ON Semiconductor

Is Now

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

G.Host JTAG Interface for Graphics Host Reference Design

Fairchild Semiconductor
Application Note 1003
August 1995

INTRODUCTION

The Graphics Host Reference Design kit is produced by the staff at Hamilton Hallmark's Technical Support Center in partnership with suppliers. A *reference design* is a working design, with all of the necessary elements in place to serve as an example of how a project might be approached. Included are schematics, application notes, program code (where appropriate), and data sheets. Request a kit by calling 1-800-605-3296 xH147 or through the World Wide Web at http://www.tsc.hh.avnet.com.

The G.Host design (the G is for Graphics) was made possible with the active support of the following manufacturers:

- Advanced Micro Devices with the 29K™ family of RISC microcontrollers, the MACH®465 complex programmable logic device (CPLD), and the AMD® 5V-only Flash memory. These components form a powerful and feature-rich design core that is easy to use. Virtually no glue is needed to interface to a wide range of peripherals.
- Fairchild Semiconductor brings several boundary-scan (JTAG) hardware elements to this design. The high-integration components used in the reference design—and the correspondingly high pin count of fine-pitch devices—need reliable methods of programming, test, and verification. The MACH465 CPLD, in particular, is best programmed in the circuit after assembly. Fairchild's Multidrop and Hierarchical Addressable JTAG Port, SCANPSC110F bridge; the Embedded Boundary Scan Controller, SCANPSC100F; and the SCAN18245T JTAG-compatible bus transceivers make it possible to interface with the JTAG capability built into the MACH465 CPLD and the AM29200™ family microcontrollers with ease.
- Corelis contributes the software for the boundary-scan process to generate boundary-scan test files. Their boundary-scan emulator allows quick access to the capability of Fairchild's boundary-scan devices, the AM29200 family of boundary-scan-compliant microcontrollers, and the AMD MACH465 CPLD. These tools have been a real boon to the test and verification of this design.
- Wind River Systems supplies the operating system for the design. The 29K family RISC microcontrollers are powerful and easy to program in their own right, with an architecture that lends itself to C or C++-language programming, but Wind River's operating system allows the programmer to concentrate on the features of the individual product. The initializations, interrupt vectors, and housekeeping details are quickly dispatched with the OS. They no longer need to be software projects in their own right.

The AM29200 family of RISC microcontrollers is especially suited for dealing with multiple complex tasks like performing real-time handshaking for what is essentially a wireless network, decompressing the data and preprocessing it for the printer. That much work in real time demands a powerful and fast instruction set and a high degree of predictability in execution. Differences in execution time (inherent in CISC architectures) can create unpredictable latencies. When operating a fast real-time data operation, such as our RF link, latencies must be minimized. This is one of the benefits of a RISC architecture, such as that used in the AM29200 microcontroller family. Each instruction in the set is completed in one instruction cycle. (Exceptions are instructions that respond to, or return from, interrupts, external memory accesses, and some math instructions.) Because of this execution stability, it is possible to reliably determine the time it takes to respond to real-time events.

The MACH465 CPLD is an important complement to the AM29200 family microcontrollers in this design. Essentially a large programmable block, the MACH465 CPLD can be set to do virtually anything to complement the power of the microcontroller.

The 5V-only Flash memory is integral to the core design idea as well. It can be reprogrammed using the AM29200 family microcontroller by simply writing to it if that memory area is set as writable in the Flash memory and in the microcontroller. Boot and operating system code can be protected either by programming the boot-block sectors of the Flash memory as non-writeable, or by configuring the microcontroller to treat the Flash memory area as read only. In the G.Host design, the boot areas of the Flash memory are selected by the ROM Chip Select 0 ($\overline{\text{ROMCS0}}$) pin; the normal operating area is selected by the ROM Chip Select 1 ($\overline{\text{ROMCS1}}$) pin. Either method will protect critical code areas.

Having the ability to update the operational portion of the program allows for maintenance and updates from a remote location. Simply place the program update algorithms in a write-protected (read-only) code area, vector to it, and begin. One scenario would be to place the new program in DRAM as data, vector to the update routine, and write the Flash memory from DRAM.

Information will arrive at the G.Host through the bidirectional serial port. This port will be connected to an RF transceiver for bidirectional communication. The microcontroller will control the handshaking protocol, the decompression of the data received, and the compression of any data sent back to the remote PC. It will preprocess the data for the HP-LaserJet-II-compatible printer so the printer does not need to do any further conversion. Both graphics processing acceleration—making the print job less time consuming—and multiple PCs sharing the resources of a single printer are realized.

The implementation of JTAG interface testing was chosen for the G.Host design for several practical reasons. The design warranted test accessibility due to its short development and debug time allotment. Once the design progressed into a standalone working unit, it was desirable to have it exercise a built-in self test to diagnose and report any physical errors that might exist during power-up. For these reasons, the G.Host design utilized as many JTAG compliant components as possible.

To keep in context with the actual design and its implementation of boundary scan testing, it is assumed that the reader is familiar with general concepts of the IEEE 1149.1 standard. Additional information on JTAG and Boundary Scan may be found in the Fairchild Semiconductor SCAN Databook.

There were several factors that contributed to the actual implementation of JTAG on this design. First, it was necessary to decide how to interface the signals to the board for testing. Second, it was important to allow for self or embedded testing of the board once the design was functional. Third, a JTAG port would be required for programming the AMD Mach device on the board. And finally, a JTAG port was necessary to allow for emulation of the AM29200 device.

Although it is possible to simply connect an external boundary scan tester to the board directly via a cable carrying the 5 JTAG lines and daisy-chaining the TDO to TDI lines from one device to another, this was not the answer for the G.Host design. It was important to find a way of allowing the microprocessor to test the board utilizing boundary scan. This would entail two things:

1. Finding a device that could accept a microprocessor interface on one side and a JTAG interface on the other.
2. Finding a device that could interface a single JTAG port into several separate chains in order to isolate the microprocessor and as few boundary scan components from the remaining JTAG devices possible.

Fairchild Semiconductor has two such components: the Embedded Boundary Scan Controller, SCANPSC100F and the Hierarchical and Multidrop Addressable JTAG Port, SCANPSC110F. The SCANPSC100F is an interface chip that connects an asynchronous parallel bus processor to a JTAG chain. Data can be written to the device via a µP/µC which is then serialized by the SCANPSC100F for boundary scan communications. Results on the boundary scan chain are sent back to the SCANPSC100F and are then readable on the µP/µC side. The SCANPSC110F is a SCAN bridge chip that supports up to 3 local scan paths which can be accessed individually from a single host JTAG port. Both of these devices were well suited for accomplishing our design goals.

The next task was determining if a suite of tools existed that would allow us to utilize these components successfully in the design. Corelis, a California based company specializing in boundary scan products, was the right choice. They have been working with both Fairchild Semiconductor and AMD for several years and have developed a powerful set of tools that will support testing a board containing the SCANPSC110F, JTAG programming a MACH device on board, and performing JTAG AM29000 emulation. The only interfacing required is several 10-pin headers placed on the G.Host board to connect a cable containing the 5 JTAG lines and 5 grounds from their hardware.

As stated earlier, the SCANPSC110 allows for multiple local scan paths. Choosing which devices to be placed on which particular chain was extremely important, based on the embedded self-testing goal. In order to boot-up the AM29000 device and execute test code, the only additional devices required on the G.Host design were a SCAN18245T transceiver, the flash memory, and a few minor discrete components. Of these devices, only the SCAN18245T includes boundary scan capability. Therefore, the AM29000 and SCAN18245T were placed on one chain while the remaining JTAG components were placed on the other. The final JTAG chain was to be reserved for the serial/RFD communications port, in case it included any boundary scan components.

As just stated, the first JTAG chain includes a AM29000 microcontroller and a Fairchild Semiconductor SCAN18245T. As seen in *Figure 1*, the boundary scan chain signals (TDO to TDI) are connected from the SCANPSC110, to the AM29000, then to SCAN18245T and back to the SCANPSC110. The TMS and TCLK lines are shown as being shared between the devices. This would be the standard configuration if JTAG emulation was not required. But, since the AM29000 needs to be isolated for emulation, 1 kΩ resistors were placed in the TMS and the TDI paths. Also, a jumper header is added to the TCLK path allowing either the SCANPSC110 or the alternate external port to drive the AM29000's TCLK line. (See *Figure 1*.)

