CSE 140
Discussion Session

Akanksha Maurya

Topics

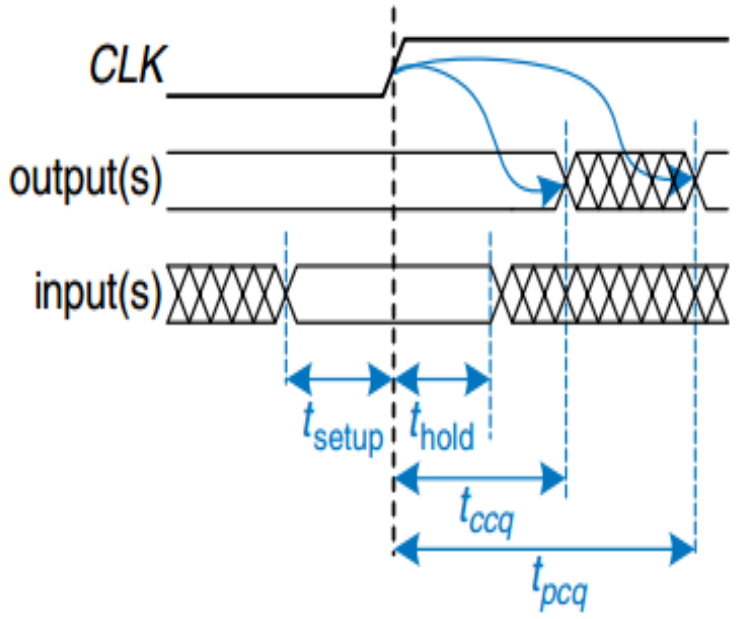
- ✓ Sequential Network: Timing
  - ✓ Setup and Hold constraint
  - ✓ Clock Skew
  - ✓ Retiming
- ✓ Standard Combinational Modules
  - ✓ Decoder
  - ✓ Multiplexer

Sequential Network: Timing

- A sequential element has an *aperture time* (t_a)around the clock edge, during which input must be stable for the flip-flop to produce a well-defined output.

$$t_a = t_{setup} + t_{hold}$$

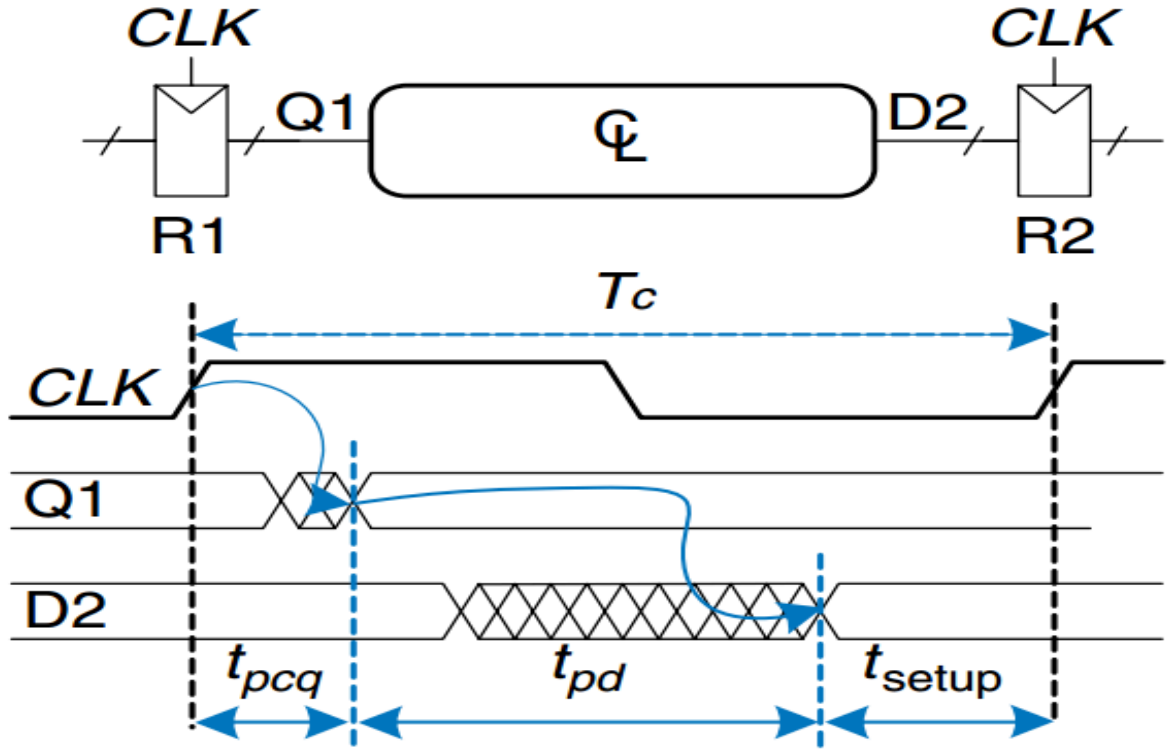
- **Setup time** (t_{setup}): time for which the input should be stable before the clock changes
- **Hold time** (t_{hold}): time for which the input should be stable after the clock changes
- **Contamination delay** (t_{ccq}): time after clock edge that output may start changing (unstable)
- **Propagation delay** (t_{pcq}): time after clock edge that output must stop changing (stable)



Setup Time Constraint (without skew)

