15-Analysis of Clocked Sequential Circuits

Text: Unit 13

ECEGR/ISSC 201
Digital Operations and Computations
Winter 2011

Overview

- Parity Checker
- Analysis by Signal Tracing
- State Tables and Graphs

Sequential Circuits

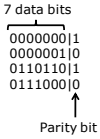
- Last lecture we discussed sequential counters
  - Sequence only dependent on previous state
- Next, we discuss sequential circuits with additional inputs
- We start with an example known as a parity checker

Sequential Parity Checker

- Error detection is important in data storage and transmittal
- Typically, an additional bit is appended to the data to assist in error detection
  - Called a "parity bit"

Sequential Parity Checker

- Example for an 8-bit word
  - Use 7 bits for data
  - One bit for parity checking
- Examples

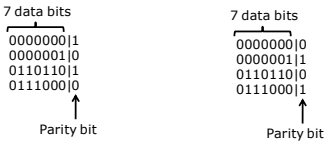


Sequential Parity Checker

- Value of parity bit is based on the number of 1s in the data
  - Odd parity: the number of 1s in the data bits is an odd number
  - Even parity: the number of 1s in the data bits is an even number

Sequential Parity Checker

- The state of the parity bit (1 or 0) depends on our convention
- Consistency is important

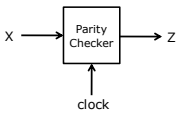


Sequential Parity Checker

- If any bit changes then the parity bit does not match the data and an error can be detected
- Using odd parity: 0000000|1
  - Error if: 0010000|1
- For odd parity: the parity bit can be thought of as adding either a 1 or 0 to make the sum always odd
- If the sum of the bits in the word are even, an error has occurred

Sequential Parity Checker

- Consider serial data X input into the parity checker
- Z = 1 if an error has occurred
- Let the checker use EVEN parity
  - Z = 1 if the total number of 1s received in an 8 bit word (7 data bits, 1 parity bit) is odd
  - Z = 0 if the total number of 1s received is even



Sequential Parity Checker

- Let 7-bit words be used
- Consider: 1 0 0 1 1 0 1
- What should the parity bit be? Remember we are using even parity
  - 1 + 0 + 0 + 1 + 1 + 0 +1 = 4
  - 4 is even, so the 8th bit is a 0

Sequential Parity Checker

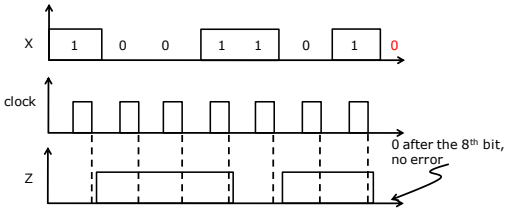
- The serial input X is then:1 0 0 1 1 0 1 0
- Do not confuse the output of Z with the parity bit
  - In the above case, the parity bit should be 0 and the output Z is 0 if no error occurs
  - If X: 1 0 0 1 1 0 0 1 (the parity bit is the final 1); the output of Z would still be 0 if no error occurs

Sequential Parity Checker

- Z may change state after each bit is received
- Z is checked after the 8th bit (parity bit) is received

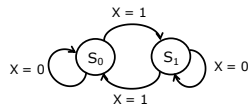
Sequential Parity Checker

- Falling edge
- X serial input: 1 0 0 1 1 0 1 0



Sequential Parity Checker

- State graph
  - S_0: Z = 0
  - S_1: Z = 1



Sequential Parity Checker

- Next-state table
- Two states, so only 1 flip-flop is needed

Next State

Present State	X = 0	X = 1	Present Output
S_0	S_0	S_1	0
S_1	S_1	S_0	1

Sequential Parity Checker

- Complete the table for a T flip-flop

Next State

Present State	X = 0	X = 1	Present Output
S_0	S_0	S_1	0
S_1	S_1	S_0	1

	Q^+		T		
Q	X = 0	X = 1	X = 0	X = 1	Z
0	0	1			0
1	1	0			1

Sequential Parity Checker

- Complete the table for a T flip-flop
  - T = X

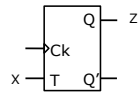
Next State

Present State	X = 0	X = 1	Present Output
S_0	S_0	S_1	0
S_1	S_1	S_0	1

	Q^+		T		
Q	X = 0	X = 1	X = 0	X = 1	Z
0	0	1	0	1	0
1	1	0	0	1	1

Sequential Parity Checker

- Resulting circuit



Analysis by Signal Tracing

- Basic procedure
  1. Assume initial state of all flip-flops (usually 0, unless otherwise noted)
  2. For the initial input data sequence, determine the circuit outputs and flip-flop inputs
  3. Determine the set of new flip-flop states after the next active clock edge
  4. Determine the new outputs
  5. Repeat 2-5 until the entire input sequence has been examined