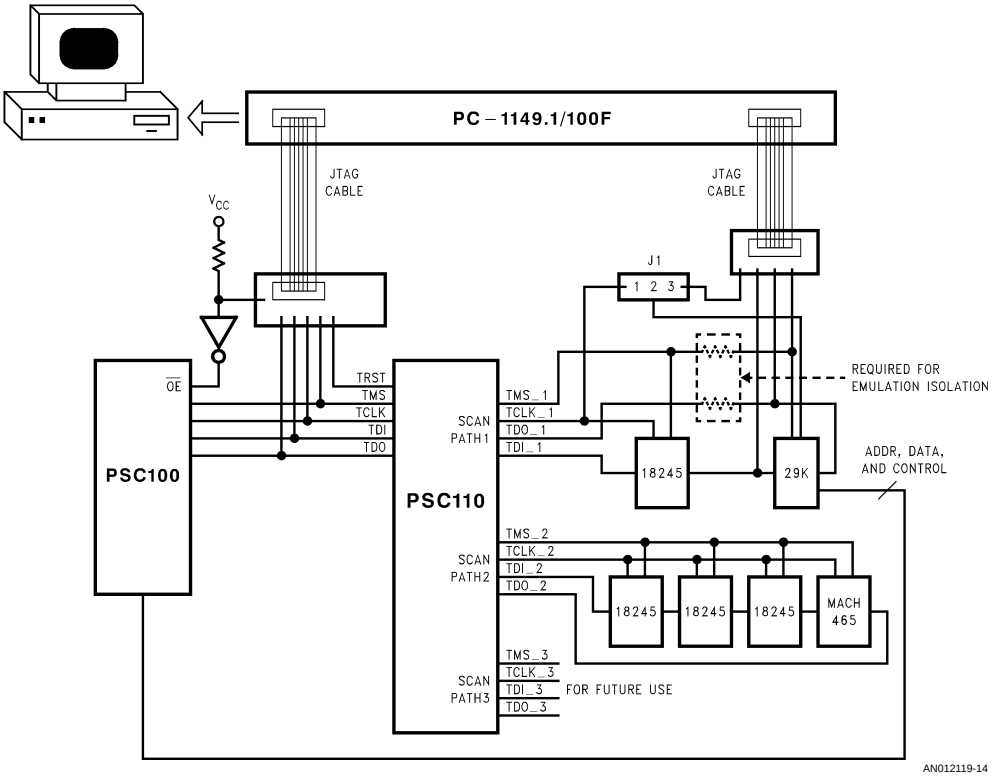

FIGURE 1. G.Host JTAG Connectivity

As a side note, when external testing is performed on the board, the SCANPSC100's output enable must be disabled in order to avoid signal conflict on the boundary scan lines. This is accomplished by placing an inverter between one of the 5 header pins on the JTAG connector (normally open) and the active low output enable ($\overline{OE}$) signal of the SCANPSC100F. The input to the inverter contains a pull-up resistor, such that the $\overline{OE}$ signal is normally low. When the 10-pin JTAG cable is attached to the board, the input to the inverter is driven low from the ground connection on the cable and causing a high on the $\overline{OE}$ of the SCANPSC100F; thereby disabling the device.

The second boundary scan path contains four JTAG components: an AMD MACH465 and three SCAN18245T. The actual connections are shown in *Figure 1*. As in the previous path, isolation has been setup for the MACH465 in order to provide an alternate path for programming the device.

Actual testing of the G.Host design was extremely simple due to the suite of tools by Corelis. All that was required were two ASCII files: a connection (CON) file and boundary scan net (BNET) file. (Request electronic files from Hamilton Hallmark Technical Resource Center, 1-800-605-3294 or on the Internet at http://www.tsc.hh.avnet.com.) The CON file is a very simple file that is required when using a SCANPSC110F device on board. It specifies the names of the local scan paths that exist on the board which are referenced from the NET file. The NET file contains the backbone of the design. First an IC number must be associated with each JTAG device. Then each device is placed in order of location in each boundary scan chain (last placed first). Following this, is a list of the actual names and pin numbers of those connections that exist between these devices.

Also required, are the Boundary Scan Description files (BSD) of the JTAG devices used. These files are readily available from the manufacturer of the components themselves. (Fairchild BSD files for all SCAN parts may be obtained by calling 1-800-272-9959. Request Lit.# 580220-001; or on the World Wide Web at http://www.fairchildsemi.com.) It is necessary to place these files into a specific directory that can be referenced by the software.

The Corelis tool used for testing was the PC-1149.1/100F/FULL. The hardware is a JTAG test board based on the Fairchild Semiconductor SCANPSC100F which contains 2 boundary scan interface ports. The software is a Windows based tool which includes Boundary Scan Automatic Test Pattern Generation (BTPG), and Enhanced Diagnostics. This package made the test development cycle extremely short and simplified due to its fully automated and admirably reported test results. The Corelis package is a powerful tool for JTAG testing.

CONTROLLERS

The AM29205 and the AM29200 microcontrollers were chosen as representative of the AMD 29K family of microprocessors and microcontrollers (see *Figure 4* and *Figure 5*). If the AM29205 microcontroller is adequate for a particular product or model, the footprint and interface for it is available. If the product or model needs more power, the bigger and faster AM29200 microcontroller may be used with no change in the rest of the design. Both devices have similar features and the same instruction set, with the AM29200 microcontroller being a superset of the AM29205 microcontroller.

The AM29205 microcontroller implements an external 16-bit instruction/data (ID) bus (still a full 32 bits internally). This presents some interesting complications when trying to use it interchangeably with the external 32-bit ID bus of the AM29200 microcontroller. Both microcontrollers have:

- A Harvard-style architecture, where program and data memory are separate
- DRAM interface drivers integrated in the microcontroller (including built-in CAS-before-RAS refresh)
- ROM drive (including support for writing to ROM space)
- Programmable peripheral-port pins.

The AM29200 microcontroller family is capable of DMA activity, and the ability to use it is built in the G.Host design as well. We are not using it in this iteration, however, because of the uncertainties involved in a DMA device possibly not releasing the bus back to the microcontroller in time to avoid data loss on the RF link. Hamilton Hallmark included it in the design to enhance the MACH465 CPLD FIFO when that is implemented. A designer wishing to use this feature could do so now, if no critical real-time operation could be jeopardized as our RF link could.

At power-on reset, both devices can be hardware-initialized to operate in a 16-bit format, and we have done that in this design. (Through the BOOTW signal, the AM29200 microcontroller may be configured to boot with 8-bit or 16-bit wide instruction fetching.) In this way, the boot code is the same for both microcontrollers, and the program can determine system configuration. The initialization code will check for microcontroller type (there is an internal processor revision level [PRL] field in the configuration register identifying each member of the family) re-initialize the appropriate registers to the proper values, then vector to the running portion of code as either a 16-bit or 32-bit external machine.

The 29K family microcontrollers are memory mapped with all features assigned to a memory address space of their own. This makes it easy to keep track of where peripherals and memory reside. Accessing a particular area of code memory, for example, is as simple as specifying the appropriate memory address range. The AM29200 and AM29205 microcontrollers will even take care of breaking the accesses into the appropriate number of wait states and bus widths as you have specified through internal control and configuration registers.

Since we have this powerful device available, we have used the MACH465 CPLD to implement a few other glue functions, such as translating levels for the parallel port, and providing the external latch functions needed by the microcontroller's parallel-port interface. The parallel port in the AM29200 family microcontrollers performs a mini bus-cycle within the larger parallel-port cycle. During a write, the microcontroller provides the appropriate control signals to place data in the parallel latch for the external system. It then completes the appropriate handshaking necessary to complete the transfer. During a read, the microcontroller uses the external parallel-port handshaking to determine when to read data from the external data latch. It then provides the necessary control signals to access the data.

Because the MACH465 CPLD is encased in a 208-pin package with a 25mm pin pitch, it could be difficult to program and then install the device on a printed circuit board (PCB) without damage due to excessive handling. Happily, AMD provided full boundary-scan (JTAG) capability in the MACH465 CPLD, which allows it to be programmed in the circuit. This is a tremendous advantage since the device only needs to be handled during assembly, and can be programmed according to need. Updates can be provided as well when desired. This is also useful in implementing different features in multiple products using the same hardware.

The MACH465 CPLD also features fixed, predictable, and deterministic delays throughout the part. This is important in this design to ensure reliable and synchronous data transmission throughout the multiplexing and demultiplexing stages of the design. Also, all macrocells in a logic block can be connected to an I/O cell through the output switch matrix. This matrix makes it possible to make significant design changes while minimizing the risk of a pinout change. All input and I/O pins have built-in pull-up resistors that help to minimize the external parts count.

