Logic Design

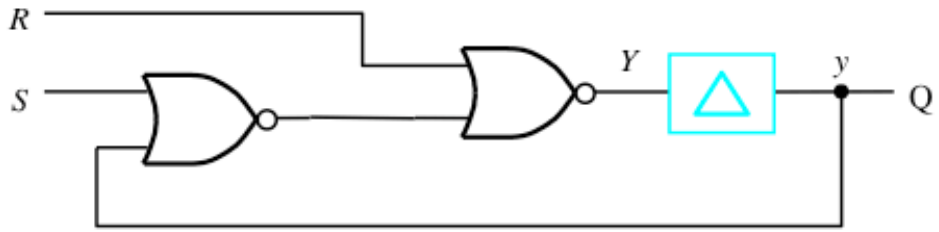
Asynchronous Sequential circuits

Asynchronous Sequential Circuits

- Synchronous sequential circuits: operation is controlled by a clock pulse (operates in pulse mode)
- Sequential circuits that do not operate in pulse mode and do not use flip-flops
- In asynchronous sequential circuits changes in state are not triggered by clock pulses but by change in the inputs
- Conditions:
  - Inputs must change one at a time
  - Sufficient time exists between changes in input signals so all internal signals stop changing
- Above conditions true: circuit is operating in the fundamental mode

McMaster University

Asynchronous behavior



(b) Circuit with modeled gate delay

Present state y	Next state			
SR =	00	01	10	11
	Y	Y	Y	Y
0	(0)	(0)	1	(0)
1	(1)	0	(1)	0

(b) State-assigned table

Figure 9.1. Analysis of the SR latch.

McMaster University

Asynchronous behavior

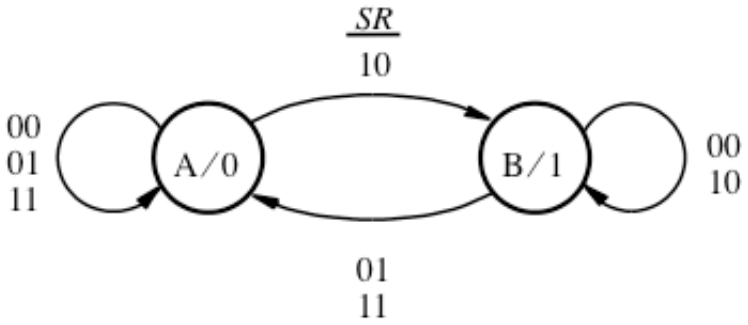
- Because of the feedback the circuit is sequential
- Box labeled Δ represents combined propagation delays through two NOR gates
- y is the current state
- Y is the next state
- After a Δ time delay y takes the value of Y
- If for a particular input y is not equal to Y, the circuit is not stable
- Stable states are indicated by circles
- We can derive state table and state diagram

McMaster University

Asynchronous behavior

Present state	Next state				Output Q
	SR = 00	01	10	11	
A	(A)	(A)	B	(A)	0
B	(B)	A	(B)	A	1

(a) State table



(b) State diagram

Figure 9.2. FSM model for the SR latch.

Asynchronous behavior

- Consider the opposite task: synthesis of an asynchronous circuit
- Using the state assigned table to find minimal SOP yields:
- Y=R'.(S+y)
- If we were doing synchronous circuits we would use a D flip-flop
- Here, we create a circuit that realizes the expression and we feed back the output signal as the present state input y

McMaster University

Asynchronous behavior

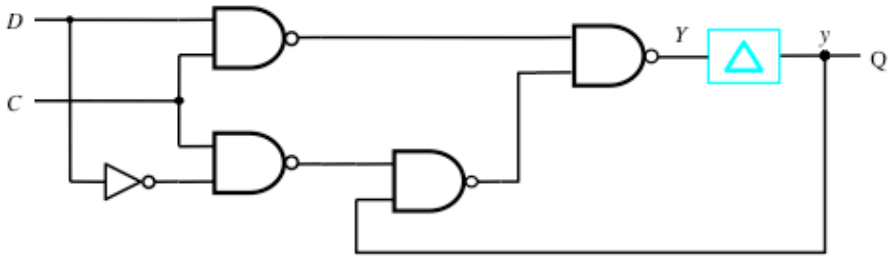
- In asynchronous circuits:
  - State table => flow table
  - State-assigned table => transition table or excitation table

