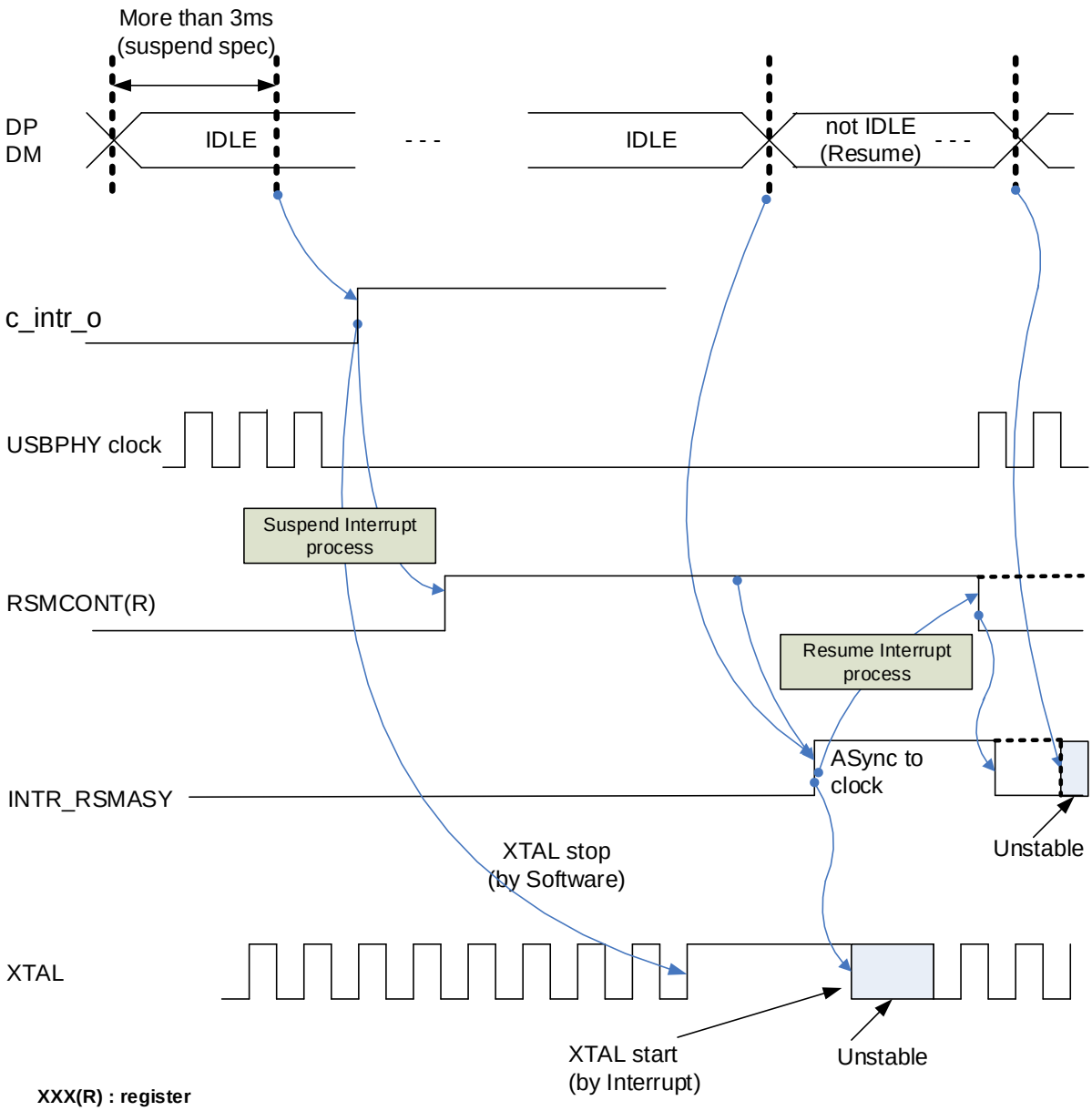

More than 3ms
(suspend spec)
DP
DM
IDLE
IDLE
not IDLE
(Resume)
c_intr_o
USBPHY clock
Suspend Interrupt process
RSMCONT(R)
Resume Interrupt process
ASync to clock
INTR_RSMASY
Unstable
XTAL stop
(by Software)
XTAL
XTAL start
(by Interrupt)
Unstable
XXX(R) : register

Figure 21-8

5.11. Power control and power-down control

ANPD pin controls the low power consumption mode of this macro when the power is turned on.

5.12. Examples of connecting and disconnecting operation

Case 1: Connecting

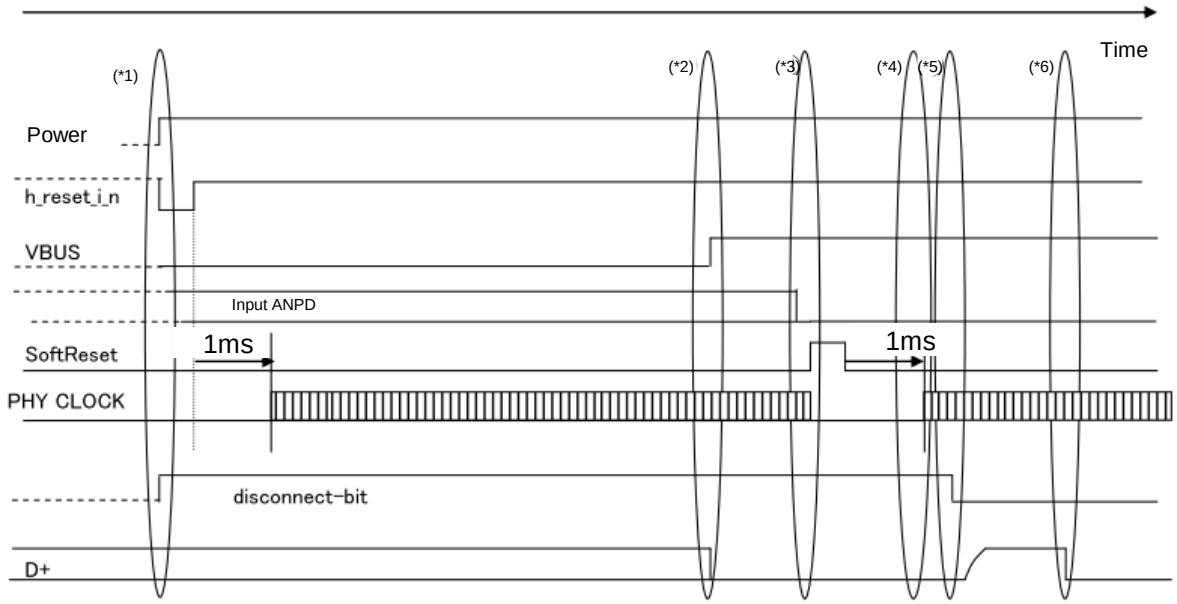


Figure 21-9

(*1) Power supply starts. Power-on-Reset is performed.

* At the time of reset, stable frequency shall be input to p_cryclk_i.

(*2) VBUS detection confirms that USB is connected.

(*3) When input ANPD is fixed at “1” and h_reset_i_n is asserted at the time of power-on, set input ANPD to “0” and perform soft reset again. Before soft reset input, input ANPD shall be set to “0”. When input ANPD is fixed at “0” and h_reset_i_n is asserted at the time of power-on, it is not necessary to perform soft reset again.

(For this product, perform soft reset.)

(*4) 1 ms after soft reset is cancelled, PHY clock starts to be supplied.

(*5) Firmware writes “0” in DEVC.disconnect, the pull-up resistor is connected, and USB host recognizes the connection.

(*6) USB host executes USB reset.

Case 2: Disconnecting

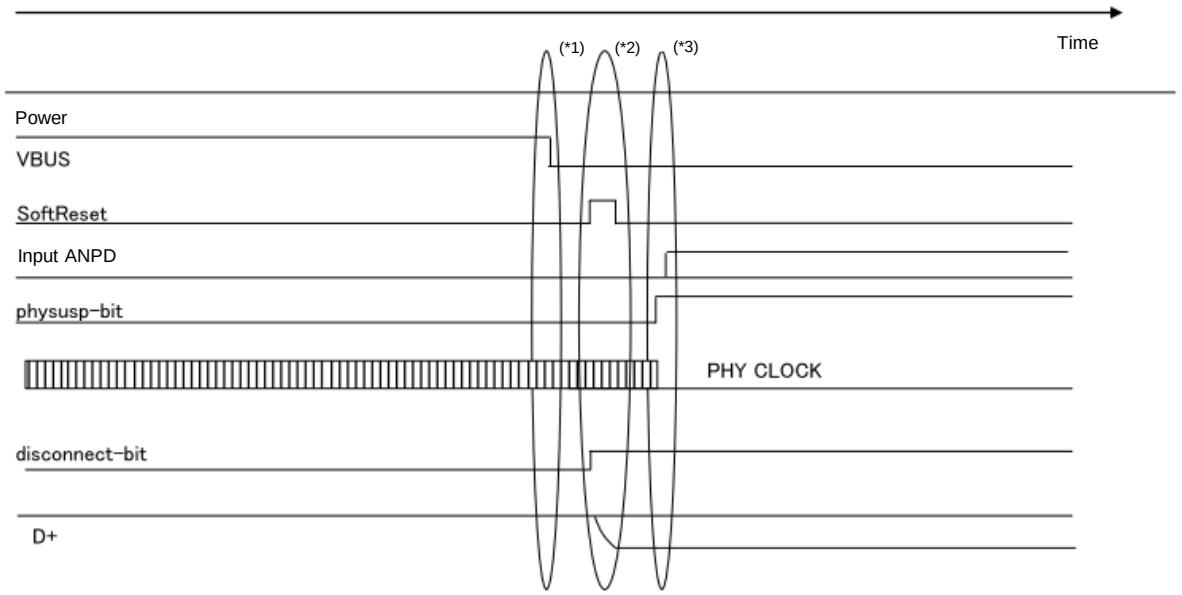


Figure 21-10

(*1) VBUS detects that USB is disconnected.

(*2) Firmware executes soft reset.

Executing soft reset sets DEVC.disconnect to “1” to disconnect the pull-up resistor.

(*3) “1” is input to input ANPD. If it does not transition to the suspend state by this time, PHY clock supply stops at this time.

Case 3: Re-connecting

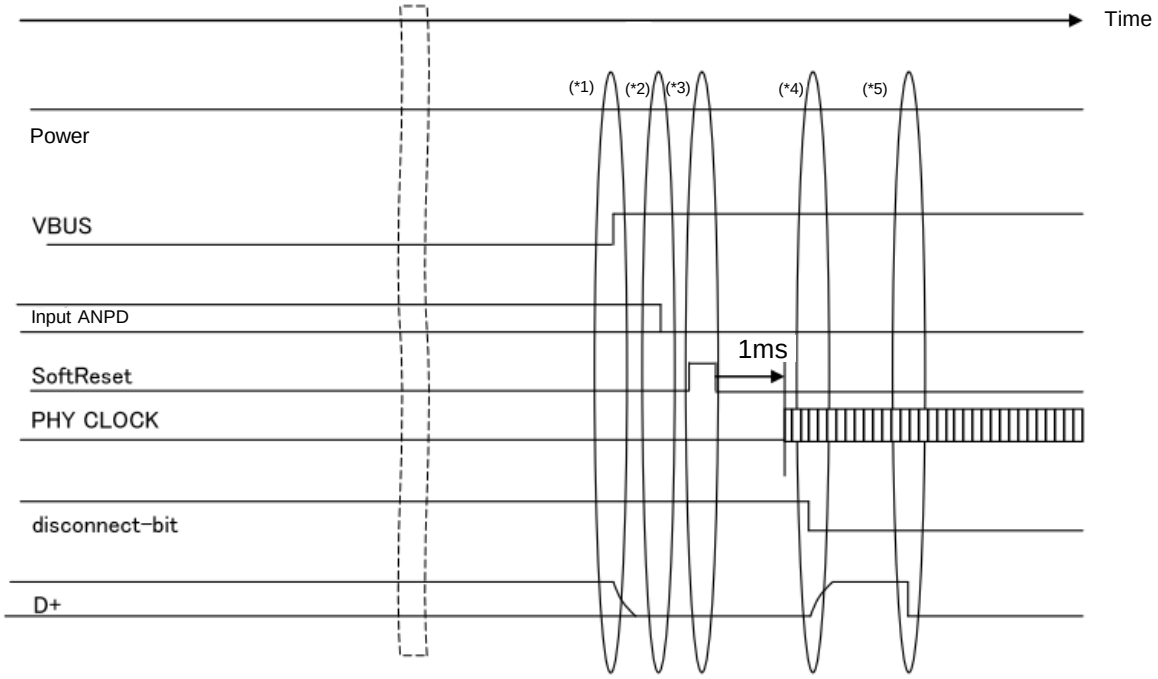


Figure 21-11

(*1) VBUS detection confirms USB is re-connected.

(*2) “0” is input to input ANPD.

By the time “0” is input to ANPD, stable frequency shall be input to p_cryclk_i.

(*3) Soft reset is executed. 1 ms later, PHY clock starts to be supplied.

(*4) Firmware writes “0” in DEVC.disconnect, the pull-up resistor is connected, and USB host recognizes the connection.

(*5) USB host executes USB reset.

21-2 Functional Overview

This chapter provides the outline of USB20DC function block in the sections below.

- USB20DC circuit configuration
- USB20DC Function

21-2-1 USB20DC circuit configuration

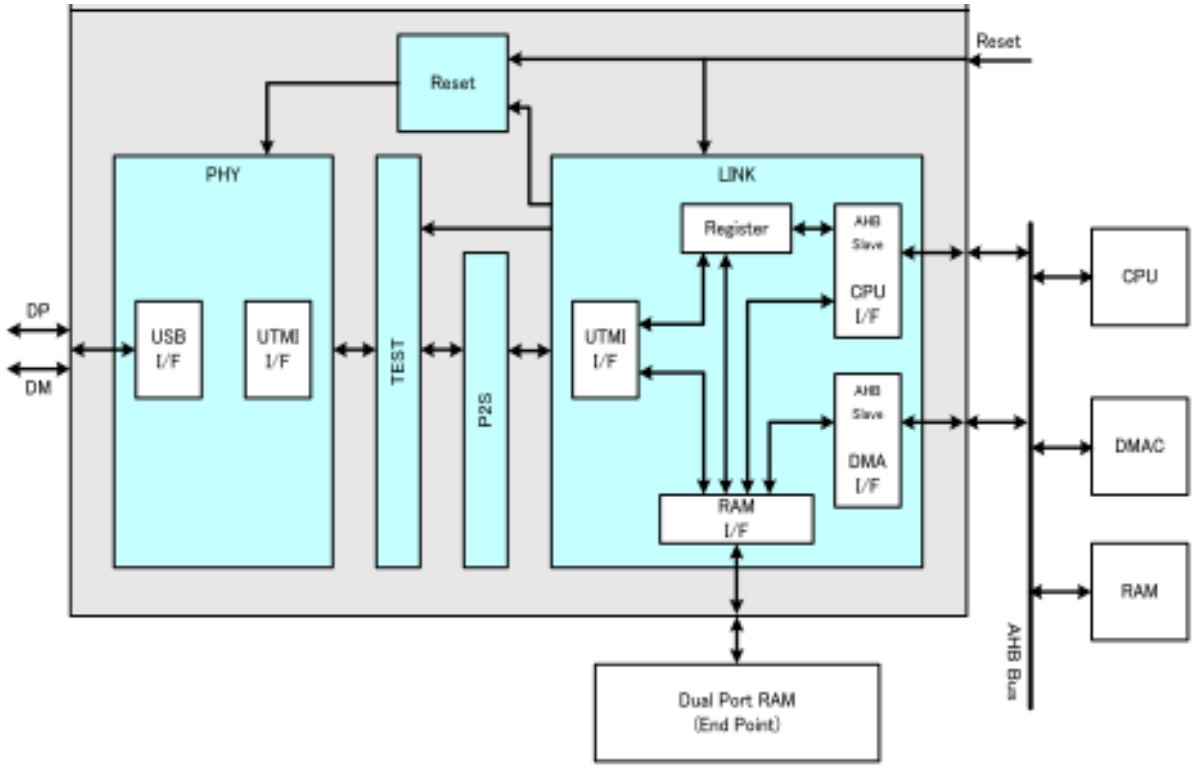


Figure 21-12

21-2-2 USB20DC Function

6. Function specification

6.1. Host function specification (This macro cannot use the host function.)

6.2. Device function specification

The following are the device functions of this macro that conform to "Universal Serial Bus Specification Rev. 2.0"

6.2.1. Endpoint

Device endpoint is hereafter referred to as EP. When a number is designated, it is numbered with # in front like EP#0. Up to 16 endpoints can be made on the connected 2-port RAM.

- EP#0 is used exclusively for control IN transfer and control OUT transfer and the buffer structure is fixed at 1 plane (1 plane for IN and OUT each).
- EP#1 to EP#15 can select any one of bulk IN/OUT transfer, interrupt IN/OUT transfer and isochronous IN/OUT transfer (IN or OUT).
- EP#1 to EP#15 each support 1- to 2-plane buffer.
- Realized by the dual port RAM externally connected to the macro. (32 KByte max, 384 Byte min) (RAM total of this product is 4.5 KByte.)
- The possible number of buffers is 32 in total of EP#0 to EP#15.
- High-speed isochronous/interrupt can designate transactions (1 to 3) that can be transmitted and received in 1 micro-frame (to support the high-band data toggle sequence).
- Each EP has 3 registers: control register (modes, EP status), configuration register (specifies buffer address, size and counter index) and counter register (PHY and APP counters). Actually, these registers are in RAM space.

See 1-3-3 Register Descriptions for the details of each register.

Figure 21-13 shows an example of the endpoint buffer structure.

(RAM total of this product is 4.5 KByte.)

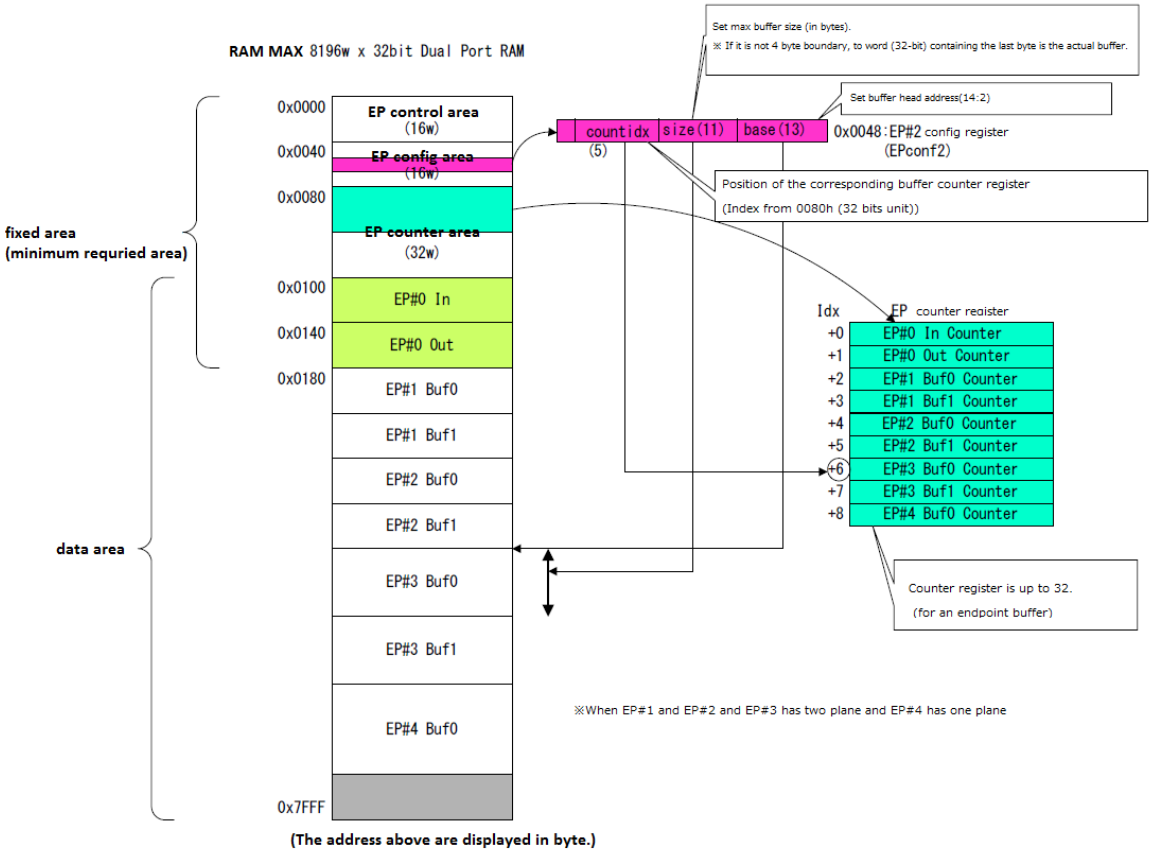


Figure 21-13 Example of the endpoint buffer structure

6.2.2. Transaction process

Table 21-1 shows the device transaction process of this macro.

Table 21-1 Transaction proces

Transfer	Dir	Token	Data	Handshake		Funciton
Control Write	Out	SETUP	-	ACK		Normal end At the time of the unjust command reception, firmware sets EP in STALL.
				none		When DATA packet is CRC error /DataToggle error, no reply (none)
	Out	OUT	-	FS	ACK	Normal end
					NAK	buffer Full state
					STALL	EP STALL state
					none	When receiving a CRC error, data more than max packet size, TimeOut
				HS	NAK	buffer Full state
					STALL	EP STALL state
					none	When receiving a CRC error, data more than max packet size, TimeOut
	-	PING	-	HS	ACK	buffer Not Full state
					NAK	buffer Full state
					STALL	EP STALL state
	In	IN	D1:NULL	-		Normal end
			-	NAK		buffer Empty state
				STALL		EP STALL state
Control Read	Out	SETUP	-	ACK		Normal end At the time of the unjust command reception, firmware sets EP in STALL.
				none		When DATA packet is CRC error /DataToggle error, no reply (none)
	In	IN	D1:(->D0)			Normal end
			-	NAK		buffer Empty state
				STALL		EP STALL state
	Out	OUT	-	ACK		Normal end
				NAK		buffer Full state
				STALL		EP STALL state
				none		When receiving a CRC error, data more than max packet size, TimeOut

* Cautions for isochronous transfer:

When IN transfer cannot receive IN token because of a packet error or other reasons, the packet is not discarded but is transmitted when the next IN token is received.

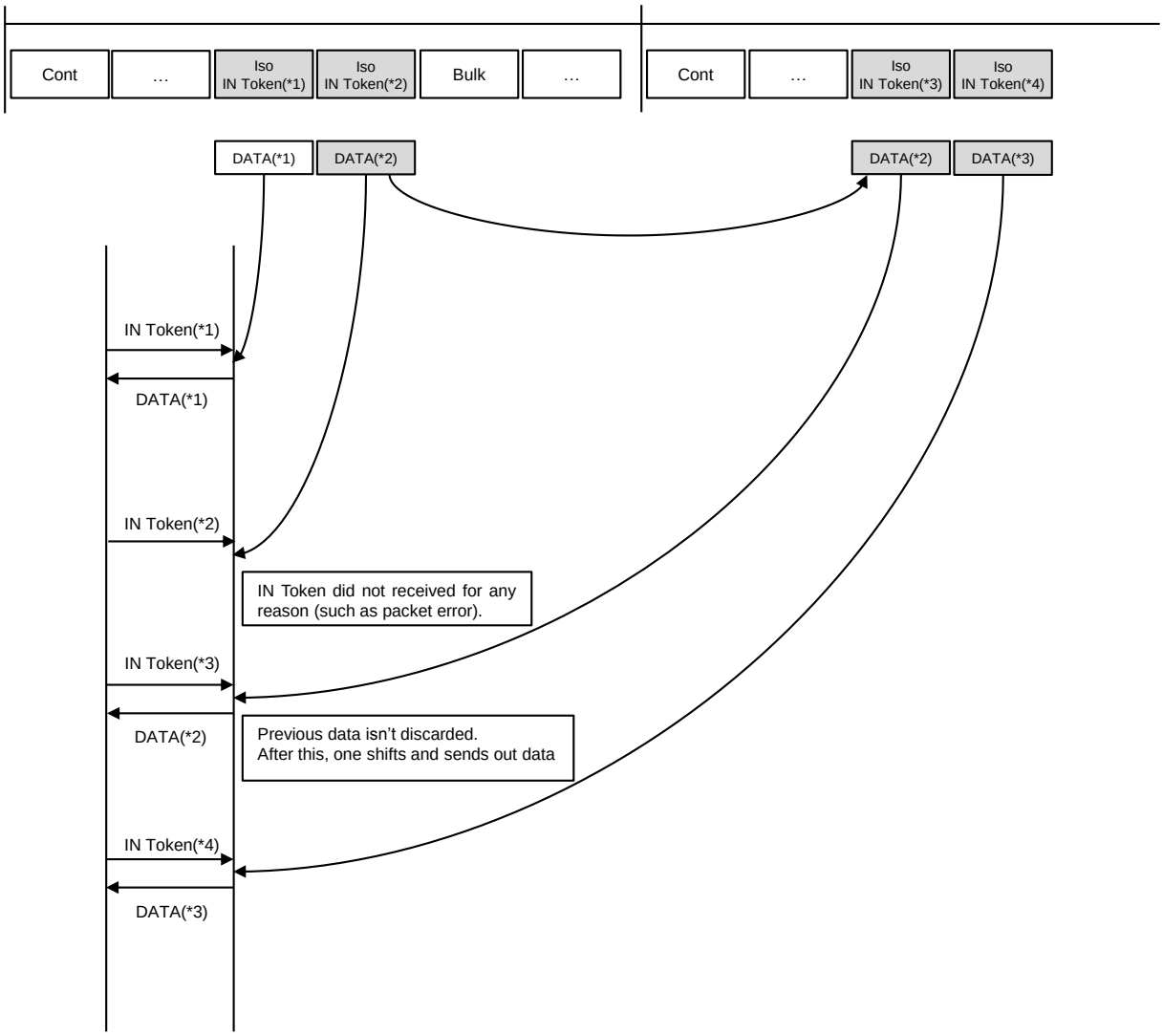


Figure 21-14

6.2.3. Standard command support

Table 21-2 shows the standard commands that need device response. This macro does not automatically respond to any standard command. So, APP needs to analyze received packets and make response packets. The process in the macro is the same as that of normal process of IN/OUT transfer data.

Table 21-2 Standard command response

Control transfer command response process

request type	for	wValue	wIndex	RqTtype (d0)	request (d1)	wValue (d3_d2)	wIndex (d5_d4)
Set Feature	dev	test_mode	Test_J	00h	03h	00_02h	01_00h
			Test_K				02_00h
			SeO_NAK				03_00h
			Test_PKT				04_00h
			Force_Eable				05_00h
Set Feature	dev	Dev_Rmt_Wkup		00h	03h	00_01h	00_00h
Clear Feature	dev	Dev_Rmt_Wkup		00h	01h	00_01h	
Set Feature	ep	EndPoint_halt	EP#n	02h	03h	00_00h	00_00h ～ 00_0Fh
Clear Feature	ep				01h		
Set/Clear Feature except the above returns Stall(ep0)							
Get Configuration	dev			80h	08h		
Get Status	dev			80h	00h		
Get Status	intf			81h	00h		
Get Status	ep			82h	00h		
Get Interface	intf			81h	0Ah		00_xxh
Get Descriptor	-			8xh	06h		
Sync Frame	ep			82h	0Ch		
Set Interface	intf			81h	0Bh		
Set Address	dev			80h	05h	00_xxh	
Set Configuration	dev			80h	09h		
Set Descriptor	-			8xh	07h		
Vendor/Class	-			x01xxxxx x10xxxxx			

6.2.4. Interrupt function

Device-issued interrupt causes are device-caused events, events that occur at each endpoint and events that occur in writing via EPCMD register.

dev_int (Bit[1]) [device cause] of INTS register is logical OR of device interrupt causes. It is cleared when each interrupt cause is cancelled.

dev_ep_int (Bit[31:16]) [each EP cause] indicates an interrupt cause that occurs at each endpoint.

After cmd_int (Bit[4]) is written via EPCMD register, an interrupt occurs.

There are some bits to enable an interrupt of each event cause, but even when it is disabled, the event itself is detected and the cause is set in the flag.

See 7.2.4 INTS, 7.4.2 DEVS, 7.2.5 EPCMD [MODE.dev_en = 1 (device mode)], 7.8.1 EPCTRL0 and 7.8.2 EPCTRL (1 to 15) for the details.

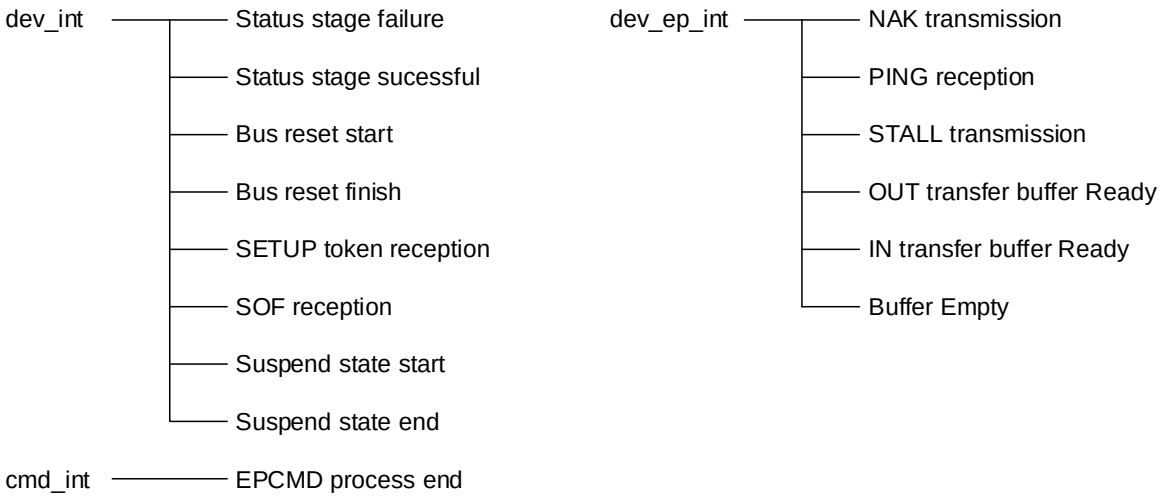


Figure 21-15

6.2.5. STALL automatic response function

Stall automatically responds when a condition below is met.

- When DATA0 of toggle error is received at the status stage of control read transfer.
- When any other data than null packet at the status stage of control read transfer.
- When the direction of a token that is received at the status stage of control transfer is opposite to the expected direction.

6.3. CPU/DMAC interface function specification

6.3.1. Address space

This macro has 2 independent ports of AHB-I/F: CPU port (c_xxxx pins) and DMAC port (d_xxxx pins). CPU port can access all the register space and RAM space in this macro, but DMAC port can access only RAM data area whose address is internally converted.

Table 21-3 shows the address map for each port.

RAM space is 32 KByte, but adequate-size RAM can be connected for necessary buffer size. However, 256 Byte needs to be set aside for other areas than data area.

(RAM total of this product is 4.5 KByte.)

Table 21-3 Address map

The area where is accessed with "C_HSEL1==1"

Start	End	word(4byte)	Area	APP	PHY
0x0000	0x7FFF		control register(FF)	R/W Burst is Single handling. Read-3cyc/Write-3cyc	R/W
0x8000	0x803F	16w	control area(RAM)	R(During initialization, writable) ※ write via EPCMD register Burst is Single handling. There is access prohibition request from the USB side. Read-3cyc/Write-3cyc	R/W
0x8040	0x807F	16w	config area(RAM)	R/W Burst is Single handling. Read-3cyc/Write-3cyc	R
0x8080	0x80FF	32w	counter area(RAM)	R/W Burst is Single handling. Read-3cyc/Write-3cyc	R/W
0x8100	0xFFFF	8Kw	data area(RAM)	R/W Read 3-1-1-1 / Write 3-1-1-1 or Read 3-2-2-2 / Write 3-2-2-2	R/W

The area where is accessed with "D_HSEL1==1"

Start	End	word	Area	APP	PHY
0x0000	0x7FFF	8Kw	DMA-CH1 address conversion area	R/W Read 3-1-1-1 / Write 3-1-1-1 or Read 3-2-2-2 / Write 3-2-2-2 HREADY = 1 is fixed output if DMA-CH1 is not enable.	inaccessible
0x8000	0xFFFF	8Kw	DMA-CH2 address conversion area	R/W Read 3-1-1-1 / Write 3-1-1-1 or Read 3-2-2-2 / Write 3-2-2-2 HREADY = 1 is fixed output if DMA-CH2 is not enable.	inaccessible

6.3.2. Access contention

In the macro, DMAC port accesses the address-conversion area that is the data area (RAM) of the address space for CPU port. So, when CPU port accesses the data area (RAM) at the same timing of DMAC port access, an access contention occurs.

(No access contention occurs when DMAC port and CPU port are connected to the same AHB bus.)

- At contention, the maximum wait is 32 cycles.

6.3.3. Byte order

In this macro, 32-bit data are always byte-accessed in little endian. A system whose AHB bus is in big endian can be supported by setting byte_order of system configuration to “1” (7.2.1 CONF register, byte_order (Bit[0]) = 1). In this case, data is swapped as shown in Figure 21-16 when data is read and imported from AHB bus or when data is output to AHB bus.

Both CPU I/F and DMA I/F swap data in units of a byte for any access: word, half word or byte.

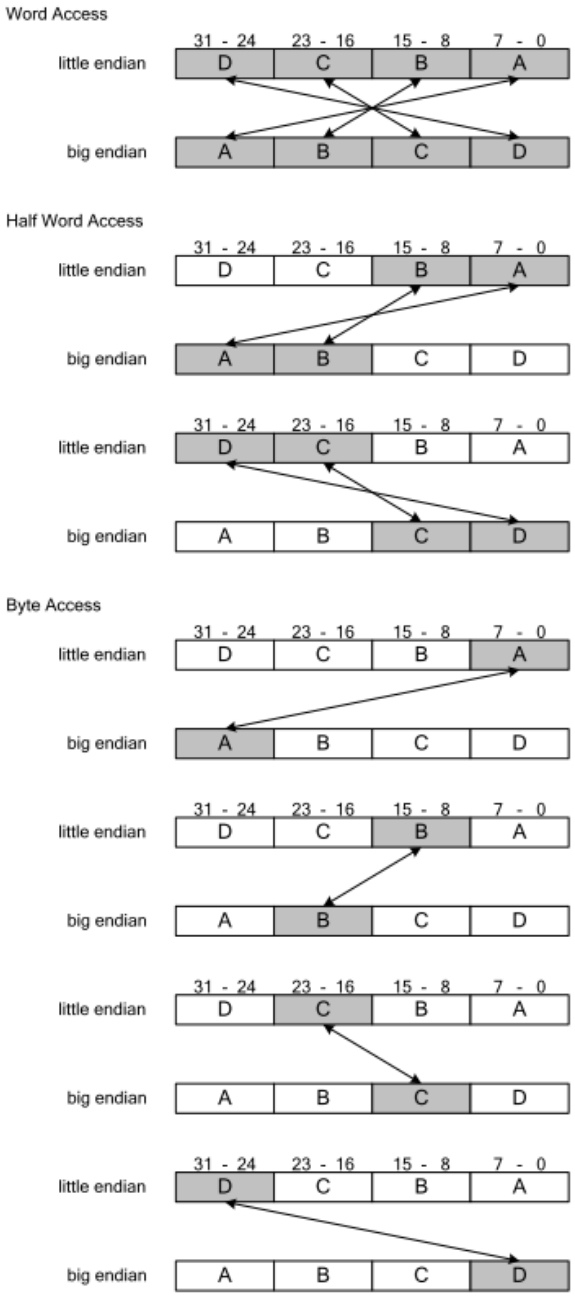


Figure 21-16

6.3.6. Limitation

The following is limitation.

- DMA transfer start address supports word-align only.

6.4. LOOPBACK function specification

DMA I/F connection can be simply confirmed with the loopback function.

The following is the area where the loopback can confirm connections.

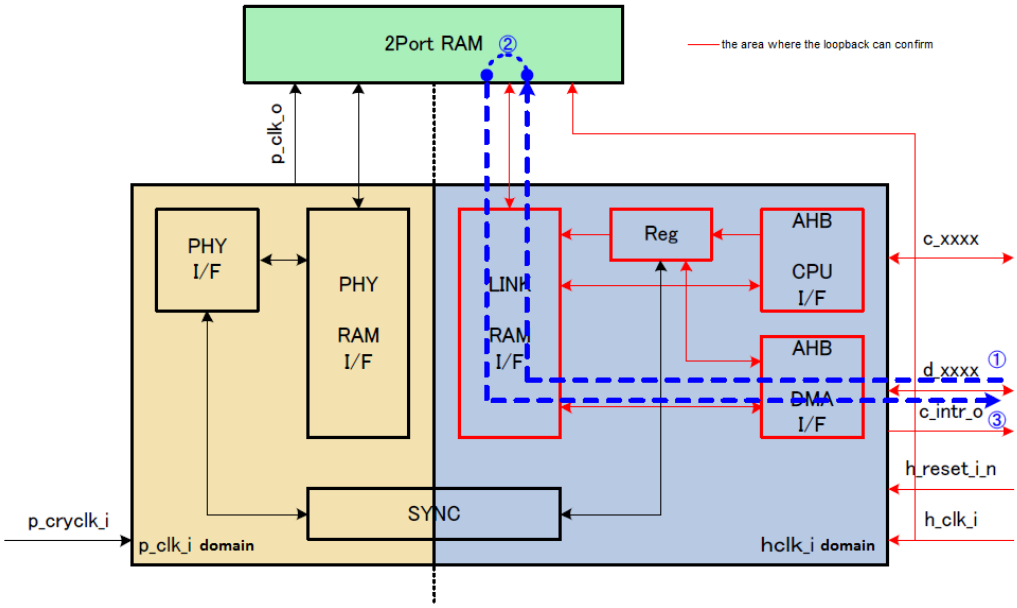


Figure 21-17

(*1) Data is written in EP (for IN transfer) kept on RAM via DMA I/F.

(*2) Setting is changed from EP (for IN transfer) to EP (for OUT transfer) via CPU I/F.

(*3) Data is read from EP (for OUT transfer) kept on RAM via DMA I/F and checked for matching the written data.

21-3 Programmer's Model

This chapter provides USB20DC registers and setting via microcontroller in the sections below.

- About the programmer's model
- Summary of USB20DC Registers
- Register Descriptions

21-3-1 About the programmer's model

USB20DC base address is not fixed, which depends on system implementation. See the system specification for base address. Each register's offset is fixed.

21-3-2 Summary of USB20DC Registers

USB20DC is controlled by the registers in "Table 21-4".

(This product cannot use registers related to the host function.)

Remark: Do not access any unspecified address from Base+0x0000 to Base+0xFFFF.

7. Register

* Do not access (write/read) the reserved address in the list of registers below.

Table 21-4 USB20DC registers

Offset	Abbreviation	Attribute	Register name	Initial value
Common				
0x0000	CONF	R/W	System configuration register	0x0200_0002
0x0004	MODE	R/W	Operation mode setting register	0x0000_0000
0x0008	INTEN	R/W	Global interrupt enable register	0x0000_0000
0x000C	INTS	R/W	Global interrupt cause register	0x0000_0000
0x0010	reserved		reserved	0x0000_0000
0x0018	CUSTOMC	R/W	PHY current amount adjustment register	0x0000_0000
~				0x0000_0000
0x003C	reserved		reserved	0x0000_0000
0x0040	EPCMD0	R/W	EP0 command register (common to host and device)	0xX000_0000
0x0044	EPCMD1	R/W	EP1 command register (common to host and device)	0xX000_0000
0x0048	EPCMD2	R/W	EP2 command register (common to host and device)	0xX000_0000
0x004C	EPCMD3	R/W	EP3 command register (common to host and device)	0xX000_0000
0x0050	EPCMD4	R/W	EP4 command register (common to host and device)	0xX000_0000
0x0054	EPCMD5	R/W	EP5 command register (common to host and device)	0xX000_0000
0x0058	EPCMD6	R/W	EP6 command register (common to host and device)	0xX000_0000
0x005C	EPCMD7	R/W	EP7 command register (common to host and device)	0xX000_0000
0x0060	EPCMD8	R/W	EP8 command register (Device)	0xX000_0000
0x0064	EPCMD9	R/W	EP9 command register (Device)	0xX000_0000
0x0068	EPCMD10	R/W	EP10 command register (Device)	0xX000_0000
0x006C	EPCMD11	R/W	EP11 command register (Device)	0xX000_0000
0x0070	EPCMD12	R/W	EP12 command register (Device)	0xX000_0000
0x0074	EPCMD13	R/W	EP13 command register (Device)	0xX000_0000
0x0078	EPCMD14	R/W	EP14 command register (Device)	0xX000_0000
0x007C	EPCMD15	R/W	EP15 command register (Device)	0xX000_0000
0x0080	reserved		reserved	
~				

0x00FC	reserved		reserved	
Host-related (Omitted)				
0x0100	reserved		reserved	
~				
0x01FC	reserved		Reserved	
Device-related				
0x0200	DEVC	R/W	Device control register	0x0000_0020
0x0204	DEVS	R/W	Device status register	0x0003_0000
0x0208	FADDR	R/W	Function address register	0x0000_0000
0x020C	TSTAMP	R	Device time stamp register	0x0000_0000
0x0210	reserved		Reserved	
~				
0x02FC	reserved		Reserved	
OTG-related				
0x0300	OTGC	R/W	OTG control register	0x0000_0000
0x 0304	reserved		reserved	
~				
0x 030C	reserved		reserved	
0x 0310	OTGSTS	R	OTG status register	0x0000_0040
0x 0314	OTGSTSC	R/W	OTG status change register	0x0000_0000
0x 0318	OTGSTSFALL	R/W	OTG status fall detection enable register	0x0000_0000
0x 031C	OTGSTSRISE	R/W	OTG status rise detection enable register	0x0000_0000
0x 0320	OTGTC	R/W	OTG timer control register	0x0000_0000
0x 0324	OTGT	R/W	OTG timer register	0x0000_0000
0x 0328	reserved		reserved	
~				
0x 03FC	reserved		reserved	
DMA-related				
0x 0400	DMAC1	R/W	DMA Ch1 control register	0x0000_0000
0x 0404	DMAS1	R	DMA Ch1 status register	0x0000_0000
0x 0408	DMATCI1	R/W	DMA Ch1 successive transfer bytes setting register	0x0000_0000
0x 040C	DMATC1	R	DMA Ch1 successive transfer bytes counter register	0x0000_0000
0x 0410	reserved		reserved	
~				
0x 041C	reserved		reserved	
0x 0420	DMAC2	R/W	DMA Ch2 control register	0x0000_0000
0x 0424	DMAS2	R	DMA Ch2 status register	0x0000_0000
0x 0428	DMATCI2	R/W	DMA Ch2 successive transfer bytes setting register	0x0000_0000
0x 042C	DMATC2	R	DMA Ch2 successive transfer bytes counter register	0x0000_0000
0x 0430	reserved		reserved	
~				
0x 04FC	reserved		reserved	
Test-related				
0x 0500	TESTC	R/W	Test control register	0x0000_0000
0x 0504	reserved		Reserved	
~				
0x 05FC	reserved		Reserved	
Device Mode RAM configuration area (Host Mode RAM area omitted)				
0x 8000	EPCTRL0	R	EP control register	0x0000_0500

0x 8004	EPCTRL1	R		0x0000_0400
0x 8008	EPCTRL2	R		0x0000_0400
0x 800C	EPCTRL3	R		0x0000_0400
0x 8010	EPCTRL4	R		0x0000_0400
0x 8014	EPCTRL5	R		0x0000_0400
0x 8018	EPCTRL6	R		0x0000_0400
0x 801C	EPCTRL7	R		0x0000_0400
0x 8020	EPCTRL8	R		0x0000_0400
0x 8024	EPCTRL9	R		0x0000_0400
0x 8028	EPCTRL10	R		0x0000_0400
0x 802C	EPCTRL11	R		0x0000_0400
0x 8030	EPCTRL12	R		0x0000_0400
0x 8034	EPCTRL13	R		0x0000_0400
0x 8038	EPCTRL14	R		0x0000_0400
0x 803C	EPCTRL15	R		0x0000_0400
0x 8040	EPCONF0	R/W	EP configuration register	0x0000_0000
0x 8044	EPCONF1	R/W		0x0000_0000
0x 8048	EPCONF2	R/W		0x0000_0000
0x 804C	EPCONF3	R/W		0x0000_0000
0x 8050	EPCONF4	R/W		0x0000_0000
0x 8054	EPCONF5	R/W		0x0000_0000
0x 8058	EPCONF6	R/W		0x0000_0000
0x 805C	EPCONF7	R/W		0x0000_0000
0x 8060	EPCONF8	R/W		0x0000_0000
0x 8064	EPCONF9	R/W		0x0000_0000
0x 8068	EPCONF10	R/W		0x0000_0000
0x 806C	EPCONF11	R/W		0x0000_0000
0x 8070	EPCONF12	R/W		0x0000_0000
0x 8074	EPCONF13	R/W		0x0000_0000
0x 8078	EPCONF14	R/W		0x0000_0000
0x 807C	EPCONF15	R/W		0x0000_0000
0x 8080	EPCOUNT0	R/W	EP counter register	0x0000_0000
0x 8084	EPCOUNT1	R/W		0x0000_0000
0x 8088	EPCOUNT2	R/W		0x0000_0000
0x 808C	EPCOUNT3	R/W		0x0000_0000
0x 8090	EPCOUNT4	R/W		0x0000_0000
0x 8094	EPCOUNT5	R/W		0x0000_0000
0x 8098	EPCOUNT6	R/W		0x0000_0000
0x 809C	EPCOUNT7	R/W		0x0000_0000
0x 80A0	EPCOUNT8	R/W		0x0000_0000
0x 80A4	EPCOUNT9	R/W		0x0000_0000
0x 80A8	EPCOUNT10	R/W		0x0000_0000
0x 80AC	EPCOUNT11	R/W		0x0000_0000
0x 80B0	EPCOUNT12	R/W		0x0000_0000
0x 80B4	EPCOUNT13	R/W		0x0000_0000
0x 80B8	EPCOUNT14	R/W		0x0000_0000
0x 80BC	EPCOUNT15	R/W		0x0000_0000
0x 80C0	EPCOUNT16	R/W		0x0000_0000
0x 80C4	EPCOUNT17	R/W		0x0000_0000
0x 80C8	EPCOUNT18	R/W		0x0000_0000
0x 80CC	EPCOUNT19	R/W		0x0000_0000

0x 80D0	EPCOUNT20	R/W		0x0000_0000
0x 80D4	EPCOUNT21	R/W		0x0000_0000
0x 80D8	EPCOUNT22	R/W		0x0000_0000
0x 80DC	EPCOUNT23	R/W		0x0000_0000
0x 80E0	EPCOUNT24	R/W		0x0000_0000
0x 80E4	EPCOUNT25	R/W		0x0000_0000
0x 80E8	EPCOUNT26	R/W		0x0000_0000
0x 80EC	EPCOUNT27	R/W		0x0000_0000
0x 80F0	EPCOUNT28	R/W		0x0000_0000
0x 80F4	EPCOUNT29	R/W		0x0000_0000
0x 80F8	EPCOUNT30	R/W		0x0000_0000
0x 80FC	EPCOUNT31	R/W		0x0000_0000

* This macro does not support USB On The Go function.

7.1. How to see the table

Bit	7	6	5	4	3	2	1	0
Field	reserved					soft_reset	burst_wait	byte_order
R/W	-					R0/W1	R/W	R/W
Initial	0					0	1	0

Bit		Bit number at access
Field		Bit field name
R/W		Read/Write attribute
	R	Can be read.
	R0	Read value is always “0”.
	R1	Read value is always “1”.
	W	Can be written.
	W0	Can be written, but only write of “0” has effect.
	W1	Can be written, but only write of “1” has effect.
Initial		Value at reset

The value read by read-modify-write or “0” shall be written in a register whose field is “reserved”.

21-3-3 Register Descriptions

7.2. Host and Device common register

21-3-3-1 System configuration register: CONF(7.2.1.)

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	soft_reset_m	burst_wait_m	byte_order_m
Reset	0	0	0	0	0	0	1	0
R/W	-	-	-	-	-	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	soft_reset	burst_wait	byte_order
Reset	0	0	0	0	0	0	1	0
R/W	-	-	-	-	-	R/W1	R/W	R/W

Name	Bit	Description
reserved	31:27	reserved
soft_reset_m	26	Mirror bit of soft_reset (Bit[2]) (Read Only)
burst_wait_m	25	Mirror bit of burst_wait (Bit[1]) (Read Only)
byte_order_m	24	Mirror bit of byte_orde (Bit[0]) (Read Only)
reserved	23:3	reserved
soft_reset	2	Performs soft reset. 0: Cancels reset. 1: Resets. * 4 cycles (h_clk_i) after reset, it automatically cancels reset.
burst_wait	1	Sets the number of waits for burst transfer cycle in the data area. 0: 0 Wait (Read 3-1-1-1/Write 3-1-1-1) 1: 1 Wait (Read 3-2-2-2/Write 3-2-2-2)
byte_order	0	Selects AHB-I/F endian. 0: Little endian 1: Big endian * See 6.3.3 Byte order for the details.

21-3-3-2 Operation mode specifying register: MODE(7.2.2.)

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	dev_addr_load_mode	dev_int_mode	dev_en	host_en
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Name	Bit	Description
reserved	31:4	reserved
dev_addr_load_mode	3	Controls address load mode. 0: After the status stage completion, application sets func_addr (Bit[6:0]) of FADDR register. * p_clk_o*3+h_clki*3 or more cycles are needed from the write in func_addr to the reflection in crt_func_addr. * If crt_func_addr is read immediately after writing func_addr, the written value cannot be read out. 1: At the status stage completion, func_addr (Bit[6:0]) of FADDR register that is preset by application is automatically loaded. The set address can be checked by crt_func_addr (Bit[22:16]) of FADDR register. * h_clk_i*3 cycles or more are needed from the status stage ACK response to the reflection in crt_func_addr. * Valid only when dev_en = 1.
dev_int_mode	2	Controls the buffer interrupt way when working as device.

<table>
<tr><td rowspan="8"></td><td rowspan="8"></td><td colspan="2">(1) In case of EP0</td></tr>
<tr><td>0</td><td>IN transfer
- Buffer Not Empty → Empty: readyi_int interrupt
OUT transfer
- Buffer Empty → Not Empty: readyo_int interrupt</td></tr>
<tr><td>1</td><td>In addition to the interrupt way of dev_int_mode = 0,

- When buffer state can be read/written, writing bufwr/bufrd of EPCMD register generates readyi_int/readyo_int interrupt.
- Writing init of EPCMD register generates readyi_int interrupt.</td></tr>
<tr><td colspan="2">(2) In case of EP1 to EP15</td></tr>
<tr><td>0</td><td>IN transfer:
- Buffer Full → Not Full: ready_int interrupt
- Buffer Not Empty → Empty: empty_int interrupt
OUT transfer:
- Buffer Empty → Not Empty: ready_int interrupt</td></tr>
<tr><td>1</td><td>In addition to the interrupt way of dev_int_mode = 0,

- When buffer state can be read/written, writing bufwr/bufrd of EPCMD register generates ready_int interrupt.
- Writing init of EPCMD register generates empty_int and ready_int interrupts at IN transfer, or empty_int interrupt at OUT transfer.
- At OUT transfer, when buffer is Not Empty → Empty, empty_int interrupt occurs.</td></tr>
<tr><td colspan="2">* Valid only when dev_en = 1.
In other cases, do not set dev_int_mode = 1.</td></tr>
<tr><td>dev_en</td><td>1</td><td colspan="2">Works as device.

0: disable
1: enable</td></tr>
<tr><td>host_en</td><td>0</td><td colspan="2">Works as host.</td></tr>
</table>

		0: disable 1: enable (Do not set this. This device’s macro cannot use the host function.)

21-3-3-3 Global interrupt enable register: INTEN(7.2.3.)

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	dev_ep_inten15	dev_ep_inten14	dev_ep_inten13	dev_ep_inten12	dev_ep_inten11	dev_ep_inten10	dev_ep_inten9	dev_ep_inten8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	dev_ep_inten7	dev_ep_inten6	dev_ep_inten5	dev_ep_inten4	dev_ep_inten3	dev_ep_inten2	dev_ep_inten1	dev_ep_inten0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	dma2_inten	dma1_inten
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	cmd_inten	phy_err_inten	otg_inten	dev_inten	host_inten
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

Name	Bit	Description
dev_ep_inten[15:0]	31:16	Controls whether to enable interrupt by dev_ep_int[15:0] (Bit[31:16]) of INTS register. 0: disable 1: enable
reserved	15:10	reserved
dma2_inten	9	Controls whether to enable interrupt by dma2_int (Bit[9]) of INTS register. 0: disable 1: enable
dma1_inten	8	Controls whether to enable interrupt by dma1_int (Bit[8]) of INTS register. 0: disable 1: enable
reserved	7:5	reserved
cmd_inten	4	Controls whether to enable interrupt by cmd_int (Bit[4]) of INTS register.

		0: disable 1: enable
phy_err_inten	3	Controls whether to enable interrupt by phy_err_int (Bit[3]) of INTS register. 0: disable 1: enable
otg_inten	2	Controls whether to enable interrupt by otg_int (Bit[2]) of INTS register. 0: disable 1: enable
dev_inten	1	Controls whether to enable interrupt by dev_int (Bit[1]) of INTS register. 0: disable 1: enable
host_inten	0	Controls whether to enable interrupt by host_int (Bit[0]) of INTS register. 0: disable 1: enable

21-3-3-4 Global interrupt cause register: INTS(7.2.4.)

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	dev_ep_int15	dev_ep_int14	dev_ep_int13	dev_ep_int12	dev_ep_int11	dev_ep_int10	dev_ep_int9	dev_ep_int8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	dev_ep_int7	dev_ep_int6	dev_ep_int5	dev_ep_int4	dev_ep_int3	dev_ep_int2	dev_ep_int1	dev_ep_int0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	dma2_int	dma1_int
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R/W0	R/W0
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	cmd_int	phy_err_int	otg_int	dev_int	host_int
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W0	R/W0	R	R	R

Name	Bit	Description
dev_ep_int[15:0]	31:16	Logical OR of interrupt causes generated at each device endpoint ■ Set condition It is set when any of conditions below is met at each device endpoint. ○ EP0 - When IN transfer buffer is ready. - When OUT transfer buffer is ready. - When PING token is received. - When NAK responds. - When STALL responds. ○ EP1 to 15 - When IN transfer buffer is ready. - When OUT transfer buffer is ready. - When PING token is received. - When NAK responds. - When STALL responds. - When buffer is empty.

		■ Clear condition - When interrupt cause at each endpoint is cancelled. - When bus reset starts. (Bus reset clears all dev_ep_int[15:0].) * Does not work when host_en = 1 (host mode) of MODE register.
reserved	15:10	reserved
dma2_int	9	Indicates interrupt caused by DMA Ch2. ■ Set condition - DMA Ch2 transfer completion - DMA Ch2 transfer disabled - DMA Ch2 transfer forced termination ■ Clear condition - Write of “0”
dma1_int	8	Indicates interrupt caused by DMA Ch1. ■ Set condition - DMA Ch1 transfer completion - DMA Ch1 transfer disabled - DMA Ch1 transfer forced termination ■ Clear condition - Write of “0”
reserved	7:5	reserved
cmd_int	4	Indicates interrupt by EPCMD write completion ■ Set condition - EPCMD write completion ■ Clear condition - Write of “0”
phy_err_int	3	Indicates interrupt by PHY_Erratic_Error. ■ Set condition When a following condition continues for 2 ms or more at HS.

		- UTMI 16 Bit: p_rxactive = 1 or p_rxvalid = 1 - UTMI 8 Bit: p_rxactive = 1 (for this product) ■ Clear condition - Write of “0”
otg_int	2	Indicates interrupt caused by OTG. ■ Set condition Interrupt occurs when any of the conditions below is met. - When it is detected that SE0 state continues for 2 ms. - When ID pin changes. - When port changes. - When OTG timer changes. ■ Clear condition - It is cleared when each OTG interrupt cause is cancelled.
dev_int	1	Logical OR of the device interrupt causes (except EP-related) ■ Set condition Interrupt occurs when any of the conditions below is met. - Status stage success/failure - Suspend start/end - SOF reception - SETUP token reception - Bus reset start/end ■ Clear condition - It is cleared when each device interrupt cause is cancelled.
host_int	0	Logical OR of the host interrupt causes. ■ Set condition Interrupt occurs when any of the conditions below is met. - SOF transmission - Frame number rollover - When connection state changes. - When an error occurs at port. - When suspend state changes.

		- When port over-current state changes. - When bus reset succeeds. - When transfer process is completed at each defined EP buffer. ■ Clear condition - It is cleared when each host interrupt cause is cancelled.

21-3-3-5 Custom control register: CUSTOMC(APPENDIX 1.1.)

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	hsdsel3	hsdsel2	hsdesel1	hsdesel0
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Name	Bit	Description
reseverd	31:4	reserved
hsdsel	3:0	Adjusts the amount of HS driver current. 4’b0000: Normal current value 4’b0001: 5% increase of normal current value 4’b0010: 10% increase of normal current value Attention: Do not set anything other than those above.

21-3-3-6 EP command register: EPCMD * (0–15) (7.2.5.)

[Base + 0x0040 ~ 0x007C]

MODE.host_en = 1 (host mode)

(Omitted because this product cannot use the host function.)

MODE.dev_en = 1 (device mode)

Bit	31	30	29	28	27	26	25	24
Name	epcmd_busy	-	hiband1	hiband0	bnum1	bnum0	dir	et1
Reset	0	X	0	0	0	0	0	0
R/W	R	-	W	W	W	W	W	W
Bit	23	22	21	20	19	18	17	16
Name	et0	nack_int_clr	stalled_int_clr	ping_int_clr	readyo_int_clr(EP#0) empty_int_clr(EP#1-15)	readyi_int_clr (EP#0) ready_int_clr (EP#1-15)	-	nack_inten
Reset	0	0	0	0	0	0	0	0
R/W	W	W1	W1	W1	W1	W1	-	W
Bit	15	14	13	12	11	10	9	8
Name	stalled_inten	ping_inten	readyo_inten(EP#0) empty_inten (EP#1-15)	readyi_inten (EP#0) ready_inten (EP#1-15)	write_en	nackresp	nullresp	toggle_clr
Reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W1
Bit	7	6	5	4	3	2	1	0
Name	toggle_set	stall_clr	stall_set	bufrd	bufwr	init	Stop	start
Reset	0	0	0	0	0	0	0	0
R/W	W1	W1	W1	W1	W1	W1	W1	W1

Name	Bit	Description
epcmd_busy	31	Indicates whether EPCMD is busy. 0: EPCMD can be written. 1: EPCMD is busy. * During epcmd_busy = 1, write-access is inhibited. During this period, write has no effect.
reserved	30	reserved
hiband[1:0]	29:28	Sets the number of packets transmitted and received for 1 μFrame (EPCTRL hiband[1:0] (Bit[15:14])). 00: 1

		01: 1 10: 2 11: 3 * Valid only when write_en = 1. * EP0 cannot be set.
bnum[1:0]	27:26	Sets the number of buffers (EPCTRL bnum[1:0] (Bit[5:4])). 00: Single 01: Double 10: reserved 11: reserved * Valid only when write_en = 1. * EP0 cannot be set. (Write “0”.)
dir	25	Sets the transfer direction (EPCTRL dir (Bit[3])). 0: OUT 1: IN * Valid only when write_en = 1. * EP0 cannot be set. (Write “0”.)
et[1:0]	24:23	Sets the transfer type (EPCTRL1-15 et[1:0] (Bit[2:1])). 00: reserved 01: Isochronous 10: Bulk 11: Interrupt * Valid only when write_en = 1. * EP0 cannot be set. (Write “0”.)
nack_int_clr	22	Clears NAK packet transmission interrupt cause (EPCTRL nack_int (Bit[30])). 1: Clears.
stalled_int_clr	21	Clears STALL interrupt cause (EPCTRL staled_int (Bit[29])). 1: Clears.
ping_int_clr	20	Clears the ping packet reception interrupt cause (EPCTRL ping_int (Bit[28])). 1: Clears.
readyo_int_clr(EP#0)	19	Clears OUT buffer ready interrupt cause (EPCTRL0 readyo_int (Bit[27])). 1: Clears.
empty_int_clr(EP#1-15)		Clears the buffer empty interrupt cause (EPCTRL#1-15

		empty_int (Bit[27])). 1: Clears.
readyi_int_clr(EP#0)	18	Clears IN buffer ready interrupt cause (EPCTRL0 readyi_int (Bit[26])). 1: Clears.
ready_int_clr(EP#1-15)		Clears the buffer ready interrupt cause (EPCTRL#1-15 ready_int (Bit[26])). 1: Clears.
reserved	17	reserved * Always write “0”.
nack_inten	16	Controls whether to enable NAK interrupt (EPCTRL nack_int (Bit[30])) function. 0: disable 1: enable * Valid only when write_en = 1.
stalled_inten	15	Controls whether to enable STALL interrupt (EPCTRL staled_int (Bit[29])) function. 0: disable 1: enable * Valid only when write_en = 1.
ping_inten	14	Controls whether to enable PING interrupt (EPCTRL ping_int (Bit[28])) function. 0: disable 1: enable * Valid only when write_en = 1.
readyo_inten(EP#0)	13	Controls whether to enable the readyo interrupt (EPCTRL#0 readyi_int (Bit[27])) function. 0: disable 1: enable * Valid only when write_en = 1.
empty_inten(EP#1-15)		Controls whether to enable the empty interrupt (EPCTRL#1-15 empty_int (Bit[27])) function.

		0: disable 1: enable * Valid only when write_en = 1.
readyi_inten(EP#0)	12	Controls whether to enable the readyi interrupt (EPCTRL#0 readyi_int (Bit[26])) function. 0: disable 1: enable * Valid only when write_en = 1.
ready_inten(EP#1-15)		Controls whether to enable the ready interrupt (EPCTRL#1-15 ready_int (Bit[26])) function. 0: disable 1: enable * Valid only when write_en = 1.
write_en	11	Enables the values set in EPCMD. 0: Enables writing EPCMD Bit0–8 and 18–22. 1: Enables writing EPCMD Bit0–29.
nackresp	10	Sets NAK response (EPCTRL#1-15 nackresp (Bit[17])). NAK response to Bulk IN/OUT transfer 0: Does not respond with NAK. 1: Responds with NAK. * Do not set this and nullresp at the same time. * Valid only when write_en = 1. * Valid only for EPCTRL#1-15. * At the time of a toggle error, it responds with ACK even when “1” is set.
nullresp	9	Sets NULL response (EPCTRL#1-15 nullresp (Bit[16]). NULL response to Bulk/Interrupt IN transfer 0: Does not respond with NULL. 1: Responds with NULL. * Do not set this and nackresp at the same time.

		* Valid only when write_en = 1. * Valid only for EPCTRL#1-15. * At the time of a toggle error, it responds with ACK even when “1” is set.
toggle_clr	8	Sets toggle = 0. 1: Sets EPCTRL toggle = 0. * Do not set this and toggle_set at the same time.
toggle_set	7	Sets toggle = 1. 1: Sets EPCTRL toggle = 1. * Do not set this and toggle_clr at the same time.
stall_clr	6	Sets stall = 0. 1: Sets EPCTRL stall = 0. * Do not set this and stall_set at the same time.
stall_set	5	Sets stall = 1. 1: Sets EPCTRL stall = 1. * Do not set this and stall_clr at the same time. * In the case of EP0, this setting has no effect even when DEVS.setup_int = 1.
bufrd	4	Sets the buffer read completion. 1: Buffer read is completed. It is set after reading buffer. appptr is automatically incremented. (When bnum is exceeded, it returns to 0.) When a new appptr equals phyptr, empty = 1 and full = 0 are set. * Do not set this and bufwr at the same time. * Do not set this and init at the same time.

		* In the case of EP0, this setting has no effect even when DEVS.setup_int = 1.
bufwr	3	Sets buffer write completion. 1: Buffer write is completed. It is set after writing buffer. appptr is automatically incremented. (When bnum is exceeded, it returns to 0.) When a new appptr equals phyptr, empty = 0 and full = 1 are set. * Do not set this and bufrd at the same time. * Do not set this and init at the same time. * In the case of EP0, this setting has no effect even when DEVS.setup_int = 1.
init	2	Initializes buffer pointer. 1: Initializes buffer pointer. Initialization sets the values as below.
		EP#0 EPCTRL0 register emptyi (Bit[8]) = 1, fulli (Bit[9]) = 0, emptyo (Bit[10]) = 1, fullo (Bit[11]) = 0
		EP#1-15, EPCTRL#1-15 register appptr (Bit[7:6]) = 0, phyptr (Bit[9:8]) = 0, empty (Bit[10]) = 1, full (Bit[11]) = 0
		* Initialization shall be performed when ep_en = 0 of EPCTRL0-15 register.
stop	1	Stops the endpoint. 1: Sets ep_en = 0 of EPCTRL register. * Do not set this and start at the same time. * When it is forcibly stopped, 2 μFrame or more shall pass at HS

		and 2 Frame or more shall pass at FS before each setting.
start	0	Starts the endpoint. 1: Sets ep_en = 1 of EPCTRL register. * Do not set this and stop at the same time.

Attention (for EPCMD setting):

Do not set EPCMD during the period from EPCMD setting to the reflection of it in HCEPCTRL register (for host) or EPCTRL register (for device).

Before setting EPCMD, make sure that EPCMD.epcmd_busy can be set (epcmd_busy = 0).

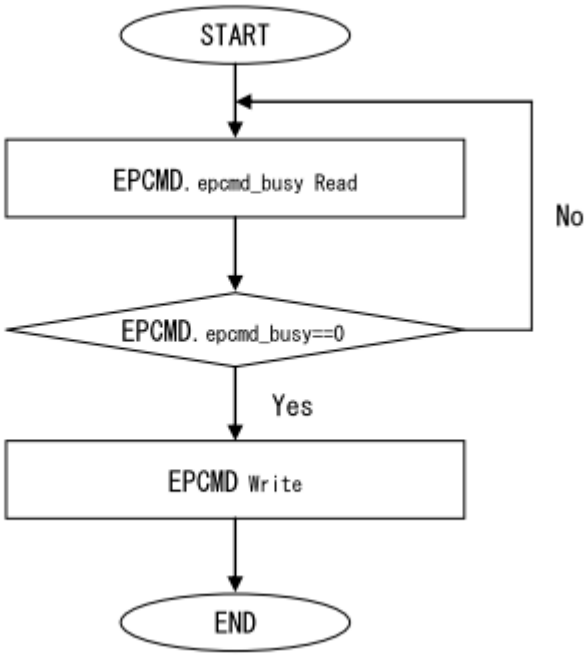


Figure 21-18

7.3. Host-related registers

(Omitted because this product cannot use the host function.)

7.4. Device-related registers

21-3-3-7 Device control register: DEVC(7.4.1.)