Among the tasks delegated for the MACH465 CPLD are implementing a FIFO, using DRAM as the memory element and the CPLD as the control element. The CPLD will allow normal data accesses between the microcontroller and DRAM. In response to a signal from the microcontroller (not yet implemented) the MACH465 CPLD will begin to interleave FIFO data bound for the printer (which has been decompressed and preprocessed prior to being placed in the FIFO) with normal code and data-bus cycles. Once the word has been fetched from DRAM, it can be transferred to the printer a byte at a time.

The ability to expand this design is one of the goals we set in the beginning. The MACH465 CPLD has tremendous power and flexibility and has not yet been taxed in its abilities, except for pin count. We deliberately routed many of the control and bus signals through the CPLD, even though they were not being used, so that expansion would be mostly a matter of reprogramming the CPLD. This can be done in circuit.

In addition to the above, there are a few collateral duties for which the MACH465 CPLD is well suited. First among these is translating signals for the AM29205 microcontroller so that the memory interface can be 32 bits consistently. Inside the CPLD, we have programmed a multiplexer/demultiplexer designed to switch the lower order of data and instruction lines to and from the upper ones for the AM29205 microcontroller, based on both the signal that indicates which processor is installed (PIO5) and the A1 address line.

Fairchild Semiconductor's boundary-scan components are used to handle the hardware aspect of boundary scan. The Multidrop and Hierarchical Addressable JTAG Port, SCANPSC110F Bridge, provides an interface from one JTAG master port on a backplane or some other source, in this case the Corelis interface described earlier. In our design, we included the SCANPSC110F to allow multiple local test access ports (TAPs)—up to three per chip—to be created. Connecting the G.Host to a Corelis PC boundary-scan interface, we can test each circuit individually and program the MACH465 CPLD in the circuit remotely, if we wish. The three ports allow for elements which may be missing at times

(e.g., the RF link). The remaining paths can still be tested without changing the boundary-scan test routines for the others.

The Embedded Boundary Scan Controller, the SCANPSC100F is an 8-bit parallel-to-TAP bidirectional translator that allows us to implement the boundary-scan power-on self-test from the microcontroller. It is routed through the SCANPSC110F bridge.

Because of its position relative to the external boundary-scan test access port, the SCANPSC100F controller can enable the AM29200 family microcontroller to act as a test-bus master. The 8-bit parallel communications are translated into boundary-scan serial format and relayed to (and from) the rest of the system. To avoid confusion, insure that the external TAPs are set to an address different than the one used for the G.Host's SCANPSC100F bridge (U7).

Note: The IEEE Std 1149.1 boundary-scan specification requires only four serial signals. These are: Test Clock (TCK), Test Mode Select (TMS), Test Data Out (TDO), and Test Data In (TDI). The SCANPSC100F implements just these four required signals. There is a fifth, optional signal called Test Reset ($\overline{\text{TRST}}$) that is used by some devices. If your external path should make use of this signal, you will have to provide for it in additional circuitry. (One of the AM29200 family microcontroller I/O-port pins driving a simple buffer would work nicely.)

MEMORY

The ROM for this design uses two AMD 29F400 5V-only Flash memories, in a 256k x 32-bit wide arrangement (that's two 29F400 memory devices, each in a 256k x 16 setup). Except for the initialization sequence at power-on reset, the memory is accessed via a 32-bit interface, although 16-bit writes will be allowed in all but the boot-block area (see *Figure 8*). This allows the program, look-up tables, or identification bytes stored in the ROM to be updated in the circuit. The core routines to accomplish the update will reside in the boot-block area where they cannot be corrupted. Communications routines necessary to this process will also be protected.

During the initialization of the system, the microcontroller will boot in 16-bit mode. The MACH465 CPLD will be under self-test, so there is a need to have an alternate path for program execution. A Fairchild SCAN18245T transceiver provides this path during initialization in response to the ROM chip-select signal that is active during initialization (when the microcontroller is using the start-up memory area). A few control signals need to be synthesized during this time and, of course, the MACH465 CPLD will be under test, so it is not available to perform this function. We solved this dilemma by adding a PALCE22V10.

DRAM is organized as 4M x 32 bits wide, for 16 Mbyte of DRAM (see *Figure 9*). This is to accommodate page buffering for the printer and allow sufficient room to implement a FIFO controlled by the MACH465 CPLD. There is also enough room to store data tables or program updates, which can be installed by the microcontroller using an algorithm in its boot-block memory area.

A portion of the DRAM is to be used as the memory element of a FIFO, which will be controlled by the MACH465 CPLD. By adding this capability, we will not have to wait for the printer's response before returning to time-critical activities. Rather, we can store information in the DRAM FIFO area in 32-bit format, and retrieve it in 8-bit format for transfer to the printer. The printer transfer part of the operation is independent of the microcontroller.

Memory map and PIO control line assignments are shown in *Table 1* and *Table 2*.

SUMMARY

The G.Host reference design is very straightforward and simple, designed for high performance while providing great flexibility. The schematics and data sheets included with the G.Host Reference Design Kit may be used as a springboard for other designs with only minor modifications required, or as it stands, if that meets your need. The components discussed here are easy to use, and they offer a great deal of power and performance—enough to tame the toughest embedded project.

RESOURCES

Hamilton Hallmark Technical Support Center

1-800-605-3296 ; http://www.tsc.hh.avnet.com

Fairchild Semiconductor Technical Resource Group

1-800-272-9959 ; http://www.fairchildsemi.com

Advanced Micro Devices

1-800-222-9323 ; http://www.amd.com

Corelis

1-310-926-6727

MEMORY MAP

The G.Host maps functions to memory address in *Table 1*.

TABLE 1. Memory Map

Address Space (hexadecimal)	AM29200 Microcontroller Selection	G.Host Assignment	Maximum Size	
			AM29200	AM29205
0000 0000— 03FF FFFF All ROM Banks	ROM Bank 0 ROM Bank 1 ROM Bank 2	Boot Code Application Code Self-Test Port	64 Mbyte Bank Size User Set	12 Mbyte Bank Size User Set
4000 0000— 43FF FFFF All DRAM Banks	DRAM	Data Memory	64 Mbyte	12 Mbyte
5000 0000— 50FF FFFF All Mapped DRAM	Mapped DRAM	Not Used	16 Mbyte	16 Mbyte
6000 0000— 63FF FFFF	VDRAM Transfers	Not Used	64 Mbyte	
8000 0000— 8000 00FC	Internal Peripheral Registers	(See the following table for addresses)		
9000 0000— 90FF FFFF	PIA Area 0	Not Used	16 Mbyte	4 Mbyte
9100 0000— 91FF FFFF	PIA Area 1	Not Used	16 Mbyte	4 Mbyte
9200 0000— 92FF FFFF	PIA Area 2	Not Used	16 Mbyte	
9300 0000— 93FF FFFF	PIA Area 3	Not Used	16 Mbyte	
9400 0000— 94FF FFFF	PIA Area 4	Not Used	16 Mbyte	
9500 0000— 90FF FFFF	PIA Area 5	Not Used	16 Mbyte	
All Others	Reserved			

PIO CONTROL LINE ASSIGNMENTS

The G.Host assigns the peripheral input/output control lines of the AM29200 family microcontrollers as follows:

TABLE 2. PIO Control Line Assignments

PIO Line	G.Host Function	Data Direction
PIO15	Serial Port CTS	Input
PIO14	Serial Port RTS	Output
PIO13–PIO8	Spares	(Not yet assigned)
PIO7–PIO0	(Not used)	