Analysis by Signal Tracing

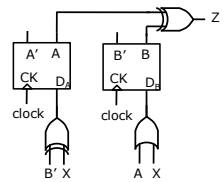
- Assumptions
  - Flip-flops change state shortly after the active edge of the clock
  - Inputs to flip-flops have been stable for a sufficient time before and after the active edge
    - Set-up and hold times are satisfied

Analysis by Signal Tracing

- Types of sequential circuits
  - Moore machine: output(s) is a function of the states only
  - Mealy machine: output(s) is a function of the states and an input

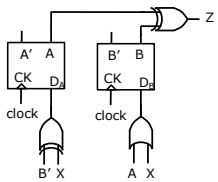
Analysis by Signal Tracing

- Is this a Moore or Mealy machine?
  - Note feedback connections are not shown



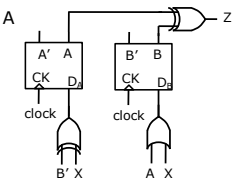
Analysis by Signal Tracing

- Is this a Moore or Mealy machine?
  - Moore machine
  - Z is a function only of the present state

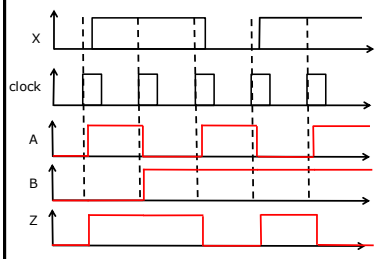


Analysis by Signal Tracing

- Assume the input sequence is X: 0 1 1 0 1
- Initial state is assumed to be A = B =0
  - D_A = B' XOR X
  - D_B = A + X
  - Z = B XOR A



Analysis by Signal Tracing



D_A = B' XOR X
D_B = A + X
Z = B XOR A

Remember, Z does not require a clock signal to change

Analysis by Signal Tracing

- Check the sequence
  - X = 0 1 1 0 1
  - A = 0 ← D_A = 1 XOR 0 = 1
  - B = 0 ← D_B = 0 + 0 = 0
  - Z = 0 ← Z = 0 XOR 1 = 1

Using the previous values for A, B, X

Using the above values for A, B

D_A = B' XOR X
D_B = A + X
Z = B XOR A

Analysis by Signal Tracing

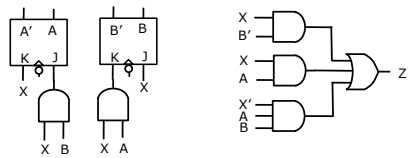
- Summary
  - X = 0 1 1 0 1
  - A = 0 1 0 1 0 1
  - B = 0 0 1 1 1 1
  - Z = 0 1 1 0 1 0

First 0 is the initial state, so ignore it

D_A = B' XOR X
D_B = A + X
Z = B XOR A

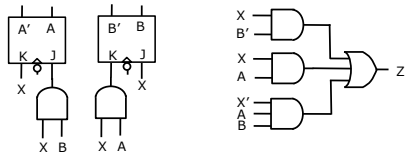
Analysis by Signal Tracing

- Is this a Moore or Mealy machine?



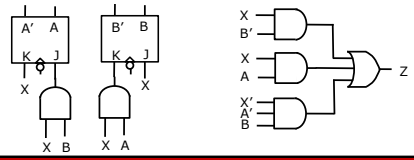
Analysis by Signal Tracing

- Is this a Moore or Mealy machine?
  - Mealy (Z has X fed into it)



Analysis by Signal Tracing

- Z = XB' + XA + X'A'B
- J_A = XB
- K_A = X
- J_B = X
- K_B = XA

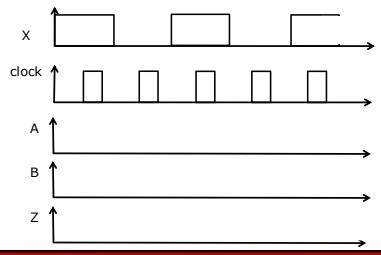


Analysis by Signal Tracing

- Output can change in response to the input as the input changes, not only after the edge of a clock