McMaster University

Gated D latch

- We consider the clock signal to be one of the input signals
- Y=CD+C'y+Dy
- Using the consensus theorem: Y=CD+C'y

McMaster University



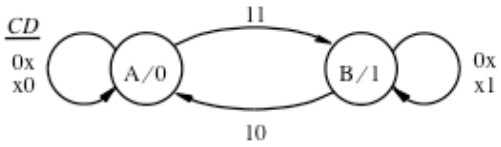
(a) Circuit

Present state y	Next state CD = 00	01	10	11	Q
	Y	Y	Y	Y	
0	(0)	(0)	(0)	1	0
1	(1)	(1)	0	(1)	1

(b) Excitation table

Present state	Next state CD = 00	01	10	11	Q
A	(A)	(A)	(A)	B	0
B	(B)	(B)	A	(B)	1

(c) Flow table



(d) State diagram

Figure 9.4. The gated D latch.

Master Slave D Flip-flop

- Ym=CD+C'ym
- Ys=C'ym+Cys

McMaster University

Master Slave D Flip-flop

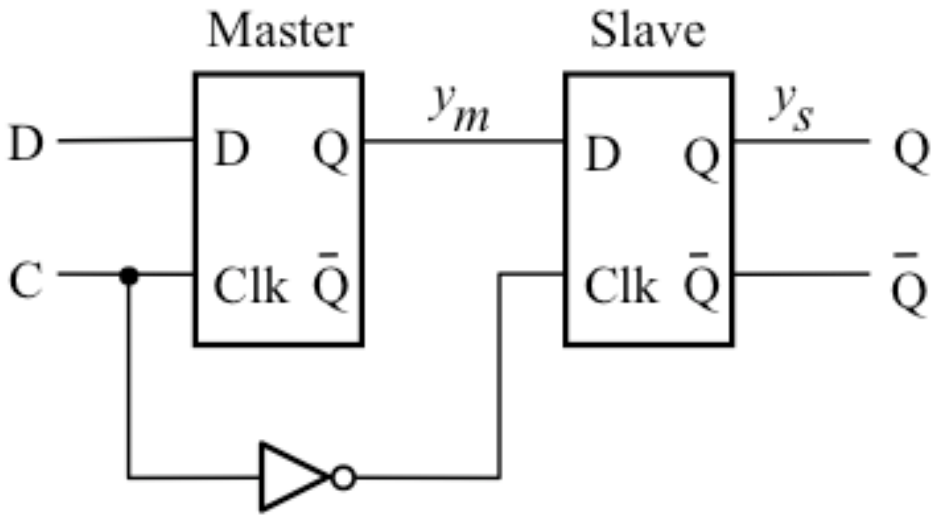


Figure 9.5. Circuit for the master-slave D flip-flop.

McMaster University

Present state $y_m\ y_s$	Next state CD = 00	01	10	11	Output Q
	$Y_m\ Y_s$				
00	(00)	(00)	(00)	10	0
01	00	00	(01)	11	1
10	11	11	00	(10)	0
11	(11)	(11)	01	(11)	1

(a) Excitation table

Present state	Next state CD = 00	01	10	11	Output Q
S1	(S1)	(S1)	(S1)	S3	0
S2	S1	S1	(S2)	S4	1
S3	S4	S4	S1	(S3)	0
S4	(S4)	(S4)	S2	(S4)	1

(b) Flow table

Present state	Next state CD = 00	01	10	11	Output Q
S1	(S1)	(S1)	(S1)	S3	0
S2	S1	–	(S2)	S4	1
S3	–	S4	S1	(S3)	0
S4	(S4)	(S4)	S2	(S4)	1

(c) Flow Table with unspecified entries

Figure 9.6. Excitation and flow tables for Example 9.2.