[Base + 0x0200]

Bit	31	30	29	28	27	26	25	24
Name	status_ng_inten	status_ok_inten	usbrstb_inten	usbrste_inten	setup_inten	sof_inten	supendb_inten	supende_inten
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	physusp
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	disconnect	-	enrmtwkup	reqresume	-	reqspeed
Reset	0	0	1	0	0	0	0	0
R/W	-	-	R/W	-	R/W	R/W1	-	R/W

Name	Bit	Description
status_ng_inten	31	Controls whether to enable interrupt by status_ng_int (Bit[31]) of DEVS register. 0: disable 1: enable * Valid only for EP0 control transfer.
status_ok_inten	30	Controls whether to enable interrupt by status_ok_int (Bit[30]) of DEVS register. 0: disable 1: enable * Valid only for EP0 control transfer.
usbrstb_inten	29	Controls whether to enable interrupt by usbrstb_int (Bit[29]) of DEVS register. 0: disable 1: enable
usbrste_inten	28	Controls whether to enable interrupt by usbrste_int (Bit[28]) of DEVS register. 0: disable 1: enable

setup_inten	27	Controls whether to enable interrupt by setup_int (Bit[27]) of DEVS register. 0: disable 1: enable
sof_inten	26	Controls whether to enable interrupt by sof_int (Bit[26]) of DEVS register. 0: disable 1: enable
suspendb_inten	25	Controls whether to enable interrupt by suspendb_int (Bit[25]) of DEVS register. 0: disable 1: enable
suspende_inten	24	Controls whether to enable interrupt by suspende_int (Bit[24]) of DEVS register. 0: disable 1: enable
reserved	23:17	reserved
physusp	16	Forcibly puts PHY block into the suspend state. 0: Normal state (Controls p_suspend_o_n according to USB protocol.) 1: Forcibly puts PHY into the suspend state. (Outputs p_suspend_o_n = 0.) * Do not set “1” during USB communication.
reserved	15:6	reserved
disconnect	5	Designates connect or disconnect. 0: Connected. 1: Disconnected. (non-Driving is output to PHY.) (p_opmode[1:0] = 01)
reserved	4	reserved
enrmtwkup	3	Sets the remote-wakeup function. 0: Disables the remote-wakeup function. 1: Enables the remote-wakeup function. When this bit is “1”, a resume request with reqresume = 1 is Valid. * It is automatically cleared by DisConnect/UsbReset.
reqresume	2	Requests a resume. Writing “1” issues a request for resuming from the suspend state. After “1” is written, PHY suspend is cancelled and p_clki is output, it is automatically cleared.

		* Do not write “1” in any other state than the suspend state. This bit’s resume request is Valid only when enrmtwkup = 1.
reserved	1	reserved
reqspeed	0	Sets PHY speed for device connection. 0: HS (When the host does not support HS, it shall be FS.) 1: FS

21-3-3-8 Device status register: DEVS(7.4.2.)

[Base + 0x0204]

Bit	31	30	29	28	27	26	25	24
Name	status_ng_int	status_ok_int	usbrstb_int	usbrste_int	setup_int	sof_int	supendb_int	supende_int
Reset	0	0	0	0	0	0	0	0
R/W	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0	R/W0
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	crtspeed	phyreset
Reset	0	0	0	0	0	0	1	1->0
R/W	-	-	-	-	-	-	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	busreset	suspend
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R	R

Name	Bit	Description
status_ng_int	31	Indicates an interrupt that control transfer fails. ■ Set condition - Failure in control transfer The following are the conditions of control transfer failure. When an interrupt occurs at the data stage and the status stage. - Sending STALL - When the next setup token is received before a status stage is completed. (When a bus reset occurs during control transfer, no interrupt occurs if SETUP is received because the state is initialized.) When an interrupt occurs only at the status stage. - Data1 of the status stage is not NULL. - Timeout - When the direction of a received token is opposite to the expected direction. - When a condition below is met at PID check after receiving IN/OUT token. - OUT transfer: PID other than Data1 is received. - IN transfer: PID other than ACK is received.

		■ Clear condition - Write of “0” * Valid only for EP0 control transfer.
status_ok_int	30	Indicates an interrupt that a status stage succeeds. ■ Set condition - When handshake is normally executed at the status stage. ■ Clear condition - Write of “0” * Valid only for EP0 control transfer.
usbrstb_int	29	Indicates an interrupt by bus reset start. ■ Set condition - Bus reset start ■ Clear condition - Write of “0”
usbrste_int	28	Indicates an interrupt by bus reset end. ■ Set condition - Bus reset end ■ Clear condition - Write of “0”
setup_int	27	Indicates an interrupt by starting the setup stage. ■ Set condition - Setup stage start (An interrupt event is timed to occur after handshake (ACK) is transmitted at the setup stage.) ■ Clear condition - Write of “0” * Settings of bufwr, bufrd and stall_set of EPCMD0 are not Valid when setup_int = 1.

sof_int	26	Indicates an interrupt by SOF or RSOF (RecoverySOF) reception. ■ Set condition - Reception of SOF or RSOF (RecoverySOF) ■ Clear condition - Write of “0” * RSOF (RecoverySOF) RSOF is an SOF signal that is generated in the macro when SOF is not transmitted from the host or when the macro cannot receive SOF from the host. RSOF generating counter is enabled by receiving SOF from the host. The counter outputs RSOF signal when SOF is not received for more than the definite period of time (HS standard micro-frame interval value (125.0625 μsec) or FS standard frame interval value (1.000500 msec)). When any of the conditions below is met, RSOF is disabled from being issued. - Bus reset - Suspend state - Resume state - Disconnect state - Phy_Erratic_Error state
suspendb_int	25	Indicates an interrupt by starting the suspend state. ■ Set condition - Suspend state start ■ Clear condition - Write of “0”
suspende_int	24	Indicates an interrupt by ending the suspend state. ■ Set condition - Suspend state end ■ Clear condition - Write of “0”

reserved	23:18	reserved
crtspeed	17	Indicates the current connection speed. 0: HS 1: FS
phyreset	16	Monitors PHY reset state. Indicates the value of p_reset_i pin.
reserved	15:2	reserved
busreset	1	Indicates that the host is resetting the bus. ■ Set condition - FS, in the case of bus reset during the non-suspend state: when D+/D- of SE0 continues for 5.3 μs. - HS, in the case of bus reset during the non-suspend state: when D+/D- of IDEL continues for 3 ms, the pull-up resistor is connected, and 500 μs later, it is in the SE0 state. - In the case of reset during the suspend state: when D+/D- changes from J-state to SE0, 3 ms later PHY block starts to input clock, and then SE0 continues for 5.3 μs. ■ Clear condition - HS connection: 5.3 μs after chirp completion. - FS connection: when the bus changes into J-state.
suspend	0	Indicates the suspend state. ■ Set condition - When IDLE state on USB bus continues for 3 ms or more at HS and then IDLE state continues for 3 ms or more at FS. - When IDLE state continues for 6 ms or more at FS. - When DEVC.physup = 1 is set. ■ Clear condition - When DEVC.physusp = 0 and the bus state changes from J-state during the suspend state.

Process of cancelling the setup command

Firmware can cancel a setup command with status_ng_int (Bit[31]) and status_ok_int (Bit[30]). See the flowchart below for cancelling.

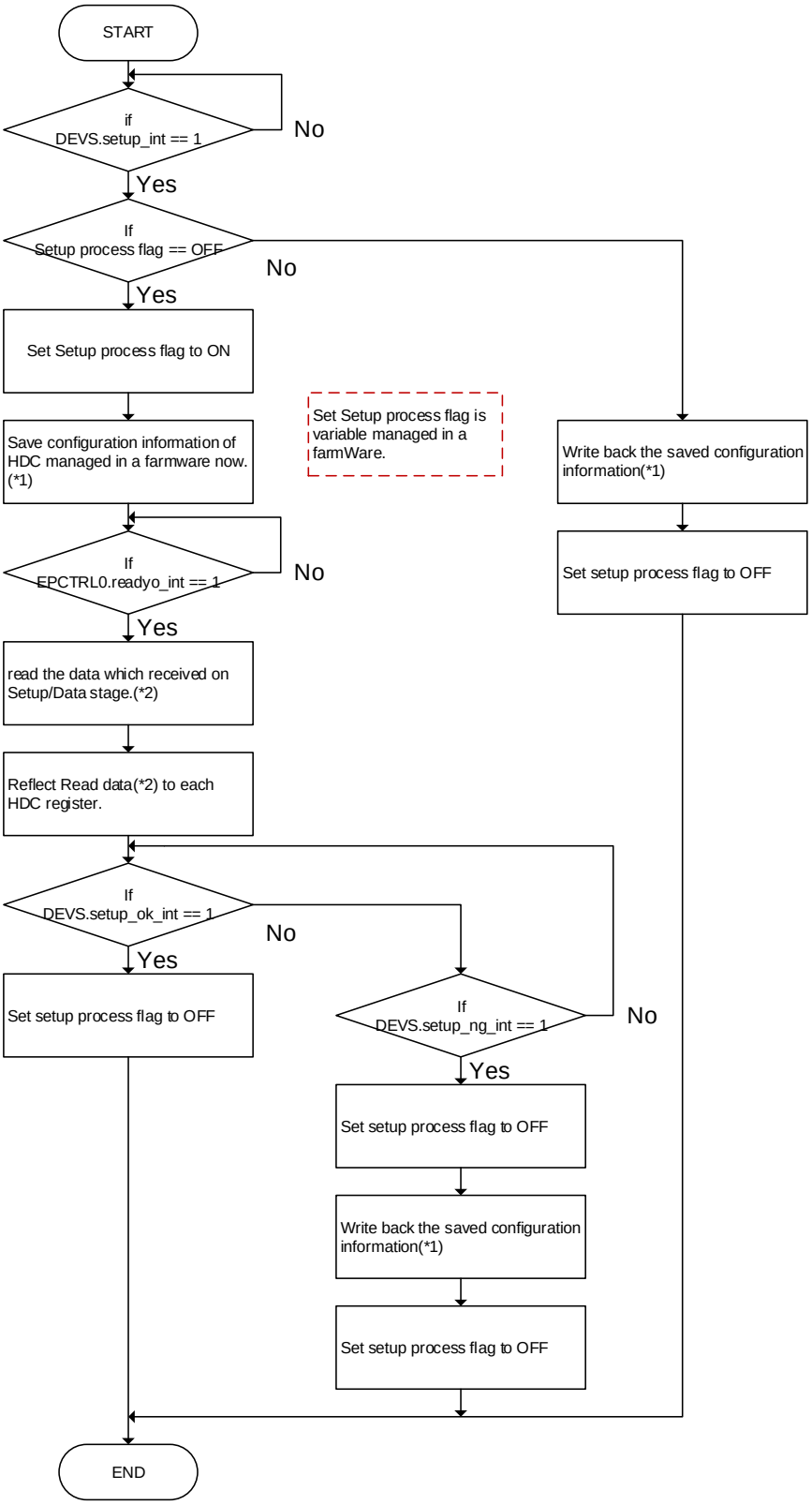


Figure 21-19

21-3-3-9 Function address register: FADDR(7.4.3.)

[Base + 0x0208]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	crt_func_addr6	crt_func_addr5	crt_func_addr4	crt_func_addr3	crt_func_addr2	crt_func_addr1	crt_func_addr0
Reset	0	0	0	0	0	0	0	0
R/W	-	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	dev_configured
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	func_addr6	func_addr5	func_addr4	func_addr3	func_addr2	func_addr1	func_addr0
Reset	0	0	0	0	0	0	0	0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Limitation: Bus reset initializes the values.

Name	Bit	Description
reserved	31:23	reserved
crt_func_addr[6:0]	22:16	Checks a device address that is currently used. * Read the same value twice for checking.
reserved	15:9	reserved
dev_configured	8	Indicates the configured state. When application recognizes the set configuration, set “1”.
reserved	7	reserved
func_addr[6:0]	6:0	Sets the function address designated by SetAddress.

21-3-3-10 Device time stamp register: TSTAMP(7.4.4.)

[Base + 0x020C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	timestamp 10	timestamp 9	timestamp 8
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	timestamp 7	timestamp 6	timestamp 5	timestamp 4	timestamp 3	timestamp 2	timestamp 1	timestamp 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Name	Bit	Description
reserved	31:11	reserved
timestamp[10:0]	10:0	Indicates a frame number when receiving SOF. * Increments when receiving RecoverySOF. (At HS, it changes 1 time out of 8 receptions. At FS, it changes each time.) * RSOF (RecoverySOF) RSOF is an SOF signal that is generated in the macro when SOF is not transmitted from the host or when the macro cannot receive SOF from the host. RSOF generating counter is enabled by receiving SOF from the host. The counter outputs RSOF signal when SOF is not received for more than the definite period of time (HS standard micro-frame interval value (125.0625 μsec) or FS standard frame interval value (1.000500 msec)). When any of the conditions below is met, RSOF is disabled from being issued. - Bus reset - Suspend state - Resume state

		- Disconnect state - Phy_Erratic_Error state

7.5. OTG-related register

* This macro does not support USB On The Go function.

21-3-3-11 OTG control register: OTGC(7.5.1.)

[Base + 0x0300]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	id_pull_up	dp_pull_down
Reset	0	0	0	0	0	0	0	0*
R/W	-	-	-	-	-	-	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	dm_pull_down	-	-	-	-	-	-	-
Reset	0*	0	0	0	0	0	0	0
R/W	R/W	-	-	-	-	-	-	-

Name	Bit	Description
reserved	31:10	reserved
id_pull_up	9	Sets ID pull-up resistor. This bit value is output to p_idpullup_o pin. 0: Does not pull up. 1: Pulls up.
dp_pull_down	8	Sets DP pull-down resistor. This bit value is output to p_dppulldown_o pin. 0: Does not pull down. 1: Pulls down.
dm_pull_down	7	Sets DM pull-down resistor. This bit value is output to p_dmpulldown_o pin. 0: Does not pull down. 1: Pulls down.
reserved	6:0	reserved

* The bits 8 to 7 above shall be set at adequate values according to USB operation mode and register values. For this product, the setting shall remain: id_pull_up = 0.

21-3-3-12 OTG status register: OTGSTS(7.5.2.)

[Base + 0x0310]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	vbus_vld	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	id	-	-	-	-	-	otg_tmrout
Reset	0	1	0	0	0	0	0	0
R/W	-	R	-	-	-	-	-	R

Name	Bit	Description
reserved	31:11	reserved
vbus_vld	10	Judges VBUS voltage, VBUS_VLD (4.4 V min). 0: VBUS < VA_VBUS_VLD 1: VBUS ≥ VA_VBUS_VLD p_vbusvalid_i pin value becomes this bit value.
reserved	9:7	reserved
id	6	Judges ID pin state. 0: mini-A plug is connected. (Host) 1: Other states than above. (Device) * When OTGC.id_pull_up = 1, p_iddig_i pin value becomes this bit value. When OTGC.id_pull_up = 0, this bit value is “1” independently of p_iddig_i pin value.
reserved	5:1	reserved
otg_tmrout	0	Judges the timeout of OTG timer. 0: No timeout. 1: Timeout. (Only at the moment when the down counter changes: 1 → 0, it becomes “1”.)

21-3-3-13 OTG status change register: OTGSTSC(7.5.3.)

[Base + 0x0314]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	vbus_vld_c	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R/W0	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	Id_c	-	-	-	-	-	otg_tmrout_c
Reset	0	0	0	0	0	0	0	0
R/W	-	R/W0	-	-	-	-	-	R/W0

Name	Bit	Description
reserved	31:11	reserved
vbus_vld_c	10	Indicates VBUS voltage change. ■ Set condition - The value of vbus_vld (Bit[10]) of OTGSTS register changes. - When vbus_vld = 1 in OTGSTS register, and vbus_vld_ren 0→1 is written in OTGSTSRISE register. Writing 1→1 does not set this bit. ■ Clear condition - Write of “0”
reserved	9:7	reserved
id_c	6	Judges ID pin state. ■ Set condition - The value of id (Bit[6]) of OTGSTS register changes. ■ Clear condition - Write of “0”
reserved	5:1	reserved
otg_tmrout_c	0	Indicates the timeout of OTG timer.

		■ Set condition - The value of otg_tmrout (Bit[0]) of OTGSTS register changes. ■ Clear condition - Write of “0”

21-3-3-14 OTG status fall detection enable register: OTGSTSFALL(7.5.4.)

[Base + 0x0318]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	vbus_vld_fen	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R/W	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	Id_fen	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	R/W	-	-	-	-	-	-

Name	Bit	Description
reserved	31:11	reserved
vbus_vld_fen	10	Controls whether to enable the function of setting vbus_vld_c (Bit[10]) of OTGSTSC to “1” when vbus_vld (Bit[10]) value of OTGSTS register changes: 1 → 0. 0: disable 1: enable
reserved	9:7	reserved
id_fen	6	Controls whether to enable the function of setting id_c (Bit[6]) of OTGSTSC register to “1” when id (Bit[6]) value of OTGSTS register changes: 1 → 0. 0: disable 1: enable
reserved	5:0	reserved

21-3-3-15 OTG status rise detection enable register: OTGSTSRISE(7.5.5.)

[Base + 0x031C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	vbus_vld_ren	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R/W	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	Id_ren	-	-	-	-	-	otg_tmrout_ren
Reset	0	0	0	0	0	0	0	0
R/W	-	R/W	-	-	-	-	-	R/W

Name	Bit	Description
reserved	31:11	reserved
vbus_vld_ren	10	Controls whether to enable the function of setting vbus_vld_c (Bit[10]) of OTGSTSC register to “1” when vbus_vld (Bit[10]) value of OTGSTS register changes: 0 → 1. 0: disable 1: enable
reserved	9:7	reserved
id_ren	6	Controls whether to enable the function of setting id_c (Bit[6]) of OTGSTSC register to “1” when id (Bit[6]) of OTGSTS register changes: 0 → 1. 0: disable 1: enable
reserved	5:1	reserved
otg_tmrout_ren	0	Controls whether to enable the function of setting otg_tmrout_c (Bit[0]) of OTGSTSC register to “1” when otg_tmrout (Bit[0]) value of OTGSTS register changes: 0 → 1 0: disable 1: enable

21-3-3-16 OTG timer control register: OTGTC(7.5.6.)

[Base + 0x0320]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	tmr_prescale7	tmr_prescale6	tmr_prescale5	tmr_prescale4	tmr_prescale3	tmr_prescale2	tmr_prescale1	tmr_prescale0
Reset	0 → Value recommended	0 → Value recommended	0 → Value recommended	0 → Value recommended	0 → Value recommended	0 → Value recommended	0 → Value recommended	0 → Value recommended
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	tmr_mode	start_tmr
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R/W	R/W

Name	Bit	Description
reserved	31:16	reserved
tmr_prescale[7:0]	15:8	Sets the frequency dividing value of p_clk_o/4 clock for OTG timer (generates 0.01 ms period). The recommended value below is automatically loaded within 6 cycles of h_clk_i after h_reset_i = 0 → 1. The recommended value (p_clk_o) 60 MHz: 149 (95h)
reserved	7:2	reserved
tmr_mode	1	Selects the timer operation mode. 0: One-shot timer (When it times out, otg_tmrout = 1 of OTGSC register is set.) 1: Interval timer (operates repeatedly)
start_tmr	0	Controls the timer (down counter) operation. 0: Timer stops. 1: Timer starts. * When it stops and restarts, it begins at the initial value set by tmr_init_val[15:0] of OTGT register.

21-3-3-17 OTG timer register: OTGT(7.5.7.)

[Base + 0x0324]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	tmr_init_val15	tmr_init_val14	tmr_init_val13	tmr_init_val12	tmr_init_val11	tmr_init_val10	tmr_init_val9	tmr_init_val8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	tmr_init_val7	tmr_init_val6	tmr_init_val5	tmr_init_val4	tmr_init_val3	tmr_init_val2	tmr_init_val1	tmr_init_val0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Name	Bit	Description
reserved	31:16	reserved
tmr_init_val[15:0]	15:0	Sets the initial value of OTG timer (the down counter initial value). Timer precision: 0.01 ms The maximum value: 655 ms * Set the value at “Set value − 1”.

7.6. DMA-related register

21-3-3-18 DMA Ch1, Ch2 control register: DMAC1, DMAC2(7.6.1.)

DMAC1 [Base + 0x0400]

DMAC2 [Base + 0x0420]

(* = 1 or 2)

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	dma*_blksize10	dma*_blksize9	dma*_blksize8
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	dma*_blksize7	dma*_blksize6	dma*_blksize5	dma*_blksize4	dma*_blksize3	dma*_blksize2	dma*_blksize1	dma*_blksize0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	dma*_ep3	dma*_ep2	dma*_ep1	dma*_ep0
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	dma*_spr	dma*_int_empty	dma*_sendnull	dma*_mode	dma*_busy	dma*_st
Reset	0	0	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R	R/W

Limitation:

Bus reset in the device mode initializes the values.

* port_reset in the host mode does not reset the values.

Name	Bit	Description
reserved	31:27	reserved
dma*_blksize[10:0]	26:16	Sets the block byte size (the number of bytes that DREQ transfers one time). * Set the number of bytes that DREQ transfers one time for the demand transfer, too.
reserved	15:12	reserved
dma*_ep[3:0]	11:8	Sets an endpoint for DMA transfer. * When dev_en = 1 of MODE register (device is operating), EP0 is

		not controlled by DMA.
reserved	7:6	reserved
dma*_spr	5	Controls whether DMA transfers a received short packet or DMA does not transfer it and completes the transfer. IN transfer is for host_en = 1 of MODE register (Host operating). OUT transfer is for dev_en = 1 of MODE register (Device operating). 0: DMA transfers a received short packet. 1: DMA does not transfer a received short packet and completes the transfer. * NULL packet reception A received NULL packet is discarded and dma_np (Bit[0]) = 1 of DMAS register is set. Additionally, at device, appptr (Bit[1:0]) of EPCTRL register is incremented. At host, appptr (Bit[1:0]) of HCEPCTRL1 register is incremented.
dma*_int_empty	4	Controls whether to complete DAM transfer after the buffer becomes empty. OUT transfer is for host_en = 1 of MODE register (Host operating). IN transfer is for dev_en = 1 of MODE register (Device operating). 0: DMA transfer is completed when DMA-to-USB20DC buffer transfer is completed. 1: DMA transfer is completed when the buffer transferred from DMA to USB20DC becomes empty (transferred to USB).
dma*_sendnull	3	Controls whether to transmit NULL packet next when the transfer end is the max packet size. OUT transfer is for host_en = 1 of MODE register (Host operating). IN transfer is for dev_en = 1 of MODE register (Device operating). 0: Does not transmit. 1: Transmits.
dma*_mode	2	Sets the DMA transfer mode.

		0: Demand transfer (d_req*_o pin outputs “1” until “1” is input to d_ack*_i pin.) 1: Block transfer (d_req*_o pin outputs “1” for 1 clock only.)
dma*_busy	1	Indicates that DMA is transferring. It is set when D_REQ is issued and cleared when the DMA transfer is completed.
dma*_st	0	Sets whether to start DMA transfer. 0: Does not start. 1: Writing “1” starts successive DMA sequences. Clear condition - Write of “0” - DMA transfer disabled - When DMA transfer completion condition is met. * Attention (when DMA is in operation): - Do not operate bufrd, bufwd and init of EPCMD* register. However, only the endpoints where DMA is operating shall not be operated. Other endpoints can be operated. - Do not use ready_int (Bit[26]) of EPCTRL#1-15 register (device) for DMA transfer when the device is operating. So, disable ready_inten (Bit[18]) at the time of starting DMA transfer. * DMA transfer disabled - When host_en = 0 and dev_en = 0 of MODE register (0004h). - When dev_en = 1 of MODE register (0004h) and the endpoint designated by dma*_ep of DMAC register (0400h/0420h) is not Valid (*1). (*1) Condition of being not Valid - When the designated endpoint is EP0. - When ep_en = 0 of EPCTRL1-15 register. - When host_en = 1 of MODE register (0004h) and the endpoint designated by dma*_ep of DMAC register (0400h/0420h) is not Valid (*1). (*1) Condition of being not Valid - When trans_en = 0 of HCEPCTRL1 register. * Attention (when DMA transfer is completed): Compare the number of transfer bytes set by DMATCI* register with

		the number of transferred bytes set by DMATC* register. If they are mismatched, that means an error occurred during DMA transfer. When DMA transfer is completed (at host IN transfer or at device OUT transfer), check empty (Bit[10]) of HCEPCTRL0 (Host) or EPCTRL (Device) register for the existence of data. When empty (Bit[10]) = 0, a new data is stored. Read it out (DMA/CPU) by software. * EPCMD stop When setting EPCMD.stop clears EPCTRL.ep_en, DMAC.dma_st that is assigned to the same endpoint as that of EPCMD.stop is also cleared. When EPCMD.stop is performed, clear the pointer by EPCMD.init at the time of (or before) setting next EPCMD.start. * Forced termination Software forced termination (dma_st = 1 → 0) de-asserts d_req*_o signal.

21-3-3-19 DMA Ch1, Ch2 status register: DMAS1, DMAS2(7.6.2.)

DMAS1 [Base + 0x0404]

DMAS2 [Base + 0x0424]

(* = 1 or 2)

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	dma*_sp	dma*_np
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R	R

Name	Bit	Description	
reserved	31:2	reserved	
dma*_sp	1	Indicates a short packet during successive transfers. At buffer write	
		0	When max-packet-size data is transferred for next transmission request.
		1	When data smaller than the max-packet size is transferred for next transmission request (including NULL packet transmission).
		At buffer read	
		0	When received data is the max-packet size.
		1	When received data is smaller than the max-packet size (including NULL packet reception).
		Writing “1” (0 →1) clears dma*_st of DMAC* register.	
dma*_np	0	Indicates a NULL packet during successive transfers. At buffer write	

		0	When the transfer request is not NULL packet.
		1	When the transfer request is NULL packet after DMA transfer is completed in the max-packet size and the last packet data written by DMA is transmitted.
		At buffer read	
		0	When received data is not NULL packet.
		1	When received data is NULL packet.
		Writing “1” (0 →1) clears dma*_st of DMAC* register.	

21-3-3-20 DMA Ch1,Ch2 successive transfer bytes setting register: DMATCI1, DMATCI2(7.6.3.)

DMATCI1 [Base + 0x0408]

DMATCI2 [Base + 0x0428]

(* = 1 or 2)

Bit	31	30	29	28	27	26	25	24
Name	dma*_tci31	dma*_tci30	dma*_tci29	dma*_tci28	dma*_tci27	dma*_tci26	dma*_tci25	dma*_tci24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	dma*_tci23	dma*_tci22	dma*_tci21	dma*_tci20	dma*_tci19	dma*_tci18	dma*_tci17	dma*_tci16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	dma*_tci15	dma*_tci14	dma*_tci13	dma*_tci12	dma*_tci11	dma*_tci10	dma*_tci9	dma*_tci8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	dma*_tci7	dma*_tci6	dma*_tci5	dma*_tci4	dma*_tci3	dma*_tci2	dma*_tci1	dma*_tci0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Limitation:

Bus reset initializes it.

* port_reset in the host mode does not reset it.

Name	Bit	Description
dma*_tci[31:0]	31:0	Sets the successive DMA transfer counter (initial value). * Attention (at DMA read): Before setting 1-packet DMA request, (DMATCI.dma_tci − DMATC.dma_tc) is compared with PHY buffer pointer (HCEPCTRL.hcptr[Host]/EPCTRL.phyptr[Device]). Then, when it is judged that all data in 1 buffer cannot be read out ((dma_tci − dma_tc) < (phyptr)), DMA request is not set and dma_st is cleared at the end. The buffer is not discarded. So, setting an adequate value in dma_tci again and restarting DMA enables DMA to start reading the packet

		data that is not read out and is completed. When the max-packet size is alternate-changed, it is necessary to read all the data in buffer or to initialize buffer before change. Short packet criteria are shown below. - When a packet whose DMA transfer is not performed is smaller than the max-packet size, dma_sp = 1. - When a packet whose DMA transfer is not performed is the max-packet size, dma_sp = 0.

21-3-3-21 DMA Ch1, Ch2 successive transfer bytes counter register: DMATC1, DMATC2(7.6.4.)

DMATC1 [Base + 0x040C]

DMATC2 [Base + 0x042C]

(* = 1 or 2)

Bit	31	30	29	28	27	26	25	24
Name	dma*_tc31	dma*_tc30	dma*_tc29	dma*_tc28	dma*_tc27	dma*_tc26	dma*_tc25	dma*_tc24
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	dma*_tc23	dma*_tc22	dma*_tc21	dma*_tc20	dma*_tc19	dma*_tc18	dma*_tc17	dma*_tc16
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	dma*_tc15	dma*_tc14	dma*_tc13	dma*_tc12	dma*_tc11	dma*_tc10	dma*_tc9	dma*_tc8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	dma*_tc7	dma*_tc6	dma*_tc5	dma*_tc4	dma*_tc3	dma*_tc2	dma*_tc1	dma*_tc0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Name	Bit	Description
dma*_tc[31:0]	31:0	Indicates the number of bytes that have been transferred by successive DMA. Clear condition: Write of “1” (0 →1) in DMAC*.dma*_st * It is updated when 1-packet data transfer is completed on AHB bus.

7.7. Test-related registers

21-3-3-22 Test control register: TESTC(7.7.1.)

TESTC [Base + 0x0500]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	testforceh s	custum_te st
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	testforece enble	testse0na ck	testk	testj	testp
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

Name	Bit	Description
reserved	31:26	reserved
testforcehs	25	A mode where PORT_RESET at host establishes HS connection without chirp sequence.
custom_test	24	DEVC.reqspeed value is directly used as PHY connection speed.
reserved	23:5	reserved
testforeceenable	4	TESET FORCE ENABLE (Exclusive to the host)
testse0nack	3	SE0NAK response test (Exclusive to the device)
testk	2	K State (common to Host/Device)
testj	1	J State (common to Host/Device)
testp	0	Test packet transmission (common to Host/Device)

7.8. Device RAM register

* In the following part of this document, the registers are in RAM configuration.

Be sure to write “0” in the reserve bits (named “-”).

21-3-3-23 EP0 control register: EPCTRL0(7.8.1.)

EPCTRL0 [Base + 0x8000]

Bit	31	30	29	28	27	26	25	24
Name	-	nack_int	stalled_int	ping_int	readyo_int	readyi_int	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	R	R	R	R	R	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	nack_inten	stalled_inten	ping_inten	readyo_inten	readyi_inten	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	R	R	R	R	R	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	toggle	stall	fullo	emptyo	fulli	emptyi
Reset	0	0	0	0	0	1	0	1
R/W	-	-	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	ep_en
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	R

Limitation:

- Be sure to initialize it with reset values before setting dev_en = 1 of MODE register.
- Do not change this register directly by writing after dev_en = 1 is set.
- Use EPCMD0 register to write and change the register setting.
- Read it several times until read-data is stable.

Name	Bit	Description
reserved	31	reserved
nack_int	30	Indicates NAK packet transmission interrupt. ■ Set condition - Transmitting NAK packet ■ Clear condition - Executing command: nack_int_clr (Bit[22]) of EPCMD0 register
stalled_int	29	Indicates STALL packet transmission interrupt. ■ Set condition - Responding with STALL ■ Clear condition

		- Executing command: stalled_int_clr (Bit[21]) of EPCMD0 register
ping_int	28	Indicates PING packet reception interrupt. ■ Set condition - Receiving PING packet ■ Clear condition - Executing command: ping_int_clr (Bit[20]) of EPCMD0 register
readyo_int	27	Indicates OUT transfer buffer ready interrupt. ■ Set condition [Setup stage] - When SETUP is received from the host and ACK is transmitted (independently of the existence of OUT transfer buffer data). [Data/Status stage] - When fullo bit changes: 0 → 1 because of data reception from the host. ■ Clear condition - Executing command: readyo_int_clr (Bit[19]) of EPCMD0 register
readyi_int	26	Indicates IN transfer buffer ready interrupt. ■ Set condition [Setup stage] - When SETUP is received from the host and ACK is transmitted (when data remains IN transfer buffer). [Data/Status stage] - When emptyi bit changes: 0 → 1 because of transmission to the host. ■ Clear condition - Executing command: readyi_int_clr (Bit[18]) of EPCMD0 register
reserved	25:23	reserved
nack_inten	22	Controls whether to enable interrupt by nack_int (Bit[30]). 0: disable 1: enable
stalled_inten	21	Controls whether to enable interrupt by stalled_int (Bit[29]). 0: disable 1: enable

ping_inten	20	Controls whether to enable interrupt by ping_int (Bit[28]). 0: disable 1: enable
readyo_inten	19	Controls whether to enable interrupt by readyo_int (Bit[27]). 0 : disable 1 : enable
readyi_inten	18	Controls whether to enable interrupt by Ireadyo_int (Bit[26]). 0: disable 1: enable
reserved	17:14	reserved
toggle	13	Indicates a data toggle that is transmitted or received next. 0: DATA0 1: DATA1 ■ Set condition - Executing command: toggle_set (Bit[7]) of EPCMD0 register - toggle = 0, IN/OUT transfer (other than ISO/PING), and ACK is transmitted or received. - Transmitting ACK by setup transfer ■ Clear condition - Executing command: toggle_clr (Bit[8]) of EPCMD0 register - toggle = 1, IN/OUT transfer (other than ISO/PING), and ACK is transmitted or received. * USB20DC hardware automatically judges SETUP/STATUS stage of SETUP. So, toggle bit value is ignored.
stall	12	STALL status flag 1: Indicates that the endpoint is in the stall state. ■ Set condition - Executing command: stall_set (Bit[5]) of EPCMD0 register ■ Clear condition

		- Executing command: stall_clr (Bit[6]) of EPCMD0 register - When an Valid SETUP token is received.
fullo	11	Buffer full (OUT) flag 1: Indicates that OUT transfer buffer has data received from the host. ■ Set condition - When data is normally received from the host and ACK is transmitted. ■ Clear condition - Executing command: bufrd (Bit[4]) of EPCMD0 register - Executing command: init (Bit[2]) of EPCMD0 register - Receiving SETUP from the host
emptyo	10	Buffer empty (OUT) flag 1: Indicates that OUT transfer buffer is empty. ■ Set condition - Executing command: bufrd (Bit[4]) of EPCMD0 register - Executing command: init (Bit[2]) of EPCMD0 register - Receiving SETUP from the host. ■ Clear condition - When data is normally received from the host and ACK is transmitted.
fulli	9	Buffer full (IN) flag 1: Indicates that IN transfer buffer is not transferred. ■ Set condition - Executing command: bufwr (Bit[3]) of EPCMD0 register ■ Clear condition - When transmitted to the host and ACK is received. - Executing command: init (Bit[2]) of EPCMD0 register - When SETUP is received from the host and ACK is transmitted.
emptyi	8	Buffer empty (IN) flag

		1: Indicates IN transfer buffer is empty. ■ Set condition - When transmitted to the host and ACK is received. - Executing command: init (Bit[2]) of EPCMD0 register - When SETUP is received from the host and ACK is transmitted. ■ Clear condition - Executing command: bufwr (Bit[3]) of EPCMD0 register
reserved	7:1	reserved
ep_en	0	Endpoint enable 0: Does not respond to transaction from the host. 1: Responds when the function address matches the endpoint number. ■ Set condition - Executing command: start (Bit[0]) of EPCMD0 register ■ Clear condition - Executing command: stop (Bit[1]) of EPCMD0 register * In the case of firmware forced stop, wait for 2 μFrame or more before setting each item. * Software stop is not recommended, except to finish the device operation.

21-3-3-24 EP (1–15) control register: EPCTRL (1–15) (7.8.2.)

EPCTRL1~15 [Base + 0x8004 ~ 0x803C]

Bit	31	30	29	28	27	26	25	24
Name	-	nack_int	stalled_int	ping_int	empty_int	ready_int	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	R	R	R	R	R	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	nack_inten	stalled_inten	ping_inten	empty_inten	ready_inten	nackresp	nullresp
Reset	0	0	0	0	0	0	0	0
R/W	-	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	hiband1	hiband0	toggle	stall	full	empty	phyptr1	phyptr0
Reset	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	appptr1	appptr0	bnum1	bnum0	dir	et1	et0	ep_en
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Limitation

- Be sure to initialize it with reset values before setting dev_en = 1 of MODE register.
- Do not change this register directly by writing after dev_en = 1 is set.
- Use EPCMD (1–15) register to write and change the register setting.
- Read it several times until read-data is stable.

Name	Bit	Description
reserved	31	reserved
nack_int	30	Indicates NAK packet transmission interrupt. ■ Set condition - Transmitting NAK packet ■ Clear condition - Executing command: nack_int_clr (Bit[22]) of EPCMD#1–15
stalled_int	29	Indicates STALL packet transmission interrupt. ■ Set condition - Responding with STALL ■ Clear condition - Executing command: stalled_int_clr (Bit[21]) of EPCMD#1-15

		register
ping_int	28	Indicates PING packet reception interrupt. ■ Set condition - Receiving PING packet ■ Clear condition - Executing command: ping_int_clr (Bit[20]) of EPCMD#1–15 register
empty_int	27	Indicates the transfer buffer empty interrupt. ■ Set condition - When empty bit changes: 0 → 1 because of IN transfer transmission to the host. - When empty bit changes: 0 → 1 because OUT transfer reads out data received from the host (when dev_int_mode = 1 of MODE register). - Setting init (Bit[2]) of EPCMD#1–15 register (when dev_int_mode = 1 of MODE register) ■ Clear condition - Executing command: empty_int_clr (Bit[19]) of EPCMD#1–15 register
ready_int	26	Indicates the transfer buffer ready interrupt. ■ Set condition - When full bit changes: 1 → 0 because of IN transfer transmission to the host. - When empty bit changes: 1 → 0 because of OUT transfer reception form the host. - When dev_int_mode = 1 of MODE register, bufrd/bufwd of EPCMD register is set, and the buffer can be read and written. - Setting init (Bit[2]) of EPCMD register (when IN transfer and dev_int_mode = 1 of MODE register) ■ Clear condition - Executing command: ready_int_clr (Bit[18]) of EPCMD#1–15 register * Attention Do not use it during DMA transfer. Disable ready_inten (Bit[18]) at DMA transfer start.

reserved	25:23	reserved
nack_inten	22	Controls whether to enable interrupt by nack_int (Bit[30]). 0: disable 1: enable
stalled_inten	21	Controls whether to enable interrupt by stalled_int (Bit[29]). 0: disable 1: enable
ping_inten	20	Controls whether to enable interrupt by ping_int (Bit[28]). 0: disable 1: enable
empty_inten	19	Controls whether to enable interrupt by empty_int (Bit[27]). 0: disable 1: enable
ready_inten	18	Controls whether to enable interrupt by ready_int (Bit[26]). 0: disable 1: enable
nackresp	17	Sets NAK response to bulk transfer IN/OUT/PING. 0: Does not respond with NAK. 1: Responds with NAK. * Attention It responds with ACK at a toggle error even when “1” is set.
nullresp	16	Sets NULL response to bulk/interrupt IN transfer. 0: Does not respond with NULL. 1: Responds with NULL.
hiband[1:0]	15:14	Sets the number of packets that are transmitted and received for 1 µFrame. 00: 1 01: 1 10: 2 11: 3

toggle	13	Indicates a data toggle that is transmitted or received next. 0: DATA0 1: DATA1 ■ Set condition - Executing command: toggle_set (Bit[7]) of EPCMD#1-15 register - toggle = 0, IN/OUT transfer (other than ISO/PING), and ACK is transmitted or received. ■ Clear condition - Executing command: toggle_clr (Bit[8]) of EPCMD#1-15 register - toggle = 1, IN/OUT transfer (other than ISO/PING), and ACK is transmitted or received.
stall	12	STALL state flag 1: Indicates that the endpoint is in the stall state. ■ Set condition - Executing command: stall_set (Bit[5]) of EPCMD#1–15 register ■ Clear condition - Executing command: stall_clr (Bit[6]) of EPCMD#1–15 register
full	11	Full flag 1: Indicates that a buffer to be transmitted to the host is full at IN transfer and that a buffer received from the host is full at OUT transfer. ■ Set condition - When, at IN transfer, appptr updated by executing command, bufwr (Bit[3]) of EPCMD#1–15 register is equal to phyptr. - When, at OUT transfer, phyptr updated by receiving data from the host is equal to appptr. ■ Clear condition - Executing command: init (Bit[2]) of EPCMD#1–15 register. - IN transfer data transmission to the host - At OUT transfer, executing command: bufrd (Bit[4]) of EPCMD#1–15 register

empty	10	Empty flag 1: Indicates that there is no buffer to be transmitted to the host at IN transfer and that there is no buffer received from the host at OUT transfer. ■ Set condition - Executing command: init (Bit[2]) of EPCMD#1–15 register - When, at IN transfer, phyptr updated by transmitting data to the host is equal to appptr. - When, OUT transfer, appptr updated by executing command, bufrd (Bit[4]) of EPCMD#1–15 register is equal to phyptr. ■ Clear condition - At IN transfer, executing command: bufwr (Bit[3]) of EPCMD#1-15 register - Receiving data from the host at OUT transfer
phyptr[1:0]	9:8	PHY access buffer pointer Indicates a buffer number to be read or written next when data is transmitted/received from the host. ■ Set condition - Increments when data is normally received and transmitted and ACK is transmitted or received. When the number of buffers (bnum) is exceeded, it returns to “00”. ■ Clear condition - Executing command: init (Bit[2]) of EPCMD#1–15 register clears it to be “00”.
appptr[1:0]	7:6	APP access buffer pointer Indicates a buffer number that APP (CPU/DMA) reads or writes next. ■ Set condition - Increments by bufwr (Bit[3]) or bufrd (Bit[4]) of EPCMD#1–15 register. When the number of buffers (bnum) is exceeded, it returns to “00”. ■ Clear condition - Executing command: init (Bit[2]) of EPCMD#1–15 register clears it to be “00”.
bnum[1:0]	5:4	Sets the number of buffers. 00: Single

		01: Double 10: reserved 11: reserved
dir	3	Sets the transfer direction. 0: OUT 1: IN
et[1:0]	2:1	Sets the transfer type. 00: reserved 01: Isochronous 10: Bulk 11: Interrupt * Control transfer cannot be performed for any other endpoint than EP0.
ep_en	0	Endpoint enable 0: Does not respond to transaction from the host. 1: Responds when the function address matches the endpoint number. ■ Set condition - Executing command: start (Bit[0]) of EPCMD#1–15 register ■ Clear condition - Executing command: stop (Bit[1]) of EPCMD#1–15 register * When ep_en = 0 is forcibly set, wait for 2 μFrame or more at HS and wait for 2 Frame or more at FS/LS before setting each item.

21-3-3-25 EP (0–15) configuration register**:**EPCONF**(**0–15**)**(7.8.3.)

EPCONF1~15 [Base + 0x8040 ~ 0x807C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	countidx4	countidx3	countidx2	countidx1	countidx0
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	size10	size9	size8	size7	size6	size5	size4	size3
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	size2	size1	size0	base12	base11	base10	base9	base8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	base7	base6	base5	base4	base3	base2	base1	base0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Limitation:

Be sure to initialize it with reset values before setting dev_en = 1 of MODE register.

Name	Bit	Description
reserved	31:29	reserved
countidx[4:0]	28:24	Sets the counter register index. * Set a number 0 to 31, not a physical address of EP counter register. See “Figure 6-1: Example of the endpoint buffer structure”.
size[10:0]	23:13	Sets the buffer size. * Set 1 to 1024 according to the transfer mode. Set the buffer size (max packet size) in conformity with USB standard below. FS: CTRL = 8/16/32/64 BYTE BULK = 8/16/32/64 BYTE INTR = 0-64 BYTE ISO = 0-1023 BYTE

		HS: CTRL = 64 BYTE (Fixed) BULK = 512 BYTE (Fixed) INTR = 0-1024 BYTE ISO = 0-1024 BYTE * Receiving a packet larger than the set size causes timeout. * To change the max packet size, it is necessary that all the data in buffer is read or buffer is initialized before change.
base[12:0]	12:0	Sets the buffer head address (word address boundary). * Set the value of a real address whose lower 2 bits and upper 1 bit (the most significant bit) are omitted. [Setting examples] A buffer head address, 0x8180 is set as base[12:0] = 0x0060. A buffer head address, 0xFF00 is set as base[12:0] = 0x1FC0.

21-3-3-26 EP (0–31) counter register: EPCOUNT**(**0–31**)**(7.8.4.)

EPCOUNT1~31 [Base + 0x8080 ~ 0x80FC]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	phycnt10	phycnt9	phycnt8
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	phycnt7	phycnt6	phycnt5	phycnt4	phycnt3	phycnt2	phycnt1	phycnt0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	appcnt10	appcnt9	appcnt8
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	appcnt7	appcnt6	appcnt5	appcnt4	appcnt3	appcnt2	appcnt1	appcnt0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Limitation:

Be sure to initialize it with reset value before setting dev_en = 1 of MODE register.

Name	Bit	Description
reserved	31:27	reserved
phycnt[10:0]	26:16	PHY counter The number of bytes (already received) ■ Update condition After transmitting a normal handshake ■ Clear condition Starting to receive packets clears it to be “0” when OUT transfer && (buffer ! = full).
reserved	15:11	reserved
appcnt[10:0]	10:0	APP counter The number of bytes (already written or read out) ■ Update condition Writing by application ■ Clear condition There is no automatic clear function.

7.9. Host RAM register

(Omitted because this product cannot use the host function.)

21-4 Operation

This section provides USB20DC setting flow.

21-4-1 Precautions for use

This product cannot use the host function of this macro.

Use another built-in host macro compliant with EHCI.

8. Recommended operation flow

8.1. Host/Device judgement

8.1.1. ID/VBUS setting

8.1.2. A-PLUG connection

8.1.3. A-PLUG disconnection

(The flows above are omitted because the host function cannot be used.)

8.1.4. VBUS connection

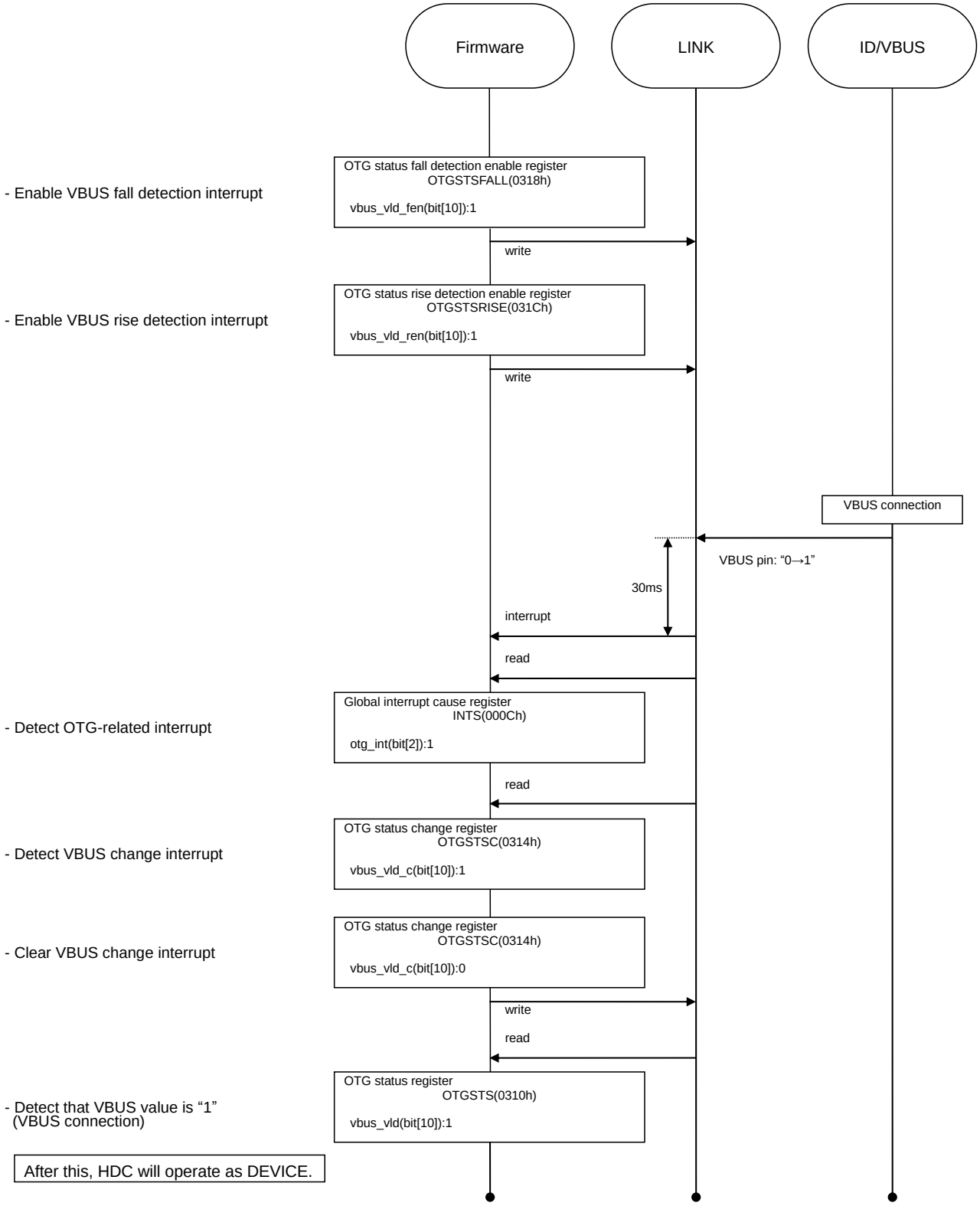


Figure 21-20

8.1.5. VBUS disconnection

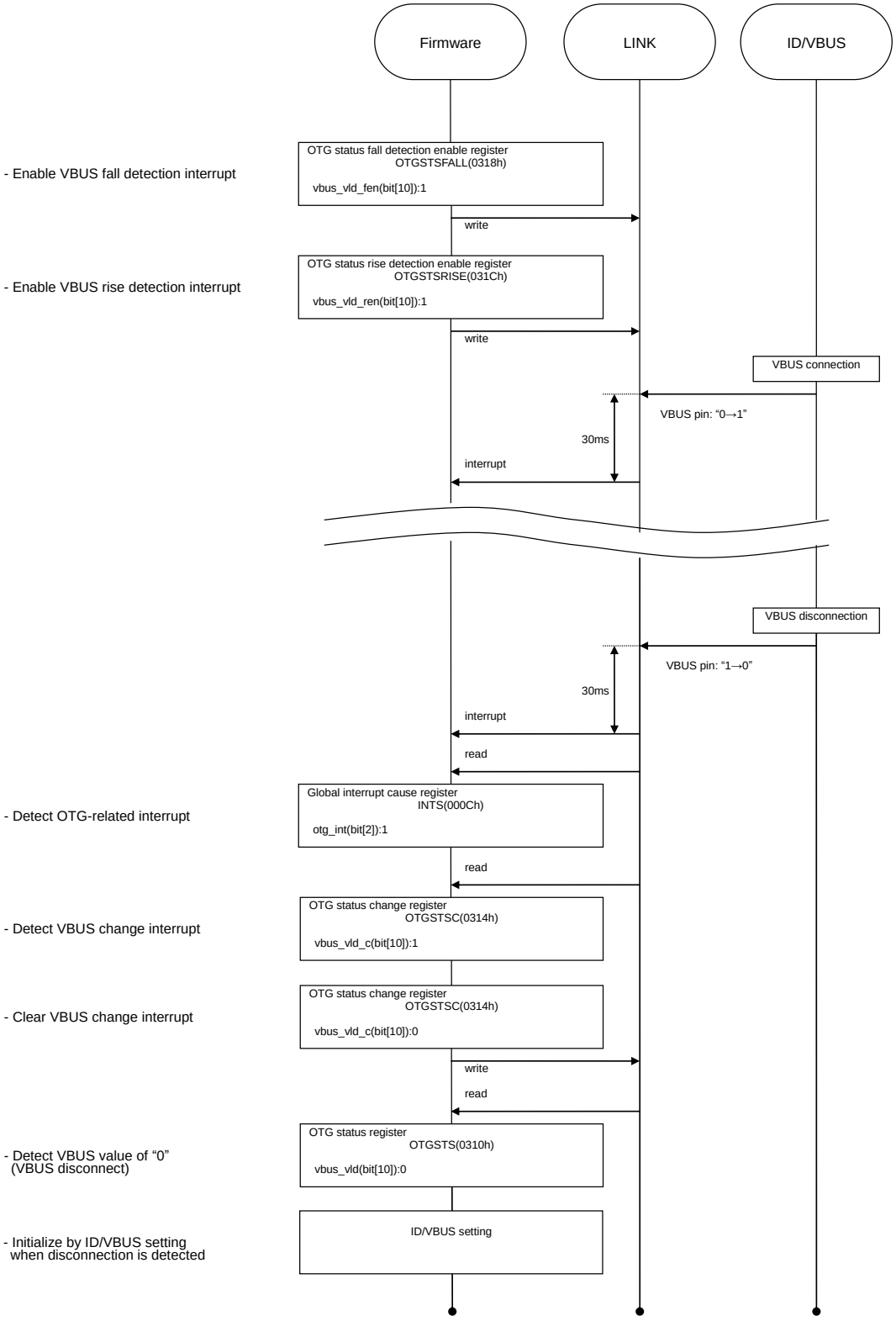


Figure 21-21

8.2. Host

Omitted because this product cannot use the host function.

8.3. Device

This chapter provides the recommended operation flow in the device mode.

8.3.1. Flow

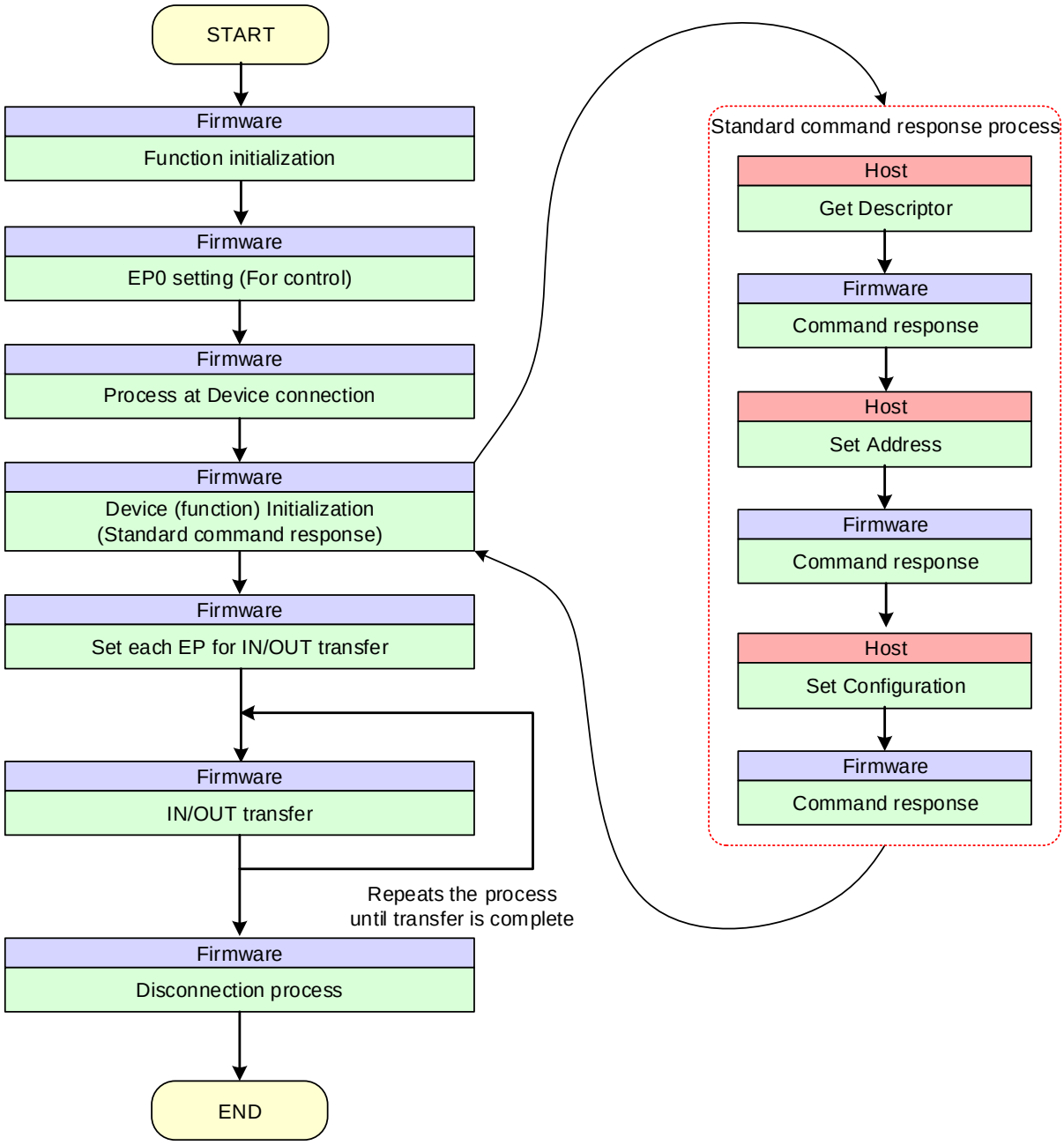


Figure 21-22

8.3.2. Function initial setting

• Soft reset setting

• Check the soft reset cancellation
*Continue to read until soft_reset = 0

• Set burst mode (0 [0Wait] or 1 [1Wait])
• Set byte order (0 [Little Endian] or 1 [Big Endian])

• Initialize RAM area.

*8000h to 80FC (max) is made of RAM. According to the making, initialize all RAM area.

• Set interrupt mode
• Set the operation mode to device

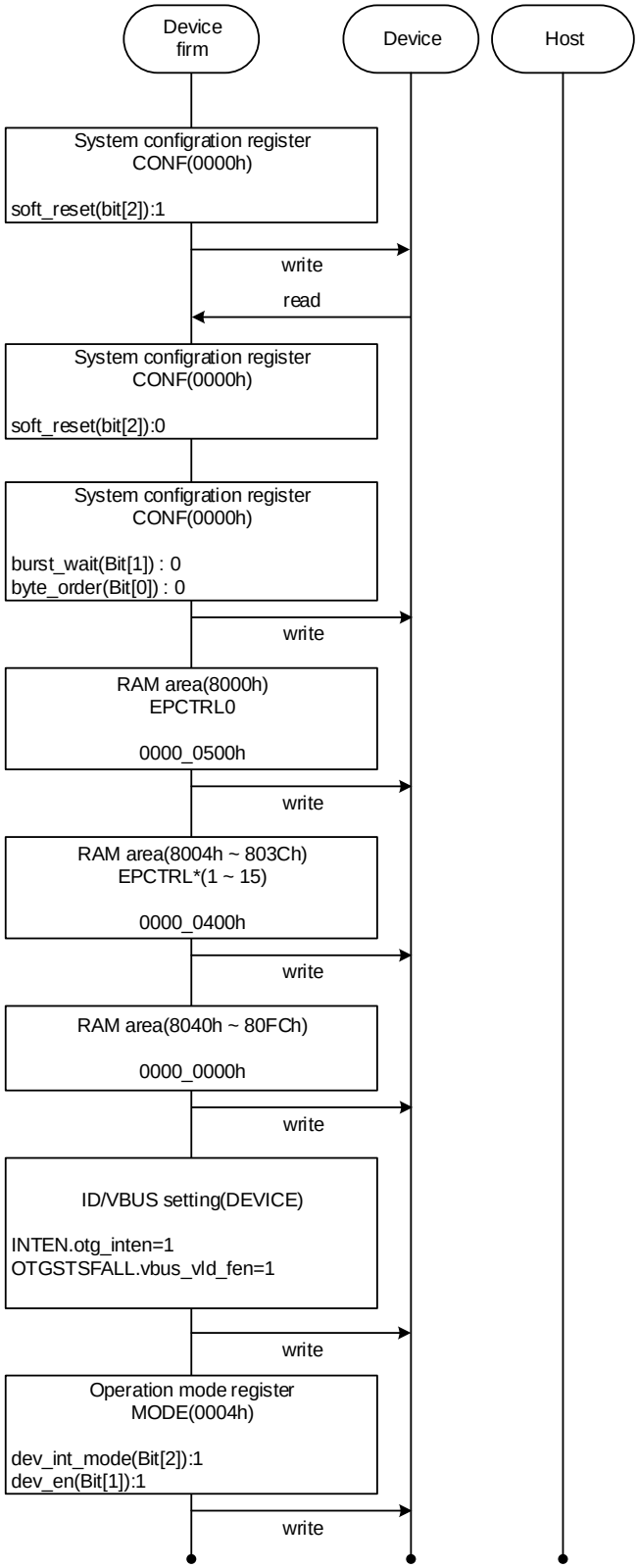


Figure 21-23

8.3.3. EP0 setting

• Soft reset setting

• Check the soft reset cancellation
*Continue to read until soft_reset = 0

• Set burst mode (0 [0Wait] or 1 [1Wait])
• Set byte order (0 [Little Endian] or 1 [Big Endian])

• Initialize RAM area.

*8000h to 80FC (max) is made of RAM. According to the making, initialize all RAM area.

• Set interrupt mode
• Set the operation mode to device

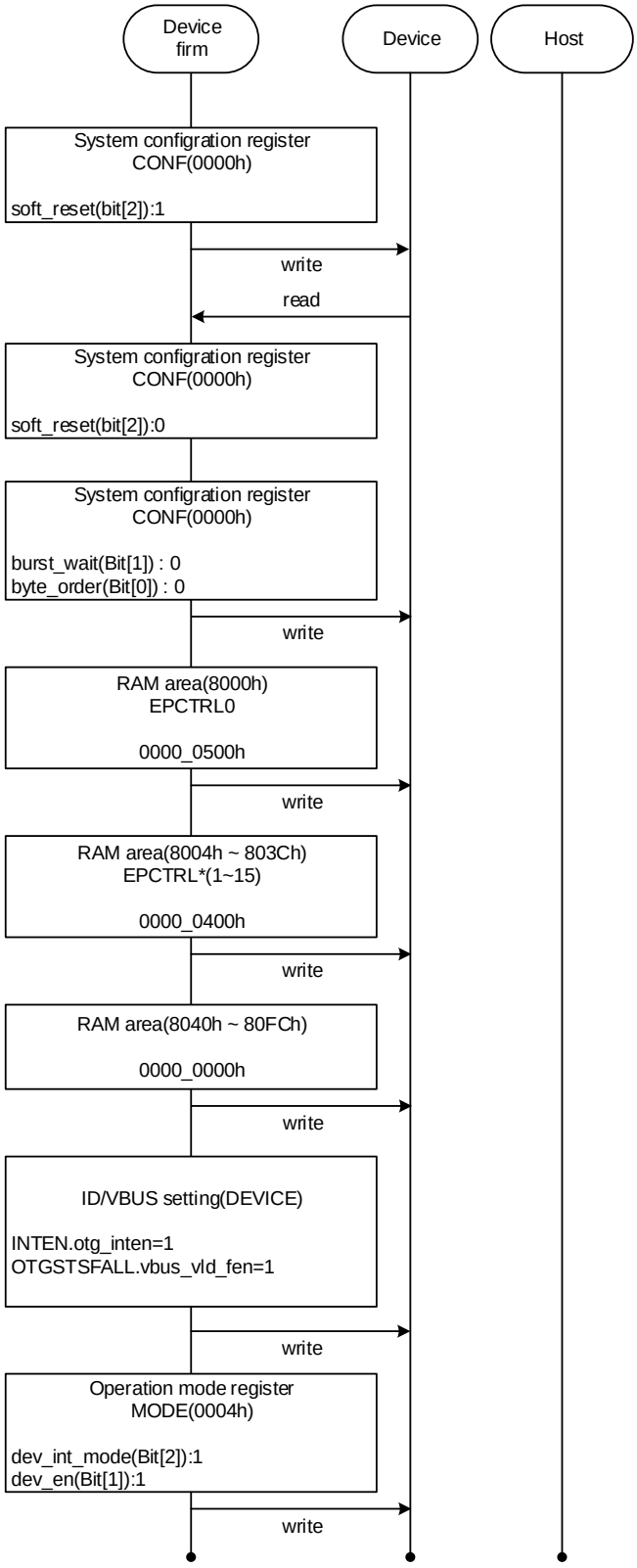


Figure 21-24

8.3.4. Device connection setting

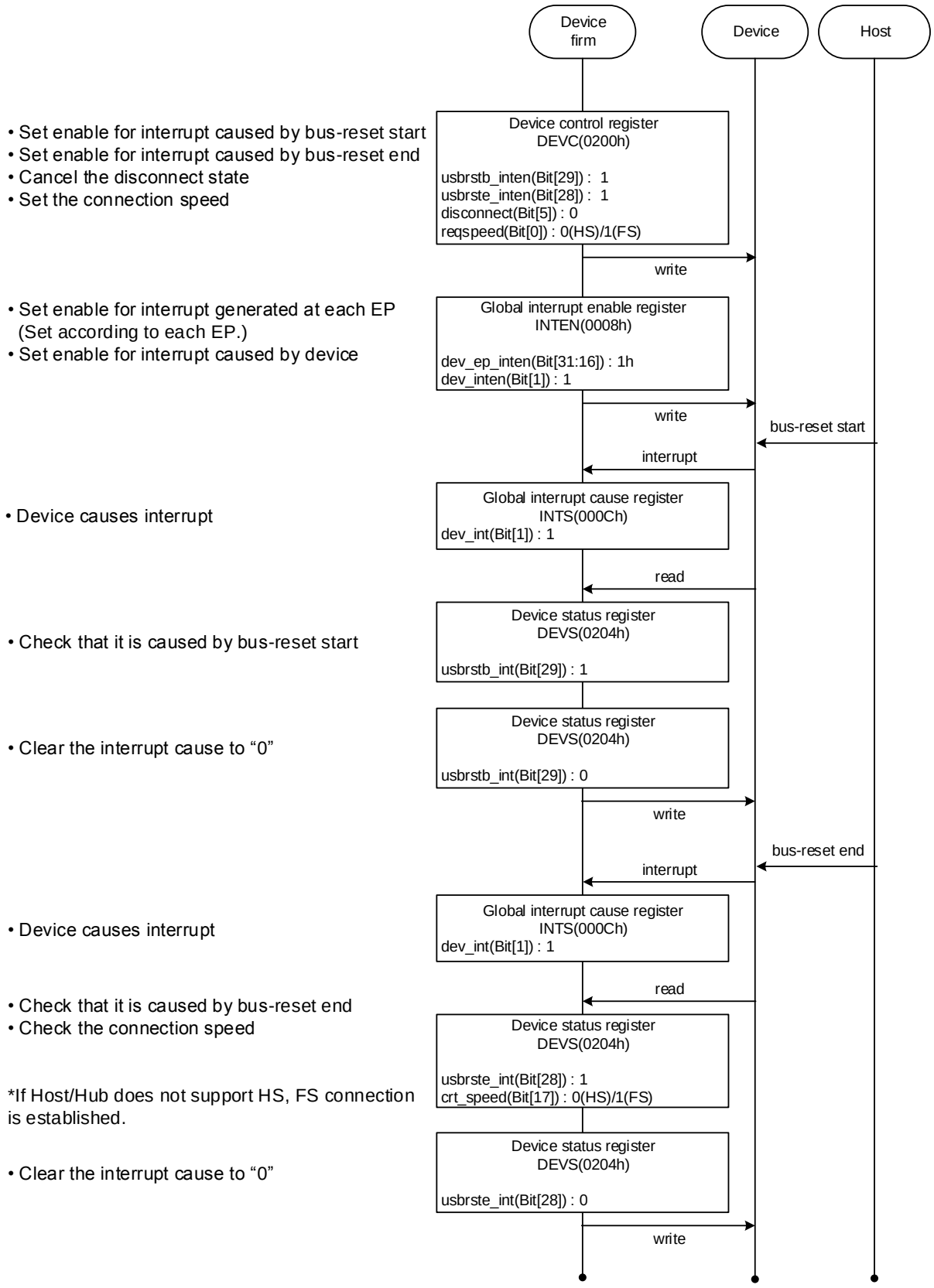


Figure 21-25

8.3.5. Control transfer

Control transfer setup stage

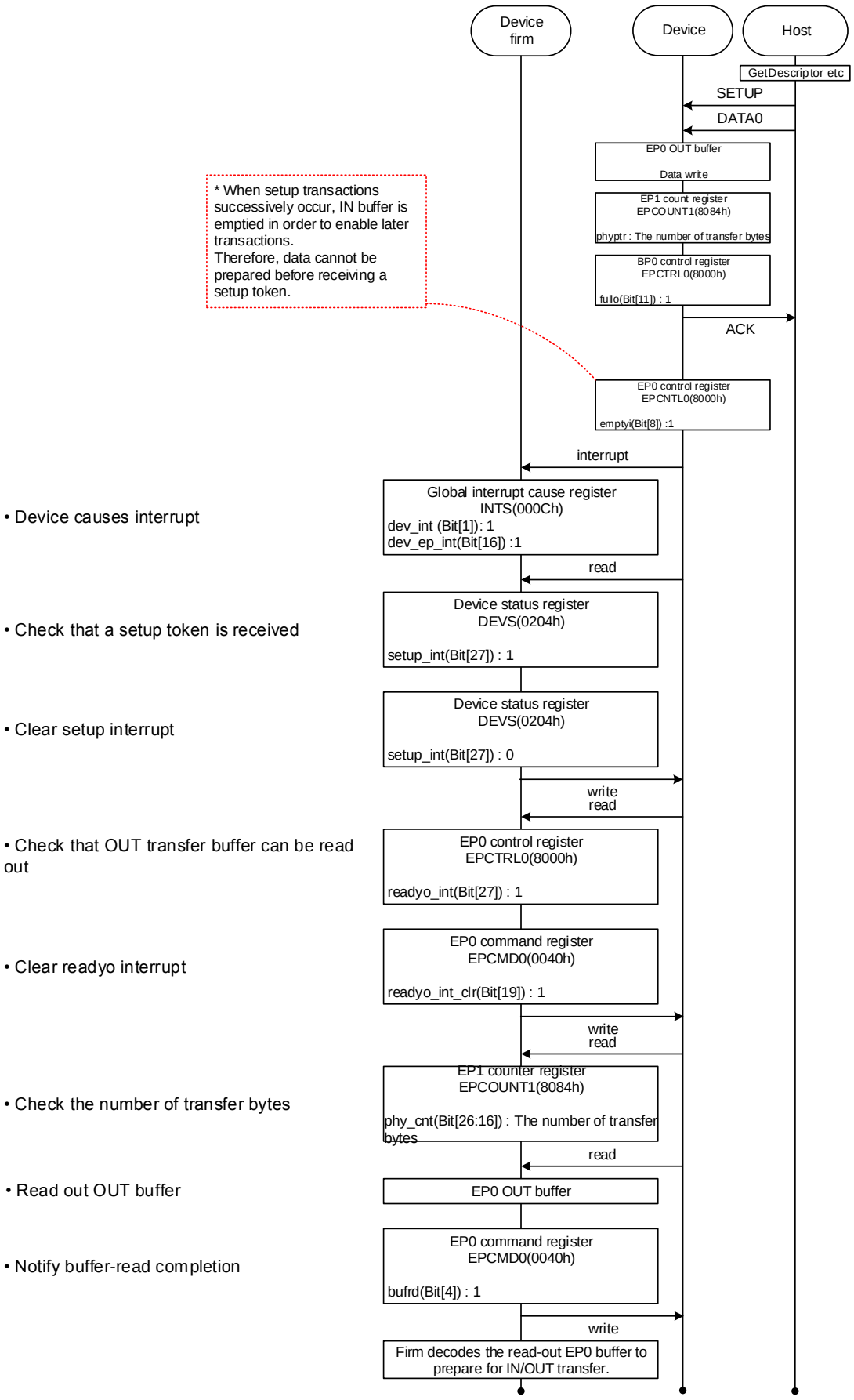

Device firm
Device
Host
GetDescriptor etc
SETUP
DATA0
EP0 OUT buffer
Data write
EP1 count register
EPCOUNT1(8084h)
phyptr : The number of transfer bytes
BP0 control register
EPCTRL0(8000h)
fullo(Bit[11]) : 1
ACK
* When setup transactions successively occur, IN buffer is emptied in order to enable later transactions.
Therefore, data cannot be prepared before receiving a setup token.
EP0 control register
EPCNTL0(8000h)
emptyi(Bit[8]) :1
interrupt
• Device causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_int (Bit[1]): 1
dev_ep_int(Bit[16]) :1
read
• Check that a setup token is received
Device status register
DEVS(0204h)
setup_int(Bit[27]) : 1
• Clear setup interrupt
Device status register
DEVS(0204h)
setup_int(Bit[27]) : 0
write
read
• Check that OUT transfer buffer can be read out
EP0 control register
EPCTRL0(8000h)
readyo_int(Bit[27]) : 1
• Clear readyo interrupt
EP0 command register
EPCMD0(0040h)
readyo_int_clr(Bit[19]) : 1
write
read
• Check the number of transfer bytes
EP1 counter register
EPCOUNT1(8084h)
phy_cnt(Bit[26:16]) : The number of transfer bytes
read
• Read out OUT buffer
EP0 OUT buffer
• Notify buffer-read completion
EP0 command register
EPCMD0(0040h)
bufrd(Bit[4]) : 1
write
Firm decodes the read-out EP0 buffer to prepare for IN/OUT transfer.

