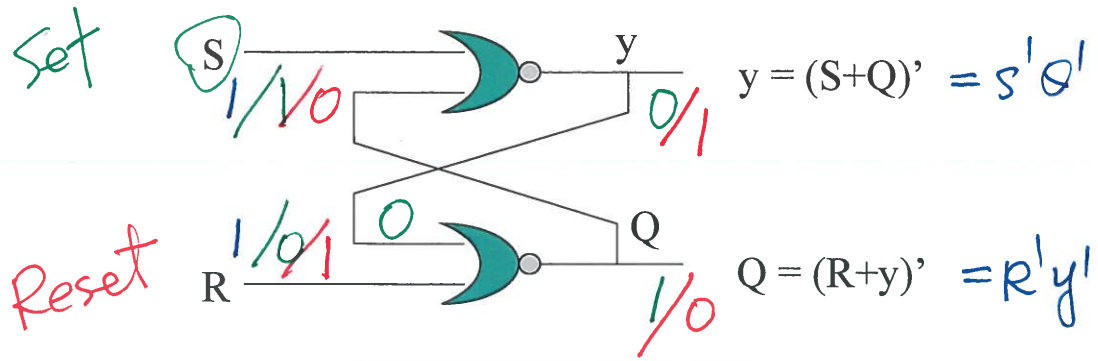


2. Basic Building Blocks: SR Latch



Inputs: S, R

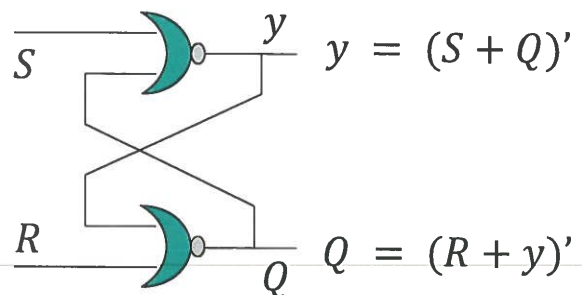
State: (Q, y)

17

id	S	R	Qt	yt	Qt	yt	Qt	yt	Qt	yt
0	0	0	0	0	1	1	0	0	1	1
1	0	0	0	1	0	1	0	1	0	1
2	0	0	1	0	1	0	1	0	1	0
3	0	0	1	1	0	0	1	1	0	0
4	0	1	0	0	0	1	0	1	0	1
5	0	1	0	1	0	1	0	1	0	1
6	0	1	1	0	0	0	0	1	0	1
7	0	1	1	1	0	0	0	1	0	1
8	1	0	0	0	1	0	1	0	1	0
9	1	0	0	1	0	0	1	0	1	0
10	1	0	1	0	1	0	1	0	1	0
11	1	0	1	1	0	0	1	0	1	0
12	1	1	0	0	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0	0	0
14	1	1	1	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0

A B

Truth table of SR latch
with incremental steps in
time



18

id	S	R	Qt	yt	Qt	yt	Qt	yt	Qt	yt
0	0	0	0	0	1	1	0	0	1	1
1	0	0	0	1	0	1	0	1	0	1
2	0	0	1	0	1	0	1	0	1	0
3	0	0	1	1	0	0	1	1	0	0
4	0	1	0	0	0	1	0	1	0	1
5	0	1	0	1	0	1	0	1	0	1
6	0	1	1	0	0	0	0	1	0	1
7	0	1	1	1	0	0	0	1	0	1
8	1	0	0	0	1	0	1	0	1	0
9	1	0	0	1	0	0	1	0	1	0
10	1	0	1	0	1	0	1	0	1	0
11	1	0	1	1	0	0	1	0	1	0
12	1	1	0	0	0	0	0	0	0	0
13	1	1	0	1	0	0	0	0	0	0
14	1	1	1	0	0	0	0	0	0	0
15	1	1	1	1	0	0	0	0	0	0

