CHAPTER 9: ASYNCHRONOUS SEQUENTIAL CIRCUITS

Chapter Objectives

- Sequential circuits that *are not synchronized by a clock* – Asynchronous circuits
- *Analysis* of Asynchronous circuits
- *Synthesis* of Asynchronous circuits
- *Hazards* that cause incorrect behavior of a circuit

Asynchronous sequential circuits

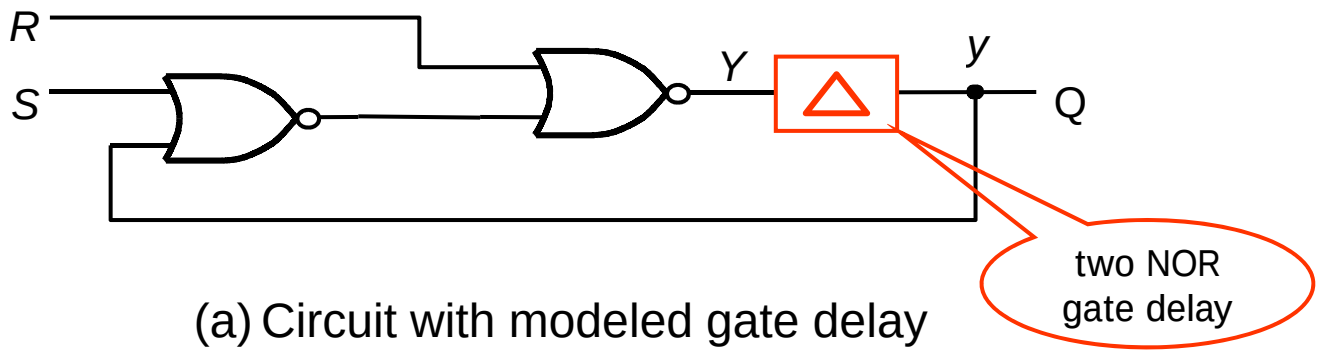
- Synchronous sequential circuits
  - state variables : F/Fs
  - controlled by a clock
  - operate in *pulse mode*

- Asynchronous sequential circuits
  - do not operate in *pulse mode*
  - do not use F/Fs to represent state variables
  - Changes in state are dependent on whether *each of inputs* to the circuit has the logic level 0 or 1 at any given time

- To achieve reliable operation (focus on the simplest case)
  - the inputs to the circuit must change one at a time
  - there must be sufficient time between the changes in input signals to allow the circuit to reach a stable state

  - A circuit that adheres to these constraints is said to operate *in the fundamental mode*

Asynchronous behavior



(a) Circuit with modeled gate delay

$$Y = \overline{\overline{(y + S)} + R}$$

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

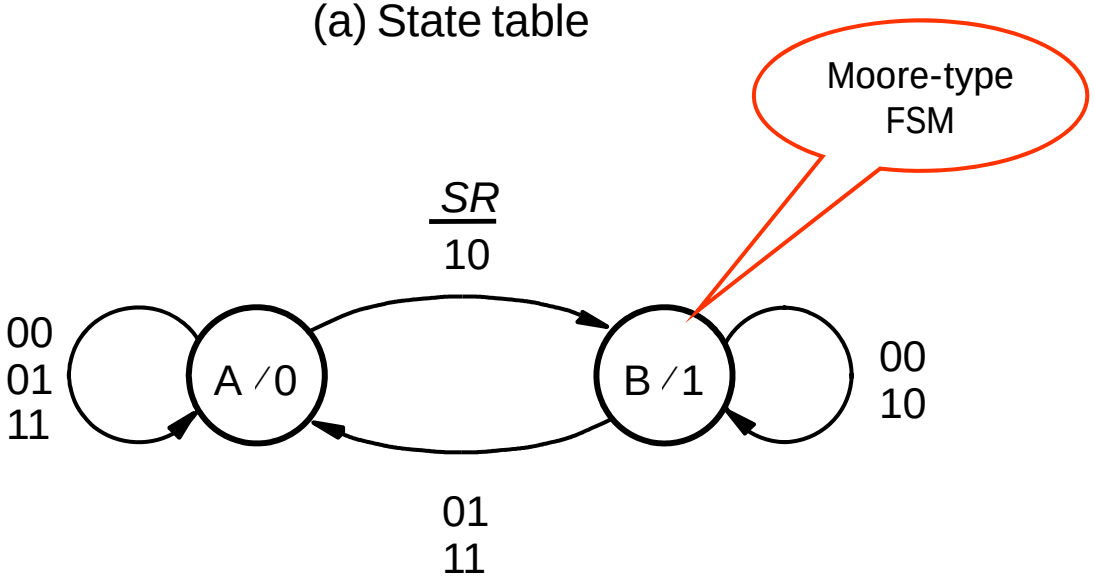
stable state

(b) State-assigned table

FSM model for the SR latch

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

Synthesis of an asynchronous circuit

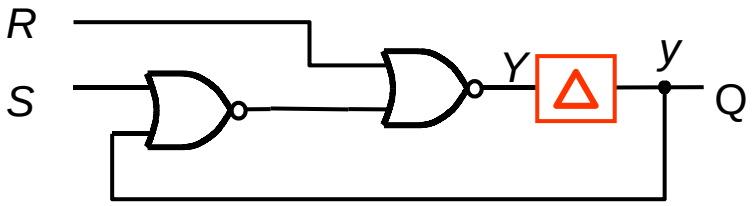
Present state	Nextstate SR = 00	01	10	11	Output Q
A	(A)	(A)	B	(A)	0
B	(B)	A	(B)	A	1

(a) State table

$$
\begin{aligned}
Y &= \overline{R} \bullet (S + y) \\
&= \overline{\overline{\overline{R} \bullet (S + y)}} \\
&= \overline{(\overline{\overline{R}} + \overline{(S + y)})} \\
&= \overline{(R + \overline{(S + y)})} \\
z &= y
\end{aligned}
$$

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

(b) State-assigned table

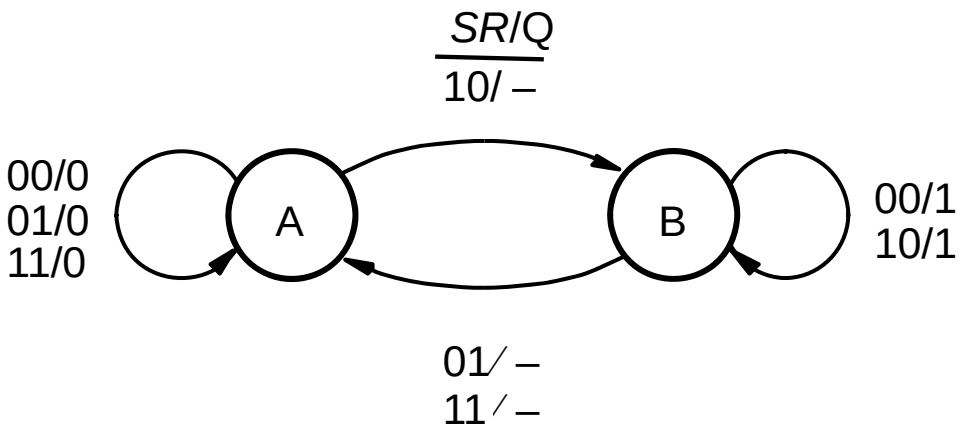


Mealy representation of the SR latch

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

there is little to be gained in trying to make output Q go to 1 a little sooner

(a) State table



(b) State diagram

$$Y = \overline{R} \bullet (S + y)$$
$$= \overline{\overline{\overline{R} \bullet (S + y)}}$$
$$= \overline{(\overline{\overline{R}} + \overline{(S + y)})}$$
$$= \overline{(R + \overline{(S + y)})}$$
$$Q = y$$