Figure 21-26

Control read transfer data stage

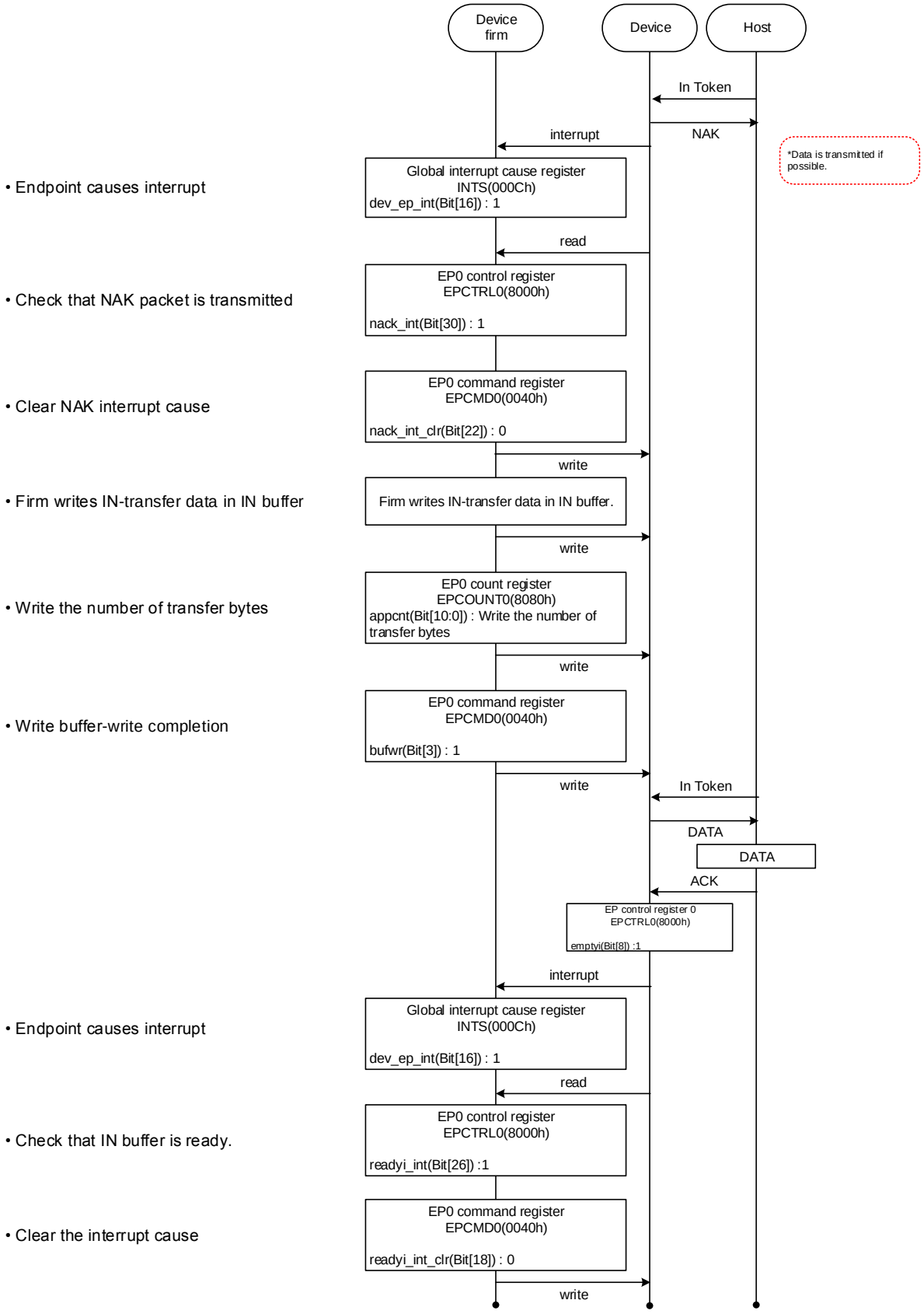

Device firm
Device
Host
In Token
NAK
interrupt
*Data is transmitted if possible.
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[16]) : 1
read
• Check that NAK packet is transmitted
EP0 control register
EPCTRL0(8000h)
nack_int(Bit[30]) : 1
• Clear NAK interrupt cause
EP0 command register
EPCMD0(0040h)
nack_int_clr(Bit[22]) : 0
write
• Firm writes IN-transfer data in IN buffer
Firm writes IN-transfer data in IN buffer.
write
• Write the number of transfer bytes
EP0 count register
EPCOUNT0(8080h)
appcnt(Bit[10:0]) : Write the number of transfer bytes
write
• Write buffer-write completion
EP0 command register
EPCMD0(0040h)
bufwr(Bit[3]) : 1
write
In Token
DATA
DATA
ACK
EP control register 0
EPCTRL0(8000h)
emptyi(Bit[8]) :1
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[16]) : 1
read
• Check that IN buffer is ready.
EP0 control register
EPCTRL0(8000h)
readyi_int(Bit[26]) :1
• Clear the interrupt cause
EP0 command register
EPCMD0(0040h)
readyi_int_clr(Bit[18]) : 0
write

Figure 21-27

Control read transfer status stage

- Endpoint causes interrupt
- Check that OUT buffer is ready.
- Clear the interrupt cause
- Write buffer-read completion

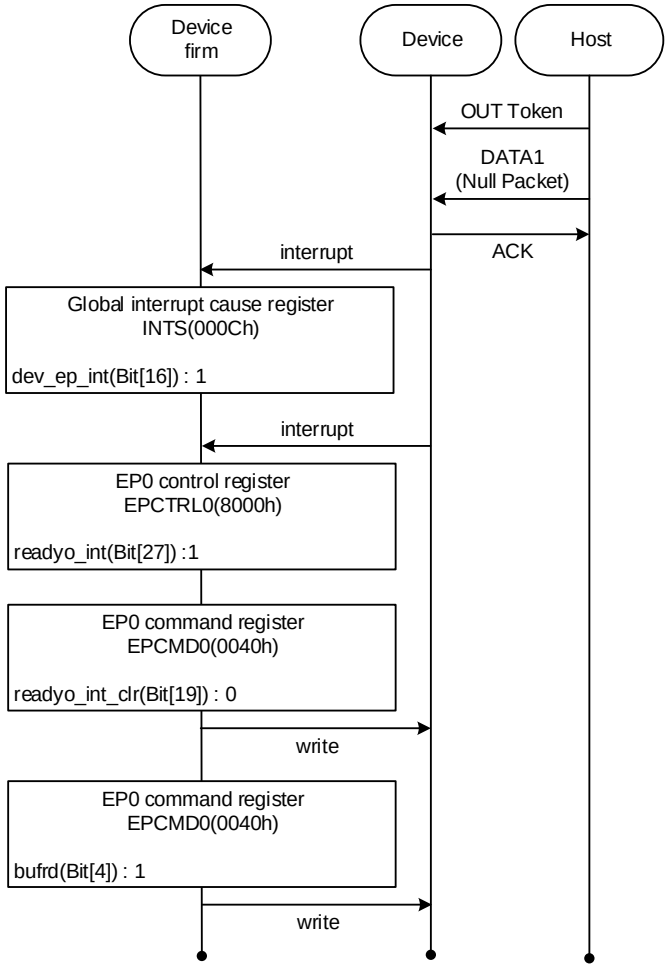
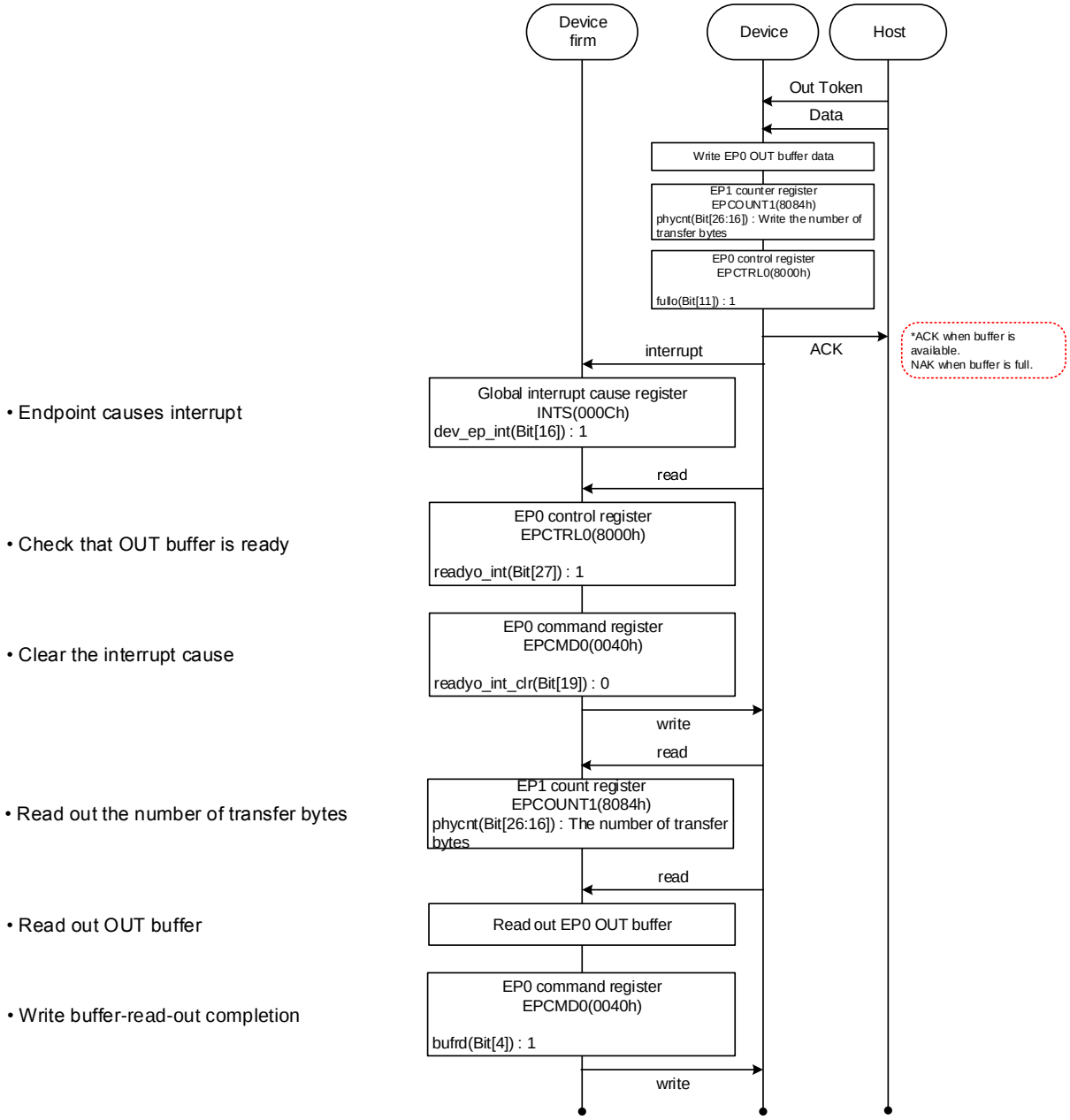


Figure 21-28

Control write transfer data stage



- Endpoint causes interrupt
- Check that OUT buffer is ready
- Clear the interrupt cause
- Read out the number of transfer bytes
- Read out OUT buffer
- Write buffer-read-out completion

Figure 21-29

Control write transfer status stage

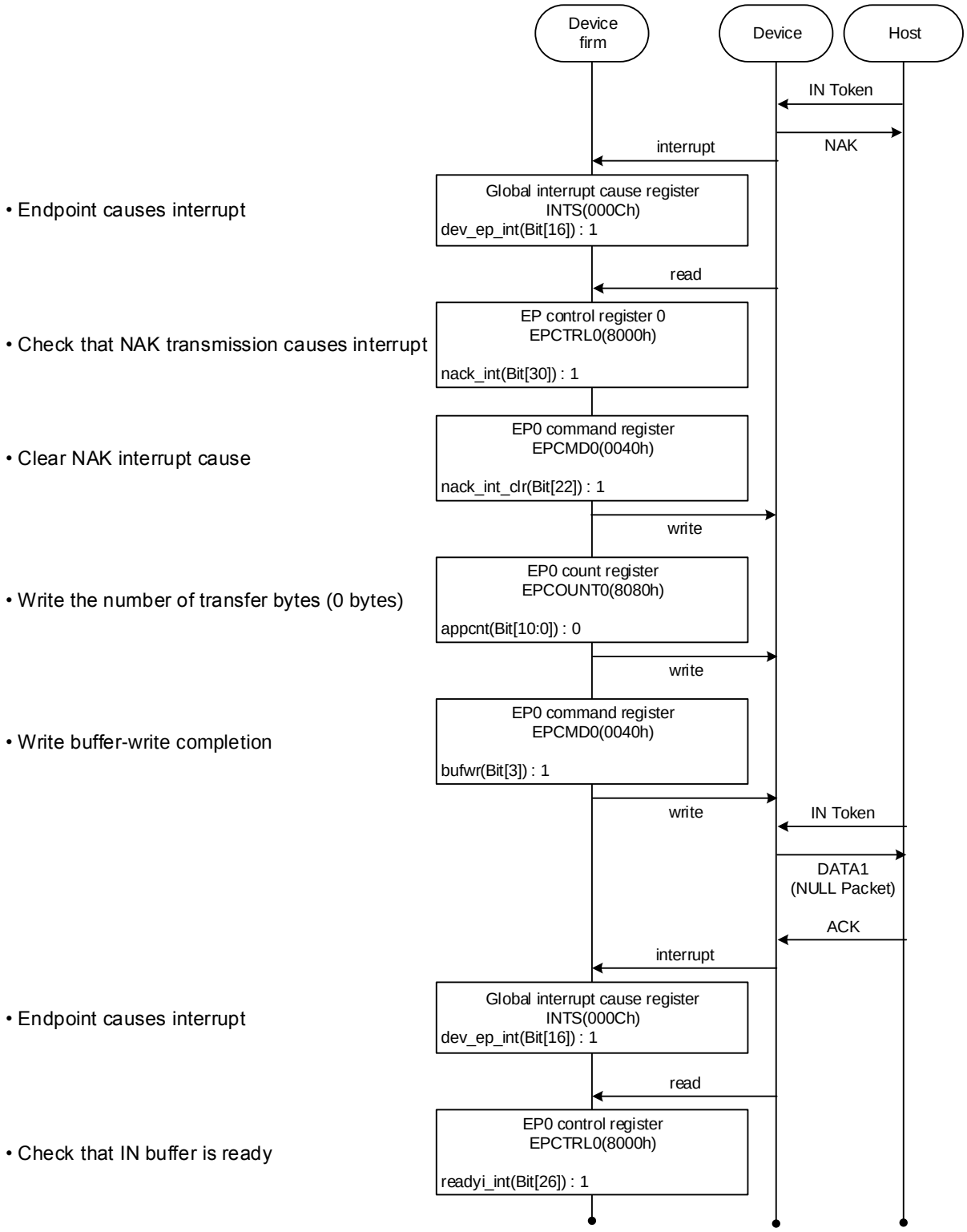


Figure 21-30

8.3.6. Control transfer (status_ok_int/status_ng_int is used.)

Control read transfer: Success

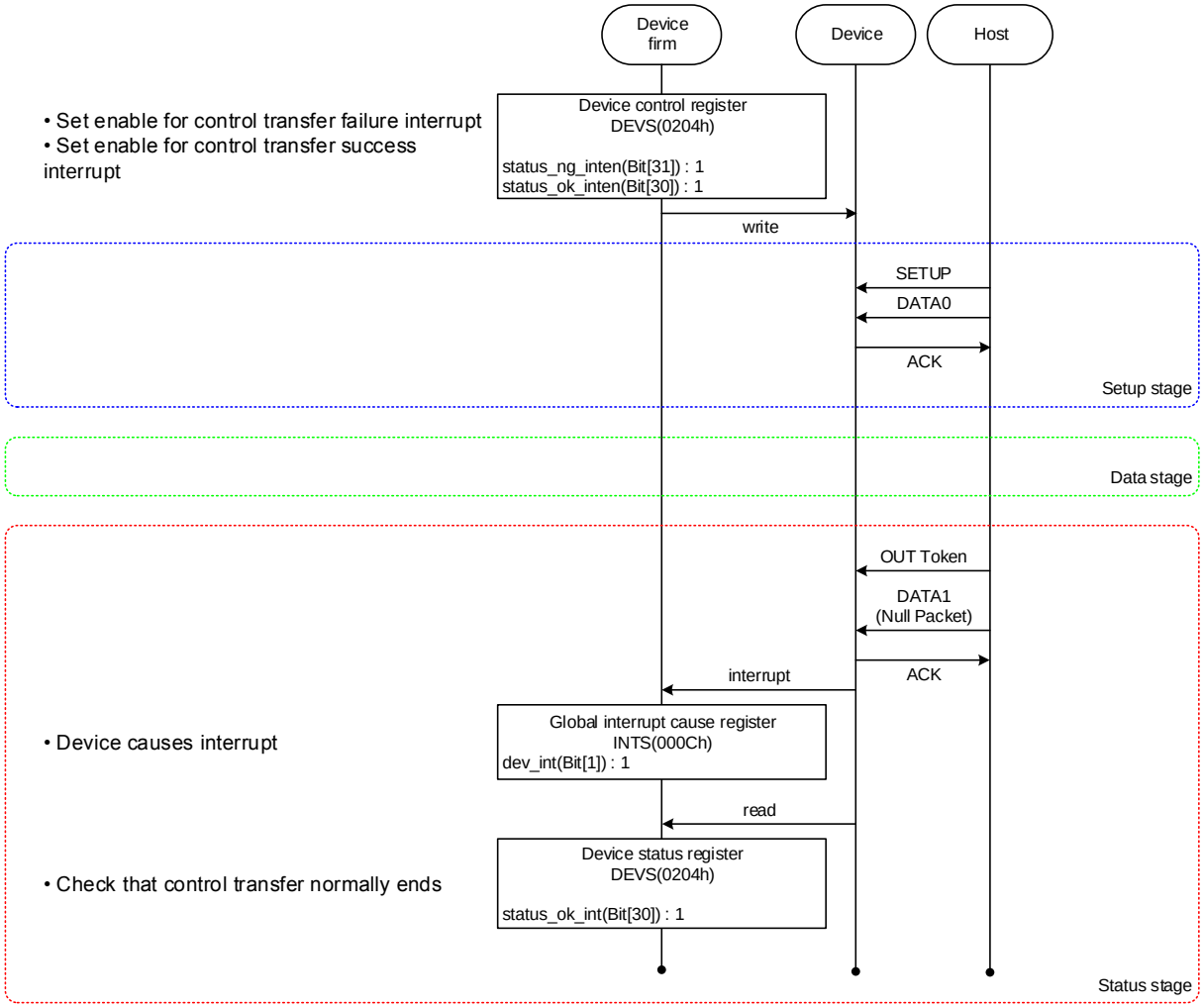


Figure 21-31

Control read transfer: Failure

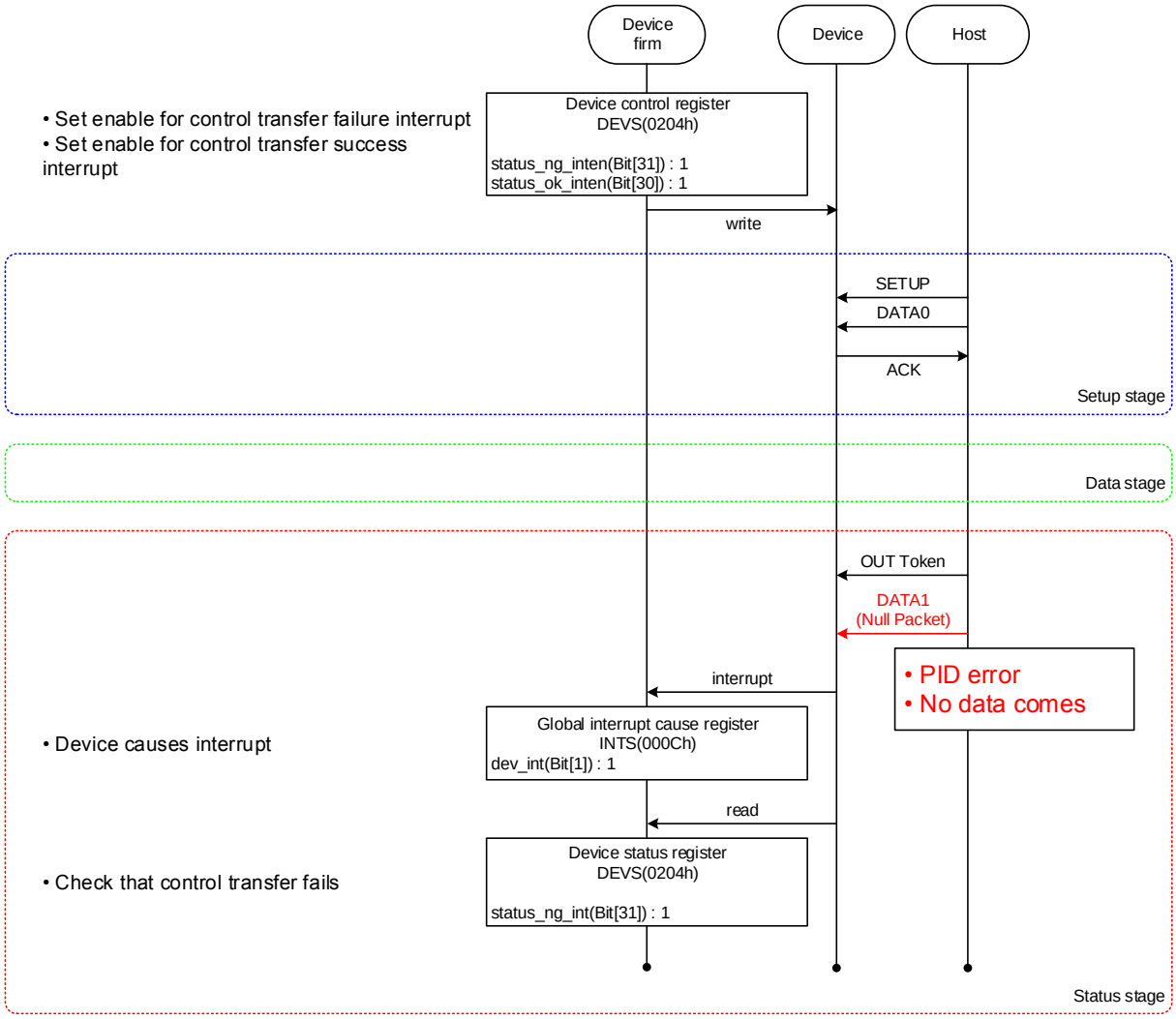


Figure 21-32

Control write transfer: Success

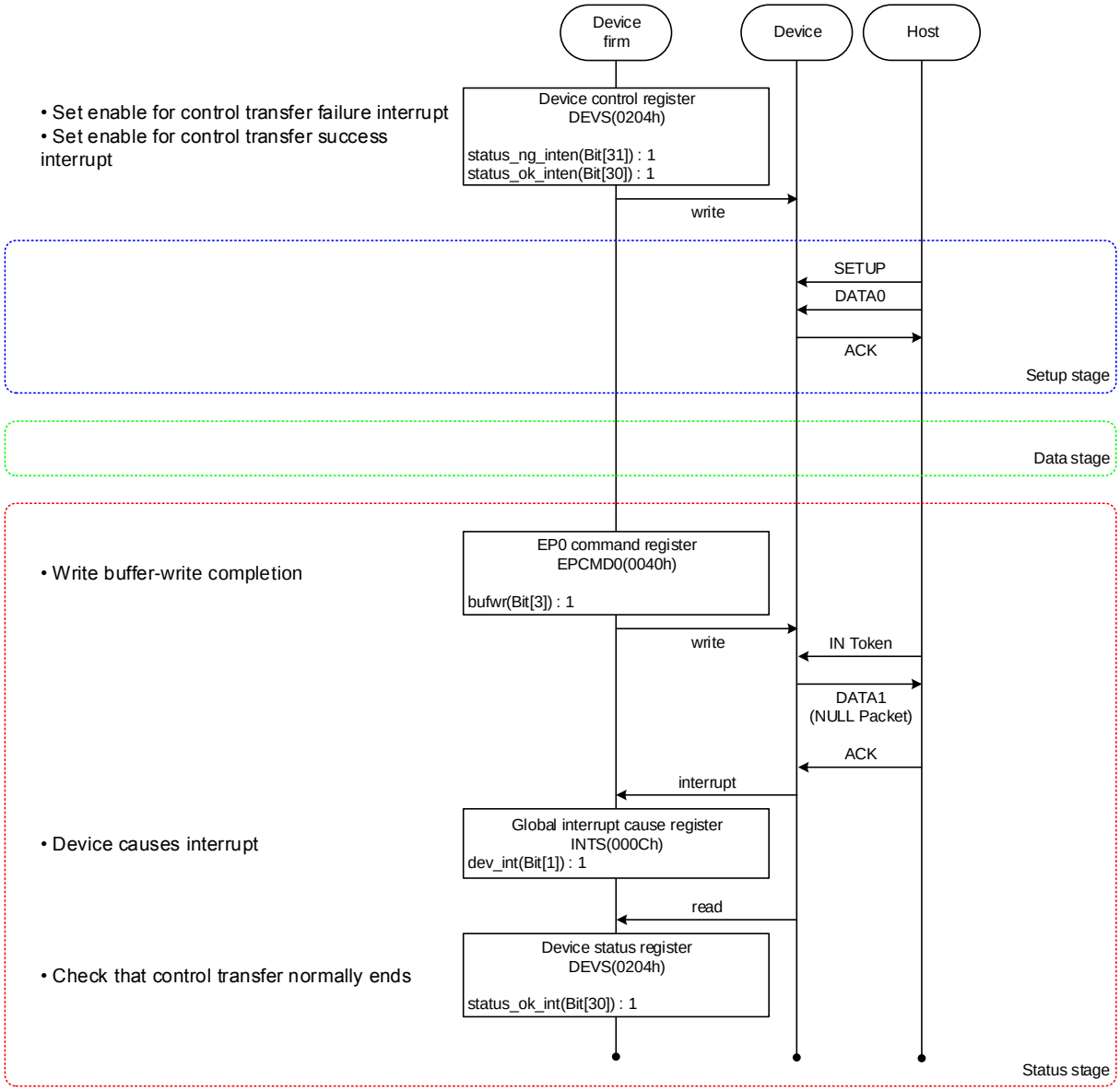


Figure 21-33

Control write transfer: Failure

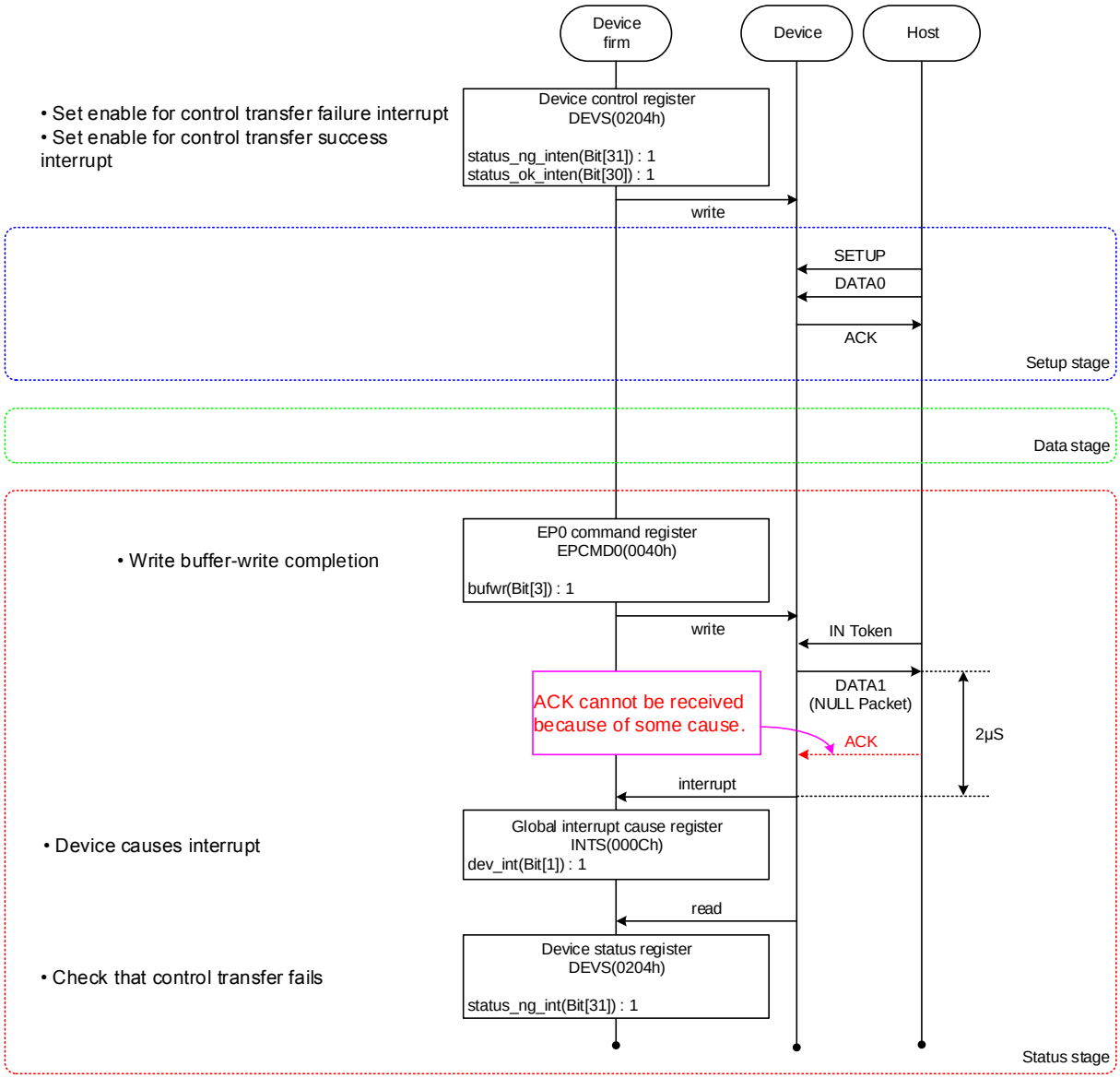


Figure 21-34

SETUP multiple reception

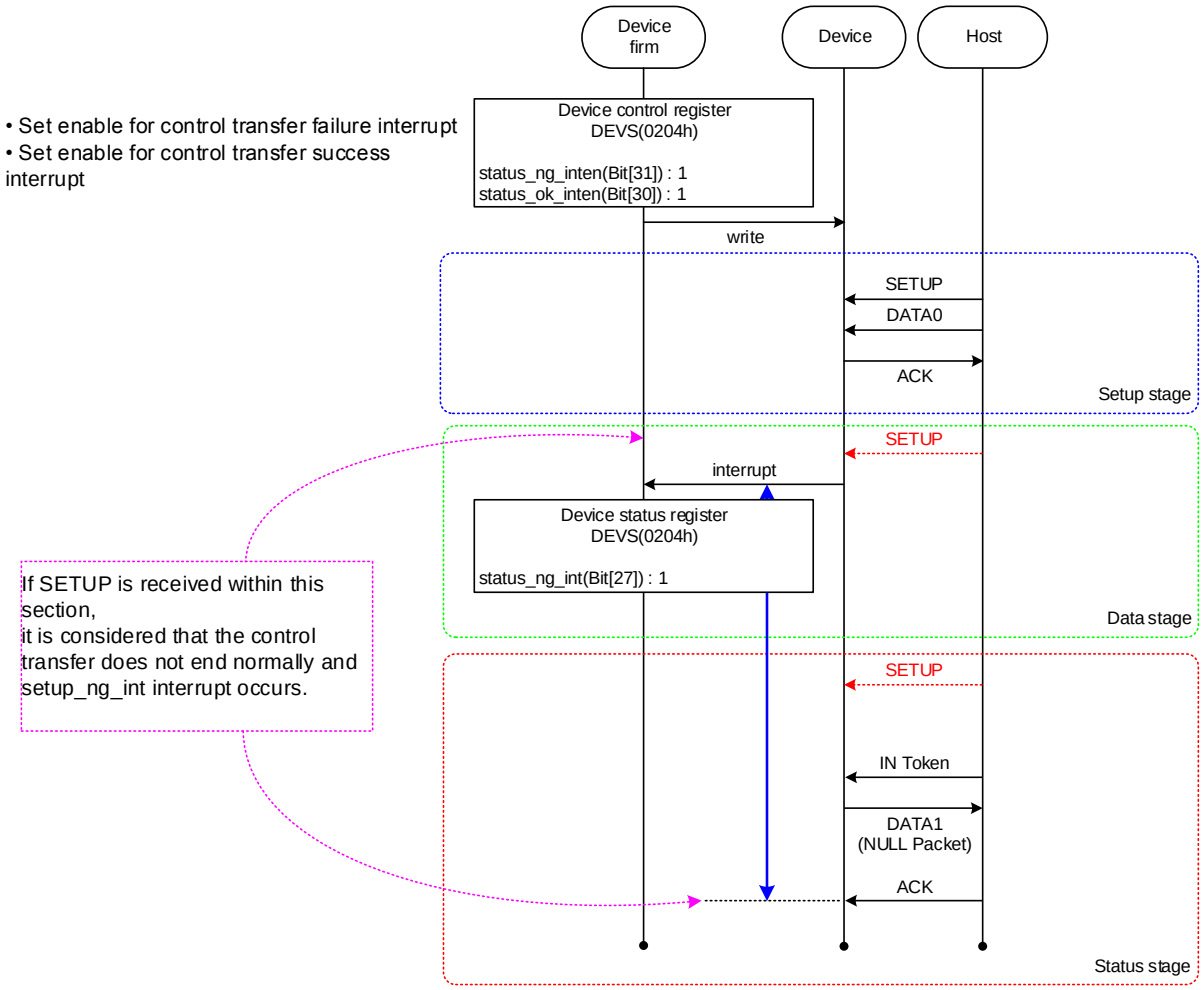


Figure 21-35

8.3.7. Data (IN) transfer of multi-plane buffer

Process of interrupt and flag

There are 2 ways below in process of interrupt and flag at IN transfer according to the setting of dev_int_mode (Bit[2]) of MODE register.

dev_int_mode = 0

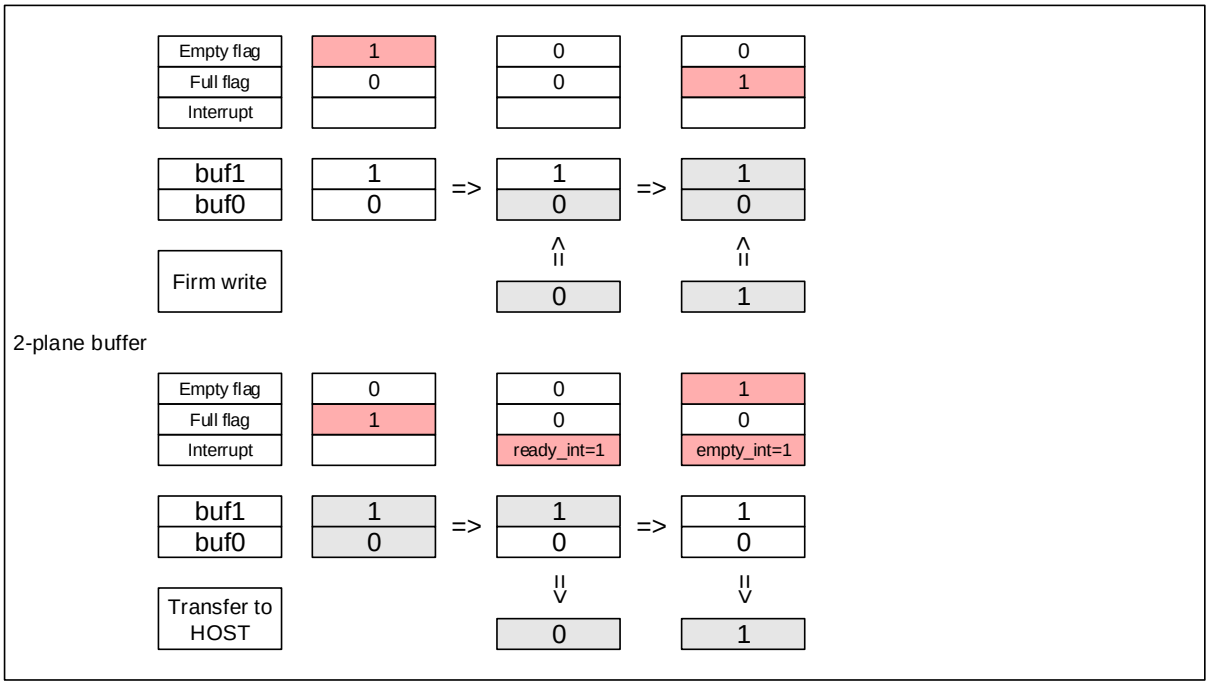


Figure 21-36

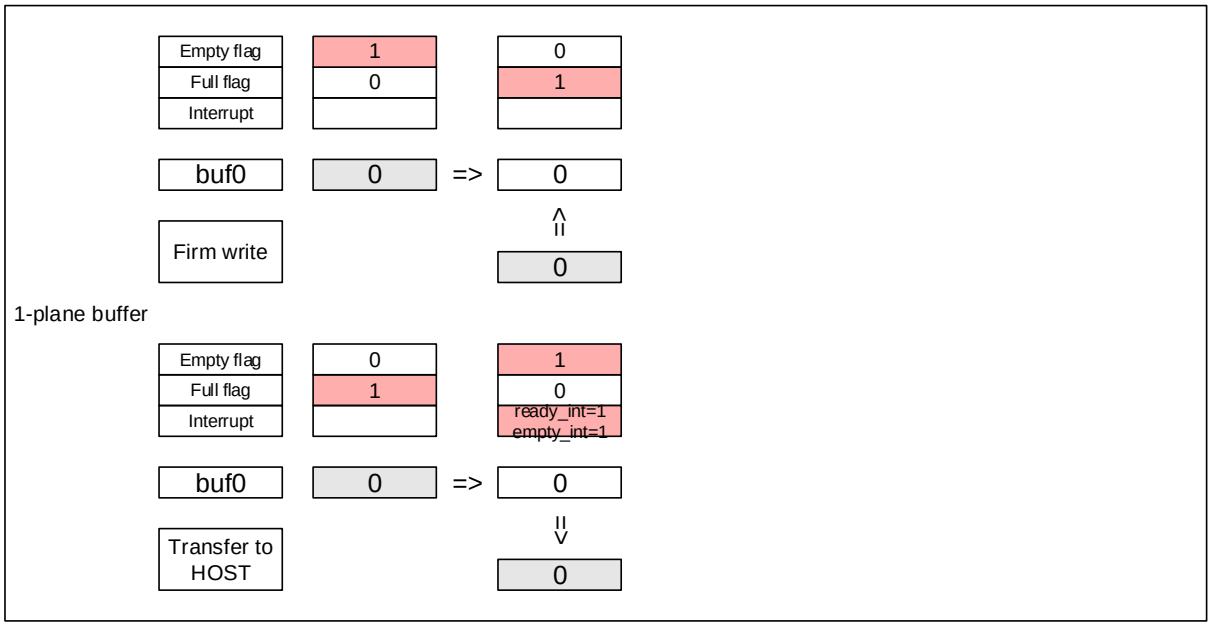


Figure 21-37

dev_int_mode = 1

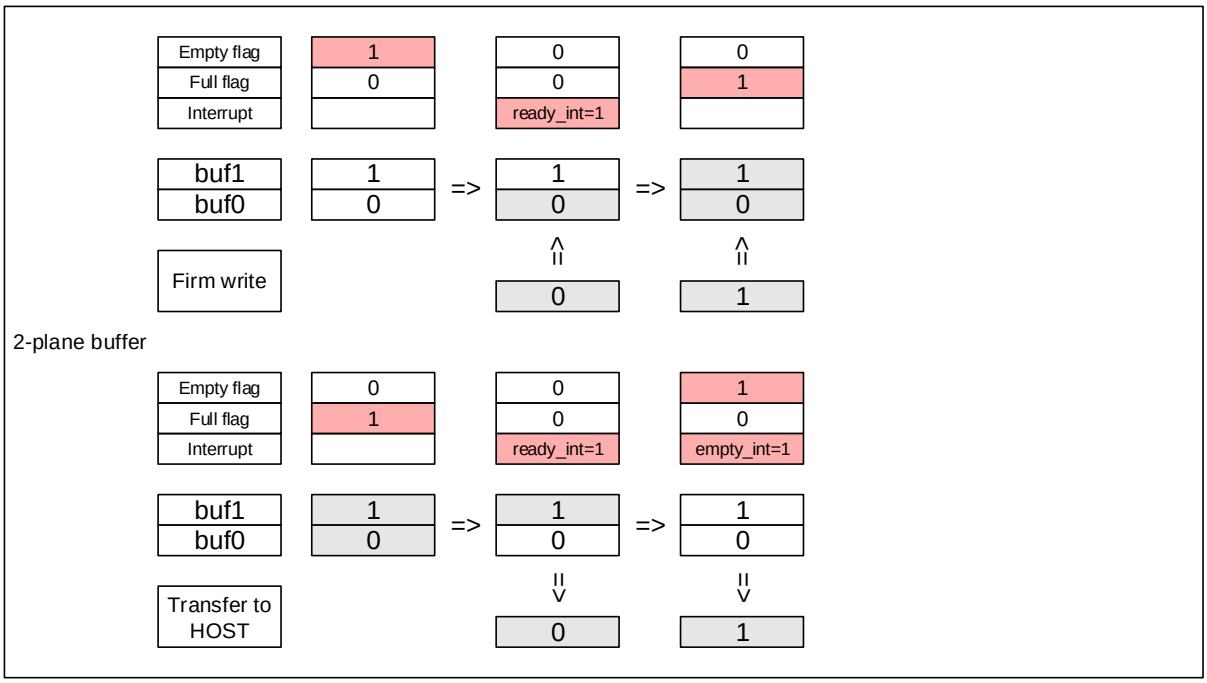


Figure 21-38

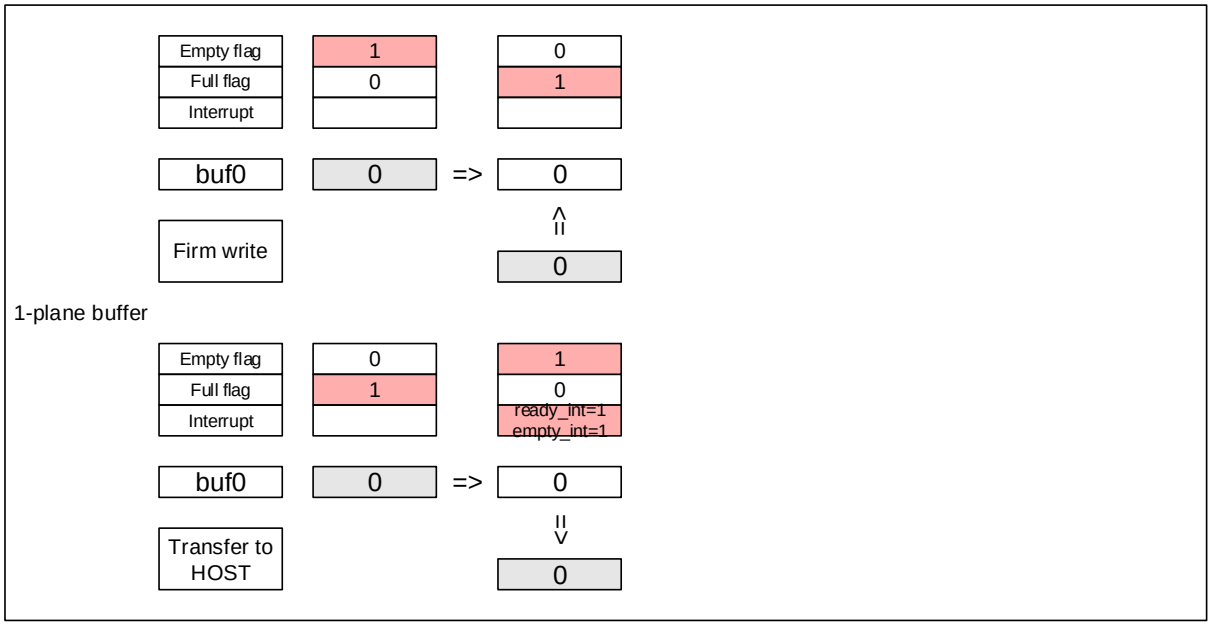


Figure 21-39

Data IN transfer of multi-plane buffer (when dev_int_mode = 0)

Preconditions
• Interrupt type: dev_int_mode = 0
• Target EP: EP1
• Transfer type: Bulk
• Transfer direction: IN
• EP size: 64 bytes
• The number of buffer planes: 2 planes
• Buffer 0 address: 8200h
• Buffer 1 address: 8240h
• The number of transfer bytes: 192 bytes

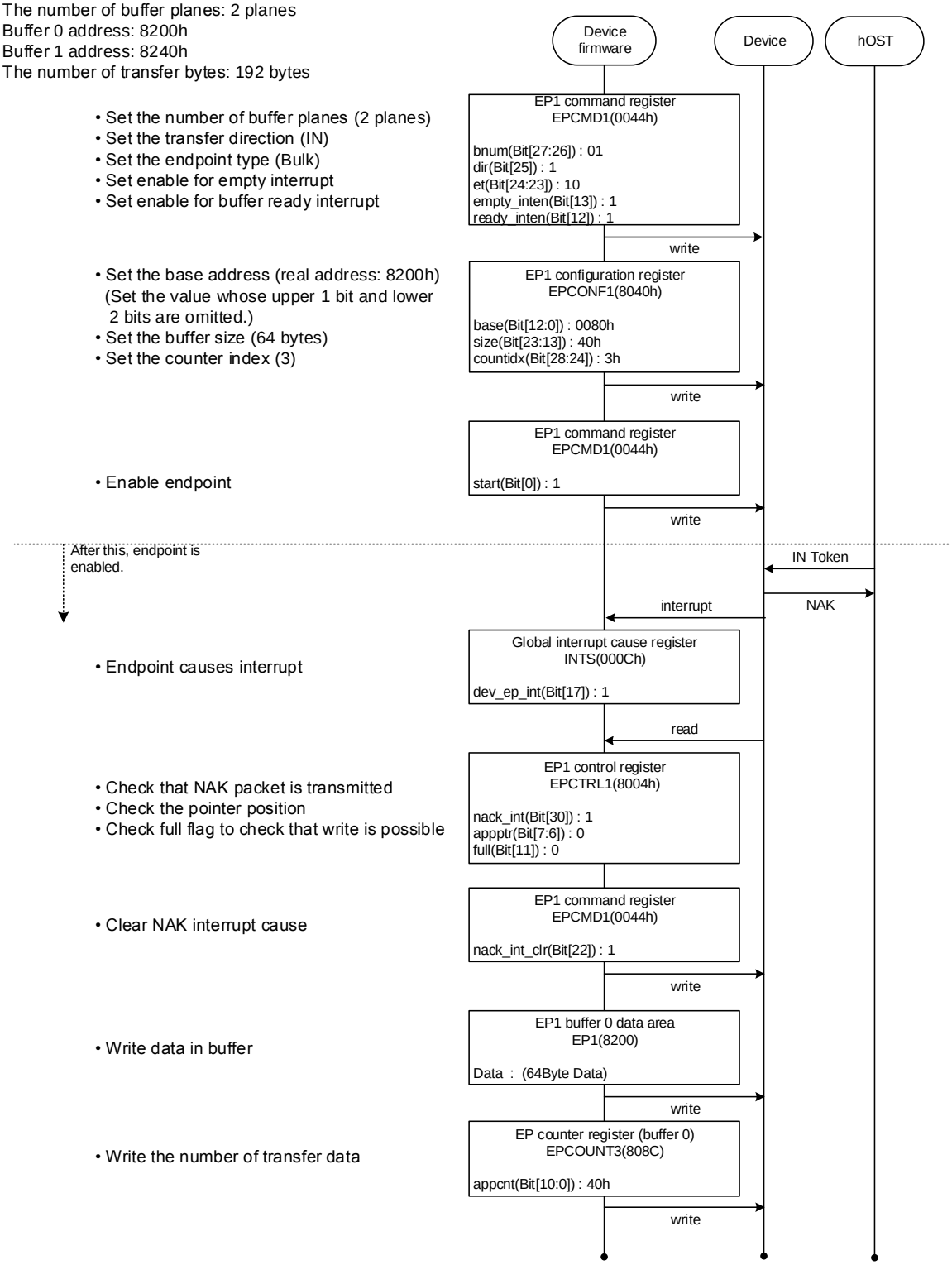


Figure 21-40

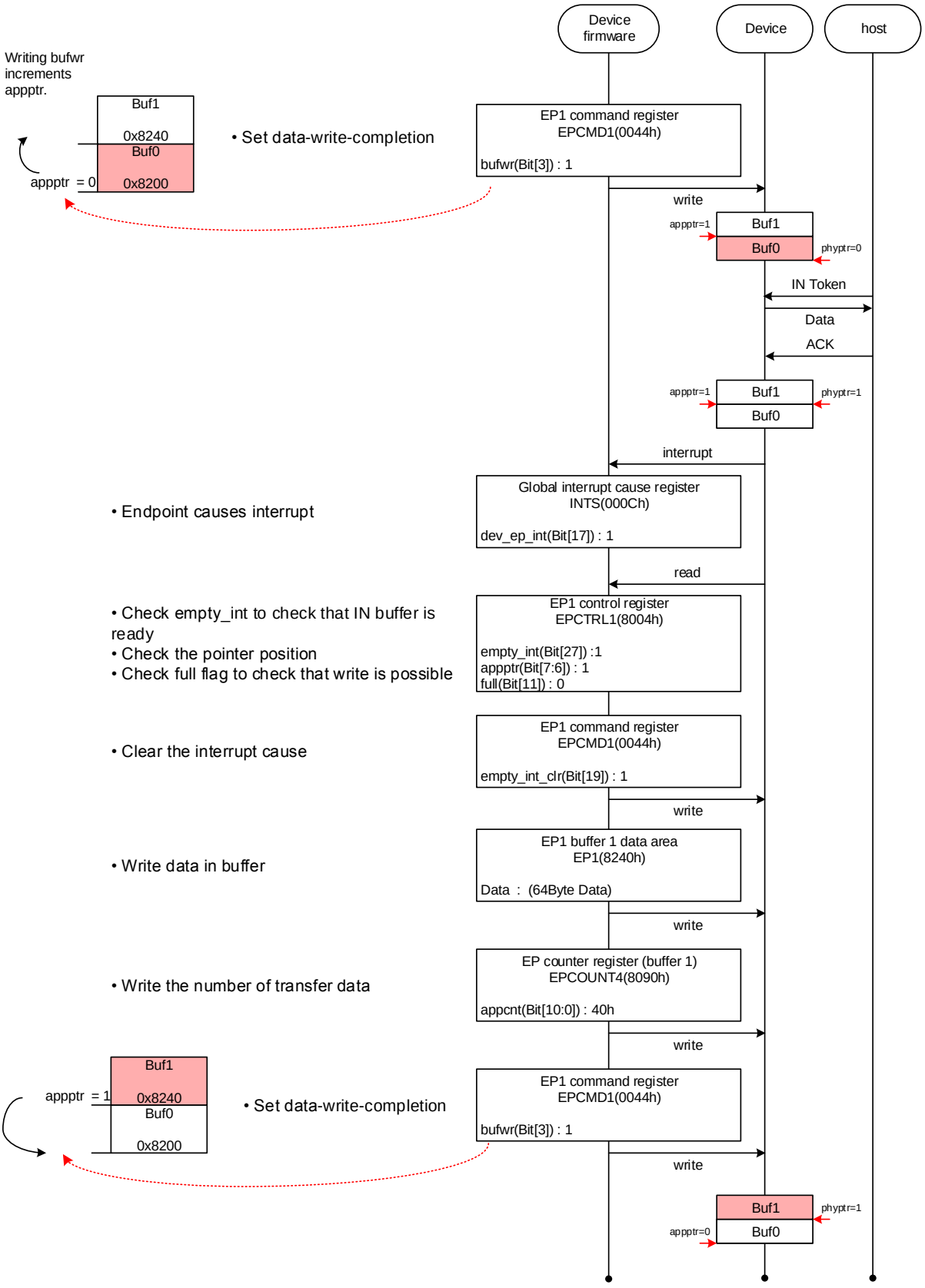

Device firmware
Device
host
Writing bufwr increments appptr.
Buf1
0x8240
Buf0
0x8200
appptr = 0
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0
IN Token
Data
ACK
appptr=1
Buf1
Buf0
phyptr=1
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) :1
appptr(Bit[7:6]) : 1
full(Bit[11]) : 0
• Check empty_int to check that IN buffer is ready
• Check the pointer position
• Check full flag to check that write is possible
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write
EP1 buffer 1 data area
EP1(8240h)
Data : (64Byte Data)
• Write data in buffer
write
EP counter register (buffer 1)
EPCOUNT4(8090h)
appcnt(Bit[10:0]) : 40h
• Write the number of transfer data
write
Buf1
0x8240
Buf0
0x8200
appptr = 1
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
Buf1
Buf0
phyptr=1
appptr=0

Figure 21-41

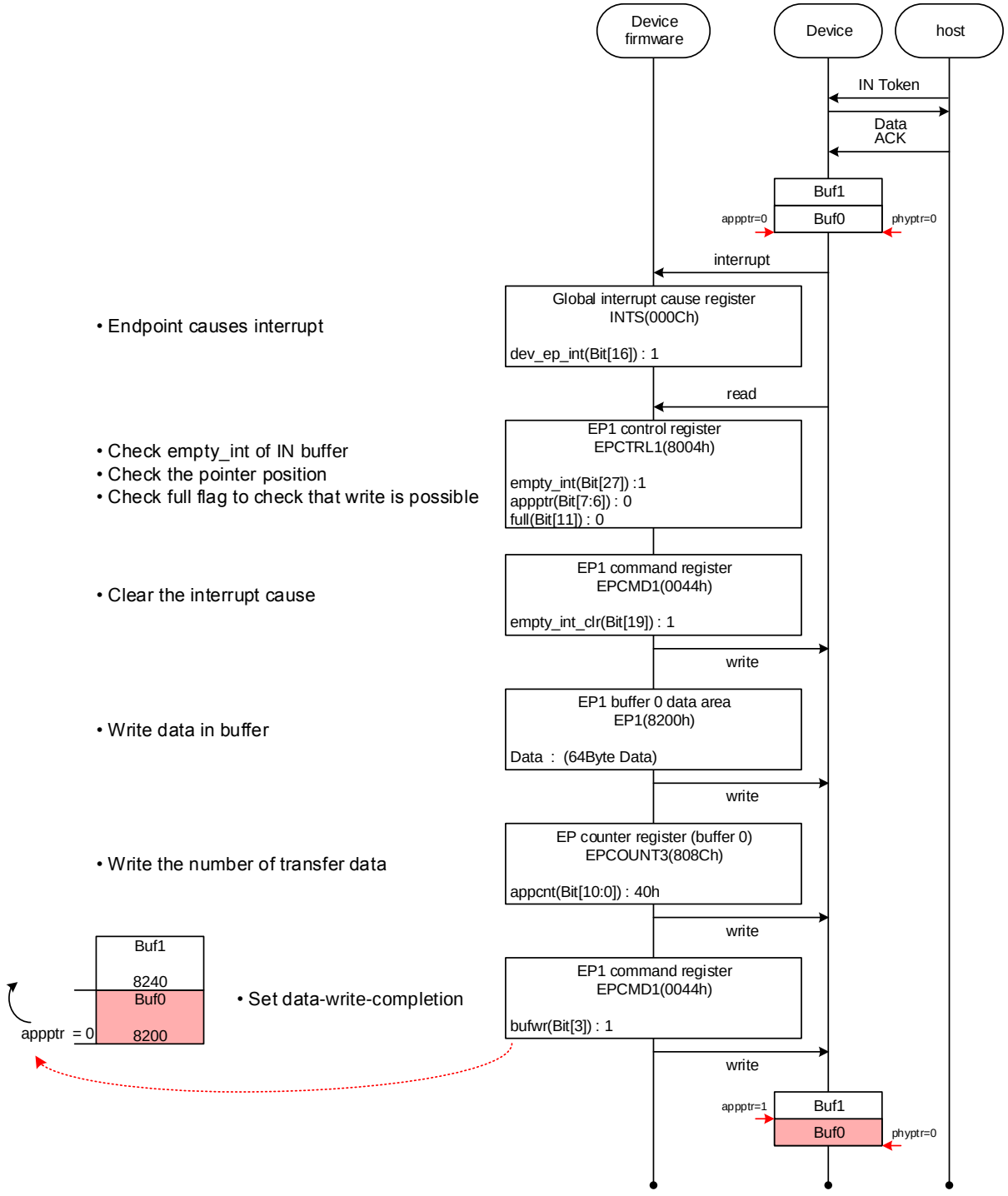

Device firmware
Device
host
IN Token
Data
ACK
Buf1
Buf0
appptr=0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[16]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) :1
appptr(Bit[7:6]) : 0
full(Bit[11]) : 0
• Check empty_int of IN buffer
• Check the pointer position
• Check full flag to check that write is possible
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write
EP1 buffer 0 data area
EP1(8200h)
Data : (64Byte Data)
• Write data in buffer
write
EP counter register (buffer 0)
EPCOUNT3(808Ch)
appcnt(Bit[10:0]) : 40h
• Write the number of transfer data
write
Buf1
8240
Buf0
8200
appptr = 0
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0

Figure 21-42

- Endpoint causes interrupt

- Check empty_int of IN buffer
- Check the pointer position
- Check full flag to check that write is possible

- Clear the interrupt cause

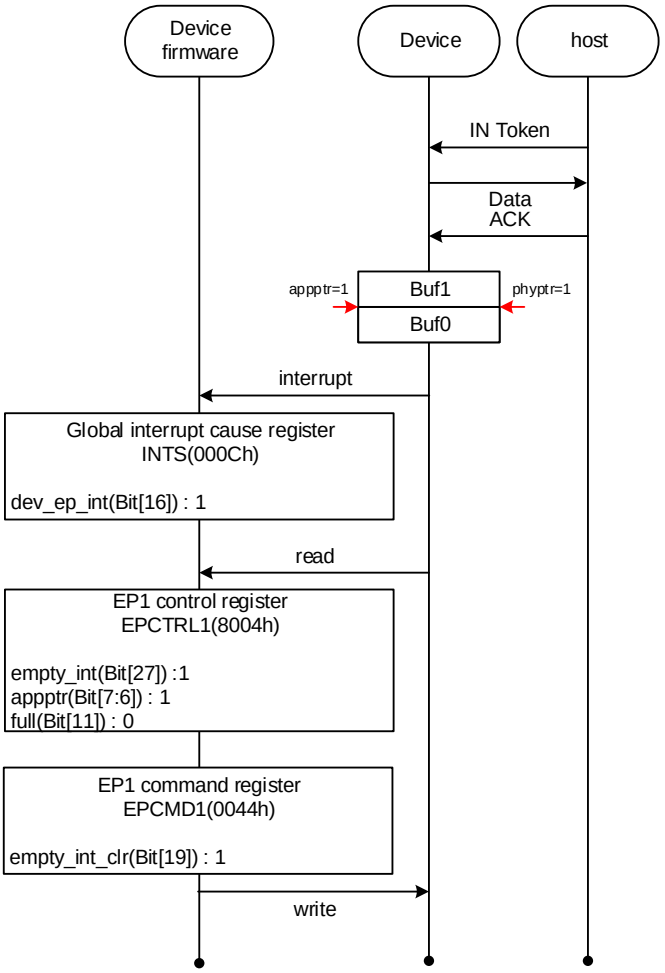


Figure 21-43

Data IN transfer of multi-plane buffer (when dev_int_mode = 1)

Preconditions
• Interrupt type: dev_int_mode = 1
• Target EP: EP1
• Transfer type: Bulk
• Transfer direction: IN
• EP size: 64 bytes
• The number of buffer planes: 2 planes
• Buffer 0 address: 8200h
• Buffer 1 address: 8240h
• The number of transfer bytes: 192 bytes