$$T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}}$$

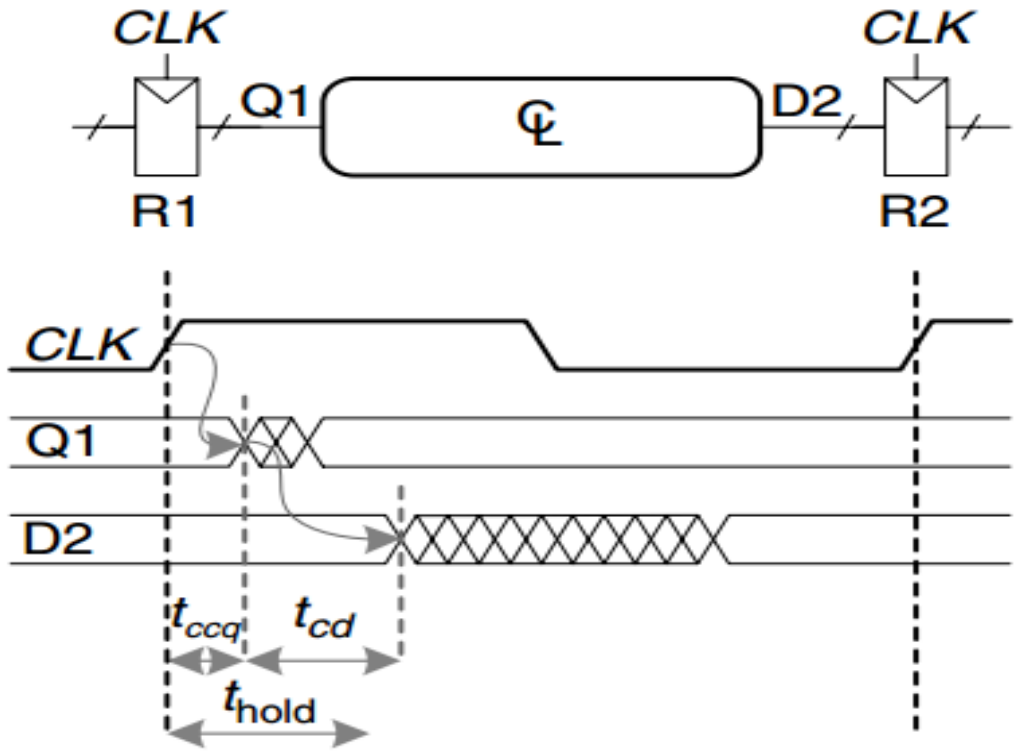
- Maximum operating frequency
- Propagation delay of longest path



Hold Time Constraint (without skew)

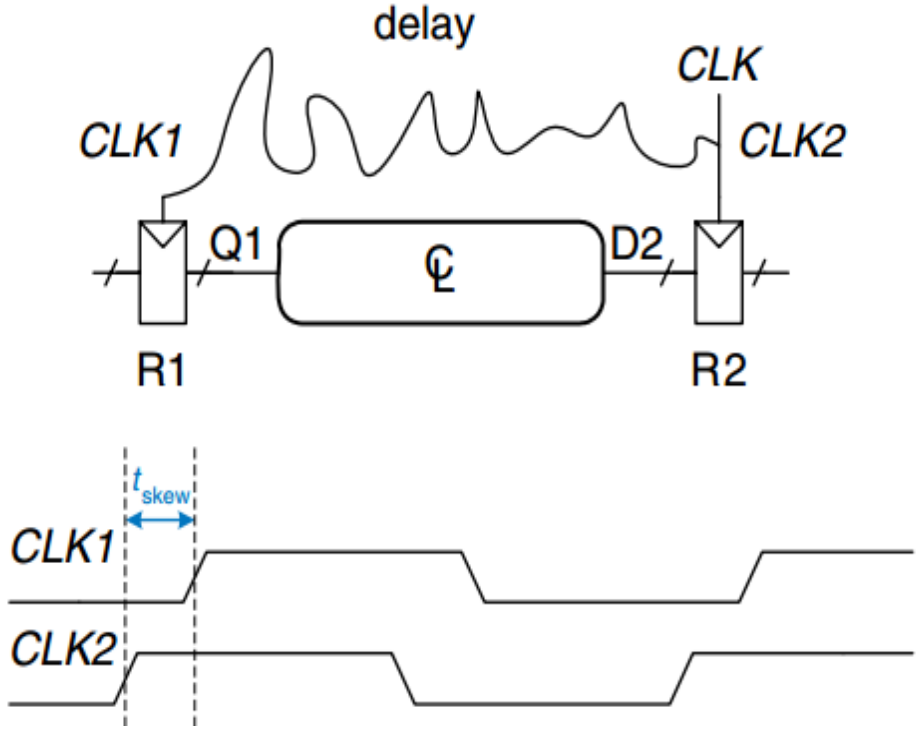
$$t_{ccq} + t_{cd} \geq t_{\text{hold}}$$

- Contamination delay of shortest path



Clock Skew

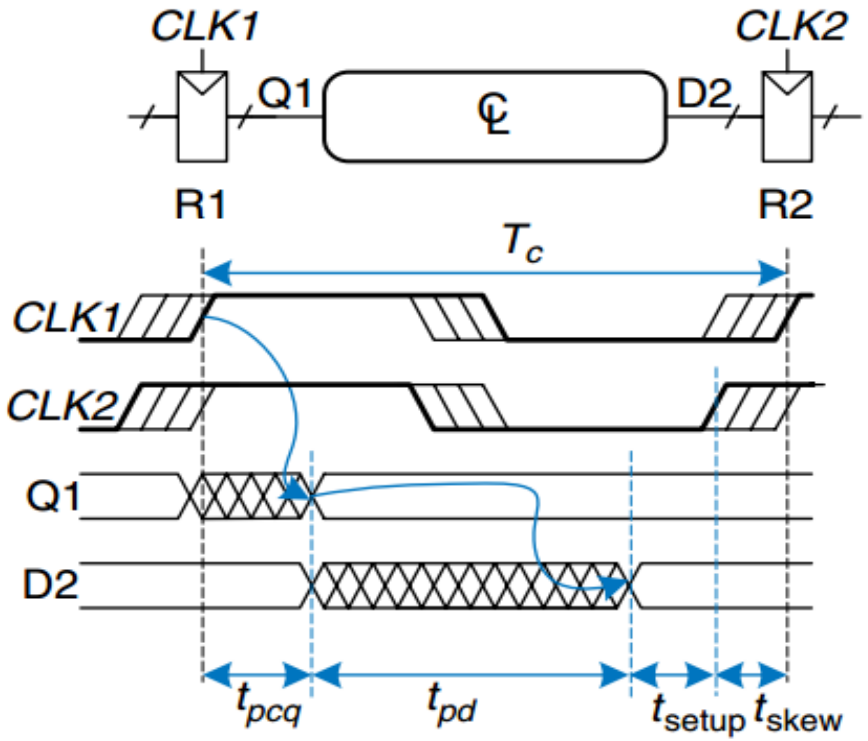
- The clock doesn't arrive at all registers at the same time.
- Clock Skew (t_{skew}) is the difference between two clock edges



