

CoE/EE460 Switching Theory

Lecture 16

Washington University
Spring 2001

<http://www.arl.wustl.edu/~lockwood/class/coe460/>

Copyright 2001, John W Lockwood

Announcements

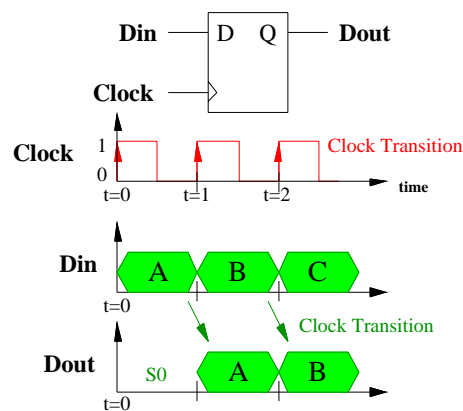
- Announcements
 - Homework 6
 - Due Monday - April 9 (5:00pm)
 - Exam 2
 - Exam 2 : Wednesday - April 18 (in class)
 - Details on Next slide
- Reading
 - Hachtel & Somenzi : Section III / Chapter 7
- Review
 - Solutions for Homework 5
- Course Evaluations

Exam 2 - Material

- Exam 2 : Wednesday - April 18 (in class)
 - Covers:
 - Lectures 8-16
 - Reading
 - Textbook, chapters 4 and 6
 - Logic Synthesis [Chapter 65 of VLSI handbook]
 - Homeworks 3-6
 - You may bring (only):
 - Boolean Algebra Theorems & Postulates sheet (on web)
 - Your own 2 pages of [2 sided] note sheets
 - You may not bring
 - Textbook
 - This is closed-book exam
 - Computation or communication device
 - Problems will be sufficiently small to work by hand

Review of Synchronous Storage Elements

- Synchronous Circuits
 - Values change at times governed by clock
- Clock
 - Input to circuit
- Clock Event
 - Time of transition
 - Example: Rising edge
- Flip/Flop
 - Transfers Value From D_{in} to D_{out} on Clock event



Finite State Machine (FSM)

- Terminology

- Time: t

- Increments on (rising) clock edge

- State: S

- State variables: $S = \{ S_1, S_2 \dots S_k \}$
- Current state: $S(t)$
- Next state: $S(t+1)$