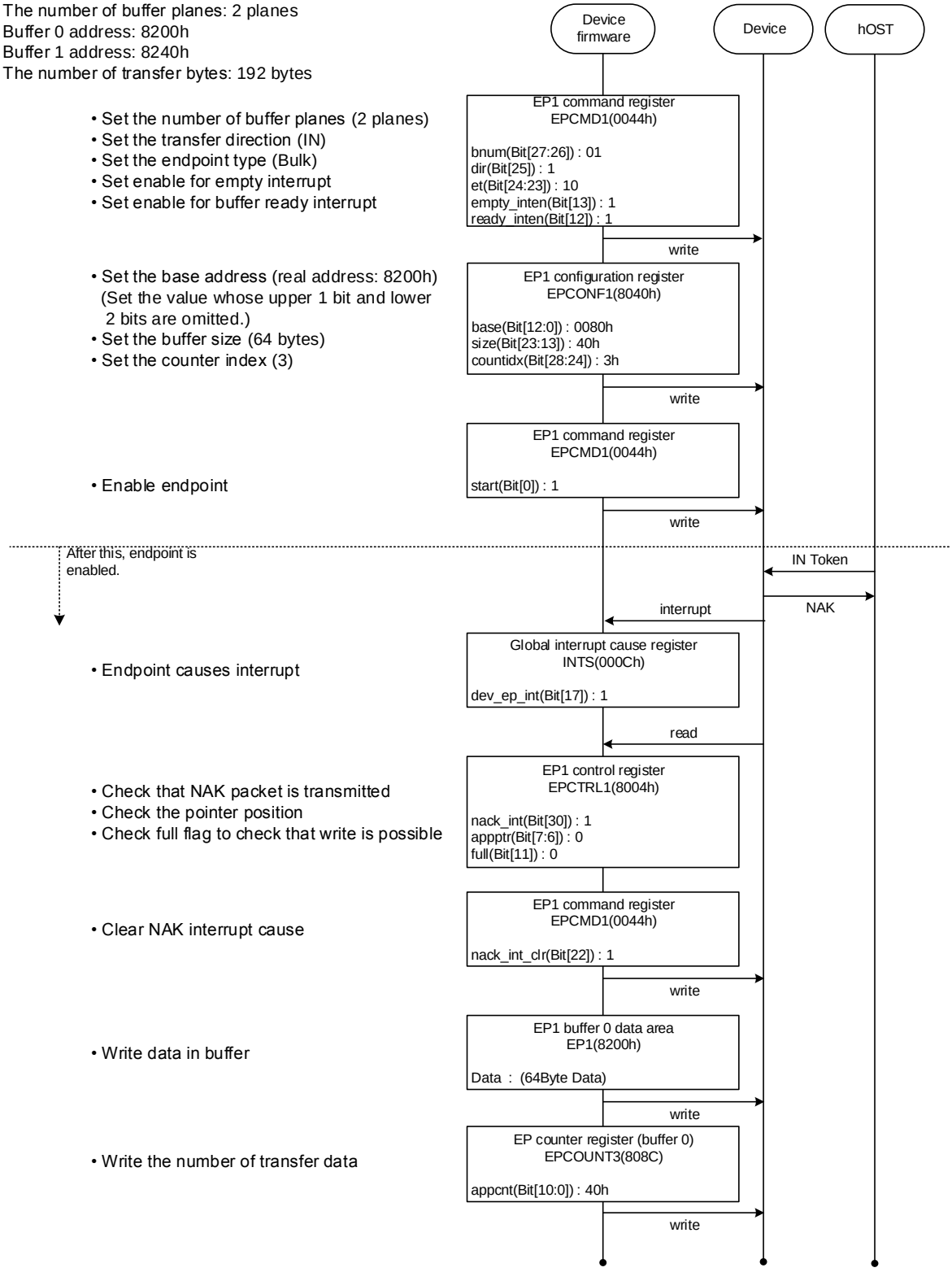


Figure 21-44

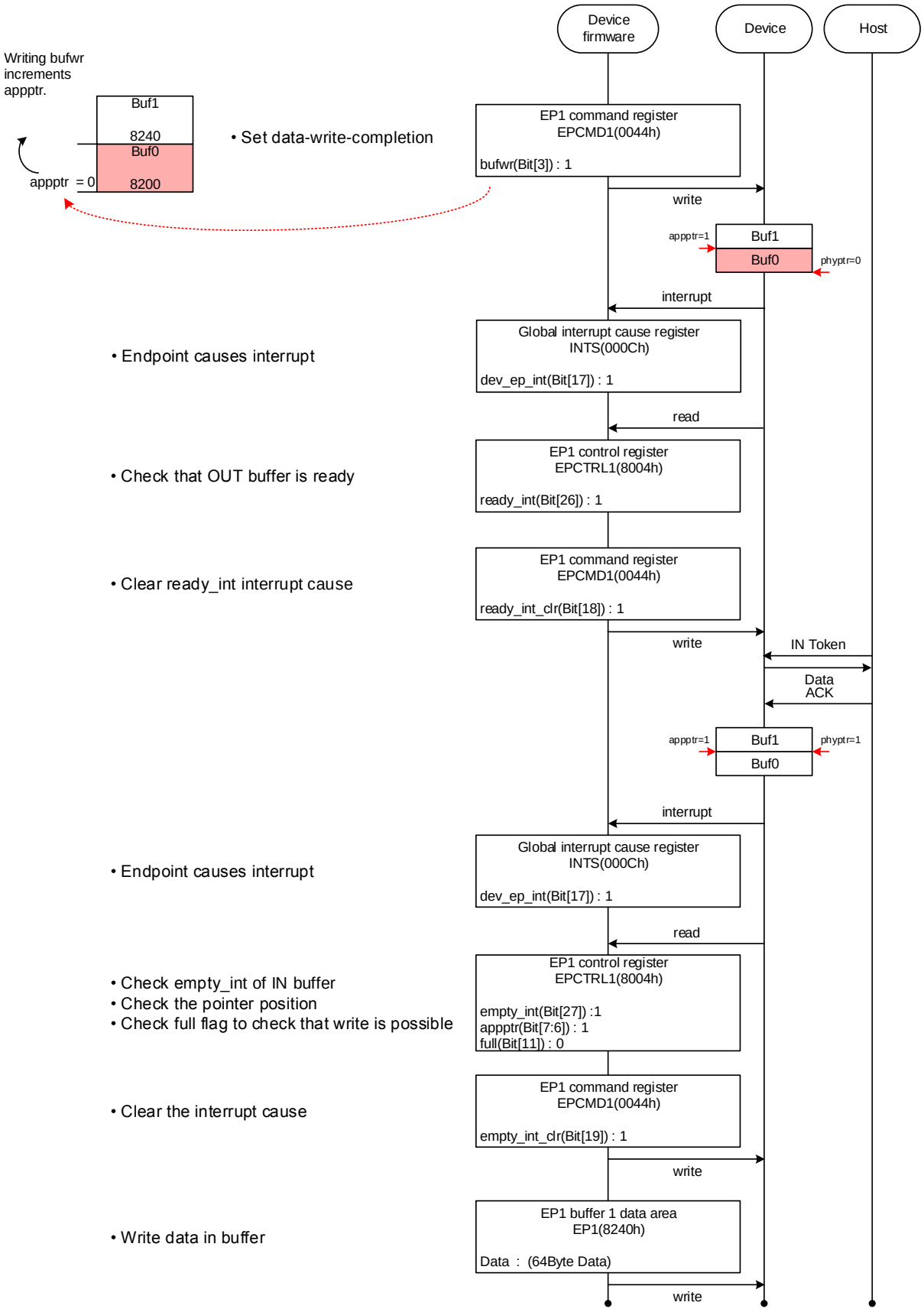

Device firmware
Device
Host
Writing bufwr increments appptr.
Buf1
8240
Buf0
8200
appptr = 0
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
• Check that OUT buffer is ready
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write
IN Token
Data
ACK
appptr=1
Buf1
Buf0
phyptr=1
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) :1
appptr(Bit[7:6]) : 1
full(Bit[11]) : 0
• Check empty_int of IN buffer
• Check the pointer position
• Check full flag to check that write is possible
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write
EP1 buffer 1 data area
EP1(8240h)
Data : (64Byte Data)
• Write data in buffer
write

Figure 21-45

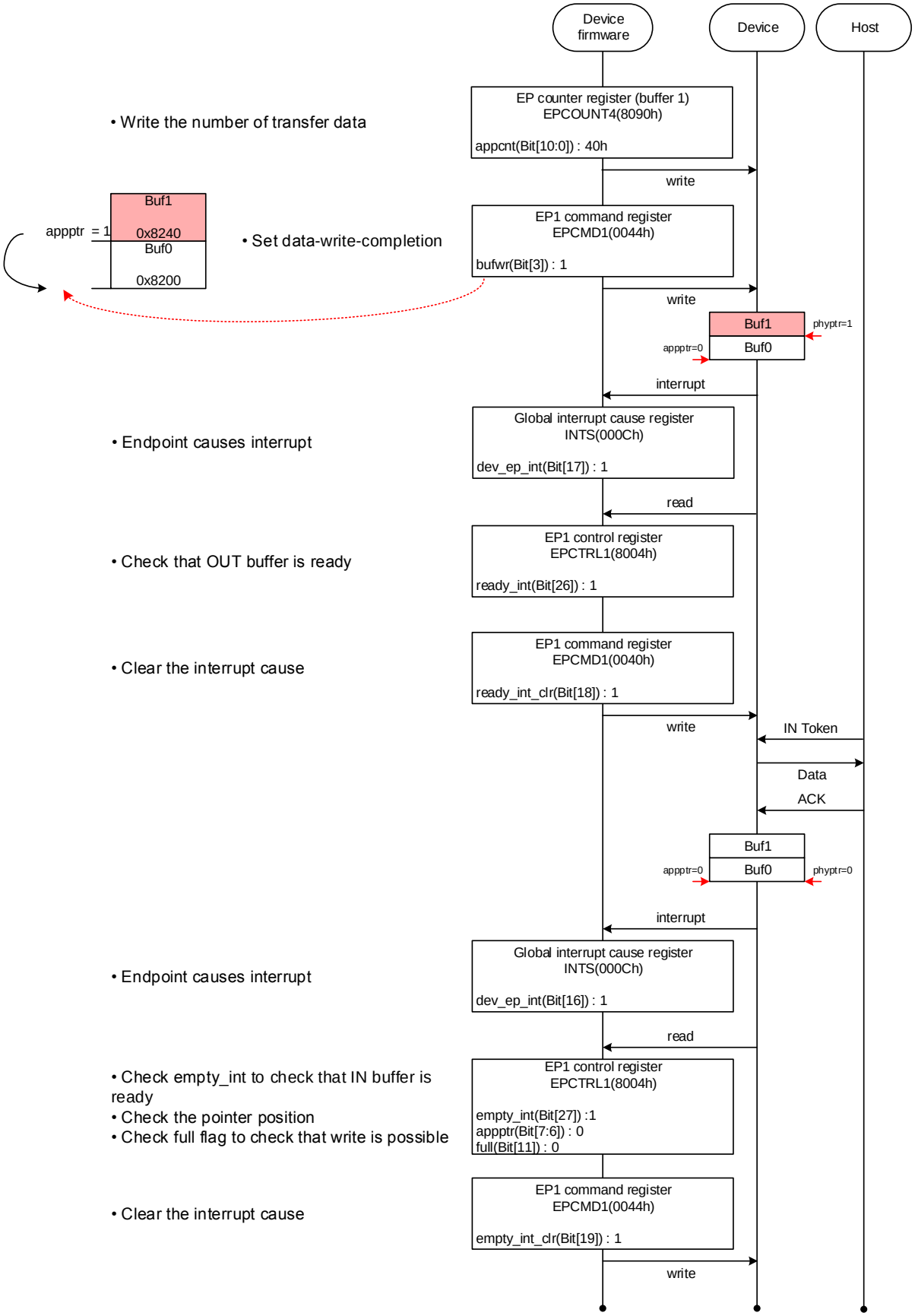

Device firmware
Device
Host
• Write the number of transfer data
EP counter register (buffer 1)
EPCOUNT4(8090h)
appcnt(Bit[10:0]) : 40h
write
Buf1
0x8240
Buf0
0x8200
appptr = 1
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
Buf1
Buf0
phyptr=1
appptr=0
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
read
• Check that OUT buffer is ready
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
• Clear the interrupt cause
EP1 command register
EPCMD1(0040h)
ready_int_clr(Bit[18]) : 1
write
IN Token
Data
ACK
Buf1
Buf0
appptr=0
phyptr=0
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[16]) : 1
read
• Check empty_int to check that IN buffer is ready
• Check the pointer position
• Check full flag to check that write is possible
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) :1
appptr(Bit[7:6]) : 0
full(Bit[11]) : 0
• Clear the interrupt cause
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
write

Figure 21-46

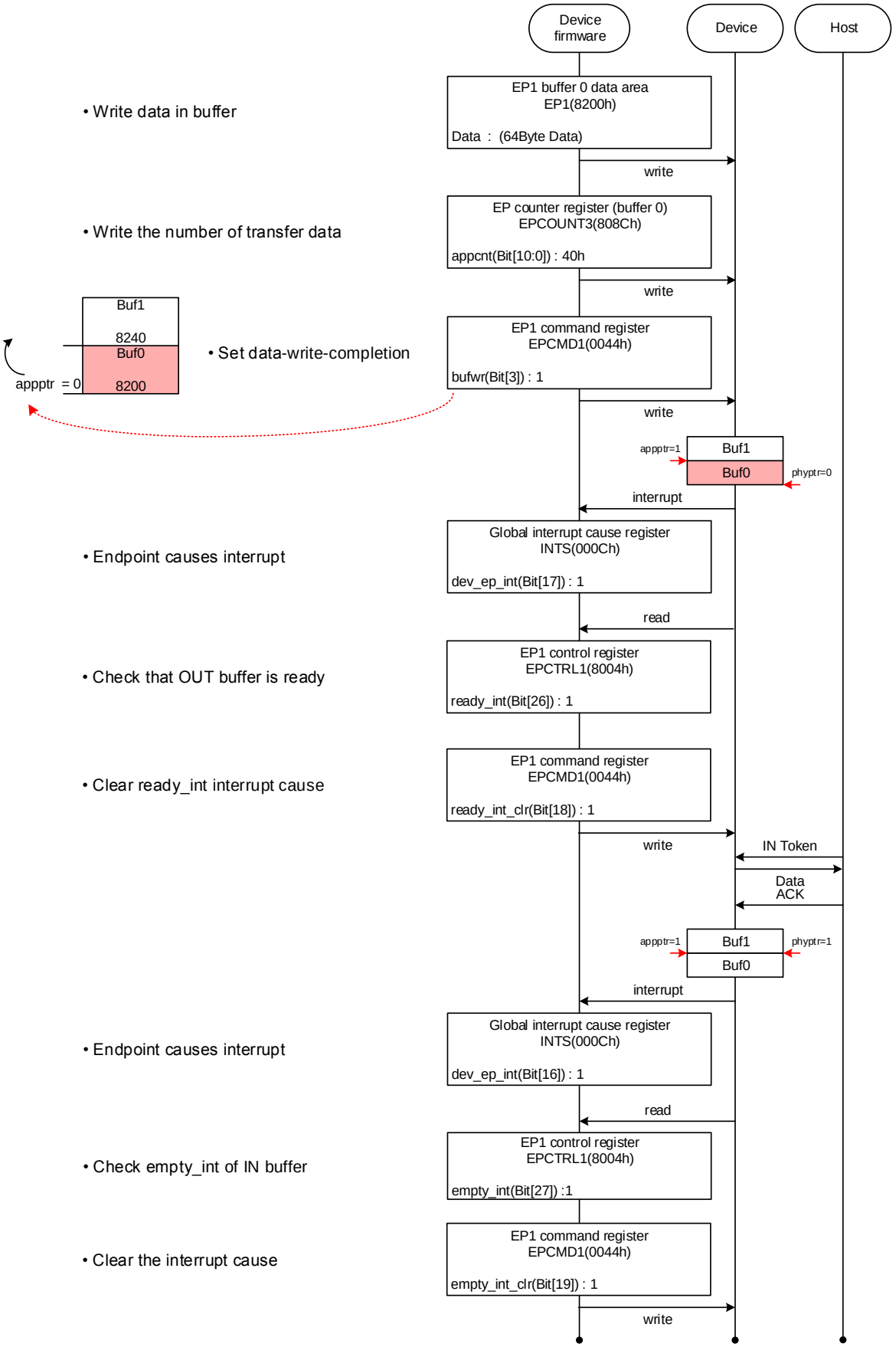

Device firmware
Device
Host
• Write data in buffer
EP1 buffer 0 data area
EP1(8200h)
Data : (64Byte Data)
write
• Write the number of transfer data
EP counter register (buffer 0)
EPCOUNT3(808Ch)
appcnt(Bit[10:0]) : 40h
write
Buf1
8240
Buf0
8200
appptr = 0
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
read
• Check that OUT buffer is ready
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
• Clear ready_int interrupt cause
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
write
IN Token
Data
ACK
appptr=1
Buf1
Buf0
phyptr=1
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[16]) : 1
read
• Check empty_int of IN buffer
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) :1
• Clear the interrupt cause
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
write

Figure 21-47

Data IN transfer of multi-plane buffer [When FULL FLAG is on] (when dev_int_mode = 0)

Preconditions
- Interrupt type: dev_int_mode = 1
- Target EP: EP1
- Transfer type: Bulk
- Transfer direction: IN
- EP size: 64 bytes
- The number of buffer planes: 2 planes
- Buffer 0 address: 8200h
- Buffer 1 address: 8240h
- The number of transfer bytes: 192 bytes

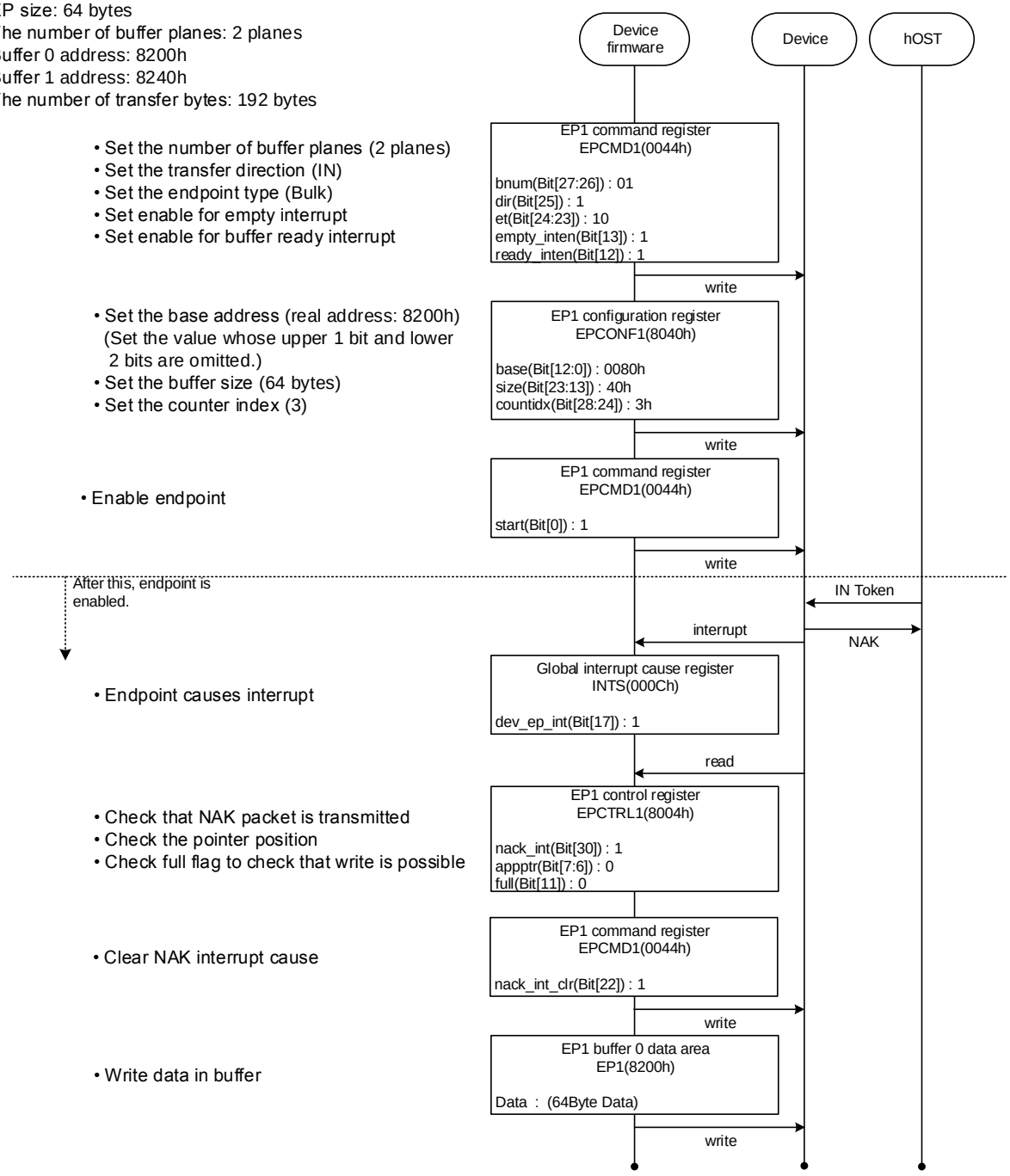


Figure 21-48

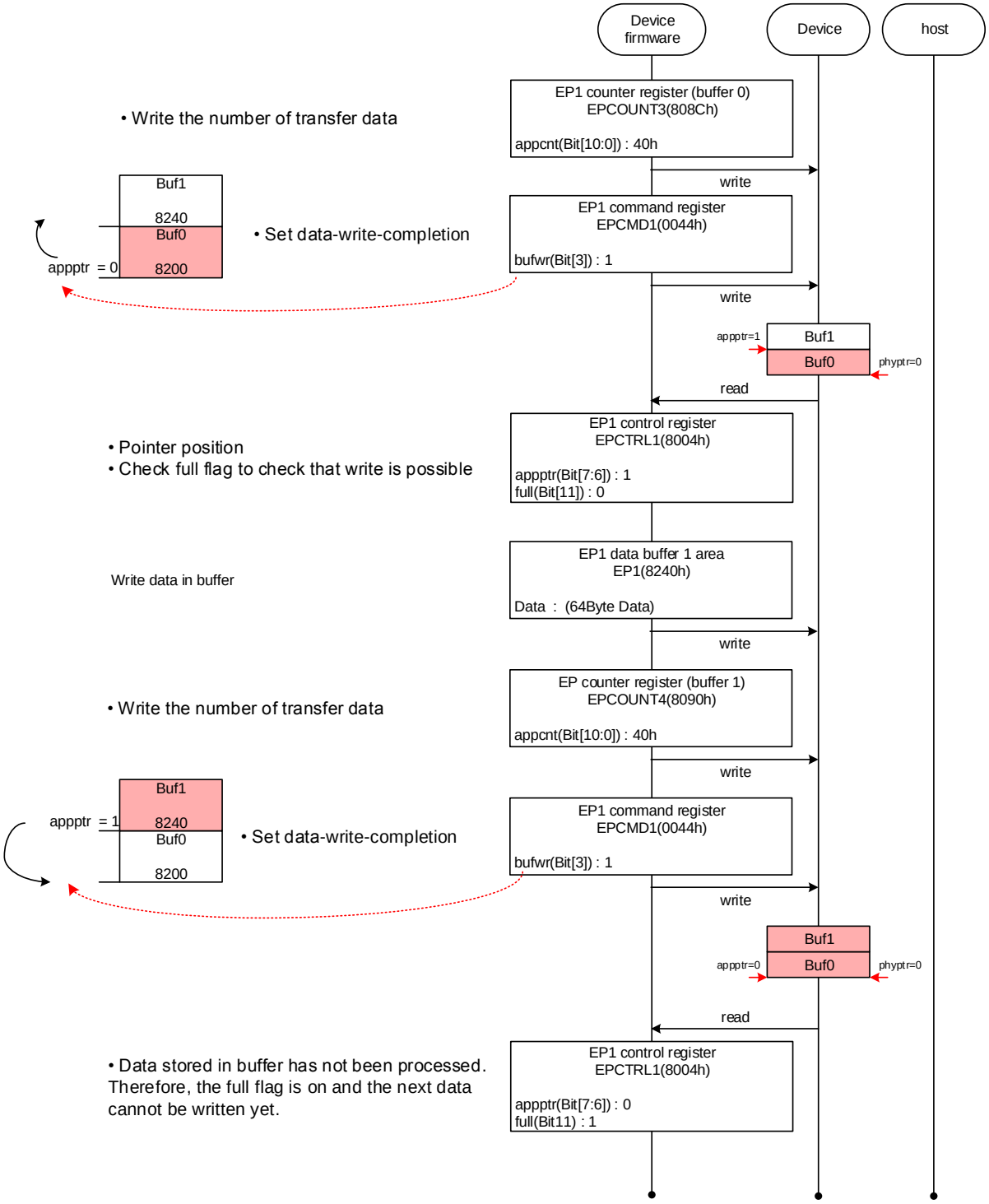
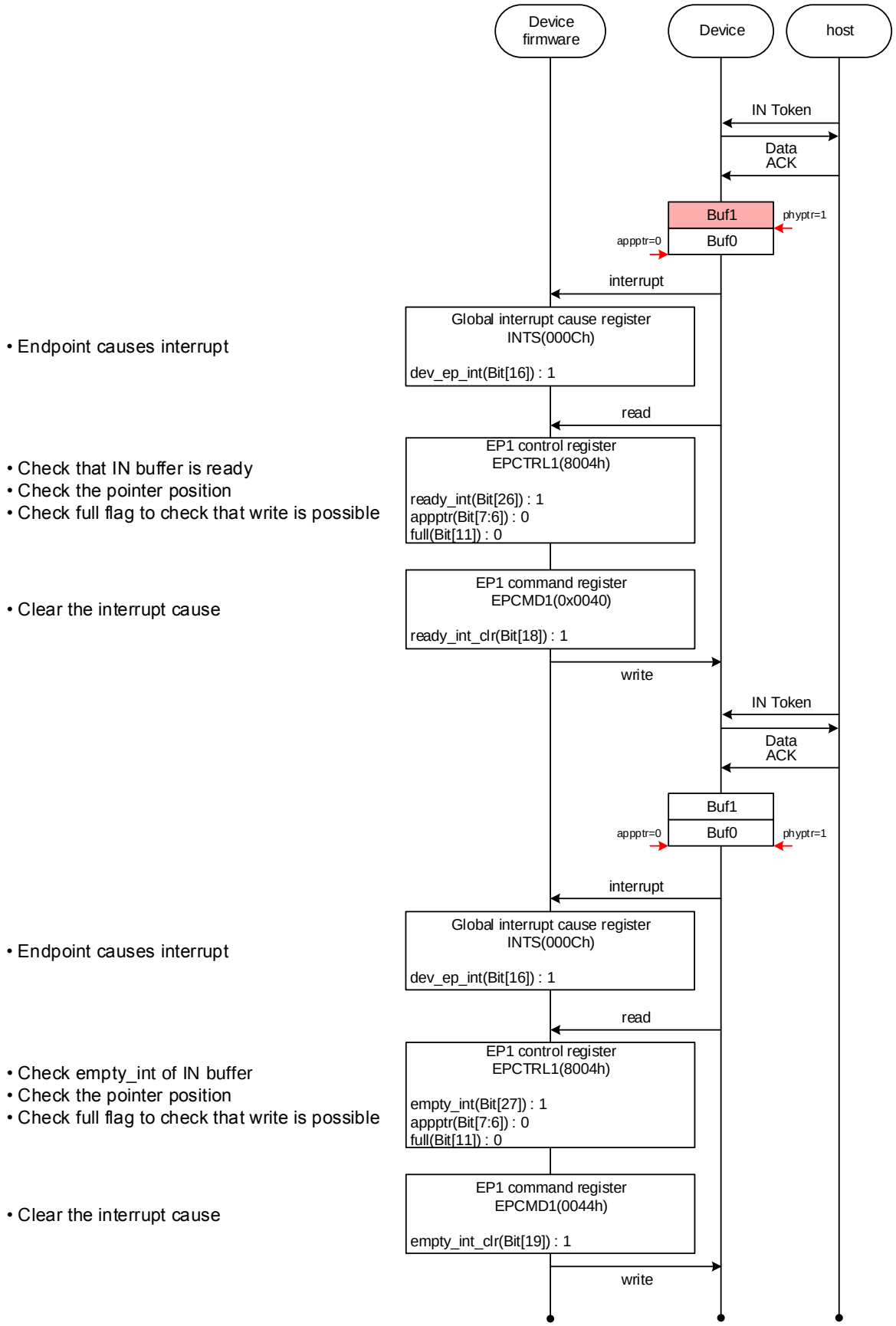

Device firmware
Device
host
EP1 counter register (buffer 0)
EPCOUNT3(808Ch)
appcnt(Bit[10:0]) : 40h
• Write the number of transfer data
write
Buf1
8240
Buf0
8200
appptr = 0
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0
read
EP1 control register
EPCTRL1(8004h)
appptr(Bit[7:6]) : 1
full(Bit[11]) : 0
• Pointer position
• Check full flag to check that write is possible
EP1 data buffer 1 area
EP1(8240h)
Data : (64Byte Data)
Write data in buffer
write
EP counter register (buffer 1)
EPCOUNT4(8090h)
appcnt(Bit[10:0]) : 40h
• Write the number of transfer data
write
Buf1
8240
Buf0
8200
appptr = 1
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
Buf1
Buf0
appptr=0
phyptr=0
read
EP1 control register
EPCTRL1(8004h)
appptr(Bit[7:6]) : 0
full(Bit11) : 1
• Data stored in buffer has not been processed. Therefore, the full flag is on and the next data cannot be written yet.

Figure 21-49



- Endpoint causes interrupt
- Check that IN buffer is ready
- Check the pointer position
- Check full flag to check that write is possible
- Clear the interrupt cause
- Endpoint causes interrupt
- Check empty_int of IN buffer
- Check the pointer position
- Check full flag to check that write is possible
- Clear the interrupt cause

Figure 21-50

Data IN transfer of multi-plane buffer [when FULL FLAG is on] (when dev_int_mode = 1)

Preconditions
- Interrupt type: dev_int_mode = 1
- Target EP: EP1
- Transfer type: Bulk
- Transfer direction: IN
- EP size: 64 bytes
- The number of buffer planes: 2 planes
- Buffer 0 address: 8200h
- Buffer 1 address: 8240h
- The number of transfer bytes: 192 bytes

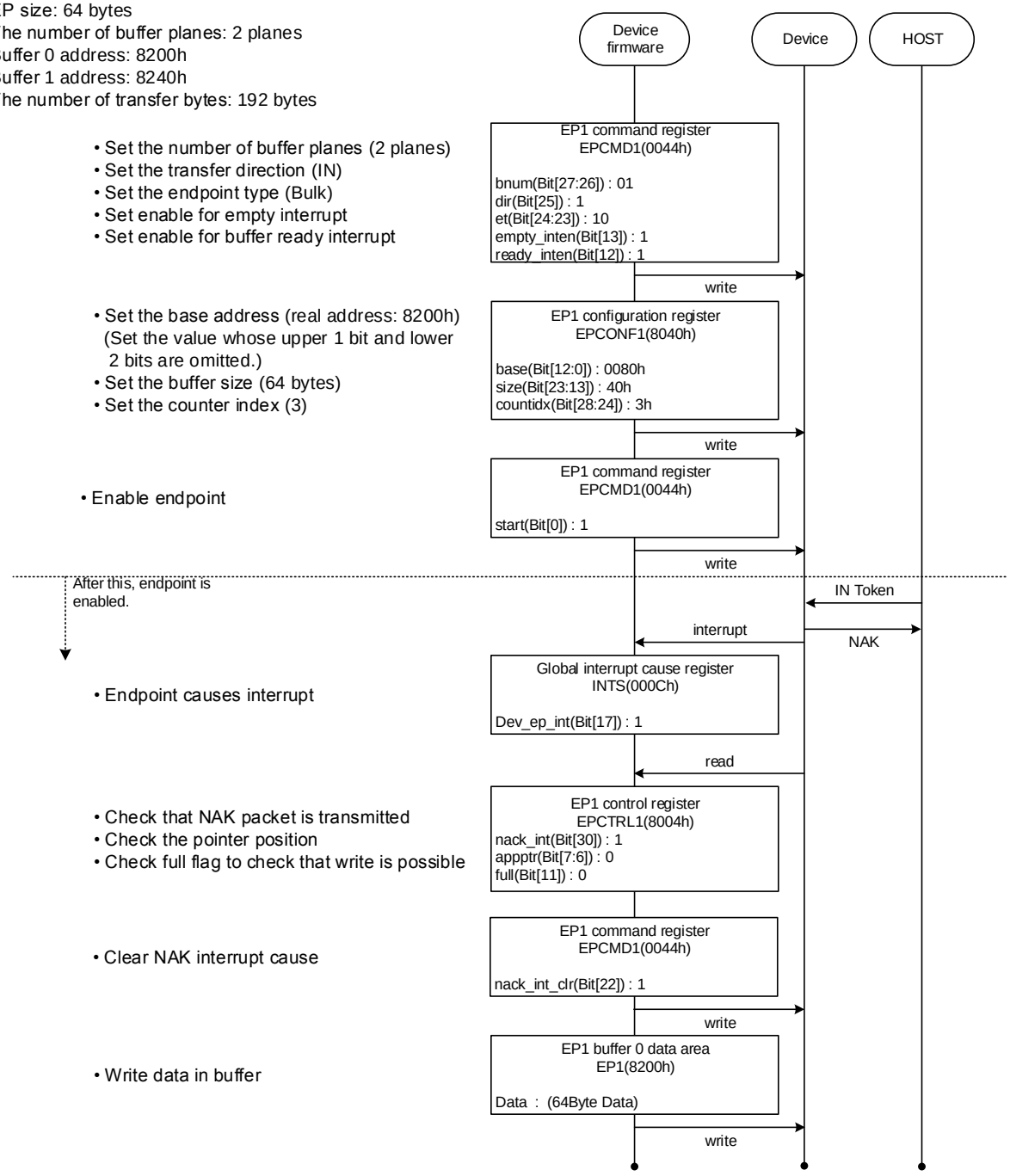


Figure 21-51

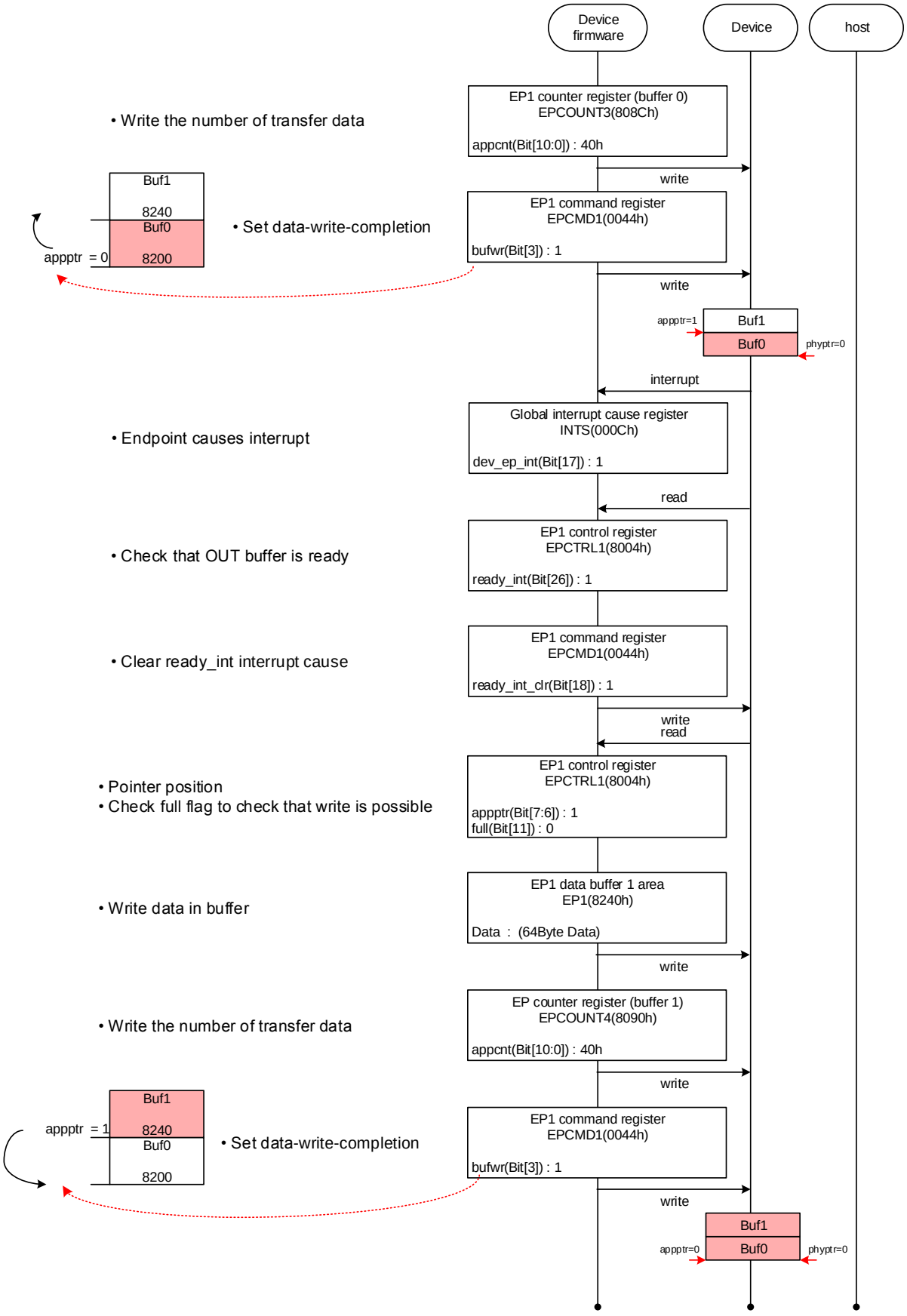
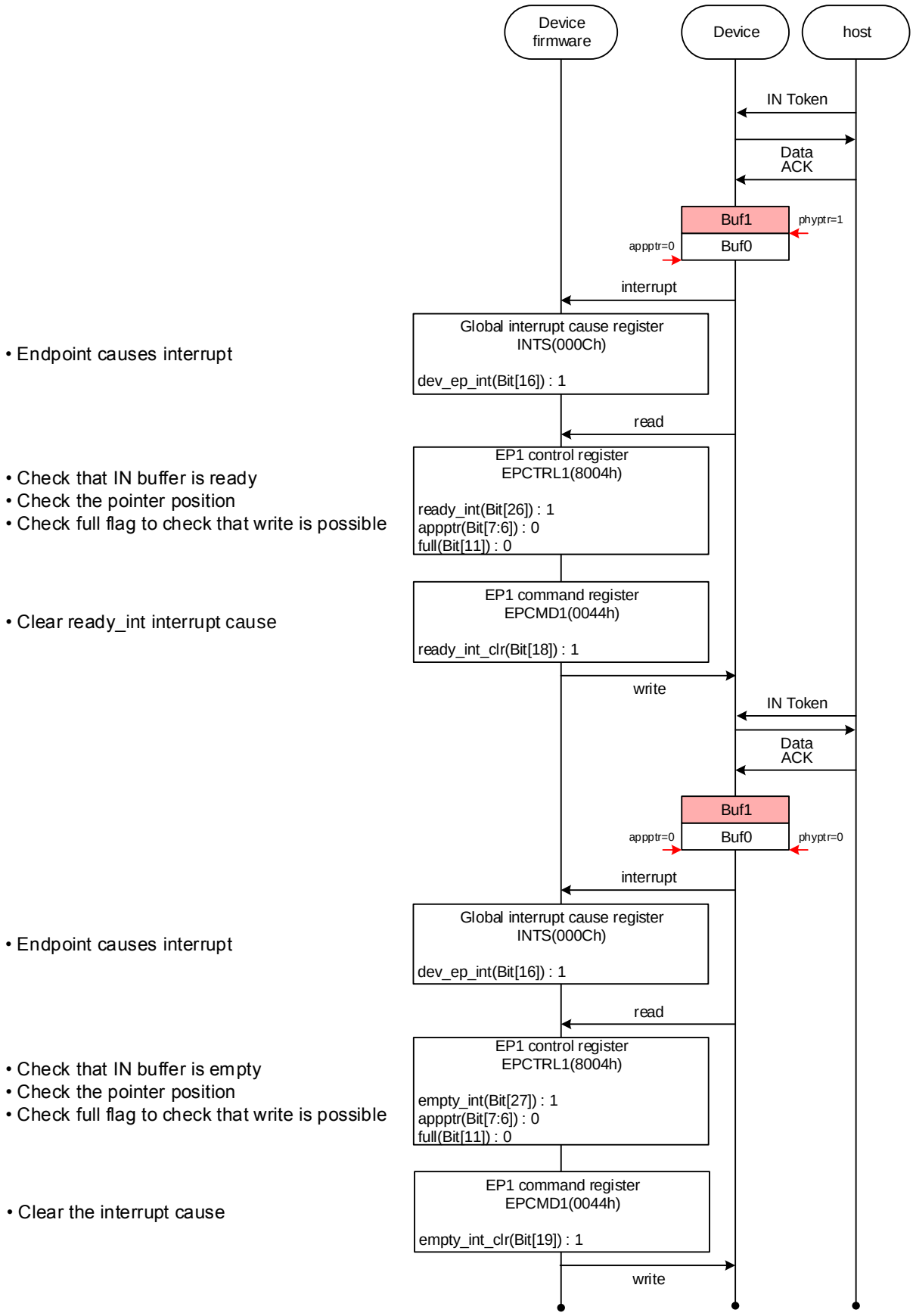

Device firmware
Device
host
• Write the number of transfer data
EP1 counter register (buffer 0)
EPCOUNT3(808Ch)
appcnt(Bit[10:0]) : 40h
write
Buf1
8240
Buf0
8200
appptr = 0
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
read
• Check that OUT buffer is ready
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
• Clear ready_int interrupt cause
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
write
read
• Pointer position
• Check full flag to check that write is possible
EP1 control register
EPCTRL1(8004h)
appptr(Bit[7:6]) : 1
full(Bit[11]) : 0
• Write data in buffer
EP1 data buffer 1 area
EP1(8240h)
Data : (64Byte Data)
write
• Write the number of transfer data
EP counter register (buffer 1)
EPCOUNT4(8090h)
appcnt(Bit[10:0]) : 40h
write
Buf1
8240
Buf0
8200
appptr = 1
• Set data-write-completion
EP1 command register
EPCMD1(0044h)
bufwr(Bit[3]) : 1
write
Buf1
Buf0
appptr=0
phyptr=0

Figure 21-52



- Endpoint causes interrupt
- Check that IN buffer is ready
- Check the pointer position
- Check full flag to check that write is possible
- Clear ready_int interrupt cause
- Endpoint causes interrupt
- Check that IN buffer is empty
- Check the pointer position
- Check full flag to check that write is possible
- Clear the interrupt cause

Figure 21-53

8.3.8. Data (OUT) transfer of multi-plane buffer

Process of interrupt and flag

There are 2 ways below in process of interrupt and flag at OUT transfer according to the setting of dev_int_mode (Bit[2]) of MODE register.

dev_int_mode = 0

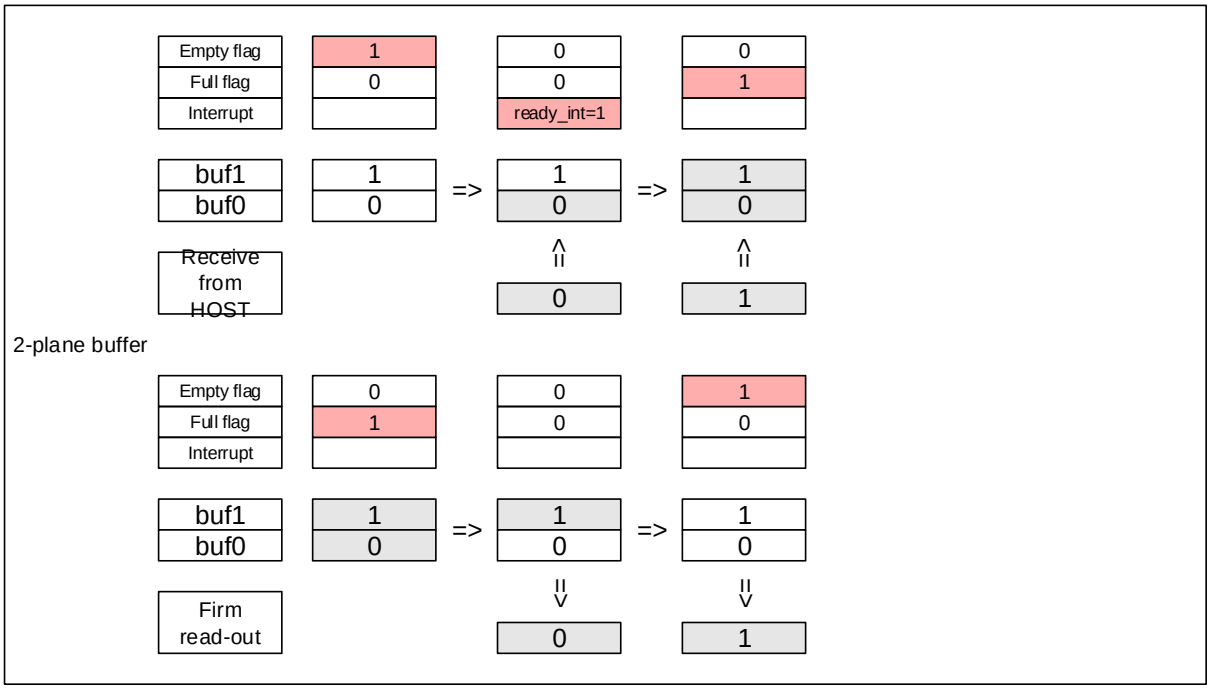


Figure 21-54

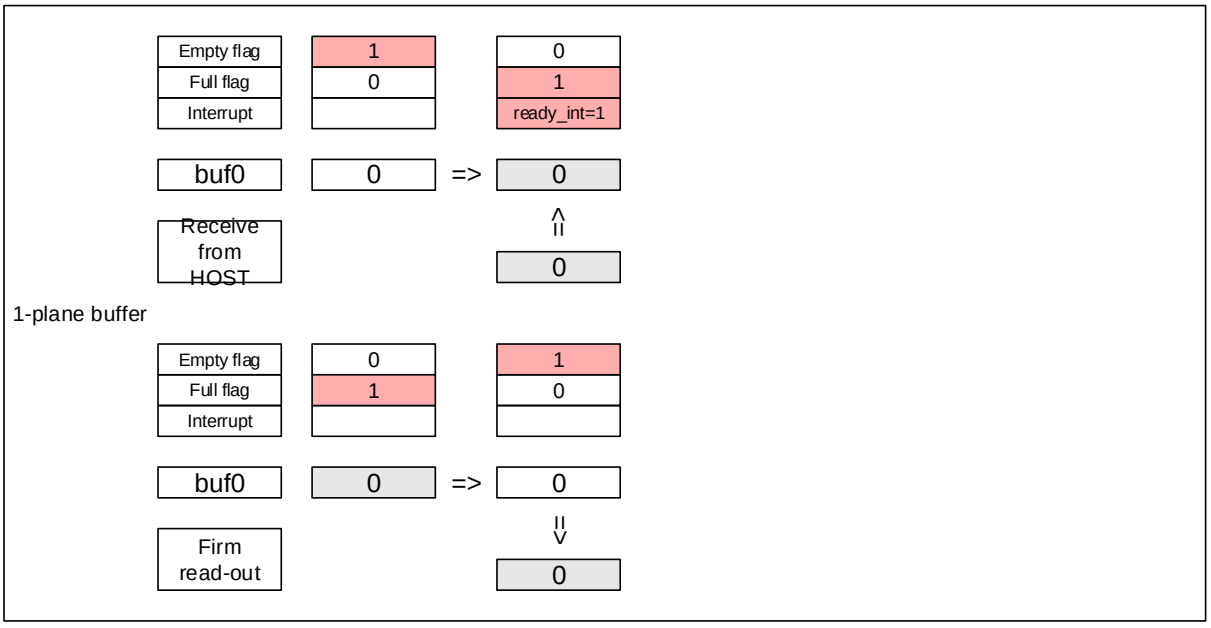


Figure 21-55

dev_int_mode = 1

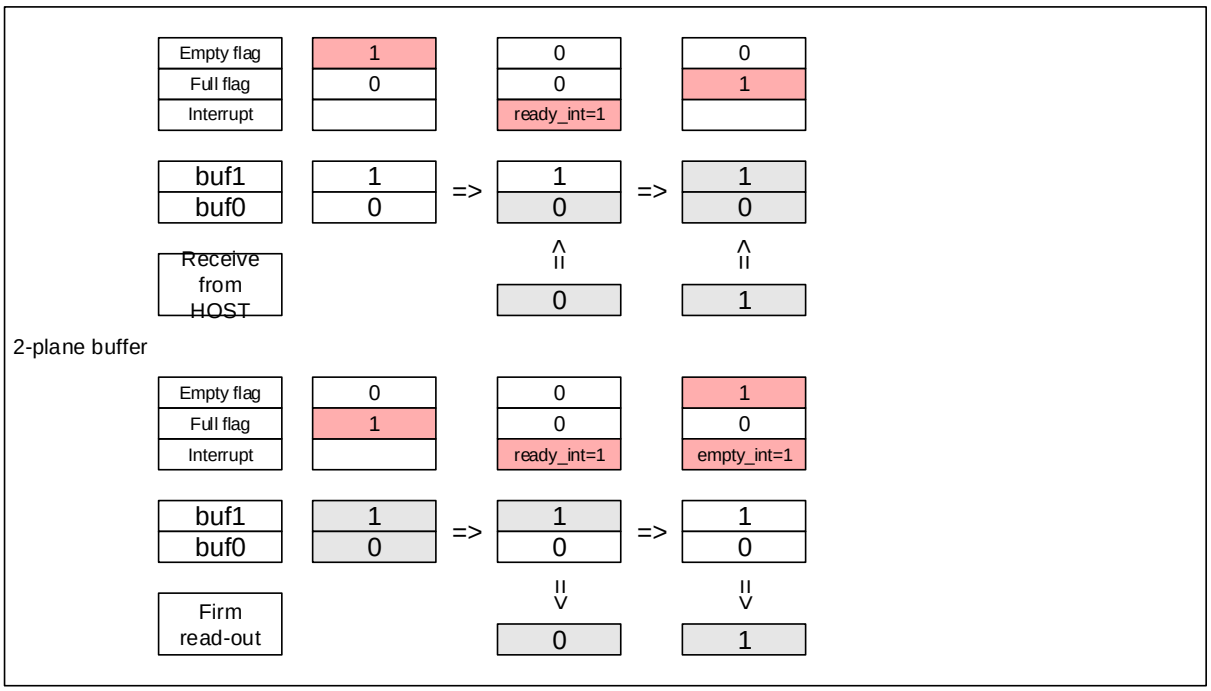


Figure 21-56

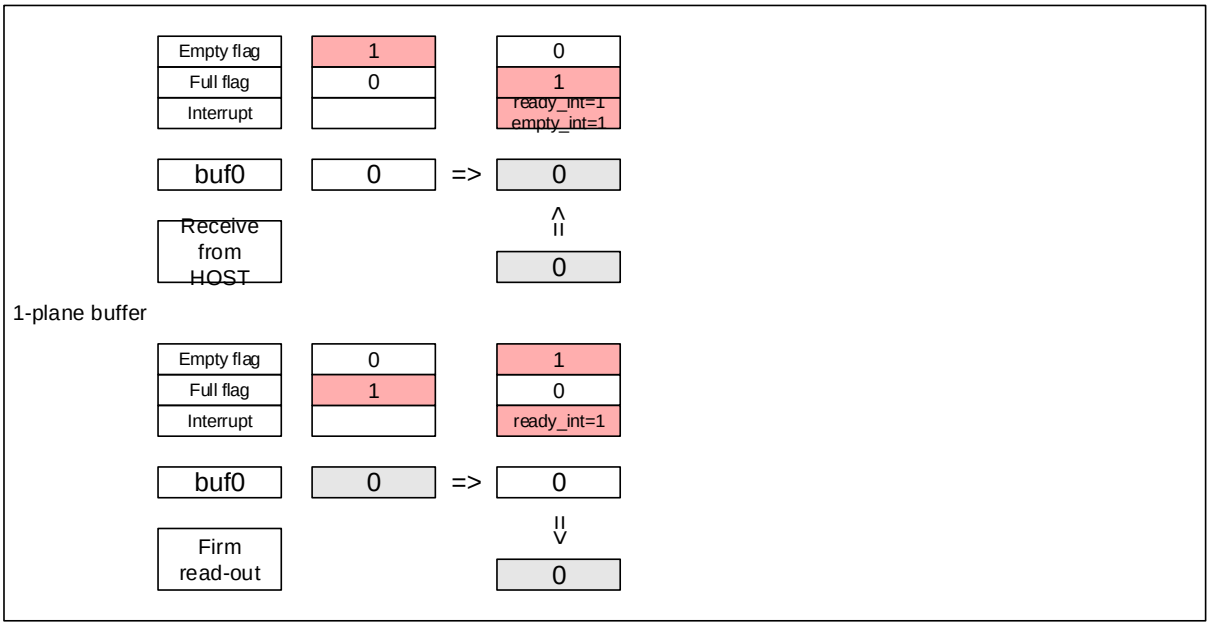


Figure 21-57

Data OUT transfer of multi-plane buffer (when dev_int_mode = 0)

Preconditions
- Interrupt type: dev_int_mode = 0
- Target EP: EP1
- Transfer type: Bulk
- Transfer direction: OUT
- EP size: 64 bytes
- The number of buffer planes: 2 planes
- Buffer 0 address: 8200h
- Buffer 1 address: 8240h
- The number of transfer bytes: 192 bytes

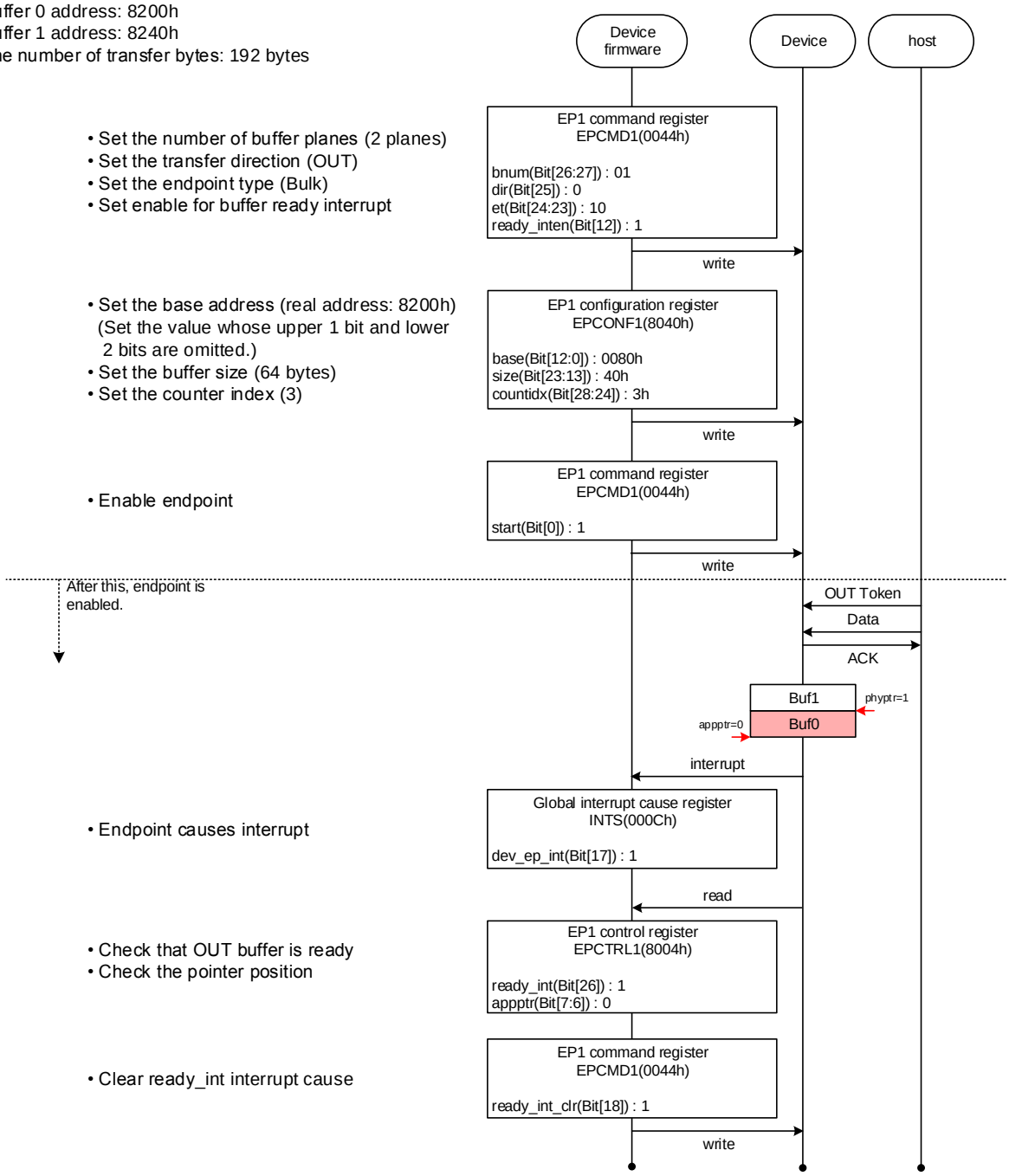


Figure 21-58

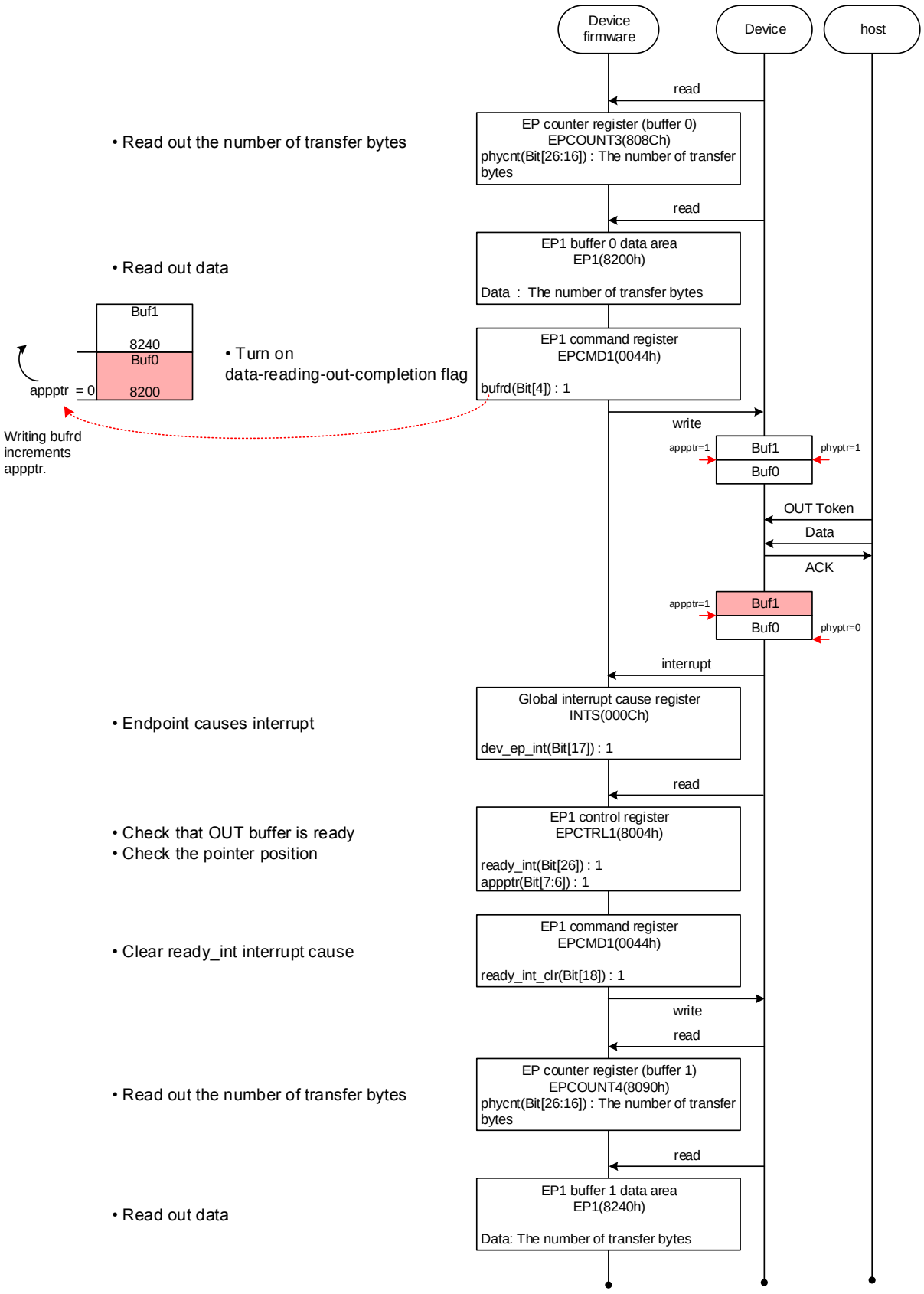

Device firmware
Device
host
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 0 data area
EP1(8200h)
Data : The number of transfer bytes
• Read out data
Buf1
8240
Buf0
8200
appptr = 0
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
Writing bufrd increments appptr.
write
appptr=1
Buf1
Buf0
phyptr=1
OUT Token
Data
ACK
appptr=1
Buf1
Buf0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 1
• Check that OUT buffer is ready
• Check the pointer position
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write
read
EP counter register (buffer 1)
EPCOUNT4(8090h)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 1 data area
EP1(8240h)
Data: The number of transfer bytes
• Read out data

Figure 21-59

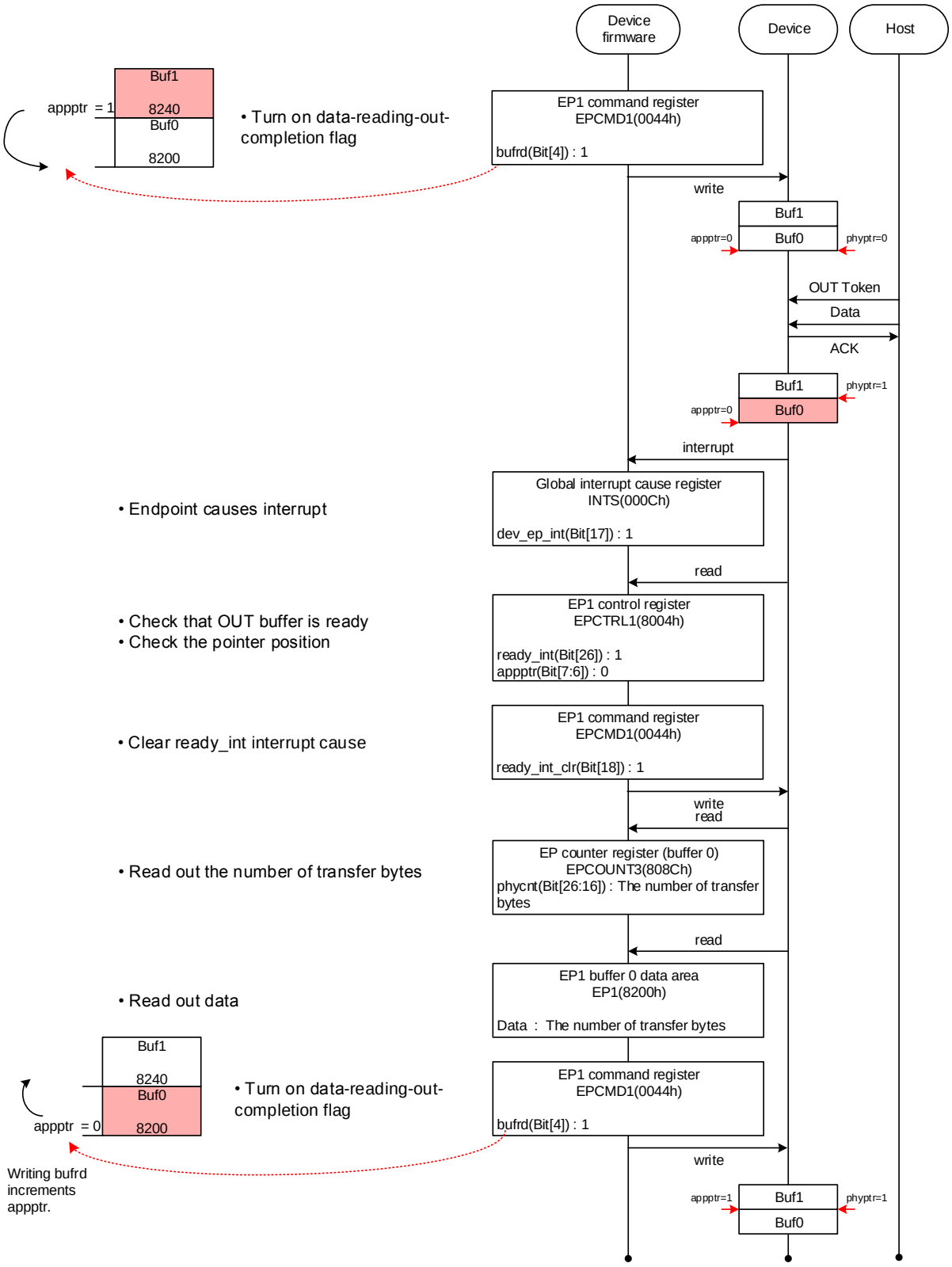

Device firmware
Device
Host
Buf1
appptr = 1
8240
Buf0
8200
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
Buf1
appptr=0
Buf0
phyptr=0
OUT Token
Data
ACK
Buf1
phyptr=1
appptr=0
Buf0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 0
• Check that OUT buffer is ready
• Check the pointer position
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 0 data area
EP1(8200h)
Data : The number of transfer bytes
• Read out data
Buf1
8240
Buf0
8200
appptr = 0
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
Writing bufrd increments appptr.
appptr=1
Buf1
phyptr=1
Buf0

Figure 21-60

Data OUT transfer of multi-plane buffer (when dev_int_mode = 1)

Preconditions
• Interrupt type: dev_int_mode = 1
• Target EP: EP1
• Transfer type: Bulk
• Transfer direction: OUT
• EP size: 64 bytes
• The number of buffer planes: 2 planes
• Buffer 0 address: 8200h
• Buffer 1 address: 8240h
• The number of transfer bytes: 192 bytes