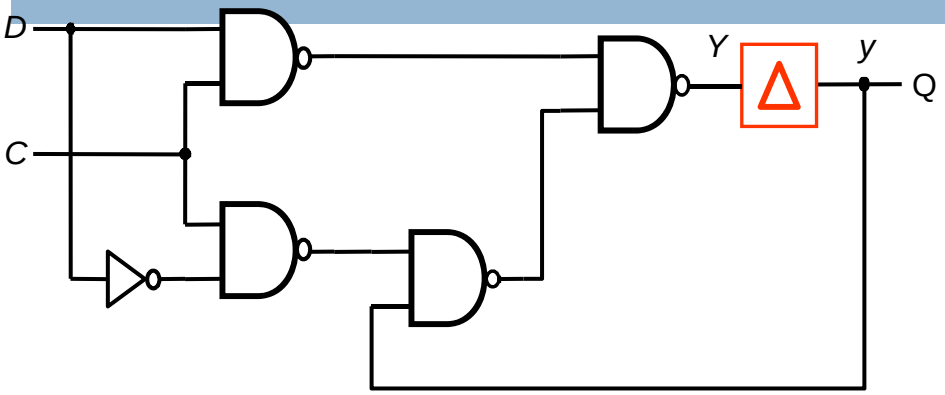
Terminology

- Asynchronous circuits
  - state table -> flow table
  - state-assigned table -> transition table or excitation table

- We will use the term *flow table and excitation table*

Analysis of Asynchronous Circuits

Analysis of Asynchronous circuits



(a)gated D latch

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

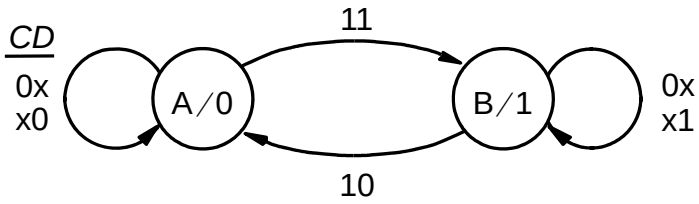
(b) Excitation table

$$Y = (\overline{\overline{C \bullet D}}) \bullet ((\overline{\overline{C \bullet \overline{D}}}) \bullet y)$$
$$= (\overline{\overline{C \bullet D}}) + ((\overline{\overline{C \bullet \overline{D}}}) \bullet y)$$
$$= CD + ((\overline{C} + D) \bullet y)$$
$$= CD + \overline{C}y + Dy$$
$$= CD + \overline{C}y$$

redundant term to solve a hazard

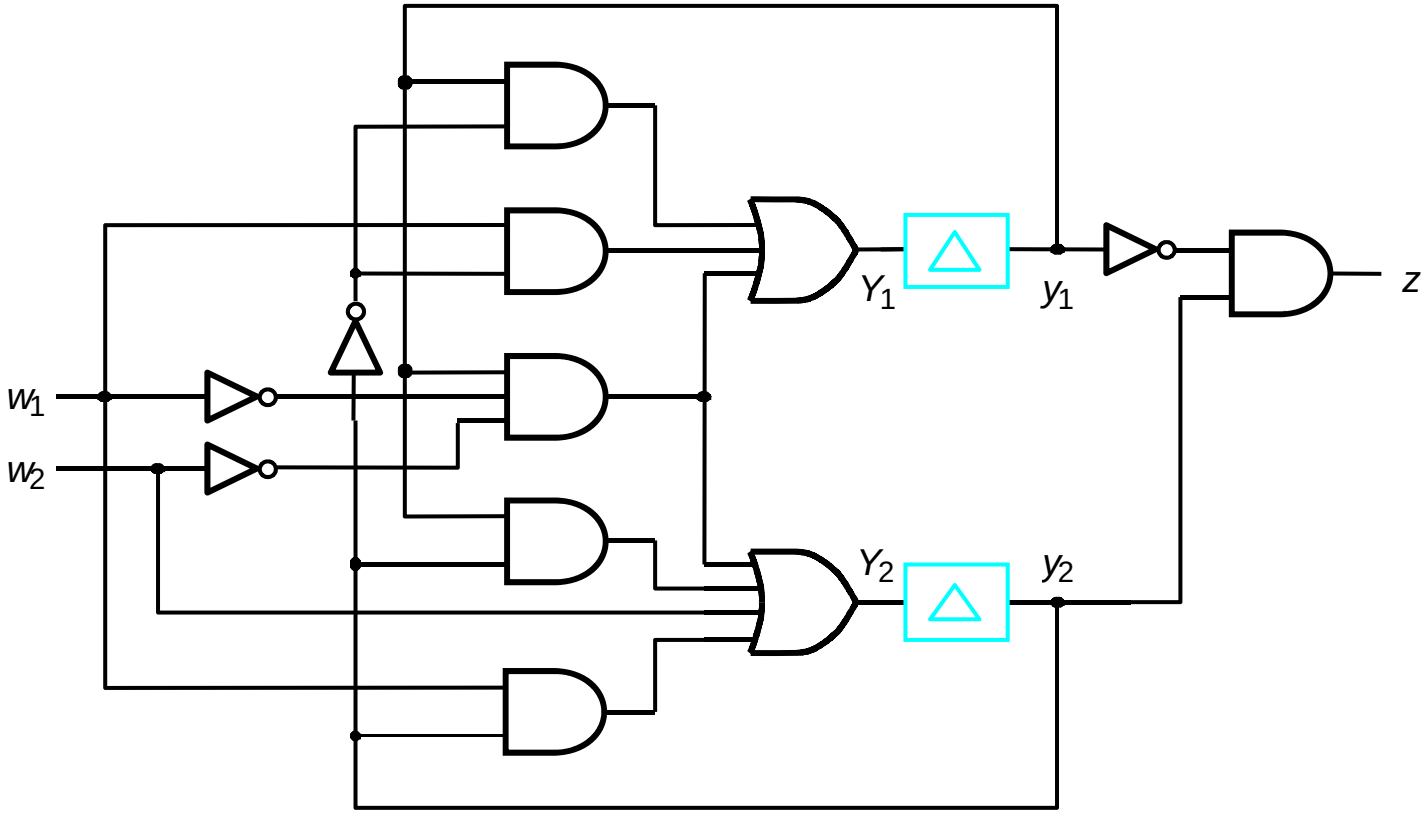
Present state	Next state CD = 00	01	10	11	Q
A	(A)	(A)	(A)	B	0
B	(B)	(B)	A	(B)	1

(c) Flow table



(d) State diagram

Analysis of the circuit in example 9.3



$$Y_1 = y_1\bar{y}_2 + w_1\bar{y}_2 + \bar{w}_1\bar{w}_2 y_1$$

$$Y_2 = y_1 y_2 + w_1 y_2 + w_2 + \bar{w}_1\bar{w}_2 y_1$$

$$z = \bar{y}_1 y_2$$

Excitation and flow tables for the circuit in example 9.3

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

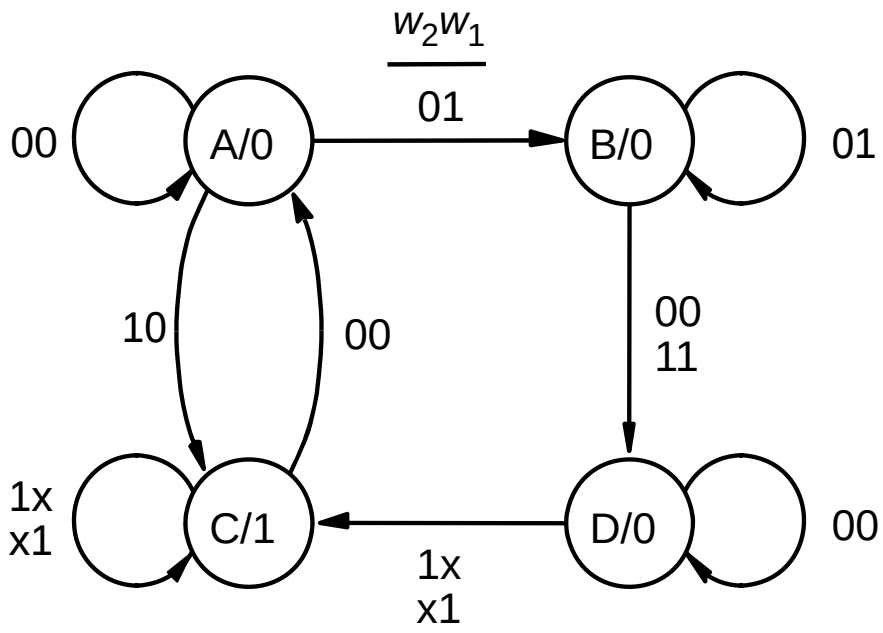
Present state	Next state w_2w_1 = 00	01	10	11	Output z
A	(A)	B	C	D	0
B	D	(B)	D	D	0
C	A	(C)	(C)	(C)	1
D	(D)	C	C	C	0

(b) Flow table

Modified flow table for Example 9.3.

