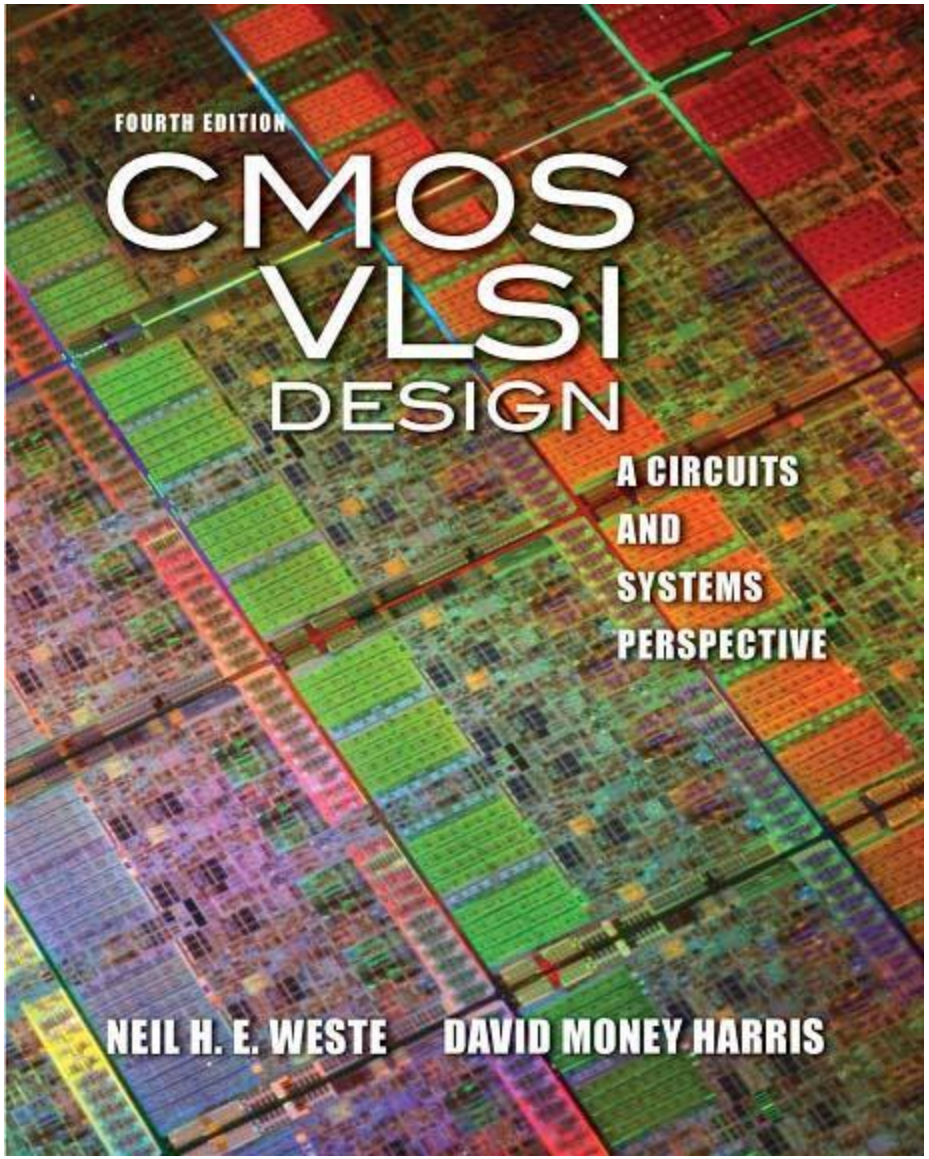


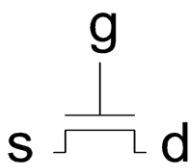
Lecture 4: CMOS Implementation of logic blocks and sequential elements

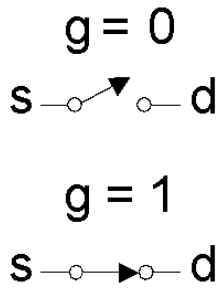
Signal Strength

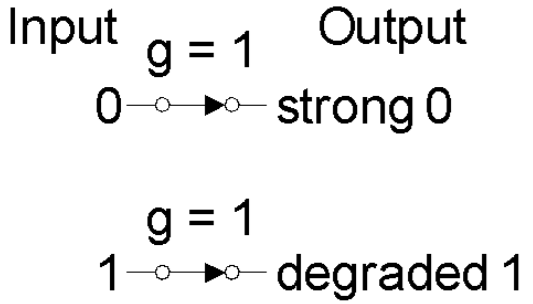
- *Strength* of signal
  - How close it approximates ideal voltage source
- V_{DD} and GND rails are strongest 1 and 0
- nMOS pass strong 0
  - But degraded or weak 1
- pMOS pass strong 1
  - But degraded or weak 0
- Thus **nMOS are best for pull-down network**
- And, **pMOS are best for pull-up network**

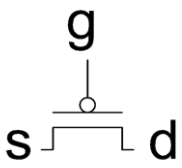
Pass Transistors

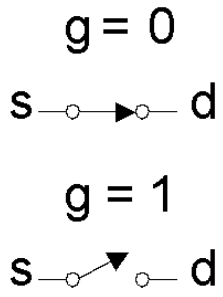
- Transistors can be used as switches

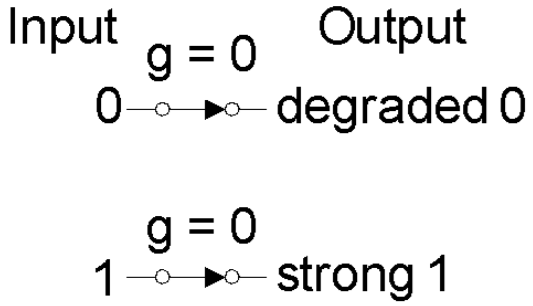






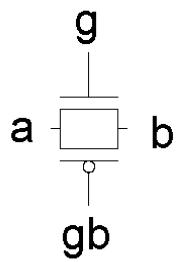


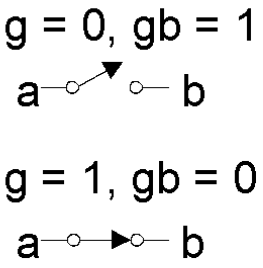


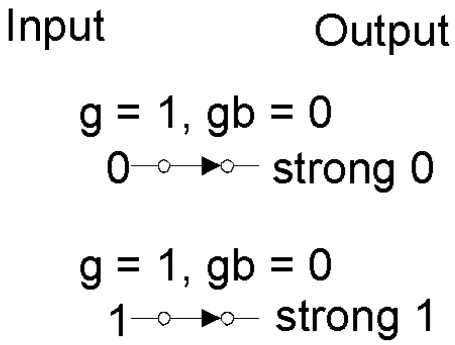


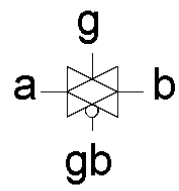
Transmission Gates

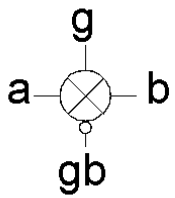
- Pass transistors produce degraded outputs
- *Transmission gates* pass both 0 and 1 well

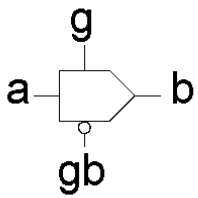








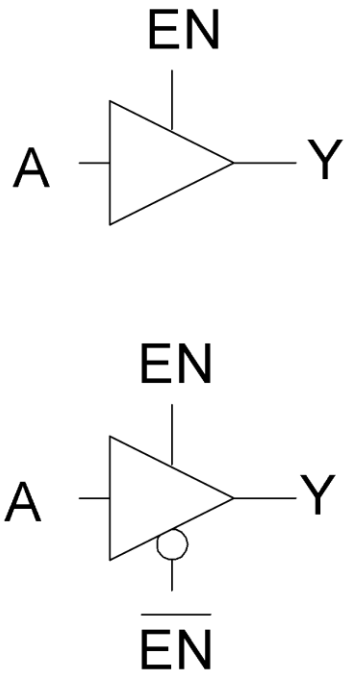




Tristates

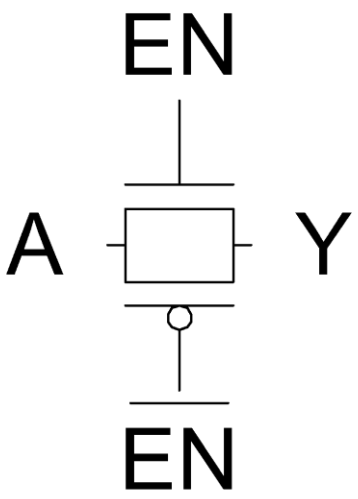
- *Tristate buffer* produces Z when not enabled