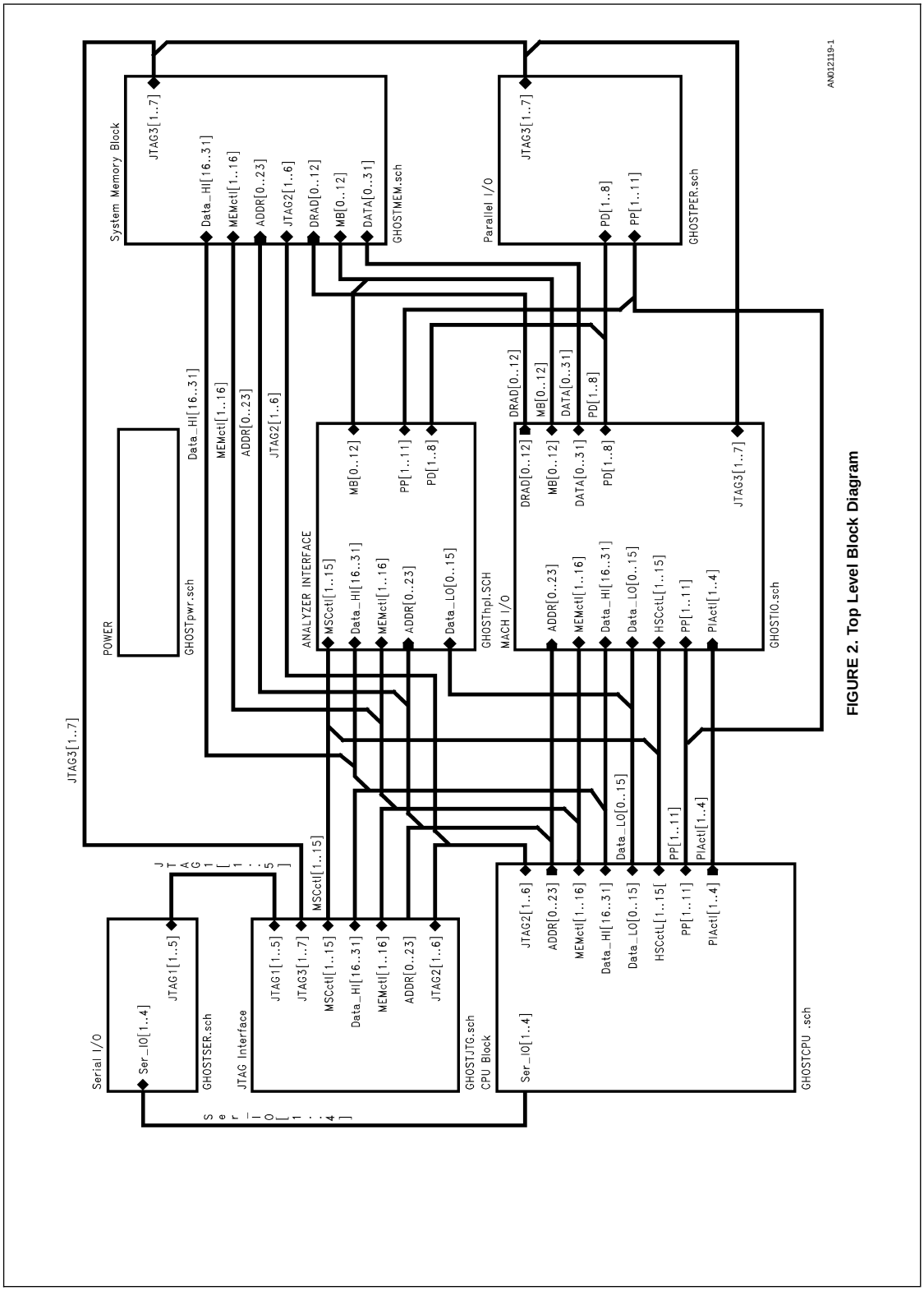

FIGURE 2. Top Level Block Diagram

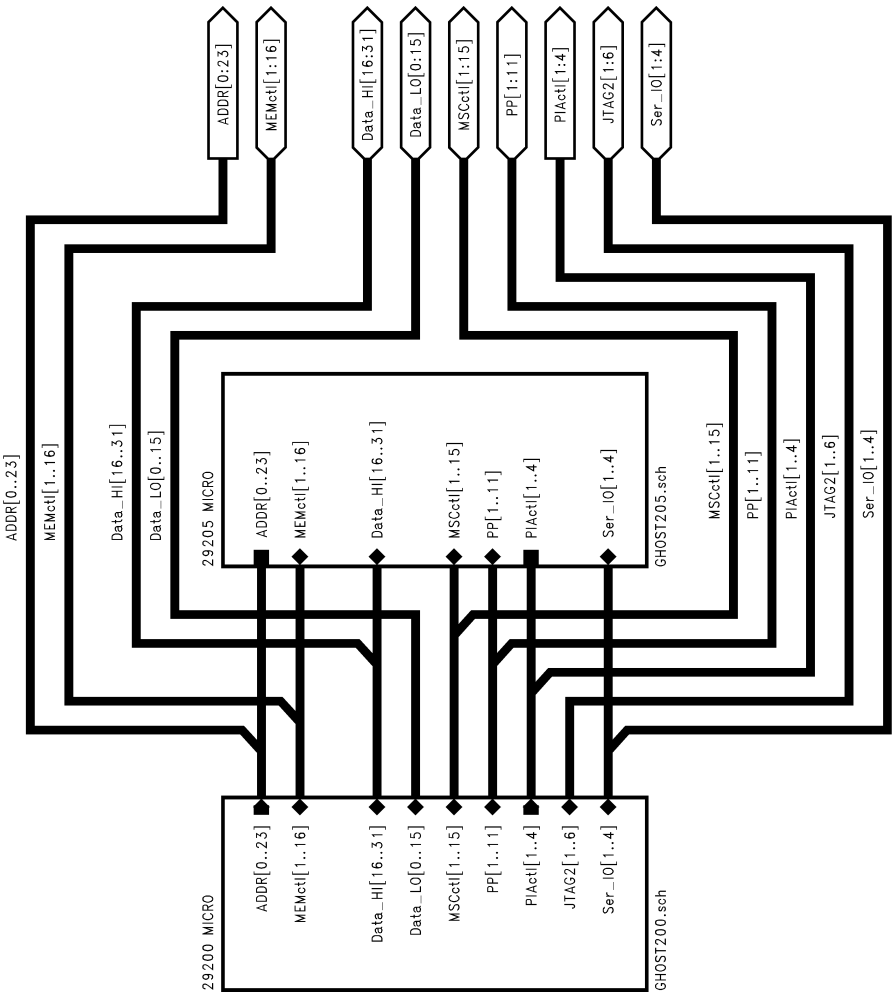

FIGURE 3. CPU Block

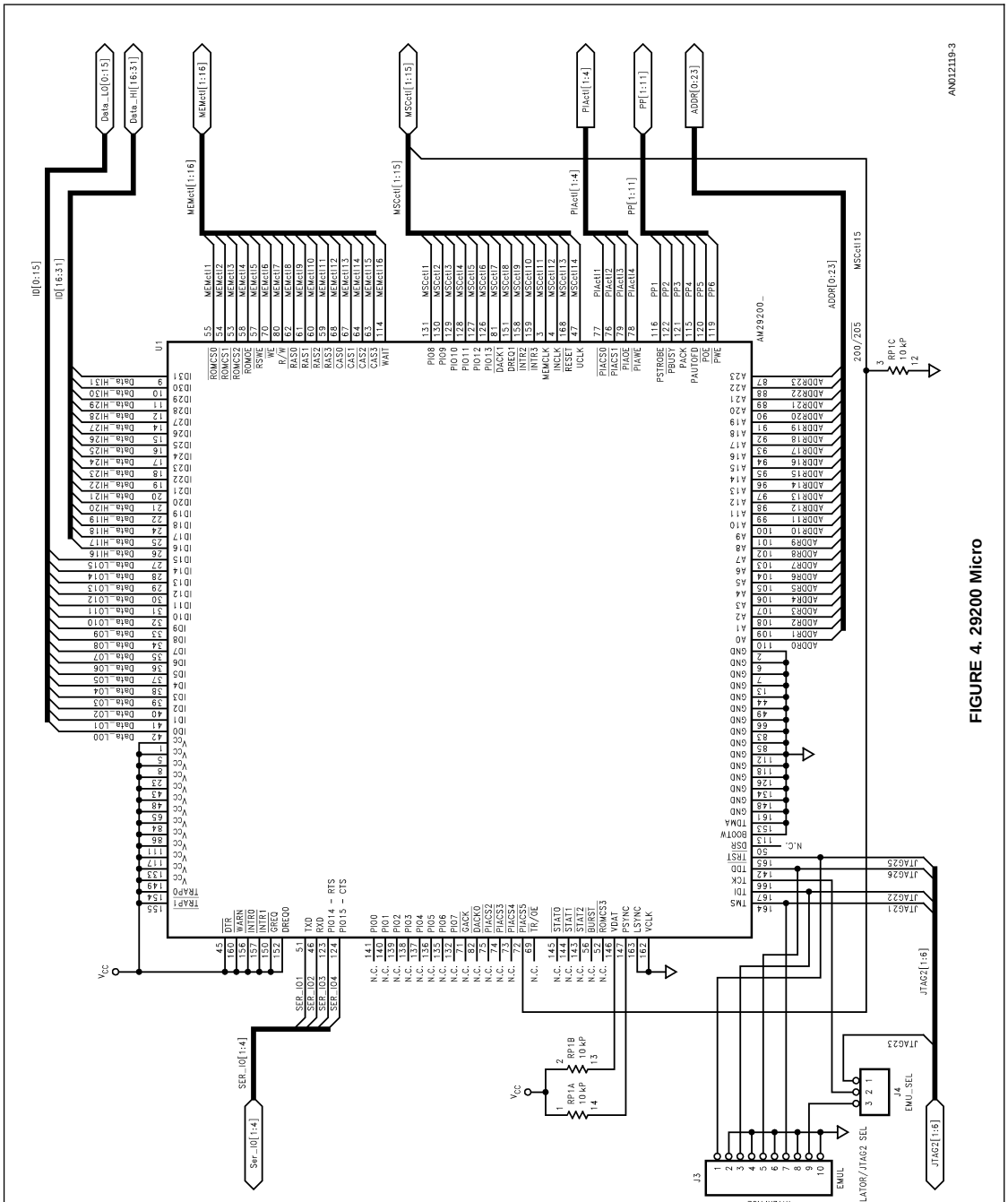

FIGURE 4. 29200 Micro

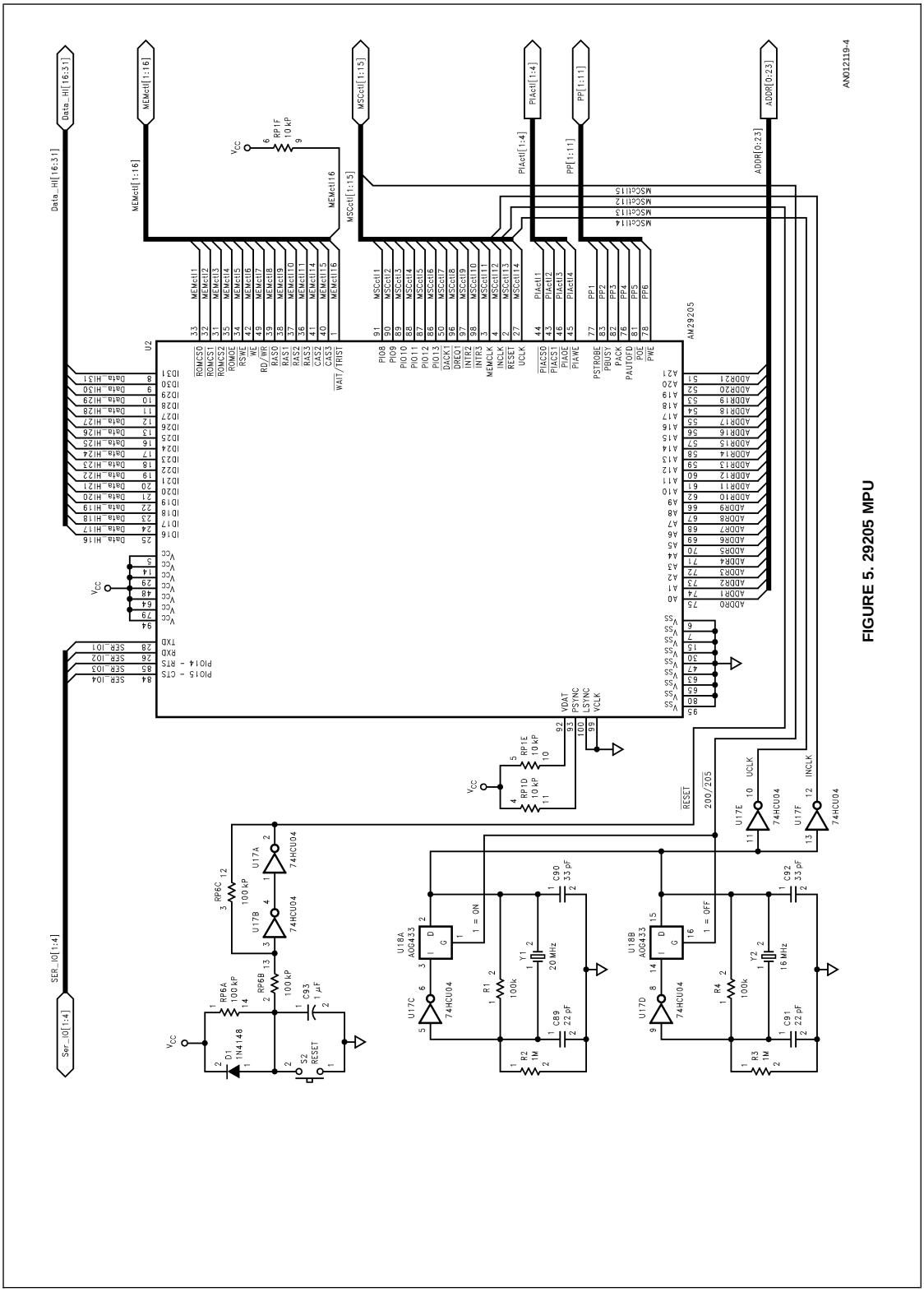

FIGURE 5. 29205 MPU