Present state	Next state				Output
	$w_2 w_1$ = 00	01	10	11	z
A	(A)	B	C	–	0
B	D	(B)	–	D	0
C	A	(C)	(C)	(C)	1
D	(D)	C	C	C	0

State table for Example 9.3



Flow table for a simple vending machine

Present state	Next state				Output
	$w_2 w_1$ = 00	01	10	11	z
A	(A)	B	C	–	0
B	D	(B)	–	–	0
C	A	(C)	(C)	–	1
D	(D)	C	C	–	0

$w_2 \equiv$ dime $w_1 \equiv$ nickel

Steps in the Analysis Process

- Each feedback path is cut
  - A delay element is inserted at the point where the cut is made
  - A cut can be made anywhere in a particular loop formed by feedback connection, as long as there is only one cut per (state variable) loop
- Next-state and output expressions are derived from the circuit
- The excitation table is derived
- A flow table is obtained
- A corresponding state diagram is derived from the flow table if desired

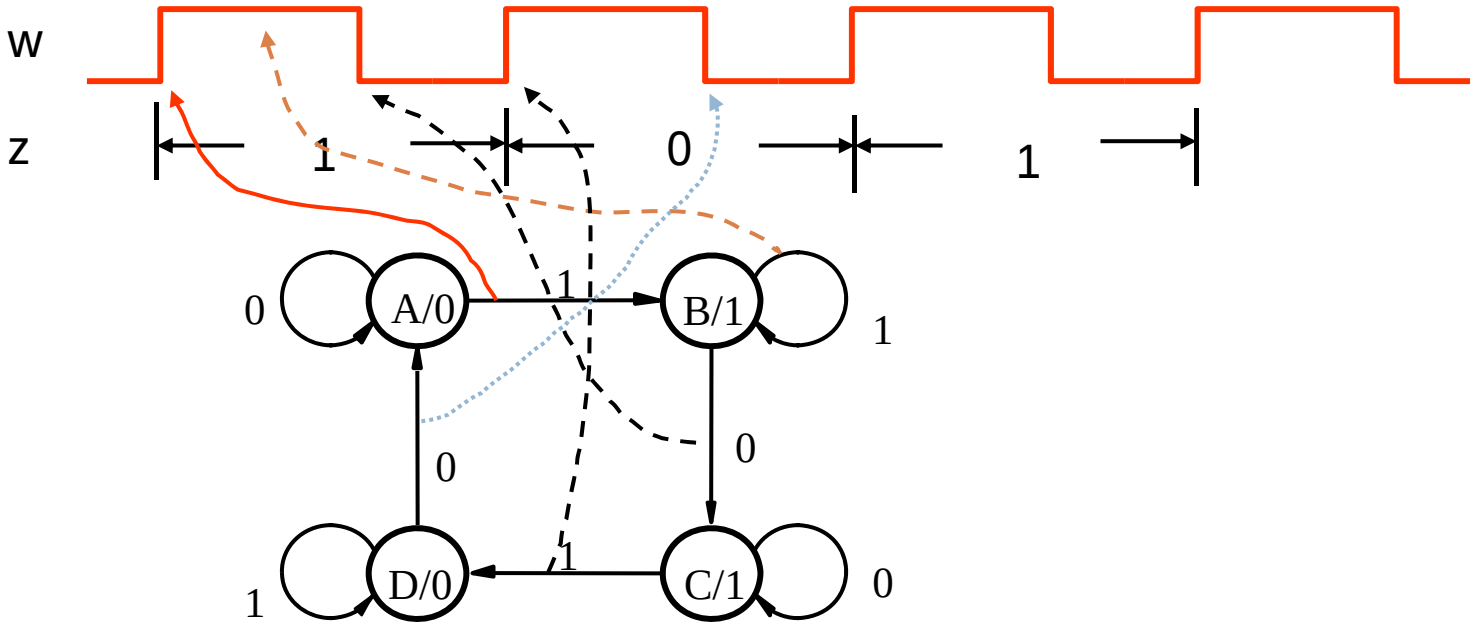
Synthesis of Asynchronous Circuits

Synthesis of Asynchronous Circuits

- the same basic steps used to synthesize the synchronous circuits
  - Devise a state diagram for an FSM
  - Derive *the flow table* and reduce the number of states if possible
  - Perform the state assignment and derive *the excitation table*
  - Obtain the next-state and output expressions
  - Construct a circuit that implements these expressions

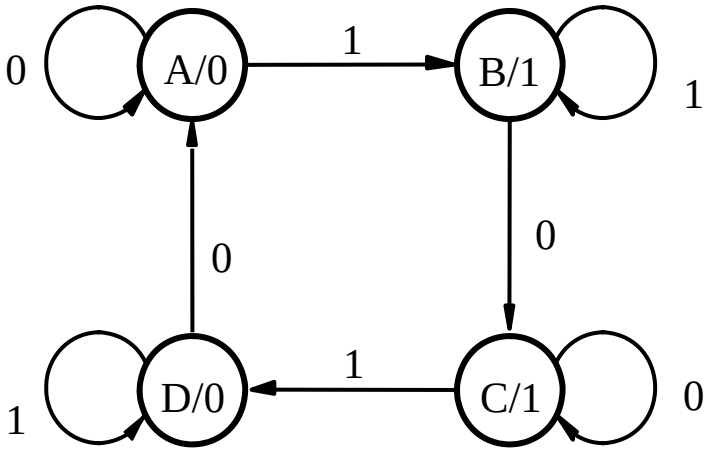
Example: serial parity generator

- Serial parity generator
  - input w : pulses are applied to w
  - output z
  - z=1 if the number of previously applied pulses is odd



(a) State diagram

Parity-generating asynchronous FSM



(a)State diagram

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

(b) Flow table

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	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

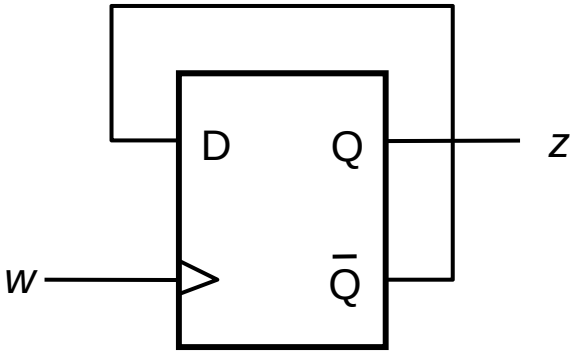
- State assignment (a) has a major flaw
  - state D =11 : w=0 -> state A
  - y_2y_1=11 -> y_2y_1=00
  - the values of the next-state variables determined by the networks of logic gates with varying delays
    - suppose y_1 changes first
      - y_2y_1=10 -> state C(10)
      - state C is stable when w=0
    - suppose y_2 changes first
      - y_2y_1=01-> state B (01)
      - try to change to y_2y_1=10 when w=0
      - if y_1 changes first, y_2y_1=00
    - *race condition occurs*

Circuit that implements the FSM

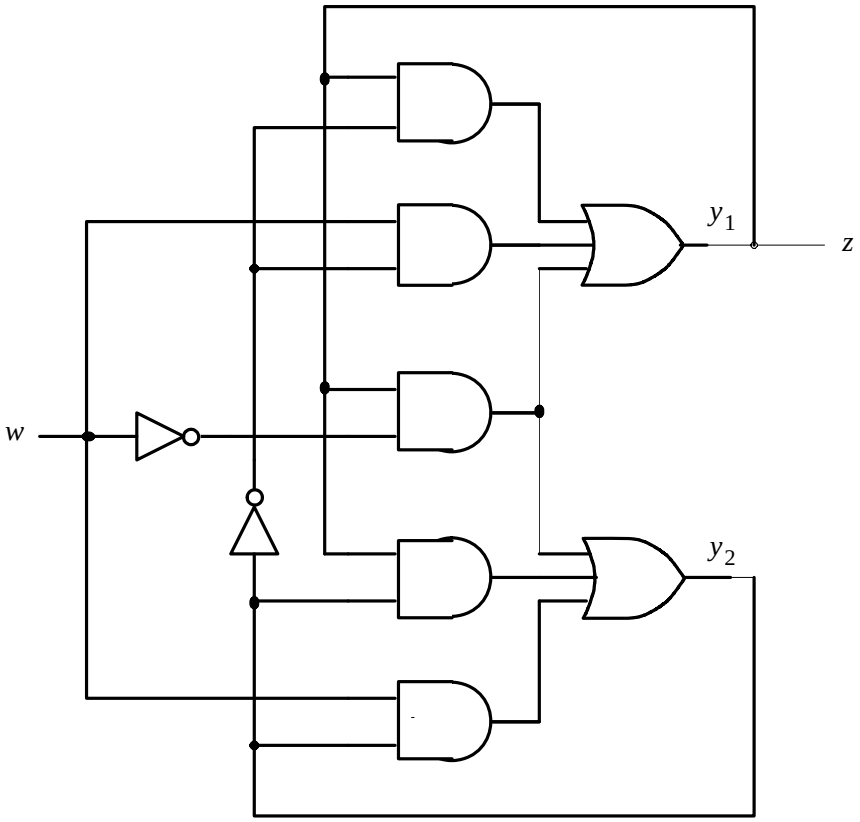
$$Y_1 = w\bar{y}_2 + \bar{w}y_1 + y_1\bar{y}_2$$

