CS1026 – Digital Logic Design

Sequential Logic Analysis

Shane Sheehan [1]

[1]ADAPT
Trinity College Dublin

March 6, 2017

Today's Overview

1 FSM – A quick recap

2 Example 1

3 Optional Homework.. :-O

Synchronous Finite State Machine Design I

Common Examples of Synchronous FSM

- Up and Down Binary Counters
- Shift Registers
- Sequence Detectors
- Controllers

Synchronous Finite State Machine Design II

We use the:

- Seven-Step Design Process for Synchronous Sequential Design
- Also called *Classical Design*

Synchronous Finite State Machine Design III

Organising the Design Specifications Use one or more of the following tools:

- Timing Diagram
- State Diagram
- ASM Chart

Synchronous Finite State Machine Design IV

Determine the number of flip-flops based on the number of states

- You may use full encoding or *one-hot* encoding
- Full encoding utilise all possible combinations of the flip-flops

How many flip flops?

To decide the number of flip-flops we use the following inequality: $2^n \geq$ No. of States

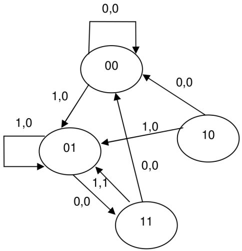
Synchronous Finite State Machine Design V

One-hot encoding $\Longrightarrow$ when the flip-flop has an output 1

Thus:

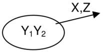
- The number of flip-flops = No. of States
- Once we know the number of flip-flops
    - Assign one variable for each of the flip-flop output

Synchronous Finite State Machine Design VI



Classic State machine
* Links show input, output in 1s and 0s
* State is inside the circles

Legend



Notes

1) State 00 is reset

2) Output Z=1 only when the input sequence is 101, so this could be "101" pattern detector.

3) State "10" is referred to as "illegal state", "unused state" or an "unreachable state".

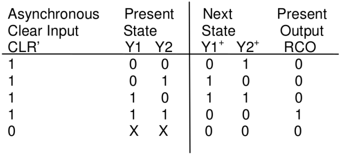
4) One way to ensure you don't end up in illegal state is to have a power on reset.

Assign a unique code to each state:

- A specific value for present state variables

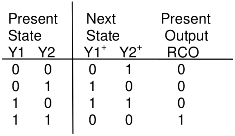
Synchronous Finite State Machine Design VII

Select the flip-flop type and draw the Present State/Next State (PS/NS) table:



Asynchronous Clear Input CLR'	Present State Y1	Y2	Next State $Y1^+$	$Y2^+$	Present Output RCO
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	0	1
0	X	X	0	0	0

Present State / Next State (PS/NS) Table



Present State Y1	Y2	Next State $Y1^+$	$Y2^+$	Present Output RCO
0	0	0	1	0
0	1	1	0	0
1	0	1	1	0
1	1	0	0	1

Simplified PS/NS Table
(Note: CLR'=0 → Y1Y2=00)

- This determines the excitation input equations and the Moore and/or Mealy output equations.

Synchronous Finite State Machine Design VIII

Remember the excitation input and next state relationship flip-flops:

- D flip-flops: $D = Y^{+}$ or $Y^{+} = D$
- J-K flip-flops: $J = Y^{+}$ or $K = Y^{+\prime}$ or $Y^{+} = JY' + K'Y$
- T flip-flops: $T = Y^{+} \oplus Y$ or $Y^{+} = T \oplus Y$

Synchronous Finite State Machine Design IX

The other *more practical* stages

- Draw the circuit schematic (paper or CAD tool).
- Perform a simulation to test the functionally of the design
- Implement the design with hardware

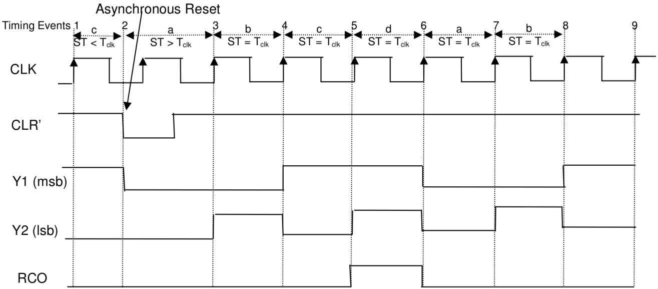
Ripple Carry Output Counter [Nelson et al., 1995] I

Our Task

- Design a 2-bit binary up-counter with a ripple carry output (*RCO*) using D flip-flops
- Note: The input CLR' denotes an asynchronous input that overrides the clock

Ripple Carry Output Counter [Nelson et al., 1995] II

A timing diagram helps..