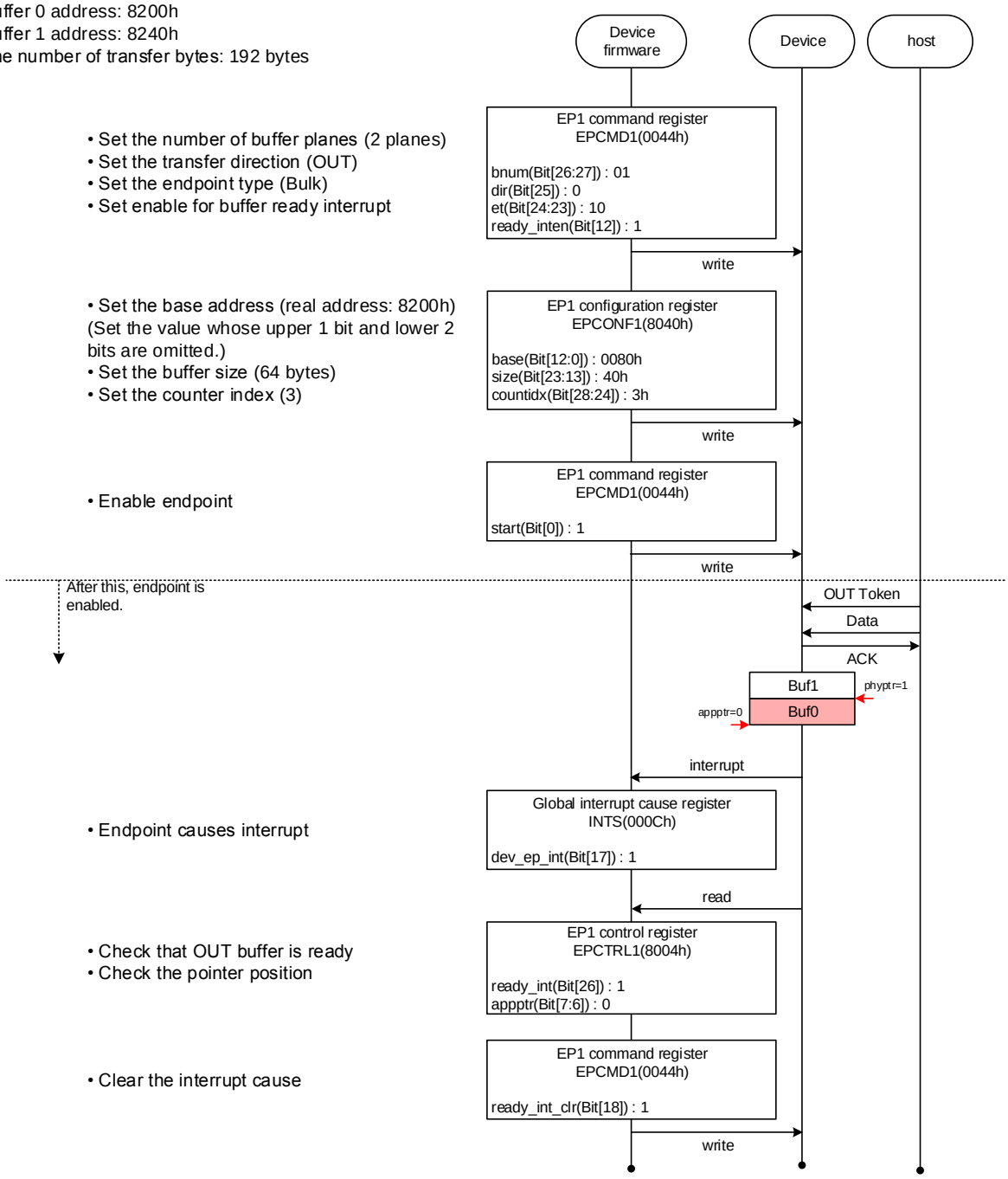


Figure 21-61

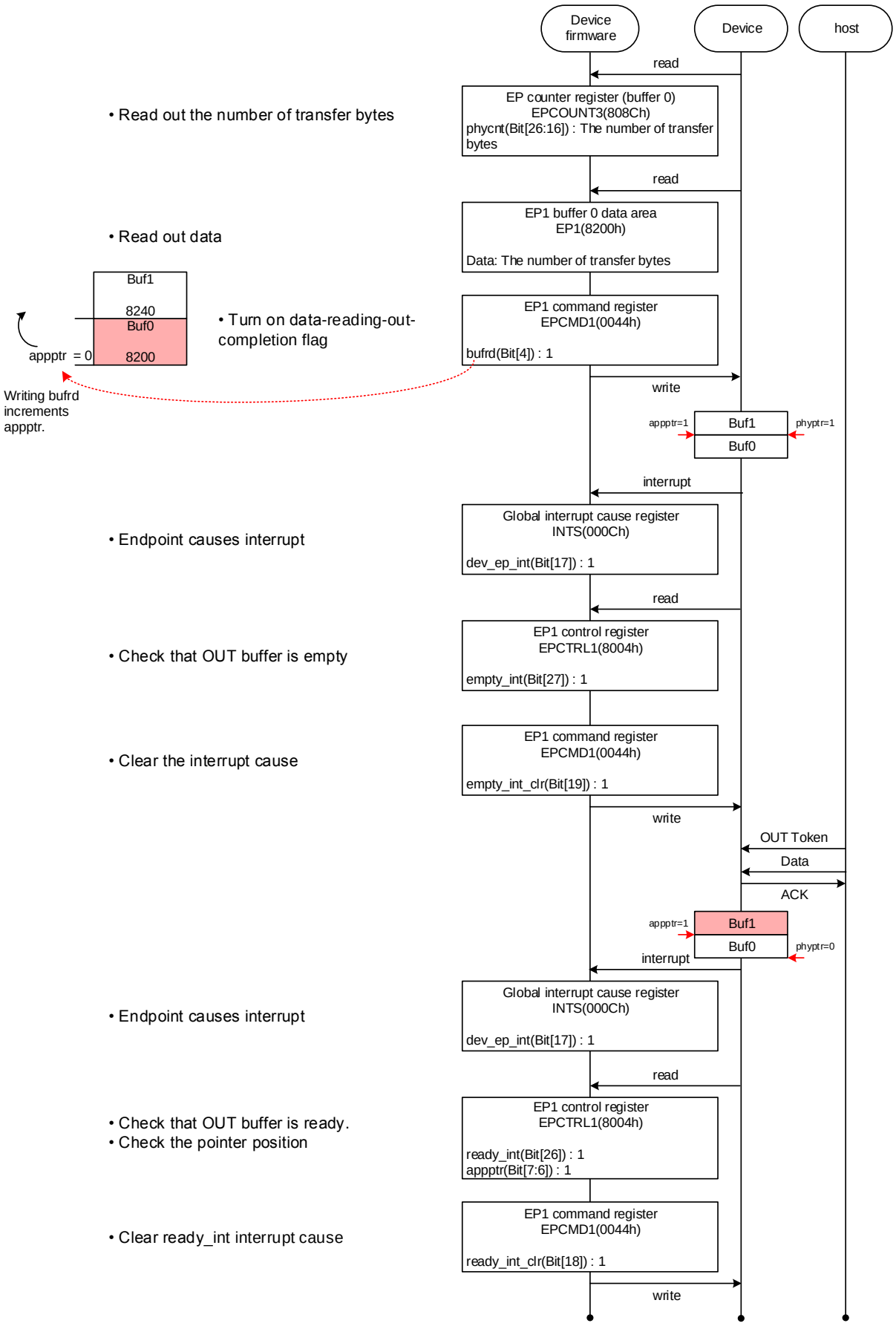

Device firmware
Device
host
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 0 data area
EP1(8200h)
Data: The number of transfer bytes
• Read out data
Buf1
8240
Buf0
8200
appptr = 0
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
Writing bufrd increments appptr.
appptr=1
Buf1
Buf0
phyptr=1
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) : 1
• Check that OUT buffer is empty
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write
OUT Token
Data
ACK
appptr=1
Buf1
Buf0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 1
• Check that OUT buffer is ready.
• Check the pointer position
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write

Figure 21-62

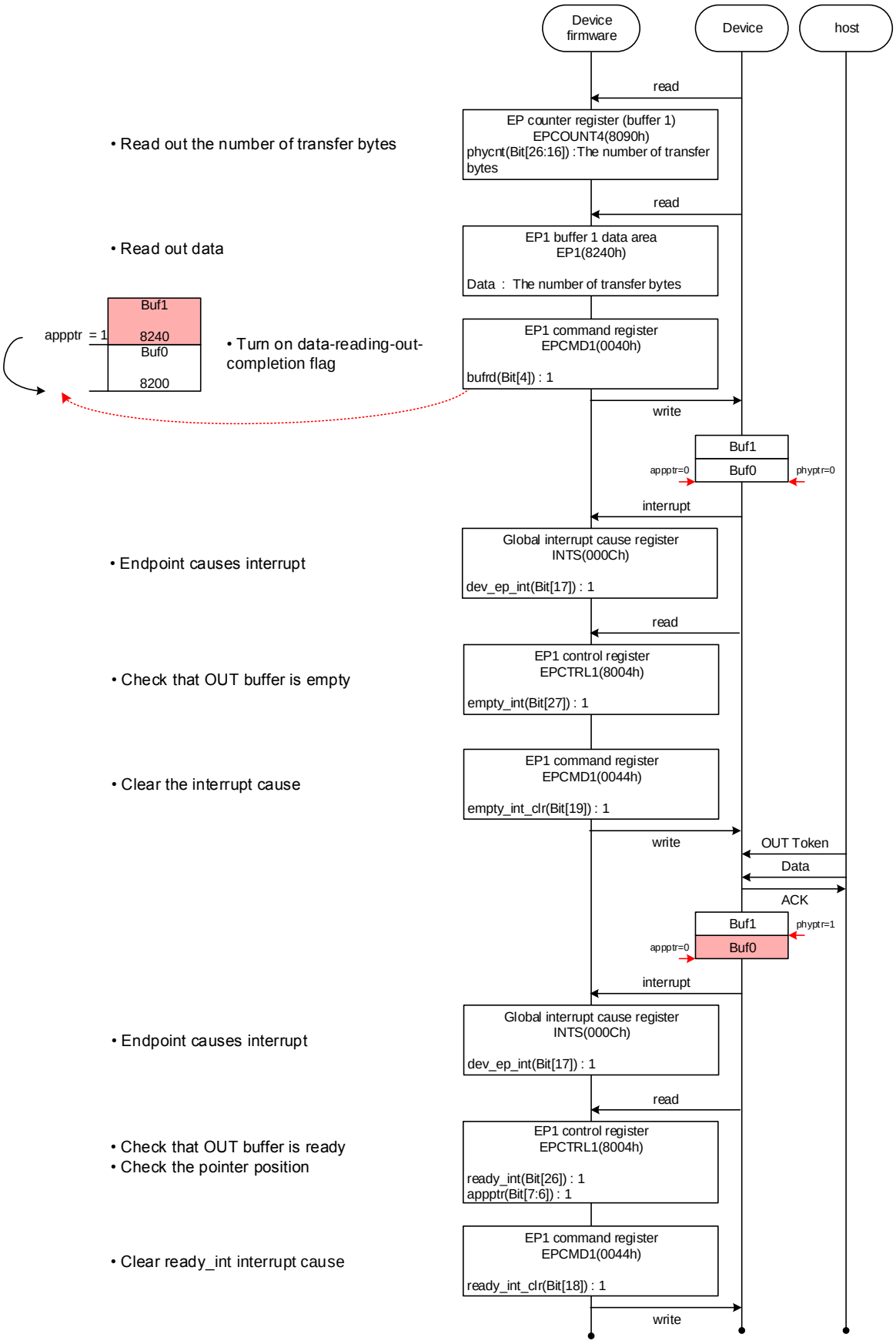

Device firmware
Device
host
read
EP counter register (buffer 1)
EPCOUNT4(8090h)
phycnt(Bit[26:16]) :The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 1 data area
EP1(8240h)
Data : The number of transfer bytes
• Read out data
Buf1
8240
appptr = 1
Buf0
8200
EP1 command register
EPCMD1(0040h)
bufrd(Bit[4]) : 1
• Turn on data-reading-out-completion flag
write
Buf1
Buf0
appptr=0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) : 1
• Check that OUT buffer is empty
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write
OUT Token
Data
ACK
Buf1
Buf0
phyptr=1
appptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 1
• Check that OUT buffer is ready
• Check the pointer position
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write

Figure 21-63

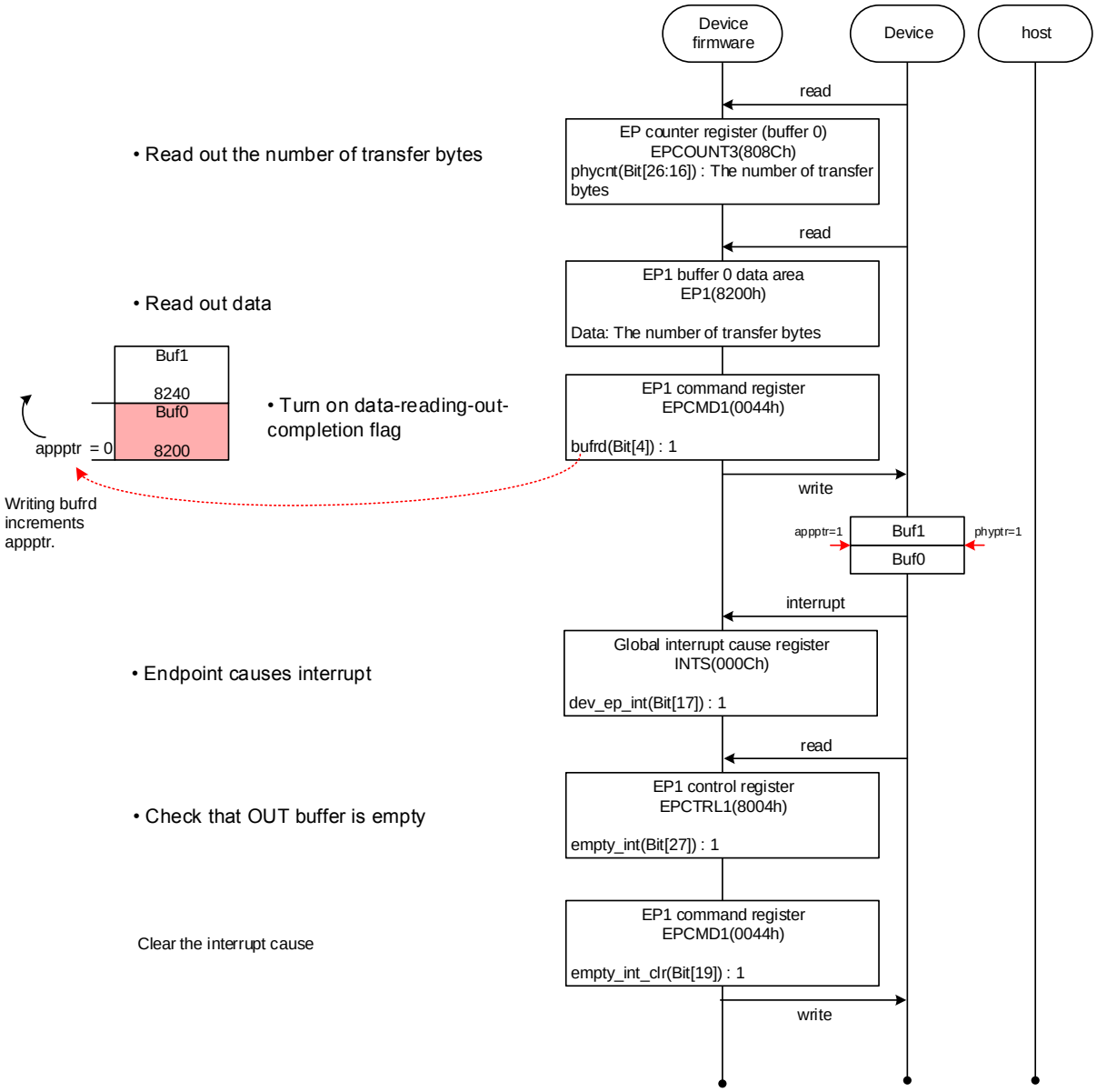

Device firmware
Device
host
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 0 data area
EP1(8200h)
Data: The number of transfer bytes
• Read out data
Buf1
8240
Buf0
8200
appptr = 0
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
Writing bufrd increments appptr.
appptr=1
Buf1
Buf0
phyptr=1
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) : 1
• Check that OUT buffer is empty
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
Clear the interrupt cause
write

Figure 21-64

Data OUT transfer of multi-plane buffer [when FULL FLAG is on] (when dev_int_mode = 0)

Preconditions
- Interrupt type: dev_int_mode = 0
- Target EP: EP1
- Transfer type: Bulk
- Transfer direction: OUT
- EP size: 64 bytes
- The number of buffer planes: 2 planes
- Buffer 0 address: 8200h
- Buffer 1 address: 8240h
- The number of transfer bytes: 192 bytes

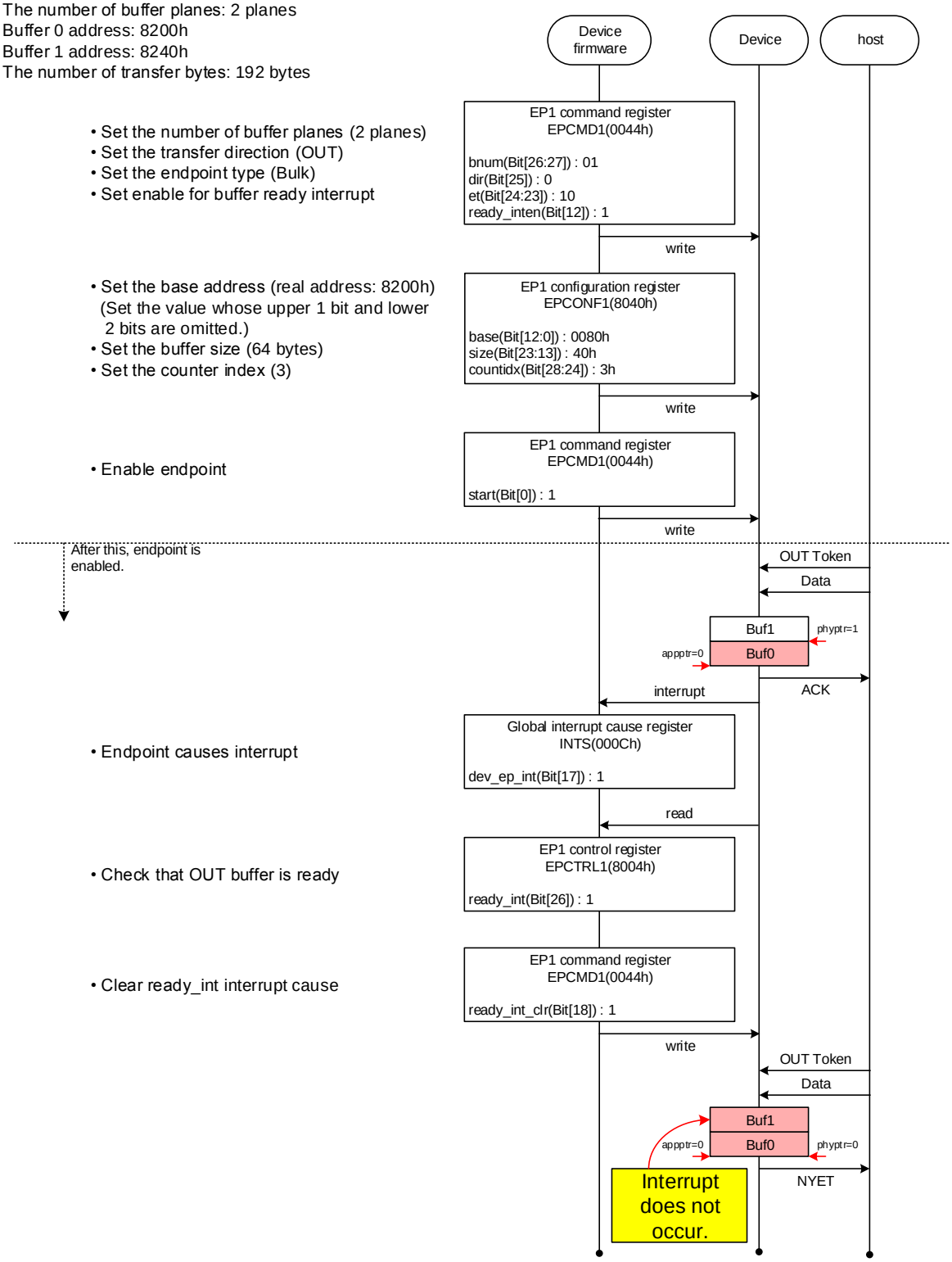


Figure 21-65

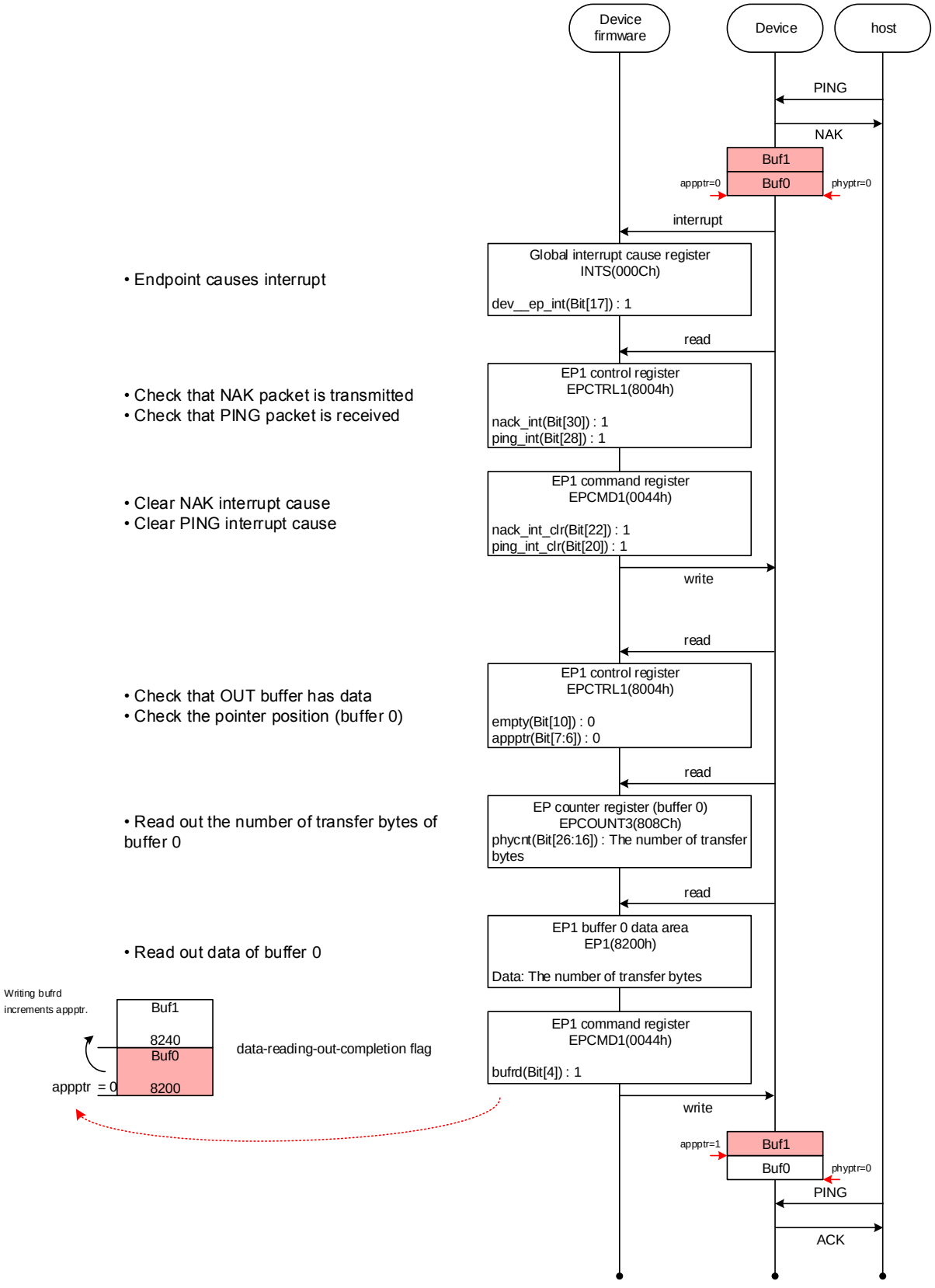

Device firmware
Device
host
PING
NAK
Buf1
Buf0
appptr=0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev__ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
nack_int(Bit[30]) : 1
ping_int(Bit[28]) : 1
• Check that NAK packet is transmitted
• Check that PING packet is received
EP1 command register
EPCMD1(0044h)
nack_int_clr(Bit[22]) : 1
ping_int_clr(Bit[20]) : 1
• Clear NAK interrupt cause
• Clear PING interrupt cause
write
read
EP1 control register
EPCTRL1(8004h)
empty(Bit[10]) : 0
appptr(Bit[7:6]) : 0
• Check that OUT buffer has data
• Check the pointer position (buffer 0)
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes of buffer 0
read
EP1 buffer 0 data area
EP1(8200h)
Data: The number of transfer bytes
• Read out data of buffer 0
Writing bufrd increments appptr.
Buf1
8240
Buf0
8200
appptr = 0
data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
appptr=1
Buf1
Buf0
phyptr=0
PING
ACK

Figure 21-66

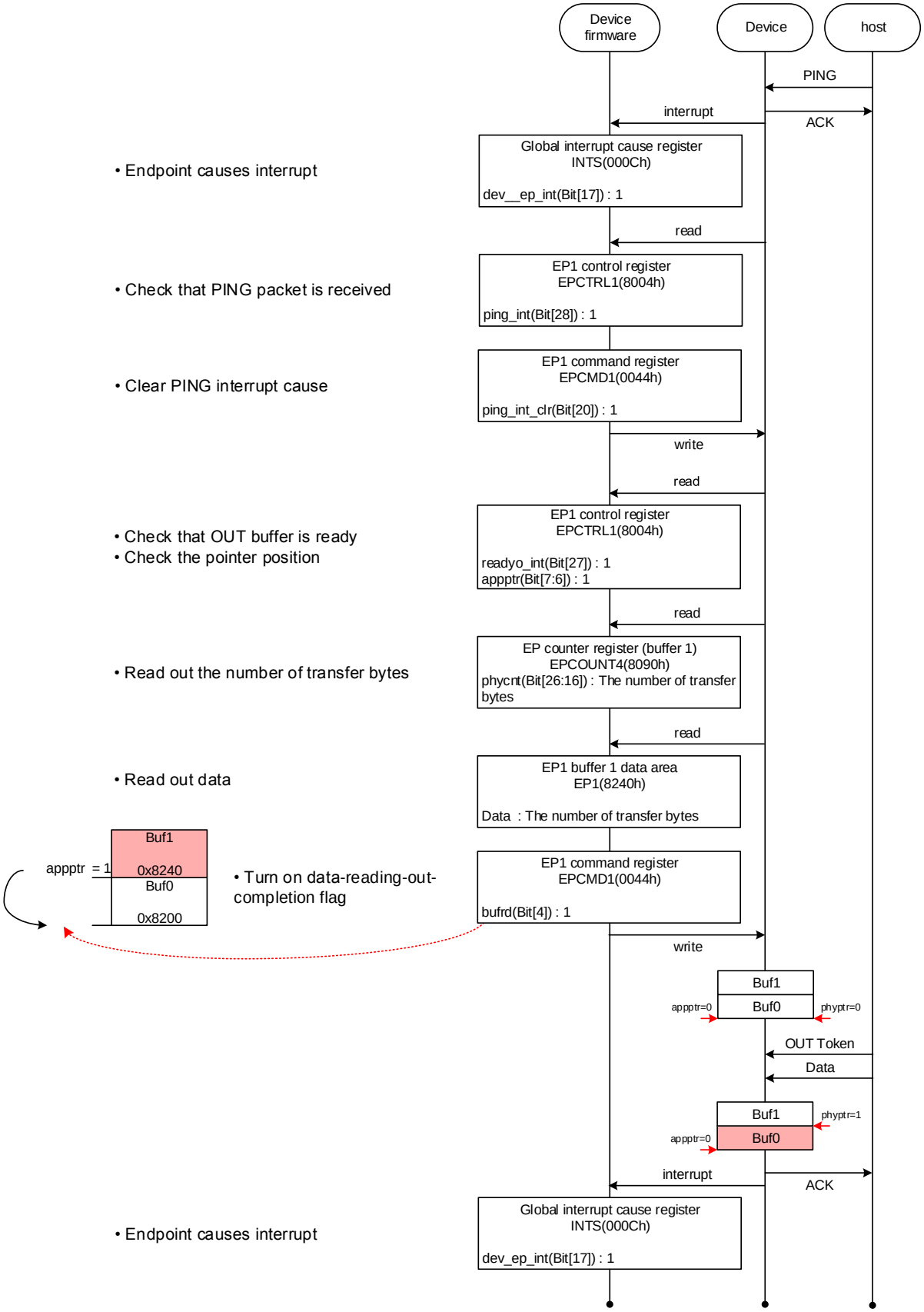

Device firmware
Device
host
PING
interrupt
ACK
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev__ep_int(Bit[17]) : 1
read
• Check that PING packet is received
EP1 control register
EPCTRL1(8004h)
ping_int(Bit[28]) : 1
• Clear PING interrupt cause
EP1 command register
EPCMD1(0044h)
ping_int_clr(Bit[20]) : 1
write
read
• Check that OUT buffer is ready
• Check the pointer position
EP1 control register
EPCTRL1(8004h)
readyo_int(Bit[27]) : 1
appptr(Bit[7:6]) : 1
read
• Read out the number of transfer bytes
EP counter register (buffer 1)
EPCOUNT4(8090h)
phycnt(Bit[26:16]) : The number of transfer bytes
read
• Read out data
EP1 buffer 1 data area
EP1(8240h)
Data : The number of transfer bytes
Buf1
appptr = 1
0x8240
Buf0
0x8200
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
Buf1
Buf0
appptr=0
phyptr=0
OUT Token
Data
Buf1
Buf0
phyptr=1
appptr=0
interrupt
ACK
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1

Figure 21-67

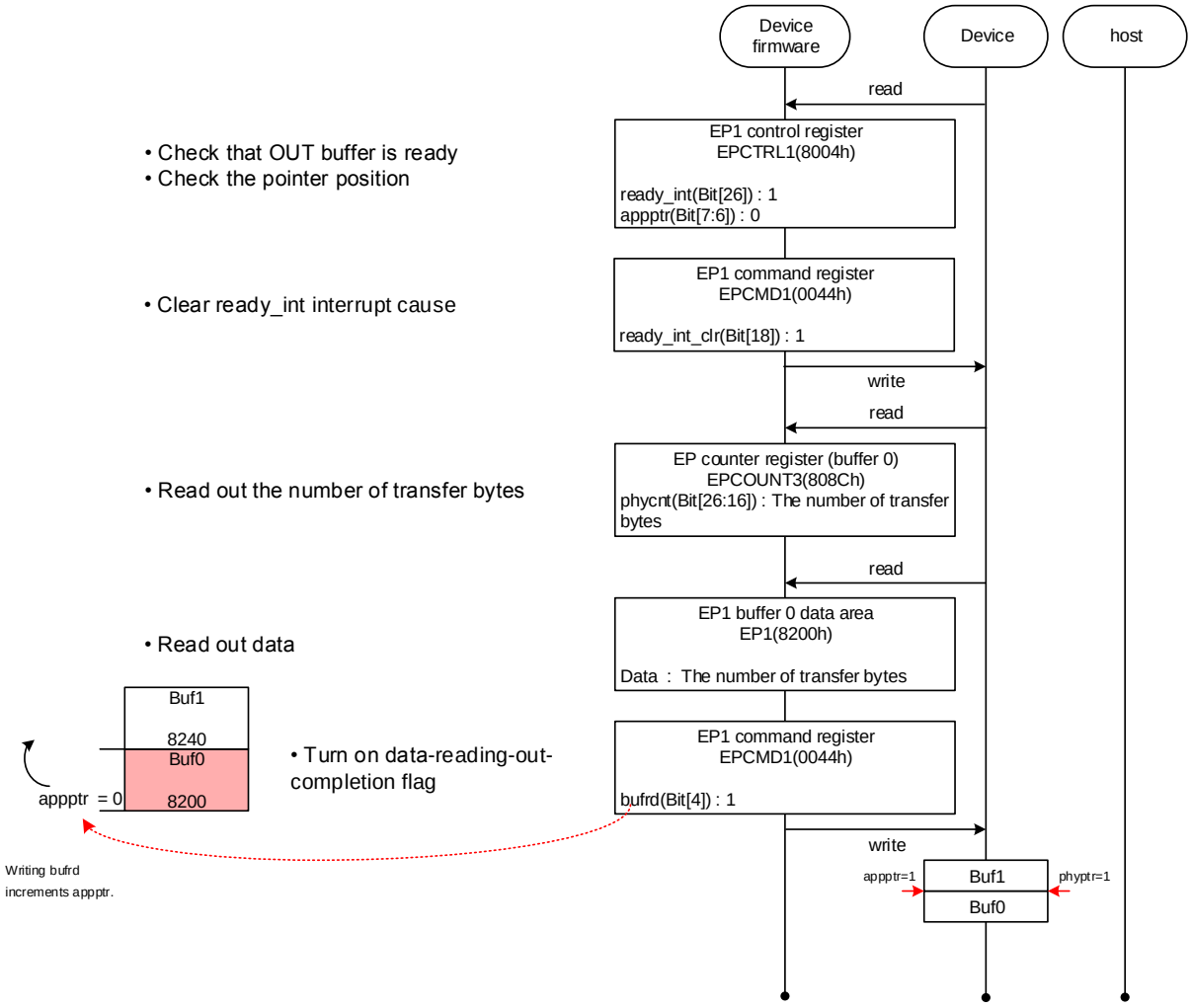

Device firmware
Device
host
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 0
• Check that OUT buffer is ready
• Check the pointer position
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 0 data area
EP1(8200h)
Data : The number of transfer bytes
• Read out data
Buf1
8240
Buf0
8200
appptr = 0
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
appptr=1
Buf1
Buf0
phyptr=1
Writing bufrd increments appptr.

Figure 21-68

Data OUT transfer of multi-plane buffer [when FULL FLAG is on] (when dev_int_mode = 1)

Preconditions
- Interrupt type: dev_int_mode = 1
- Target EP: EP1
- Transfer type: Bulk
- Transfer direction: OUT
- EP size: 64 bytes
- The number of buffer planes: 2 planes
- Buffer 0 address: 8200h
- Buffer 1 address: 8240h
- The number of transfer bytes: 192 bytes

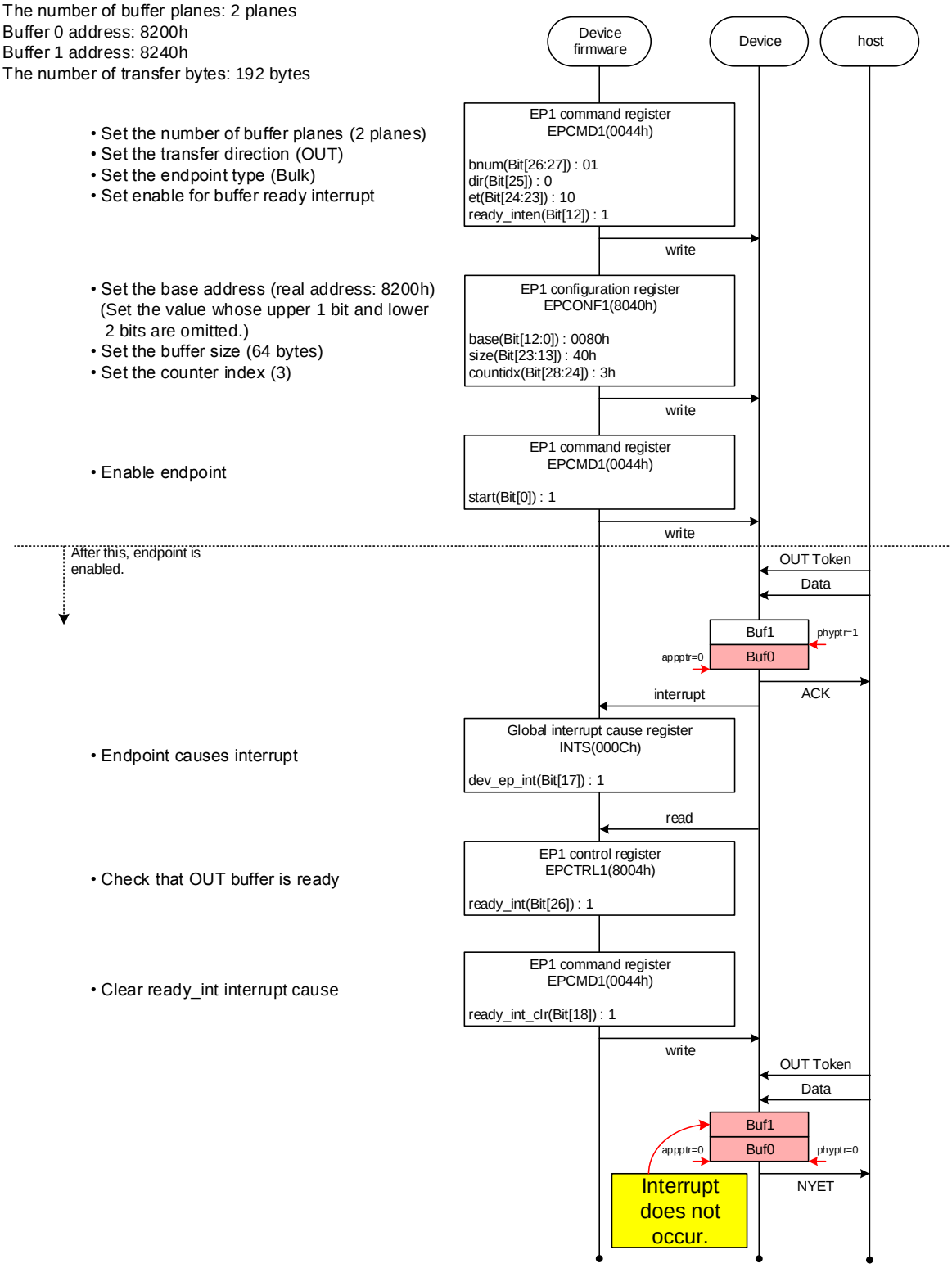


Figure 21-69

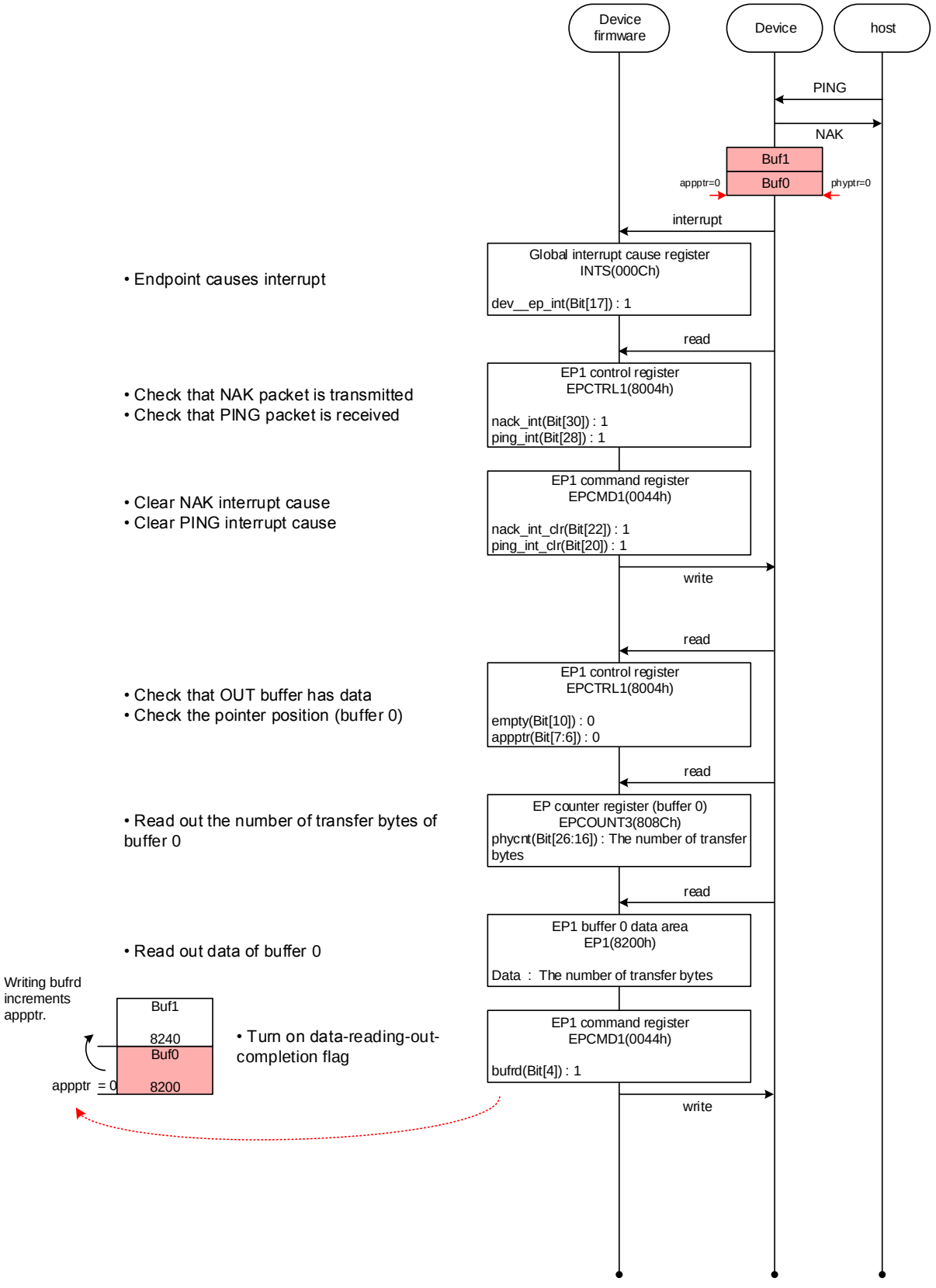

Device firmware
Device
host
PING
NAK
Buf1
Buf0
appptr=0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev__ep_int(Bit[17]) : 1
read
• Endpoint causes interrupt
EP1 control register
EPCTRL1(8004h)
nack_int(Bit[30]) : 1
ping_int(Bit[28]) : 1
• Check that NAK packet is transmitted
• Check that PING packet is received
EP1 command register
EPCMD1(0044h)
nack_int_clr(Bit[22]) : 1
ping_int_clr(Bit[20]) : 1
• Clear NAK interrupt cause
• Clear PING interrupt cause
write
read
EP1 control register
EPCTRL1(8004h)
empty(Bit[10]) : 0
appptr(Bit[7:6]) : 0
• Check that OUT buffer has data
• Check the pointer position (buffer 0)
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes of buffer 0
read
EP1 buffer 0 data area
EP1(8200h)
Data : The number of transfer bytes
• Read out data of buffer 0
Writing bufrd increments appptr.
Buf1
8240
Buf0
8200
appptr = 0
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
• Turn on data-reading-out-completion flag
write

Figure 21-70

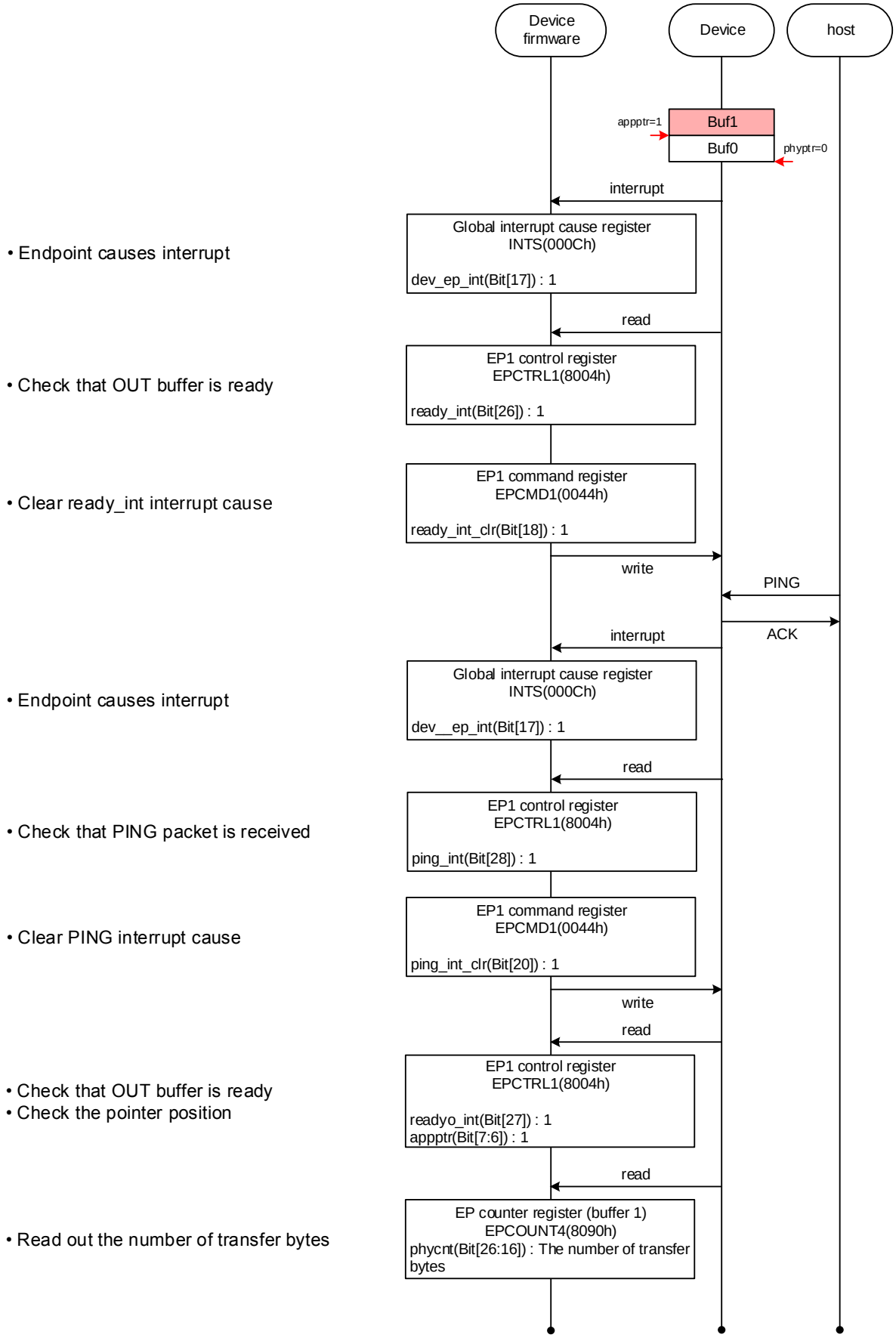

Device firmware
Device
host
appptr=1
Buf1
Buf0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
• Check that OUT buffer is ready
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write
PING
ACK
interrupt
Global interrupt cause register
INTS(000Ch)
dev__ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ping_int(Bit[28]) : 1
• Check that PING packet is received
EP1 command register
EPCMD1(0044h)
ping_int_clr(Bit[20]) : 1
• Clear PING interrupt cause
write
read
EP1 control register
EPCTRL1(8004h)
readyo_int(Bit[27]) : 1
appptr(Bit[7:6]) : 1
• Check that OUT buffer is ready
• Check the pointer position
read
EP counter register (buffer 1)
EPCOUNT4(8090h)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes

Figure 21-71

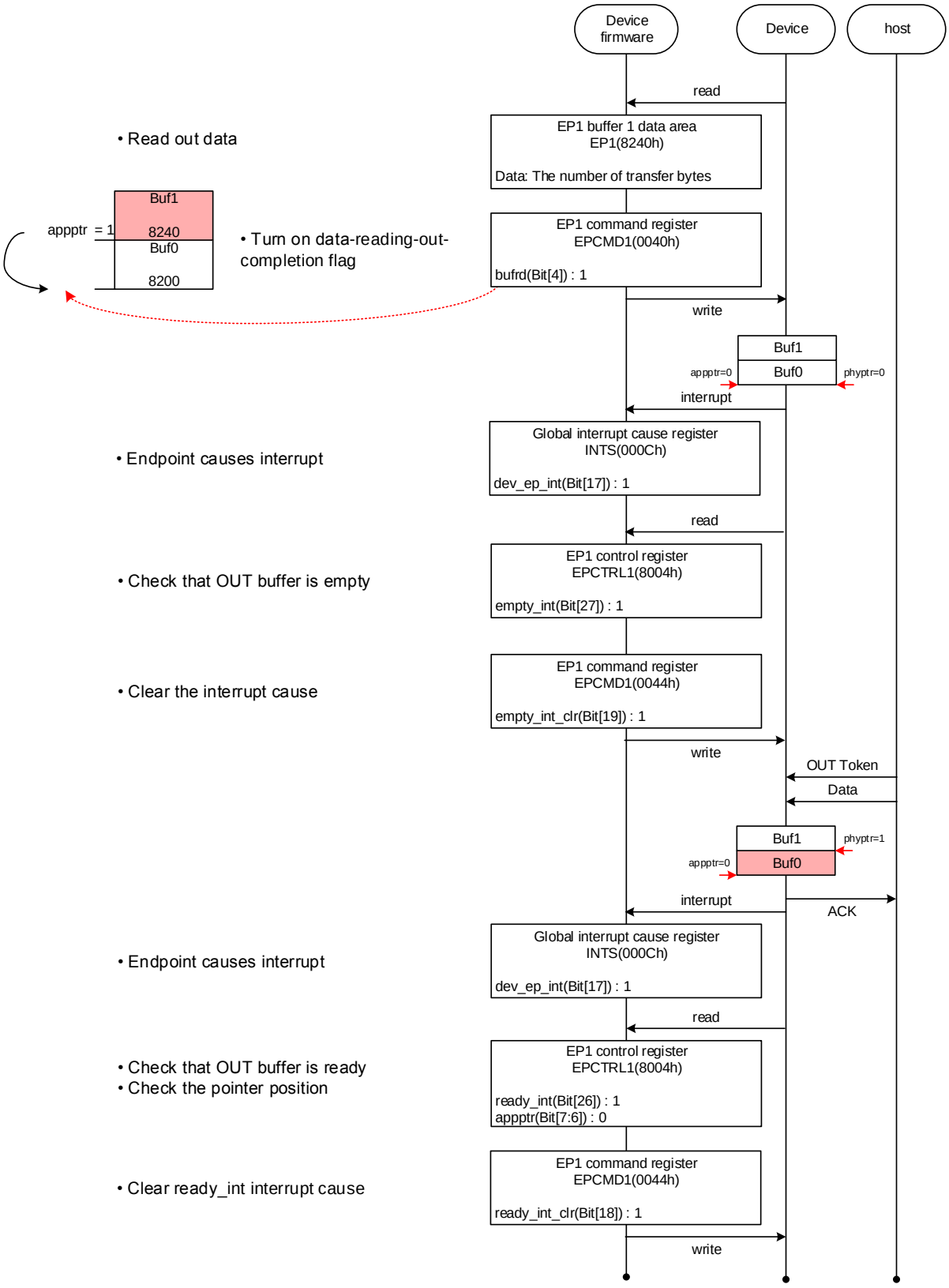

Device firmware
Device
host
read
EP1 buffer 1 data area
EP1(8240h)
Data: The number of transfer bytes
• Read out data
Buf1
appptr = 1
8240
Buf0
8200
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0040h)
bufrd(Bit[4]) : 1
write
Buf1
Buf0
appptr=0
phyptr=0
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) : 1
• Check that OUT buffer is empty
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write
OUT Token
Data
Buf1
Buf0
appptr=0
phyptr=1
interrupt
ACK
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 0
• Check that OUT buffer is ready
• Check the pointer position
EP1 command register
EPCMD1(0044h)
ready_int_clr(Bit[18]) : 1
• Clear ready_int interrupt cause
write

Figure 21-72

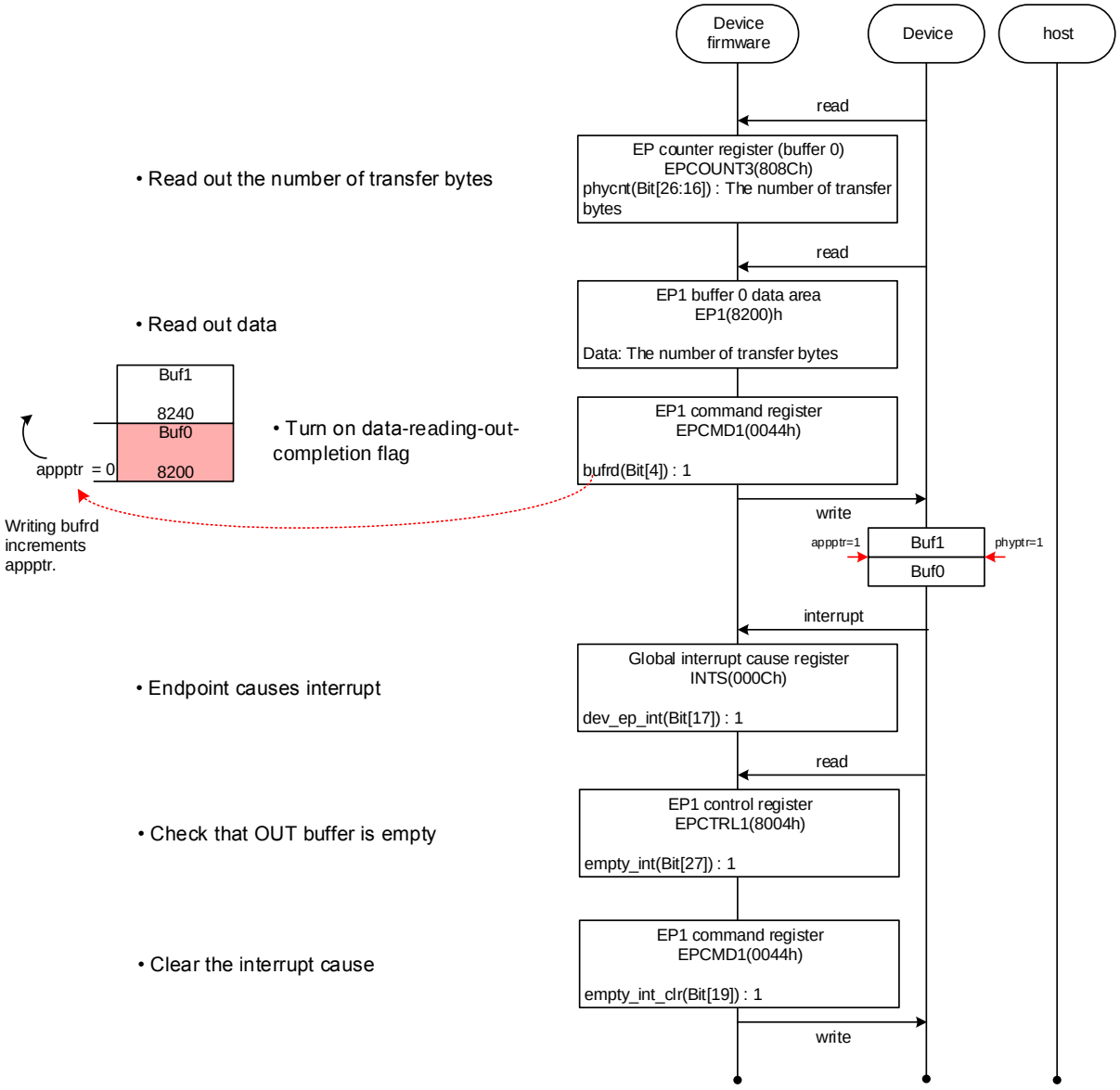

Device firmware
Device
host
read
EP counter register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
• Read out the number of transfer bytes
read
EP1 buffer 0 data area
EP1(8200)h
Data: The number of transfer bytes
• Read out data
Buf1
8240
Buf0
8200
appptr = 0
• Turn on data-reading-out-completion flag
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
appptr=1
Buf1
Buf0
phyptr=1
Writing bufrd increments appptr.
interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
• Endpoint causes interrupt
read
EP1 control register
EPCTRL1(8004h)
empty_int(Bit[27]) : 1
• Check that OUT buffer is empty
EP1 command register
EPCMD1(0044h)
empty_int_clr(Bit[19]) : 1
• Clear the interrupt cause
write

Figure 21-73

8.3.9. Standard command response process

SET ADDRESS

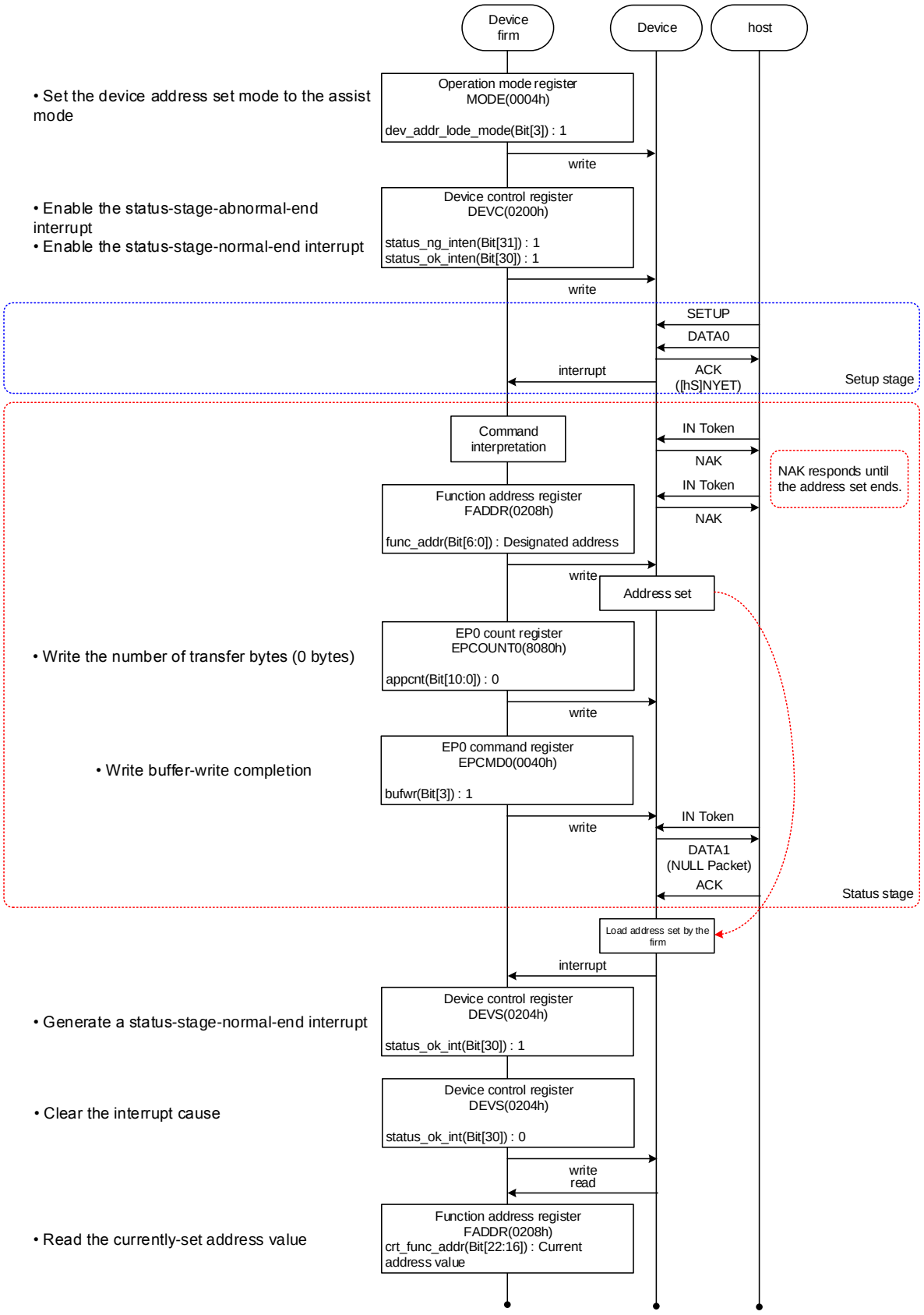


Figure 21-74

SET CONFIGURATION

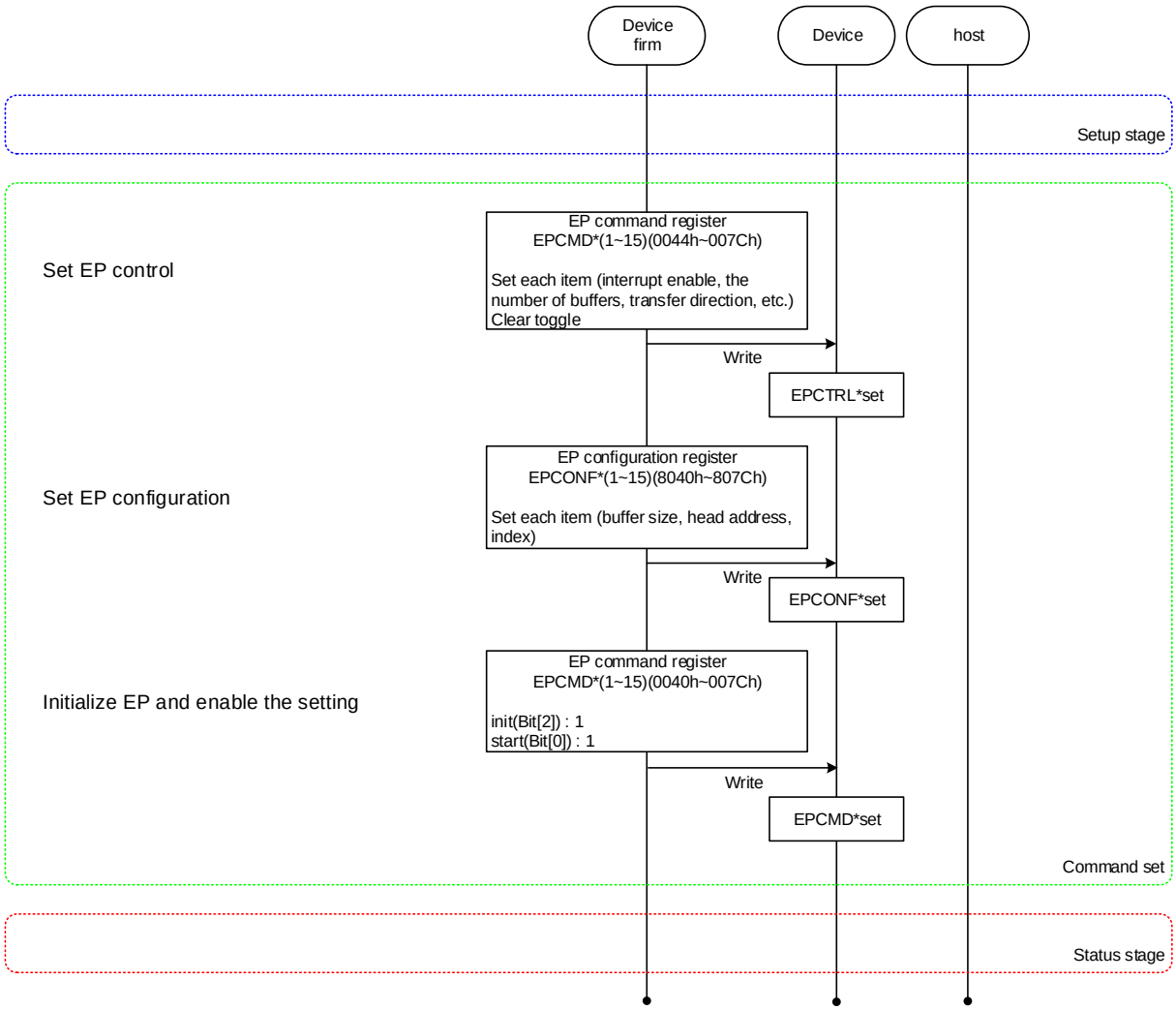


Figure 21-75

SET INTERFACE

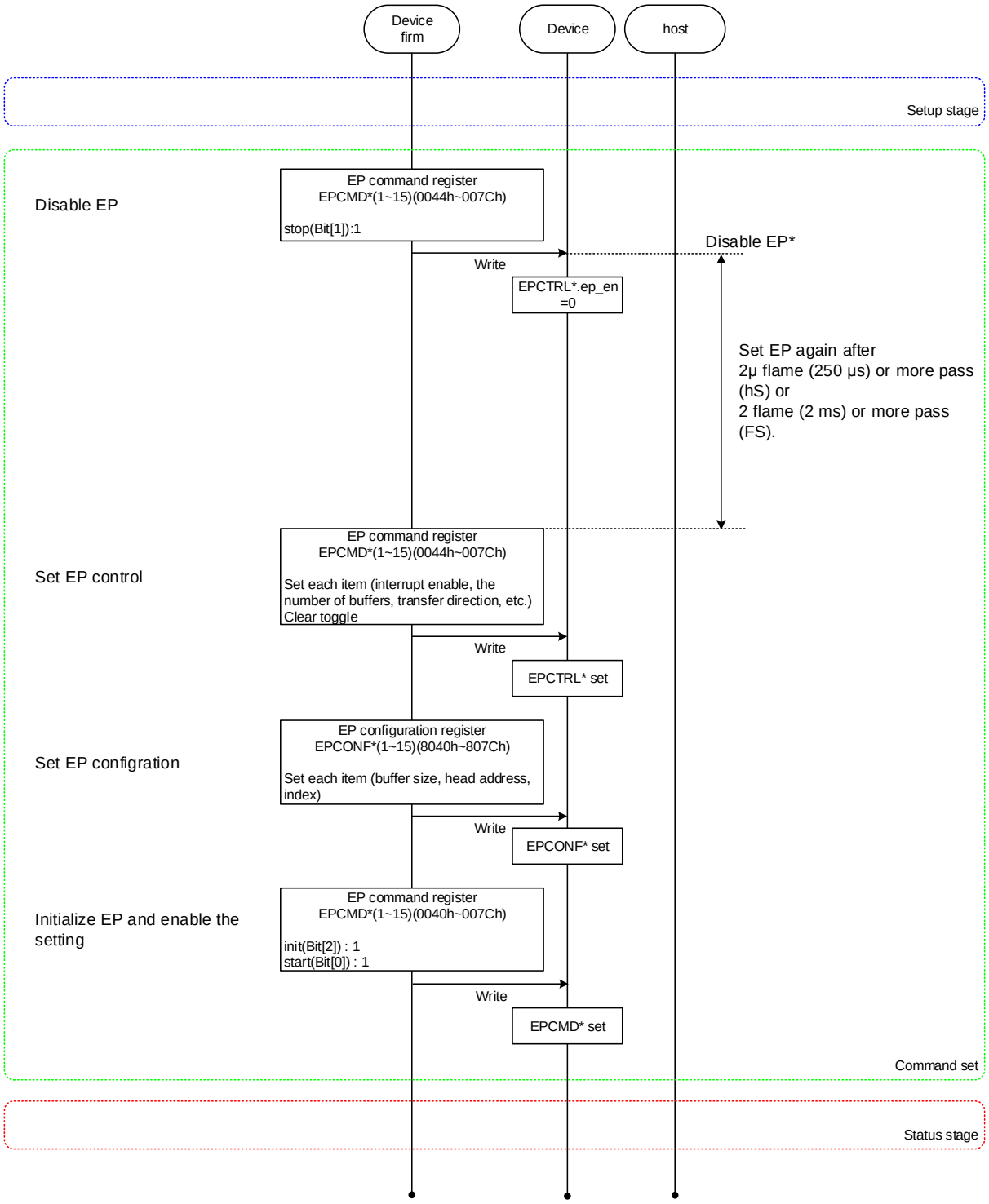


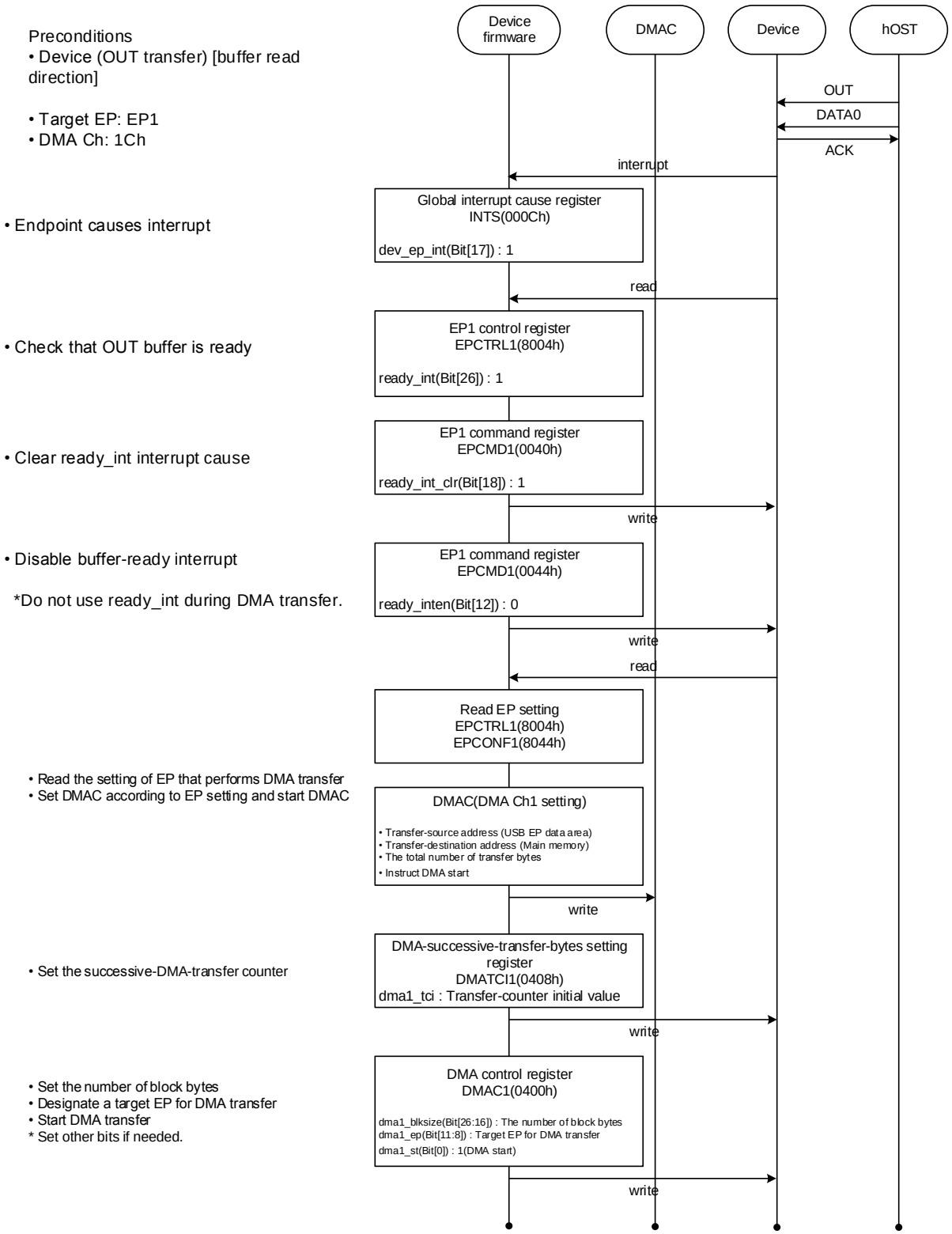
Figure 21-76

8.3.10. DMA transfer

DMA READ (OUT) transfer

Preconditions
• Device (OUT transfer) [buffer read direction]

• Target EP: EP1
• DMA Ch: 1Ch



• Endpoint causes interrupt

• Check that OUT buffer is ready

• Clear ready_int interrupt cause

• Disable buffer-ready interrupt

*Do not use ready_int during DMA transfer.