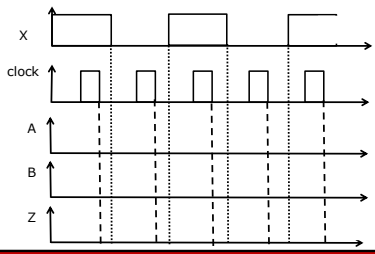
Analysis by Signal Tracing

- X = 1 0 1 0 1



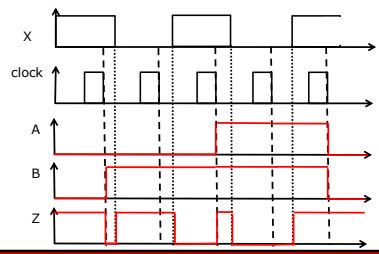
Analysis by Signal Tracing

- Need to track the active edge and any change in X



Analysis by Signal Tracing

- X = 1 0 1 0 1



$J_A = XB$
$K_A = X$

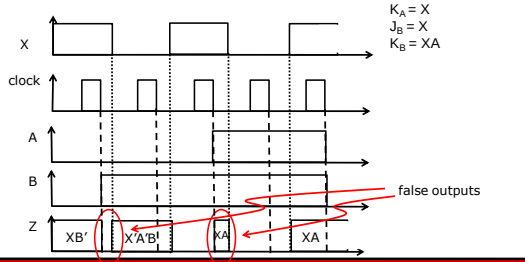
$J_B = X$
$K_B = XA$

$Z = XB' + XA + X'A'B$

Analysis by Signal Tracing

- X = 1 0 1 0 1

$Z = XB' + XA + X'A'B$
$J_A = XB$
$K_A = X$
$J_B = X$
$K_B = XA$



Analysis by Signal Tracing

- False outputs are possible for Mealy machines
  - Output temporarily changes prior to the active edge as a result of a change in input (X)
- Glitch: 0 to 1 prior to active edge
- Spike: 1 to 0 prior to active edge
- Solution: read the output just prior to the active edge

Analysis by Signal Tracing

- Summary
  - X = 1 0 1 0 1
  - A = 0 0 0 1 1 0
  - B = 0 1 1 1 1 0
  - Z = 1 1 0 0 1

State Tables and Graphs

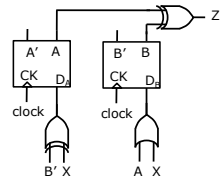
1. Determine the flip-flop input equations and the output equations from the circuit
2. Derive the next-state equation for each flip-flop
3. Plot the next-state k-map
4. Combine the maps to form the state table

State Tables and Graphs

- Next state equations
  - D flip-flop
    - $Q^+ = D$
  - D-CE flip-flop
    - $Q^+ = D(CE) + Q(CE)'$
  - T flip-flop
    - $Q^+ = T \oplus Q$
  - S-R flip-flop
    - $Q^+ = S + R'Q$
  - J-K flip-flop
    - $Q^+ = JQ' + K'Q$

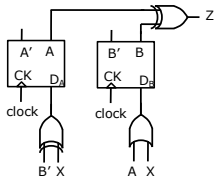
State Tables and Graphs

- Example circuit
- Step 1: Determine the flip-flop input equations and the output equations from the circuit



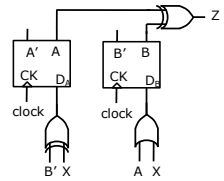
State Tables and Graphs

- Step 1: Determine the flip-flop input equations and the output equations from the circuit
  - $D_A = B'$ XOR X
  - $D_B = A + X$
  - Z = B XOR A



State Tables and Graphs

- Step 2: Derive the next-state equation for each flip-flop
  - $A^+ = X$ XOR B'
  - $B^+ = X + A$



State Tables and Graphs

- Step 3: Create the next-state k-map
  - $A^+ = X \text{ XOR } B'$
  - $B^+ = X + A$

AB \ X	0	1
00	1	0
01	0	1
11	0	1
10	1	0

A^+

AB \ X	0	1
00	0	1
01	0	1
11	1	1
10	1	1

B^+

State Tables and Graphs

- Step 4: Combine the maps to form the state table Z = B XOR A

A	B	X = 0 A+	X = 0 B+	X = 1 A+	X = 1 B+	Z
0	0	1	0	0	1	0
0	1	0	0	1	1	1
1	1	0	1	1	1	0
1	0	1	1	0	1	1

State Tables and Graphs

- For convenience, assign states (AB)
  - $S_0 = 00$
  - $S_1 = 01$
  - $S_2 = 11$
  - $S_3 = 10$