Setup Time Constraint (clock skew)

$$T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}} + t_{\text{skew}}$$

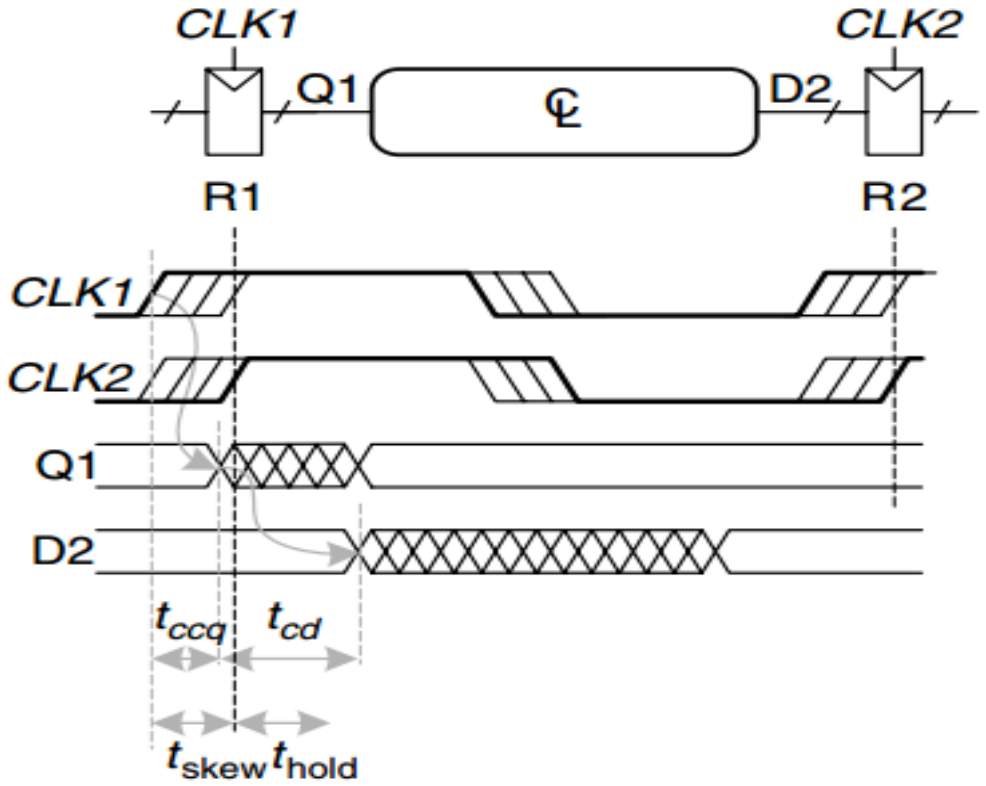
- In the worst case, R1 receives the latest skewed clock and R2 receives the earliest skewed clock, leaving as little time as possible for data to propagate between the registers.



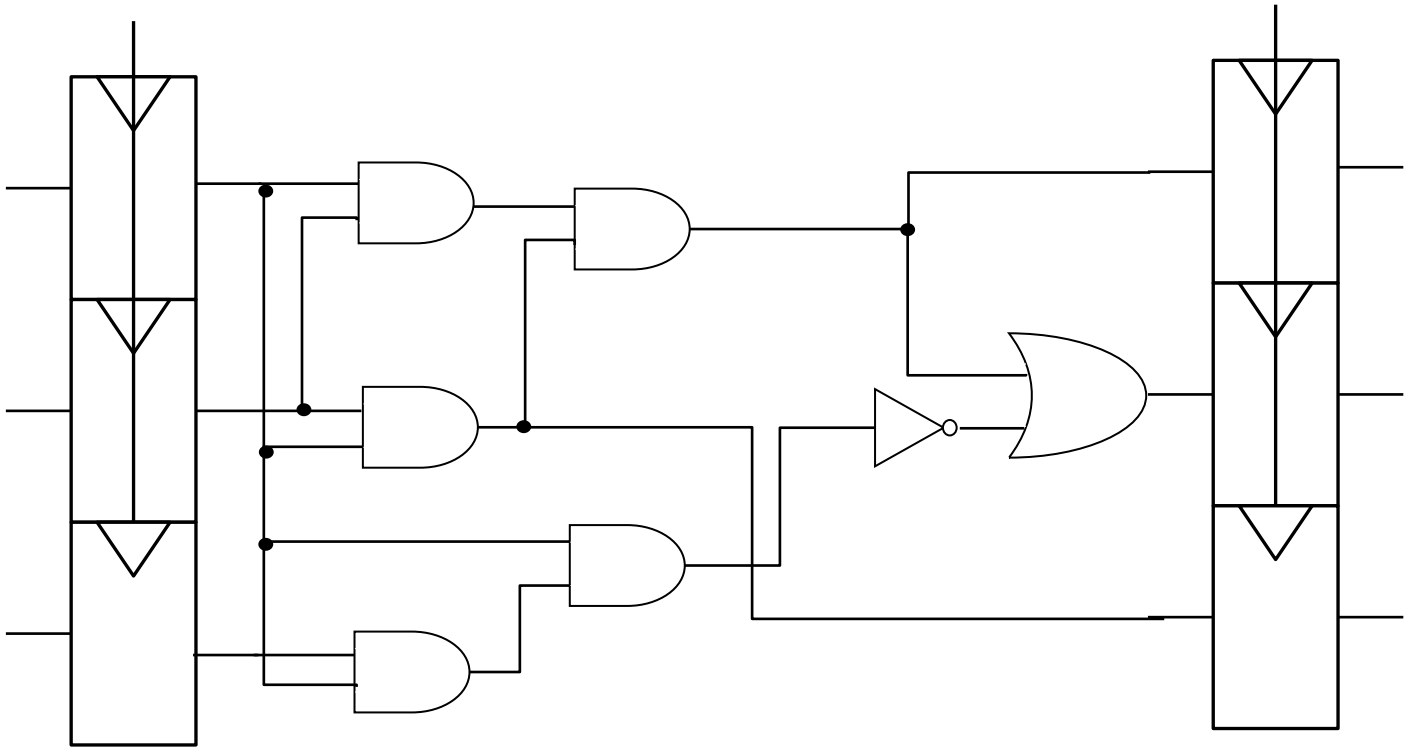
Hold Time Constraint (clock skew)

$$t_{ccq} + t_{cd} \geq t_{\text{hold}} + t_{\text{skew}}$$

- In the worst case, R1 receives an early skewed clock, *CLK*1, and R2 receives a late skewed clock, *CLK*2. The data zips through the register and combinational logic but must not arrive until a hold time after the late clock.