EN	A	Y
0	0	
0	1	
1	0	
1	1	



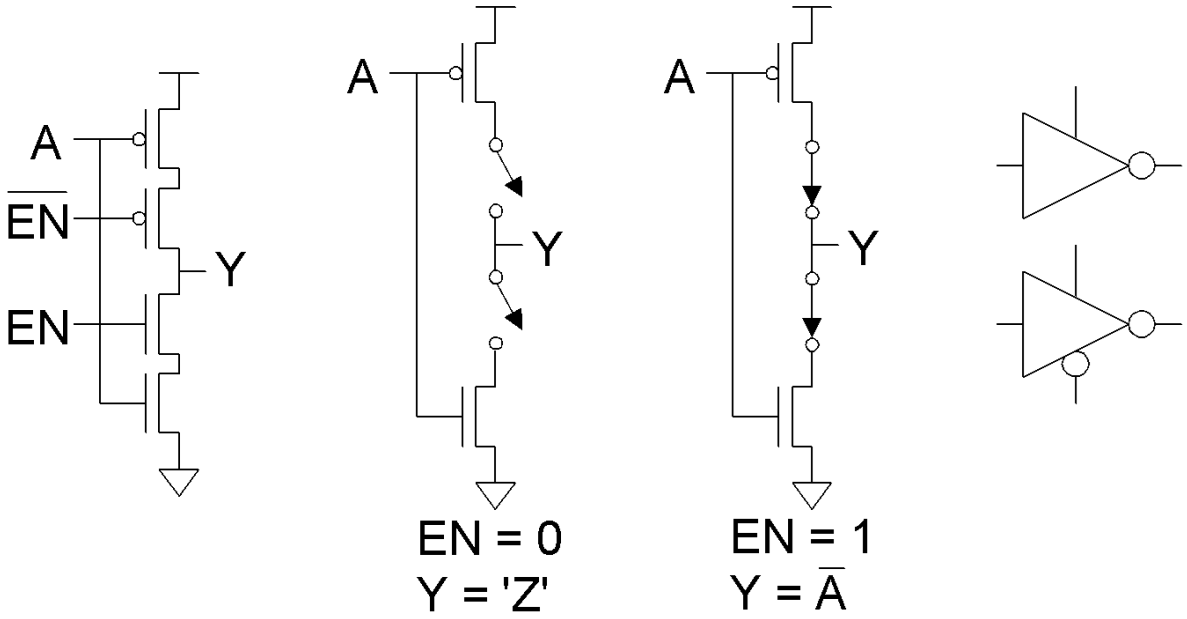
Nonrestoring Tristate

- Transmission gate acts as tristate buffer
  - Only two transistors
  - But *nonrestoring*
    - Noise on A is passed on to Y



Tristate Inverter

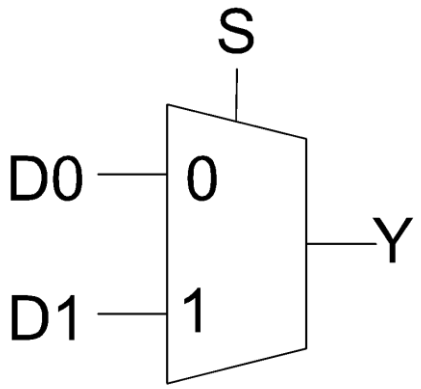
- Tristate inverter produces restored output
  - Violates conduction complement rule
  - Because we want a Z output



Multiplexers

- 2:1 multiplexer chooses between two inputs

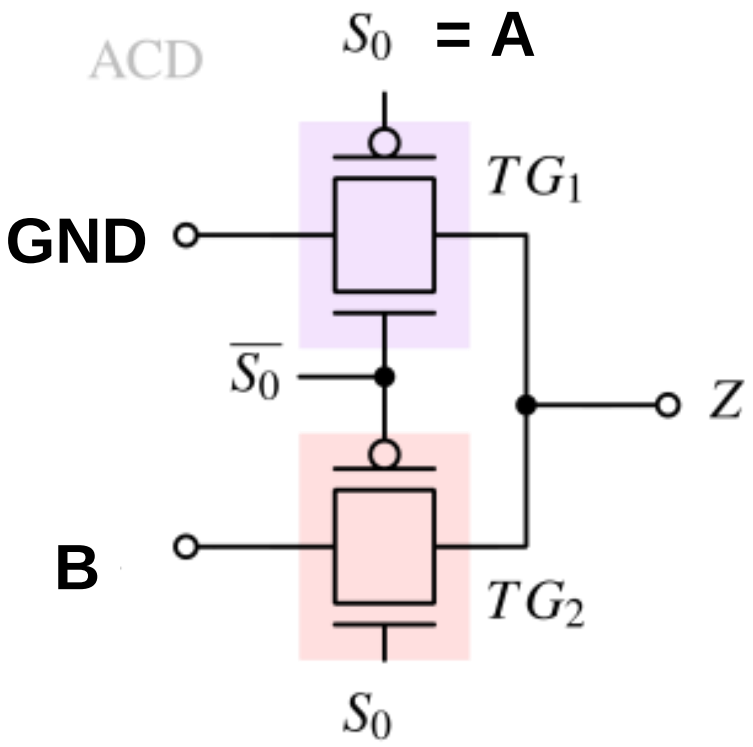
S	D1	D0	Y
0	X	0	
0	X	1	
1	0	X	
1	1	X	



Gate-Level Mux Design

- $Y = SD_1 + \overline{S}D_0$ (too many transistors)
- How many transistors are needed?