"State Table" of SR latch

input ↙

\SR	00	01	10	11
Qy				
00	11	01	10	00
01	01	01	10	00
10	10	10	01	10
11	00	01	10	00

Present
state

Next
state

19

CASES:

SR=01: (Q,y) = (0,1)

SR=10: (Q,y) = (1,0)

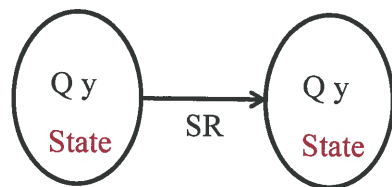
SR=11: (Q,y) = (0,0)

SR= 00: (Q,y) does not change if (Q,y)=(1,0) or (0,1)

However, when (Q,y) = (0,0) or (1,1), the output keeps changing

Remark: To verify the design, we need to enumerate all combinations.

State Table and State Diagram

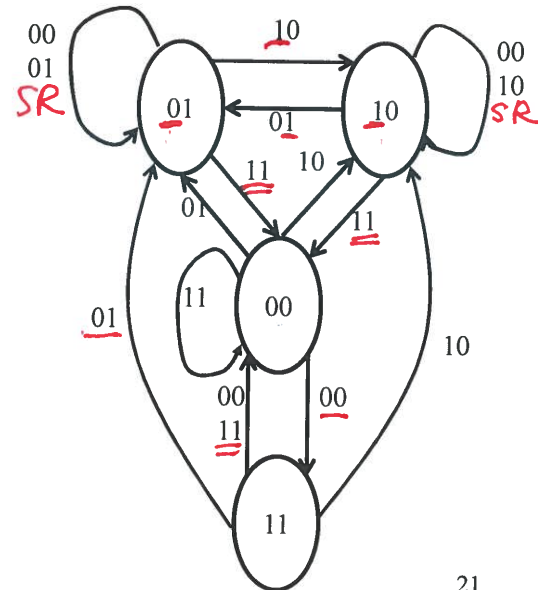


Present State Transition Next State

State diagram

State Table

\SR Qy	00	01	10	11
00	11	01	10	00
01	01	01	10	00
10	10	01	10	00
11	00	01	10	00



21

CASES:

SR=01: $(Q,y) = (0,1)$

SR=10: $(Q,y) = (1,0)$

SR=11: $(Q,y) = (0,0)$

SR= 00: (Q,y) does not change if $(Q,y)=(1,0)$ or $(0,1)$

However, when $(Q,y) = (0,0)$ or $(1,1)$, the output keeps changing

Q. Suppose that we can set the initial state $(Q,y)=(0,1)$. To avoid the SR latch output from toggling or behaving in an undefined way which input combinations should be avoided:

- A. $(S, R) = (0, 0)$
- B. $(S, R) = (1, 1)$
- C. None of the above

CASES:

SR=01: $(Q,y) = (0,1)$

SR=10: $(Q,y) = (1,0)$

SR=11: $(Q,y) = (0,0)$

SR= 00: (Q,y) does not change if $(Q,y)=(1,0)$ or $(0,1)$

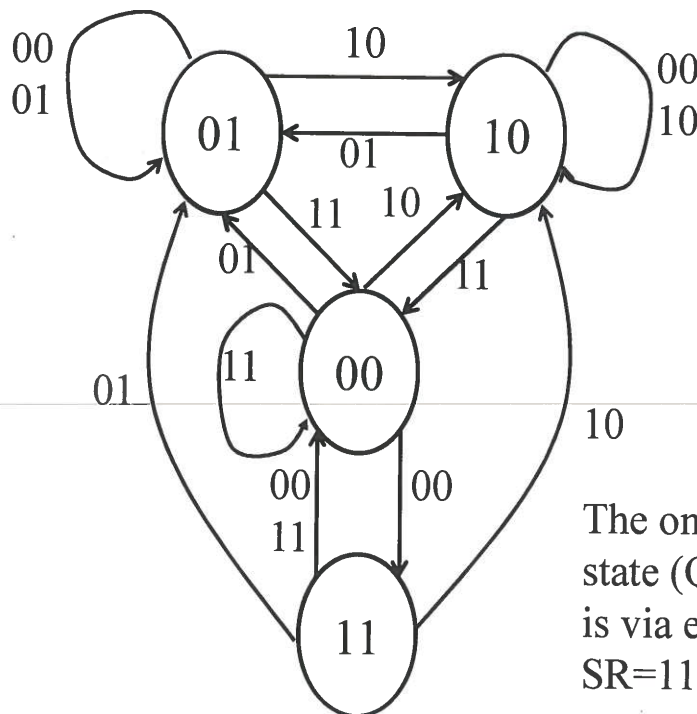
However, when $(Q,y) = (0,0)$ or $(1,1)$, the output keeps changing