• Read the setting of EP that performs DMA transfer
• Set DMAC according to EP setting and start DMAC

• Set the successive-DMA-transfer counter

• Set the number of block bytes
• Designate a target EP for DMA transfer
• Start DMA transfer
* Set other bits if needed.

Figure 21-77

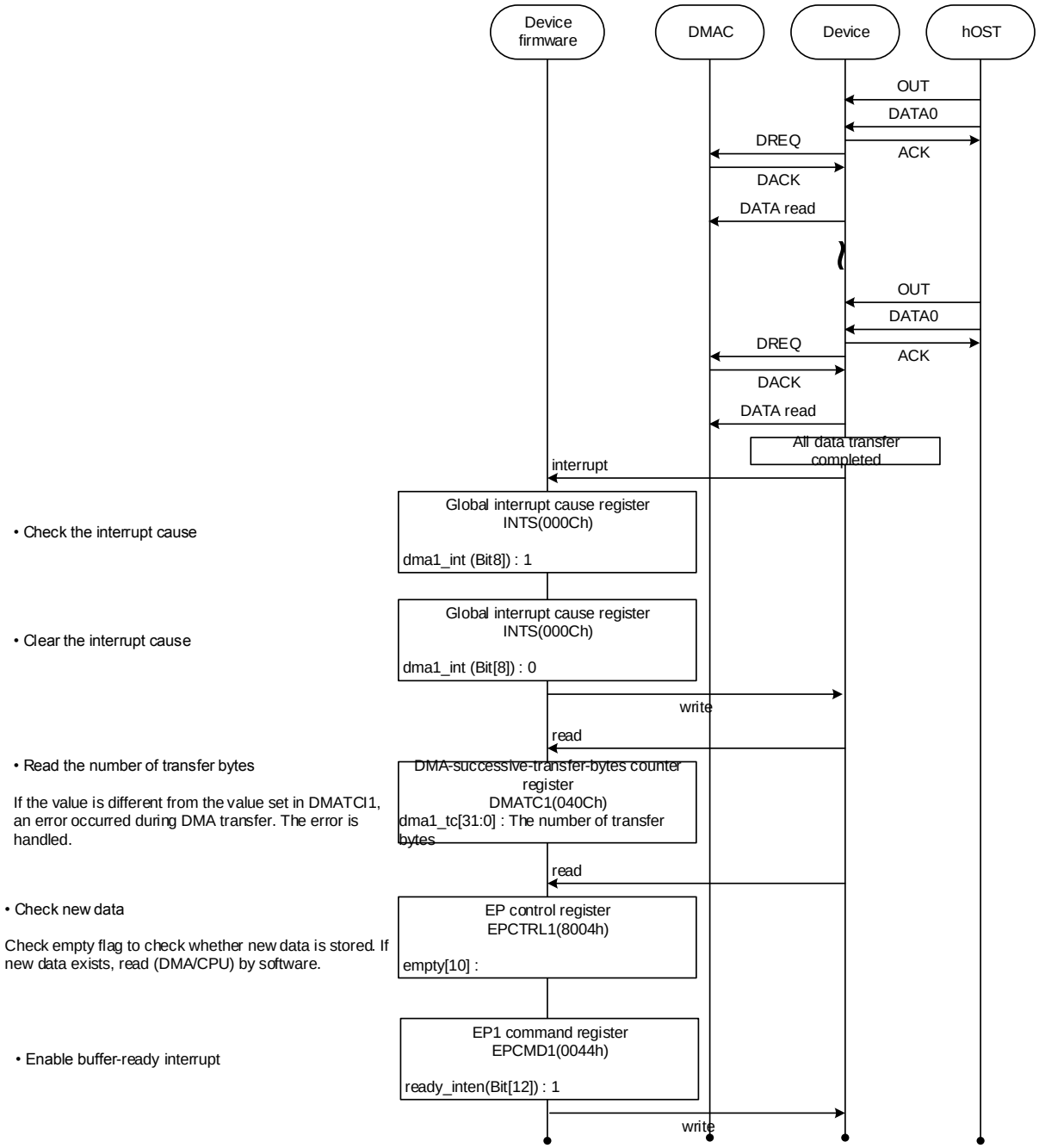

Device firmware
DMAC
Device
hOST
OUT
DATA0
DREQ
ACK
DACK
DATA read
OUT
DATA0
DREQ
ACK
DACK
DATA read
All data transfer completed
interrupt
• Check the interrupt cause
Global interrupt cause register
INTS(000Ch)
dma1_int (Bit8]) : 1
• Clear the interrupt cause
Global interrupt cause register
INTS(000Ch)
dma1_int (Bit[8]) : 0
write
read
• Read the number of transfer bytes
If the value is different from the value set in DMATCI1, an error occurred during DMA transfer. The error is handled.
DMA-successive-transfer-bytes counter register
DMATC1(040Ch)
dma1_tc[31:0] : The number of transfer bytes
read
• Check new data
Check empty flag to check whether new data is stored. If new data exists, read (DMA/CPU) by software.
EP control register
EPCTRL1(8004h)
empty[10] :
• Enable buffer-ready interrupt
EP1 command register
EPCMD1(0044h)
ready_inten(Bit[12]) : 1
write

Figure 21-78

DMA Write (IN) transfer

Preconditions
• Device (IN transfer) [buffer write direction]
• Target EP: EP1
• DMA Ch: 1Ch

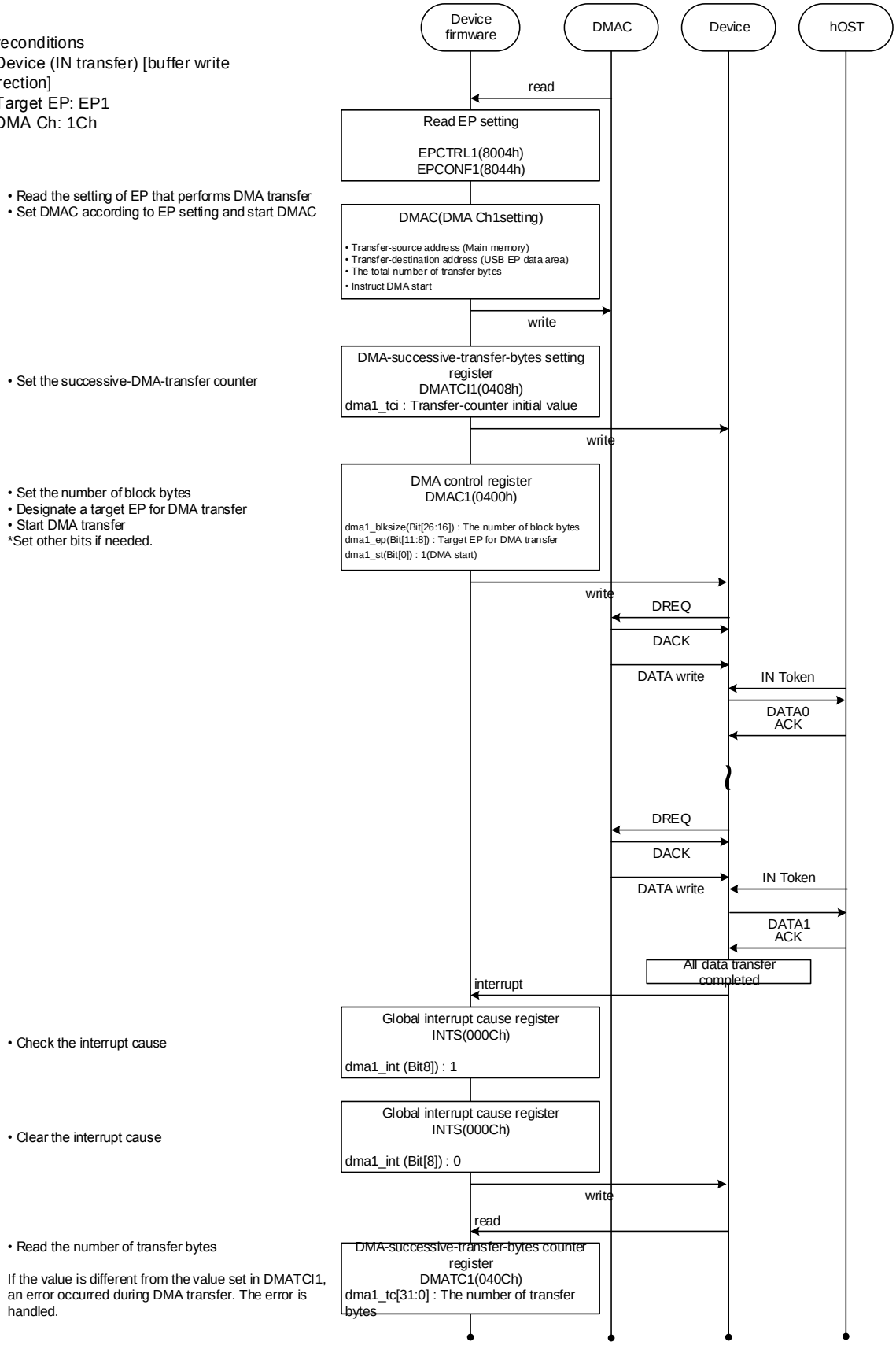


Figure 21-79

8.3.11. PING NYET

USB20DC automatically responds to PING.

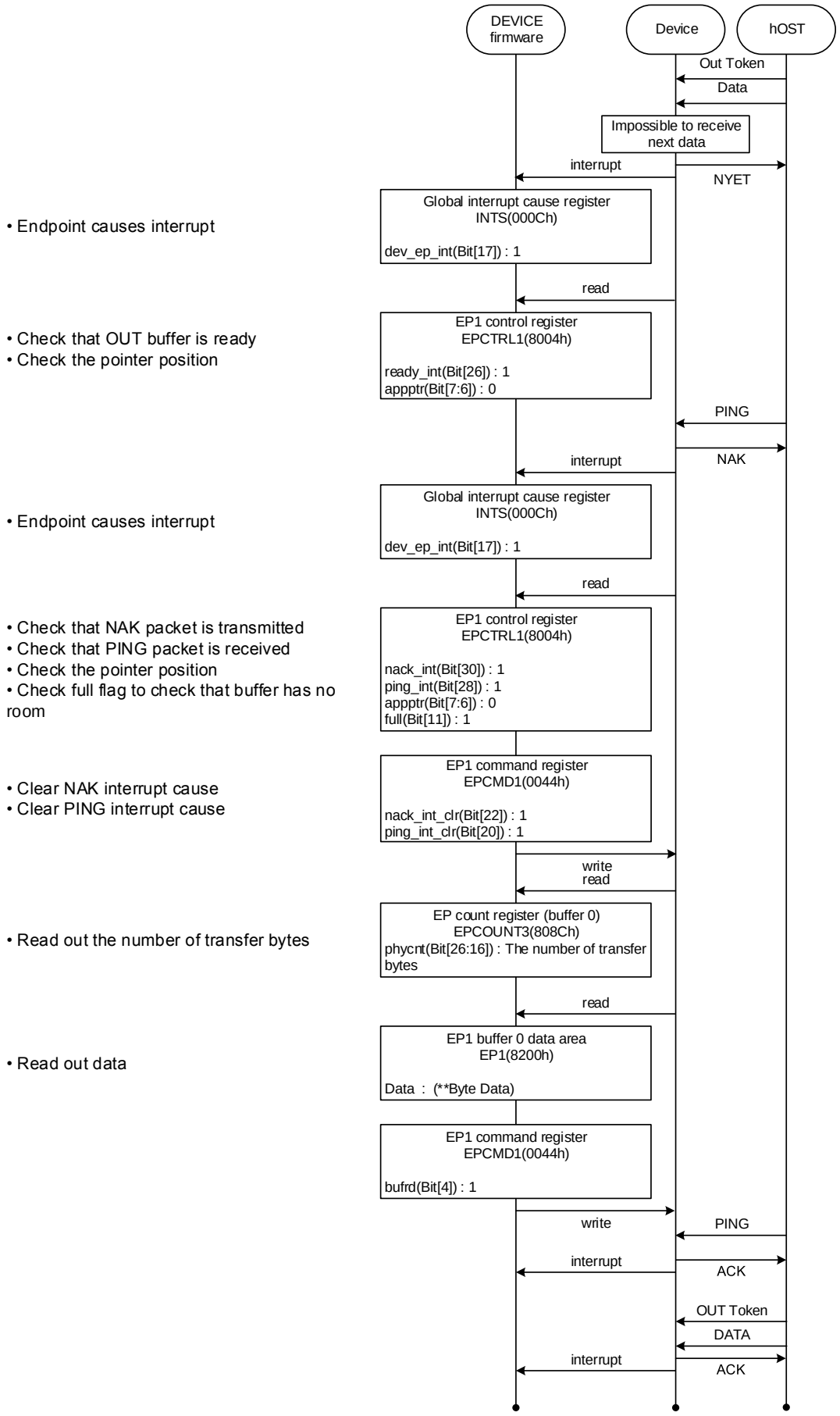

DEVICE firmware
Device
hOST
Out Token
Data
Impossible to receive next data
interrupt
NYET
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
read
• Check that OUT buffer is ready
• Check the pointer position
EP1 control register
EPCTRL1(8004h)
ready_int(Bit[26]) : 1
appptr(Bit[7:6]) : 0
PING
NAK
interrupt
• Endpoint causes interrupt
Global interrupt cause register
INTS(000Ch)
dev_ep_int(Bit[17]) : 1
read
• Check that NAK packet is transmitted
• Check that PING packet is received
• Check the pointer position
• Check full flag to check that buffer has no room
EP1 control register
EPCTRL1(8004h)
nack_int(Bit[30]) : 1
ping_int(Bit[28]) : 1
appptr(Bit[7:6]) : 0
full(Bit[11]) : 1
• Clear NAK interrupt cause
• Clear PING interrupt cause
EP1 command register
EPCMD1(0044h)
nack_int_clr(Bit[22]) : 1
ping_int_clr(Bit[20]) : 1
write
read
• Read out the number of transfer bytes
EP count register (buffer 0)
EPCOUNT3(808Ch)
phycnt(Bit[26:16]) : The number of transfer bytes
read
• Read out data
EP1 buffer 0 data area
EP1(8200h)
Data : (**Byte Data)
EP1 command register
EPCMD1(0044h)
bufrd(Bit[4]) : 1
write
PING
interrupt
ACK
OUT Token
DATA
interrupt
ACK

Figure 21-80

8.3.12. Suspend

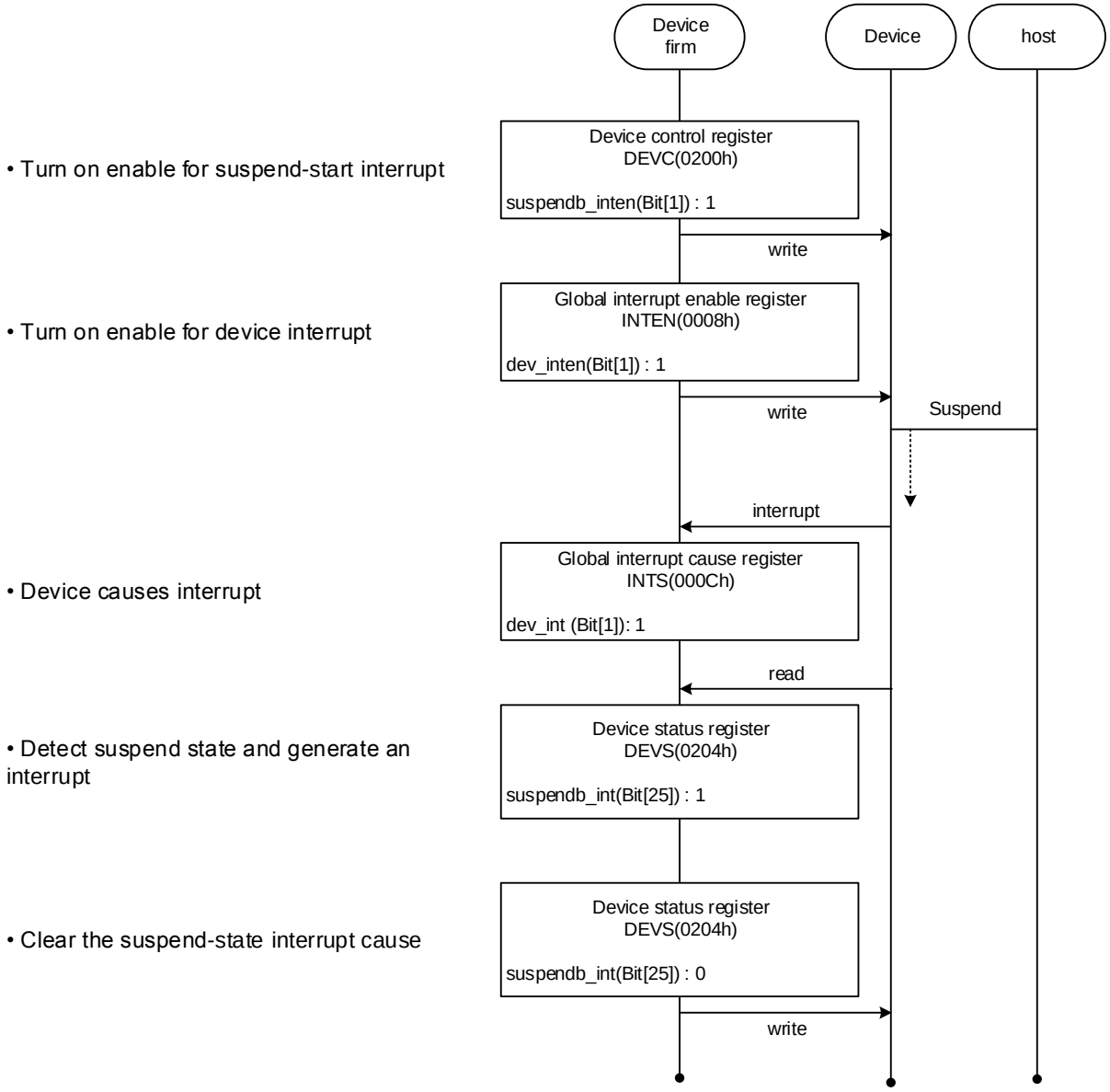


Figure 21-81

To stop XTAL oscillation at suspend and start XTAL oscillation at resume automatically, see page 702, “XTAL (p_cryclk_i) oscillation stop at suspend and the oscillation start by resume” and set USBCNT:RSMCONT register to process suspend interrupt and resume interrupt.

8.3.13. Resume

Host cause

• Turn on enable for suspend-state-end interrupt

• Device causes interrupt

• Check suspend-state-end interrupt

• Clear the suspend-state-end interrupt cause

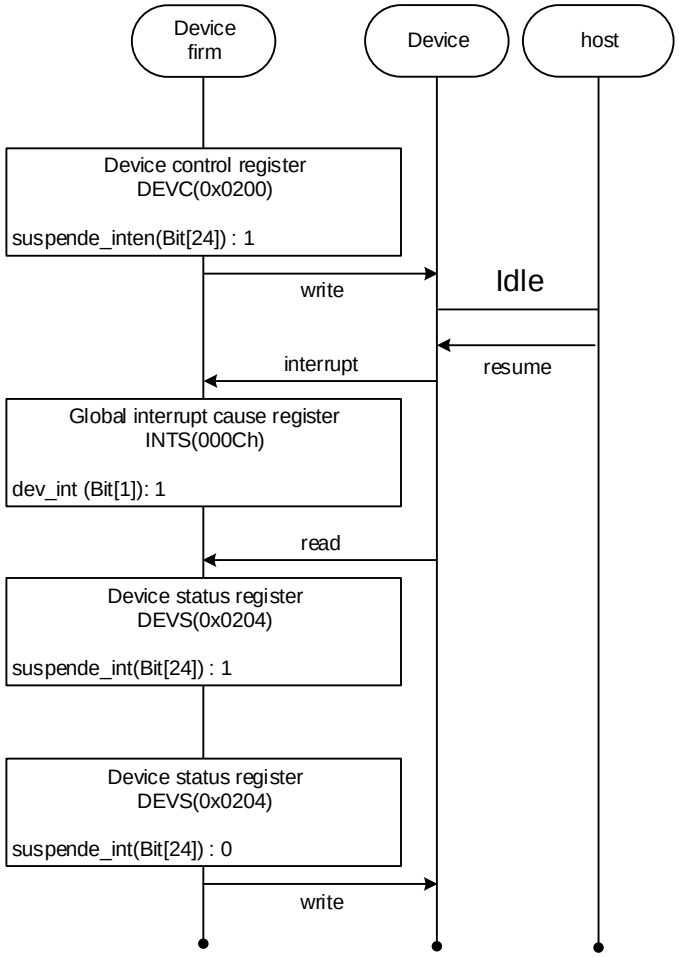


Figure 21-82

Device cause

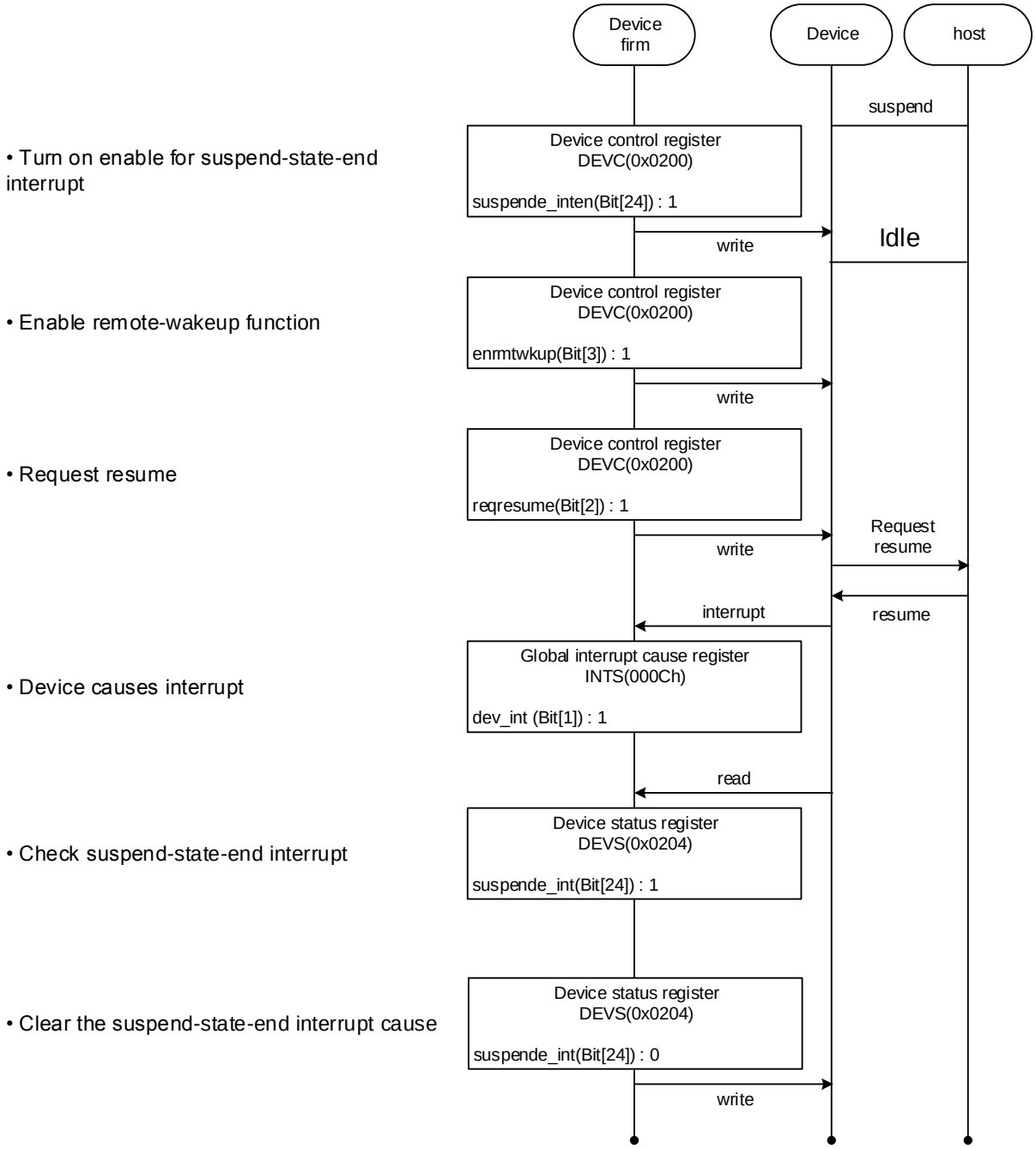


Figure 21-83

8.3.14. STALL and return

When EP0 is STALL

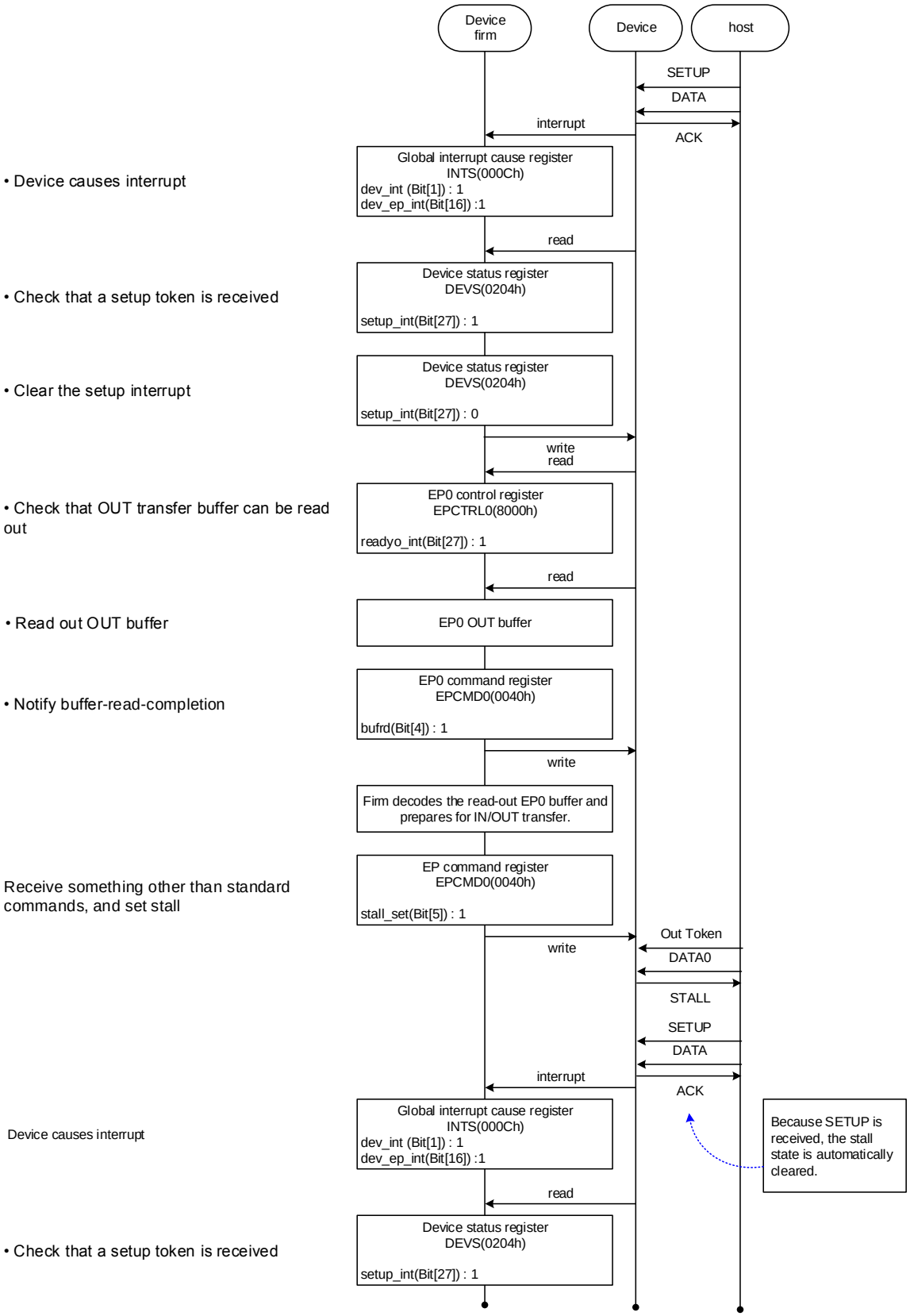


Figure 21-84

When any other endpoint than EP0 is STALL

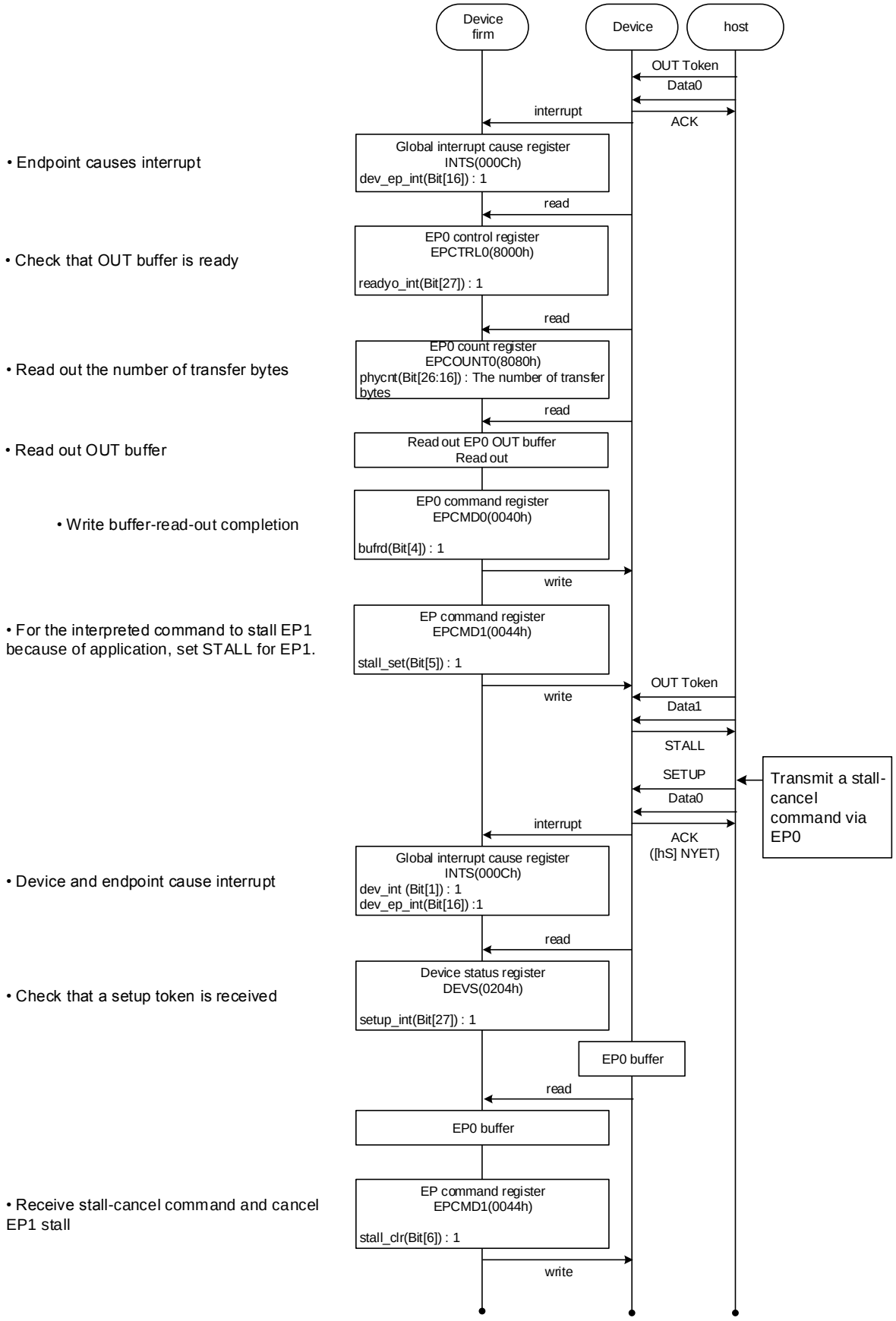


Figure 21-85

22 USB20HC

22-1 Introduction

This chapter provides the outline of USB20HC in the section below.

- The outline of USB20HC

22-1-1 The outline of USB20HC

1. The outline of macro

USB20HC is a USB host controller macro.

It has a controller (LINK) and a physical layer (PHY). The number of ports is 1 port only.

The controller has one EHCI compliant controller + OHCI compliant controller. It supports three modes: HS, FS, and LS.

The system interface conforms to AHB. It operates AHB slave and AHB master. AHB master operates DMA with the system memory in coordination with EHCI and OHCI.

2. Features

- With PHY block/LINK block
- With EHCI compliant controller + OHCI compliant controller
- Handling USB2.0 HS/FS/LS protocol
- Supports UTMI+ interface

(Necessary to support FS Serial mode)

- With Packet Buffer 512 Byte
- Not supports the legacy support function
- The list of outlined functions

The table below shows the list of USB20HC functions.

Table 1-1: The list of USB20HC functions_1

Item		**Function**	
OHCI 1.0a Standard HC	OHCI-standard (1.0a)-compliant Full/Low-Speed USB transfer		See Table 1-3: The list of the detailed functions of OHCI Full/Low-Speed transfer.
	AHB Slave I/F (sharing signals with EHCI Slave I/F)	HRESP	Generates OK/ERROR only.
		HBURST	Supports SINGLE only.
		HSIZE	Supports WORD only.
	AHB Master I/F	Endian	Little Endian
		HRESP	Supports OK/RETRY/SPLIT/ERROR.
		HBURST	Generates SINGLE/INCR only.
		HSIZE	Generates BYTE/WORD.
	Legacy support function		Not supports.
EHCI 1.0 Standard HC	EHCI-standard (1.0)-compliant High-Speed USB transfer	High-Speed transfer	SPLIT transfer can be processed with siTD BackPointer and FSTN. See Table 1-4: The list of detailed functions of EHCI High-Speed transfer.
		Debug Port Number	None.
		Port Indicator	None.
		Number of Companion Controller	The number of OHCIs: 1
		Number of Ports per Companion Controller	The number of ports per OHCI: 1
		Port Routing Rule	Routing Port1 to OHCI0.
		Port Power Control	PortPower control function
		Number of Ports	1
		Extended Capabilities Pointer	See Capability List in PCI Configuration Space. OFF
		Asynchronous PARK Mode	Can be turned on/off by Asynchronous PARK Mode program. Sets PARK value at the time of turn-on.
		Programmable Frame List	Setting function by Frame List Size register value
		64 bit Addressing	OFF

Table 1-2: The list of USB20HC functions_2

Item		Function	
EHCI 1.0 Standard HC	AHB Slave I/F (sharing signals with OHCI Slave I/F)	HRESP	Generates OK/ERROR only.
		HBURST	Supports SINGLE only.
		HSIZE	Supports WORD only.
	AHB Master I/F	Endian	Little Endian
		HRESP	Supports OK/RETRY/SPLIT/ERROR.
		HBURST	Generates SINGLE/INCR only.
		HSIZE	Generates BYTE/WORD.
	Not compliant with EHCI standard (Function unique to macro)	Programmable Microframe Base	Register value sets 1 μFrame length
		Programmable PacketBuffer Threshold	Setting function by the register value of PacketBuffer Threshold.
		Programmable PacketBuffer Depth	Setting function by the register value of PacketBuffer Depth.
		HCSPARAMS setting function	For debug.
		HCCPARAMS setting function	For debug.
		Port Enumeration short setting	For Sim.
		ON/OFF of Nak Reload correction function	Register value turns on/off the correction function. (Alwasy on)
		SOF period adjustment	Adjusts SOF period by the external pin or the register setting.

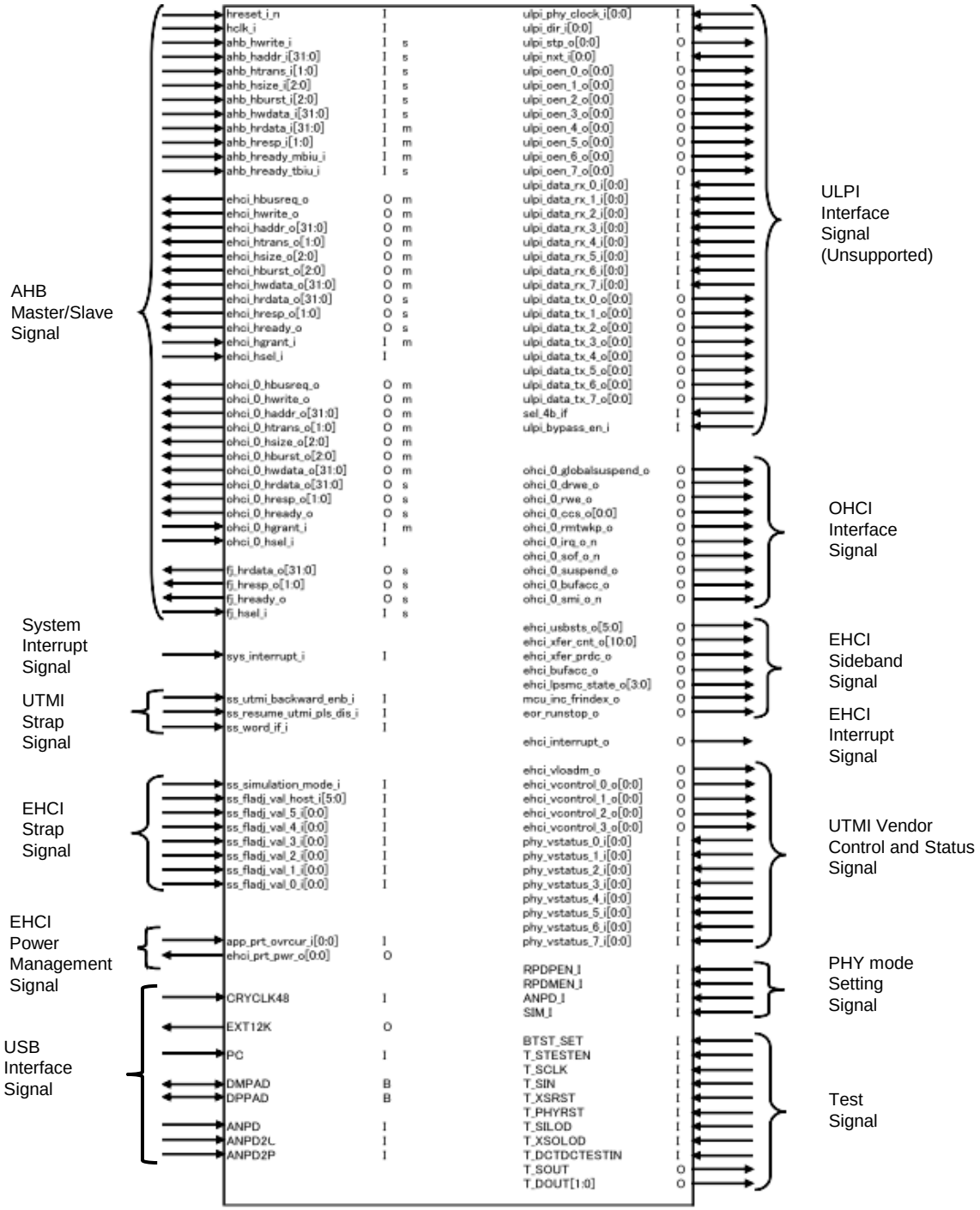
Table 1-3: The list of the detailed functions of OHCI Full/Low-Speed transfer

Bulk transfer	
	DATA toggle setting
	PID (IN/OUT) setting
	Transfer data (the number of bytes) setting
	Setting whether to judge an error when receiving a short packet
	Setting offset value in PacketDataBuffer
	Setting the interrupt when transfer is completed and the interrupt frame
Control transfer	
	DATA toggle setting
	PID (IN/OUT/SETUP) setting
	Transfer data (the number of bytes) setting
	Setting whether to judge an error when receiving a short packet
	Setting offset value in PacketDataBuffer
	Setting the interrupt when transfer is completed and the interrupt frame
	Full/Low-Speed setting (Setting whether PRE is used when FullSpeed-Hub is connected)
Interrupt transfer	
	DATA toggle setting
	PID (IN/OUT) setting
	Transfer data (the number of bytes) setting
	Setting whether to judge an error when receiving a short packet
	Setting offset value in PacketDataBuffer
	Setting the interrupt when transfer is completed and the interrupt frame
	Full/Low-Speed setting (Setting whether PRE is used when FullSpeed-Hub is connected.)
Isochronous transfer	
	PID (IN/OUT) setting
	Transfer data (the number of bytes) setting
	Setting whether to judge an error when receiving a short packet
	Setting offset value in PacketDataBuffer
	Setting the interrupt when transfer is completed and the interrupt frame
	Transfer execution frame setting

Table 1-4: The list of detailed functions of EHCI High-Speed transfer

Asynchronous (Control/Bulk) transfer	
	DATA toggle setting
	PID (IN/OUT/SETUP) setting
	Transfer data (the number of bytes) setting
	Setting offset value in PacketDataBuffer
	Setting next transfer data when ShortPkt/NormalPkt is received
	NakCount setting to limit the number of PINGs re-transmitted at High-Speed
	PARK setting the number of successive transfers to the same EndPoint at High-Speed
	Setting to execute split transfer when hub is connected
	Setting the interrupt when transfer is completed
	List-head indicating bit (H-bit) setting
Interrupt transfer (At High-Speed EndPoint)	
	DATA toggle setting
	PID (IN/OUT) setting
	Transfer data (the number of bytes) setting
	Setting offset value in PacketDataBuffer
	Setting next transfer data when ShortPkt/NormalPkt is received
	Mult setting the number of successive transfers for 1 μFrame in the case of High-Bandwidth EndPoint
	Setting the interrupt when transfer is completed
Interrupt transfer (At Full/Low-Speed EndPoint)	
	DATA toggle setting
	PID (IN/OUT) setting
	Transfer data (the number of bytes) setting
	Setting offset value in PacketDataBuffer
	Setting next transfer data when ShortPkt/NormalPkt is received
	Random setting of StartSplit/CompleteSplit execution μFrame: FSTN 2Frame transferred according to combination of the number of transfer bytes and the transfer start μFrame
	Setting the interrupt when transfer is completed
	Setting of transfer temporary stop (I-bit setting)
Isochronous transfer (At High-Speed EndPoint)	
	Mult setting the number of successive transfers for 1 μFrame in the case of High-Bandwidth EndPoint
	Transfer data (the number of bytes) setting
	Setting offset value in PacketDataBuffer
	Setting the interrupt when transfer is completed
Isochronous transfer (At Full-Speed EndPoint)	
	Transfer data (the number of bytes) setting
	Setting offset value in PacketDataBuffer
	Setting of StartSplit/CompleteSplit execution μFrame: BackPointer 2Frame transferred according to combination of the number of transfer bytes and the transfer start μFrame
	Setting the interrupt when transfer is completed

2. Macro pins



Attention: When needed, functions are explained with these macro-pin names, but not all macro pins are explained.

The table below shows where main macro pins used in this specification are connected to.

- Pin connection

Macro pins	Connection place
hclk_i	Bus clock, BASIC_CLK (See the datasheet.)
CRYCLK48	XT1 oscillation (See the datasheet.)
hreset_i_n	Internal signal that is hard-reset or reset by MRSTCNTBASIC:USB RSTB register. MRSTCNTBASIC: USB RSTB register (See the specification of System Controller.)
ahb_haddr_i	AHB address bus
ehci_prt_pwr_o	USBSTAT: EHCI_PRT_PWR register (See the specification of System Controller.)
app_prt_ovrcur_i	USBCNT:APP_PRT_OVRCUR register (See the specification of System Controller.)
sys_interrupt_i	USBCNT:SYS_INTERRUPT register (See the specification of System Controller.)
ulpi_bypass_en_i	Input fixed at 1’b1
phy_vstatus_[7:0]_i	Input fixed at 6’b000000
ss_fladj_val_[5:0]_i	Input fixed at 6’b100000
ss_fladj_val_host_[5:0]_i	Input fixed at 6’b100000
ss_simulation_mode_i	Input fixed at 1’b0
ss_word_if_i	Input fixed at 1’b0
ss_resume_utmi_pls_dis_i	Input fixed at 1’b0
ss_utmi_backward_enb_i	Input fixed at 1’b0
RPDPEN_I	Input fixed at 1’b1
RPDMEN_I	Input fixed at 1’b1
ANPD_I(ANPD)	USBCNT:ANPD register (See the specification of System Controller.)
SIM_I	Input fixed at 1’b0
DPPAD	USBDP pin (See the datasheet.)
DMPAD	USBDM pin (See the datasheet.)
EXT12K	USBEXT12 pin (See the datasheet.)

See also the specification of System Controller for other registers necessary to control USB20HC.

8. Clock

This macro has 3 clocks: PHY-side logic operation clock (CRYCLK48 pin), PHY reference clock (PHYCLK. The pin is not described in this specification.), and application-side (AHB bus access and the like) logic operation clock (hclk_i pin).

* Attention: The clock frequency of hclk_i shall be set as below.

30 MHz ≤ hclk_i ≤ BASIC_CLK maximum operation frequency (See the datasheet.)

9. Reset

9.1. Reset timing

When hreset_i_n is asserted, USB macro gets reset immediately and USB macro does not output the undefined.

The reset generation circuit inside USB macro generates reset signals needed in USB macro. Reset signals generated by the reset generation circuit are cancelled after being continued until the specified number of cycles of CRYCLK48 clock. When CRYCLK48 clock is not input, reset is not cancelled.

After hreset_i_n is de-asserted, AHB slave access cannot be made until 64 cycles of CRYCLK48 + 10 cycles of hclk_i because a reset signal continues in USB macro and the internal registers (LINK mode setting Register, PHY mode setting1 Register) are initialized.

9.2. Timing chart

The chart below shows the timing of USB20HC clock and reset.

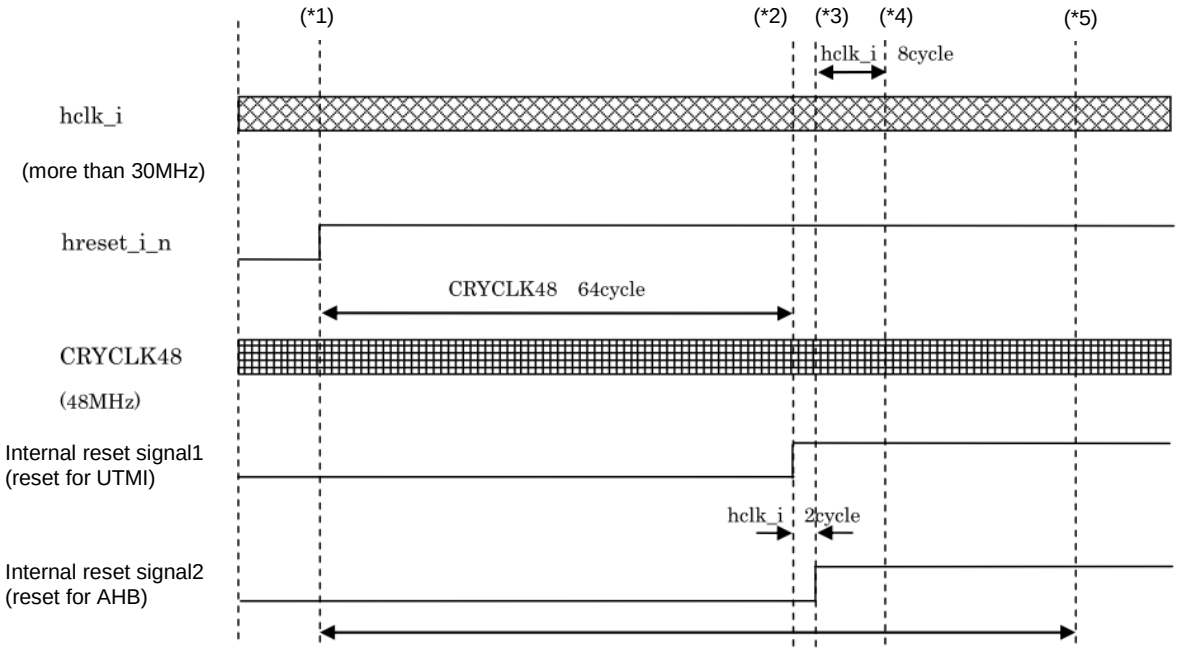


Figure 22-1

(*1): The system reset is cancelled. It is necessary that stable CRYCLK48 is being input.

(*2): Internal reset signal 1 (UTMI reset) is cancelled.

(*3): Internal reset signal 2 (AHB reset) is cancelled.

(*4): The external pin value is set in the internal registers (LINK mode setting Register, PHY mode setting1 Register).

(*5): PHY starts. It takes 1 ms from (*1) to (*5).

- Register access in macro can be made after (*4).
- USB communication can be established after (*5).
- Before hreset_i_n is asserted, set System Controller HF_SEL = 1 (host), ANPD = 0 (PHY power down cancellation), CLKSEL [2:0], and CRYCNTSW [1:0]. (See the specification of System Controller.)

- Assertion period of hreset_i_n shall be 200 us or more, and during assertion, CRYCLK48 rising edge shall be input until 4 cycles or more.

10. Precautions for macro use

10.1. Packet data storage area

Set transfer addresses so that 1 packet data does not cross a 4 kB-page boundary in the packet data storage area as explained below.

The figure below shows data structure example at the time of EHCI bulk transfer, control transfer, and interrupt transfer. 4 kByte pages are assigned in the buffer area. Bufrer Ptr designates the head address of 4 kByte page. The example below shows that Buffer Ptr assigns 5 pages of 4 kByte.

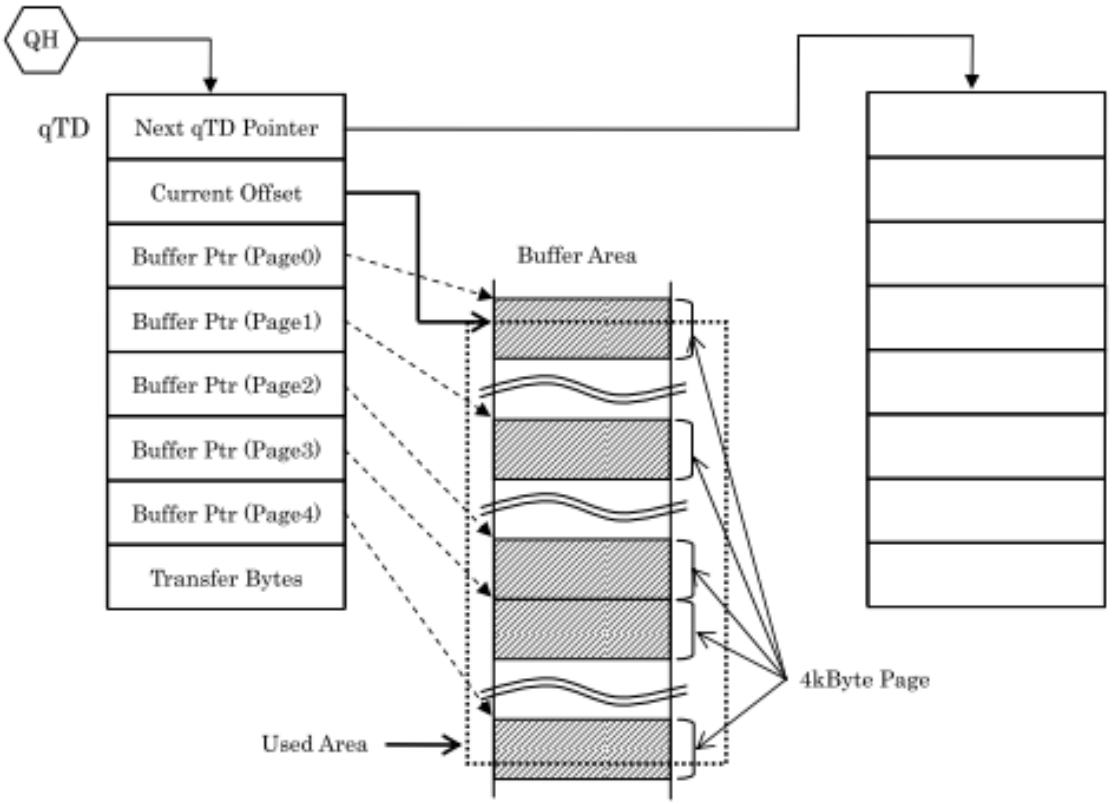


Figure 22-2

Example 1 shows a case where 4 kByte page is divisible by EpMaxPktSize.
Example 2 shows a case where 4 kByte page is not divisible by EpMaxPktSize.

Example 1: When 4 kByte page is divisible by EpMaxPktSize, the initial setting value of current offset shall be set at an integer multiple of EpMaxPktSize. In this case, 5 or less pages of 4 kByte can be used for transfer.

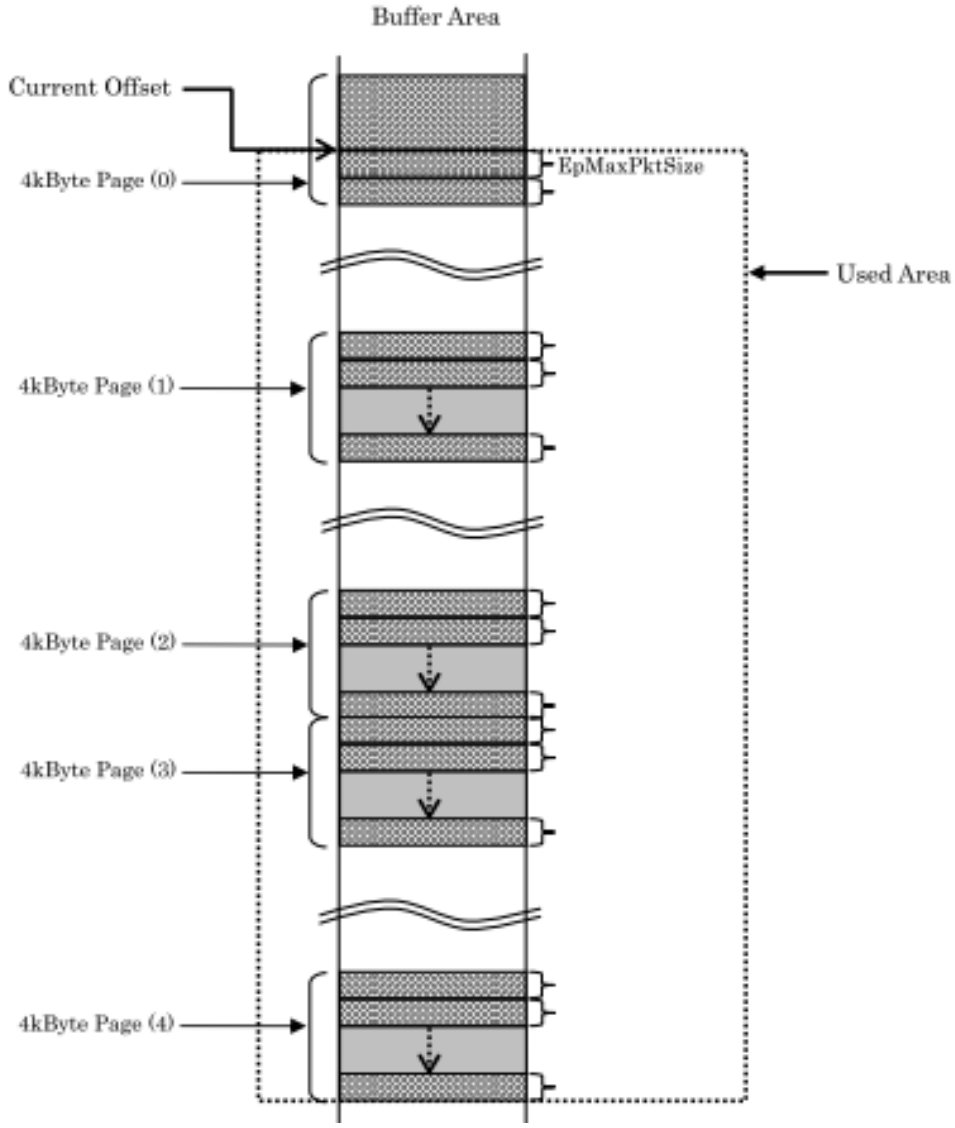


Figure 22-3

Example 2: When 4 kByte page is not divisible by EpMaxPktSize, the initial setting value of current offset shall be set at “4 k – (An integer multiple of EpMaxPktSize)”. In this case, 2 or less pages of 4 kByte can be used.

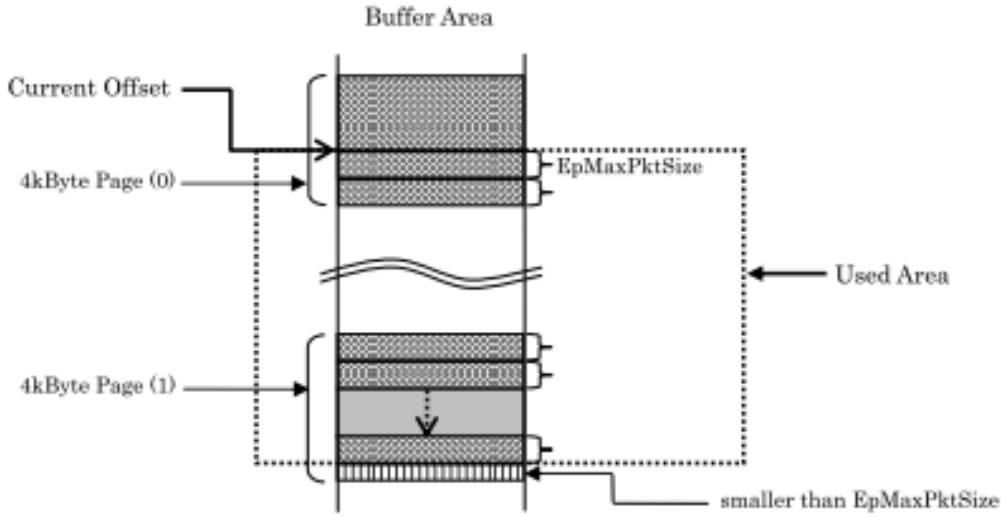


Figure 22-4

10.2. Clock input in the suspend state

The suspend state of this macro needs both CRYCLK48 and hclk_i to be input.

10.3. Example of VBUS control circuit connection

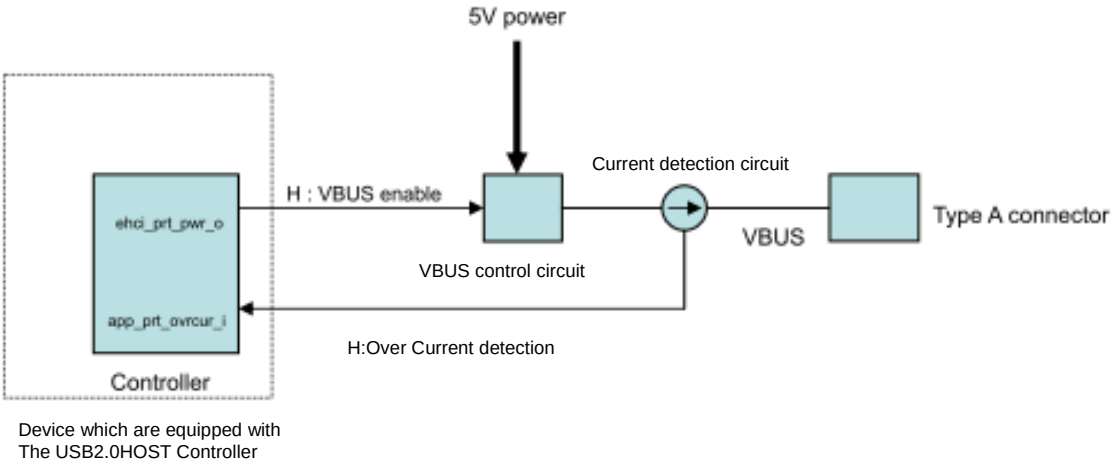


Figure 22-5

USB20HC signal of ehci_prt_pwr_o controls VBUS. When the signal is high, 5 V is supplied to VBUS.

When high is input to App_prt_ovrcur_i, the controller detects the state of over-current. Each signal shall be connected to VBUS control circuit according to the signal polarities.

Both signal pins are 3 V signals. 5 V connection needs a voltage conversion circuit.

10.4. Power control and power-down control

To put this macro into the low-power-consumption mode when this macro is turned on, use ANPD pin.

- The operation when USB macro is not used

When USB macro on LSI is not used, perform the operation as below.

Set ANPD = 1 to put USB into the power-down state. ANPD = 1 can be set at any time. DPPAD/DMPAD/EXT12K shall be connected via 10 k–100 kΩ to GND.

22-2 Functional Overview

This chapter provides the outline of USB20HC function block in the section below.

- USB20HC circuit configuration

22-2-1 USB20HC circuit configuration

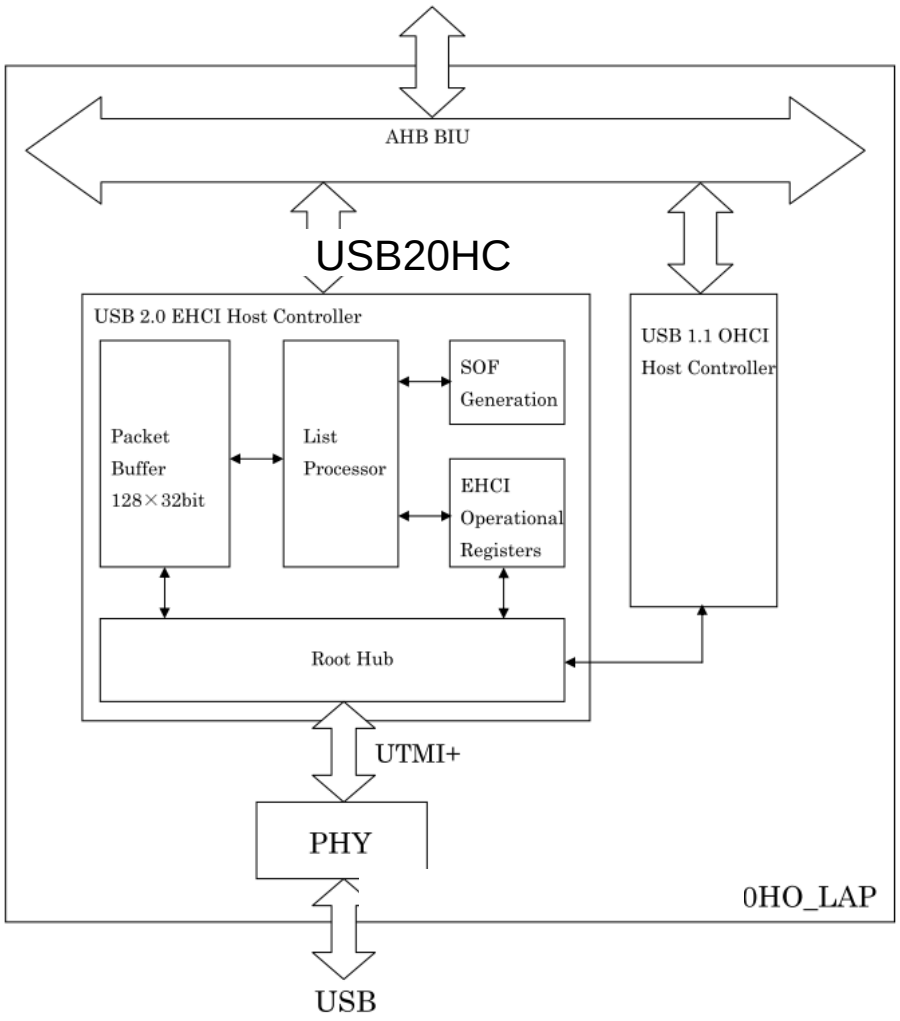


Figure 22-6

22-3 Programmer's Model

This chapter provides USB20HC registers and setting via microcontroller in the sections below.

- About the programmer's model
- Summary of USB20HC registers
- Register Descriptions

22-3-1 About the programmer’s model

USB20HC base address is not fixed, which depends on system implementation. See the system specification for base address. Each register’s offset is fixed.

22-3-2 Summary of USB20HC registers

USB20HC is controlled by the registers in “Table 22-1”.