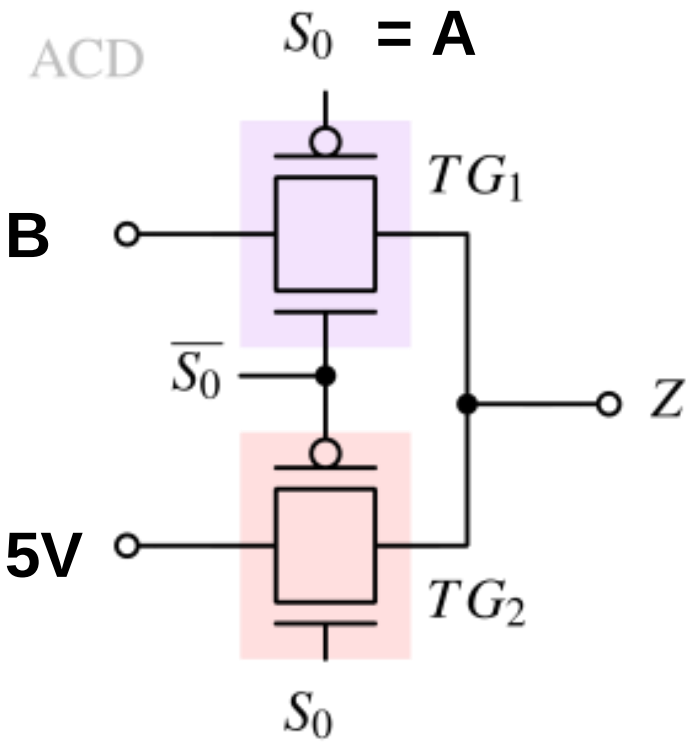
Logic Gate using TG

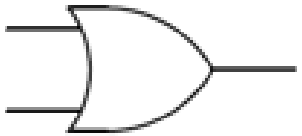


AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Logic Gate using TG

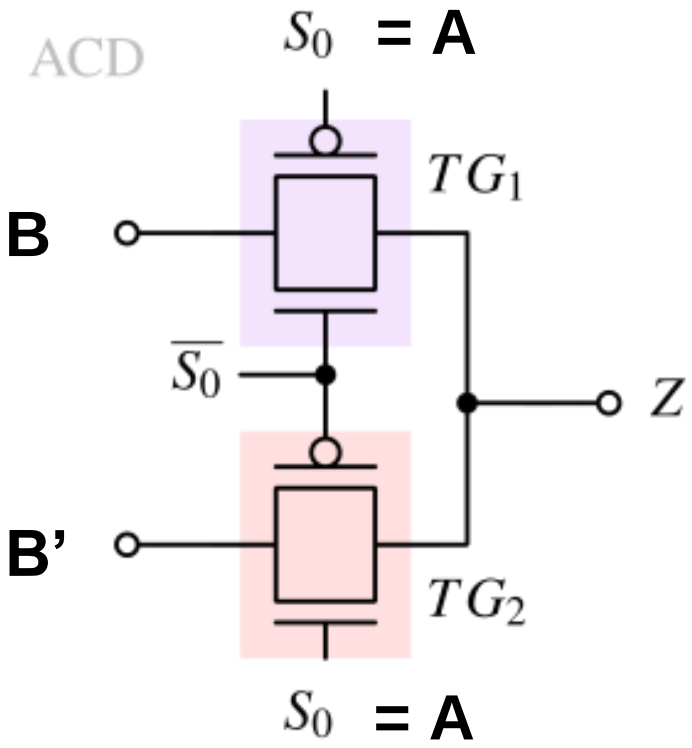


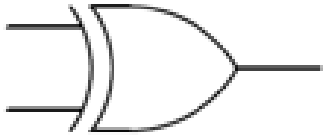


OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Logic Gate using TG



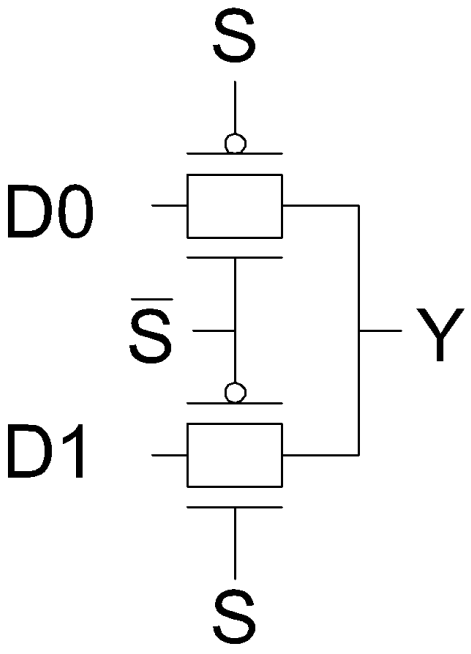


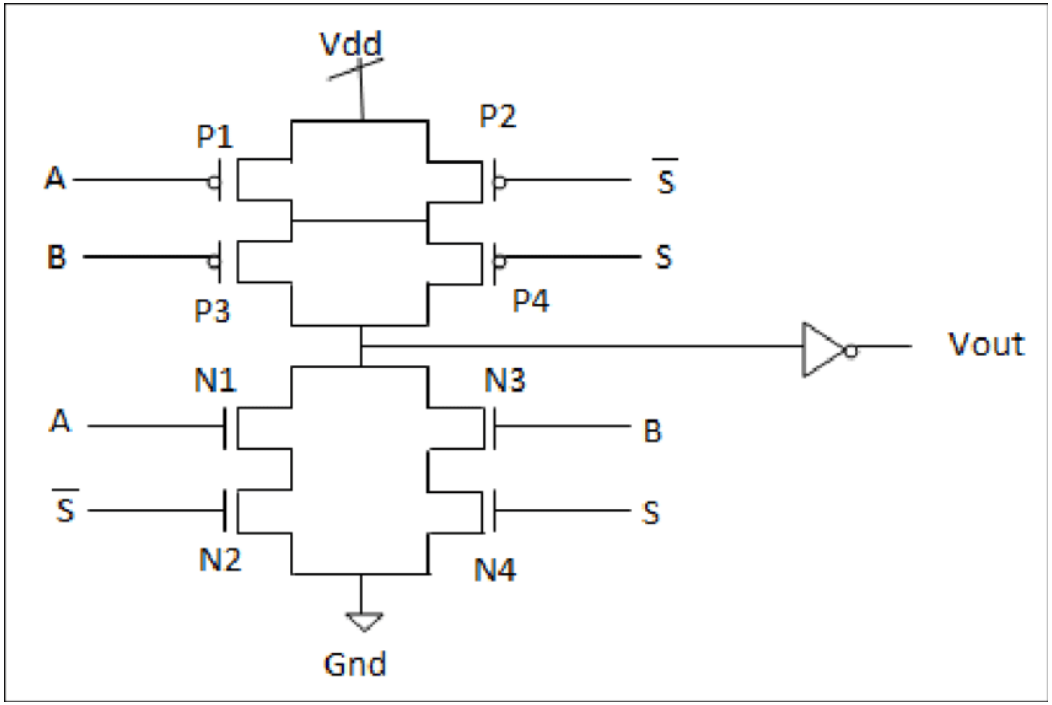
XOR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

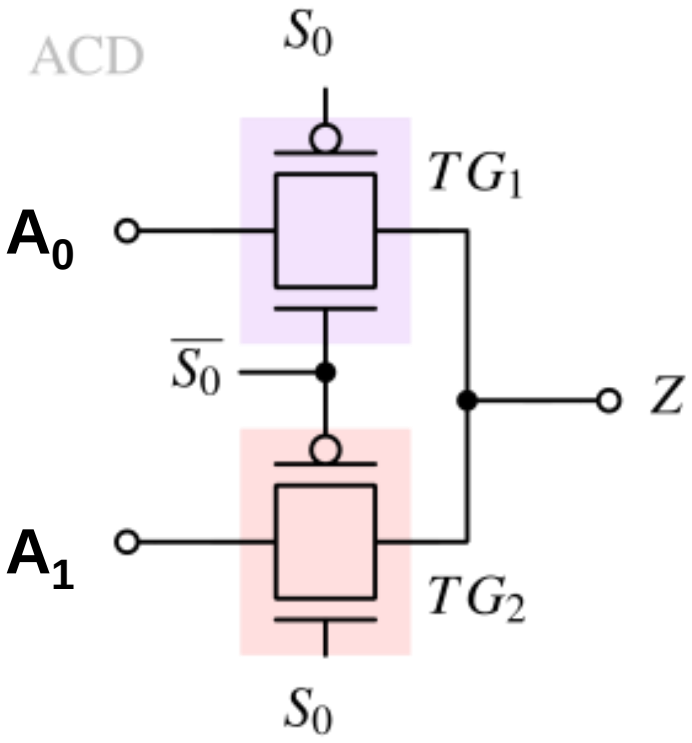
Transmission Gate Mux

- Nonrestoring mux uses two transmission gates
  - Only 4 transistors

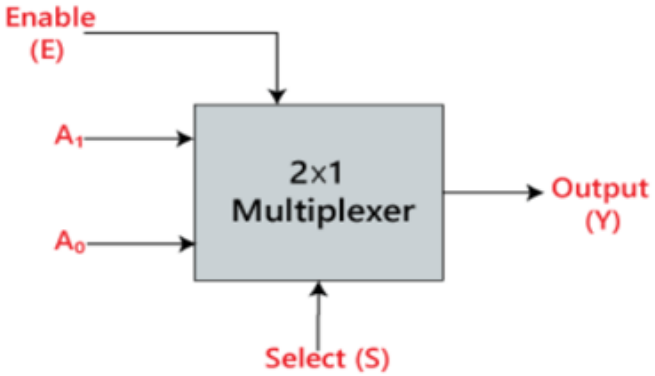




2:1 MUX using TG



Block Diagram:

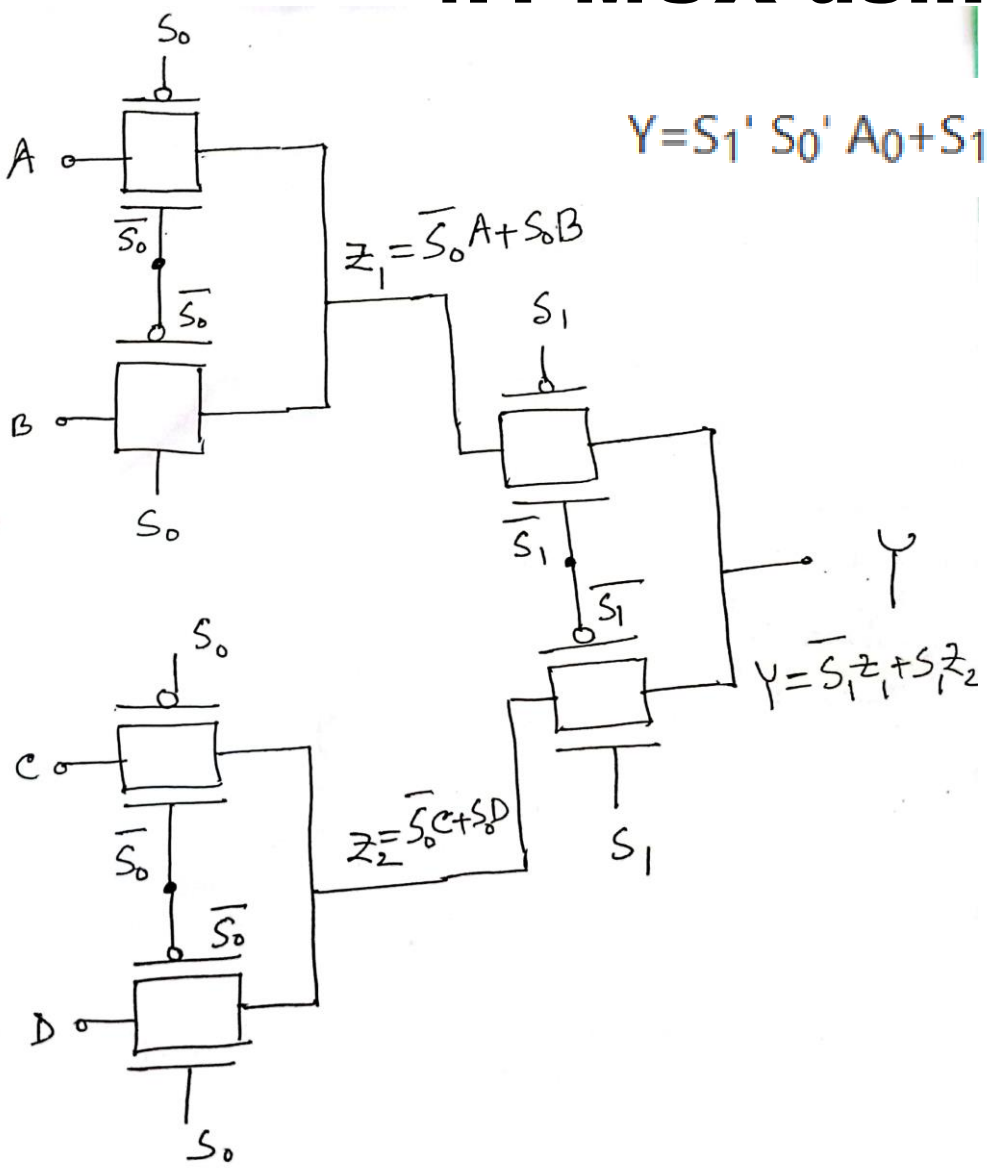


INPUTS	Output
S_0	Y
0	A_0
1	A_1

The logical expression of the term Y is as follows:

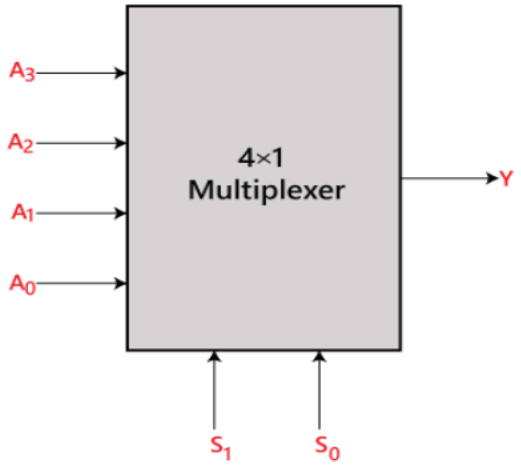
$Y=S_0'.A_0+S_0.A_1$

4:1 MUX using TG



$$Y = S_1' S_0' A_0 + S_1' S_0 A_1 + S_1 S_0' A_2 + S_1 S_0 A_3$$

Block Diagram:

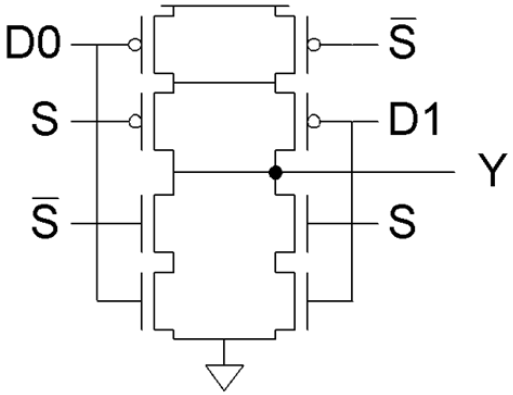


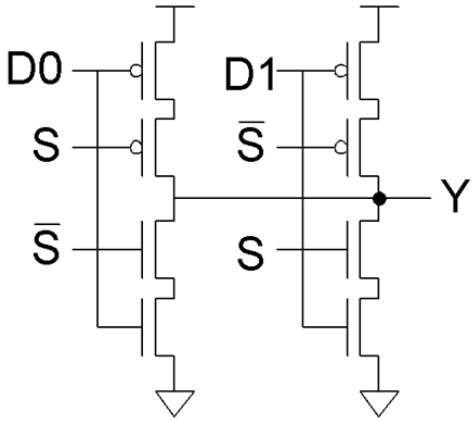
Truth Table:

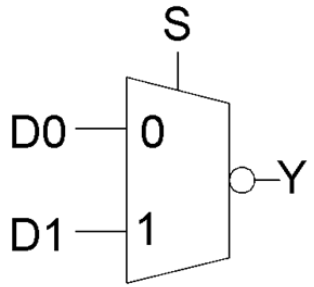
INPUTS		Output
S_1	S_0	Y
0	0	A_0
0	1	A_1
1	0	A_2
1	1	A_3

Inverting Mux

- Inverting multiplexer
  - Use compound AOI22
  - Or pair of tristate inverters
  - Essentially the same thing
- Noninverting multiplexer adds an inverter

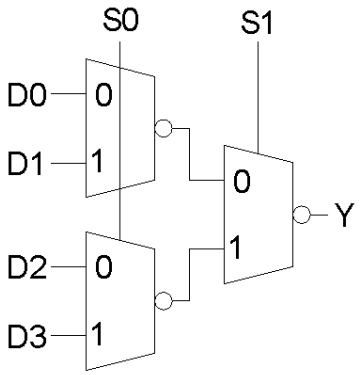


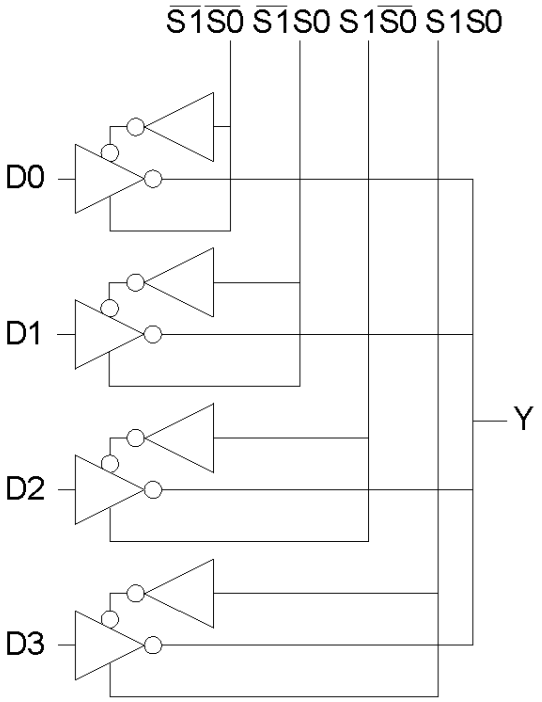




4:1 Multiplexer

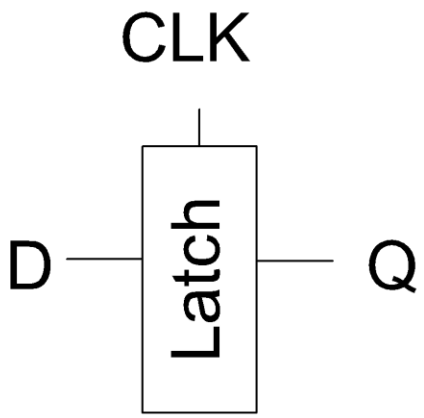
- 4:1 mux chooses one of 4 inputs using two selects
  - Two levels of 2:1 muxes
  - Or four tristates

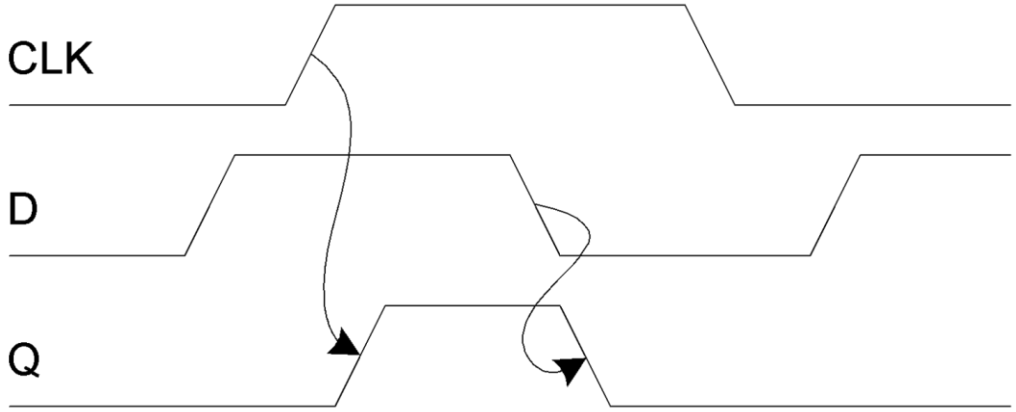




D Latch

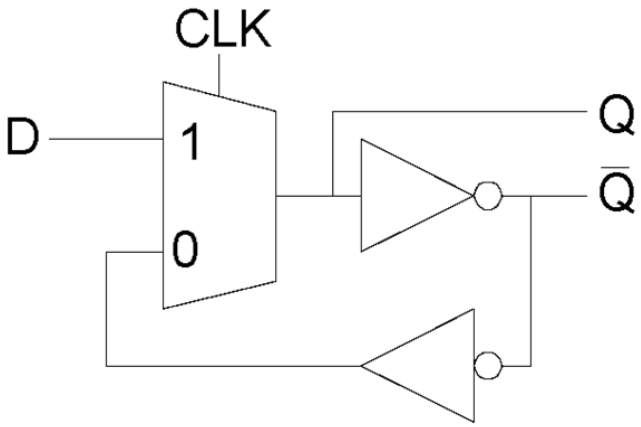
- When CLK = 1, latch is *transparent*
  - D flows through to Q like a buffer
- When CLK = 0, the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*