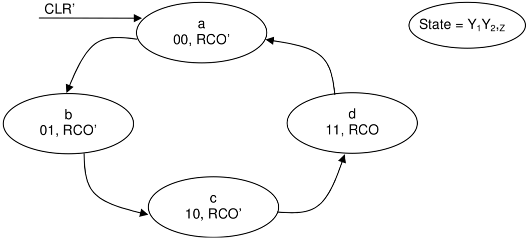
Ripple Carry Output Counter [Nelson et al., 1995] III

A few things to remember:

1. T_{clk} is the clock period and ST is the state time.
2. The first two events are less and then more than T_{clk}.
3. $Y1Y2$ are the states (counts).
4. RCO is a Moore output indicating when the maximum count has been reached.

Ripple Carry Output Counter [Nelson et al., 1995] IV

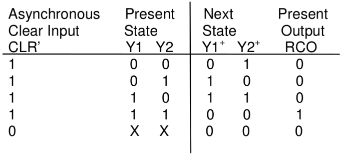
Although a timing diagram appears more complete, we can understand state diagram better:



- Since it does not contain the clock timing information.

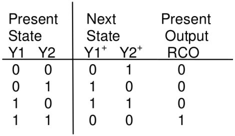
Ripple Carry Output Counter [Nelson et al., 1995] V

We can use a PS/NS table to describe the functionality (Step 1):



Asynchronous Clear Input	Present State		Next State		Present Output
CLR'	Y1	Y2	$Y1^+$	$Y2^+$	RCO
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0
1	1	1	0	0	1
0	X	X	0	0	0

Present State / Next State (PS/NS) Table



Present State		Next State		Present Output
Y1	Y2	$Y1^+$	$Y2^+$	RCO
0	0	0	1	0
0	1	1	0	0
1	0	1	1	0
1	1	0	0	1

Simplified PS/NS Table
(Note: CLR'=0 → Y1Y2=00)

Top hint

The *Next state, NS* defines the state of the machine

- During the next clock cycle

Ripple Carry Output Counter [Nelson et al., 1995] VI

Step 2:

- Determine the no. of flip-flops based on the no. of states
- For full encoding – no. of states $= 4 \leq 2^2$

Step 3:

- Assign Unique code to each state
  - We already did this in the state diagram

Ripple Carry Output Counter [Nelson et al., 1995] VII

Step 4:

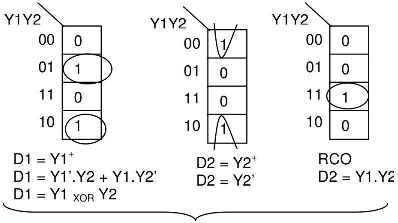
- Write the excitation-input equations

Remember

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

Ripple Carry Output Counter [Nelson et al., 1995] VIII

More Step 4:



Y1Y2	
00	0
01	1
11	0
10	1

$D1 = Y1^{+}$
$D1 = Y1'.Y2 + Y1.Y2'$
$D1 = Y1 \text{ XOR } Y2$

Y1Y2	
00	1
01	0
11	0
10	1

$D2 = Y2^{+}$
$D2 = Y2'$

Y1Y2	
00	0
01	0
11	1
10	0

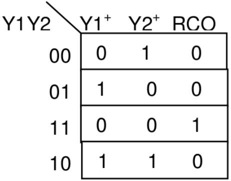
RCO
$D2 = Y1.Y2$

Excitation-inputs and output RCO equations
derived from separate K maps
(These equations are also called design equations)

- The K-map for each of the desired outputs: $Y1^{+}$, $Y2^{+}$, RCO helps us determine excitation-input equations

Ripple Carry Output Counter [Nelson et al., 1995] IX

Last bit of step 4:



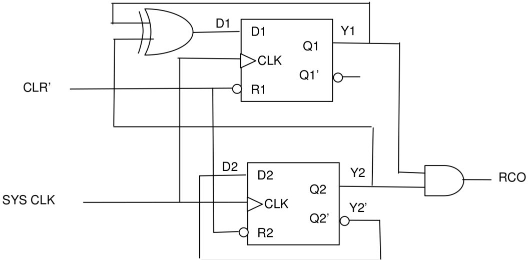
Y1Y2	$Y1^+$	$Y2^+$	RCO
00	0	1	0
01	1	0	0
11	0	0	1
10	1	1	0

A composite K-map is a short hand for multiple K-maps.

- Draw a composite K-map for each of the desired outputs:
    - *Y*1, *Y*2, *RCO*

Ripple Carry Output Counter [Nelson et al., 1995] X

Now relax.. (Step 5)



- Draw the Circuit Schematic

Ripple Carry Output Counter [Nelson et al., 1995] XI

We skip steps 6 and 7

- Remember to do them for labs
- And note them when doing exams
    - It shows understanding!

Some for you to try

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

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.