$$Y_2 = wy_2 + \bar{w}y_1 + y_1y_2$$

$$z = y_1$$



Synchronous solution



Asynchronous solution

Circuit that implements a parity-generating asynchronous FSM

- The asynchronous implementation is *more complex* than the synchronous one*?*
- It's a negative-edge-triggered master/slave F/F
  - With the complement of its output connected to its D input

Master-slave D F/F(example 9.2)

- Analyze synchronous circuit as if it were an asynchronous circuit.
  - Actually all circuits are asynchronous

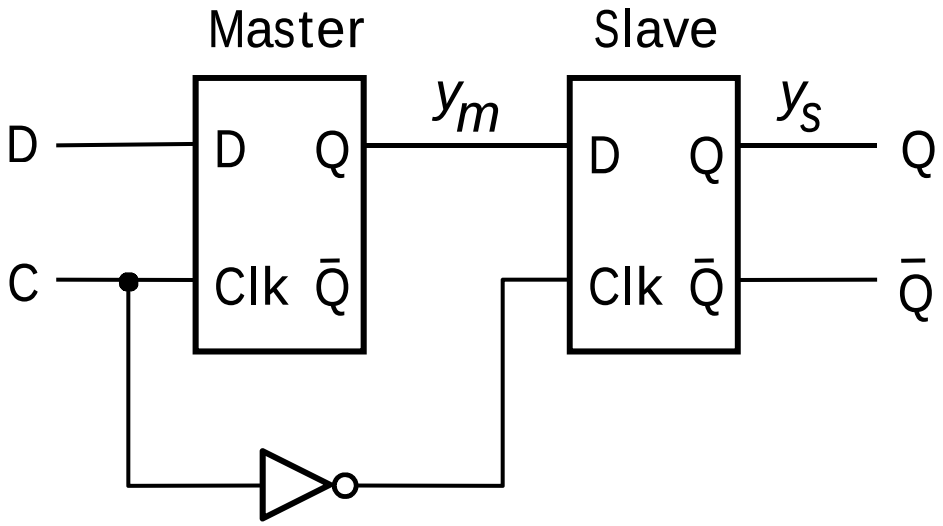
$$Y = CD + \overline{C}y + Dy$$

in the previous example

of gated D - Latch

$$Y_m = CD + \overline{C}y_m + Dy_m$$

$$Y_s = \overline{C}y_m + Cy_s + y_m y_s$$



Circuit for the master-slave D flip-flop.

Excitation table for example 9.2

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) Excitation table

Flow tables for Example 9.2

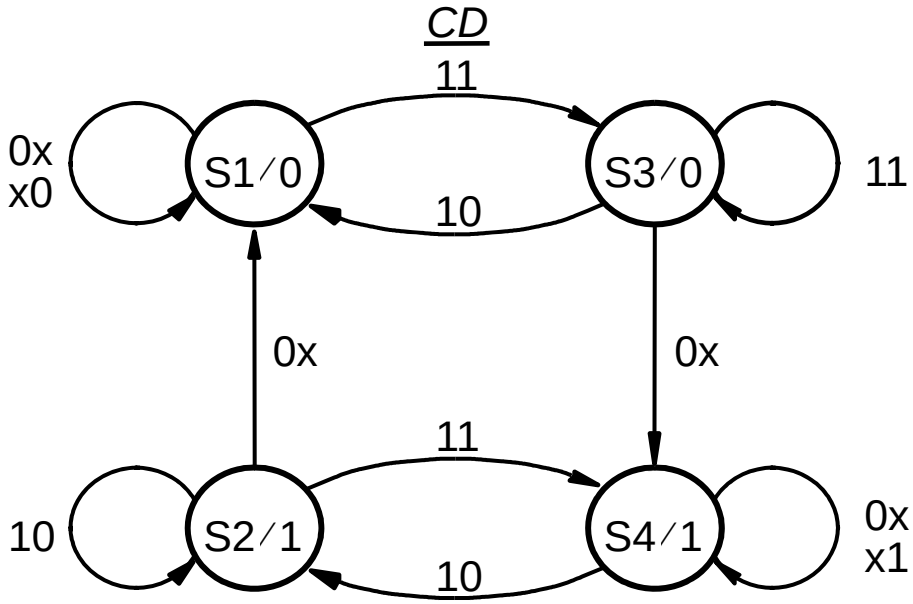
Present state	Next state				Output Q
	CD = 00	01	10	11	
S1	(S1)	(S1)	(S1)	S3	0
S2	S1	S1	(S2)	S4	1
S3	S4	S4	S1	(S3)	0
S4	(S4)	(S4)	S2	(S4)	1

(b) Flow table

Present state	Next state				Output Q
	CD = 00	01	10	11	
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

State diagram for the master-slave D Flip/Flop



Parity generating FSM and Master–slave D F/F

$$Y_1 = w\bar{y}_2 + \bar{w}y_1 + y_1\bar{y}_2$$
$$Y_2 = wy_2 + \bar{w}y_1 + y_1y_2$$
$$z = y_1$$

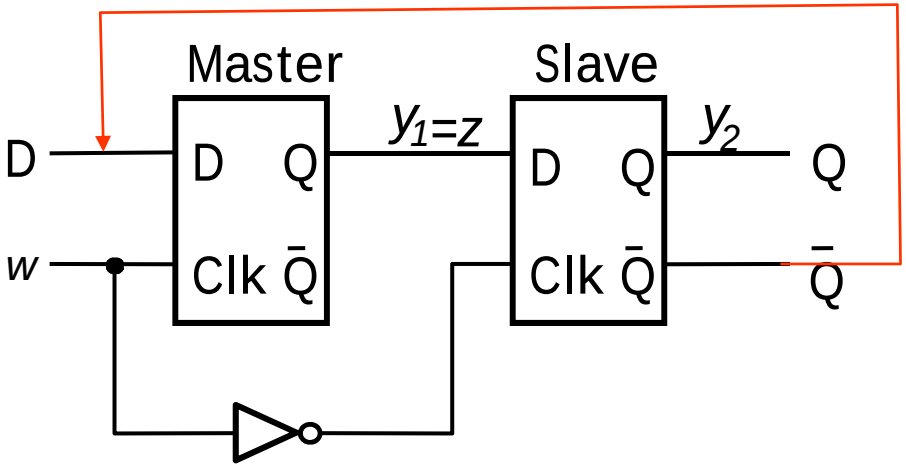
$Y = CD + \bar{C}y + Dy$

in the previous example

of gated D - Latch

$$Y_m = CD + \bar{C}y_m + Dy_m$$
$$Y_s = \bar{C}y_m + Cy_s + y_my_s$$

$$y_1 = y_m, y_2 = y_s$$
$$w = C, \bar{y}_2 = D,$$
$$z = y_1 = y_m$$



Hazard and Glitches

Hazards and glitches

- In asynchronous circuits
  - undesirable glitches on signals should not occur
  - hazards
    - the glitches cause by the structure of a given circuit and propagation delays in the circuit
  - two types of hazards
    - static
      - the signal undergoes a momentary change in its required value
    - dynamic
      - when a signal is supposed to change from 1 to 0 or from 0 to 1
      - a change involves a short oscillation before the signal settles into its new level