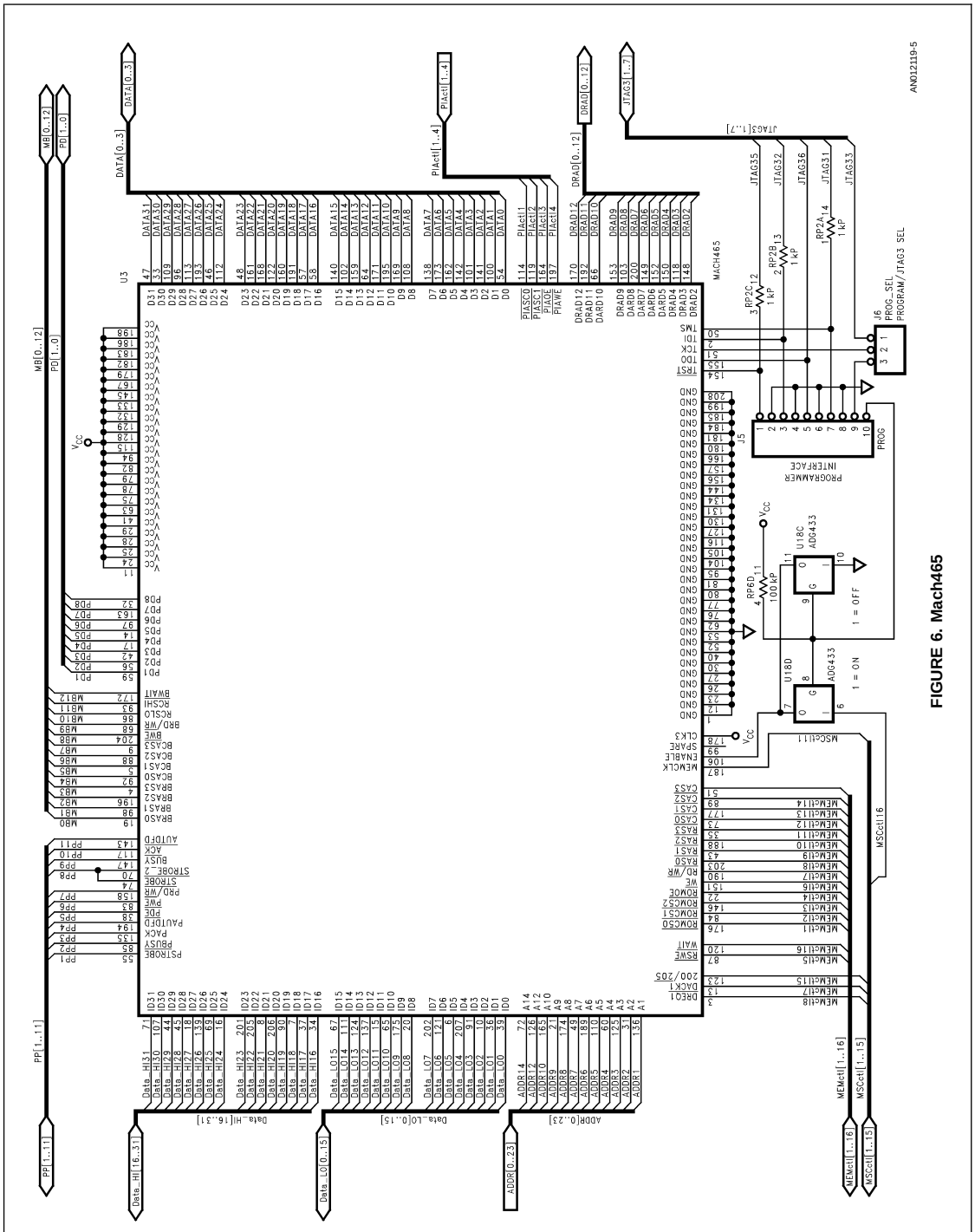

FIGURE 6. Mach465

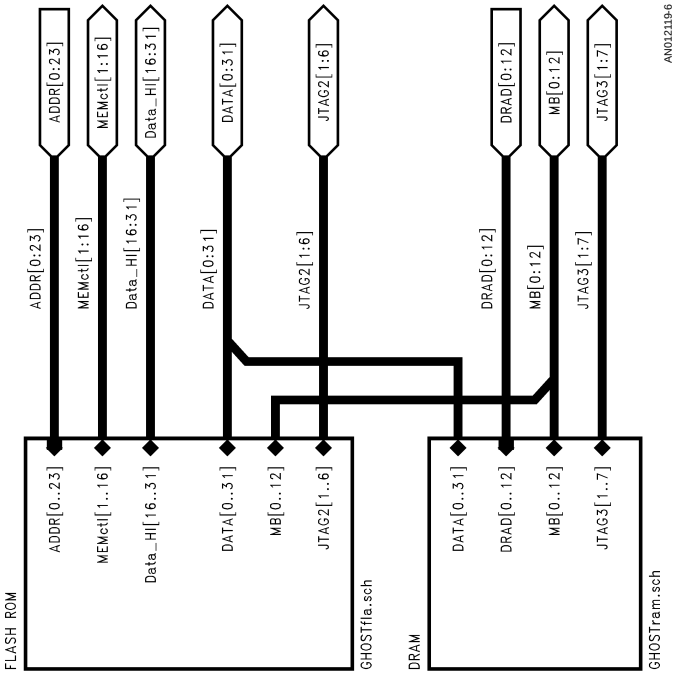

FIGURE 7. System Memory Block

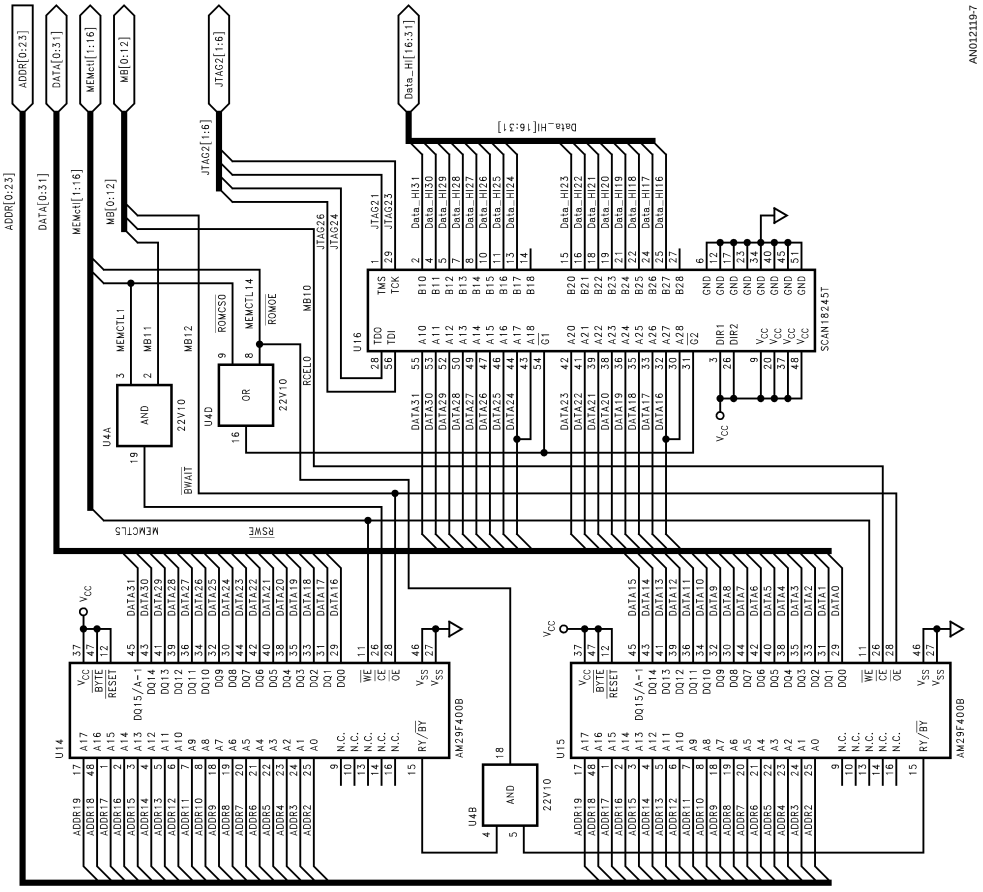

FIGURE 8. FLASH Memory

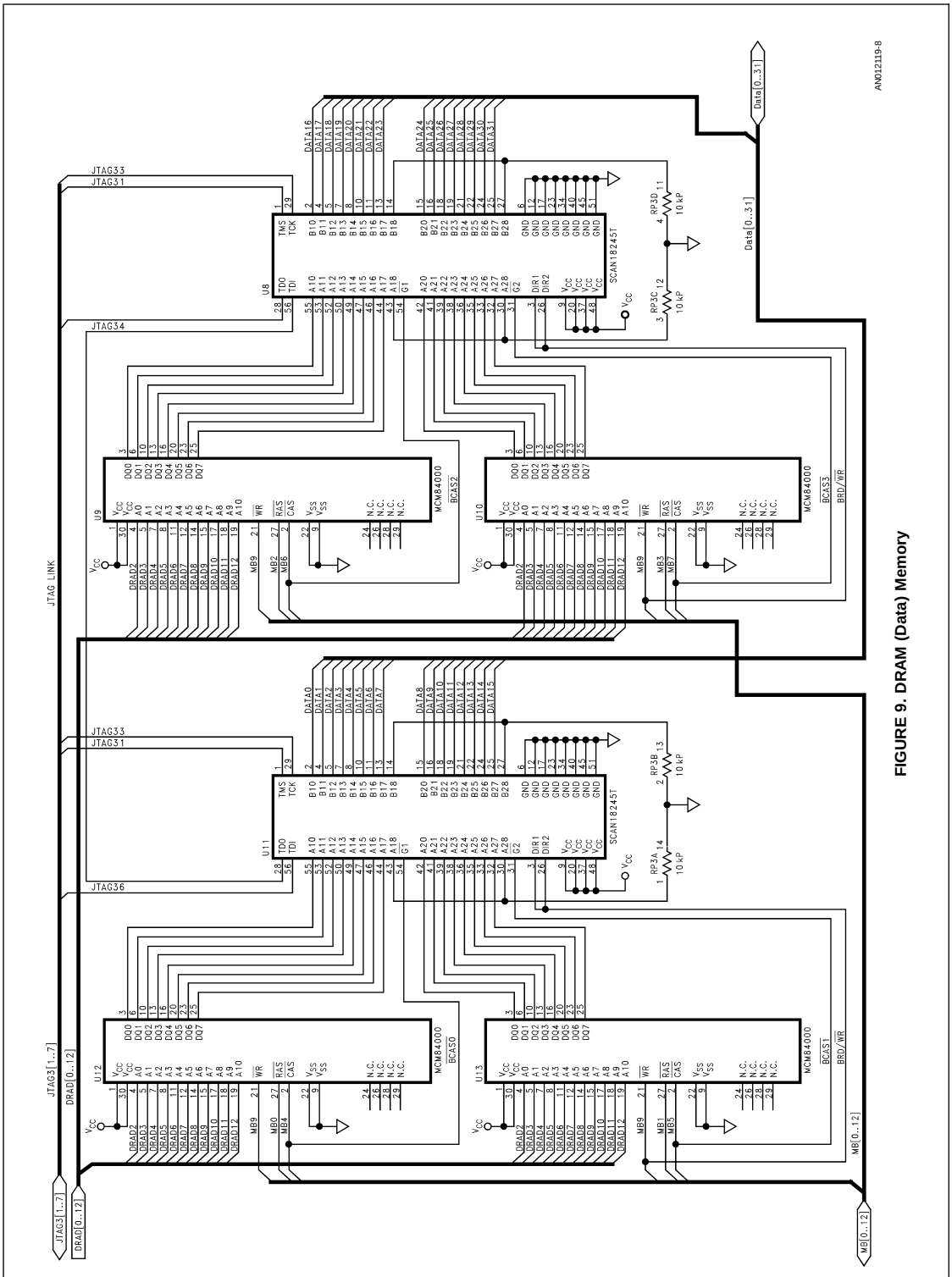

FIGURE 9. DRAM (Data) Memory