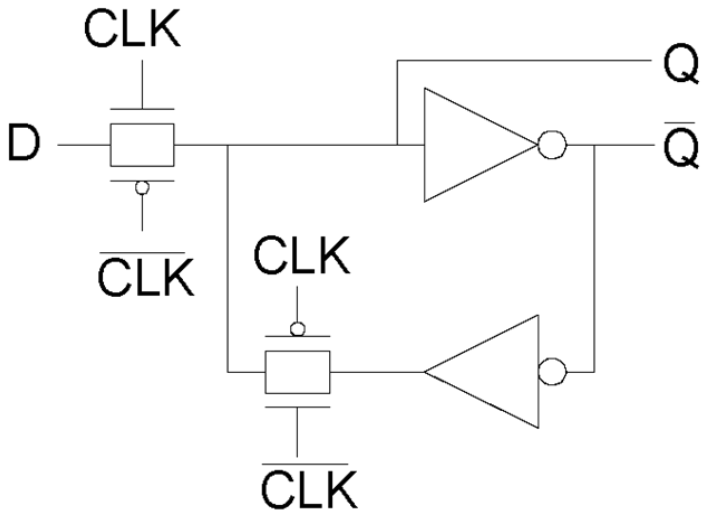




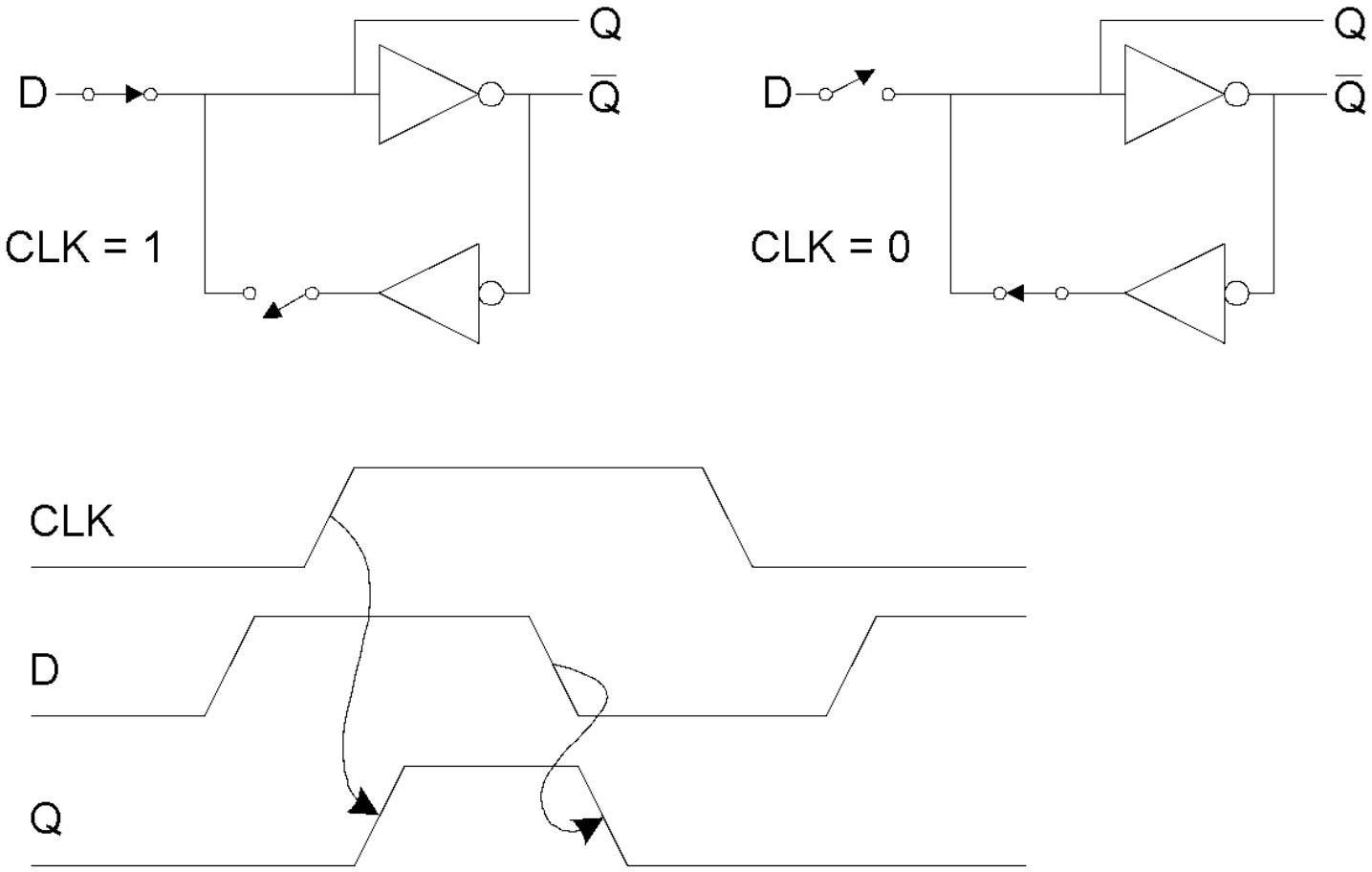
D Latch Design

- Multiplexer chooses D or old Q



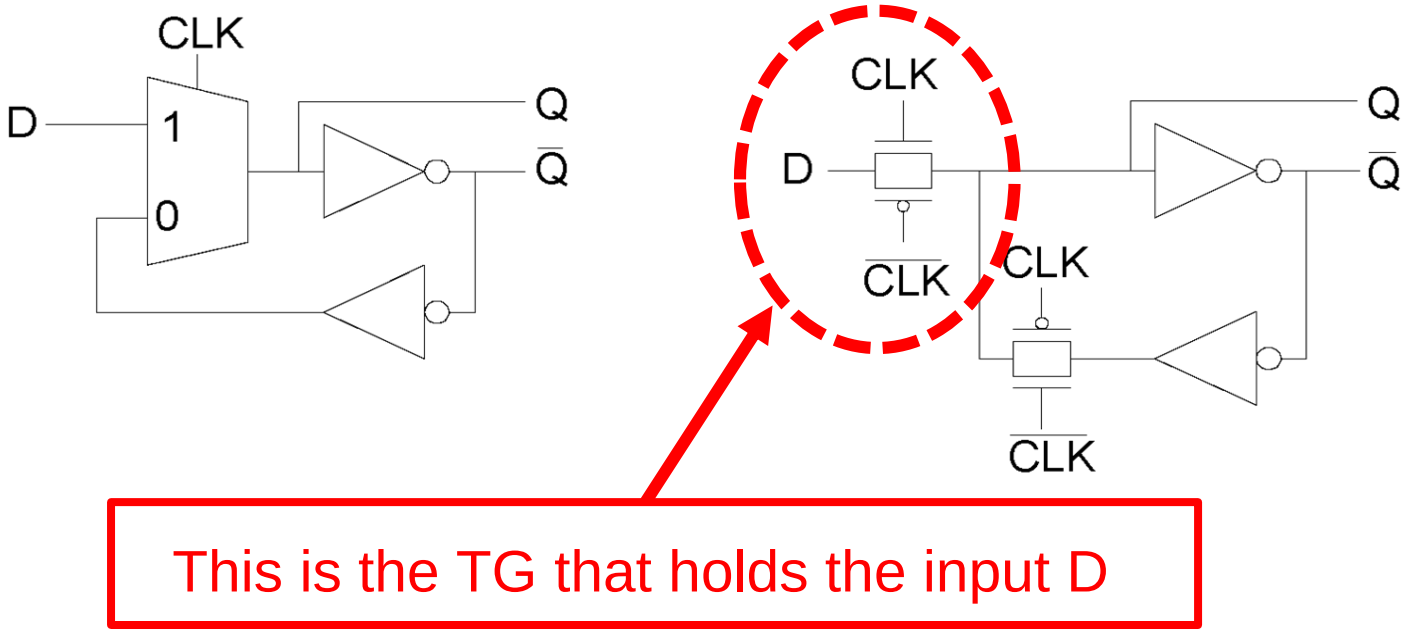


D Latch Operation

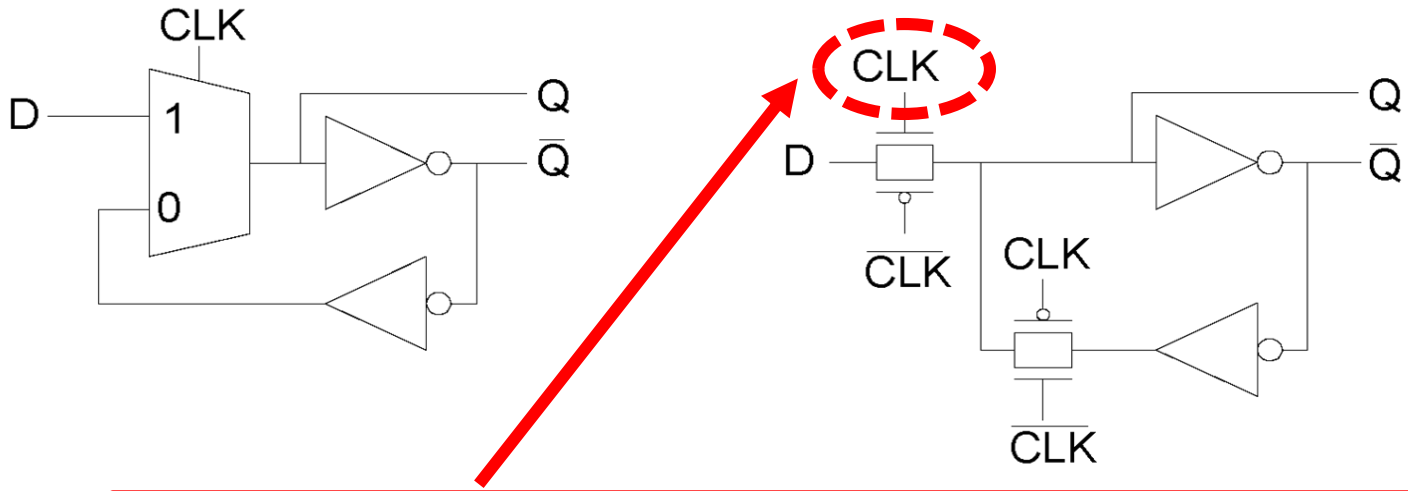


How to identify the Latch

Find out the TG which holds the input of the latch. Now look at the gate of the NMOS of that TG. If gate of the NMOS of that TG have CLK as it's input. Then the latch is a positive latch. Otherwise it's a negative latch.