Remark: Do not access any unspecified address from Base+0x0000 to Base+0xFFFF.

7. Register

EHCIBASE = 0x000, OHCIBASE = 0x200, and OTHBASE = 0x400 in the table below.

Actually, host macro decodes and uses only lower 9 bits (ahb_haddr_i[8:0]) of ahb_haddr_i.

Table 22-1 USB20HC registers

Offset	Abbreviation	Attribute	Register name	Initial value
EHCI Operational Registers(ahb_haddr_i[7:0])				
EHCIBASE +0x00	HCCAPBASE	R	Capability Register	0x0100_0010
EHCIBASE +0x04	HCSPARAMS	R	Structural Parameter	0x0000_1212
EHCIBASE +0x08	HCCPARAMS	R	Capability Parameter	0x0000_0016
EHCIBASE + 0x10	USBCMD	R/W	USB Command	0x0008_0B00
EHCIBASE + 0x14	USBSTS	R/W	USB Status	0x0000_1000
EHCIBASE + 0x18	USBINTR	R/W	USB Interrupt Enable	0x0000_0000
EHCIBASE + 0x1c	FRINDEX	R/W	USB Frame Index	0x0000_0000
EHCIBASE + 0x20	CTRLDSSEGMENT	R/W	4G Segment Selector	0x0000_0000
EHCIBASE + 0x24	PERIODICLISTBASE	R/W	Periodic Frame List Base Address Register	0x0000_0000
EHCIBASE + 0x28	ASYNCLISTADDR	R/W	Asynchronous List Address	0x0000_0000
EHCIBASE + 0x50	CONFIGFLAG	R/W	Configured Flag Register	0x0000_0000
EHCIBASE + 0x54	PORTSC_1	R/W	Port Status/Control	0x0000_2000
EHCIBASE + 0x58	Reserved		Do not access.	
EHCIBASE + 0x90	INSNREG00	R/W	Programmable Microframe Base Value	0x0000_0000
EHCIBASE + 0x94	INSNREG01	R/W	Programmable Packet Buffer OUT/IN Thresholds	0x0020_0020
EHCIBASE + 0x98	INSNREG02	R/W	Programmable Packet Buffer Depth	0x0000_0080
EHCIBASE + 0x9C	INSNREG03	R/W	Break Memory Transfer	0x0000_0000
EHCIBASE + 0xA0	INSNREG04	R/W	Debug Register	0x0000_0000
EHCIBASE + 0xA4	INSNREG05	R/W	UTMI Control Status Registers	0x0000_0000
OHCI Operational Registers(ahb_haddr_i[8:0])				
OHCIBASE + 0x000	HcRevision	R		0x0000_0010
OHCIBASE +0x004	HcControl	R/W		0x0000_0000
OHCIBASE +0x008	HcCommandStatus	R/W		0x0000_0000
OHCIBASE +0x00C	HcInterruptStatus	R/W		0x0000_0000
OHCIBASE +0x010	HcInterruptEnable	R/W		0x0000_0000
OHCIBASE +0x014	HcInterruptDisable	R/W		0x0000_0000
OHCIBASE +0x018	HcHCCA	R/W		0x0000_0000
OHCIBASE +0x01C	HcPeriodCurrentED	R/W		0x0000_0000
OHCIBASE +0x020	HcControlHeadED	R/W		0x0000_0000
OHCIBASE +0x024	HcControlCurrentED	R/W		0x0000_0000
OHCIBASE +0x028	HcBulkHeadED	R/W		0x0000_0000
OHCIBASE +0x02C	HcBulkCurrentED	R/W		0x0000_0000
OHCIBASE +0x030	HcDoneHead	R/W		0x0000_0000
OHCIBASE +0x034	HcFmInterval	R/W		0x0000_2EDF
OHCIBASE +0x038	HcFmRemaining	R		0x0000_0000
OHCIBASE +0x03C	HcFmNumber	R		0x0000_0000
OHCIBASE +0x040	HcPeriodicStart	R/W		0x0000_0000
OHCIBASE +0x044	HcLSThreshold	R/W		0x0000_0628
OHCIBASE +0x048	HcRhDescriptorA	R/W		0x0200_0902
OHCIBASE +0x04C	HcRhDescriptorB	R/W		0x0006_0000
OHCIBASE +0x050	HcRhStatus	R/W		0x0000_0000
OHCIBASE +0x054	HcRhPortStatus[1]	R/W		0x0000_0000
OHCIBASE +0x058	Reserved		Do not access.	
OHCIBASE +0x100	Reserved		Do not access.	
OHCIBASE +0x104	Reserved		Do not access.	
OHCIBASE +0x108	Reserved		Do not access.	

OHCIBASE +0x10C	Reserved		Do not access.	
Other Registers(ahb_haddr_i[3:0])				
OTHBASE +0x0	Link mode setting	R/W		0x0060_8000
OTHBASE +0x4	PHY mode setting1	R/W		0x0700_4018
OTHBASE +0x8	PHY mode setting2	R/W		0x0020_0018
OTHBASE +0xC	PHY mode setting3	R/W		0x0000_0000
OTHBASE +0x10	Reserved		Do not access.	

22-3-3 Register Descriptions

8.1. EHCI Operational Registers

22-3-3-1 Capability Register:HCCAPBASE(8.1.1.)

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	HCIVERSION 15	HCIVERSION 14	HCIVERSION 13	HCIVERSION 12	HCIVERSION 11	HCIVERSION 10	HCIVERSION 9	HCIVERSION 8
Reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	HCIVERSION 7	HCIVERSION 6	HCIVERSION 5	HCIVERSION 4	HCIVERSION 3	HCIVERSION 2	HCIVERSION 1	HCIVERSION 0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	CAPLENGTH 7	CAPLENGTH 6	CAPLENGTH 5	CAPLENGTH 4	CAPLENGTH 3	CAPLENGTH 2	CAPLENGTH 1	CAPLENGTH 0
Reset	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit31-16: HCIVERSION
Indicates EHCI Rev number.
31-24: major revision number
23-16: minor revision number

Bit7-0: CAPLENGTH
Operational Register space offset

22-3-3-2 Structural Parameter Register: HCSPARAMS(8.1.2.)

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	Debug Port Number3	Debug Port Number2	Debug Port Number1	Debug Port Number0	-	-	-	P_INDICATOR
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	N_CC3	N_CC2	N_CC1	N_CC0	N_PCC3	N_PCC2	N_PCC1	N_PCC0
Reset	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	Port Routing Rule	-	-	PPC	N_PORTS3	N_PORTS2	N_PORTS1	N_PORTS0
Reset	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R

Bit23-20: Debug Port Number
For this macro, “0” is read to indicate that there is no debug port.

Bit16: P_INDICATOR
Port Indicator
For this macro, “0” is read to indicate that port indicator control is not supported.
1’b0: Not supports.
1’b1: Supports.

Bit15-12: N_CC
Number of Companion Controller
Indicates the number of USB1.1 Host Controllers.
For this macro, “1” is read to indicate that there is one USB1.1 Host Controller.

Bit11-8: N_PCC
Number of Ports per Companion Controller
Indicates the number of ports that USB1.1 Host Controller supports.
When these bits are read out, “2” is read to indicate that USB1.1 Host Controller supports 2 ports.
Actually, this macro supports 1 port (Port1).

Bit7: Port Routing Rule
Indicates how all ports are assigned to USB1.1 Host Controller.
1’b0: Assigns in ascending order of port number.
1’b1: Depends on the first N_PORTS elements of HCSP-PORTROUTE array.
“0” is read to indicate that this macro assigns ports in ascending order of port number.

Bit4: PPC
Port Power Control
Indicates whether Host Controller can control port power.
1’b0: Cannot control port power.

1’b1: Can control port power.

For this macro, “1” is read to indicate that Host Controller can control port power.

Bit3-0: N_PORTS

Indicates the number of port registers that can be accessed in the operational register space. When these bits are read out, “2” is read to indicate that 2 port registers can be accessed. Actually, this macro supports 1 port (Port1).

22-3-3-3 Capability Parameter Register:HCCPARAMS(8.1.3.)

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	EECP7	EECP6	EECP5	EECP4	EECP3	EECP2	EECP1	EECP0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	Isochronous Scheduling Threshold3	Isochronous Scheduling Threshold2	Isochronous Scheduling Threshold1	Isochronous Scheduling Threshold0	-	Asynchronous Schedule Park Capability	Programmable Frame List Flag	64 bit Addressing Capability
Reset	0	0	0	1	0	1	1	0
R/W	R	R	R	R	R	R	R	R

Bit15-8: EECP
EHCI Extended Capabilities Pointer
Indicates the offset in PCI configuration space.

Bit7-4: Isochronous Scheduling Threshold
When Bit [7] is “0”, it indicates what micro frame the software can update isochronous schedule at.

Bit2: Asynchronous Schedule Park Capability
For this macro, “1” is read to indicate that Host Controller supports the park feature for hi-speed queue head of asynchronous schedule.
1’b0: Not supports.
1’b1: Supports.

Bit1: Programmable Frame List Flag
For this macro, “1” is read to indicate that the size can be determined by Frame List Size of USBCMD register.
1’b0: Fixed at 1024.
1’b1: Determines the size by Frame List Size of USBCMD register.

Bit0: 64 bit Addressing Capability
For this macro, “0” is read to indicate that the data structure uses 32-bit address.
1’b0: Data structure uses 32-bit address pointer.
1’b1: Data structure uses 64-bit address pointer.

22-3-3-4 USB Command Register:USBCMD(8.1.4.)

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	Interrupt Threshold Control7	Interrupt Threshold Control6	Interrupt Threshold Control5	Interrupt Threshold Control4	Interrupt Threshold Control3	Interrupt Threshold Control2	Interrupt Threshold Control1	Interrupt Threshold Control0
Reset	0	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	Asynchronous Schedule Park Mode Enable	-	Asynchronous Schedule Park Mode Count1	Asynchronous Schedule Park Mode Count0
Reset	0	0	0	0	1	0	1	1
R/W	R	R	R	R	R/W	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	Light Host Controller Reset	Interrupt on Async Advance Doorbell	Asynchronous Schedule Enable	Periodic Schedule Enabe	Frame List Size1	Frame List Size0	HCRESET	RS
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit23-16: Interrupt Threshold Control
Indicates the frame period when Host Controller issues an interrupt.
00h: Reserved
01h: 1 micro frame
02h: 2 micro frame
04h: 4 micro frame
08h: 8 micro frame (= 1ms)
10h: 16 micro frame (= 2ms)
20h: 32 micro frame (= 4ms)
40h: 64 micro frame (= 8ms)
Do not write any other value than those above.

Bit11: Asynchronous Schedule Park Mode Enable
For this macro, Asynchronous Park Capability bit of HCCPARAMS register is “1”. So, this bit can be read/written.
1’b0: park mode disable
1’b1: park mode enable

Bit9-8: Asynchronous Schedule Park Mode Count
For this macro, Asynchronous Park Capability bit of HCCPARAMS register is “1”. So, this bit can be read/written.
Effective values are 1h-3h. Do not write 0h.

Bit7: Light Host Controller Reset
Resets EHCI host controller, which does not affect the port state or OHCI host controller.

At the time of read
1'b0: Light host controller reset is completed.
1'b1: Light host controller reset is being executed.

Bit6: Interrupt on Async Advance Doorbell
Informs the host controller that the next asynchronous schedule advances to issue an interrupt.
After "1" is set for the interrupt of Async Advance Status bit of USBSTS register, Host Controller sets "0" in this bit.
When Async Schedule is disabled, "1" cannot be written in this bit.

Bit5: Asynchronous Schedule Enable
This bit controls whether the host controller skips async schedule processing.
1'b0: Not executes async schedule processing.
1'b1: Executes async schedule processing.

Bit4: Periodic Schedule Enabe
This bit controls whether the host controller skips periodic schedule processing.
1'b0: Not executes periodic schedule processing.
1'b1: Executes periodic schedule processing.

Bit3-2: Frame List Size
For this macro, Programmable Frame List Flag of HCCPARAMS register is "1". So, this bit is read/written.
Determines the frame list size.
00 b: 1024 elements (4096 bytes)
01 b: 512 elements (2048 bytes)
10 b: 256 elements (1024 bytes)
11 b: Reserved

Bit1: HCRESET
Host Controller Reset
When "1" is written in this bit, the host controller initializes pipeline, timer, counter and state machine, and port registers and port state machine are set at the initial values.
The port ownership returns to USB1.1 host controller. When a reset is completed, this bit is set to "0" by the host controller. When HCHalted bit of USBSTS register is "0", do not write "1" in this bit.

Bit0: RS
Run/Stop
When "1" is written, the host controller executes the schedule. When "0" is written, current transaction is completed to a stop.
1'b0: Stop
1'b1: Run
When HCHalted of USBSTS register is "0", do not write "1" in this bit.

22-3-3-5 USB Status Register:USBSTS(8.1.5.)

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	Asynchronous Schedule Status	Periodic Schedule Status	Reclamation	HCHalted	-	-	-	-
Reset	0	0	0	1	0	0	0	0
R/W	R	R	R	R	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	-	Interrupt on Async Advance	Host System Error	Frame List Rollover	Port Change Detect	USBERRINT	USBINT
Reset	0	0	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit15: Asynchronous Schedule Status
This bit indicates the current state of async schedule.
1'b0: Async Schedule Disable
1'b1: Async Schedule Enable

Bit14: Periodic Schedule Status
This bit indicates the current state of periodic schedule.
1'b0: Periodic Schedule Disable
1'b1: Periodic Schedule Enable

Bit13: Reclamation
This bit indicates whether async schedule is empty.

Bit12: HCHalted
When Run/Stop bit is "1 (Run)", this bit becomes "0".
After Run/Stop bit is set to "0 (Stop)" and the process stops, the host controller sets this bit to "1".

Bit5: Interrupt on Async Advance
Writing "1" in Interrupt on Async Advance Doorbell bit of USBCMD register makes the host controller issue an interrupt the next time async schedule advances. Writing "1" in this bit clears it to be "0".

Bit4: Host System Error
When the host controller accesses the memory and is responded to with an error or when "1" is input in sys_interrupt_i, the host controller sets this bit to "1".
When this error occurs, Run/Stop bit of Command register is cleared to prevent the schedule from being executed. Writing "1" in this bit clears it to be "0".

Bit3: Frame List Rollover
When Frame List Index returns form the maximum value to 0, the host controller sets this bit to "1".

Writing “1” in this bit clears it to be “0”.

Bit2: Port Change Detect

When a port whose Port Owner bit is set to “0” changes from “0” to “1”, when Force Port Resume bit changes from “0” to “1” because J-K is detected on a suspended port, or when EHCI releases a port by writing “1” in Port Owner bit, the host controller sets this bit to “1”. Writing “1” in this bit clears it to be “0”.

Bit1: USBERRINT

USB Error Interrupt

When USB transfer results in an error, the host controller sets this bit to “1”. When TD is error-interrupted and the IOC bit is set, both this bit and USBINT bit are set.

Writing “1” in this bit clears it to be “0”.

Bit0: USBINT

When USB transfer is completed and a transfer descriptor whose IOC bit is set retires, the host controller sets this bit to “1”.

When a short packet is detected, the host controller sets this bit to “1”. Writing “1” in this bit clears it to be “0”.

22-3-3-6 USB Interrupt Enable Register:USBINTR(8.1.6.)

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	Interrupt on Async Advance Enable	Host System Error Enable	Frame List Rollover Enable	Port Change Interrupt Enable	USB Error Interrupt Enable	USB Interrupt Enable
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit5: Interrupt on Async Advance Enable
Enables interrupt by Interrupt on Async Advance of USBSTS register.
1‘b0: Disables interrupt.
1‘b1: Enables interrupt.

Bit4: Host System Error Enable
Enables interrupt by Host System Error Status of USBSTS register.
1‘b0: Disables interrupt.
1‘b1: Enables interrupt.

Bit3: Frame List Rollover Enable
Enables interrupt by Frame List Rollover of USBSTS register.
1‘b0: Disables interrupt.
1‘b1: Enables interrupt.

Bit2: Port Change Interrupt Enable
Enables interrupt by Port Change Detect of USBSTS register.
1‘b0: Disables interrupt.
1‘b1: Enables interrupt.

Bit1: USB Error Interrupt Enable
Enables interrupt by USBERRINT of USBSTS register.
1‘b0: Disables interrupt.
1‘b1: Enables interrupt.

Bit0: USB Interrupt Enable
Enables interrupt by USBINT of USBSTS register.
1‘b0: Disables interrupt.
1‘b1: Enables interrupt.

22-3-3-7 USB Frame Index Register:FRINDEX(8.1.7.)

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	Frame Index13	Frame Index12	Frame Index11	Frame Index10	Frame Index9	Frame Index8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	Frame Index7	Frame Index6	Frame Index5	Frame Index4	Frame Index3	Frame Index2	Frame Index1	Frame Index0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit13-0: Frame Index

This register value is incremented at the end of each frame. [N:3] is used for the current frame list number. The value of N is based on the value of Frame List Size of USBCMD register as below.

Frame List Size	Number Elements	N
00 b	(1024)	12
01 b	(512)	11
10 b	(256)	10
11 b	Reserved	

22-3-3-8 4G Segment Selector Register:CTRLDSSEGMENT(8.1.8.)

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	CTRLDSSEGMENT31	CTRLDSSEGMENT30	CTRLDSSEGMENT29	CTRLDSSEGMENT28	CTRLDSSEGMENT27	CTRLDSSEGMENT26	CTRLDSSEGMENT25	CTRLDSSEGMENT24
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	CTRLDSSEGMENT23	CTRLDSSEGMENT22	CTRLDSSEGMENT21	CTRLDSSEGMENT20	CTRLDSSEGMENT19	CTRLDSSEGMENT18	CTRLDSSEGMENT17	CTRLDSSEGMENT16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	CTRLDSSEGMENT15	CTRLDSSEGMENT14	CTRLDSSEGMENT13	CTRLDSSEGMENT12	CTRLDSSEGMENT11	CTRLDSSEGMENT10	CTRLDSSEGMENT9	CTRLDSSEGMENT8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	CTRLDSSEGMENT7	CTRLDSSEGMENT6	CTRLDSSEGMENT5	CTRLDSSEGMENT4	CTRLDSSEGMENT3	CTRLDSSEGMENT2	CTRLDSSEGMENT1	CTRLDSSEGMENT0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-0: CTRLDSSEGMENT

Control Data Structure Segment Register.

This register is equivalent to address bit [63:32] of EHCI data structure.

For this macro, this register is not used because 64 bit addressing Capability bit of HCCPARAMS register is “0”.

22-3-3-9 Periodic Frame List Base Address Register:PERIODICLISTBASE(8.1.9.)

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	Base Address19	Base Address18	Base Address17	Base Address16	Base Address15	Base Address14	Base Address13	Base Address12
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	Base Address11	Base Address10	Base Address9	Base Address8	Base Address7	Base Address6	Base Address5	Base Address4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	Base Address3	Base Address2	Base Address1	Base Address0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-12: Base Address
Base address of periodic frame list in the memory area.

Bit11-0: Reserved
Read/Write 0’s

22-3-3-10 Asynchronous List Address Register:ASYNCLISTADDR(8.1.10.)

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	Link Pointer Low26	Link Pointer Low25	Link Pointer Low24	Link Pointer Low23	Link Pointer Low22	Link Pointer Low21	Link Pointer Low20	Link Pointer Low19
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	Link Pointer Low18	Link Pointer Low17	Link Pointer Low16	Link Pointer Low15	Link Pointer Low14	Link Pointer Low13	Link Pointer Low12	Link Pointer Low11
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	Link Pointer Low10	Link Pointer Low9	Link Pointer Low8	Link Pointer Low7	Link Pointer Low6	Link Pointer Low5	Link Pointer Low4	Link Pointer Low3
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	Link Pointer Low2	Link Pointer Low1	Link Pointer Low0	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R

Bit31-5: Link Pointer Low

Includes the address for asynchronous queue head that is executed next.

22-3-3-11 Configured Flag Register:CONFIGFLAG(8.1.11.)

[Base + 0x0050]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	CF
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-1: Reserved.
Read/Write 0's

Bit0: CF
Configure Flag
This bit controls port routing. "1" is written in this bit as the last process for the host controller setting.
1'b0: OHCI controller owns the port routing control.
1'b1: EHCI controller owns the port routing control.

22-3-3-12 Port Status/Control Register:PORTSC_1(8.1.12.)

[Base + 0x0054]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	WKOC_E	WKDSCNNT_E	WKCNNT_E	Port Test Control3	Port Test Control2	Port Test Control1	Port Test Control0
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	Port Indicator Control1	Port Indicator Control0	Port Owner	PP	Line Status1	Line Status0	-	Port Reset
Reset	0	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W
Bit	7	6	5	4	3	2	1	0
Name	Suspend	Force Port Resume	Over-current Change	Over-current Active	Port Enable/Disable Change	Port Enable/Disable	Connect Status Change	Current Connect Status
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R

Bit22: WKOC_E
Wake on Over-current Enable
When “1” is written in this bit, an over-current state (app_prt_ovrcur_i = 1) is treated as a wake-up event. When Port Power is “0”, this bit is “0”.

Bit21: WKDSCNNT_E
Wake on Disconnect Enable
When “1” is written in this bit, a disconnect event is treated as a wake-up event. When Port Power is “0”, this bit is “0”.

Bit20 : WKCNNT_E
Wake on Connect Enable
When “1” is written in this bit, a connect event is treated as a wake-up event. When Port Power is “0”, this bit is “0”.

Bit19-16: Port Test Control

When this area is “0”, a port does not work in test mode.
When this area is not “0”, a port works in test mode.
(0110 b–1111 b are reserved.)

Bits	Test Mode
0000 b	Test mode not enable
0001 b	Test J_STATE
0010 b	Test K_STATE
0011 b	Test SE0_NAK
0100 b	Test Packet
0101 b	Test FORCE_ENABLE

Bit15-14: Port Indicator Control

For this macro, writing in these bits has no effect because P_INDICATOR bit of HCPARAMS register is “0”.

Bits	Meaning
00 b	Port Indicators are off
01 b	Amber
10 b	Green
11 b	Undefined

When Port Power is “0”, this bit is “0”.

Bit13: Port Owner

Indicates which host controller owns a port.
When Configured bit of CONFIGFLAG register changes from “0” to “1”, this bit becomes “0”.
When Configured bit is “0”, this bit is “1”.
1‘b0: EHCI host controller owns a port.
1‘b1: OHCI host controller owns a port.

Bit12: PP

Port Power
This bit function depends on the value in Port Power Control (PPC) area of HCSPARAMS register.

PPC	PP	Operation
0 b	1 b	Host Controller cannot control port power.
1 b	1 b/0 b	Host Controller can control port power. PP: 0 b off PP: 1 b on When port power is turned off, the port does not work and connect or disconnect events are not reported. When over-current is detected (app_prt_ovrcur_i = 1), this bit becomes “0”.

Bit11-10: Line Status

Bits[11:10]	USB State
00 b	SE0
10 b	J-State
01 b	K-State
11 b	Undefined

When Port Power is “0”, the value of this area is not defined.

Bit8: Port Reset

When “1” is written in this bit, a bus reset sequence starts. When “0” is written in this bit, a bus reset sequence ends.
1’b0: NOT Reset
1’b1: Reset
When Port Power is “0”, this bit is “0”.

Bit7: Suspend
Port Enable bit and Suspend bit define port states as below.

Bits [Port Enable, Suspend]	Port State
0Xb	Disable
10 b	Enable
11 b	Suspend

Writing “0” has no effect.
When Port Power is “0”, this bit is “0”.

Bit6: Force Port Resume
When this bit is “0”, writing “1” starts a resume.
When this bit is “1”, writing “0” finishes a resume.
1’b0: No Resume (K-State)
1’b1: Resume detect/driven on port
When Port Power is “0”, this bit is “0”.

Bit5: Over-current Change
When Over-current Active changes, this bit is set to “1”.
Writing “1” in this bit clears this bit.

Bit4: Over-current Active
1’b0: NOT Over-current state
app_prt_ovrcur_i = 0
1’b1: Over-current state
app_prt_ovrcur_i = 1

Bit3: Port Enable/Disable Change
This bit is set only when a port error changes port status into disable at the point of EOF2. When this bit is set, Port Enable/Disable bit (PORTSC.bit2) is also cleared. Writing “1” in this bit clears this bit. When Port Power is “0”, this bit is “0”.

Bit2: Port Enable/Disable

1'b0: Port Disable
1'b1: Port Enable
Only the host controller can enable a port. Software cannot enable a port by writing "1".
Only when a reset sequence verifies that a high-speed device is connected, the host controller sets this bit to "1".
When Port Power is "0", this bit is "0".

Bit1: Connect Status Change
Indicates that Current Connect Status changes at a port. Writing "1" in this bit clears this bit.
1'b0: Current Connect Status does not change.
1'b1: Current Connect Status changes.
When Port Power is "0", this bit is "0".

Bit0: Current Connect Status
Indicates the device status on port.
1'b0: Device is not connected.
1'b1: Device is connected.

22-3-3-13 Programmable Microframe Base Value Register:INSNREG00(8.1.13.)

[Base + 0x0090]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	1-microframe counter12	1-microframe counter11	1-microframe counter10	1-microframe counter9	1-microframe counter8	1-microframe counter7
Reset	0	0	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	1-microframe counter6	1-microframe counter5	1-microframe counter4	1-microframe counter3	1-microframe counter2	1-microframe counter1	1-microframe counter0	Enable
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit13-1: 1-microframe counter
1 microframe length can be changed to reduce simulation time.
It is used as 1-microframe counter.
Example: When setting value is “1D4Ch”, 1 microframe = 125 μs.

Bit0: Enable
Selects enable/disable of this register.
1’b0: Disable
1’b1: Enable
Note: Do not enable this register.

22-3-3-14 Programmable Packet Buffer OUT/IN Thresholds Register:INSNREG01(8.1.14.)

[Base + 0x0094]

Bit	31	30	29	28	27	26	25	24
Name	OUT Threshold15	OUT Threshold14	OUT Threshold13	OUT Threshold12	OUT Threshold11	OUT Threshold10	OUT Threshold9	OUT Threshold8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	OUT Threshold7	OUT Threshold6	OUT Threshold5	OUT Threshold4	OUT Threshold3	OUT Threshold2	OUT Threshold1	OUT Threshold0
Reset	0	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	IN Threshold15	IN Threshold14	IN Threshold13	IN Threshold12	IN Threshold11	IN Threshold10	IN Threshold9	IN Threshold8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	IN Threshold7	IN Threshold6	IN Threshold5	IN Threshold4	IN Threshold3	IN Threshold2	IN Threshold1	IN Threshold0
Reset	0	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit31-16: OUT Threshold

When an OUT Threshold amount of data is acquired from the system memory, transfer to USB starts.
Unit: DWORD
Minimum value:16 Bytes (0004h)
Maximum value: [INSNREG02 setting value – 2]
Example: When “80h” is set in INSNREG02, the maximum value of OUT Threshold is “7Eh”.
[31:24] cannot be written.

Bit15-0: IN Threshold

When an IN Threshold amount of data is stored in the packet buffer, transfer to memory starts.
Unit: DWORD
Minimum value: 16 Bytes (0004h)
Maximum value: [INSNREG02 setting value – 2]
Example:
When “80h” is set in INSNREG02, the maximum value of IN Threshold is “7Eh”.
[15:8] cannot be written.

- Setting values of INSNREG01 register and INSNREG02 register and operation of this macro

1. At Bulk transfer

In the case of OUT transfer

INSNREG01[31:16]	INSNREG02	Host macro operation
0080h	80h	This setting cannot be used.
0040h	80h	When 256 bytes of data is acquired from memory, transfer to USB starts.

In the case of IN transfer

INSNREG01[15:0]	INSNREG02	Host macro operation
0080h	80h	This setting cannot be used.
0040h	80h	When 256 Bytes of data is stored in packet buffer, transfer to memory starts.
0004h	80h	When 16 Bytes of data is stored in packet buffer, transfer to memory starts.

2. At Isochronous/Interrupt transfer

In the case of OUT transfer

INSNREG01[31:16]	INSNREG02	Host macro operation
0080h	80h	This setting cannot be used.
0040h	80h	When 256 bytes of data is acquired from memory, transfer to USB starts.

In the case of IN transfer

INSNREG01[15:0]	INSNREG02	Host macro operation
0080h	80h	This setting cannot be used.
0040h	80h	When 256 Bytes of data is stored in packet buffer, transfer to memory starts.

3. At Control transfer

In the case of OUT transfer

INSNREG01[31:16]	INSNREG02	Host macro operation
0080h	80h	This setting cannot be used.
0004h	80h	When 16 bytes of data is acquired from memory, transfer to USB starts.

In the case of IN transfer

INSNREG01[15:0]	INSNREG02	Host macro operation
0080h	80h	This setting cannot be used.
0004h	80h	When 16 Bytes of data is stored in packet buffer, transfer to memory starts.

- This is about the timing of transfer from packet buffer in the case of a short packet that is less than the threshold set in INSNREG01.

In the case of out transfer, after the memory on AHB bus writes a shot packet data in packet buffer, transfer starts from packet buffer to USB.
In the case of IN transfer, after USB writes a shot packet data in packet buffer, transfer starts from packet buffer to the memory on AHB bus.

22-3-3-15 Programmable Packet Buffer Depth Register:INSNREG02(8.1.15.)

[Base + 0x0098]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	INSNREG02 11	INSNREG02 10	INSNREG02 9	INSNREG02 8
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	INSNREG02 7	INSNREG02 6	INSNREG02 5	INSNREG02 4	INSNREG02 3	INSNREG02 2	INSNREG02 1	INSNREG02 0
Reset	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit11-0: INSNREG02

Defines the packet buffer depth with Dword.

22-3-3-16 Time-Available Offset Register:INSNREG03(8.1.16.)

[Base + 0x009C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	Time-Available Offset7
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	R/W
Bit	7	6	5	4	3	2	1	0
Name	Time-Available Offset6	Time-Available Offset5	Time-Available Offset4	Time-Available Offset3	Time-Available Offset2	Time-Available Offset1	Time-Available Offset0	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit8-1: Time-Available Offset. Reserved.
Do not write anything other than “00h”.

Bit0: Reserved
Do not write “1”.

22-3-3-17 Debug Register:INSNREG04(8.1.17.)

[Base + 0x00A0]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(Used only for debug)

Bit31-6: Reserved.
This bit shall remain the initial value.

Bit5:
This bit can be changed by writing.
This bit shall remain the initial value.

Bit4:.
1'b0: NAK reload fix enable
1'b1: NAK reload fix disable
This bit shall remain the initial value.

Bit3: Reserved.
This bit shall remain the initial value.

Bit2:
Port enumeration time measurement function
When this function is enabled, device chirp detection time is approximately 3.5 μs and host chirp width is approximately 400 ns.
When this function is enabled, the host macro does not detect approximately 19 μs-or-more-continued device chirp as a device chirp. When this function is enabled, device chirp shall continue for 19 μs or less.
1'b0: Disables measurement function (Normal operation)
1'b1: Enables measurement function (Simulation and the like)
For normal operation, this bit shall remain the initial value.

Bit1:
Controls writing in HCCPARAMS register.
1'b0: Disables writing.

1'b1: Enables writing.

This bit shall remain the initial value.

Bit0:
Controls writing in HCSPARAMS register.
1‘b0: Disables writing.
1‘b1: Enables writing.
This bit shall remain the initial value.

22-3-3-18 UTMI Control Status Registers:INSNREG05(8.1.18.)

[Base + 0x00A4]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

The register function changes depending on the interface system with PHY.

– UTMI Control Status Registers (When UTMI+ I/F is enabled, [ulpi_bypass_en_i = 1])

8.2. OHCI Operational Registers

22-3-3-19 HcRevision Register:HcRevision(8.2.1.)

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0
Reset	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit7-0: REV.
Revision
Indicates the number of OHCI specification revisions.

22-3-3-20 HcControl Register:HcControl(8.2.2.)

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	RWE	RWC	IR
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	HCFS1	HCFS0	BLE	CLE	IE	PLE	CBSR1	CBSR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcControl register sets the operating mode of the host controller.

Other than RemoteWakeupConnected bit, only Host Controller Driver can be changed by writing.

Bit31-11: Reserved
Read/Write 0's

Bit10: RWE
RemoteWakeupEnable
When the host controller supports remote wakeup signal, this bit enables the operation.

Bit9: RWC
RemoteWakeupConnected
This bit indicates whether the host controller supports remote wakeup signal.

Bit8: IR
InterruptRouting
This bit indicates the interrupt routing.
1'b0: Interrupt routing is normal function (INT).
1'b1: Interrupt routing is SMI.

Bit7-6: HCFS
HostControllerFunctionalState
In this area, this macro state is set. When a resume signal from DownPort is detected, this macro itself changes the state from USBSuspend to USBResume.
00: USBReset 01: USBResume 10: USBOperational 11: USBSuspend

Bit5: BLE
BulkListEnable

When “1” is written in this bit, the bulk list process is enabled.

Bit4: CLE
ControlListEnable
When “1” is written in this bit, the control list process is enabled.

Bit3: IE
IsochronousEnable
When this bit is cleared, the isochronous list process is disabled even if PeriodicListEnable is set. (In this case, interrupt ED is processed.) This macro checks this bit before processing isochronous ED.

Bit2: PLE
PeriodicListEnable
When “1” is written in this bit, the periodic (interrupt and isochronous) list process is enabled. This macro checks this bit before transferring frames periodically.

Bit1-0: CBSR
ControlBulkServiceRatio
Indicates the number of services for the control endpoint of each bulk endpoint.
N-1 indicates N times services of the control endpoint.
(Example: ‘00’ = 1 Control Endpoint, ‘11’ = 4 Control Endpoint)

22-3-3-21 HcCommandStatus Register:HcCommandStatus (8.2.3.)

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	SOC1	SOC0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	OCR	BLF	CLF	HCR
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcCommandStatus register reflects the host controller state. It is also used to receive a command from the host controller driver.

Bit31-18: Reserved
Read/Write 0's

Bit17-16: SOC
SchedulingOverrunCount
This area is incremented each time ScheduleOverrunbit of HcInterruptStatus register is set.
In counting, "11" is followed by "00".

Bit15-4: Reserved
Read/Write 0's

Bit3: OCR
OwnershipChangeRequest
When "1" is written by software, this bit sets OwnershipChange of HcInterruptStatus register. This bit is cleared by software.

Bit2: BLF
BulkListFilled
Indicates that active ED is in the bulk list.
Software or this macro itself sets this bit. It is cleared each time this macro itself starts to process the head of the bulk list.

Bit1: CLF
ControlListFilled
Indicates that active ED is in the control list.
Software or this macro itself sets this bit. It is cleared each time this macro itself starts to process the head of the control list.

Bit0: HCR
HostControllerReset
This bit is written to start software reset.
This bit is cleared by macro after reset operation is completed.

22-3-3-22 HcInterruptStatus Register:HcInterruptStatus(8.2.4.)

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	OC	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	RHSC	FNO	UE	RD	SF	WDH	SO
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcInterruptStatus register indicates the state to cause a hardware interrupt. To enable a hardware interrupt, set MIE bit of HcInterruptEnable register. The host controller sets each bit but does not clear it. Host Controller Driver bit can be cleared by writing “1” but cannot be set.

Bit31: Reserved
Read/Write 0’s

Bit30: OC
OwnershipChange
When OwnershipChangeRequest bit of HcCommandStatus register is set, this bit is set.

Bit29-7: Reserved
Read/Write 0’s

Bit6: RHSC
RootHubStatusChange
When the contents of HcRhStatus or HcRhPortStatus register change, this bit is set.

Bit5: FNO
FrameNumberOverflow
When bit 15 of FrameNumber changes, this bit is set.

Bit4: UE
UnrecoverableError
When the host controller accesses the memory and is responded to with an error or when “1” is input in sys_interrupt_i, this bit is set.

Bit3: RD
ResumeDetected
When this macro detects a resume signal at port, this bit is set.

Bit2: SF
StartofFrame
When the frame management block generates an event signal of “Start of Frame”, this bit is set.

Bit1: WDH
WritebackDoneHead
When this macro finishes writing HcDoneHead in HccaDoneHead, this bit is set.

Bit0 :SO
SchedulingOverrun
When ListProcessor confirms that ScheduleOverrun occurs, this bit is set.

22-3-3-23 HcInterruptEnable Register:HcInterruptEnable(8.2.5.)

[Base + 0x0010]

Bit	31	30	29	28	27	26	25	24
Name	MIE	OC	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	-	-	-	-	-	-
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-
Bit	7	6	5	4	3	2	1	0
Name	-	RHSC	FNO	UE	RD	SF	WDH	SO
Reset	0	0	0	0	0	0	0	0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcInterruptEnable register controls hardware interrupt generation. When an interrupt cause is enabled and MIE is set, the hardware interrupt becomes enabled.

Bit31: MIE
MasterInterruptEnable
This bit enables interrupts globally.
When “1” is written in this bit, each enable bit above can enable the interrupt.

Bit30: OC
OwnershipChangeEnable
1‘b0: No effect.
1‘b1: Enables Ownership Change interrupt.

Bit6: RHSC
RootHubStatusChangeEnable
1‘b0: No effect.
1‘b1: Enables Root Hub Status Change interrupt.

Bit5: FNO
FrameNumberOverflowEnable
1‘b0: No effect.
1‘b1: Enables Frame Number Overflow interrupt.

Bit4: UE
UnrecoverableErrorEnable
1‘b0: No effect.
1‘b1: Enables UnrecoverableError interrupt.

Bit3: RD
ResumeDetectedEnable
1‘b0: No effect.

1‘b1: Enables Resume Detected interrupt.

Bit2: SF
StartOfFrameEnable
1‘b0: No effect.
1‘b1: Enables Start Of Frame interrupt.

Bit1: WDH
WritebackDoneHeadEnable
1‘b0: No effect.
1‘b1: Enables Writeback Done Head interrupt.

Bit0: SO
SchedulingOverrunEnable
1‘b0: No effect.
1‘b1: Enables Schedule Overrun interrupt.

22-3-3-24 HcInterruptDisable Register:HcInterruptDisable(8.2.6.)

[Base + 0x0014]

Bit	31	30	29	28	27	26	25	24
Name	MIE	OC	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	RHSC	FNO	UE	RD	SF	WDH	SO
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcInterruptDisable register is combined with HcInterruptEnable register. When “1” is written, the corresponding bit of HcInterruptEnable register is cleared. When “0” is written, HcInterruptEnable register value does not change. In the case of read, not HcInterruptDisable register value but HcInterruptEnable register value is read out.

Bit31: MIE
This bit enables interrupts globally.
When “1” is written in this bit, each enable bit above can disable the interrupt.

Bit30: OC
1‘b0: No effect.
1‘b1: Disables Ownership Change interrupt.

Bit6: RHSC
1‘b0: No effect.
1‘b1: Disables Root Hub Status Change interrupt.

Bit5: FNO
1‘b0: No effect.
1‘b1: Disables Frame Number Overflow interrupt.

Bit4: UE
1‘b0: No effect.
1‘b1: Disables UnrecoverableError interrupt.

Bit3: RD
1‘b0: No effect.
1‘b1: Disables Resume Detected interrupt.

Bit2: SF
1‘b0: No effect.
1‘b1: Disables Start Of Frame interrupt.

Bit1 :WDH
1‘b0: No effect.
1‘b1: Disables Writeback Done Head interrupt.

Bit0 : SO
1‘b0: No effect.
1‘b1: Disables Schedule Overrun interrupt.

22-3-3-25 HcHCCA Register:HcHCCA(8.2.7.)

[Base + 0x0018]

Bit	31	30	29	28	27	26	25	24
Name	HCCA23	HCCA22	HCCA21	HCCA20	HCCA19	HCCA18	HCCA17	HCCA16
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	HCCA15	HCCA14	HCCA13	HCCA12	HCCA11	HCCA10	HCCA9	HCCA8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	HCCA7	HCCA6	HCCA5	HCCA4	HCCA3	HCCA2	HCCA1	HCCA0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

HcHCCA register indicates the physical address of Host Controller Communication Area. Because the alignment minimum unit is 256 bytes, lower 7–0 bits are fixed at “0”. See Chapter 4 of the OpenHCI specification for Host Controller Communication Area.

Bit31-8: HCCA
HCCA
Pointer to HCCA base address

22-3-3-26 HcPeriodCurrentED Register:HcPeriodCurrentED(8.2.8.)

[Base + 0x001C]

Bit	31	30	29	28	27	26	25	24
Name	PCED27	PCED26	PCED25	PCED24	PCED23	PCED22	PCED21	PCED20
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	PCED19	PCED18	PCED17	PCED16	PCED15	PCED14	PCED13	PCED12
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	PCED11	PCED10	PCED9	PCED8	PCED7	PCED6	PCED5	PCED4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	PCED3	PCED2	PCED1	PCED0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcPeriodCurrentED register indicates the physical address of isochronous or interrupt endpoint descriptor that is being currently processed.

Bit31-4: PCED
PeriodCurrentED
Pointer to periodic list ED that is being currently processed.

Bit3-0: Reserved
Read/Write 0’s

22-3-3-27 HcControlHeadED Register:HcControlHeadED(8.2.9.)

[Base + 0x0020]

Bit	31	30	29	28	27	26	25	24
Name	CHED27	CHED26	CHED25	CHED24	CHED23	CHED22	CHED21	CHED20
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	CHED19	CHED18	CHED17	CHED16	CHED15	CHED14	CHED13	CHED12
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	CHED11	CHED10	CHED9	CHED8	CHED7	CHED6	CHED5	CHED4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	CHED3	CHED2	CHED1	CHED0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcControlHeadED register indicates the physical address of the head endpoint descriptor of the control list.

Bit31-4: CHED
ControlHeadED
Pointer to control list head ED.

Bit3-0: Reserved
Read/Write 0’s

22-3-3-28 HcControlCurrentED Register:HcControlCurrentED(8.2.10.)

[Base + 0x0024]

Bit	31	30	29	28	27	26	25	24
Name	CCED27	CCED26	CCED25	CCED24	CCED23	CCED22	CCED21	CCED20
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	CCED19	CCED18	CCED17	CCED16	CCED15	CCED14	CCED13	CCED12
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	CCED11	CCED10	CCED9	CCED8	CCED7	CCED6	CCED5	CCED4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	CCED3	CCED2	CCED1	CCED0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcControlCurrentED register indicates the physical address of the currently processed endpoint descriptor in the control list.

Bit31-4: CCED
ControlCurentED
Pointer to control list ED that is being currently processed.
Only when ControlListEnable of HcControl register is “0”, this bit can be written. When ControlListEnable of HcControl register is “1”, this bit can be only read.

Bit3-0: Reserved
Read/Write 0’s

22-3-3-29 HcBulkHeadED Register:HcBulkHeadED(8.2.11.)

[Base + 0x0028]

Bit	31	30	29	28	27	26	25	24
Name	BHED27	BHED26	BHED25	BHED24	BHED23	BHED22	BHED21	BHED20
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	BHED19	BHED18	BHED17	BHED16	BHED15	BHED14	BHED13	BHED12
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	BHED11	BHED10	BHED9	BHED8	BHED7	BHED6	BHED5	BHED4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	BHED3	BHED2	BHED1	BHED0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcBulkHeadED register indicates the physical address of the head endpoint descriptor of the bulk list.

Bit31-4: BHED
HcBulkHeadED
Pointer to bulk list head ED.

Bit3-0: Reserved
Read/Write 0’s

22-3-3-30 HcBulkCurrentED Register:HcBulkCurrentED(8.2.12.)

[Base + 0x002C]

Bit	31	30	29	28	27	26	25	24
Name	BCED27	BCED26	BCED25	BCED24	BCED23	BCED22	BCED21	BCED20
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	BCED19	BCED18	BCED17	BCED16	BCED15	BCED14	BCED13	BCED12
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	BCED11	BCED10	BCED9	BCED8	BCED7	BCED6	BCED5	BCED4
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	BCED3	BCED2	BCED1	BCED0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcBulkCurrentED register indicates the physical address of the currently processed endpoint descriptor in the bulk list.

Bit31-4: BCED
BulkCurrentED
Pointer to bulk list ED that is being currently processed.
Only when BulkListEnable of HcControl register is “0”, this bit can be written.
When BulkListEnable of HcControl register is “1”, this bit can be only read.

Bit3-0: Reserved
Read/Write 0’s

22-3-3-31 HcDoneHead Register:HcDoneHead(8.2.13.)

[Base + 0x0030]

Bit	31	30	29	28	27	26	25	24
Name	DH27	DH26	DH25	DH24	DH23	DH22	DH21	DH20
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	DH19	DH18	DH17	DH16	DH15	DH14	DH13	DH12
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	DH11	DH10	DH9	DH8	DH7	DH6	DH5	DH4
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	DH3	DH2	DH1	DH0	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

HcDoneHead register indicates the physical address of the transfer descriptor that is completed last in Done queue.

Bit31-4: DH
DoneHead
Pointer to done head ED.

Bit3-0: Reserved
Read/Write 0’s

22-3-3-32 HcFmInterval Register:HcFmInterval(8.2.14.)

[Base + 0x0034]

Bit	31	30	29	28	27	26	25	24
Name	FIT	FSMPS14	FSMPS13	FSMPS12	FSMPS11	FSMPS10	FSMPS9	FSMPS8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	FSMPS7	FSMPS6	FSMPS5	FSMPS4	FSMPS3	FSMPS2	FSMPS1	FSMPS0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	FI13	FI12	FI11	FI10	FI9	FI8
Reset	0	0	1	0	1	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	FI7	FI6	FI5	FI4	FI3	FI2	FI1	FI0
Reset	1	1	0	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

13-0 bit value of HcFmInterval register indicates the bit time interval (SOF period of successive frames). 30-16 bit value indicates the full-speed maximum packet size.

Bit31: FIT
FrameIntervalToggle
The host controller driver toggles this bit each time the host controller driver loads a new value into FrameInterval register.

Bit30-16: FSMPS
FSLargestDataPacket
This area indicates the value loaded into Largest Data Packet Counter at the beginning of each frame.

Bit15-14: Reserved
Read/Write 0's

Bit13-0: FI
FrameInterval
This area indicates the frame length (bit time – 1).
Example: When 1 frame is 12000 bit time, 11,999 is indicated.

22-3-3-33 HcFmRemaining Register:HcFmRemaining(8.2.15.)

[Base + 0x0038]

Bit	31	30	29	28	27	26	25	24
Name	FRT	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	FR13	FR12	FR11	FR10	FR9	FR8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

HcFmRemaining register indicates the current-frame remaining time with 14-bit down counter.

Bit31: FRT
FrameRemainingToggle
When FrameRemaining is loaded, FrameIntervalToggle is loaded.

Bit30-14: Reserved
Read/Write 0’s

Bit13-0: FR
FrameRemaining
When this macro is in USBOperational state, this 14-bit area is decremented at a clock frequency of 12 MHz.
When the count value becomes “0” (at the end of a frame), FrameInterval value is loaded. Additionally, when this macro transitions to USBOperational state, the value is loaded.

22-3-3-34 HcFmNumber Register:HcFmNumber(8.2.16.)

[Base + 0x003C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	FN15	FN14	FN13	FN12	FN11	FN10	FN9	FN8
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
Name	FN7	FN6	FN5	FN4	FN3	FN2	FN1	FN0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

HcFmNumber register gives the timing reference to the event occurrence period of the host controller and the host controller driver with a 16-bit counter.

Bit31-16: Reserved
Read/Write 0’s

Bit15-0: FN
FrameNumber
This 16-bit increment counter area is incremented at the same time when FrameRemaining is loaded. In counting, “FFFFh” is followed by “0h”.

22-3-3-35 HcPeriodicStart Register:HcPeriodicStart(8.2.17.)

[Base + 0x0040]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	PS13	PS12	PS11	PS10	PS9	PS8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

With 14-bit value, HcPeriodicStart register gives the first time when the host controller should start processing the periodic list.

Bit31-14: Reserved
Read/Write 0’s

Bit13-0: PS
PeriodicStart
This bit sets the time for the list processor to start processing the periodic list in the frame.

22-3-3-36 HcLSThreshold Register:HcLSThreshold (8.2.18.)

[Base + 0x0044]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	LST11	LST10	LST9	LST8
Reset	0	0	0	0	0	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	LST7	LST6	LST5	LST4	LST3	LST2	LST1	LST0
Reset	0	0	1	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

With 11-bit value, HcLSThreshold register determines whether the host controller transmits 8-byte LS packet before EOF.

Bit31-12: Reserved
Read/Write 0’s

Bit11-0: LST
LSThreshold
This bit indicates the value to determine whether the frame management block starts low-speed transaction in the current frame.
Only when FrameRemaining is larger than this bit value, low-speed transaction starts.

22-3-3-37 HcRhDescriptorA Register:HcRhDescriptorA(8.2.19.)

[Base + 0x0048]

Bit	31	30	29	28	27	26	25	24
Name	POTPGT7	POTPGT6	POTPGT5	POTPGT4	POTPGT3	POTPGT2	POTPGT1	POTPGT0
Reset	0	0	0	0	0	0	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	NOCP	OCPM	DT	NPS	PSM
Reset	0	0	0	0	1	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	NDP7	NDP6	NDP5	NDP4	NDP3	NDP2	NDP1	NDP0
Reset	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R

HcRhDescriptorA register is the first register to indicate the root hub setting.

Bit31-24: POTPGT
PowerOnToPowerGoodTime
Indicates the time from power-on to the stable state in units of 2 ms.

Bit23-13: Reserved
Read/Write 0's

Bit12: NOCP
NoOverCurrentProtection
1'b0: Over-current status is reported.
1'b1: Over-current status is not reported.

Bit11: OCPM
OverCurrentProtectionMode
This bit is effective only when NoOverCurrentProtection is "0".
1'b0: Global over-current
1'b1: Individual over-current

Bit10: DT
DeviceType
This macro is not a compound device.

Bit9: NPS
NoPowerSwitching
1'b0: Port is controlled with the power switch.
1'b1: Port is always on.

Bit8: PSM
PowerSwitchingMode
This bit is effective only when NoPowerSwitching is "0".
1'b0: Global Switching

1’b1: Individual Switching

Bit7-0: NDP
NumberDownstreamPorts
When these bits are read out, “2” is read to indicate that 2 ports are supported. Actually, this macro supports 1 port (Port1).

22-3-3-38 HcRhDescriptorB Register:HcRhDescriptorB(8.2.20.)

[Base + 0x004C]

Bit	31	30	29	28	27	26	25	24
Name	PPCM15	PPCM14	PPCM13	PPCM12	PPCM11	PPCM10	PPCM9	PPCM8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	PPCM7	PPCM6	PPCM5	PPCM4	PPCM3	PPCM2	PPCM1	PPCM0
Reset	0	0	0	0	0	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

HcRhDescripterB register is the second register to indicate the root hub setting. This register is written at the time of initialization so that it can match the system implementation.

Bit31-16: PPCM
PortPowerControlMask
Indicates whether Global Power Control command has an effect.
This area is effective when NoPowerSwitching is “0” and PowerSwitchingMode is “1”.
1‘b0: Affected by Global Power Control command.
1‘b1: Not affected by Global Power Control command.

Relation between bits and ports
Bit 16:reserved
Bit 17:port1
Bit 18:This bit indicates port2, but actually, this macro supports port1 only.
Bit 19–31: No support. Read/write ‘0’

Bit15-0: DR
DeviceRemovable
This macro default setting enables a port to remove a device.
1‘b0: A device can be removed.
1‘b1: A device cannot be removed.

Relations between bits and ports
Bit 0: reserved
Bit 1: port1
Bit 2: This bit indicates port2, but actually, this macro supports port1 only.
Bit 3–15: No support. Read/write ‘0’

22-3-3-39 HcRhStatus Register:HcRhStatus (8.2.21.)

[Base + 0x0050]

Bit	31	30	29	28	27	26	25	24
Name	CRWE	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	OCIC	LPSC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	DRWE	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	OCI	LPS
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

HcRhStatus register consists of two parts. The lower 16 bits are the hub status field and the upper 16 bits are the hub status change field. “0” shall be always written in the reserved bits.

Bit31: CRWE
(write) ClearRemoteWakeupEnable
1’b0: No effect.
1’b1: Clears DeviceRemoteWakeupEnable.

Bit30-18: Reserved
Read/Write 0’s

Bit17: OCIC
OverCurrentIndicatorChange
This bit is set when OverCurrentIdicator changes.
(write)
1‘b0: No effect.
1‘b1: Clears this bit.

Bit16: LPSC
(read) LocalPowerStatusChange
Not supported. “0” is always read.
(write) SetGlobalPower
1‘b0: No effect.
1‘b1: Issues SetGlobalPower command.

Bit15: DRWE
(read) DeviceRemoteWakeupEnable
This bit enables port ConnectStatusChange by a remote-wakeup event.

1‘b0: Disables.
1‘b1: Enables.
(write) SetRemoteWakeupEnable
1‘b0: No effect.
1‘b1: Sets SetRemoteWakeupEnable.

Bit14-2: Reserved
Read/Write 0’s

Bit1: OCI
OverCurrentIndicator
This bit reflects the state of app_prt_ovrcur_i pin.
It is effective when NoOverCurrentProtection and OverCurrentProtectionMode of HcRhDescriptorA register are “0”.

1‘b0: No Over Current
(app_prt_ovrcur_i is “0”.)
1‘b1: Over Current condition
(app_prt_ovrcur_i is “1”.)

Bit0: LPS
(read) LocalPowerStatus
Not supported. “0” is always read.
(write) ClearGlobalPower
1‘b0: No effect.
1‘b1: Issues ClearGlobalPower command to port.

22-3-3-40 HcRhPortStatus[1] Register:HcRhPortStatus[1](8.2.22.)

[Base + 0x0054]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	PRSC	OCIC	PSSC	PESC	CSC
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	LSDA	PPS
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	PRS	POCI	PSS	PES	CCS
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

HcRhPortStatus register controls each port and reports port events. HcRhPortStatus[1] is for port 1. The lower 16 bits reflect the port status and the upper 16 bits reflect the status change bit. Some status bits have a special writing way. See the explanation about each bit for the details. If a transaction proceeds when the port status is changed by writing, writing shall wait until the transaction is completed and the resulting port status shall be written.

Bit31-21: Reserved
Read/Write 0's

Bit20: PRSC
PortResetStatusChange
This bit indicates the port reset signal completion.
(read)
1'b0: Port reset is not completed.
1'b1: Port reset is completed.
(write)
1'b0: No effect.
1'b1: Clears this bit.

Bit19: OCIC
PortOverCurrentIndicatorChange
This bit is set when OverCurrentIndicator changes.
(read)
1'b0: OverCurrentIndicator bit is not changed.
1'b1: OverCurrentIndicator bit is changed.
(write)
1'b0: No effect.
1'b1: Clears this bit.

Bit18: PSSC
PortSuspendStatusChange
This bit indicates the port resume sequence completion. When ResetStatusChange is set to “1”, this bit is cleared.
(read)
1‘b0: Port is not resumed.
1‘b1: Port is resumed completely.
(write)
1‘b0: No effect.
1‘b1: Clears this bit.

Bit17: PESC
PortEnableStatusChange
This bit is set when PortEnableStatus changes. When software changes PortEnableStatus, this bit is not set.
(read)
1‘b0: PortEnableStatus bit is not changed.
1‘b1: PortEnableStatus bit is changed.
(write)
1‘b0: No effect.
1‘b1: Clears this bit.

Bit16: CSC
ConnectStatusChange
(read) This bit indicates the detection of connect or disconnect event.
1‘b0: No event of connect or disconnect.
1‘b1: Connect or disconnected is detected.
(write)
1‘b0: No effect.
1‘b1: Clears this bit.

Bit15-10: Reserved
Read/Write 0’s

Bit9: LSDA
(read) LowSpeedDeviceAttached
This bit indicates the connected device speed. It is effective when CurrentConnectStatus is “1”.
1‘b0: Full-speed device
1‘b1: Low-speed device
(write) ClearPortPower
1‘b0: No effect.
1‘b1: Clears PortPowerStatus.

Bit8: PPS
(read) PortPowerStatus
This bit indicates the port power state independently of power switching mode. This bit is cleared when over-current state is detected (PortOverCurrentIndicator = 1).
1‘b0: Port power is off.
1‘b1: Port power is on.
Note: This bit is always read as “1” when NoPowerSwitching is set.
(write)SetPortPower

1‘b0: No effect.
1‘b1: PortPowerStatus is set.

Bit7-5: Reserved
Read/Write 0's

Bit4: PRS
(read) PortResetStatus
This bit is cleared when a reset is completed.
This bit cannot be set when CurrentConnectStatus is "0".
1'b0: Reset signal is not active.
1'b1: Reset signal is active.
(write) SetPortReset
1'b0: No effect.
1'b1: PortResetStatus is set.

Bit3: POCI
(read) PortOverCurrentIndicator
Reports the over-current condition of each port.
This bit reflects the state of app_prt_ovrcur_i pin.
It is effective only when NonOverCurrentProtection of HcRhDescriptorA register is "0" and OverCurrentProtectionMode of HcRhDescriptorA register is "1".
1'b0: This port current is normal. (app_prt_ovrcur_i = 0)
1'b1: This port has over-current. (app_prt_ovrcur_i = 1)
(write) ClearSuspendStatus
1'b0: No effect.
1'b1: Executes port resume sequence.

Bit2: PSS
(read) PortSuspendStatus
This bit cannot be set when CurrentConnectStatus is "0". When a port reset is completed or the host controller gets in USBRESUME state, this bit is cleared.
1'b0: Port is not suspended.
1'b1: Port is suspended.
(write) SetPortSuspend
When CurrentConnectStatus is "0", writing this bit cannot set PortSuspendStatus. Instead, ConnectStatusChange is set.
1'b0: No effect.
1'b1: Sets PortSuspendStatus.

Bit1: PES
(read) PortEnableStatus
When CurrentConnectStatus is "0", this bit cannot be set. This bit is set when a port reset or a port suspend is completed.
1'b0: Port is disabled.
1'b1: Port is enabled.
(write) SetPortEnable
When CurrentConnectStatus is "0", writing this bit cannot set PortEnableStatus. Instead, ConnectStatusChange is set.
1'b0: No effect.
1'b1: Sets PortEnableStatus.

Bit0: CCS
(read) CurrentConnectStatus

1‘b0: No device is connected to this port currently.
1‘b1: A device is connected to this port currently.
Note: When DeviceRemoveable is “1”, this bit is always “1”.
(write) ClearPortEnable
1‘b0: No effect.
1‘b1: Clears PortEnableStatus.

8.3. Other Registers

22-3-3-41 LINK mode setting Register:LINK mode setting(8.3.1.)

[Base + 0x0000]

Bit	31	30	29	28	27	26	25	24
Name	phy_vstatus_7_o	phy_vstatus_6_o	phy_vstatus_5_o	phy_vstatus_4_o	phy_vstatus_3_o	phy_vstatus_2_o	phy_vstatus_1_o	phy_vstatus_0_o
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	ss_ulpi_pp2vbus_o	ss_autoppd_on_overcur_en_o	ss_fladj_val_5_o	ss_fladj_val_4_o	ss_fladj_val_3_o	ss_fladj_val_2_o	ss_fladj_val_1_o	ss_fladj_val_0_o
Reset	0	1	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	ss_fladj_val_host_o5	ss_fladj_val_host_o4	ss_fladj_val_host_o3	ss_fladj_val_host_o2	ss_fladj_val_host_o1	ss_fladj_val_host_o0	ss_simulation_mode_o	ss_word_if_o
Reset	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	ss_nxt_power_state_valid_o	ss_power_state_valid_o	ss_next_power_state_o1	ss_next_power_state_o0	ss_power_state_o1	ss_power_state_o0	ss_resume_utmi_pls_dis_o	ss_utmi_backward_enb_o
Reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit initial values that are determined by external pins are as shown in “Interface Signals”.

Attention: See page 12 for the input state of interface signals.

Bit31-24: phy_vstatus_[7:0]_o
Vendor Status
Fixed at “00h” in normal use.
The initial value is determined immediately after reset by phy_vstatus_[7:0]_i external pin.
When any other value than “00h” is set in phy_vstatus_[7:0]_i, “00h” shall be set in these bits.

Bit23: ss_ulpi_pp2vbus_o
LINK mode setting
This bit shall remain the initial value.

Bit22: ss_autoppd_on_overcur_en_o
LINK mode setting
This bit shall remain the initial value.

Bit21-16: ss_fladj_val_[5:0]_o
Frame Length Adjustment Register Per Port

The same value as that of ss_fladj_val_host_i[5:0] shall be input.
Fixed at “20h” in normal use.
The initial value is determined immediately after reset by ss_fladj_val_[5:0]_i external pin.
When any other value than “20h” is set in ss_fladj_val_[5:0]_i, “20h” shall be set in these bits.

Bit15-10: ss_fladj_val_host_o[5:0]
Adjusts the frame length.

Frame Length (decimal)	FLADJ Value (decimal)
59488	0(0x00)
59504	1(0x01)
59520	2(0x02)
…	…
59984	31(0x1F)
60000	32(0x20)
…	…
60496	63(0x3F)

Fixed at “20h” in normal use.
The initial value is determined immediately after reset by ss_fladj_val_host_i[5:0] external pin.
When any other value than “20h” is set in ss_fladj_val_host_i[5:0], “20h” shall be set in these bits.

Bit9: ss_simulation_mode_o
Simulation mode signal
0: Normal mode
1: Simulation mode
When set to “1”, PHY is set to be in non-driving mode during no communication. This signal is used only for simulation.
It shall be “0” in normal use.
The initial value is determined immediately after reset by ss_simulation_mode_i external pin.
When ss_simulation_mode_i is set to “1”, this bit shall be set to “0”.

Bit8: ss_word_if_o
Word interface
Selects a data width of UTMI/UTMI+.
1‘b1: 16-bit interface (Not supported.)
1’b0: 8-bit interface
Fixed at “0” in normal use.
The initial value is determined immediately after reset by ss_word_if_i external pin.
When ss_word_if_i is set to “1”, this bit shall be set to “0”.

Bit7: ss_nxt_power_state_valid_o
LINK mode setting
This bit shall remain the initial value.

Bit6: ss_power_state_valid_o
LINK mode setting
This bit shall remain the initial value.

Bit5-4: ss_next_power_state_o[1:0]
LINK mode setting
This bit shall remain the initial value.

Bit3-2: ss_power_state_o[1:0]
LINK mode setting
This bit shall remain the initial value.

Bit1: ss_resume_utmi_pls_dis_o
Signal for debug
Fixed at “0” in normal use.
The initial value is determined immediately after reset by ss_resume_utmi_pls_dis_i external pin.
When ss_resume_utmi_pls_dis_i is set to “1”, this bit shall be set to “0”.

Bit0: ss_utmi_backward_enb_o
Signal for debug
Fixed at “0” in normal use.
The initial value is determined immediately after reset by ss_utmi_backward_enb_i external pin.
When ss_utmi_backward_enb_i is set to “1”, this bit shall be set to “0”.

22-3-3-42 PHY mode setting1 Register:PHY mode setting1(8.3.2.)

[Base + 0x0004]

Bit	31	30	29	28	27	26	25	24
Name	RESET	-	-	-	SIM	ANPD	RPDMEN	RPDPEN
Reset	0	0	0	0	0	1	1	1
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
Name	-	EOP_TIMER_EN	-	-	-	-	CONTCDR	-
Reset	0	1	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R/W	R
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	ENFLTR	RPUENXEN	RPUEN2	-	dbus16
Reset	0	0	0	1	1	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R	R/W

Bit initial values that are determined by external pins are as shown in “Interface Signals”.

Attention: See page 12 for the input state of interface signals.

Bit31: RESET
PHY bock reset signal
When “1” is written in this bit, PHY block is reset.
When “0” is written in this bit, PHY block reset is cancelled.

Bit27: SIM
Inputs simulation time reduction mode setting.
Fixed at “0” in normal use.
When “1” is written in this bit, the time from the cancellation of PHYmode setting1 register reset bit to PHY activation is reduced from 1 ms to 2.1 us.
The initial value is determined immediately after reset by SIM_I external pin.
When SIM_I is set to “1”, this bit shall be set to “0”.

Bit26: ANPD
Power-down signal of PHY block analog circuit
It shall not be active during USB communication.
The initial value is determined immediately after reset by ANPD_I external pin.
When ANPD_I is set to “1”, this bit shall be set to “0”.

Bit25: RPDMEN
D- Pull-down resistor ON/OFF control signal
Fixed at “1” in normal use.
The initial value is determined immediately after reset by RPDMEN_I external pin.

When RPDMEN_I is set to “0”, this bit shall be set to “1”.

Bit24: RPDPEN
D+ Pull-down resistor ON/OFF control signal
Fixed at “1” in normal use.
The initial value is determined immediately after reset by RPDPEN_I external pin.
When RPDPEN_I is set to “0”, this bit shall be set to “1”.

Bit14: EOP_TIMER_EN
PHY mode setting
This bit shall remain the initial value.

Bit9: CONTCDR
PHY mode setting
This bit shall remain the initial value.

Bit4: ENFLTR
PHY mode setting
This bit shall remain the initial value.

Bit3: RPUENXEN
PHY mode setting
This bit shall remain the initial value.

Bit2: RPUEN2
PHY mode setting
This bit shall remain the initial value.

Bit0: dbus16
PHY mode setting
Fixed at “0” in normal use.
The initial value is determined immediately after reset by ss_word_if_i external pin.

22-3-3-43 PHY mode setting2 Register:PHY mode setting2(8.3.3.)

[Base + 0x0008]

Bit	31	30	29	28	27	26	25	24
Name	COMPMEASREG4	COMPMEASREG3	COMPMEASREG2	COMPMEASREG1	COMPTEST4	COMPTEST3	COMPTEST2	COMPTEST1
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	23	22	21	20	19	18	17	16
Name	COMPTESTSEL	-	EARLYSYNC	REFLMASK	-	-	SQSEL	-
Reset	0	0	1	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R/W	R
Bit	15	14	13	12	11	10	9	8
Name	-	COMPDISCON6	COMPDISCON5	COMPDISCON4	COMPDISCON3	COMPDISCON2	COMPDISCON1	COMPDISCON0
Reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	COMPUPDPLS	-	COMPUPDCTL3	COMPUPDCTL2	COMPUPDCTL1	LEOP2	LEOP1	LEOP0
Reset	0	0	0	1	1	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit initial values that are determined by external pins are as shown in “Interface Signals”.

Bit31: COMPMEASREG4
Reserved

Bit30: COMPMEASREG3
Reserved

Bit29: COMPMEASREG2
Reserved

Bit28: COMPMEASREG1
Reserved

Bit27: COMPTEST4
PHY mode setting
This bit shall remain the initial value.

Bit26: COMPTEST3
PHY mode setting
This bit shall remain the initial value.

Bit25: COMPTEST2
PHY mode setting
This bit shall remain the initial value.

Bit24: COMPTEST1
PHY mode setting
This bit shall remain the initial value.

Bit23: COMPTESTSEL
PHY mode setting
This bit shall remain the initial value.

Bit21: EARLYSYNC
PHY mode setting
This bit shall remain the initial value.

Bit20: REFLMASK
PHY mode setting
This bit shall remain the initial value.

Bit17: SQSEL
PHY mode setting
This bit shall remain the initial value.