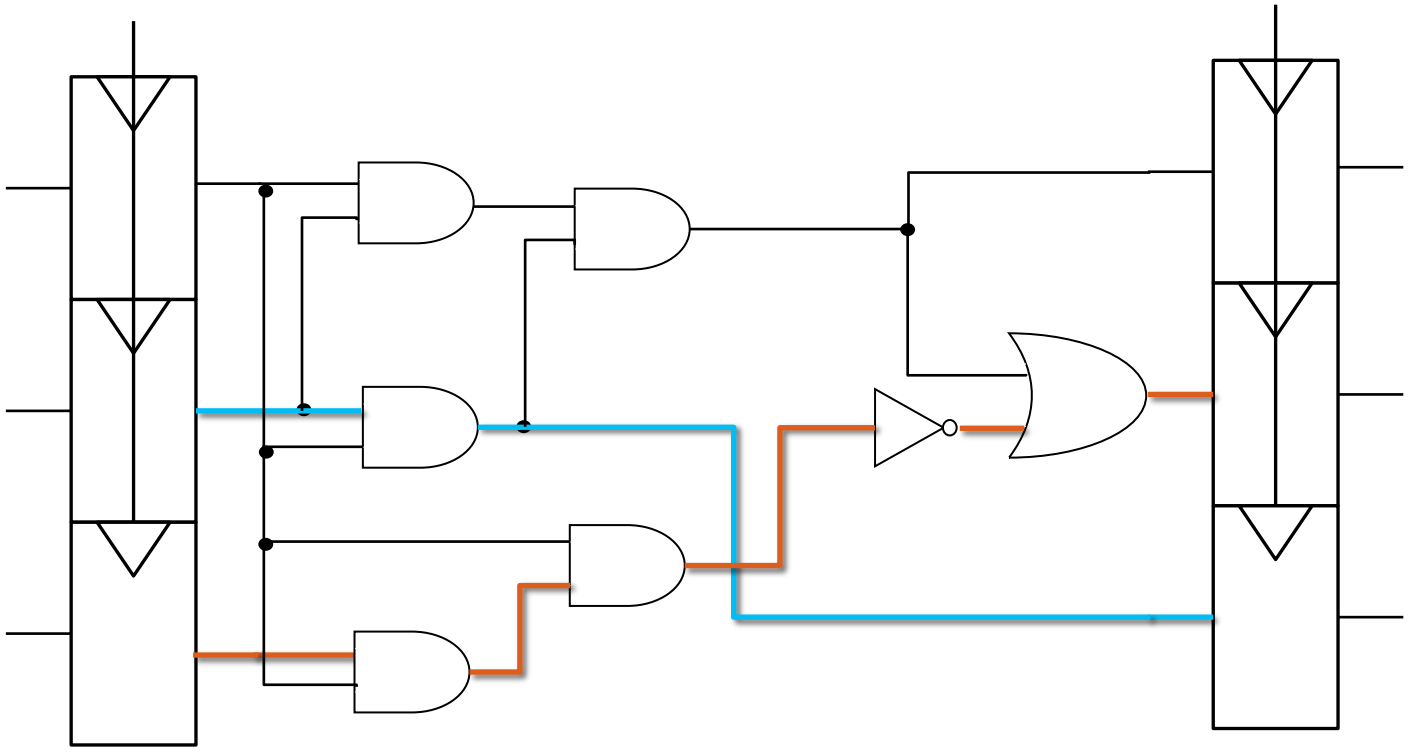


Timing Analysis: Example



- D-FF:
  - t_{ccq} = 40ps
  - t_{pcq} = 60ps
  - t_{setup} = 70ps
  - t_{hold} = 80ps
- AND:
  - t_{pd} = 40ps
  - t_{cd} = 30ps
- OR:
  - t_{pd} = 40ps
  - t_{cd} = 30ps
- NOT:
  - t_{pd} = 30ps
  - t_{cd} = 20ps
- BUF:
  - t_{pd} = 30ps
  - t_{cd} = 20ps