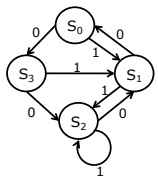
State Tables and Graphs

Present State	Next State X = 1	Next State X = 0	Z
S_0	S_3	S_1	0
S_1	S_0	S_2	1
S_2	S_1	S_2	0
S_3	S_2	S_1	1

State Tables and Graphs

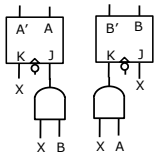
- Construct the state graph
- The numbers beside the path refer to the value of X

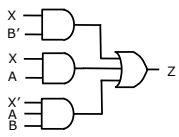
Present State	Next State X = 0	Next State X = 1	Z
S_0	S_3	S_1	0
S_1	S_0	S_2	1
S_2	S_1	S_2	0
S_3	S_2	S_1	1



State Tables and Graphs

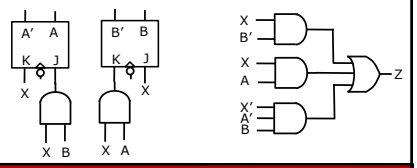
- Now consider the Mealy machine





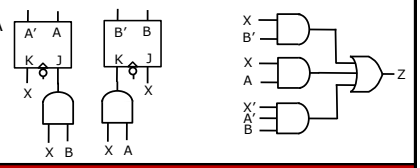
State Tables and Graphs

- Step 1: Determine the flip-flop input equations and the output equations from the circuit



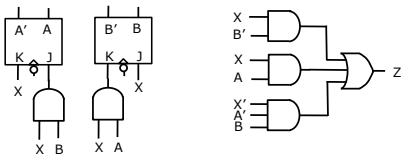
State Tables and Graphs

- Step 1: Determine the flip-flop input equations and the output equations from the circuit
  - $Z = XB' + XA + X'A'B$
  - $J_A = XB$
  - $K_A = X$
  - $J_B = X$
  - $K_B = XA$



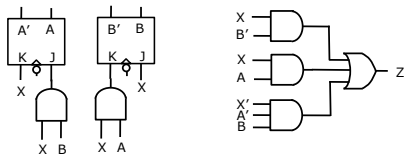
State Tables and Graphs

- Step 2: Derive the next-state equation for each flip-flop: $Q^+ = JQ' + K'Q$



State Tables and Graphs

- Step 2: Derive the next-state equation for each flip-flop
  - $A^+ = J_A A' + K'_A A = XBA' + X'A$
  - $B^+ = J_B B' + K'_B B = XB' + (AX)'B = XB' + X'B + A'B$



State Tables and Graphs

- Step 3: Complete the next-state k-map

AB \ X	0	1
00		
01		
11		
10		

A^+

AB \ X	0	1
00		
01		
11		
10		

B^+

AB \ X	0	1
00		
01		
11		
10		

Z

State Tables and Graphs

- Step 3: Complete the next-state k-map

$A^+ = XBA' + X'A$

AB \ X	0	1
00	0	0
01	0	1
11	1	0
10	1	0

A^+

$B^+ = XB' + X'B + A'B$

AB \ X	0	1
00	0	1
01	1	1
11	1	0
10	0	1

B^+

$Z = XB' + XA + X'A'B$

AB \ X	0	1
00	0	1
01	1	0
11	0	1
10	0	1

Z

State Tables and Graphs

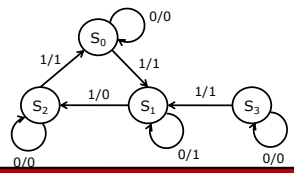
A	B	X = 0 A+	X = 0 B+	X = 1 A+	X = 1 B+	X =0 Z	X =1 Z
0	0	0	0	0	1	0	1
0	1	0	1	1	1	1	0
1	1	1	1	0	0	0	1
1	0	1	0	0	1	0	1

State Tables and Graphs

Present State	Next State X = 0	Next State X = 1	X =0 Z	X =1 Z
S_0	S_0	S_1	0	1
S_1	S_1	S_2	1	0
S_2	S_2	S_0	0	1
S_3	S_3	S_1	0	1

State Tables and Graphs

- F/G
  - F: input value
  - G: output value



State Tables and Graphs

- Read text page 371 through 377 for details on serial adders and a general model