CS1026 – Digital Logic Design

More Synchronous Logic

Shane Sheehan [1]

[1]ADAPT
Trinity College Dublin

March 20, 2017

Today's Overview

1 One more example

2 How fast can we go

3 Adding an Enable

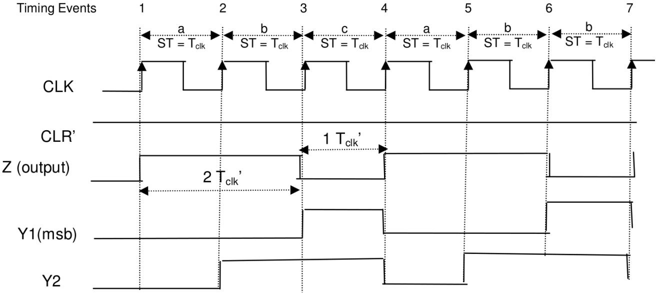
Classical Design Revisted [Nelson et al., 1995] I

Example Style Question

Design a synchronous sequential circuit called *Div-by-3* having an output Z that divides the system clock frequency f CLK by 3.

- Use an output duty cycle of two-thirds
  - 2 CLK cycle high, 1 cycle low
- Design the circuit using positive-edge-triggered flip-flops

Classical Design Revisted [Nelson et al., 1995] II



Step 1: Start with a timing diagram

Classical Design Revisted [Nelson et al., 1995] III

Step 2: Determine the number of flip-flop based on the number of State

- No. states $= 3 \leq 2$
- Assuming Full Coding

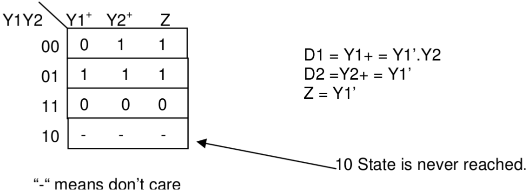
Classical Design Revisted [Nelson et al., 1995] IV

Step 3: Assign a unique code to each state

- A – 00
- B – 01
- C – 11

Classical Design Revisted [Nelson et al., 1995] V

Step 4: Write the excitation-input equations



Y1Y2	$Y1^+$	$Y2^+$	Z
00	0	1	1
01	1	1	1
11	0	0	0
10	-	-	-

D1 = Y1+ = Y1’.Y2
D2 =Y2+ = Y1’
Z = Y1’

10 State is never reached.

“-“ means don’t care

- The D flip flop excitation equation: $D = Y+$

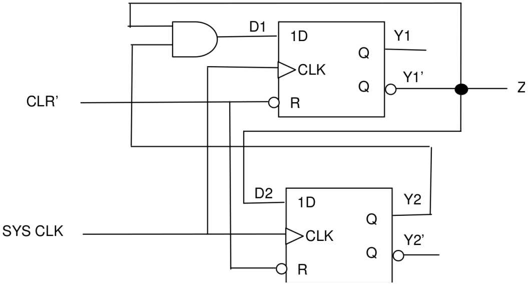
Classical Design Revisted [Nelson et al., 1995] VI

More step 4..

- We need the composite K-map for each of the desired outputs: $Y1$, $Y2$, Z

Classical Design Revisted [Nelson et al., 1995] VII

Step 5: Draw the Circuit Schematic



- We skip steps 6 and 7..
    - But you should do this in the lab/exam!

Classical Design Revisted [Nelson et al., 1995] VIII

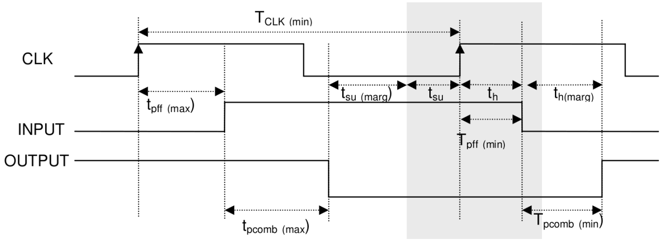
The Next Lab

Design a synchronous sequential circuit identical to the previous example, except implement the design using JK flip-flops instead of D flip-flops

Determining the Maximum Clock I

The maximum clock frequency that a system can handle is driven by the set-up, hold and margin times required by the flip flops in the synchronous system.

Determining the Maximum Clock II



We can see that the clock frequency becomes limited by:
$f_{max} = 1/T_{CLK(min)}$

Determining the Maximum Clock III

$$T_{CLK(min)} = t_{pff(max)} + t_{pcomb(max)} + t_{h(marg)}) + t_{su} + t_h \text{ where:}$$

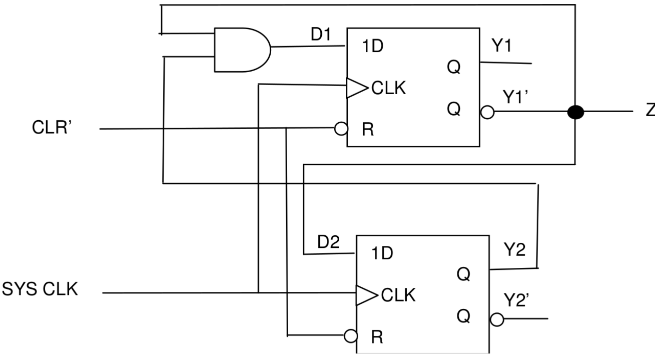
- $t_{pff(max)}$ – Maximum propagation delay time through flip-flop from the clock tick to Q output
- $t_{pcomb(max)}$ – Maximum propagation delay time through combinational logic
- $t_{h(marg)}$ – Margin time, it is always a good design practice to allow for tolerances.
- t_{su} – Set-up time requirement
- t_h – Hold time requirement

Determining the Maximum Clock IV

An Example

Determine the absolute maximum clock frequency for the divide-by-3 synchronous machine

Determining the Maximum Clock V



Remember the Schematic!