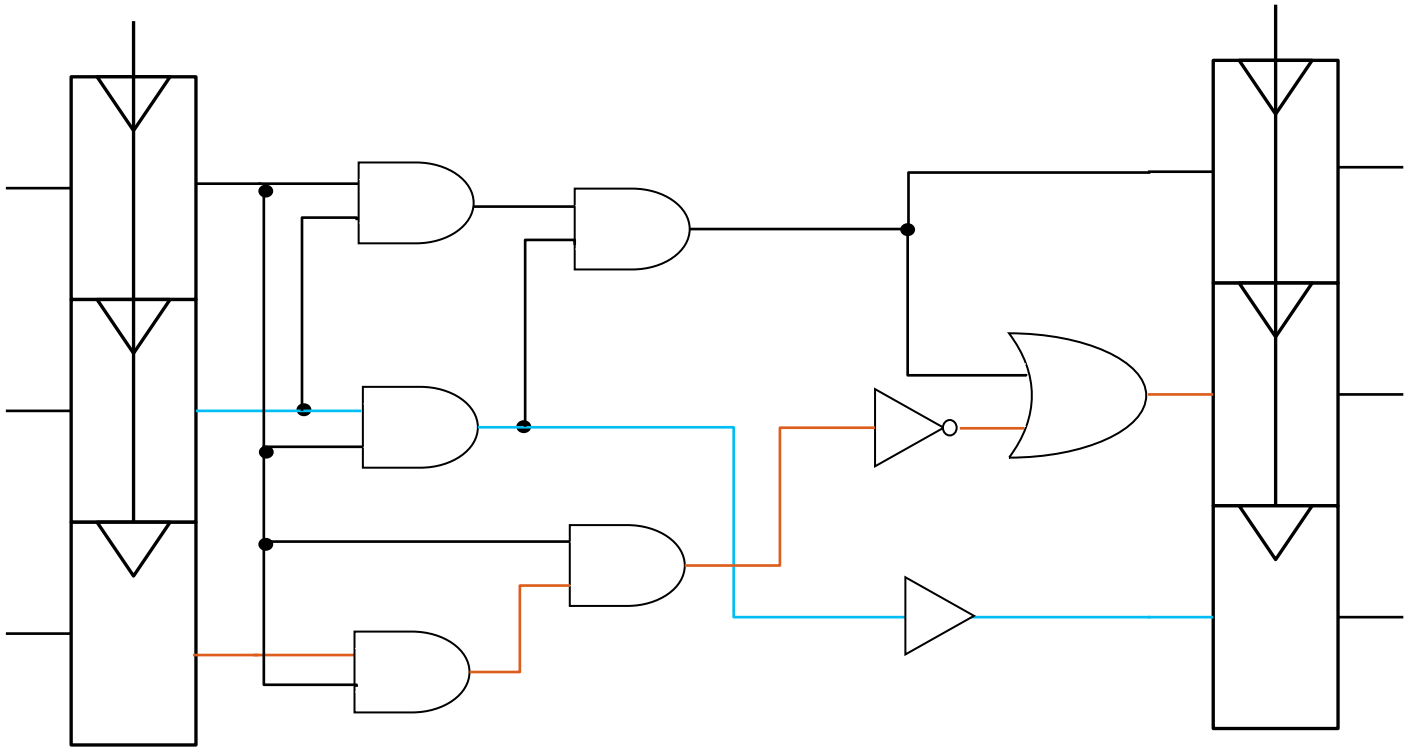
Timing Analysis: Example



Longest path Shortest path

- D-FF:
  - t_{ccq} = 40ps
  - t_{pcq} = 60ps
  - t_{setup} = 70ps
  - t_{hold} = 80ps
- AND:
  - t_{pd} = 40ps
  - t_{cd} = 30ps
- OR:
  - t_{pd} = 40ps
  - t_{cd} = 30ps
- NOT:
  - t_{pd} = 30ps
  - t_{cd} = 20ps
- BUF:
  - t_{pd} = 30ps
  - t_{cd} = 20ps

Timing Analysis: Example



- Setup time constraint

  $T_c \geq t_{pcq} + t_{pd} + t_{setup}$

  $T_c \geq (60) + (3*40 + 30) + 70$

  $T_c \geq 280$ ps

Max. operating frequency:

F= 1/T = 3.57 GHz

- Hold time constraint

  $t_{hold} \leq t_{ccq} + t_{cd}$

  $80 \leq 40 + 30$

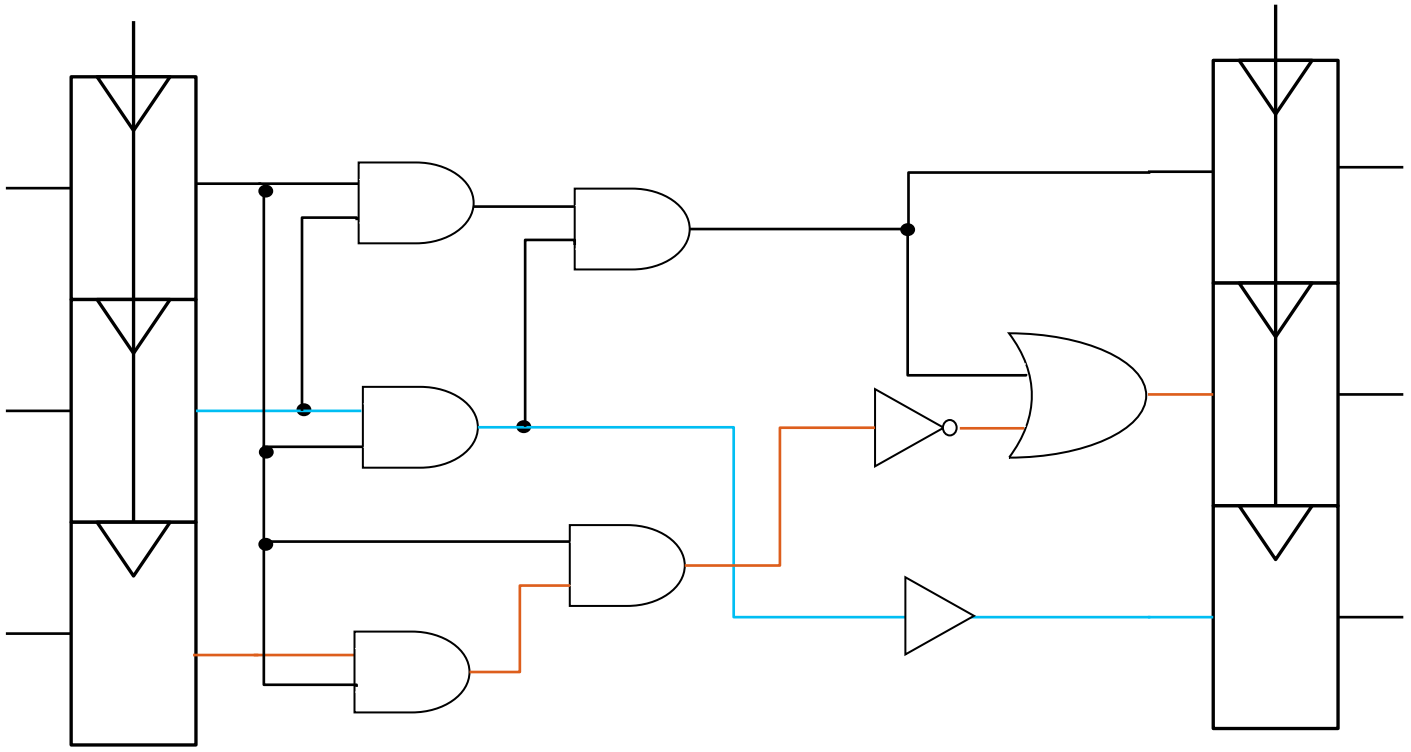
Hold time violation!!!!

- Add Buffer

  $t_{hold} \leq t_{ccq} + t_{cd}$

  $80 \leq 40 + 30 + 20$

Timing Analysis: Example (with skew)