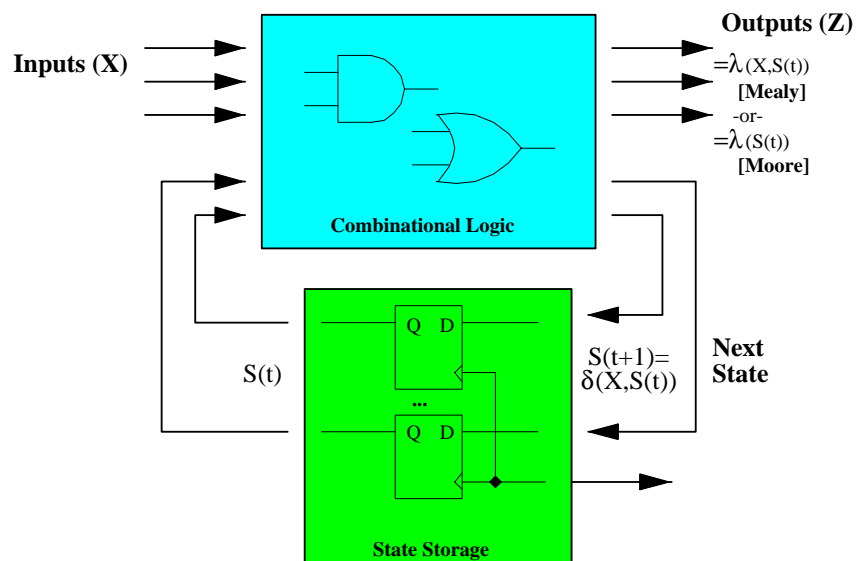
- State transition function $\delta()$

- Assigns next state

- State Transition

- $S(t+1) = \delta(X , S(t))$

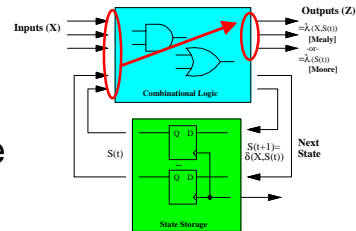
Operation of the FSM



Mealy and Moore FSMs

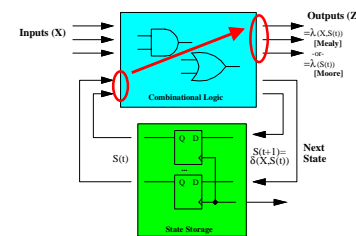
- Mealy Machine

- Output (Z) = $\lambda(X , S(t))$
- Output is a combinatorial function of input and state



- Moore Machine

- Output (Z) = $\lambda(S(t))$
- Output is a combinatorial function only of state
- Consider effect of input transition



Informal Synthesis Techniques

- Example

- Count the difference between the number of 1's minus number of 0's received on a network link during the last four bit transmission periods

- Informal Method

- Allocate Flip/Flops to store state
- Compute state transition function
 - Next State $S(t+1) = \delta(X , S(t))$
- Compute output function
 - Output (Z) = $\lambda(X , S(t))$ or $\lambda(S(t))$

Calculate Outputs

- $Z = \lambda(S(t))$

2's Complement
Encoding of Count

S_3	S_2	S_1	S_0	Count	Z_3	Z_2	Z_1	Z_0
0	0	0	0	0-4=-4	1	1	0	0
0	0	0	1	1-3=-2	1	1	1	0
0	0	1	0	1-3=-2	1	1	1	0
0	0	1	1	2-2=0	0	0	0	0
0	1	0	0	1-3=-2	1	1	1	0
0	1	0	1	2-2=0	0	0	0	0
0	1	1	0	2-2=0	0	0	0	0
0	1	1	1	3-1=+2	0	0	1	0
1	0	0	0	1-3=-2	1	1	1	0
		
1	1	1	1	4-1=0	0	1	0	0

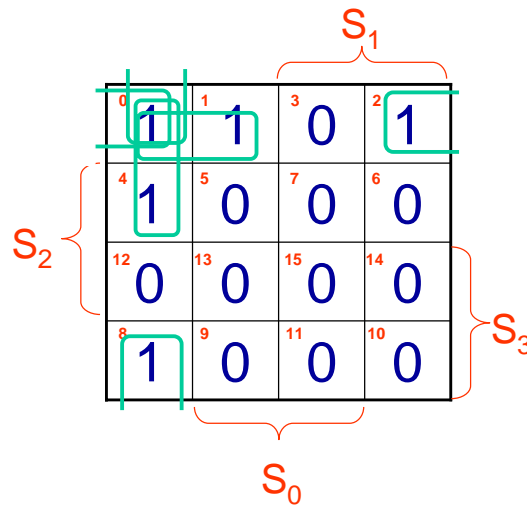
Minimize Combinatorial logic

- Minimize

- $Z = \lambda(S(t))$

- Example

- $Z_3 = \lambda(S_3 S_2 S_1 S_0) =$



Implementation of Combinatorial Logic & State

- Observations

- LUTs contain logic for

- $\lambda_3(S_3S_2S_1S_0)$:
- $\lambda_2(S_3S_2S_1S_0)$
- $\lambda_1(S_3S_2S_1S_0)$
- $\lambda_0(S_3S_2S_1S_0)$

- Further, Note

- $Z_1 = \lambda(S_3S_2S_1S_0) = 0$
 - Count is always even

- No logic required for $\delta(X, S(t))$

- Use Carry chain