Master Slave D Flip-flop

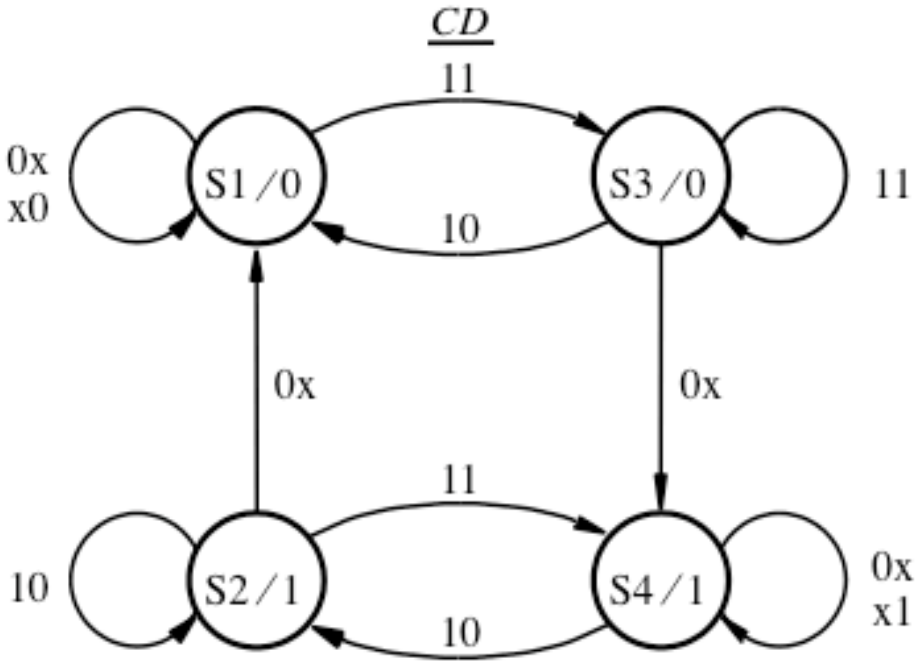


Figure 9.7. State diagram for the master-slave D flip-flop.

McMaster University

Master Slave D Flip-flop

- Since we assumed that inputs must change one at a time, if the circuit is stable in state S2 for which CD=10 it cannot go to S1 under the input valuation CD=01 because simultaneous change in both inputs cannot happen
- These changes are labeled unspecified in the table.