Determining the Maximum Clock VI

Going through the ICs:

- 74LS08 AND gate
  - t_{pcomb} : Min at 3 ns and Max. at 18 ns
- 75LS175 D-flip-flop
  - t_{pff}: Min at 0 ns and Max. at 42 ns
  - t_{su}: Min at 20 ns
  - t_h: Min at 0 ns

Determining the Maximum Clock VII

So let's find the fastest clock speed:

- $T_{CLK(min)} = t_{pff(max)} + t_{pcomb(max)} + t_{h(marg)} + t_{su} = 42 + 18 + 0 + 20 = 80$ ns
- $T_{CLK(max)} = 1/T_{CLK(min)} = 1/80 \times 10^{-9} = 12.5$ MHz

How does this compare?

- 12.5 MHz $\implies$ Slow
  - In comparison to modern tech

Using Enables I

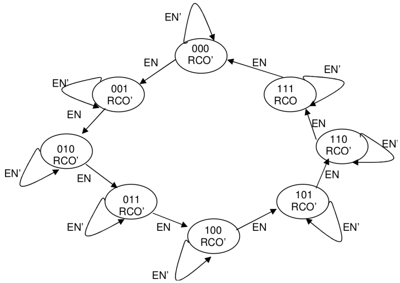
Used to stop the count at times and then continue counting

- Basis of *Full-Encoded Stoppable Counter*
    - Counter will count up as long as EN high
    - Else STOP counting

Using Enables II

State Diagram for a three-bit $(Y1, Y2, Y3)$

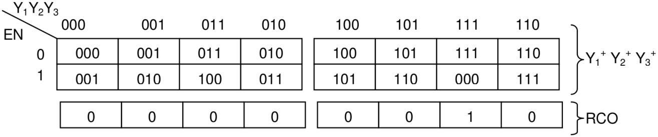
- Also called a *Full-Encoded Stoppable Counter*



Using Enables III

Composite K-map for a 3-bit binary up stoppable counter with:

- Enable input EN
- Asynchronous clear input CLR
- Ripple-carry out RCO.



EN \ $Y_1Y_2Y_3$	000	001	011	010	100	101	111	110	
0	000	001	011	010	100	101	111	110	$Y_1^+ Y_2^+ Y_3^+$
1	001	010	100	011	101	110	000	111	
	0	0	0	0	0	0	1	0	RCO

Note: CLR=1 → $Y_1Y_2Y_3 = 000$

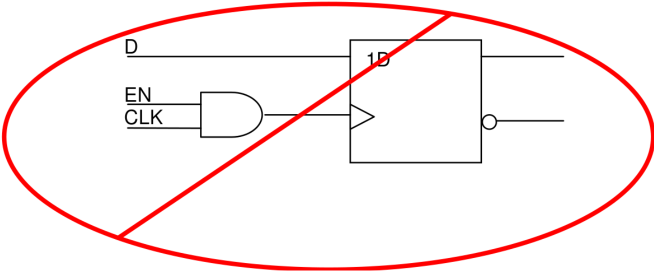
Using Enables IV

We derive the flip-flop input excitation equation and RCO output equation from the composite K-map or (need 3 flip-flops):

- $D1 = Y1^{+} = EN.Y1'.Y2.Y3 + Y1.Y2' + Y1.Y3' + EN'.Y1$
- $D2 = Y2^{+} = EN.Y2'.Y3 + Y2.Y3' + EN'.Y2$
- $D3 = Y3^{+} = EN.Y3' + EN'.Y3$
- $RCO = Y1.Y2.Y3$

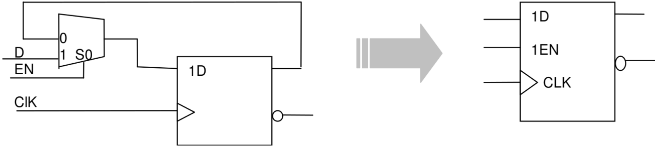
Using Enables V

In order to maintain the benefits of a synchronous system (avoiding clock glitches), it is important that the clock to all of the components remains the same.



- I.E. NOT TO DO THIS!

Using Enables VI



Instead, if you need to enable a flip-flop, use one with enable capability designed in or use the MUX as shown

Using Enables VII

Moral of the story:

- Flip-flops with enable allows the designers to focus on input/output sync
- Enabled flip-flops simply require a connection to the enable pin

References (Homework) I

Nelson, V. P., Nagle, H. T., Carroll, B. D., and Irwin, J. D. (1995).
Digital logic circuit analysis and design.
Prentice-Hall, Inc.