Definition of hazards

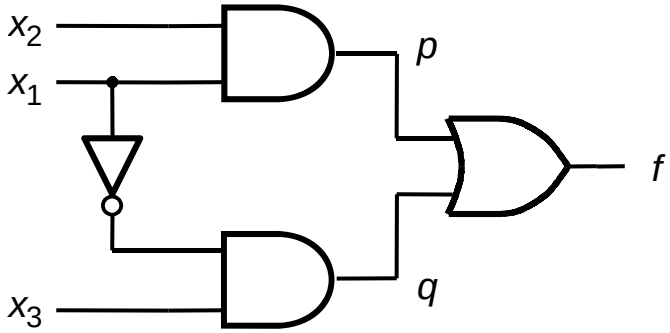
(a) Static hazard

(b) Dynamic hazard

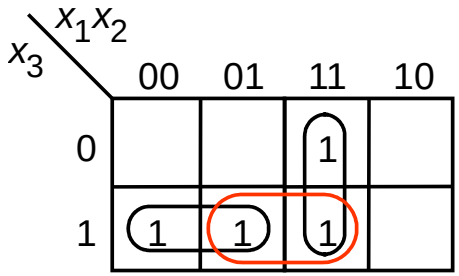
Hazards and glitches

- Usual solutions
  - wait until signals are stable by using a clock
    - preferable
    - easiest to design when there is a clock
    - synchronous circuits
  - design hazard-free circuits
    - sometimes necessary
    - asynchronous design

Static hazards



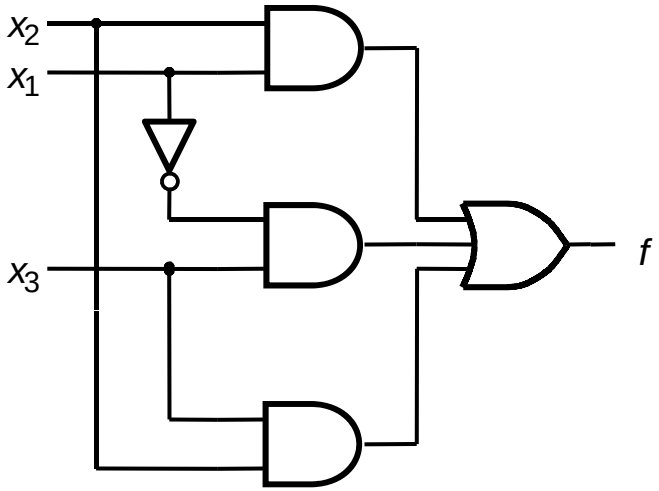
(a) Circuit with a hazard



(b) Karnaugh map

$$f = x_1x_2 + \bar{x}_1x_3$$

$$f = x_1x_2 + \bar{x}_1x_3 + x_2x_3$$



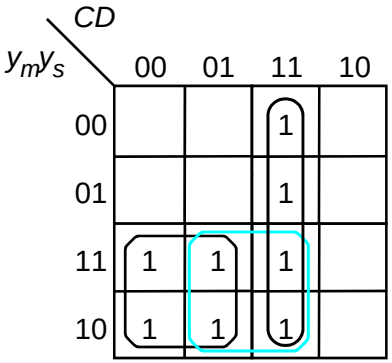
(c) Hazard-free circuit

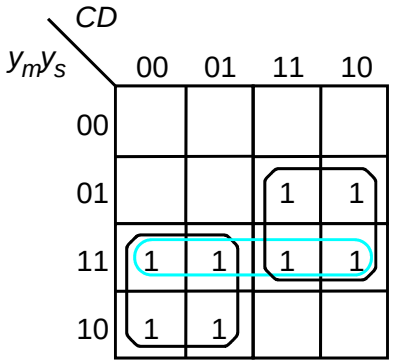
hazard-free if more than one bit of inputs change simultaneously ?

Two-level implementation of master-slave D flip-flop

Present state $y_m\ y_s$	Next state $Y_m\ Y_s$ for CD = 00	01	10	11	Output Q
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



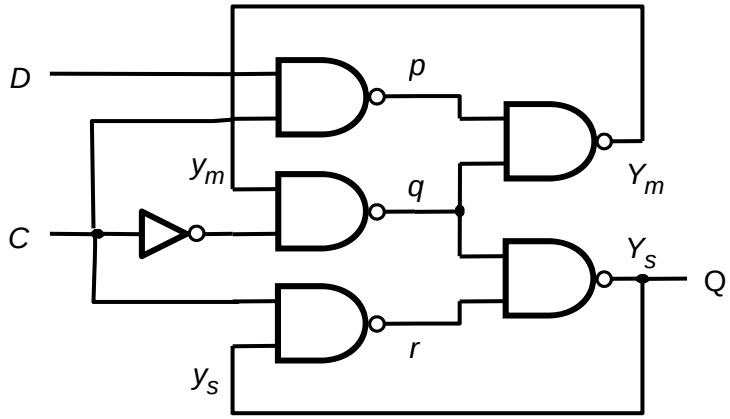


(b) Karnaugh maps for Y_m and Y_s in Figure 9.6*a*

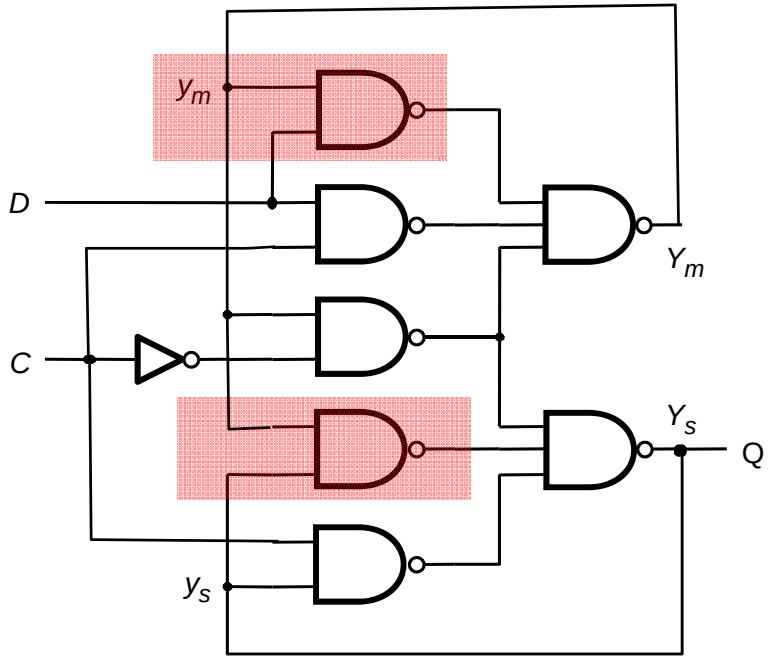
$$Y_m = CD + \overline{C}y_m + Dy_m$$

$$Y_s = \overline{C}y_m + Cy_s + y_m y_s$$

Two-level implementation of master-slave D flip-flop (2)



(a) Minimum-cost circuit

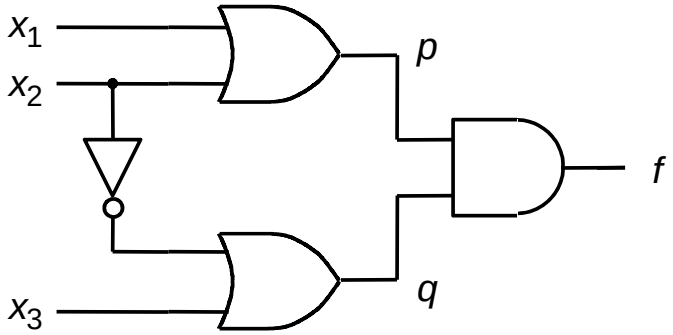


(c) Hazard-free circuit

$$Y_m = CD + \overline{C}y_m + Dy_m$$

$$Y_s = \overline{C}y_m + Cy_s + y_m y_s$$

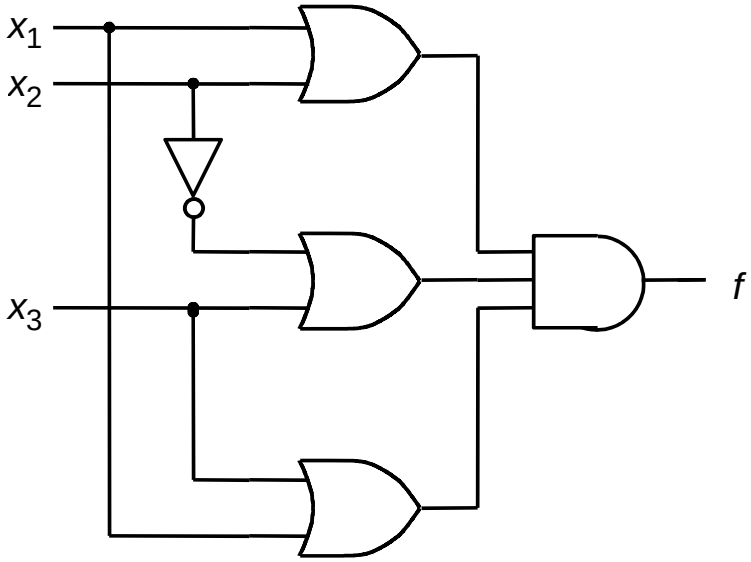
Static hazard in a POS circuit (0–hazard)