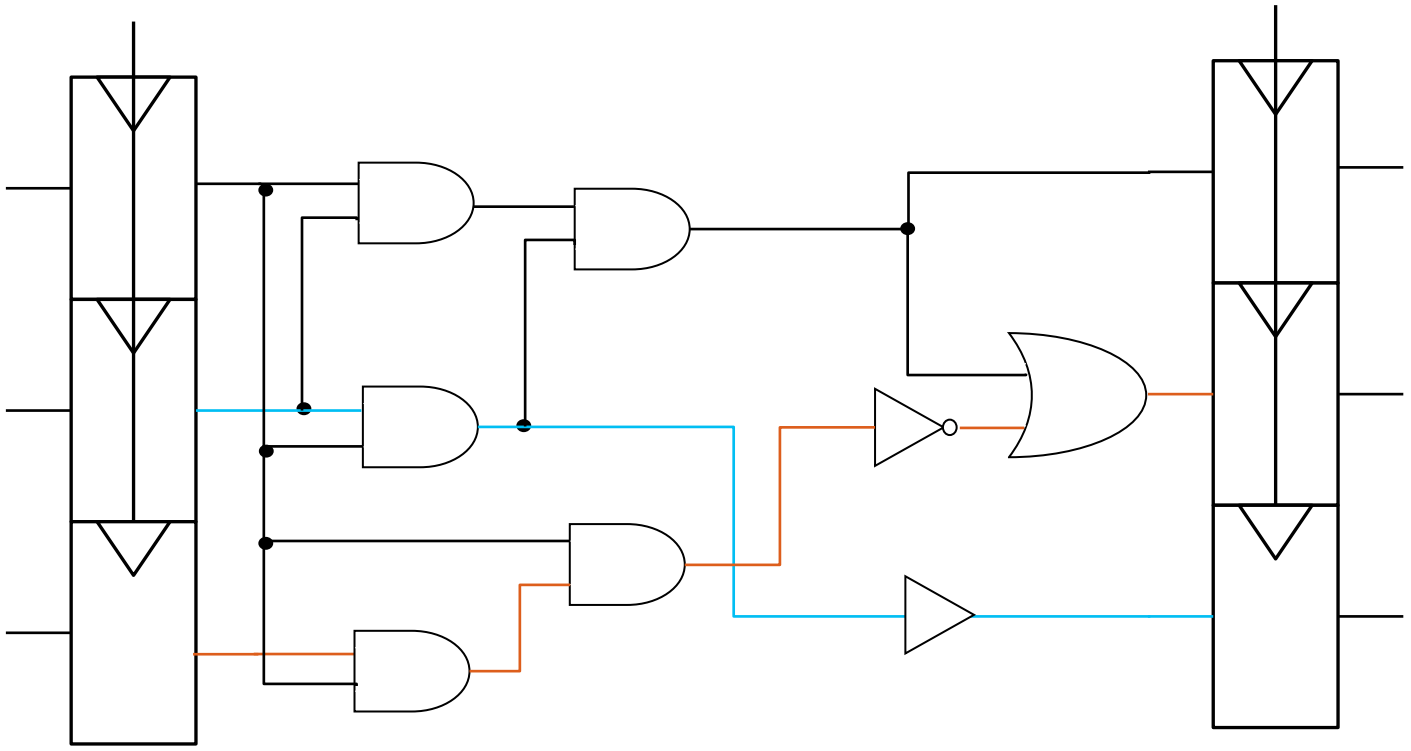


- Setup time constraint
  - t_{skew} = 20ps
  - $T_c \geq t_{pcq} + t_{pd} + t_{setup} + t_{skew}$
  - $T_c \geq (60) + (3*40 + 30) + 70 + 20$
  - $T_c \geq 300$ ps

Max. operating frequency:
F= 1/T = 3.33 GHz

- Hold time constraint
  - Tolerable skew
    - $t_{hold} + t_{skew} \leq t_{ccq} + t_{cd}$
    - $80 + t_{skew} \leq 40 + 30 + 20$
    - $t_{skew} \leq 10$ps

Retiming



- Adjust clock skew so that max operating frequecy can be reduced
- Setup time constraint

  $t_{skew} = \Delta$

  $T_c \geq t_{pcq} + t_{pd} + t_{setup} - \Delta$

  $T_c + \Delta \geq (60) + (3*40 + 30) + 70$

  $T_c + \Delta \geq 280$ ps
- Hold time constraint

  $t_{hold} + \Delta \leq t_{ccq} + t_{cd}$

  $80 + \Delta \leq 40 + 30 + 20$

  $\Delta \leq 10$ps

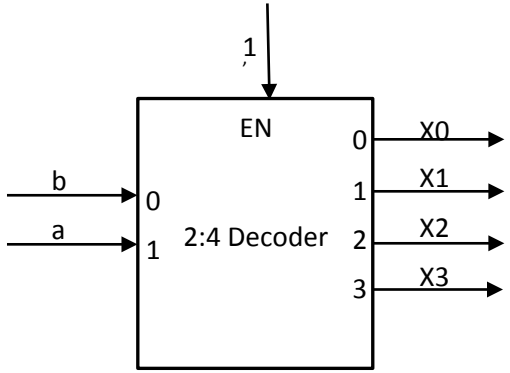
➔ T = 270 ps

➔ Max operating frequency:

F= 1/T = 3.7GHz

Standard Combinational Modules

- **Decoder**
  - n to 2^n decoder
  - Only 1 output HIGH at most

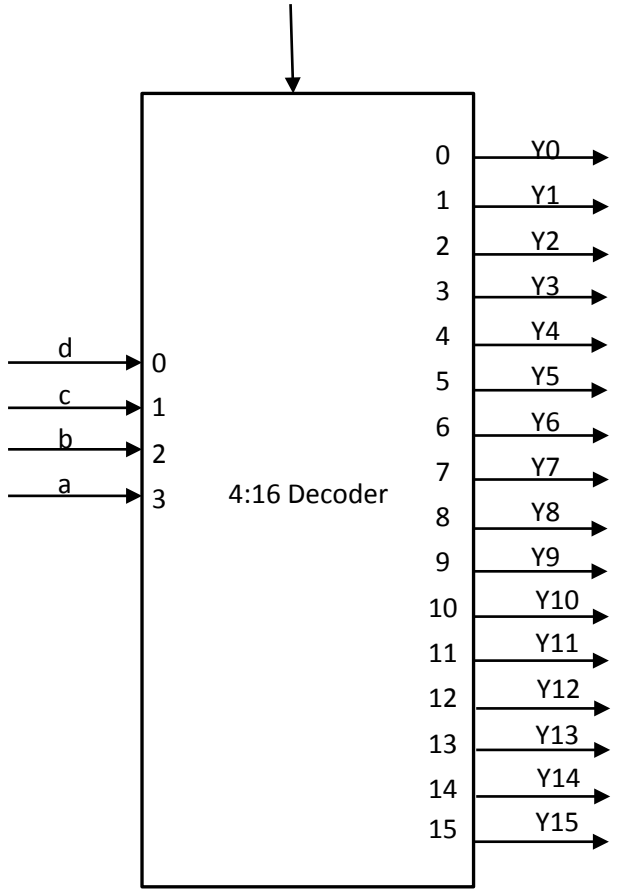


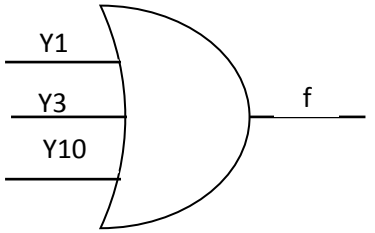
a	b	X3	X2	X1	X0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