We set the initial state $(Q,y)=(0,1)$ or $(1,0)$. To avoid the state $(Q,y)= (0,0)$ or $(1,1)$, we block the input SR=11.

Thus, without input SR=11, the state can only be $(Q,y)=(0,1)$ or $(1,0)$.

23

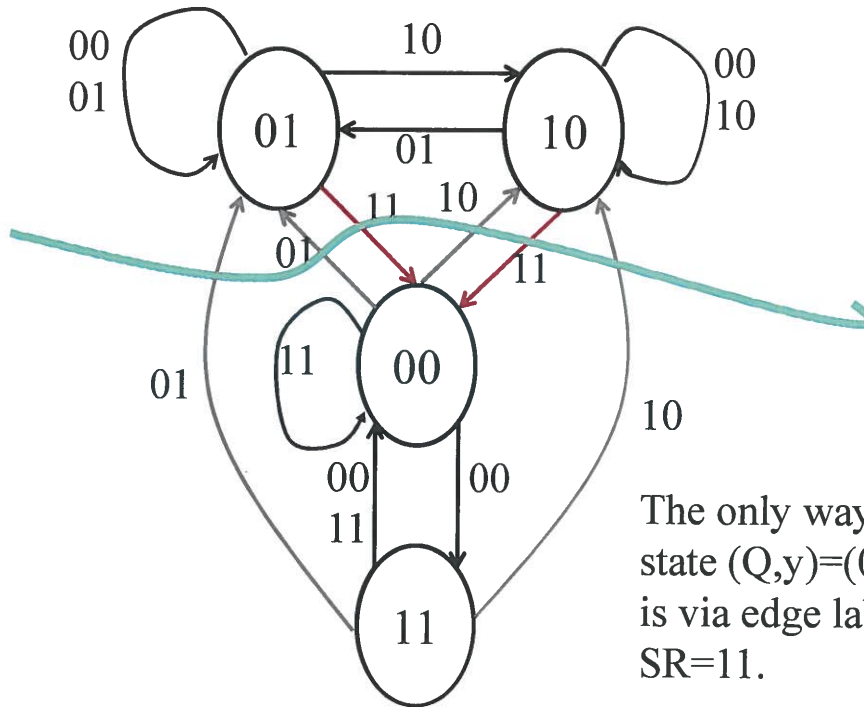
SR Latch



The only way to reach state $(Q,y)=(0,0)$ or $(1,1)$ is via edge labeled SR=11.

24

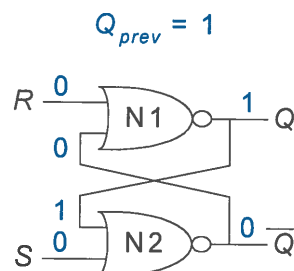
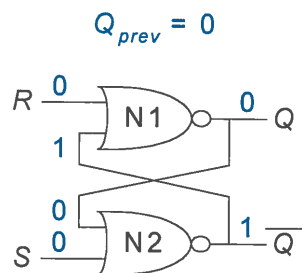
SR Latch