(a) Circuit with a hazard

x_3 \ x_1x_2	00	01	11	10
0	0	0	0	1
1	0	1	1	1

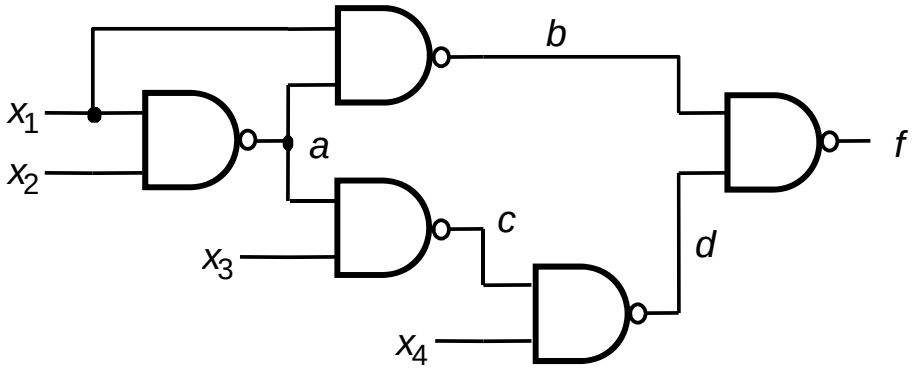
(b) Karnaugh map



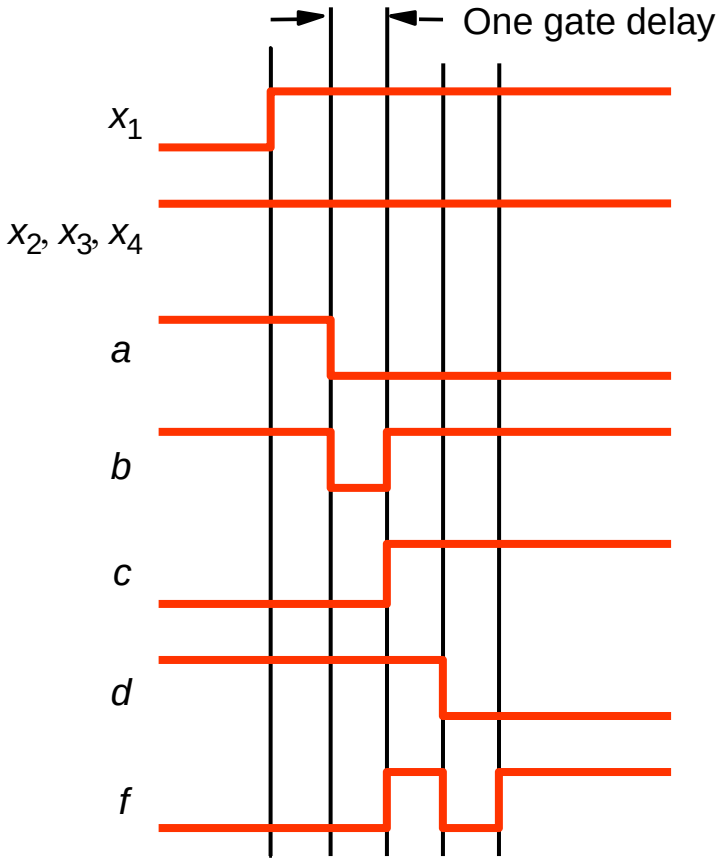
(c) Hazard-free circuit

dynamic hazards

- there exist multiple paths for a given signal change to propagate along
- neither easy to detect nor easy to deal with
- using two-level hazard-free circuits



(a) Circuit

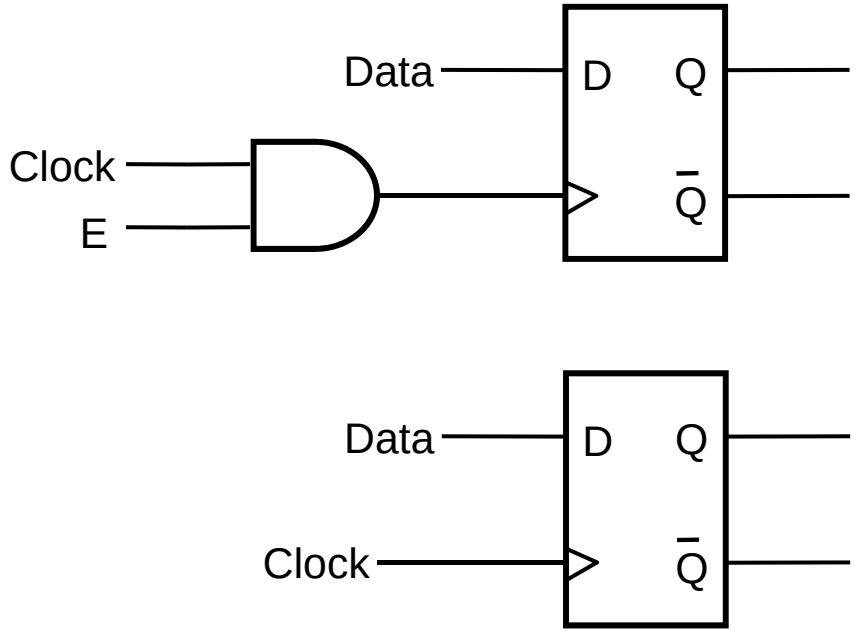


(b) Timing diagram

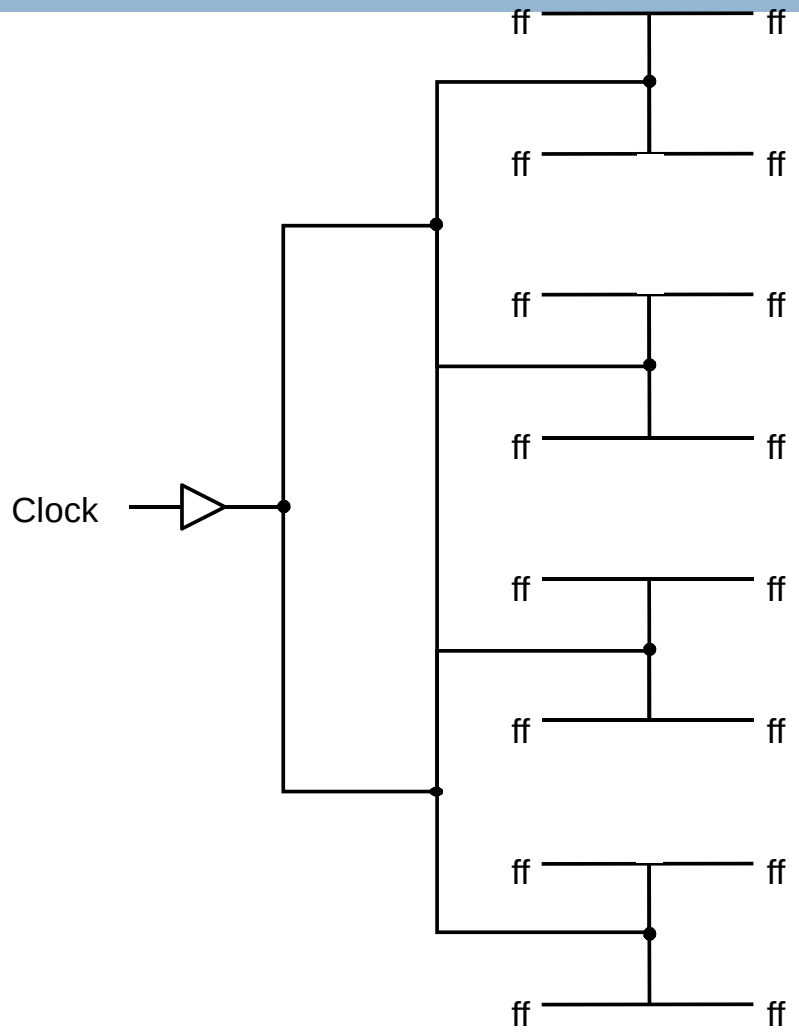
CLOCK SYNCHRONIZATION (CHAPTER 10.3)