An other example

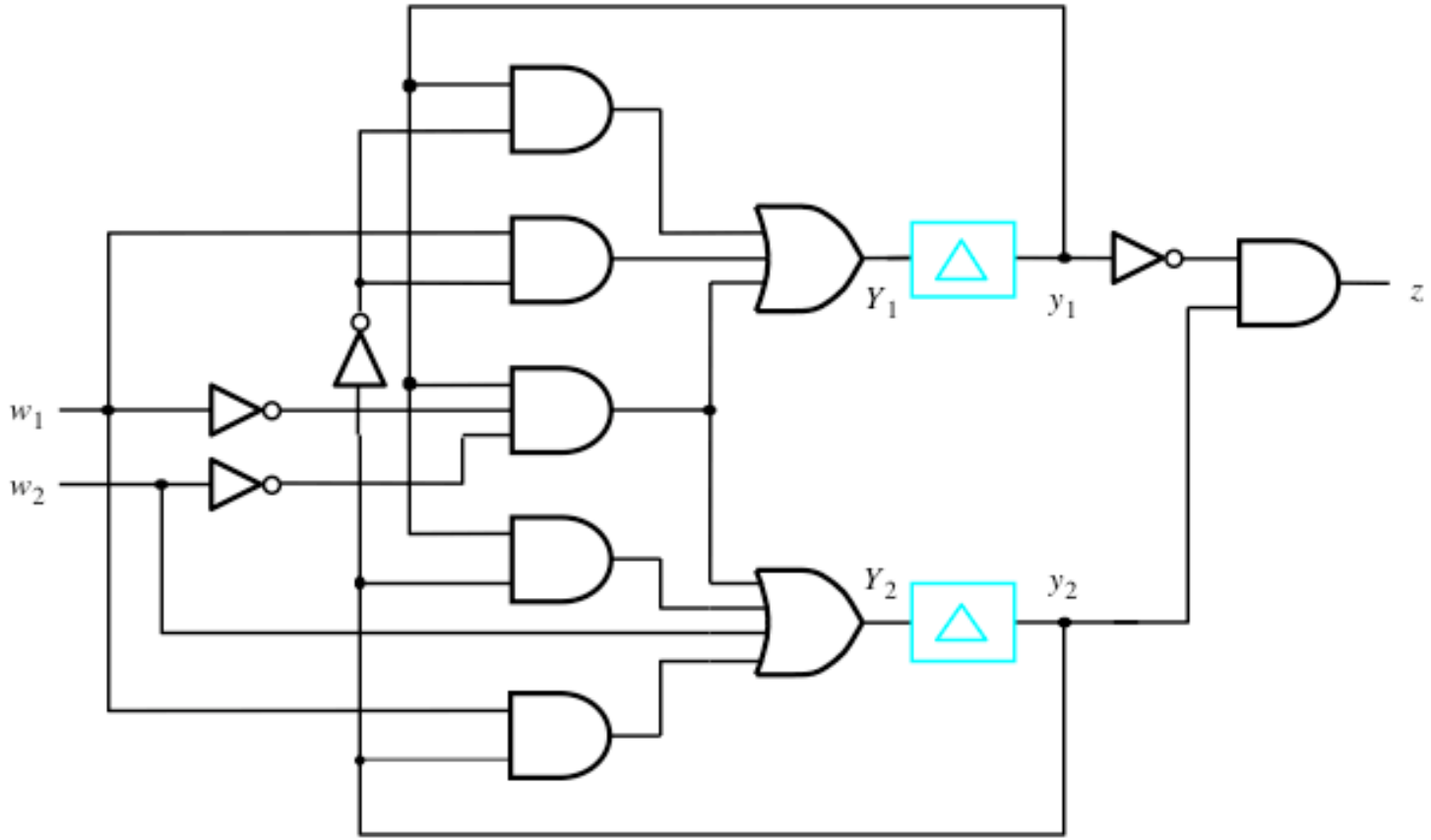


Figure 9.8. Circuit for Example 9.3.

- Y1=y1y2’+w1y2’+w1’w2’y1
- Y2=y1y2+w1y2+w2+w1’w2’y1
- Z=y1’y2

McMaster University

Present state y_2y_1	Next state w_2w_1 = 00	01	10	11	Output z
	Y_2Y_1	Y_2Y_1	Y_2Y_1	Y_2Y_1	
00	(00)	01	10	11	0
01	11	(01)	11	11	0
10	00	(10)	(10)	(10)	1
11	(11)	10	10	10	0

(a) Excitation table

Present state	Next state w_2w_1 = 00	01	10	11	Output z
A	(A)	B	C	D	0
B	D	(B)	D	D	0
C	A	(C)	(C)	(C)	1
D	(D)	C	C	C	0

(b) Flow table

Figure 9.9. Excitation and flow tables for the circuit in Figure 9.8.

Present state	Next state				Output z
	$w_2w_1 =$ 00	01	10	11	
A	(A)	B	C	—	0
B	D	(B)	—	D	0
C	A	(C)	(C)	(C)	1
D	(D)	C	C	C	0

Figure 9.10. Modified flow table for Example 9.3.

McMaster University

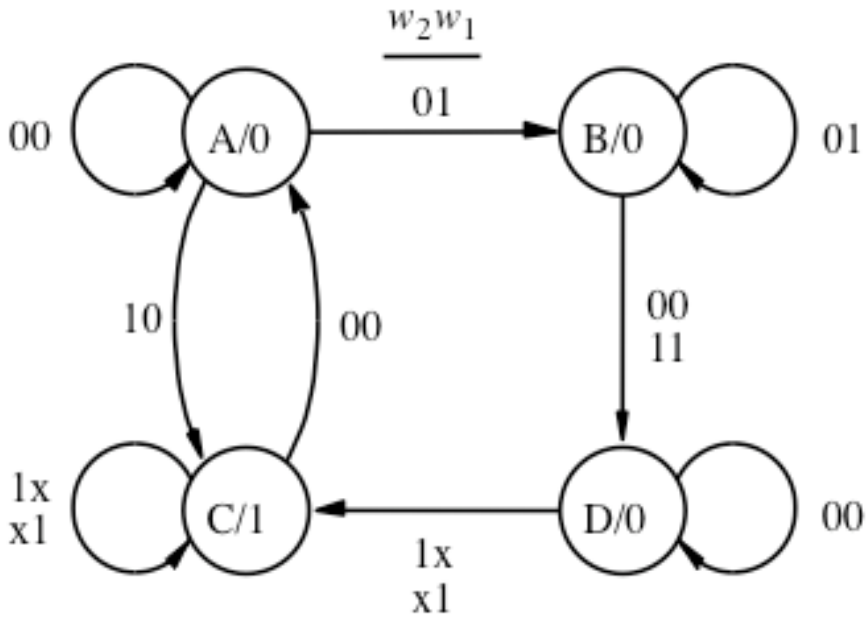


Figure 9.11. State table for Example 9.3.