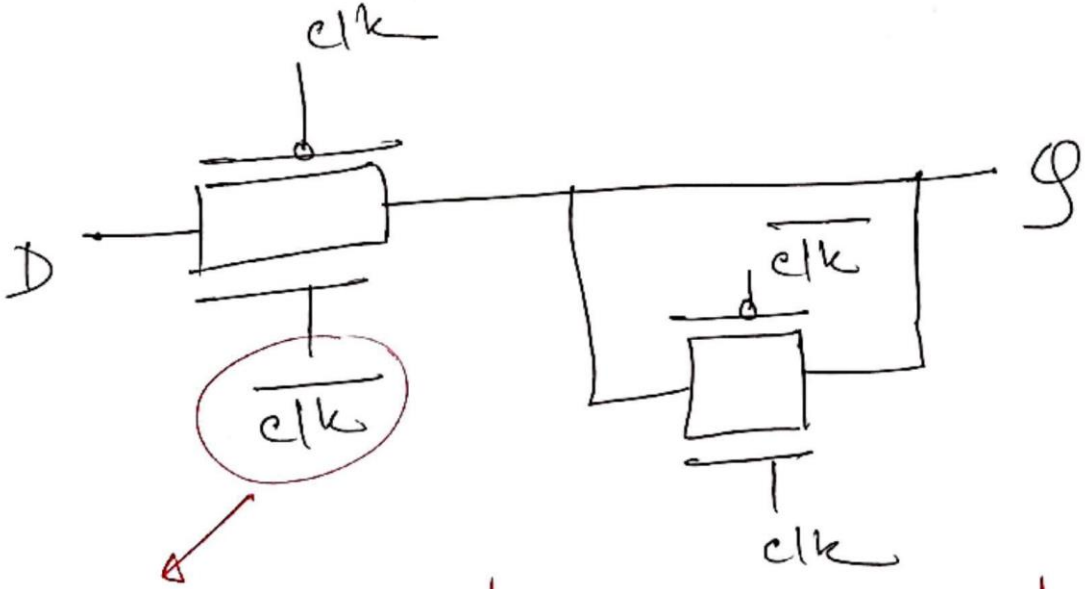
How to identify the Latch



Look at the input of the NMOS gate of the TG that holds input of the latch

As the gate of this NMOS have CLK, it's a positive latch

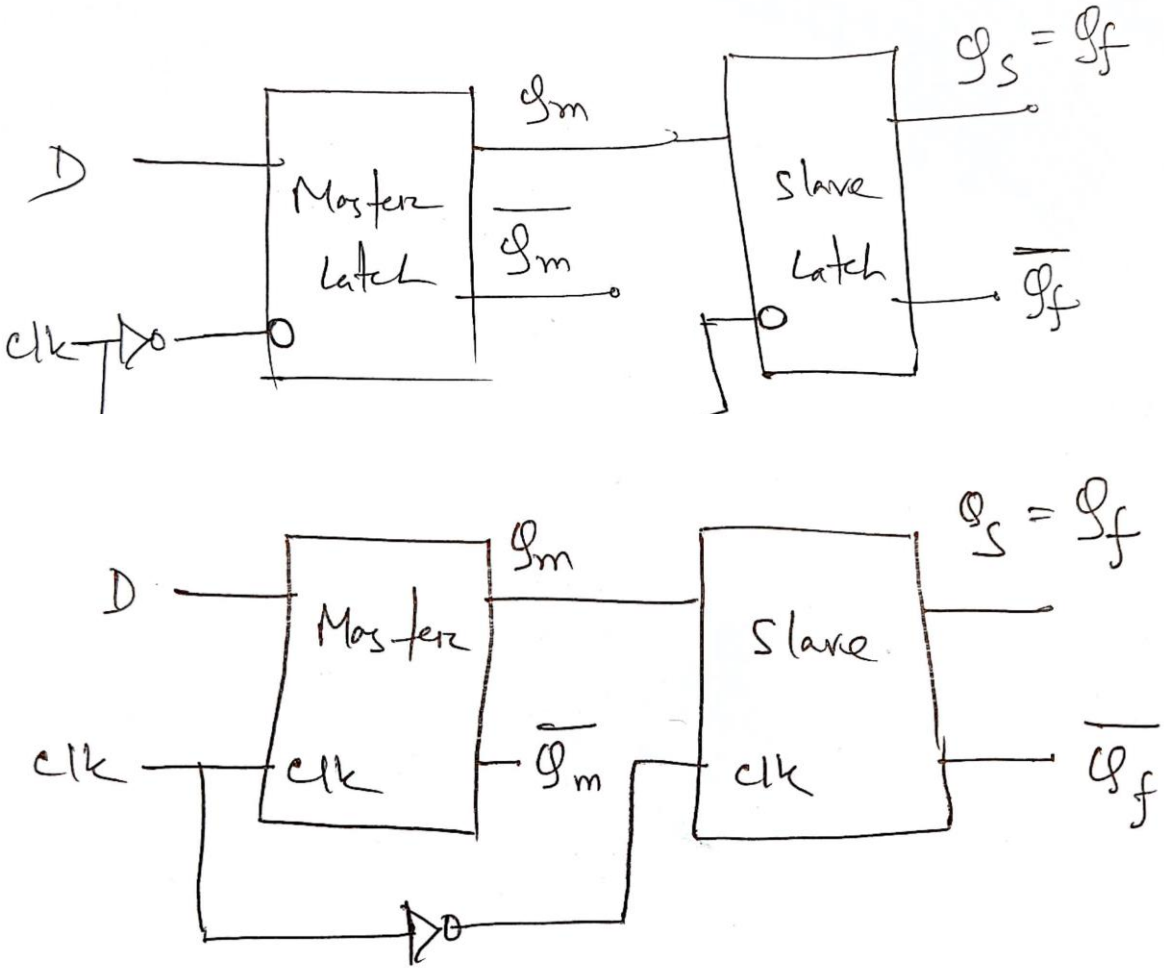
How to identify the Latch



This clock is ~~negative~~ inverted. That's why, it's a negative latch.

D Flip-flop Operation

Negative Edge Triggered D-FF:



D Flip-flop Operation

Negative Edge Triggered D-FF:

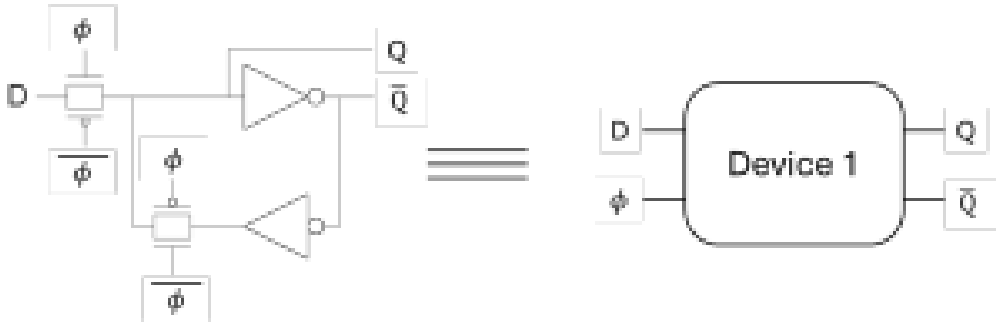


Figure 1: Circuit on the left is represented by the block - **Device 1**

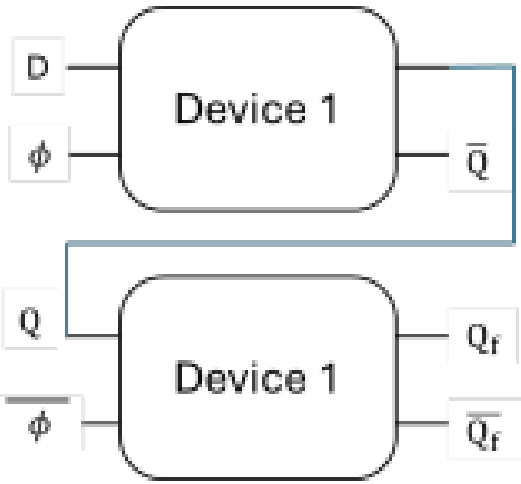


Figure 2: A circuit made up of two **Device 1** blocks **from Fig. 1**

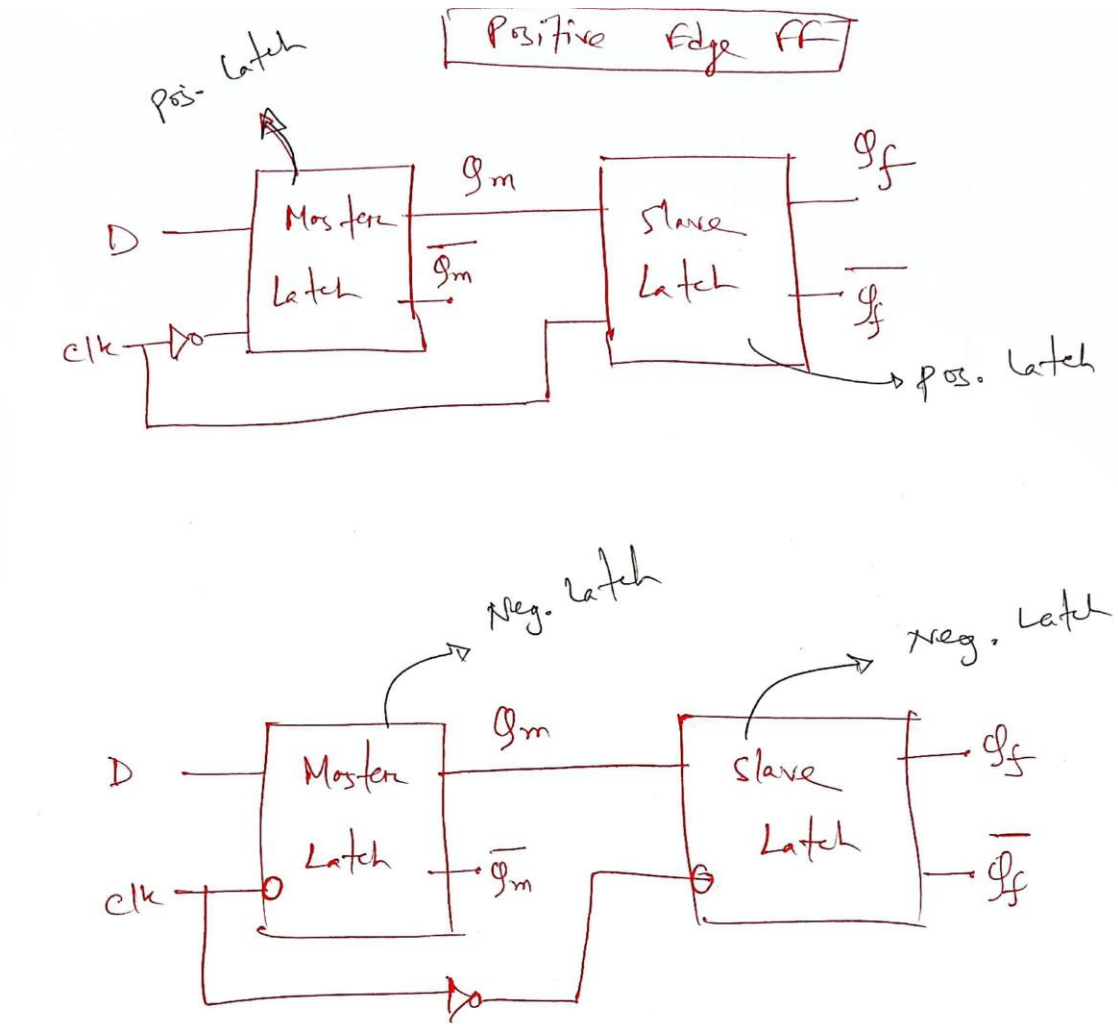
(d) **Q=D → *Positive level triggered Latch***

(e)

Node ↓ / Time →	5 ns	10 ns	15 ns	20 ns	25 ns
ϕ	0	1	0	1	0
D	0	1	1	1	0
Q	X/1	1	1	1	1
Q_f	1	1	1	1	1

Negative-edge-triggered flip-flop.