Clock skew

- the clock signal arrives at different times at different F/Fs
  - with or without clock enable circuits
  - wires whose lengths vary appreciably

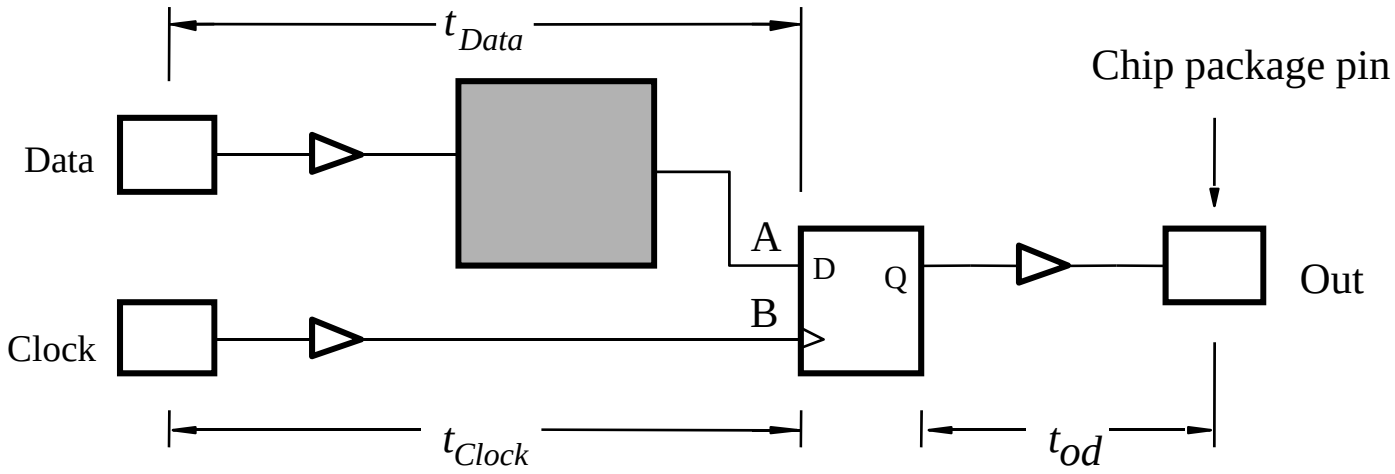


An H-tree clock distribution network



F/F timing parameters

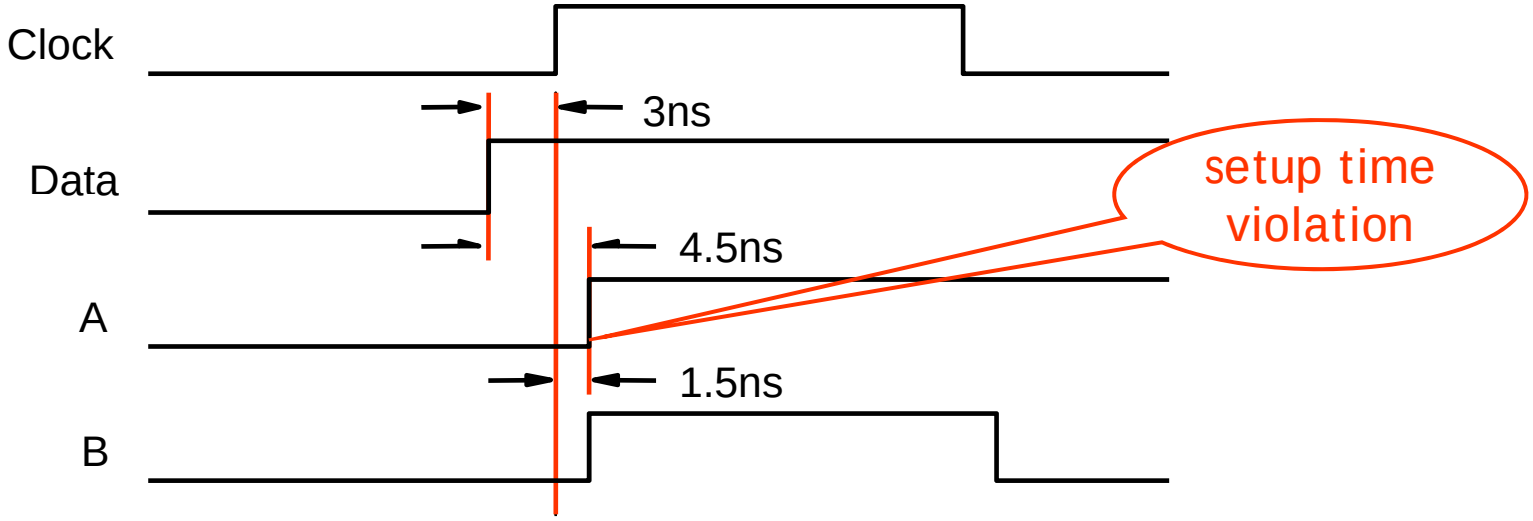
- setup time t_{su}
- hold time t_h
- register delay or propagation delay t_{rd}
- output delay time t_{od}
  - required for the change in Q to propagate to an output pin on the chip



A flip-flop in an integrated circuit

F/F timing parameters, cont'd

- t_{CO} delay : active clock edge -> output change at an output pin
  - $t_{Clock} + t_{rd} + t_{od}$
  - Example
    - t_{Clock} = 1.5ns, t_{rd}=1ns, t_{od}=2ns -> t_{co} =4.5 ns
- F/F timing in a chip
  - t_{Clock}=1.5 ns, t_{Data}=4.5 ns, t_{su}=3 ns



Metastability and Asynchronous Inputs

Asynchronous Inputs Are Dangerous!

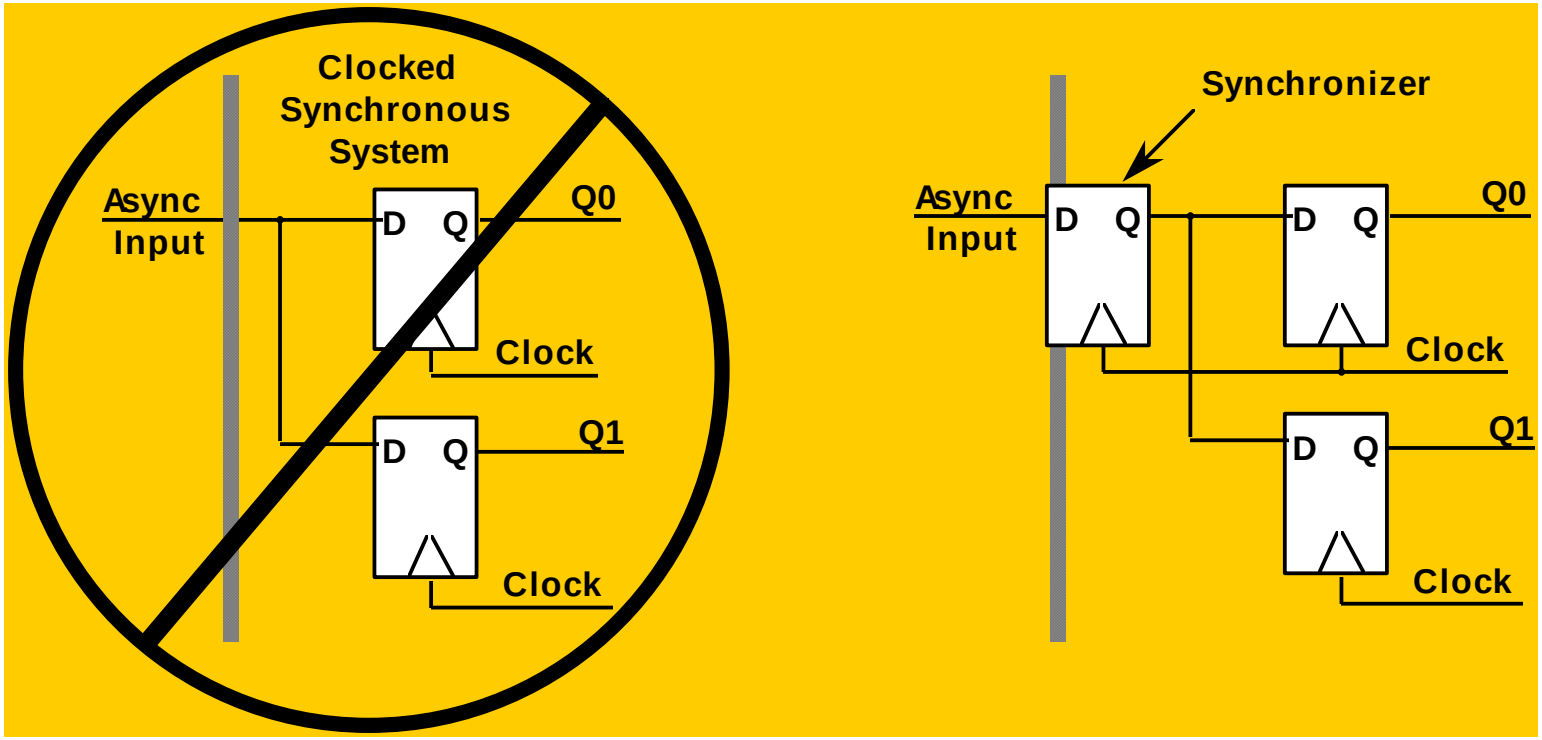
Since they take effect immediately, glitches can be disastrous