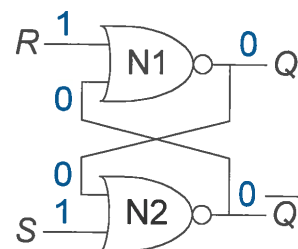
25

2. Basic Building Blocks: SR Latch Analysis

- $S = 0, R = 0$: then $Q = Q_{prev}$ and $\bar{Q} = \bar{Q}_{prev}$ (**memory!**)



- $S = 1, R = 1$: then $Q = 0$ and $\bar{Q} = 0$ (**invalid state: $Q \neq \text{NOT } \bar{Q}$**)



26

2. Basic Building Blocks: SR Latch

CASES

SR=01: $(Q,y) = (0,1)$

SR=10: $(Q,y) = (1,0)$

SR=11: $(Q,y) = (0,0)$

SR = 00: if $(Q,y) = (0,0)$ or $(1,1)$, the output keeps changing

Solutions: Avoid the case that $SR = (1,1)$.

State table

PS \ inputs		SR			
		00	01	10	11
<u>Q(t)</u>	0	0	0	1	-
	1	1	0	1	-

Characteristic Expression
 $Q(t+1) = S(t) + R'(t)Q(t)$

Handwritten diagram showing a 2x4 grid with inputs R and S, and outputs Q and $\bar{Q}$. The grid contains the values 0, 0, -, 1 in the top row and 1, 0, -, 1 in the bottom row. A red arrow points from the state table to this diagram.

Handwritten equation: $Q(t+1) = S(t) + R'(t)Q(t)$

Labels: Q(t+1) NS (next state) S

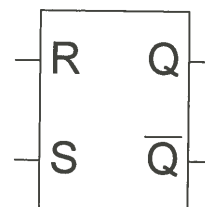
27

2. Basic Building Blocks: SR Latch Symbol

- SR stands for Set/Reset Latch
 - Stores one bit of state (Q)
- Control what value is being stored with S , R inputs
 - **Set:** Make the output 1 ($S = 1$, $R = 0$, $Q = 1$)
 - **Reset:** Make the output 0 ($S = 0$, $R = 1$, $Q = 0$)

- **Must do something to avoid invalid state (when $S = R = 1$)**

SR Latch Symbol

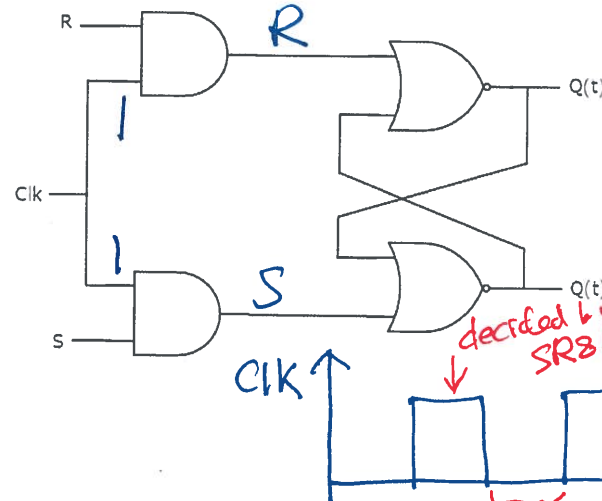
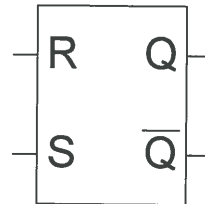


28

2. Basic Building Blocks: SR Latch Symbol



SR Latch
Symbol



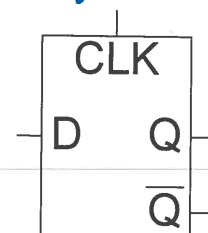
clk	SR	$Q(t+1)$
0	--	$Q(t+1)=Q(t)$
1	00	$Q(t+1)=Q(t)'$
1	01	0
1	10	1