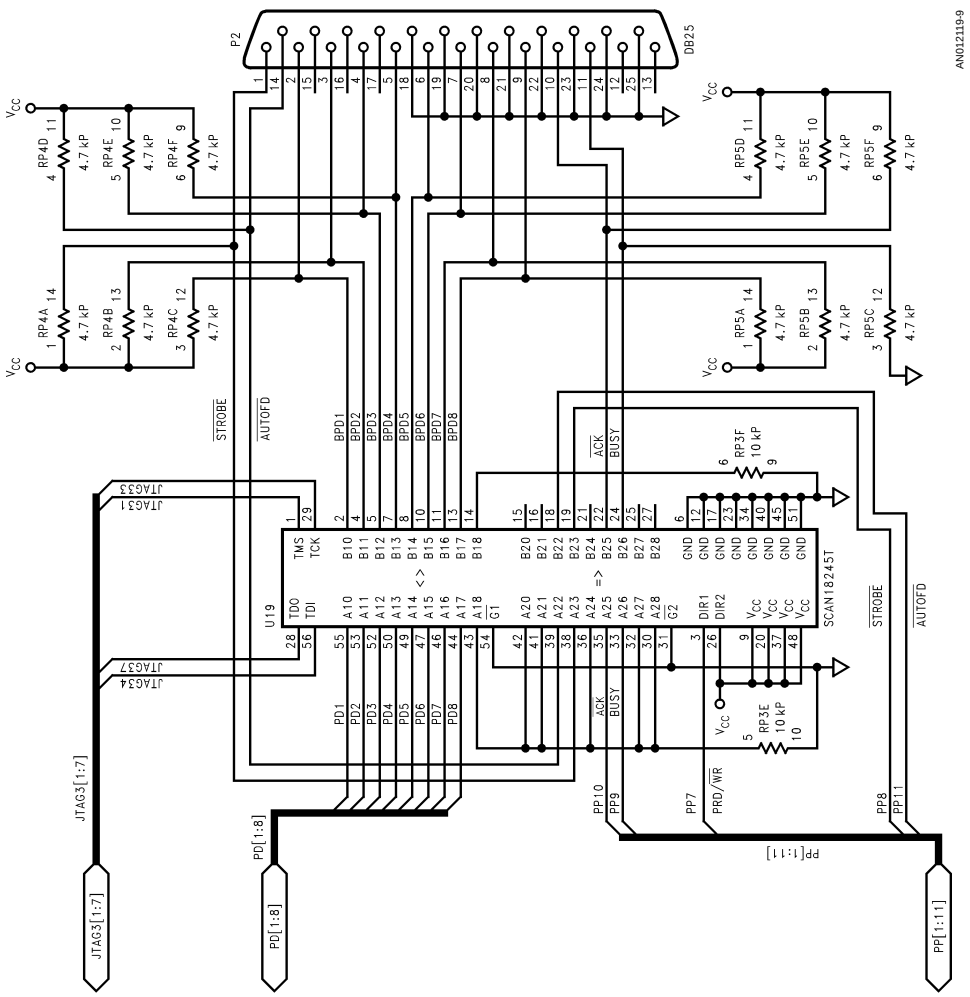

FIGURE 10. Parallel Interface

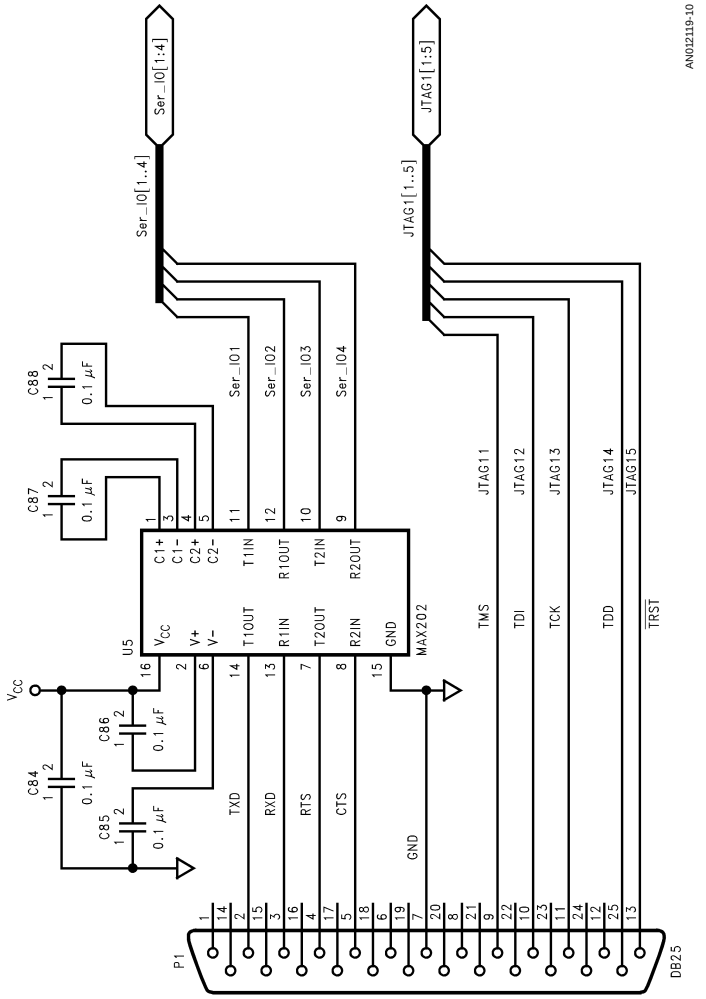

FIGURE 11. Serial Port

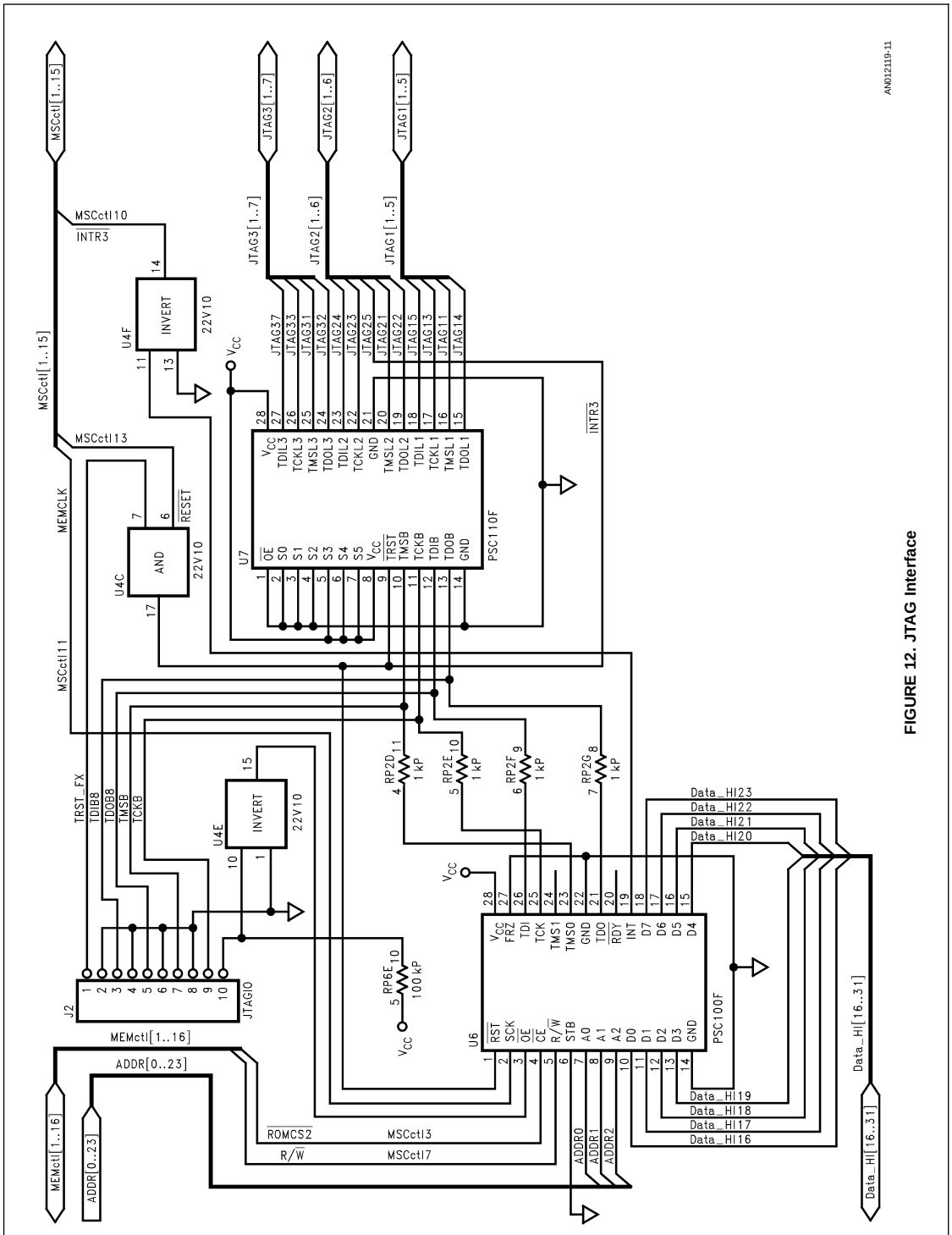

FIGURE 12. JTAG Interface

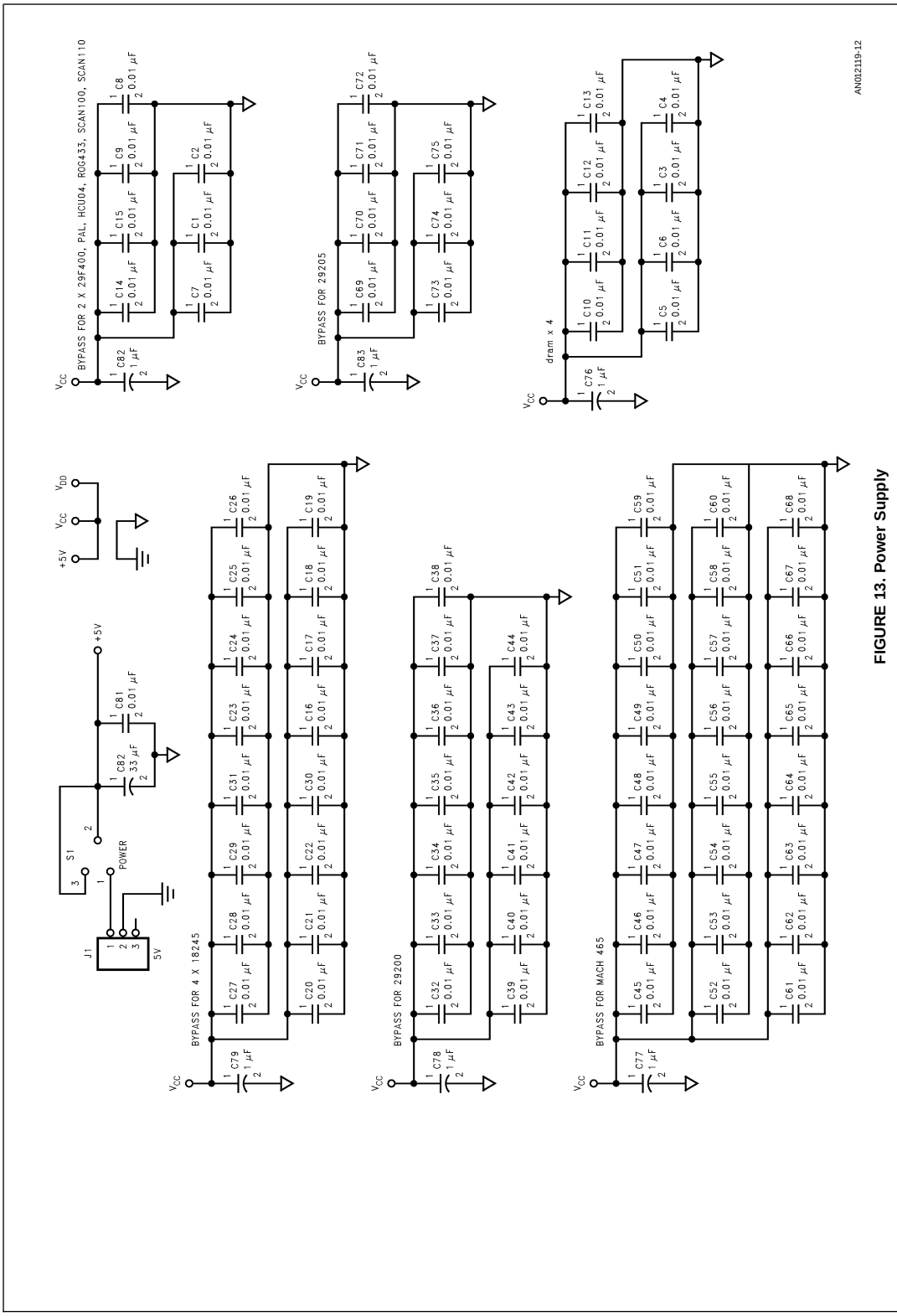

FIGURE 13. Power Supply