McMaster University

Analysis Process

- Each feedback path is cut and a delay element is inserted at the point where the cut is made.
  - The input to the delay is a next state variable Yi and the output is present state variable yi
  - Cut can be make anywhere in the loop
  - The number of cuts is the smallest number that results in being no feedback anywhere in the circuit (cut set)
- Next state and output expressions are derived from the circuit
- Excitation table is derived
- Flow table is obtained
- State diagram is derived

Synthesis

1. Devise a state diagram for an FSM that realizes the required functional behavior
2. Derive the flow table and reduce the number of states if possible
3. Perform the state assignment and derive the excitation table
4. Obtain the next state and output expressions
5. Construct a circuit that implements these expressions

- Note: When devising a state diagram, it is essential to ensure when the circuit is in stable state correct output signals are generated. Should it be necessary to pass through an unstable state, this state must not produce an undesirable output

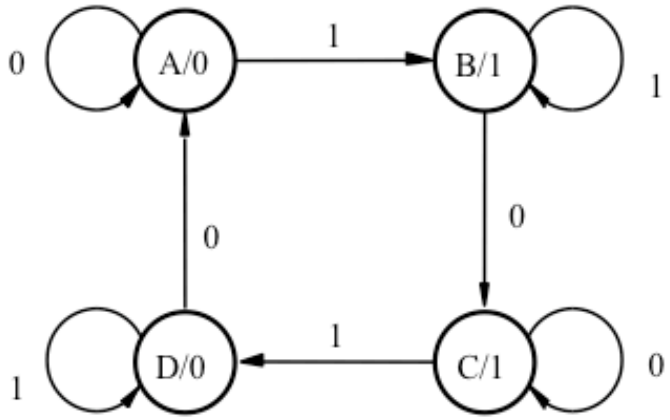
Serial Parity Generator

- We want to design a circuit with input w and output z such that z is equal to 0 if the number of previously applied pulses to w is even and z is equal to 1 if the number of pulses is odd.

Serial Parity Generator

- State A: an even number of pulses have arrived and the current value of the input is 0
- State B: an odd number of pulses have arrived and the current value of the input is 1
- State C: an odd number of pulses have arrived and the current value of the input is 0
- State D: an even number of pulses have arrived and the current value of the input is 1

Serial Parity Generator



(a) State diagram

Present State	Next state $w = 0$	Next state $w = 1$	Output z
A	(A)	B	0
B	C	(B)	1
C	(C)	D	1
D	A	(D)	0

(b) Flow table

Figure 9.13. Parity-generating asynchronous FSM.

Serial Parity Generator

- States B and C cannot be combined because it is impossible to have B both stable under w=1 and change to D for w=1

McMaster University

Serial Parity Generator

Present state y_2y_1	Next state Y_2Y_1 $w = 0$	Next state Y_2Y_1 $w = 1$	Output z
00	(00)	01	0
01	10	(01)	1
10	(10)	11	1
11	00	(11)	0

(a) Poor state assignment

Present state y_2y_1	Next state Y_2Y_1 $w = 0$	Next state Y_2Y_1 $w = 1$	Output z
00	(00)	01	0
01	11	(01)	1
11	(11)	10	1
10	00	(10)	0

(b) Good state assignment

Figure 9.14. State assignment for Figure 9.13*b*.