2. Basic Building Blocks: D Latch

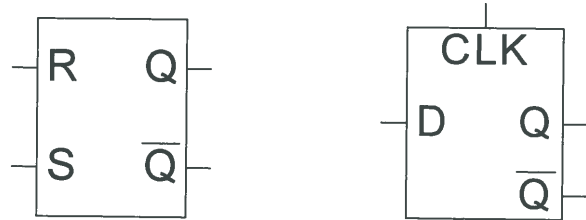
Data/Delay

- Two inputs: CLK , D
 - CLK : controls *when* the output changes
 - D (the data input): controls *what* the output changes to
- Function
 - When $CLK = 1$, D passes through to Q (the latch is *transparent*)
 - When $CLK = 0$, Q holds its previous value (the latch is *opaque*)
- Avoids invalid case when $Q \neq \text{NOT } \bar{Q}$

D Latch
Symbol

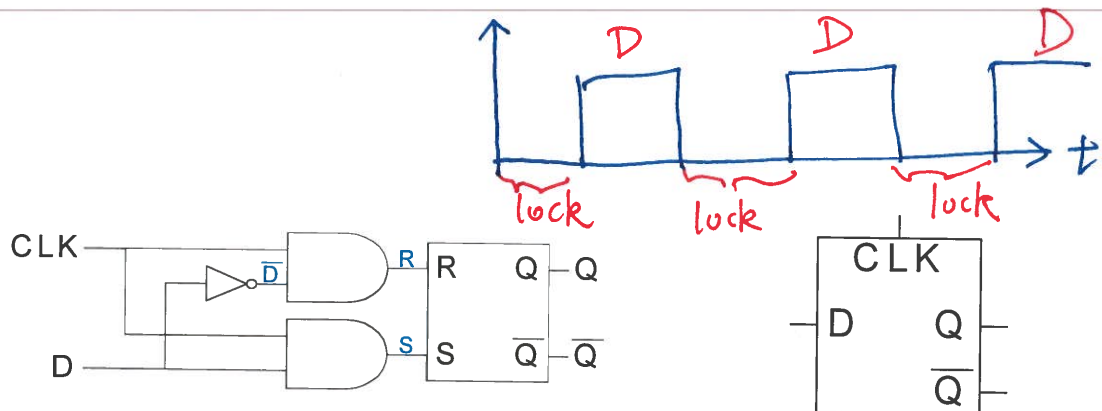


2. Basic Building Blocks: D Latch Internal Circuit



31

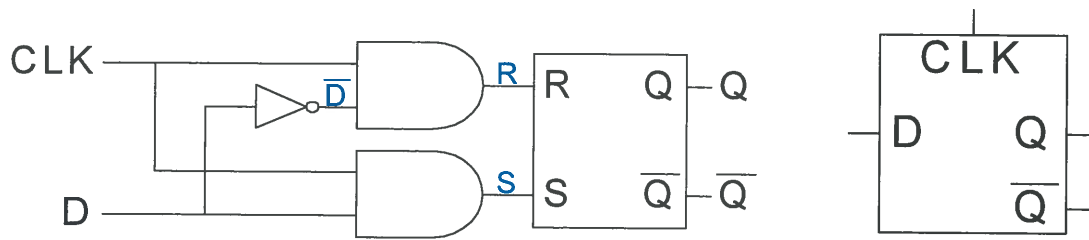
2. Basic Building Blocks: D Latch Internal Circuit



CLK	D	$\overline{D}$	S	R	Q	$\overline{Q}$
0	X		0	0	$Q(t+1) = Q(t)$	
1	0		0	1	0	
1	1		1	0	1	

32

2. Basic Building Blocks: D Latch Internal Circuit



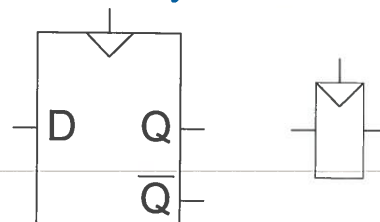
CLK	D	$\overline{D}$	S	R	Q	$\overline{Q}$
0	X	$\overline{X}$	0	0	Q_{prev}	$\overline{Q}_{prev}$
1	0	1	0	1	0	1
1	1	0	1	0	1	0