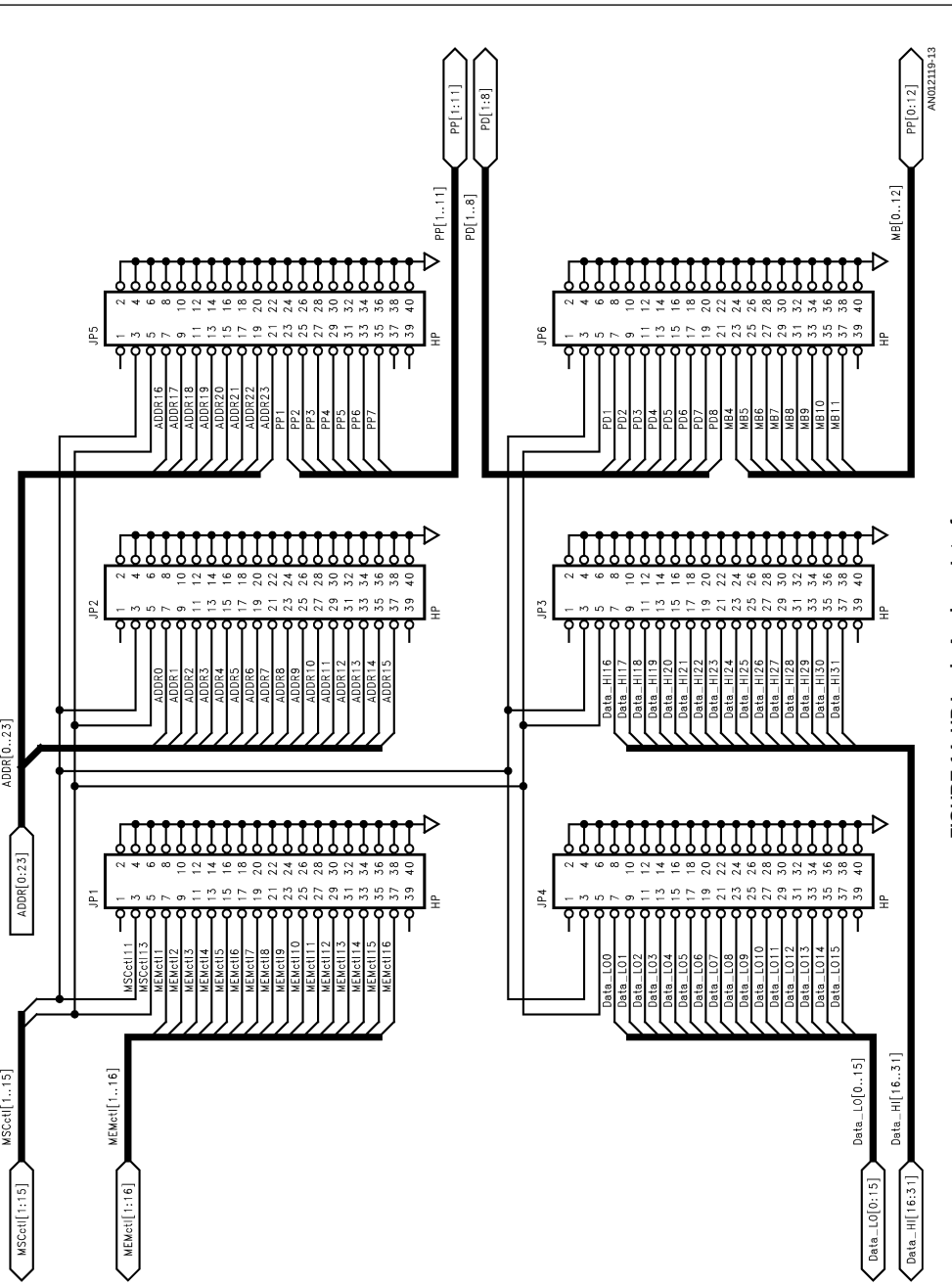

FIGURE 14. HP Logic Analyzer Interface

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Fairchild Semiconductor Corporation
Americas
Customer Response Center
Tel: 1-888-522-5372

Fairchild Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 8 141-35-0
English Tel: +44 (0) 1 793-85-68-56
Italy Tel: +39 (0) 2 57 5631

Fairchild Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: +852 2737-7200
Fax: +852 2314-0061

National Semiconductor Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179

www.fairchildsemi.com