Serial Parity Generator

- Problem with the first state assignment:
- Assume circuit is stable in state 11 with w=1
- Input changes to w=0
- State should change from 11 to 00.
- Both state variables must change their values
- In asynchronous circuits values of the next states are determined by logic gates with varying propagation delays.
- One variable will change slightly before the other

Race condition

- Scenario 1: y1 changes first
  - Circuit goes from 11 to 10.
  - As soon as it reaches 10 with w=0 it will stay there
- Scenario 2: y2 changes first
  - Circuit goes from 11 to 01
  - According to the table with w=0 it will try to change to 10
  - This again requires both y's to change.
  - If y1 changes first the circuits is going to be stable in 00 with w=0
- Uncertainty caused by multiple changes in the state variables in response to input: race condition

McMaster University

Serial Parity Generator

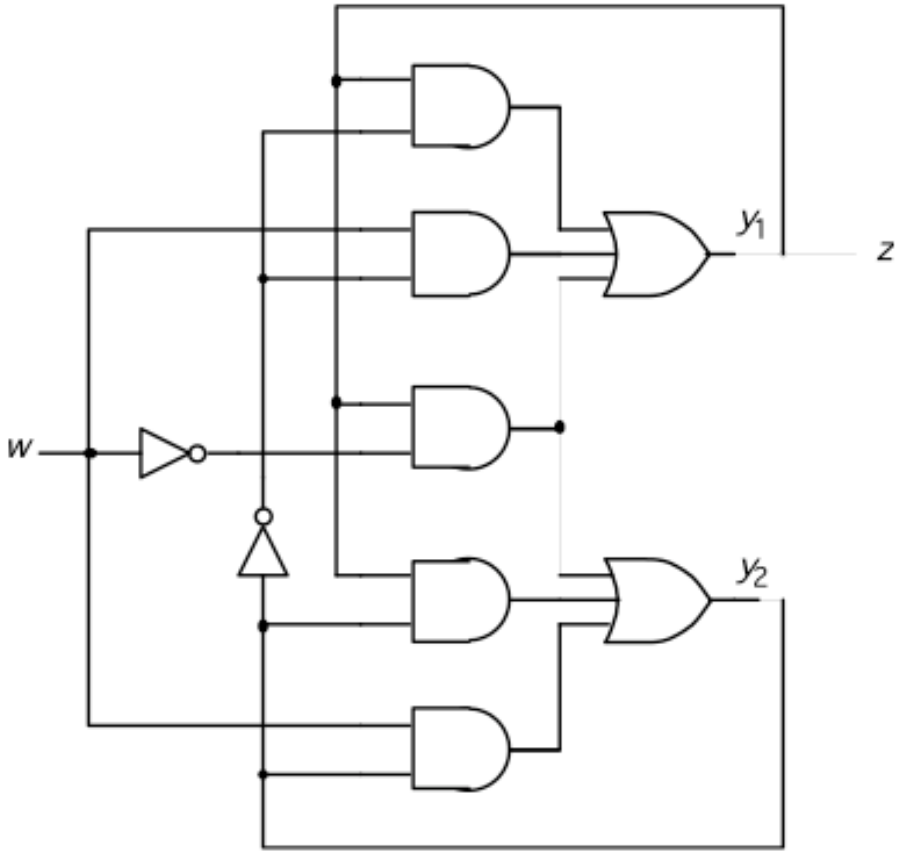


Figure 9.15. Circuit that implements the FSM in Figure 9.13*b*.