Synchronous inputs are greatly preferred!

But sometimes, asynchronous inputs cannot be avoided
e.g., reset signal, memory wait signal

Metastability and Asynchronous Inputs

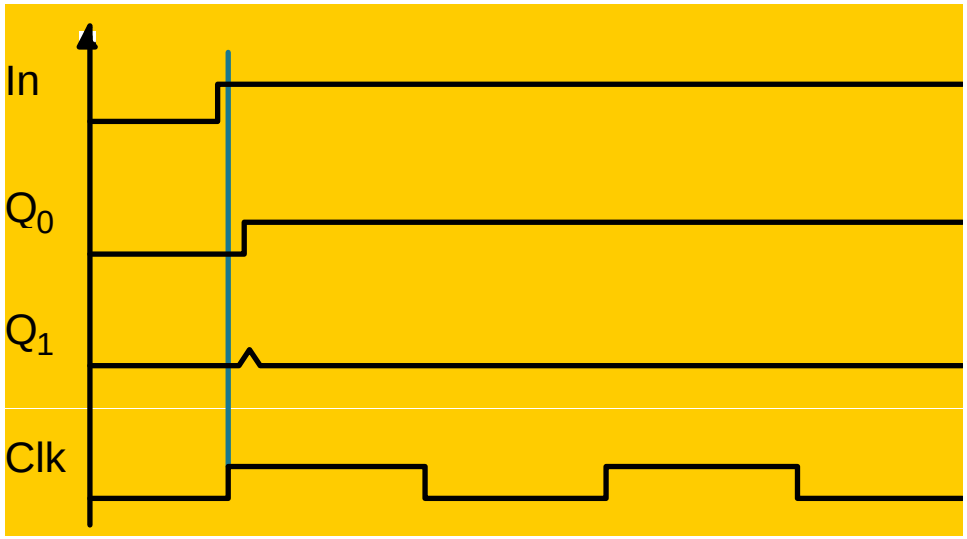
Handling Asynchronous Inputs



Never allow asynchronous inputs to be fanned out to more than one FF within the synchronous system

Metastability and Asynchronous Inputs

What Can Go Wrong



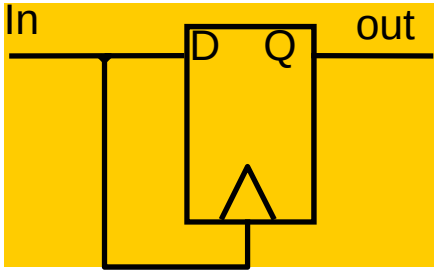
In is asynchronous
Fans out to D0 and D1
One FF catches the signal, one does not

impossible state might be reached!

Single FF that receives the asynchronous signal is *a synchronizer*

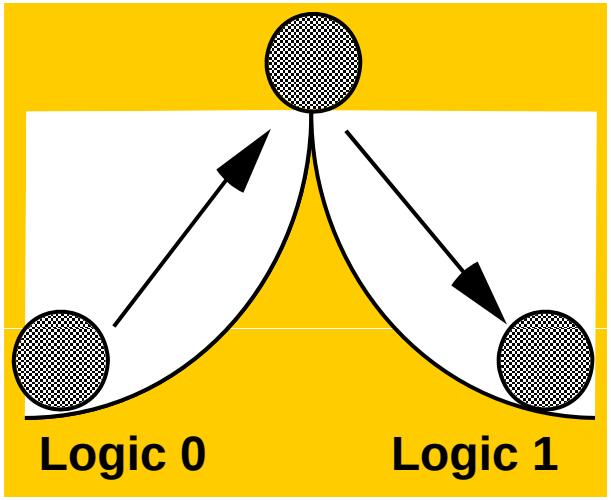
Metastability and Asynchronous Inputs

Synchronizer Failure

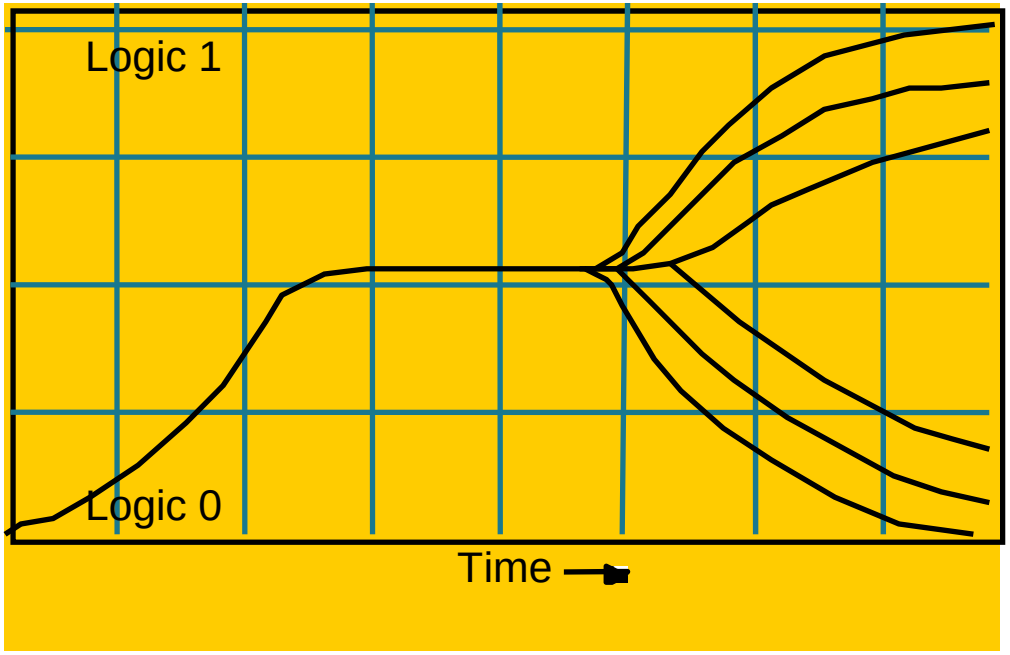


When FF input changes close to clock edge, the FF may enter the *metastable* state: neither a logic 0 nor a logic 1

It may stay in this state an indefinite amount of time, although this is not likely in real circuits



Small, but non-zero probability that the FF output will get stuck in an in-between state



Oscilloscope Traces Demonstrating Synchronizer Failure and Eventual Decay to Steady State

Metastability and Asynchronous Inputs

Solutions to Synchronizer Failure

- **the probability of failure can never be reduced to 0, but it can be reduced**

- **slow down the system clock**
  this gives the synchronizer more time to decay into a steady state
  synchronizer failure becomes a big problem for very high speed systems

- **use fastest possible logic in the synchronizer**
  this makes for a very sharp "peak" upon which to balance
  S or AS TTL D-FFs are recommended

- **cascade two synchronizers**