33

2. Basic Building Blocks: D Flip-Flop

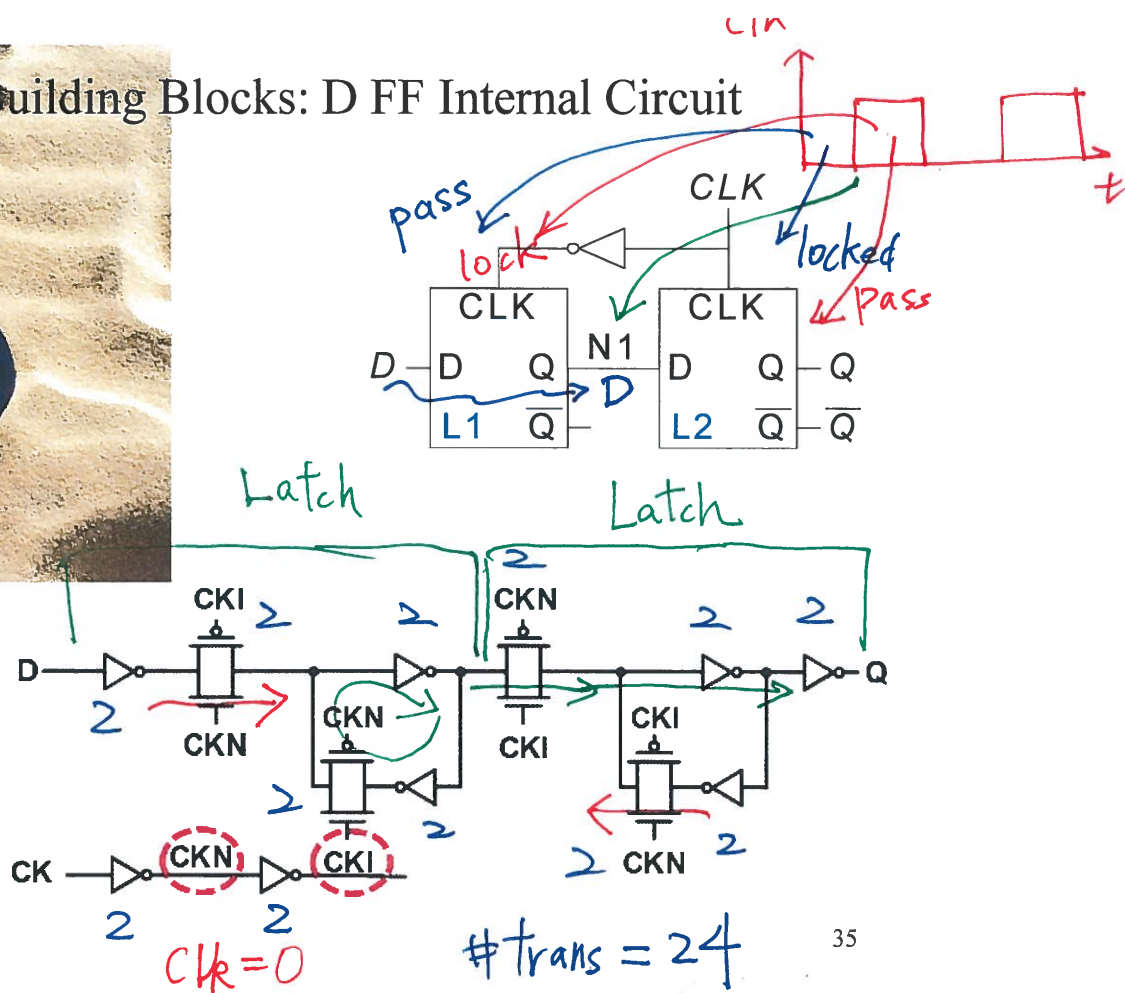
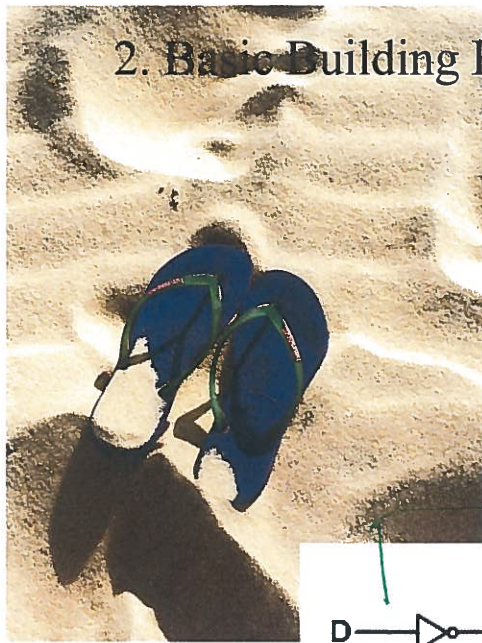
- Two inputs: CLK , D
- *Function*
 - The flip-flop “samples” D on the rising edge of CLK
 - When CLK rises from 0 to 1, D passes through to Q
 - Otherwise, Q holds its previous value
 - Q changes only on the rising edge of CLK
- A flip-flop is called an *edge-triggered* device because it is activated on the clock edge