McMaster University

Modulo-4 counter

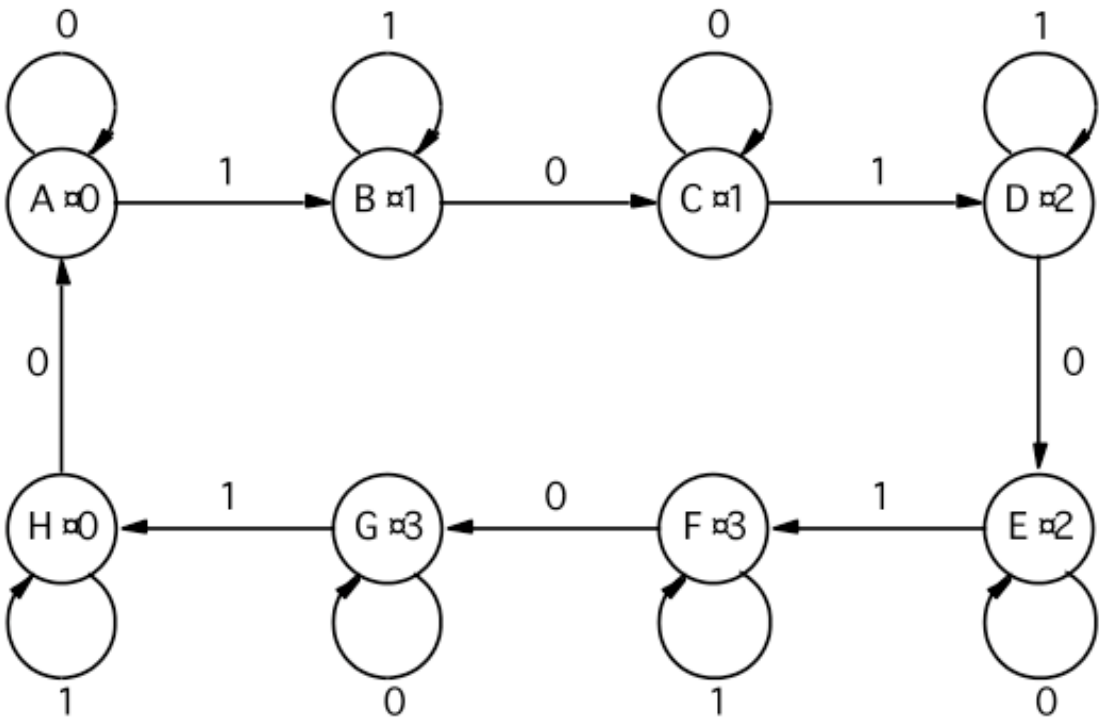


Figure 9.17. State diagram for a modulo-4 counter.

McMaster University

Present state	Next state $w = 0$	Next state $w = 1$	Output z
A	(A)	B	0
B	C	(B)	1
C	(C)	D	1
D	E	(D)	2
E	(E)	F	2
F	G	(F)	3
G	(G)	H	3
H	A	(H)	0

(a) Flow table

Present state $y_3y_2y_1$	Next state $w = 0$ $Y_3Y_2Y_1$	Next state $w = 1$ $Y_3Y_2Y_1$	Output z_2z_1
000	(000)	001	00
001	011	(001)	01
011	(011)	010	01
010	110	(010)	10
110	(110)	111	10
111	101	(111)	11
101	(101)	100	11
100	000	(100)	00

(b) Excitation table

Mod-8 output $z_3z_2z_1$
000
001
010
011
100
101
110
111

(c) Output for counting the edges

Figure 9.18. Flow and excitation tables for a modulo-4 counter.

Hazards

- In asynchronous sequential circuits it is important to avoid glitches
- Glitches caused by the structure of a given circuit and propagation delays are referred to as hazard
- Two types of hazards: static and dynamic

Hazards

- Static hazard: a signal is supposed to remain at a particular logic value when an input changes it value but instead the signal undergoes a momentary change
- Dynamic hazard: a signal is supposed to change from 0 to 1 (or 1 to 0) but the change involves a short oscillation

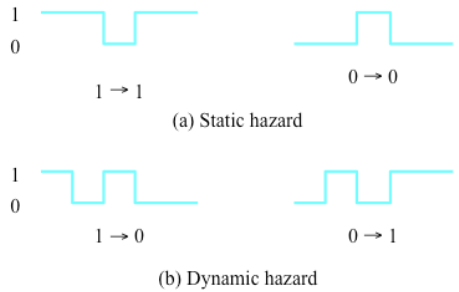


Figure 9.61. Definition of hazards.

McMaster University

Hazards

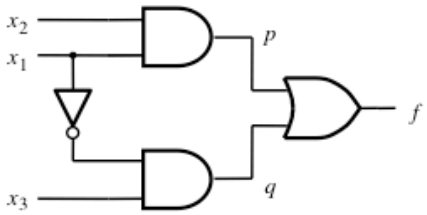
- Consider the circuit in the next slide
- Let's assume x1=x2=x3=1, f=1
- Suppose x1 changes to 0
- Point p will see the change before point q
- For a short time both p and q are zero dropping f to zero before recovering back to 1

$$f = x_1 x_2 + \bar{x}_1 x_3$$

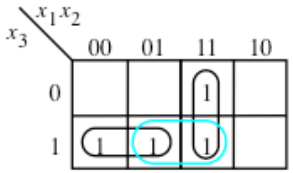
$$f = x_1 x_2 + \bar{x}_1 x_3 + x_2 x_3$$