Bit14-8: COMPDISCON
PHY mode setting
This bit shall remain the initial value.

Bit7: COMPUPDPLS
PHY mode setting
This bit shall remain the initial value.

Bit5: COMPUPDCTL3
PHY mode setting
This bit shall remain the initial value.

Bit4: COMPUPDCTL2
PHY mode setting
This bit shall remain the initial value.

Bit3: COMPUPDCTL1
PHY mode setting
This bit shall remain the initial value.

Bit2-0: LEOP
PHY mode setting
This bit shall remain the initial value.

22-3-3-44 PHY mode setting3 Register:PHY mode setting3(8.3.4.)

[Base + 0x000C]

Bit	31	30	29	28	27	26	25	24
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	15	14	13	12	11	10	9	8
Name	-	-	-	-	-	-	-	-
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	HSDSEL3	HSDSEL2	HSDSEL1	HSDSEL0
Reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit initial values that are determined by external pins are as shown in “Interface Signals”.

Bit31-30: Reserved.
The value changes after reset, but read value has no meaning.

Bit29-20: Reserved
Reserved

Bit19-16: Reserved.
This bit shall remain the initial value.

Bit15-12: Reserved
Reserved

Bit11-8: Reserved.
This bit shall remain the initial value.

Bit7-5: Reserved
Reserved

Bit4: Reserved.
This bit shall remain the initial value.

Bit3-0: HSDSEL
4’b0000: Normal current value
4’b0001: 5% increase of normal current value
4’b0010: 10% increase of normal current value
Attention: Do not set anything other than those above.

22-4 Operation

This section provides USB20HC setting flow.

22-4-1 Precautions for use

The remainder of this page is intentionally left blank.

Appendix

There are various interrupt factors which are generated from the peripheral circuit in LC823450.

The following table indicates the relationship between the interrupt factor and each controller register such as Factor confirmation register, Source register, Factor clear register and Mask register.

These interrupt factors are connected to each built-in NVIC in Cortex-M3 Core0 or Core1 respectively, can be also controlled by NVIC register. The controller registers other than NVIC register are listed in the following table. As for the detail of NVIC register, please refer to "DDI0337I_cortexm3_r2p1_trm" or "DDI0403D_arm_architecture_v7m_reference_manual".

The number of interrupt vector corresponds to the number of Core in the following table.

The number of Core NVIC Bit in the following table corresponds to the Bit count, (m+(32*n)) of the following NVIC register. (m:0-31, n:0-2)

0xE000_E100-0xE000_E108 : NVIC_ISER0-2 Interrupt Set-Enable Register

0xE000_E180-0xE000_E188 : NVIC_ICER0-2 Interrupt Clear-Enable Register

0xE000_E200-0xE000_E208 : NVIC_ISPR0-2 Interrupt Set-Pending Register

0xE000_E280-0xE000_E288 : NVIC_ICPR0-2 Interrupt Clear-Pending Register

0xE000_E300-0xE000_E308 : NVIC_IABR0-2 Interrupt Active-Bit Register

0xE000_E400-0xE000_E454 : NVIC_IPR0-16 Interrupt Priority Register

In addition, the interrupt to LPDSP32 is connected to the interrupt controller of LPDSP32 through NVIC, the relationship of interrupt factor is shown in the follows.

The management number of LPDSP32 in the following table corresponds to "n" in "Interrupt factor n" of 4. INTC.

Please set it in LPDSP32 interrupt setting register 0 to 7 (0x0850 ~ 0x086C) which interrupt signal of the interrupt controller of LPDSP32 you output to from INTC.

attention) AudioBuffer module, AUDCTRL module and AudioTimer module are included in Audio Module.

Table A : Interrupt Source List Table

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
0			SP_main	SP_main				
1			Reset	Reset			Entry to the corresponding vector table	
2			NMI	Non Maskable Interrupt	ICSR register		Entry to the corresponding vector table Or, control ICSR register	
3			HardFault	Hard Fault	HFSR register		Entry to the corresponding vector table	
4			MemManage	Mem Manage Fault	MMFSR register		Entry to the corresponding vector table	FAULTMASK register
5			BusFault	Bus Fault	BFSR register		Entry to the corresponding vector table	FAULTMASK register
6			UsageFault	Usage Fault	UFSR register		Entry to the corresponding vector table	FAULTMASK register
7			Reserved	Reserved				
8			Reserved	Reserved				
9			Reserved	Reserved				
10			Reserved	Reserved				
11			SVCall	SuperVisor Call			Entry to the corresponding vector table	
12			Debug Monitor	Debug Monitor	MON_PEND Bit in DEMCR register		Entry to the corresponding vector table	TRCENA Bit in DEMCR register
13			Reserved	Reserved				
14			Pend SV	Pend SuperVisor	ICSR register		Entry to the corresponding vector table Or, control ICSR register	
15			SysTick	Systick Timmer	ICSR register		Entry to the corresponding vector table Or, control ICSR register	SYST_CSR register
16	0	0	CortexM3_00	0x4000_3000 IPIREG Bit0 INTISR0_0 1Write	0x4000_3000 IPIREG Bit0 INTISR0_0 Read		0x4000_3004 IPICLR Bit0 INTISR0CLR_0 1Write	
17	1	1	CortexM3_01	0x4000_3000 IPIREG Bit1 INTISR0_1 1Write	0x4000_3000 IPIREG Bit1 INTISR0_1 Read		0x4000_3004 IPICLR Bit1 INTISR0CLR_1 1Write	
18	2	2	CortexM3_02	0x4000_3000 IPIREG Bit2 INTISR0_2 1Write	0x4000_3000 IPIREG Bit2 INTISR0_2 Read		0x4000_3004 IPICLR Bit2 INTISR0CLR_2 1Write	
19	3	3	CortexM3_03	0x4000_3000 IPIREG Bit3 INTISR0_3 1Write	0x4000_3000 IPIREG Bit3 INTISR0_3 Read		0x4000_3004 IPICLR Bit3 INTISR0CLR_3 1Write	
20	4	4	CortexM3_10	0x4000_3000 IPIREG Bit8 INTISR1_0 1Write	0x4000_3000 IPIREG Bit8 INTISR1_0 Read		0x4000_3004 IPICLR Bit8 INTISR1CLR_0 1Write	
21	5	5	CortexM3_11	0x4000_3000 IPIREG Bit9 INTISR1_1 1Write	0x4000_3000 IPIREG Bit9 INTISR1_1 Read		0x4000_3004 IPICLR Bit9 INTISR1CLR_1 1Write	
22	6	6	CortexM3_12	0x4000_3000 IPIREG Bit10 INTISR1_2 1Write	0x4000_3000 IPIREG Bit10 INTISR1_2 Read		0x4000_3004 IPICLR Bit10 INTISR1CLR_2 1Write	
23	7	7	CortexM3_13	0x4000_3000 IPIREG Bit11 INTISR1_3 1Write	0x4000_3000 IPIREG Bit11 INTISR1_3 Read		0x4000_3004 IPICLR Bit11 INTISR1CLR_3 1Write	
24	8	8	LPDSP32 0	0x4000_3000 IPIREG Bit16 INTISR0_DSP 1Write	0x4000_3000 IPIREG Bit16 INTISR0_DSP Read		0x4000_3004 IPICLR Bit16 INTISR0CLR_DSP 1Write	
25	9	9	LPDSP32 1	0x4000_3000 IPIREG Bit17 INTISR1_DSP 1Write	0x4000_3000 IPIREG Bit17 INTISR1_DSP Read		0x4000_3004 IPICLR Bit17 INTISR1CLR_DSP 1Write	

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
26	10	10	LPDSP32 2	0x4000_3000 IPIREG Bit18 INTISR2_DSP 1Write	0x4000_3000 IPIREG Bit18 INTISR2_DSP Read		0x4000_3004 IPICLR Bit18 INTISR2CLR_DSP 1Write	
27	11	11	LPDSP32 3	0x4000_3000 IPIREG Bit19 INTISR3_DSP 1Write	0x4000_3000 IPIREG Bit19 INTISR3_DSP Read		0x4000_3004 IPICLR Bit19 INTISR3CLR_DSP 1Write	
28	12	12	WatchDogTimer0	Interrupt condition of Watch Dog Timer was established	0x4004_7004 rPTnSTS Bit0 fWTnICSTS Read		0x4004_7004 rPTnSTS Bit0 fWTnICSTS 1Write	
29	13	13	WatchDogTimer1	Interrupt condition of Watch Dog Timer was established	0x4004_8004 rPTnSTS Bit0 fWTnICSTS Read		0x4004_8004 rPTnSTS Bit0 fWTnICSTS 1Write	
30	14	14	WatchDogTimer2	Interrupt condition of Watch Dog Timer was established	0x4004_9004 rPTnSTS Bit0 fWTnICSTS Read		0x4004_9004 rPTnSTS Bit0 fWTnICSTS 1Write	
31	15	15	BasicTimer0	BTMn overflow	0x4004_7004 rPTnSTS Bit4 fBTnOVF Read		0x4004_7004 rPTnSTS Bit4 fBTnOVF 1Write	
32	16	16	BasicTimer1	BTMn overflow	0x4004_8004 rPTnSTS Bit4 fBTnOVF Read		0x4004_8004 rPTnSTS Bit4 fBTnOVF 1Write	
33	17	17	BasicTimer2	BTMn overflow	0x4004_9004 rPTnSTS Bit4 fBTnOVF Read		0x4004_9004 rPTnSTS Bit4 fBTnOVF 1Write	
34	18	18	MultipleTimer_00	Ch0 overflow Ch0 input capture/compare match B Ch0 input capture/compare match A	0x4004_3070 rMTn0TISR Bit2 fMTn0OVS Read Bit1 fMTn0IMSB Read Bit0 fMTn0IMSA Read	0x4004_304C rMTn0STS Bit2 fMTn0OVF Read Bit1 fMTn0BEVT Read Bit0 fMTn0AEVT Read	0x4004_304C rMTn0STS Bit2 fMTn0OVF 1Write Bit1 fMTn0BEVT 1Write Bit0 fMTn0AEVT 1Write	0x4004_306C rMTn0TIER Bit2 fMTn0OVE 0Write Bit1 fMTn0IMEB 0Write Bit0 fMTn0 IMEA 0Write
35	19	19	MultipleTimer_01	Ch1 overflow Ch1 input capture/compare match B Ch1 input capture/compare match A	0x4004_30B0 rMTn1TISR Bit2 fMTn1OVS Read Bit1 fMTn1IMSB Read Bit0 fMTn1IMSA Read	0x4004_308C rMTn1STS Bit2 fMTn1OVF Read Bit1 fMTn1BEVT Read Bit0 fMTn1AEVT Read	0x4004_308C rMTn1STS Bit2 fMTn1OVF 1Write Bit1 fMTn1BEVT 1Write Bit0 fMTn1AEVT 1Write	0x4004_30AC rMTn1TIER Bit2 fMTn1OVE 0Write Bit1 fMTn1IMEB 0Write Bit0 fMTn1 IMEA 0Write
36	20	20	MultipleTimer_10	Ch0 overflow Ch0 input capture/compare match B Ch0 input capture/compare match A	0x4004_4070 rMTn0TISR Bit2 fMTn0OVS Read Bit1 fMTn0IMSB Read Bit0 fMTn0IMSA Read	0x4004_404C rMTn0STS Bit2 fMTn0OVF Read Bit1 fMTn0BEVT Read Bit0 fMTn0AEVT Read	0x4004_404C rMTn0STS Bit2 fMTn0OVF 1Write Bit1 fMTn0BEVT 1Write Bit0 fMTn0AEVT 1Write	0x4004_406C rMTn0TIER Bit2 fMTn0OVE 0Write Bit1 fMTn0IMEB 0Write Bit0 fMTn0 IMEA 0Write
37	21	21	MultipleTimer_11	Ch1 overflow Ch1 input capture/compare match B Ch1 input capture/compare match A	0x4004_40B0 rMTn1TISR Bit2 fMTn1OVS Read Bit1 fMTn1IMSB Read Bit0 fMTn1IMSA Read	0x4004_408C rMTn1STS Bit2 fMTn1OVF Read Bit1 fMTn1BEVT Read Bit0 fMTn1AEVT Read	0x4004_408C rMTn1STS Bit2 fMTn1OVF 1Write Bit1 fMTn1BEVT 1Write Bit0 fMTn1AEVT 1Write	0x4004_40AC rMTn1TIER Bit2 fMTn1OVE 0Write Bit1 fMTn1IMEB 0Write Bit0 fMTn1 IMEA 0Write
38	22	22	MultipleTimer_20	Ch0 overflow Ch0 input capture/compare match B Ch0 input capture/compare match A	0x4004_5070 rMTn0TISR Bit2 fMTn0OVS Read Bit1 fMTn0IMSB Read Bit0 fMTn0IMSA Read	0x4004_504C rMTn0STS Bit2 fMTn0OVF Read Bit1 fMTn0BEVT Read Bit0 fMTn0AEVT Read	0x4004_504C rMTn0STS Bit2 fMTn0OVF 1Write Bit1 fMTn0BEVT 1Write Bit0 fMTn0AEVT 1Write	0x4004_506C rMTn0TIER Bit2 fMTn0OVE 0Write Bit1 fMTn0IMEB 0Write Bit0 fMTn0 IMEA 0Write
39	23	23	MultipleTimer_21	Ch1 overflow Ch1 input capture/compare match B Ch1 input capture/compare match A	0x4004_50B0 rMTn1TISR Bit2 fMTn1OVS Read Bit1 fMTn1IMSB Read Bit0 fMTn1IMSA Read	0x4004_508C rMTn1STS Bit2 fMTn1OVF Read Bit1 fMTn1BEVT Read Bit0 fMTn1AEVT Read	0x4004_508C rMTn1STS Bit2 fMTn1OVF 1Write Bit1 fMTn1BEVT 1Write Bit0 fMTn1AEVT 1Write	0x4004_50AC rMTn1TIER Bit2 fMTn1OVE 0Write Bit1 fMTn1IMEB 0Write Bit0 fMTn1 IMEA 0Write
40	24	24	MultipleTimer_30	Ch0 overflow Ch0 input capture/compare match B Ch0 input capture/compare match A	0x4004_6070 rMTn0TISR Bit2 fMTn0OVS Read Bit1 fMTn0IMSB Read Bit0 fMTn0IMSA Read	0x4004_604C rMTn0STS Bit2 fMTn0OVF Read Bit1 fMTn0BEVT Read Bit0 fMTn0AEVT Read	0x4004_604C rMTn0STS Bit2 fMTn0OVF 1Write Bit1 fMTn0BEVT 1Write Bit0 fMTn0AEVT 1Write	0x4004_606C rMTn0TIER Bit2 fMTn0OVE 0Write Bit1 fMTn0IMEB 0Write Bit0 fMTn0 IMEA 0Write
41	25	25	MultipleTimer_31	Ch1 overflow Ch1 input capture/compare match B Ch1 input capture/compare match A	0x4004_60B0 rMTn1TISR Bit2 fMTn1OVS Read Bit1 fMTn1IMSB Read Bit0 fMTn1IMSA Read	0x4004_608C rMTn1STS Bit2 fMTn1OVF Read Bit1 fMTn1BEVT Read Bit0 fMTn1AEVT Read	0x4004_608C rMTn1STS Bit2 fMTn1OVF 1Write Bit1 fMTn1BEVT 1Write Bit0 fMTn1AEVT 1Write	0x4004_60AC rMTn1TIER Bit2 fMTn1OVE 0Write Bit1 fMTn1IMEB 0Write Bit0 fMTn1 IMEA 0Write

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
42	26		UsbHost EHCI	[Interrupt on Async Advance] Host Controller processed the following Async Schedule by writing "1" to Interrupt on Async Advance Doorbell Bit in USBCMD register. [Host System Error] ERROR response was received when Host Controller accessed to memory. Or, "1" was input to F47sys_interrupt_i. [Frame List Rollover] Frame List Index returned to 0 from maximum. [Port Change Detect] The port which was set to "0" in Port Owner Bit was changed to "1". Force Port Resume Bit was changed from "0" to "1" by the detection of J-K on the suspended port. EHCI released the port by the wrting "1" to Port Owner Bit [USBERRINT (USB Error Interrupt)] USB transfer was exited with error If IOC Bit was set at the TD which Error interrupt generated, both of USBERRINT and USBINT generate. [USBINT] Transfer Descriptor which IOC Bit was set became retire after the USB transfer completed Short packet was detected		0x4000_2014 USBSTS Bit5 Interrupt on Async Advance Read Bit4 Host System Error Read Bit3 Frame List Rollover Read Bit2 Port Change Detect Read Bit1 USBERRINT Read Bit0 USBINT Read	0x4000_2014 USBSTS Bit5 Interrupt on Async Advance 1Write Bit4 Host System Error 1Write Bit3 Frame List Rollover 1Write Bit2 Port Change Detect 1Write Bit1 USBERRINT 1Write Bit0 USBINT 1Write	0x4000_2018 USBINTR Bit5 Interrupt on Async Advance Enable 0Write Bit4 Host System Error Enable 0Write Bit3 Frame List Rollover Enable 0Write Bit2 Port Change Interrupt Enable 0Write Bit1 USB Error Interrupt Enable 0Write Bit0 USB Interrupt Enable 0Write
43	27		UsbHost OHCI	[OC (OwnershipChange)] WnershipChangeRequest Bit in HcCommandStatus register was set. [RHSC (RootHubStatusChange)] Content of HcRhStatus or HcRhPortStatus register was changed. [FNO (FrameNumberOverflow)] Bit15 in FrameNumber was changed. [UE (UnrecoverableError)] ERROR response was received when Host Controller accessed to memory Or, "1" was input in F47sys_interrupt_i. [RD (ResumeDetected)] Resume signal was detected. [SF (StartofFrame)] Event signal of "Start of Frame" was generated by Frame Managemanet block. [WDH (WritebackDoneHead)] Writing HcDoneHead was finished. [SO (SchedulingOverrun)] Generation of ScheduleOverrun was judged by ListProcessor.		0x4000_220C HcInterruptStatus Bit30 OC Read Bit6 RHSC Read Bit5 FNO Read Bit4 UE Read Bit3 RD Read Bit2 SF Read Bit1 WHD Read Bit0 SO Read	0x4000_220C HcInterruptStatus Bit30 OC 1Write Bit6 RHSC 1Write Bit5 FNO 1Write Bit4 UE 1Write Bit3 RD 1Write Bit2 SF 1Write Bit1 WHD 1Write Bit0 SO 1Write	Mask Disable 0x4000_2210 HcInterruptEnable Bit31 MIE Master 1Write Bit30 OC 1Write Bit6 RHSC 1Write Bit5 FNO 1Write Bit4 UE 1Write Bit3 RD 1Write Bit2 SF 1Write Bit1 WHD 1Write Bit0 SO 1Write Mask Enable 0x4000_2214 HcInterruptDisable Bit31 MIE Master 1Write Bit30 OC 1Write Bit6 RHSC 1Write Bit5 FNO 1Write Bit4 UE 1Write Bit3 RD 1Write Bit2 SF 1Write Bit1 WHD 1Write Bit0 SO 1Write

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
44	28		SerialFlash	Transfer completed Receive FIFO Underflow Receive FIFO Overflow Receive FIFO remaining is 8 byte Receive FIFO remaining is over 4 byte Receive FIFO remaining is 0 byte Transmit FIFO Overflow Transmit FIFO remaining is 8 byte Transmit FIFO remaining is under 5 byte Transmit FIFO remaining is 0 byte		0x4000_1014 SF_STATUS Bit16 FIN Read Bit13 R_U_ERR Read Bit12 R_O_ERR Read Bit10 R_FULL Read Bit9 R_HALF Read Bit8 R_EMP Read Bit4 T_O_ERR Read Bit2 T_FULL Read Bit1 T_HALF Read Bit0 T_EMP Read	0x4000_1014 SF_STATUS Bit16 FIN 0Write 0x4000_1020 SF_SRSTB Bit0 SRSTB 0Write (Reset process) 0x4000_1020 SF_SRSTB Bit0 SRSTB 0Write (Reset process) 0x4000_101C SF_FIFO_NUM Bit11-8 R_NUM other than 8 0x4000_101C SF_FIFO_NUM Bit11-8 R_NUM < 4 0x4000_101C SF_FIFO_NUM Bit11-8 R_NUM > 0 0x4000_1020 SF_SRSTB Bit0 SRSTB 0Write (Reset process) 0x4000_101C SF_FIFO_NUM Bit3-0 T_NUM other than 8 0x4000_101C SF_FIFO_NUM Bit3-0 T_NUM > 4 0x4000_101C SF_FIFO_NUM Bit3-0 T_NUM > 0	0x4000_1018 SF_IRQEN Bit16 FIN_EN 0Write 0x4000_1018 SF_IRQEN Bit13 R_U_ERR_EN 0Write 0x4000_1018 SF_IRQEN Bit12 R_O_ERR_EN 0Write 0x4000_1018 SF_IRQEN Bit10 R_FULL_EN 0Write 0x4000_1018 SF_IRQEN Bit9 R_HALF_EN 0Write 0x4000_1018 SF_IRQEN Bit8 R_EMP_EN 0Write 0x4000_1018 SF_IRQEN Bit4 T_O_ERR_EN 0Write 0x4000_1018 SF_IRQEN Bit2 T_FULL_EN 0Write 0x4000_1018 SF_IRQEN Bit1 T_HALF_EN 0Write 0x4000_1018 SF_IRQEN Bit0 T_EMP_EN 0Write
45	29	26	DmaController	DMA error DMA transfer completed	0x4000_4004 DMACIntTCStatus Bit7-0 IntTCStatus Read 0x4000_400C DMACIntErrorStatus Bit7-0 IntErrorStatus Read	0x4000_4014 DMACRawIntTCStatus Bit7-0 RawIntStatus Read 0x4000_4018 DMACRawIntErrorStatus Bit7-0 RawIntErrorStatus Read	0x4000_4008 DMACIntTCClear Bit7-0 IntTCClear 1Write 0x4000_4010 DMACIntErrClr Bit7-0 IntErrClr 1Write	0x4000_4110 DMACC0 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_4130 DMACC1 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_4150 DMACC2 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_4170 DMACC3 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_4190 DMACC4 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_41B0 DMACC5 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_41D0 DMACC6 Configuration Bit15 ITC 0Write Bit14 IE 0Write 0x4000_41F0 DMACC7 Configuration Bit15 ITC 0Write Bit14 IE 0Write
46	30		SDCardSync0	SD Card Ch0 Synchronization interrupt	0x4004_A038 SD_INFO1 Read 0x4004_A03C SD_INFO2 Read 0x4004_A06C SDIO_INFO Read		0x4004_A038 SD_INFO1 0Write 0x4004_A03C SD_INFO2 0Write 0x4004_A06C SDIO_INFO 0Write	0x4004_A040 SD_INFO1_MASK 1Write 0x4004_A044 SD_INFO2_MASK 1Write 0x4004_A070 SDIO_INFO_MASK 1Write

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
47	31		SDCardSync1	SD Card Ch1 Synchronization interrupt	0x4004_B038 SD_INFO1 Read 0x4004_B03C SD_INFO2 Read 0x4004_B06C SDIO_INFO Read		0x4004_B038 SD_INFO1 0Write 0x4004_B03C SD_INFO2 0Write 0x4004_B06C SDIO_INFO 0Write	0x4004_B040 SD_INFO1_MASK 1Write 0x4004_B044 SD_INFO2_MASK 1Write 0x4004_B070 SDIO_INFO_MASK 1Write
48	32		SDCardSync2	SD Card Ch2 Synchronization interrupt	0x4004_C038 SD_INFO1 Read 0x4004_C03C SD_INFO2 Read 0x4004_C06C SDIO_INFO Read		0x4004_C038 SD_INFO1 0Write 0x4004_C03C SD_INFO2 0Write 0x4004_C06C SDIO_INFO 0Write	0x4004_C040 SD_INFO1_MASK 1Write 0x4004_C044 SD_INFO2_MASK 1Write 0x4004_C070 SDIO_INFO_MASK 1Write
49	33		SDCardAsync0	SD Card Ch0 SDIO Asynchronization interrupt	0x4004_A06C SDIO_INFO Read		0x4004_A06C SDIO_INFO 0Write	0x4004_A070 SDIO_INFO_MASK 1Write
50	34		SDCardAsync1	SD Card Ch1 SDIO Asynchronization interrupt	0x4004_B06C SDIO_INFO Read		0x4004_B06C SDIO_INFO 0Write	0x4004_B070 SDIO_INFO_MASK 1Write
51	35		SDCardAsync2	SD Card Ch2 SDIO Asynchronization interrupt	0x4004_C06C SDIO_INFO Read		0x4004_C06C SDIO_INFO 0Write	0x4004_C070 SDIO_INFO_MASK 1Write
52	36		MemoryStick	-	-	-	-	-
53	37		MemoryStickIns	-	-	-	-	-

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
54	38	27	DspCmdInterface	M3_0: 0x4000_6044 HOST0_STS7_REG Write 0x4000_6040 HOST0_STS6_REG Write 0x4000_603C HOST0_STS5_REG Write 0x4000_6038 HOST0_STS4_REG Write 0x4000_6034 HOST0_STS3_REG Write 0x4000_6030 HOST0_STS2_REG Write 0x4000_602C HOST0_STS1_REG Write 0x4000_6028 HOST0_STS0_REG Write M3_1: 0x4000_6064 HOST1_STS7_REG Write 0x4000_6060 HOST1_STS6_REG Write 0x4000_605C HOST1_STS5_REG Write 0x4000_6058 HOST1_STS4_REG Write 0x4000_6054 HOST1_STS3_REG Write 0x4000_6050 HOST1_STS2_REG Write 0x4000_604C HOST1_STS1_REG Write 0x4000_6048 HOST1_STS0_REG Write DSP: 0x4000_6004 DSP_COMM1_REG Write 0x4000_6000 DSP_COMM0_REG Write 0x4000_6024 DSP_PARA7_REG Write 0x4000_6020 DSP_PARA6_REG Write 0x4000_601C DSP_PARA5_REG Write 0x4000_6018 DSP_PARA4_REG Write 0x4000_6014 DSP_PARA3_REG Write 0x4000_6010 DSP_PARA2_REG Write 0x4000_600C DSP_PARA1_REG Write 0x4000_6008 DSP_PARA0_REG Write	M3_0:0x4000_6088 HOST0_STSIRQ_REG Bit7 HOST0_STSIRQ7 Read Bit6 HOST0_STSIRQ6 Read Bit5 HOST0_STSIRQ5 Read Bit4 HOST0_STSIRQ4 Read Bit3 HOST0_STSIRQ3 Read Bit2 HOST0_STSIRQ2 Read Bit1 HOST0_STSIRQ1 Read Bit0 HOST0_STSIRQ0 Read M3_1:0x4000_608C HOST1_STSIRQ_REG Bit7 HOST1_STSIRQ7 Read Bit6 HOST1_STSIRQ6 Read Bit5 HOST1_STSIRQ5 Read Bit4 HOST1_STSIRQ4 Read Bit3 HOST1_STSIRQ3 Read Bit2 HOST1_STSIRQ2 Read Bit1 HOST1_STSIRQ1 Read Bit0 HOST1_STSIRQ0 Read DSP:0x4000_6080 DSP_COMMIRQ_REG Bit1 DSP_COMMIRQ1 Read Bit0 DSP_COMMIRQ0 Read 0x4000_6084 DSP_PARAIRQ_REG Bit7 DSP_PARAIRQ7 Read Bit6 DSP_PARAIRQ6 Read Bit5 DSP_PARAIRQ5 Read Bit4 DSP_PARAIRQ4 Read Bit3 DSP_PARAIRQ3 Read Bit2 DSP_PARAIRQ2 Read Bit1 DSP_PARAIRQ1 Read Bit0 DSP_PARAIRQ0 Read	M3_0: 0x4000_6078 HOST0_STSIRQSRC_REG Bit7 HOST0_STSIRQSRC7 Read Bit6 HOST0_STSIRQSRC6 Read Bit5 HOST0_STSIRQSRC5 Read Bit4 HOST0_STSIRQSRC4 Read Bit3 HOST0_STSIRQSRC3 Read Bit2 HOST0_STSIRQSRC2 Read Bit1 HOST0_STSIRQSRC1 Read Bit0 HOST0_STSIRQSRC0 Read M3_1: 0x4000_607C HOST1_STSIRQSRC_REG Bit7 HOST1_STSIRQSRC7 Read Bit6 HOST1_STSIRQSRC6 Read Bit5 HOST1_STSIRQSRC5 Read Bit4 HOST1_STSIRQSRC4 Read Bit3 HOST1_STSIRQSRC3 Read Bit2 HOST1_STSIRQSRC2 Read Bit1 HOST1_STSIRQSRC1 Read Bit0 HOST1_STSIRQSRC0 Read DSP: 0x4000_6074 DSP_PARAIRQSRC_REG Bit7 DSP_PARAIRQRC7 Read Bit6 DSP_PARAIRQRC6 Read Bit5 DSP_PARAIRQRC5 Read Bit4 DSP_PARAIRQRC4 Read Bit3 DSP_PARAIRQRC3 Read Bit2 DSP_PARAIRQRC2 Read Bit1 DSP_PARAIRQRC1 Read Bit0 DSP_PARAIRQRC0 Read	M3_0: 0x4000_6028 HOST0_STS0_REG Read 0x4000_602C HOST0_STS1_REG Read 0x4000_6030 HOST0_STS2_REG Read 0x4000_6034 HOST0_STS3_REG Read 0x4000_6038 HOST0_STS4_REG Read 0x4000_603C HOST0_STS5_REG Read 0x4000_6040 HOST0_STS6_REG Read 0x4000_6044 HOST0_STS7_REG Read M3_1: 0x4000_6048 HOST1_STS0_REG Read 0x4000_604C HOST1_STS1_REG Read 0x4000_6050 HOST1_STS2_REG Read 0x4000_6054 HOST1_STS3_REG Read 0x4000_6058 HOST1_STS4_REG Read 0x4000_605C HOST1_STS5_REG Read 0x4000_6060 HOST1_STS6_REG Read 0x4000_6064 HOST1_STS7_REG Read DSP: 0x4000_6000 DSP_COMM0_REG Read 0x4000_6004 DSP_COMM1_REG Read 0x4000_6008 DSP_PARA0_REG Read 0x4000_600C DSP_PARA1_REG Read 0x4000_6010 DSP_PARA2_REG Read 0x4000_6014 DSP_PARA3_REG Read 0x4000_6018 DSP_PARA4_REG Read 0x4000_601C DSP_PARA5_REG Read 0x4000_6020 DSP_PARA6_REG Read 0x4000_6024 DSP_PARA7_REG Read	M3_0: 0x4000_606C HOST0_STSIRQEN_REG Bit7 HOST0_STSIRQEN7 0Write Bit6 HOST0_STSIRQEN6 0Write Bit5 HOST0_STSIRQEN5 0Write Bit4 HOST0_STSIRQEN4 0Write Bit3 HOST0_STSIRQEN3 0Write Bit2 HOST0_STSIRQEN2 0Write Bit1 HOST0_STSIRQEN1 0Write Bit0 HOST0_STSIRQEN0 0Write M3_1: 0x4000_6070 HOST1_STSIRQEN_REG Bit7 HOST1_STSIRQEN7 0Write Bit6 HOST1_STSIRQEN6 0Write Bit5 HOST1_STSIRQEN5 0Write Bit4 HOST1_STSIRQEN4 0Write Bit3 HOST1_STSIRQEN3 0Write Bit2 HOST1_STSIRQEN2 0Write Bit1 HOST1_STSIRQEN1 0Write Bit0 HOST1_STSIRQEN0 0Write DSP: 0x4000_6068 DSP_PARAIRQEN_REG Bit7 DSP_PARAIRQEN7 0Write Bit6 DSP_PARAIRQEN6 0Write Bit5 DSP_PARAIRQEN5 0Write Bit4 DSP_PARAIRQEN4 0Write Bit3 DSP_PARAIRQEN3 0Write Bit2 DSP_PARAIRQEN2 0Write Bit1 DSP_PARAIRQEN1 0Write Bit0 DSP_PARAIRQEN0 0Write

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
55	39		AD Converter	Interrupt by AD conversion completed	0x4008_702C rADSTS Bit0 fADCMPL Read		0x4008_702C rADSTS Bit0 fADCMPL 1Write	
56	40		SPI	Continuous transfer completed Receive FIFO Underflow error Transmit FIFO Underflow error Receive FIFO Overwrite error Transmit FIFO Overwrite error Receive FIFO Data-full Transmit FIFO Data-full Receive FIFO Data-empty Transmit FIFO Data-empty Receive FIFO Water-level-match Transmit FIFO Water-level-match Receive register Overwrite error Receive register Overrun error SPI transfer completed	0x4008_802C SPI_INT Bit13 I_BURST_END Read Bit12 I_RxORE Read Bit11 I_TxORE Read Bit10 I_RxOWE Read Bit9 I_TxOWE Read Bit8 I_RxFULL Read Bit7 I_TxFULL Read Bit6 I_RxEMP Read Bit5 I_TxEMP Read Bit4 I_RxWLM Read Bit3 I_TxWLM Read Bit2 I_ROWE Read Bit1 I_OVE Read Bit0 I_SPIF Read	0x4008_8014 SPI_ISR Bit13 BURST_END Read Bit12 RxORE Read Bit11 TxORE Read Bit10 RxOWE Read Bit9 TxOWE Read Bit8 RxFULL Read Bit7 TxFULL Read Bit6 RxEMP Read Bit5 TxEMP Read Bit4 RxWLM Read Bit3 TxWLM Read Bit2 ROWE Read Bit1 OVE Read Bit0 SPIF Read	0x4008_8014 SPI_ISR Bit13 BURST_END 1Write Bit12 RxORE 1Write Bit11 TxORE 1Write Bit10 RxOWE 1Write Bit9 TxOWE 1Write Bit8 RxFULL 1Write Bit7 TxFULL 1Write Bit6 RxEMP 1Write Bit5 TxEMP 1Write Bit4 RxWLM 1Write Bit3 TxWLM 1Write Bit2 ROWE 1Write Bit1 OVE 1Write Bit0 SPIF 1Write	0x4008_8028 SPI_MSK Bit13 M_BURST_END 0Write Bit12 M_RxORE 0Write Bit11 M_TxORE 0Write Bit10 M_RxOWE 0Write Bit9 M_TxOWE 0Write Bit8 M_RxFULL 0Write Bit7 M_TxFULL 0Write Bit6 M_RxEMP 0Write Bit5 M_TxEMP 0Write Bit4 M_RxWLM 0Write Bit3 M_TxWLM 0Write Bit2 M_ROWE 0Write Bit1 M_OVE 0Write Bit0 M_SPIF 0Write
57	41		I2C0	Start condition + Address generation sequence completed Stop condition generation sequence completed 1byte Transmit sequence completed 1byte Receive sequence completed State clear sequence completed		0x4008_9004 I2CSTR Bit6 IREQ Read	0x4008_9004 I2CSTR Bit6 IREQ 0Write or 0x4008_9000 I2CCTL Bit12 BTRIG 1Write (start transfer or receive and IREQ clear)	0x4008_9000 I2CCTL Bit4 IREQEN 0Write
58	42		I2C1	Start condition + Address generation sequence completed Stop condition generation sequence completed 1byteTransmit sequence completed 1byte Receive sequence completed State clear sequence completed		0x4008_A004 I2CSTR Bit6 IREQ Read	0x4008_A004 I2CSTR Bit6 IREQ 0Write or 0x4008_A000 I2CCTL Bit12 BTRIG 1Write (start transfer or receive and IREQ clear)	0x4008_A000 I2CCTL Bit4 IREQEN 0Write
59	43		UART0	Receive FIFO Water-level-match Receive FIFO Underflow error Receive FIFO Overwrite error Transmit FIFO Water-level-match Transmit FIFO Overwrite error Bus access error Framing error Receive register Overwrite error Receive parity error Transmit register exit Receive register exit	0x4008_B024 UART0_UINT Bit14 RxWLM_INT Read Bit13 RxUFE_INT Read Bit12 RxOWE_INT Read Bit10 TxWLM_INT Read Bit8 TxOWE_INT Read Bit7 BUSERR_INT Read Bit4 FE_INT Read Bit3 ROWE_INT Read Bit2 PE_INT Read Bit1 UARTTF_INT Read Bit0 UARTRF_INT Read	0x4008_B018 UART0_UISR Bit14 RxWLM_LS Read Bit13 RUFE_IS Read Bit12 RxOWE_IS Read Bit10 TxWLM_IS Read Bit8 TxOWE_IS Read Bit7 BUSERR Read Bit4 FE Read Bit3 ROWE Read Bit2 PE Read Bit1 UARTTF Read Bit0 UARTRF Read	0x4008_B018 UART0_UISR Bit14 RxWLM_LS 1Write Bit13 RUFE_IS 1Write Bit12 RxOWE_IS 1Write Bit10 TxWLM_IS 1Write Bit8 TxOWE_IS 1Write Bit7 BUSERR 1Write Bit4 FE 1Write Bit3 ROWE 1Write Bit2 PE 1Write Bit1 UARTTF 1Write Bit0 UARTRF 1Write	0x4008_B020 UART0_UIEN Bit14 RxWLM_IEN 0Write Bit13 RxUFE_IEN 0Write Bit12 RxOWE_IEN 0Write Bit10 TxWLM_IEN 0Write Bit8 TxOWE_IEN 0Write Bit7 BUSERR_IEN 0Write Bit4 FE_IEN 0Write Bit3 ROWE_IEN 0Write Bit2 PE_IEN 0Write Bit1 UARTTF_IEN 0Write Bit0 UARTRF_IEN 0Write

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
60	44		UART1	Receive FIFO Water-level-match Receive FIFO Underflow error Receive FIFO Overwrite error Transmit FIFO Water-level-match Transmit FIFO Overwrite error Bus access error Framing error Receive register Overwrite error Receive parity error Transmit register completed Receive register completed	0x4008_C024 UART1_UINT Bit14 RxWLM_INT Read Bit13 RxUFE_INT Read Bit12 RxOWE_INT Read Bit10 TxWLM_INT Read Bit8 TxOWE_INT Read Bit7 BUSERR_INT Read Bit4 FE_INT Read Bit3 ROWE_INT Read Bit2 PE_INT Read Bit1 UARTTF_INT Read Bit0 UARTRF_INT Read	0x4008_C018 UART1_UISR Bit14 RxWLM_LS Read Bit13 RUFE_IS Read Bit12 RxOWE_IS Read Bit10 TxWLM_IS Read Bit8 TxOWE_IS Read Bit7 BUSERR Read Bit4 FE Read Bit3 ROWE Read Bit2 PE Read Bit1 UARTTF Read Bit0 UARTRF Read	0x4008_C018 UART1_UISR Bit14 RxWLM_LS 1Write Bit13 RUFE_IS 1Write Bit12 RxOWE_IS 1Write Bit10 TxWLM_IS 1Write Bit8 TxOWE_IS 1Write Bit7 BUSERR 1Write Bit4 FE 1Write Bit3 ROWE 1Write Bit2 PE 1Write Bit1 UARTTF 1Write Bit0 UARTRF 1Write	0x4008_C020 UART1_UIEN Bit14 RxWLM_IEN 0Write Bit13 RxUFE_IEN 0Write Bit12 RxOWE_IEN 0Write Bit10 TxWLM_IEN 0Write Bit8 TxOWE_IEN 0Write Bit7 BUSERR_IEN 0Write Bit4 FE_IEN 0Write Bit3 ROWE_IEN 0Write Bit2 PE_IEN 0Write Bit1 UARTTF_IEN 0Write Bit0 UARTRF_IEN 0Write
61	45		UART2	Receive FIFO Water-level-match Receive FIFO Underflow error Receive FIFO Overwrite error Transmit FIFO Water-level-match Transmit FIFO Overwrite error Bus access error Framing error Receive register Overwrite error Receive parity error Transmit register completed Receive register completed	0x4008_D024 UART2_UINT Bit14 RxWLM_INT Read Bit13 RxUFE_INT Read Bit12 RxOWE_INT Read Bit10 TxWLM_INT Read Bit8 TxOWE_INT Read Bit7 BUSERR_INT Read Bit4 FE_INT Read Bit3 ROWE_INT Read Bit2 PE_INT Read Bit1 UARTTF_INT Read Bit0 UARTRF_INT Read	0x4008_D018 UART2_UISR Bit14 RxWLM_LS Read Bit13 RUFE_IS Read Bit12 RxOWE_IS Read Bit10 TxWLM_IS Read Bit8 TxOWE_IS Read Bit7 BUSERR Read Bit4 FE Read Bit3 ROWE Read Bit2 PE Read Bit1 UARTTF Read Bit0 UARTRF Read	0x4008_D018 UART2_UISR Bit14 RxWLM_LS 1Write Bit13 RUFE_IS 1Write Bit12 RxOWE_IS 1Write Bit10 TxWLM_IS 1Write Bit8 TxOWE_IS 1Write Bit7 BUSERR 1Write Bit4 FE 1Write Bit3 ROWE 1Write Bit2 PE 1Write Bit1 UARTTF 1Write Bit0 UARTRF 1Write	0x4008_D020 UART2_UIEN Bit14 RxWLM_IEN 0Write Bit13 RxUFE_IEN 0Write Bit12 RxOWE_IEN 0Write Bit10 TxWLM_IEN 0Write Bit8 TxOWE_IEN 0Write Bit7 BUSERR_IEN 0Write Bit4 FE_IEN 0Write Bit3 ROWE_IEN 0Write Bit2 PE_IEN 0Write Bit1 UARTTF_IEN 0Write Bit0 UARTRF_IEN 0Write
62	46		RTC	Periodic Alerm Update Cycle	0x4008_E044 RTCSTAT Bit6 PF Read Bit5 AF Read Bit4 UF Read		0x4008_E044 RTCSTAT Bit7 PCLR 0Write Bit6 PF 1Write Bit5 AF 1Write Bit4 UF 1Write	0x4008_E044 RTCSTAT Bit7 PCLR 0Write 0x4008_E03C RTCINT Bit6 PIE 0Write Bit5 AIE 0Write Bit4 UIE 0Write
63	47		RTCKEY	Signal which was set in the following Bit was input to KEYINT0~2. 0x4008_E0094 KEYPOL1 Bit2 POL1_2 Bit1 POL1_1 Bit0 POL1_0	0x4008_E0A4 KEYINTMON1 Bit2 IMON1_2 Read Bit1 IMON1_1 Read Bit0 IMON1_0 Read		0x4008_E0A4 KEYINTMON1 Bit2 IMON1_2 1Write Bit1 IMON1_1 1Write Bit0 IMON1_0 1Write	0x4008_E09C KEYINTEN1 Bit2 EN1_2 0Write Bit1 EN1_1 0Write Bit0 EN1_0 0Write
64	48	28	AudioBuffer0	Overlevel, Underlevel, Overflow or Underflow of Buffer A~N was generated. Following register can select that Overlevel, Underlevel, Overflow or Underflow of Buffer A~N enter which AudioBuffer from 0 to 3. 0x4006_0010 ABUFIRQSEL0 BnISEL0-1 0x4006_0014 ABUFIRQSEL1 BnISEL0-1 0x4006_0018 ABUFIRQSEL2 BnISEL0-1 0x4006_001C ABUFIRQSEL3 BnISEL0-1		0x4006_0034 ABUFSTS1 Bit29-16 BnOLVL Read Bit13-0 BnULVL Read 0x4006_0038 ABUFSTS2 Bit29-16 BnOVF Read Bit13-0 BnUDF Read	Overlevel interrupt : It becomes clear when the number of data in the corresponding buffer was less than the setting value of Overlevel. (Setting : 0x4006_0180-0x4006_01B4 LVL15-0) Underlevel interrupt : It becomes clear when the number of data in the corresponding buffer was more than the setting value of Underlevel. (Setting : 0x4006_0140-0x4006_0174 LVL15-0) Overflow interrupt : 0x4006_0000 ABUFCLR Bit13-0 BUFnCLR 1Write Underflow interrupt : 0x4006_0000 ABUFCLR Bit13-0 BUFnCLR 1Write	0x4006_0008 ABUFIRQEN0 Bit29-16 BnOLIRQEN 0Write Bit13-0 BnULIRQEN 0Write 0x4006_000C ABUFIRQEN1 Bit29-16 BnOFIRQEN 0Write Bit13-0 BnUFIRQEN 0Write
65	49	29	AudioBuffer1					
66	50	30	AudioBuffer2					
67	51	31	AudioBuffer3					

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
68	52	32	AudioStatus0	Refer to "Audio Interrupt Source List Table" Core Exception No. (Core NVIC Bit No., LPDSP32 control No.) can be changed by 0x4006_4010 - 0x4006_4014 IRQSET0 - 1.				
69	53	33	AudioStatus1					
70	54	34	AudioStatus2					
71	55	35	AudioStatus3					
72	56	36	AudioTimer0	Counter underflow	0x4006_1710 ATMIRQ Bit0 ATMINT Read		0x4006_1710 ATMIRQ Bit0 ATMINT 1Write	
73	57	37	AudioTimer1	Counter underflow	0x4006_1810 ATMIRQ Bit0 ATMINT Read		0x4006_1810 ATMIRQ Bit0 ATMINT 1Write	
74	58		USB Device	IN transfer buffer Ready OUT transfer buffer Ready PING Token receive NAK response STALL response Buffer Empty DMA Ch1,2 transfer completed DMA Ch1,2 transfer incompleted DMA Ch1,2 transfer interrupted EPCMD completed PHY_Erratic_Error OTG factor Status stage Failure/Success Bus reset Start/End SETUP token receive SOF receive Suspend status Start/End Host factor	0x4001_0314 OTGSTSC Bit10 vbus_vld_c Read Bit6 id_c Read Bit0 otg_tmrout_c Read	0x4001_000C INTS Bit31-16 dev_ep_int Read Bit9 dma2_int Read Bit8 dma1_int Read Bit4 cmd_int Read Bit3 phy_err_int Read Bit2 otg_int Read Bit1 dev_int Read Bit0 host_int Read 0x4001_8000-0x4001_803C EPCTRL0-15 (MODE.dev_en=1) Bit30 nack_int Read Bit29 stalled_int Read Bit28 ping_int Read Bit27 EP0:readyo_int Read / EP1-15:empty_int Read Bit26 EP0:readyi_int Read / EP1-15:ready_int Read 0x4001_0204 DEVS Bit31 status_ng_int Read Bit30 status_ok_int Read Bit29 usbrstb_int Read Bit28 usbrste_int Read Bit27 setup_int Read Bit26 sof_int Read Bit25 supendb_int Read Bit24 supende_int Read	0x4001_000C INTS Bit9 dma2_int 0Write Bit8 dma1_int 0Write Bit4 cmd_int 0Write Bit3 phy_err_int 0Write 0x4001_0040-0x4001_007C EPCMD0-15 (MODE.dev_en=1) Bit22 nack_int_clr 1Write Bit21 stalled_int_clr 1Write Bit20 ping_int_clr 1Write Bit19 EP0:readyo_int_clr 1Write / EP1-15:empty_int_clr 1Write Bit18 EP0:readyi_int_clr 1Write / EP1-15:ready_int_clr 1Write 0x4001_0314 OTGSTSC Bit10 vbus_vld_c 0Write Bit6 id_c 0Write Bit0 otg_tmrout_c 0Write 0x4001_0204 DEVS Bit31 status_ng_int 0Write Bit30 status_ok_int 0Write Bit29 usbrstb_int 0Write Bit28 usbrste_int 0Write Bit27 setup_int 0Write Bit26 sof_int 0Write Bit25 supendb_int 0Write Bit24 supende_int 0Write	0x4001_0008 INTEN Bit31-16 dev_ep_inten 0Write Bit9 dma2_inten 0Write Bit8 dma1_inten 0Write Bit4 cmd_inten 0Write Bit3 phy_err_inten 0Write Bit2 otg_inten 0Write Bit1 dev_inten 0Write Bit0 host_inten 0Write 0x4001_0040-0x4001_007C EPCMD0-15 (MODE.dev_en=1) Bit16 nack_int_en 0Write Bit15 stalled_int_en 0Write Bit14 ping_int_en 0Write Bit13 EP0:readyo_int_en 0Write / EP1-15:empty_int_en 0Write Bit12 EP0:readyi_int_en 0Write / EP1-15:ready_int_en 0Write 0x4001_0318 OTGSTSFALL Bit10 vbus_vld_fen 0Write Bit6 id_fen 0Write 0x4001_031C OTGSTSRISE Bit10 vbus_vld_ren 0Write Bit6 id_ren 0Write Bit0 otg_tmrout_ren 0Write 0x4001_0200 DEVC Bit31 status_ng_inten 0Write Bit30 status_ok_inten 0Write Bit29 usbrstb_inten 0Write Bit28 usbrste_inten 0Write Bit27 setup_inten 0Write Bit26 sof_inten 0Write Bit25 supendb_inten 0Write Bit24 supende_inten 0Write
75	59	38	EXTINT0	Generate interrupt to external interrupt terminal (EXTINT00~0F)	0x4000_3400 EXTINT0 Read	0x4000_3418 EXTINT0S Read	0x4000_3490 EXTINT0CLR 1Write	0x4000_3430 EXTINT0M 1Write

Core Exception No.	Core NVIC Bit No.	LPDSP32 Control No.	Function (Name)	Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear	Factor Mask
76	60	39	EXTINT1	Generate interrupt to external interrupt terminal (EXTINT10~1F)	0x4000_3404 EXTINT1 Read	0x4000_341C EXTINT1S Read	0x4000_3494 EXTINT1CLR 1Write	0x4000_3434 EXTINT1M 1Write
77	61	40	EXTINT2	Generate interrupt to external interrupt terminal (EXTINT20~2F)	0x4000_3408 EXTINT2 Read	0x4000_3420 EXTINT2S Read	0x4000_3498 EXTINT2CLR 1Write	0x4000_3438 EXTINT2M 1Write
78	62	41	EXTINT3	Generate interrupt to external interrupt terminal (EXTINT30~3F)	0x4000_340C EXTINT3 Read	0x4000_3424 EXTINT3S Read	0x4000_349C EXTINT3CLR 1Write	0x4000_343C EXTINT3M 1Write
79	63	42	EXTINT4	Generate interrupt to external interrupt terminal (EXTINT40~4F)	0x4000_3410 EXTINT4 Read	0x4000_3428 EXTINT4S Read	0x4000_34A0 EXTINT4CLR 1Write	0x4000_3440 EXTINT4M 1Write
80	64	43	EXTINT5	Generate interrupt to external interrupt terminal (EXTINT50~59)	0x4000_3414 EXTINT5 Read	0x4000_342C EXTINT5S Read	0x4000_34A4 EXTINT5CLR 1Write	0x4000_3444 EXTINT5M 1Write
		63	LPDSP32 Bus Error	Generated by that DSP access to seg[n] in PM at the state of 0x4000_8014 MEMEN2P Bit[n]=0. (n:0~9) Generated by that DSP access to seg[n] in DMA at the state of 0x4000_8018 MEMEN2A Bit[n]=0. (n:0~8) Generated by that DSP access to seg[n] in DMB at the state of 0x4000_801C MEMEN2B Bit[n]=0. (n:0~8)	0x4000_387C DSPDMASTS[n] Read[n] 0x4000_3880 DSPDMBSTS[n] Read[n] 0x4000_3884 DSPPMSTS[n] Read[n] n:0~9		0x4000_3870 DSPDMACLR[n] 1Write 0x4000_3874 DSPDMBCLR[n] 1Write 0x4000_3878 DSPPMCLR[n]1Write n:0~9	

Table B : Audio Interrupt Source List Table

Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear 1	Factor Clear 2	Factor Mask 1	Factor Mask 2
Meter of PCM1 P/S zero level detect	0x4006_4008 AUDIRQ Bit31 MTRPS1IRQ1 Read	0x4006_44DC MTRINTSTS Bit9 MTRZINTRSTS Read Bit8 MTRZINTLSTS Read	0x4006_4008 AUDIRQ Bit31 MTRPS1IRQ1 1Write	0x4006_44C0 MTRCLR Bit5 METERRCCLR 1Write Bit4 METERLCCLR 1Write	0x4006_4008 AUDIRQ Bit31 MTRPS1MASK1 1Write	0x4006_44D0 MTRINTCTL Bit10 MTRZINTMASK 1Write
Meter of PCM1 P/S level detect	0x4006_4008 AUDIRQ Bit30 MTRPS1IRQ0 Read	0x4006_44DC MTRINTSTS Bit1 MTRINTRSTS Read Bit0 MTRINTLSTS Read	0x4006_4008 AUDIRQ Bit30 MTRPS1IRQ0 1Write	0x4006_44C0 MTRCLR Bit1 METERRCLR 1Write Bit0 METERLCLR 1Write	0x4006_4008 AUDIRQ Bit30 MTRPS1MASK0 1Write	0x4006_44D0 MTRINTCTL Bit2 MTRINTMASK 1Write
Volume of PCM1 P/S error	0x4006_4008 AUDIRQ Bit29 VOLPS1IRQ1 Read	0x4006_439C ERRSTS Bit1 ERRSTSR Read Bit0 ERRSTSL Read	0x4006_4008 AUDIRQ Bit29 VOLPS1IRQ1 1Write	0x4006_4398 ERRSTSCLR Bit1 ERRCLRR 1Write Bit0 ERRCLRL 1Write	0x4006_4008 AUDIRQ Bit29 VOLPS1MASK1 1Write	
Volume of PCM1 P/S gain change completed	0x4006_4008 AUDIRQ Bit28 VOLPS1IRQ0 Read		0x4006_4008 AUDIRQ Bit28 VOLPS1IRQ0 1Write		0x4006_4008 AUDIRQ Bit28 VOLPS1MASK0 1Write	
Meter of PCM1 S/P zero level detect	0x4006_4008 AUDIRQ Bit27 MTRSP1IRQ1 Read	0x4006_44AC MTRINTSTS Bit9 MTRZINTRSTS Read Bit8 MTRZINTLSTS Read	0x4006_4008 AUDIRQ Bit27 MTRSP1IRQ1 1Write	0x4006_4490 MTRCLR Bit5 METERRCCLR 1Write Bit4 METERLCCLR 1Write	0x4006_4008 AUDIRQ Bit27 MTRSP1MASK1 1Write	0x4006_44A0 MTRINTCTL Bit10 MTRZINTMASK 1Write
Meter of PCM1 S/P level detect	0x4006_4008 AUDIRQ Bit26 MTRSP1IRQ0 Read	0x4006_44AC MTRINTSTS Bit1 MTRINTRSTS Read Bit0 MTRINTLSTS Read	0x4006_4008 AUDIRQ Bit26 MTRSP1IRQ0 1Write	0x4006_4490 MTRCLR Bit1 METERRCLR 1Write Bit0 METERLCLR 1Write	0x4006_4008 AUDIRQ Bit26 MTRSP1MASK0 1Write	0x4006_44A0 MTRINTCTL Bit2 MTRINTMASK 1Write
Volume of PCM1 S/P error	0x4006_4008 AUDIRQ Bit25 VOLSP1IRQ1 Read	0x4006_437C ERRSTS Bit1 ERRSTSR Read Bit0 ERRSTSL Read	0x4006_4008 AUDIRQ Bit25 VOLSP1IRQ1 1Write	0x4006_4378 ERRSTSCLR Bit1 ERRCLRR 1Write Bit0 ERRCLRL 1Write	0x4006_4008 AUDIRQ Bit25 VOLSP1MASK1 1Write	
Volume of PCM1 S/P gain change completed	0x4006_4008 AUDIRQ Bit24 VOLSP1IRQ0 Read		0x4006_4008 AUDIRQ Bit24 VOLSP1IRQ0 1Write		0x4006_4008 AUDIRQ Bit24 VOLSP1MASK0 1Write	
BEEP output completed	0x4006_4008 AUDIRQ Bit23 BEEPIRQ Read		0x4006_4008 AUDIRQ Bit23 BEEPIRQ 1Write		0x4006_4008 AUDIRQ Bit23 BEEPMASK 1Write	
EQ3 error	0x4006_4008 AUDIRQ Bit22 EQ3IRQ Read	0x4006_1118 EQ_STATE Bit16 ERR_STATE Read	0x4006_4008 AUDIRQ Bit22 EQ3IRQ 1Write	0x4006_1118 EQ_STATE Bit17 ERR_CLR 1Write	0x4006_4008 AUDIRQ Bit22 EQ3MASK 1Write	
Mute of PCM0 P/S error	0x4006_4008 AUDIRQ Bit21 MUTEPS0IRQ1 Read	0x4006_4240 ERRSTS Bit1 ERRSTSR Read Bit0 ERRSTSL Read	0x4006_4008 AUDIRQ Bit21 MUTEPS0IRQ1 1Write	0x4006_423C ERRSTSCLR Bit1 ERRCLRR 1Write Bit0 ERRCLRL 1Write	0x4006_4008 AUDIRQ Bit21 MUTEPS0MASK1 1Write	
Mute of PCM0 P/S transition completed	0x4006_4008 AUDIRQ Bit20 MUTEPS0IRQ0 Read		0x4006_4008 AUDIRQ Bit20 MUTEPS0IRQ0 1Write		0x4006_4008 AUDIRQ Bit20 MUTEPS0MASK0 1Write	
Meter of PCM0 P/S zero level detect	0x4006_4008 AUDIRQ Bit19 MTRPS0IRQ1 Read	0x4006_447C MTRINTSTS Bit9 MTRZINTRSTS Read Bit8 MTRZINTLSTS Read	0x4006_4008 AUDIRQ Bit19 MTRPS0IRQ1 1Write	0x4006_4460 MTRCLR Bit5 METERRCCLR 1Write Bit4 METERLCCLR 1Write	0x4006_4008 AUDIRQ Bit19 MTRPS0MASK1 1Write	0x4006_4470 MTRINTCTL Bit10 MTRZINTMASK 1Write
Meter of PCM0 P/S level detect	0x4006_4008 AUDIRQ Bit18 MTRPS0IRQ0 Read	0x4006_447C MTRINTSTS Bit1 MTRINTRSTS Read Bit0 MTRINTLSTS Read	0x4006_4008 AUDIRQ Bit18 MTRPS0IRQ0 1Write	0x4006_4460 MTRCLR Bit1 METERRCLR 1Write Bit0 METERLCLR 1Write	0x4006_4008 AUDIRQ Bit18 MTRPS0MASK0 1Write	0x4006_4470 MTRINTCTL Bit2 MTRINTMASK 1Write
Volume of PCM0 P/S error	0x4006_4008 AUDIRQ Bit17 VOLPS0IRQ1 Read	0x4006_435C ERRSTS Bit1 ERRSTSR Read Bit0 ERRSTSL Read	0x4006_4008 AUDIRQ Bit17 VOLPS0IRQ1 1Write	0x4006_4358 ERRSTSCLR Bit1 ERRCLRR 1Write Bit0 ERRCLRL 1Write	0x4006_4008 AUDIRQ Bit17 VOLPS0MASK1 1Write	
Volume of PCM0 P/S gain change completed	0x4006_4008 AUDIRQ Bit16 VOLPS0IRQ0 Read		0x4006_4008 AUDIRQ Bit16 VOLPS0IRQ0 1Write		0x4006_4008 AUDIRQ Bit16 VOLPS0MASK0 1Write	
Meter of PCM0 S/P zero level detect	0x4006_4008 AUDIRQ Bit15 MTRSP0IRQ1 Read	0x4006_444C MTRINTSTS Bit9 MTRZINTRSTS Read Bit8 MTRZINTLSTS Read	0x4006_4008 AUDIRQ Bit15 MTRSP0IRQ1 1Write	0x4006_4430 MTRCLR Bit5 METERRCCLR 1Write Bit4 METERLCCLR 1Write	0x4006_4008 AUDIRQ Bit15 MTRSP0MASK1 1Write	0x4006_4440 MTRINTCTL Bit10 MTRZINTMASK 1Write
Meter of PCM0 S/P level detect	0x4006_4008 AUDIRQ Bit14 MTRSP0IRQ0 Read	0x4006_444C MTRINTSTS Bit1 MTRINTRSTS Read Bit0 MTRINTLSTS Read	0x4006_4008 AUDIRQ Bit14 MTRSP0IRQ0 1Write	0x4006_4430 MTRCLR Bit1 METERRCLR 1Write Bit0 METERLCLR 1Write	0x4006_4008 AUDIRQ Bit14 MTRSP0MASK0 1Write	0x4006_4440 MTRINTCTL Bit2 MTRINTMASK 1Write
Volume of PCM0 S/P error	0x4006_4008 AUDIRQ Bit13 VOLSP0IRQ1 Read	0x4006_433C ERRSTS Bit1 ERRSTSR Read Bit0 ERRSTSL Read	0x4006_4008 AUDIRQ Bit13 VOLSP0IRQ1 1Write	0x4006_4338 ERRSTSCLR Bit1 ERRCLRR 1Write Bit0 ERRCLRL 1Write	0x4006_4008 AUDIRQ Bit13 VOLSP0MASK1 1Write	
Volume of PCM0 S/P gain change completed	0x4006_4008 AUDIRQ Bit12 VOLSP0IRQ0 Read		0x4006_4008 AUDIRQ Bit12 VOLSP0IRQ0 1Write		0x4006_4008 AUDIRQ Bit12 VOLSP0MASK0 1Write	

Generating factor	Factor Confirmation	Factor Source Confirmation	Factor Clear 1	Factor Clear 2	Factor Mask 1	Factor Mask 2
Meter of Decoder zero level detect	0x4006_4008 AUDIRQ Bit11 MTRDECIRQ1 Read	0x4006_441C MTRINTSTS Bit9 MTRZINTRSTS Read Bit8 MTRZINTLSTS Read	0x4006_4008 AUDIRQ Bit11 MTRDECIRQ1 1Write	0x4006_4400 MTRCLR Bit5 METERRCCLR 1Write Bit4 METERLCCLR 1Write	0x4006_4008 AUDIRQ Bit11 MTRDECMASK1 1Write	0x4006_4410 MTRINTCTL Bit10 MTRZINTMASK 1Write
Meter of Decoder level detect	0x4006_4008 AUDIRQ Bit10 MTRDECIRQ0 Read	0x4006_441C MTRINTSTS Bit1 MTRINTRSTS Read Bit0 MTRINTLSTS Read	0x4006_4008 AUDIRQ Bit10 MTRDECIRQ0 1Write	0x4006_4400 MTRCLR Bit1 METERRCLR 1Write Bit0 METERLCLR 1Write	0x4006_4008 AUDIRQ Bit10 MTRDECMASK0 1Write	0x4006_4410 MTRINTCTL Bit2 MTRINTMASK 1Write
Volume of Decoder error	0x4006_4008 AUDIRQ Bit9 VOLDECIRQ1 Read	0x4006_431C ERRSTS Bit1 ERRSTSR Read Bit0 ERRSTSL Read	0x4006_4008 AUDIRQ Bit9 VOLDECIRQ1 1Write	0x4006_4318 ERRSTSCLR Bit1 ERRCLRR 1Write Bit0 ERRCLRL 1Write	0x4006_4008 AUDIRQ Bit9 VOLDECMASK1 1Write	
Volume of Decoder gain change completed	0x4006_4008 AUDIRQ Bit8 VOLDECIRQ0 Read		0x4006_4008 AUDIRQ Bit8 VOLDECIRQ0 1Write		0x4006_4008 AUDIRQ Bit8 VOLDECMASK0 1Write	
Asynchronous SRC remaining data amount monitor	0x4006_4008 AUDIRQ Bit7 ASRCIRQ Read		0x4006_4008 AUDIRQ Bit7 ASRCIRQ 1Write		0x4006_4008 AUDIRQ Bit7 ASRCMASK 1Write	
PCM1slave clock error	0x4006_4008 AUDIRQ Bit6 PCKGENIRQ1 Read	0x4006_160C SLVERR_STS Bit6 PCK1PRDERR Read Bit5 PCK1SLTERR Read	0x4006_4008 AUDIRQ Bit6 PCKGENIRQ1 1Write	0x4006_160C SLVERR_STS Bit4 PCK1ERRCLR 1Write	0x4006_4008 AUDIRQ Bit6 PCKGENMASK1 1Write	
PCM0slave clock error	0x4006_4010 AUDIRQ Bit5 PCKGENIRQ0 Read	0x4006_160C SLVERR_STS Bit4 PCK0PRDERR Read Bit3 PCK0SLTERR Read	0x4006_4008 AUDIRQ Bit5 PCKGENIRQ0 1Write	0x4006_160C SLVERR_STS Bit0 PCK0ERRCLR 1Write	0x4006_4008 AUDIRQ Bit5 PCKGENMASK0 1Write	
MP3 Encoder encode completed	0x4006_4012 AUDIRQ Bit4 MP3ENCIRQ Read	0x4006_301C MP3ENC_ENDSTAT Bit0 BUFEND Read	0x4006_4008 AUDIRQ Bit4 MP3ENCIRQ 1Write		0x4006_4008 AUDIRQ Bit4 MP3ENCMASK 1Write	
MP3 Decoder status ready	0x4006_4013 AUDIRQ Bit2 MP3DECIRQ2 Read	0x4006_2008 MP3ERR Bit2 MP3ERR2 Read	0x4006_4008 AUDIRQ Bit2 MP3DECIRQ2 1Write	0x4006_2004 MP3CTL Bit2 STSCLR 1Write	0x4006_4008 AUDIRQ Bit2 MP3DECMASK2 1Write	
MP3 Decoder CRC error	0x4006_4014 AUDIRQ Bit1 MP3DECIRQ1 Read	0x4006_2008 MP3ERR Bit1 MP3ERR1 Read	0x4006_4008 AUDIRQ Bit1 MP3DECIRQ1 1Write		0x4006_4008 AUDIRQ Bit1 MP3DECMASK1 1Write	
MP3 Decoder SYNC error	0x4006_4015 AUDIRQ Bit0 MP3DECIRQ0 Read	0x4006_2008 MP3ERR Bit0 MP3ERR0 Read	0x4006_4008 AUDIRQ Bit0 MP3DECIRQ0 1Write		0x4006_4008 AUDIRQ Bit0 MP3DECMASK0 1Write	

ARM, the ARM logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere.

ON Semiconductor and the ON Semiconductor logo are trademarks of Semiconductor Components Industries, LLC dba ON Semiconductor or its subsidiaries in the United States and/or other countries. ON Semiconductor owns the rights to a number of patents, trademarks, copyrights, trade secrets, and other intellectual property. A listing of ON Semiconductor's product/patent coverage may be accessed at www.onsemi.com/site/pdf/Patent-Marking.pdf. ON Semiconductor reserves the right to make changes without further notice to any products herein. ON Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does ON Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation special, consequential or incidental damages. Buyer is responsible for its products and applications using ON Semiconductor products, including compliance with all laws, regulations and safety requirements or standards, regardless of any support or applications information provided by ON Semiconductor. "Typical" parameters which may be provided in ON Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. ON Semiconductor does not convey any license under its patent rights nor the rights of others. ON Semiconductor products are not designed, intended, or authorized for use as a critical component in life support systems or any FDA Class 3 medical devices or medical devices with a same or similar classification in a foreign jurisdiction or any devices intended for implantation in the human body. Should Buyer purchase or use ON Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold ON Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that ON Semiconductor was negligent regarding the design or manufacture of the part. ON Semiconductor is an Equal Opportunity/Affirmative Action Employer. This literature is subject to all applicable copyright laws and is not for resale in any manner.