D Flip-Flop Symbols



34

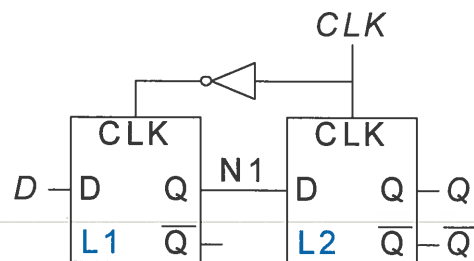
2. Basic Building Blocks: D FF Internal Circuit



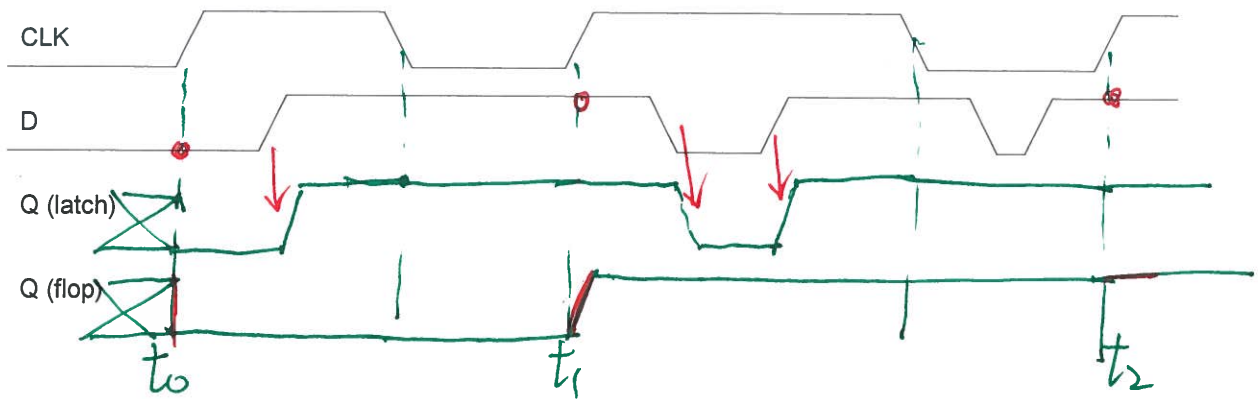
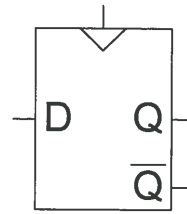
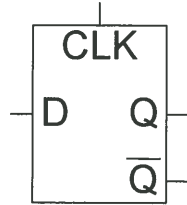
35