Positive Edge Triggered D-FF:



Positive Edge Triggered D-FF:

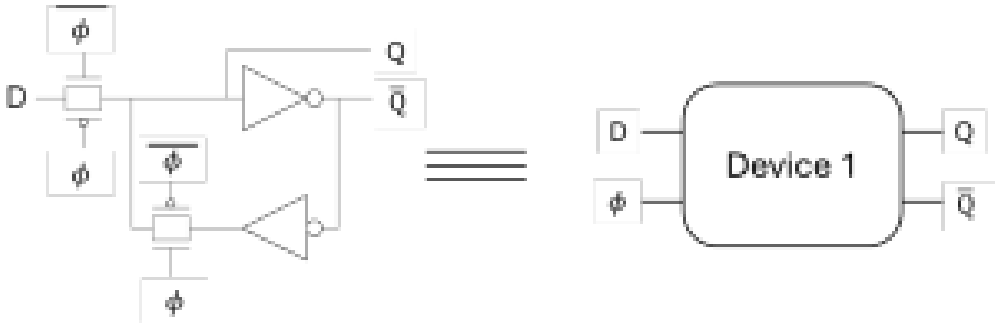


Figure 1: Circuit on the left is represented by the block - **Device 1**

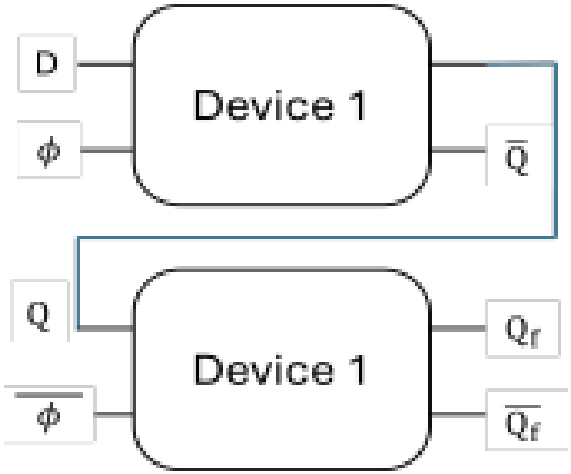


Figure 2: A circuit made up of two **Device 1** blocks **from Fig. 1**

(d) **Q=D → *Negative level triggered Latch***

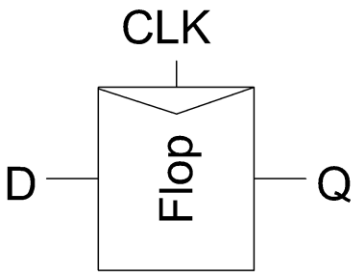
(e)

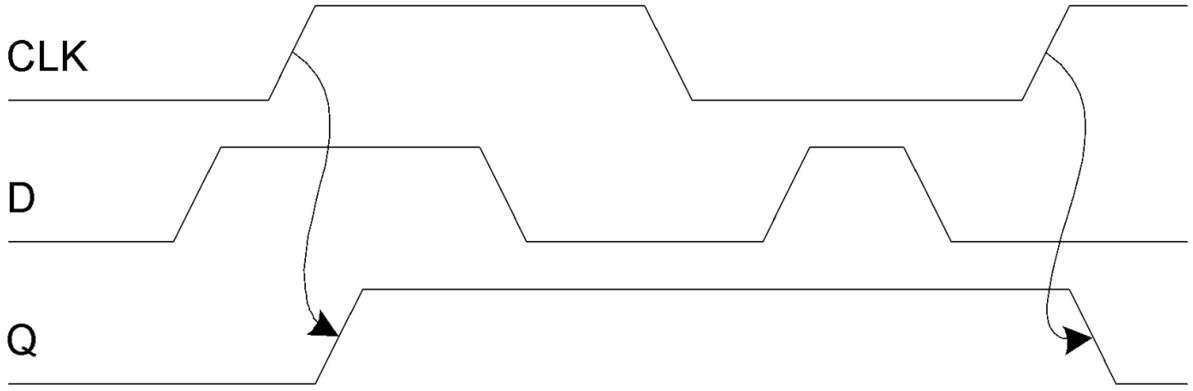
Node ↓ / Time →	5 ns	10 ns	15 ns	20 ns	25 ns
ϕ	0	1	0	1	0
D	0	1	1	1	0
Q	0	0	1	1	0
Q_f	1	0	0	1	1

Positive-edge triggered flip-flop.

D Flip-flop

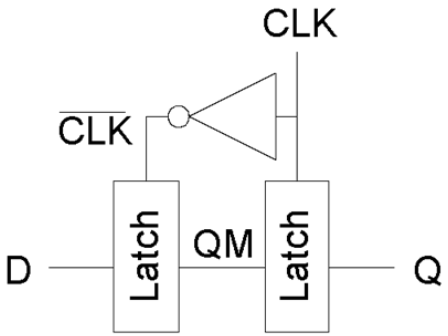
- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop*, *master-slave flip-flop*

