Odd Number Divide By Counters With 50% Outputs and Synchronous Clocks

Prepared by: Cleon Petty and Paul Shockman
Product Applications
ON Semiconductor

The application inquiries handled by the Product Applications gives opportunities to solve customer needs with new ideas and learn of ways the customer has used our devices in new applications. A couple of these calls lead to techniques of designing odd number counters with synchronous clocks and 50% outputs.

The first technique requires a differential clock, that has a 50% duty cycle, a extra Flip Flop, and a gate to allow Odd integers, such as 3, 5, 7, 9, to have 50% duty cycle outputs and a synchronous clock. The frequency of operations is limited by Tpd of the driving FF, Setup, and Hold of the extra FF, and the times cannot exceed one half on the incoming clock cycle time.

The design begins with producing a odd number counter (Divide By 3 for this discussion) by any means one wishes and add a flip flop, and a couple of gates to produce the desired function. Karnaugh maps usually produce counters that are lockup immune.

Example:
Specify, Divide By 3,
50% duty cycle on the output
Synchronous clocking
50% duty cycle clock in

Using D type Flop flips and karnaugh maps we find;
\[ Ad = A*B* \] and \[ Bd = A \]
(Note: * indicates BAR function)

Figure 1 shows schematic and timing of such a design.
Using the technique, we add a gate on the clock to get differential Clock and Clock bar, a flip flop that triggers on the Clock Bar rising edge (Clock Neg.) to shift the output of "B" by 90 degrees and a gate to AND/OR two FF output to produce the 50% output. We get Figure 2, a Divide By 3 that clocks synchronously with 50% output duty cycle.

The Max frequency of the configuration (figure 2) is calculated as Clock input freq./2 = Tpd of FF "B" + Setup of "C" + Hold of "C".

Example:

Tpd = 1Ns, Setup = !NS and Hold time = 0Ns.

with these numbers the Max Frequency the configuration can expect is; Cycle time = 2*(1 + 1)Ns or 4 Ns that converts to 250MHZ.

The Method is usable on other divide by "N" counters as well by using the same methodology. The use of different types of Flip Flops (J,K, S,R, Toggle, ETC.) may produce fewer components. The type logic used may also dictate configuration. The configuration should always be checked for lockup conditions before the design is committed to a production.

Example:

A Divide By 3 design has all possible states shown in chart 1 but uses only the states shown in chart 2 leaving the states 2,3,4,5, & 7 for possible lockup.

<table>
<thead>
<tr>
<th>Chart 1</th>
<th>Chart 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C</td>
<td>A B C</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td>2 0 1</td>
<td>6 0 1</td>
</tr>
<tr>
<td>3 1 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4 0 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>5 1 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>6 0 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>7 1 1</td>
<td>1 1 0</td>
</tr>
</tbody>
</table>
We need to know that the counter will go into the flow, shown in chart 2, if it happens to come up in one of the unused states at powerup or for any other reason. Figure 3 shows the resulting flow chart of the analysis of the Divide By 3 counter of Figure 2. There is no state that the counter can begin in that doesn’t lead to the desired flow after one clock cycle.

![Flow Chart](image)

**Figure 3.**

Observation shows that FF "C" follows FF "B" by a half a clock cycle and will never be able to lockup making the analysis of the Divide By 3 sufficient to assure the whole configuration will have no lockup flow. So; only the 1 1 state of the divide by three needed to be confirmed.

The method is extendible to other odd larger divide by "N" numbers by following the same design flow.

a) Design a stable UP or Down divide by "N" counter
b) Make the Clock input a 50% duty cycle differential signal
c) Add a FF to follow one of the FF's in the counter by 1/2 clock cycle
d) OR/AND the shifted FF with the one that is driving it to obtain the desired 50% output

**Example:**

Design a 50% Divide By 9
Use "D" type FF's, other types may give smaller component count
Karnaugh maps yield:
\[\begin{align*}
Ad &= A*B* \\
Bd &= A*B + AB* \\
Cd &= ABC* + CB* + A*C \\
Dd &= ABC
\end{align*}\]
Choosing to use "C" as the flip flop to delay by a 1/2 clock cycle is necessary to accomplish the 50% output required when "ANDed" with "E".

Another Synchronous 50% counter for Divide By 6, 10, 12, 14, 18, etc. can be realized by the additions of a J K FF and some gates. Other types of FF’s may be used.
Take the before mentioned Divide By 3 add a J K and a divide by 6, 50% duty cycle, synchronous counter is realized as shown in Figure 5.

Of course, there are better ways to realize a Divide By 6 but it does demonstrate how the method works. Note this configuration does not require a 50% input clock duty cycle and it is synchronous. This type of configuration could be useful in a clock generating PLL chip where a Divide By 3 and Divide By 6 are needed to synchronize two signals as shown in figure 6.
Notice FF "A" was chosen as the FF to drive FF "E" in order to align the positive edges of the clock, Divide By 3, and divide by 6. The overall skew of the output could be better matched if all the same type of FF and gates are used.

We already know the Divide By 3 is lockup immune, following flow chart Figure 7 shows that the addition of the J K does not change that situation for the Divide By 6.

The flow shows no lockup, but if one observes that the J K is a sort of toggle device it is obvious that it can't lock up the counter.

The J K may need bigger input AND gates to accomplish larger divide numbers. As an example, pick a Divide By 12 and use J K type FF’s to do the function.
Maps show:
Ja = 1  JB = AC  Jc = AB
Ka = 1  Kb = A  Kc = A

Figure 8 shows the implementation.

*Examination of the truth table shows that the FF "D" must
decode a 5 and a 13 in order to make the desire 50% function.
The inputs to the "D" FF are J = ACD and K = ACD and
requires 3 input AND gates. For larger counters the inputs
on the AND gates will need to increase to reach the desired
configuration. However for the single digit integers such as
3, 5, 7, & 9 to realize 6, 10, 14, & 18 a fan in of three is max.

The methods are expandable. A little observation,
thinking, and logic typing will allow the designer to
minimize the component count and skew on this type of
counter.