McMaster University

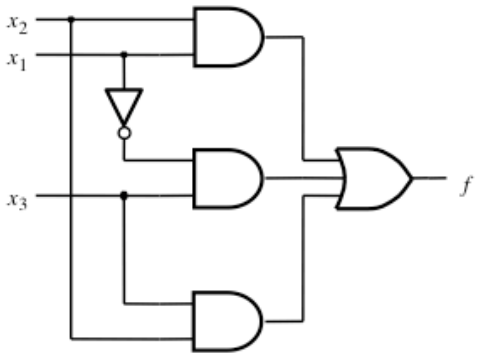
Hazards



(a) Circuit with a hazard



(b) Karnaugh map



(c) Hazard-free circuit

Figure 9.62. An example of a static hazard.

Master Slave D Flip Flop

Present state $y_m\ y_s$	Next state $Y_m\ Y_s$				Output Q
	CD = 00	01	10	11	
00	(00)	(00)	(00)	10	0
01	00	00	(01)	11	1
10	11	11	00	(10)	0
11	(11)	(11)	01	(11)	1

(a) Excitationtable

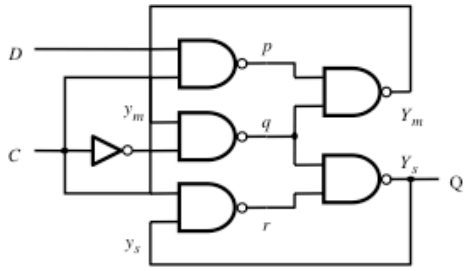
$$Y_m = CD + \bar{C}\, y_m$$

$$= (C \uparrow D) \uparrow (\bar{C} \uparrow y_m)$$

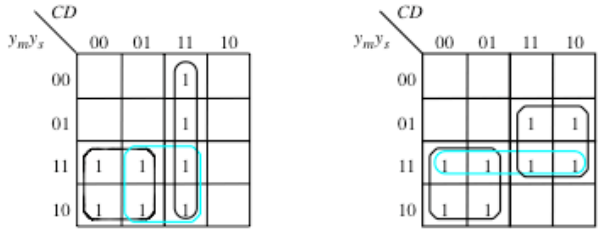
$$Y_s = Cy_s + \bar{C}\, y_m$$

$$= (C \uparrow y_s) \uparrow (\bar{C} \uparrow y_m)$$

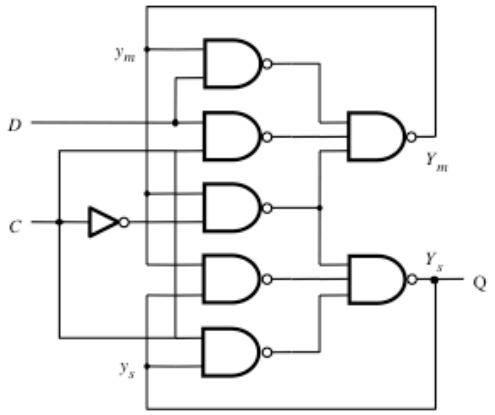
McMaster University



(a) Minimum-cost circuit



(b) Karnaugh maps for Y_m and Y_s in Figure 9.6a



(c) Hazard-free circuit

Figure 9.63. Two-level implementation of master-slave D flip-flop.

$$Y_m = CD + \bar{C}\, y_m + D y_m$$

$$= (C \uparrow D) \uparrow ((C \uparrow \bar{D}) \uparrow y_m)$$

$$Y_s = C y_s + \bar{C}\, y_m + y_m y_s$$

$$= ((\bar{C} \uparrow \bar{y}_m) \uparrow y_s) \uparrow (\bar{C} \uparrow y_m)$$

McMaster University

Dynamic Hazard

- A dynamic hazard is caused by the structure of the circuit where there exist multiple paths for a given signal change to propagate along
- Dynamic hazards are encountered in multilevel circuits obtained using factoring or decomposition techniques
- Dynamic hazards are neither easy to detect nor easy to deal with
- They can be avoided using two-level circuits and ensuring that there are not static hazards