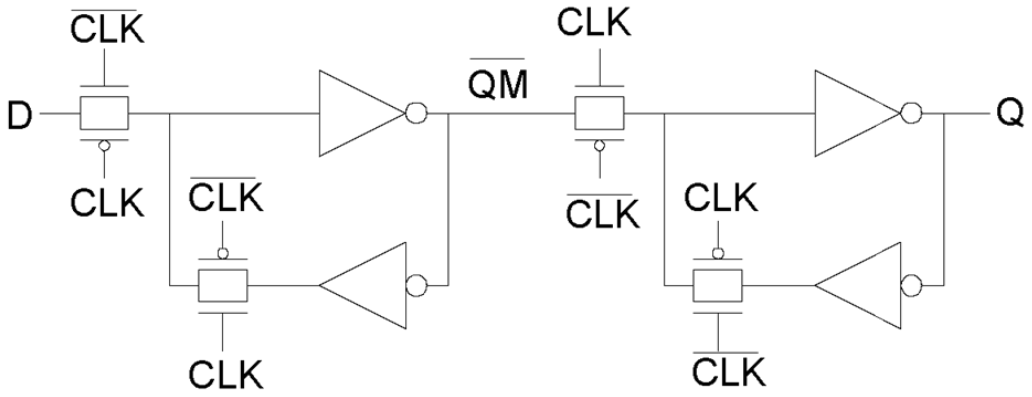




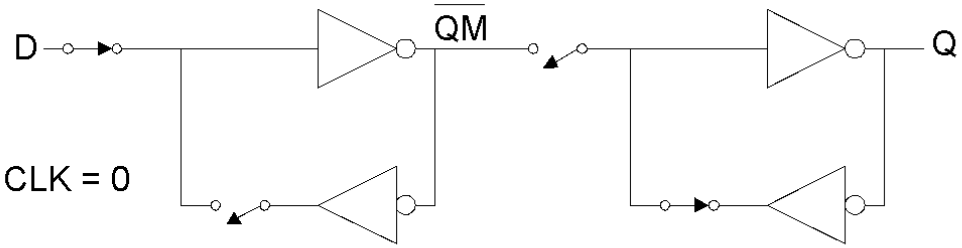
D Flip-flop Design

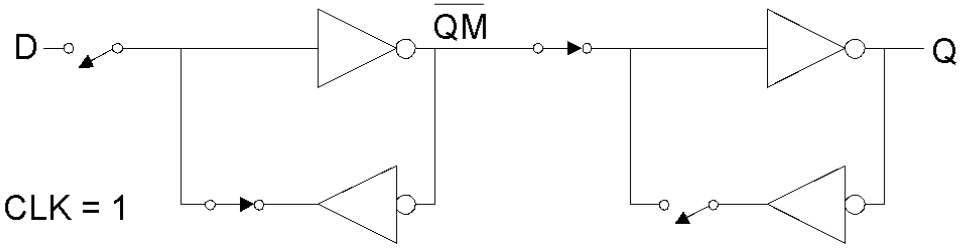
- Built from master and slave D latches

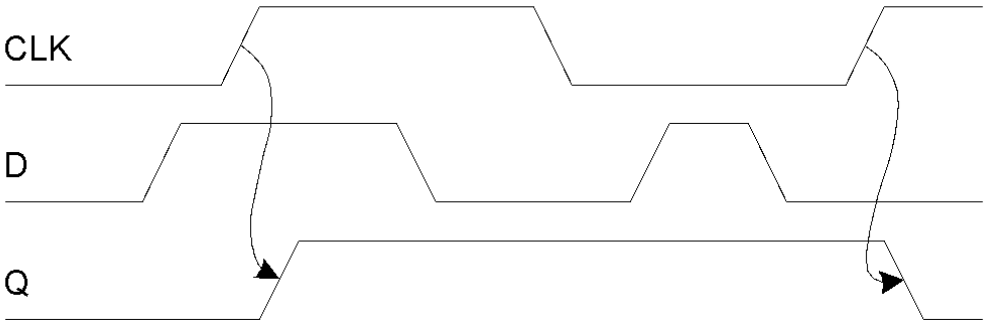




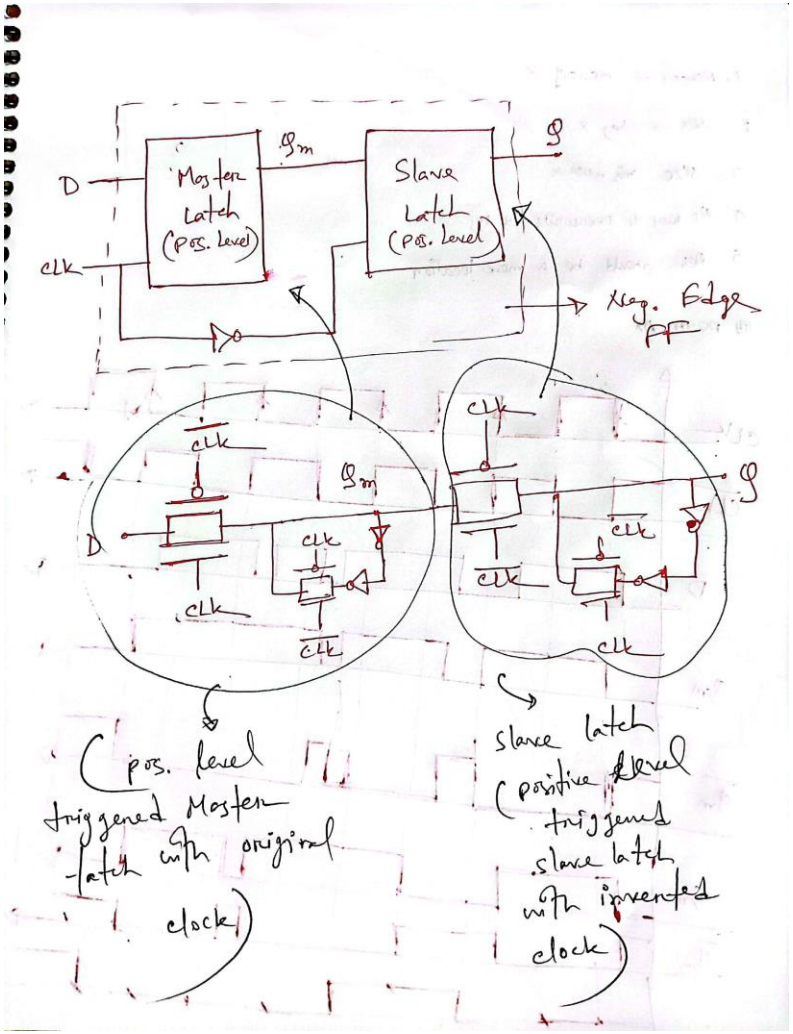
D Flip-flop Operation

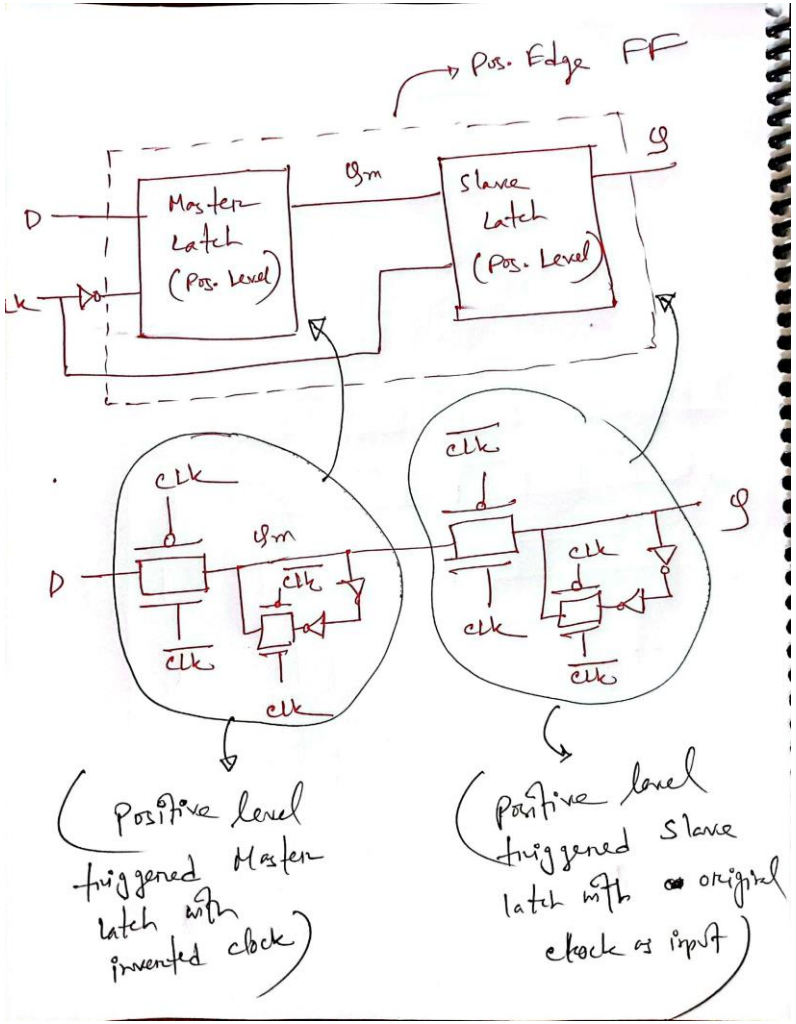




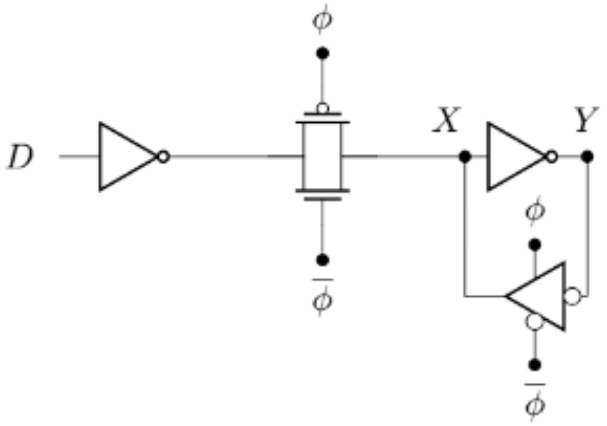


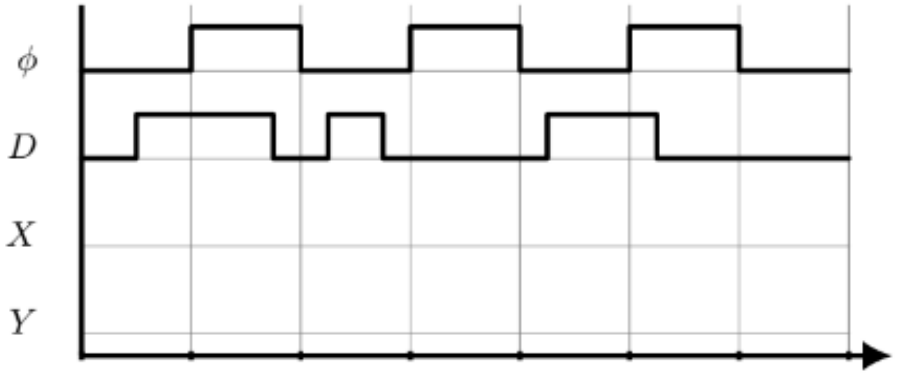
D Flip-flop Operation

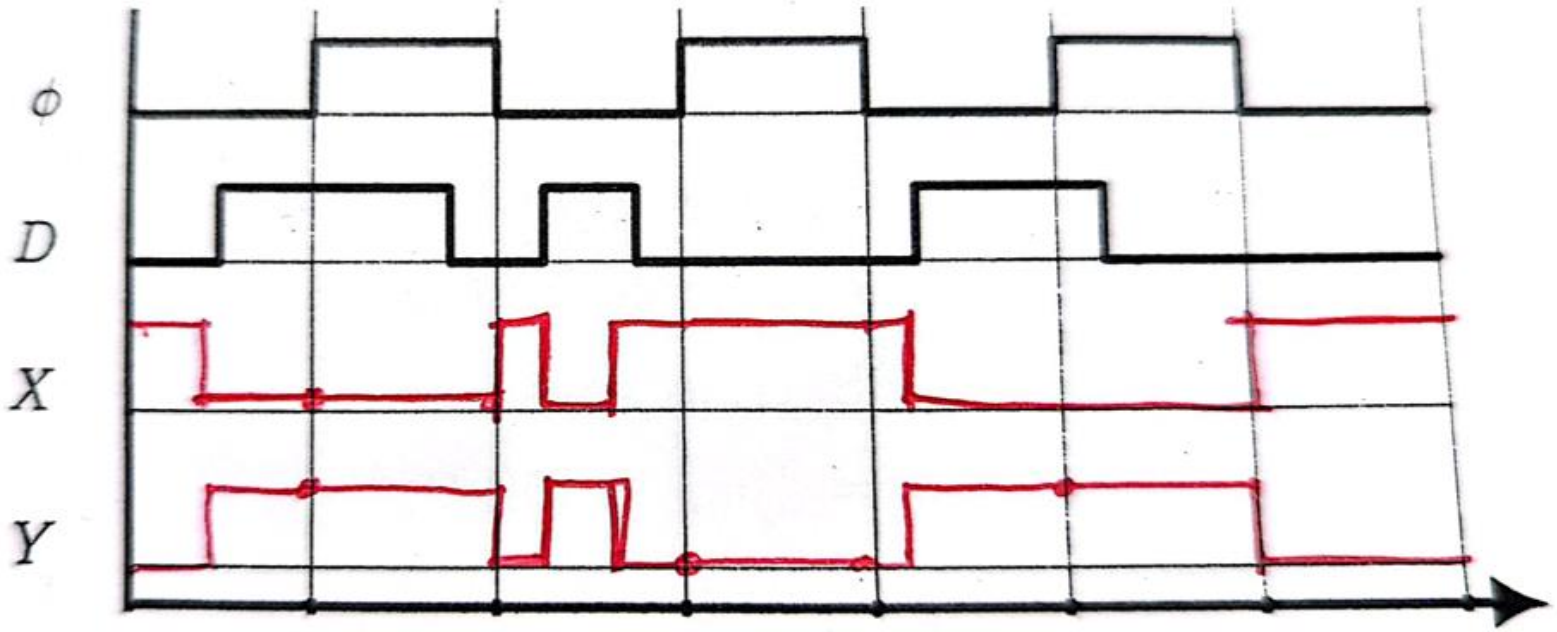




[**3 marks**] **Complete** the timing diagram shown below on the right, in the question paper, based on the circuit diagram presented below on the left. Ensure that your completed timing diagram accurately reflects the logic behavior of the circuit, taking into account the states of ϕ and D to determine the corresponding outputs X and Y.






ϕ
D
X
Y