- Apply to implement Boolean function
  - Decoder produces min-terms when E=1. (X0 = a'b', X1 = a'b, X2 = ab', X3 = ab)
  - Use an OR gate to collect the min-terms to cover the On-set
  - For the Don't Care-Set, we can just ignore the terms

Decoder: Example

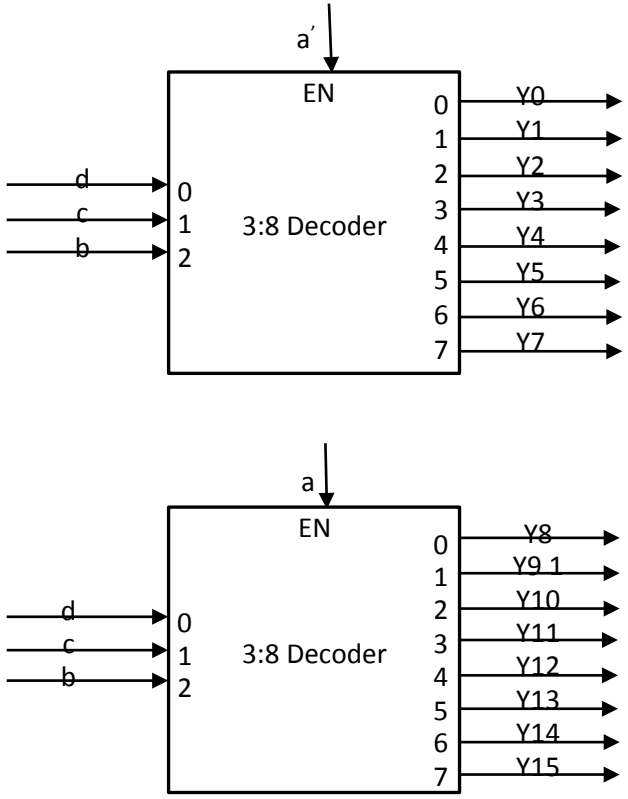
- f(a,b,c,d) = ∑ m(1, 3, 10) + d(2,5)
  - Implement using :
    - 4:16 decoder
    - OR gate

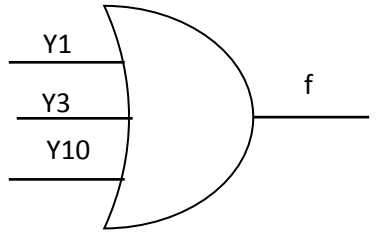




Decoder: Example

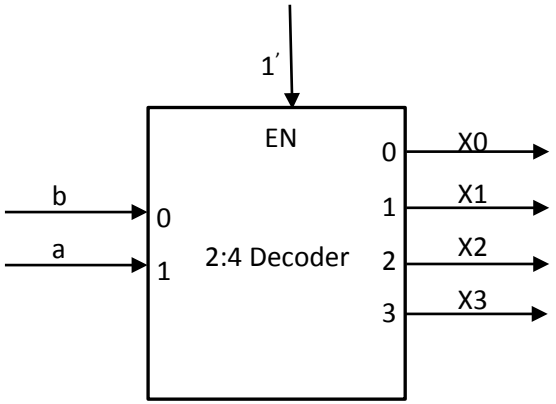
- f(a,b,c,d) = ∑ m(1, 3, 10) + d(2,5)
  - Implement using :
    - 3:8 decoder
    - OR gate

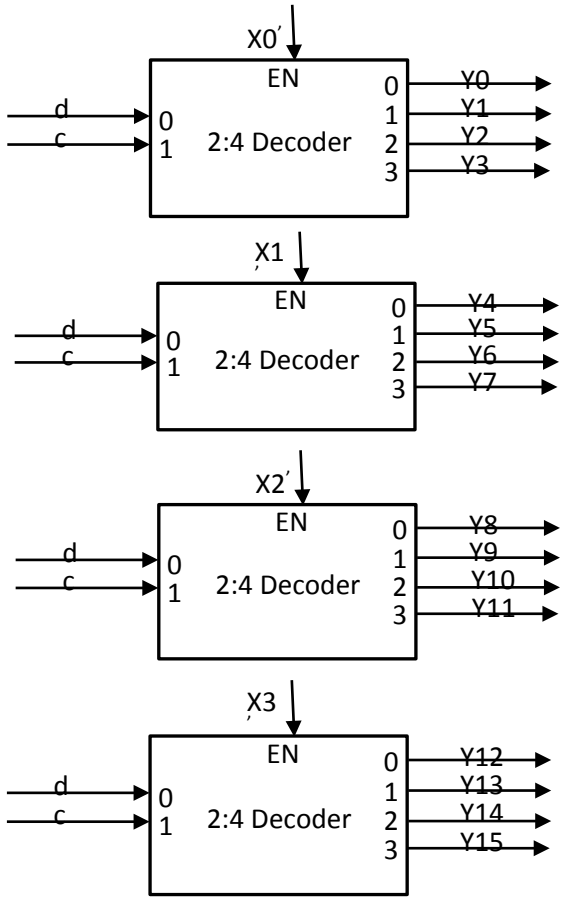


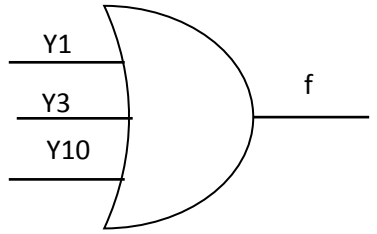


Decoder: Example

- Implement using :
  - 2:4 decoder
  - OR gate

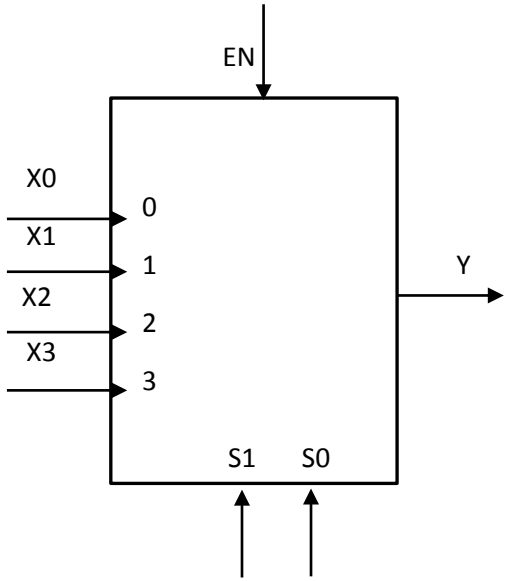






Standard Combinational Modules

- **Multiplexer**
  - Selects one of N inputs to connect to the output
  - log2N-bit select input