D Flip-Flop Internal Circuit

- Two back-to-back latches (L1 and L2) controlled by complementary clocks
- When $CLK = 0$
 - L1 is transparent, L2 is opaque
 - D passes through to N1
- When $CLK = 1$
 - L2 is transparent, L1 is opaque
 - N1 passes through to Q
- Thus, on the edge of the clock (when CLK rises from 0 to 1)
 - D passes through to Q

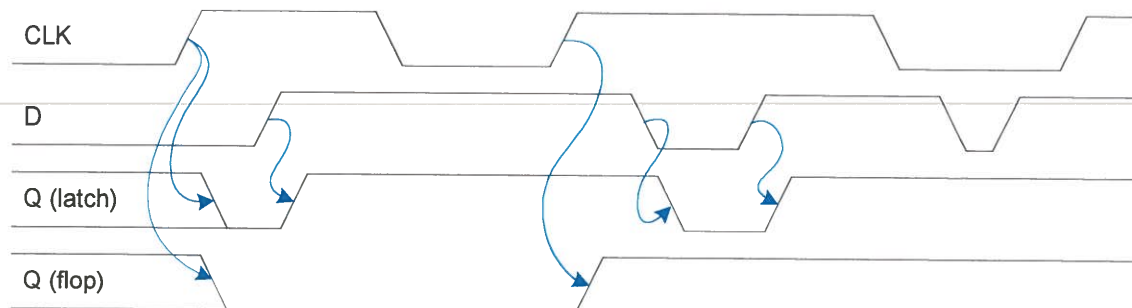
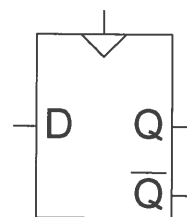
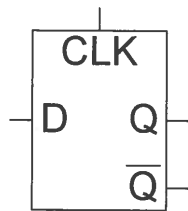


D Flip-Flop vs. D Latch



37

D Flip-Flop vs. D Latch



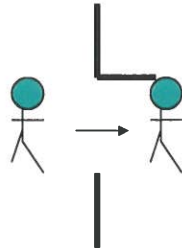
38

2. Basic Building Blocks: Latch and FF (two latches)

A latch can be considered as a door

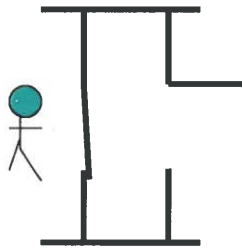


CLK = 0, door is shut

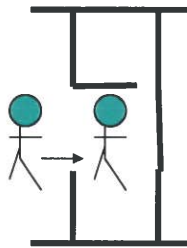


CLK = 1, door is unlocked

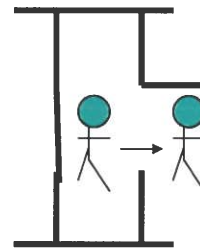
A flip-flop is a two-door entrance



CLK = 1



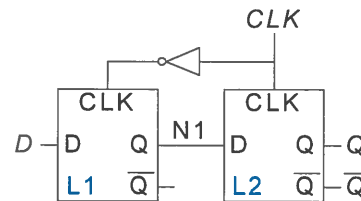
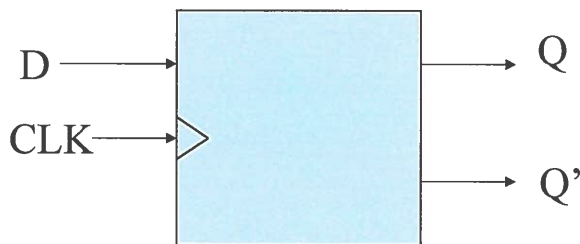
CLK = 0



CLK = 1

39

2. Basic Building Blocks: D Flip-Flop (Delay)



Id	D	Q(t)	Q(t+1)
0	0	0	0
1	0	1	0
2	1	0	1
3	1	1	1

State table

PS \ D	0	1
0	0	1
1	0	1

NS = Q(t+1)

Characteristic Expression: $Q(t+1) = D(t)$

40