S1	S0	Y
0	0	X0
0	1	X1
1	0	X2
1	1	X3

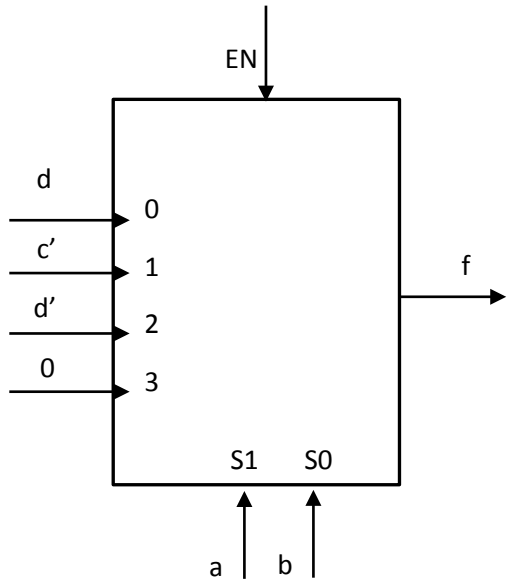
$$Y = X0*S1'*S0' + X1*S1'*S0 + X2*S1*S0' + X3*S1*S0$$

Multiplexer: Example

- f(a,b,c,d) = ∑ m(1, 3, 4, 5, 10) + d(8,12)
  - Implement using 4:1 multiplexer

a	b	f(a,b,0,0)	f(a,b,0,1)	f(a,b,1,0)	f(a,b,1,1)	D(c,d)
0	0	0	1	0	1	**d**
0	1	1	1	0	0	**c'**
1	0	x	0	1	0	**d'**
1	1	x	0	0	0	**0**

	a	b	c	d	f
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	x
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	x
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0



Multiplexer: Example

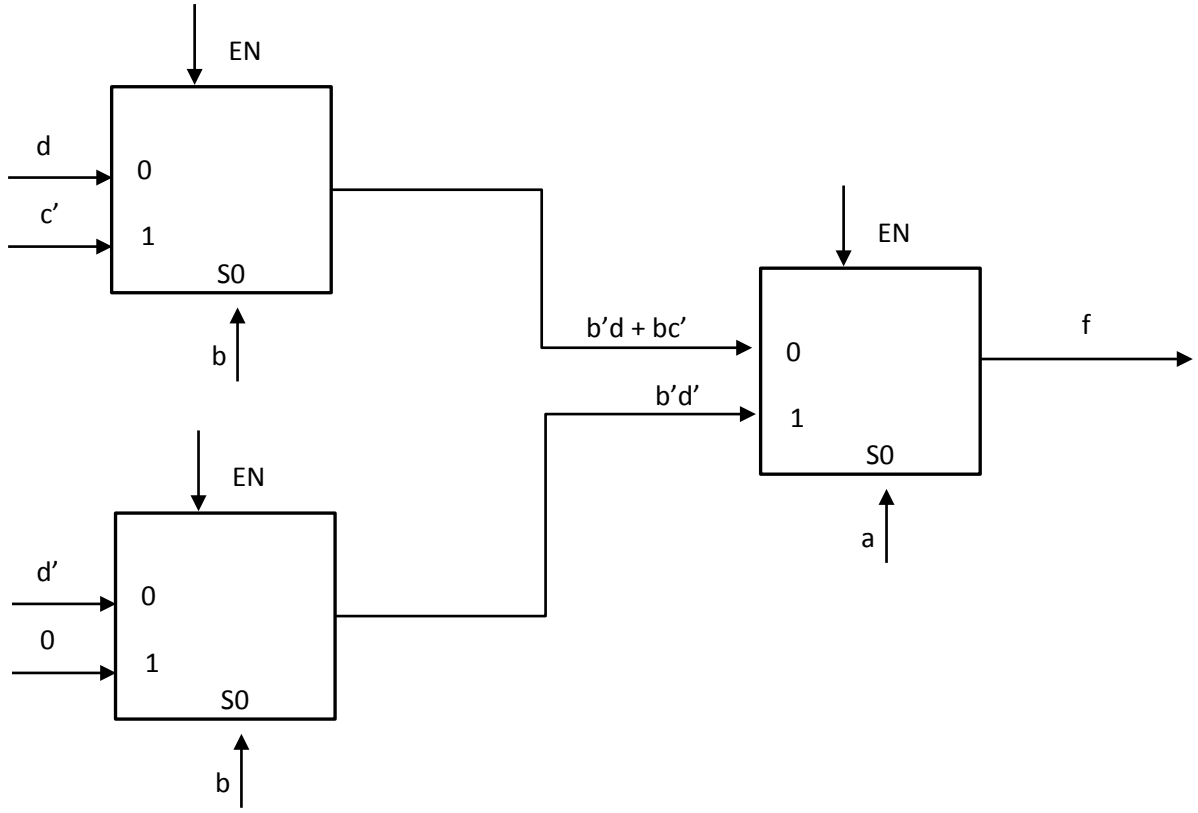
- f(a,b,c,d) = ∑ m(1, 3, 4, 5, 10) + d(8,12)
  - Implement using 2:1 multiplexer

a=0

b	f(a,b,0,0)	f(a,b,0,1)	f(a,b,1,0)	f(a,b,1,1)	D(c,d)
0	0	1	0	1	D0 =d
1	1	1	0	0	D1 =c’

a=1

b	f(a,b,0,0)	f(a,b,0,1)	f(a,b,1,0)	f(a,b,1,1)	D(c,d)
0	x	0	1	0	D2 =d’
1	x	0	0	0	D3 =0



Thank You !

- Remember:
  - Homework due on Thursday 3/3/16 11:59 PM