

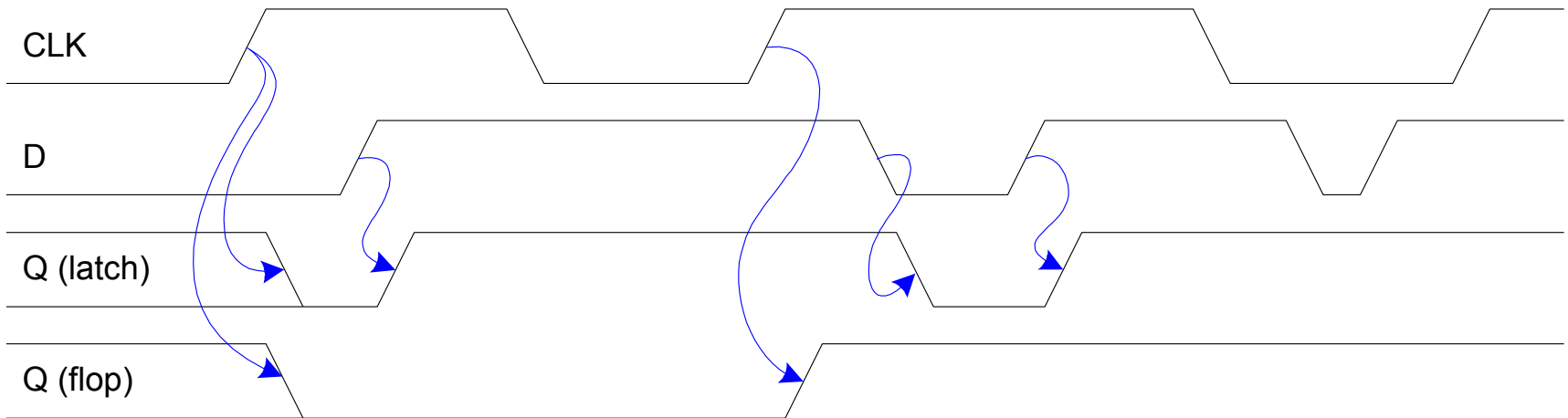
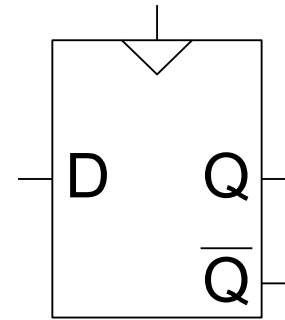
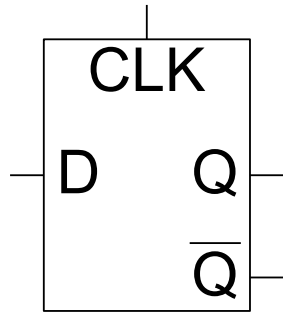
Lecture 12:

Sequential Networks – Flip flops and registers

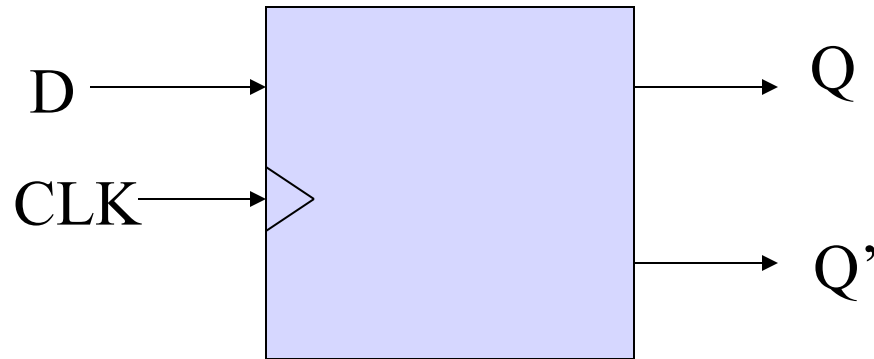
CSE 140: Components and Design Techniques for
Digital Systems

Diba Mirza
Dept. of Computer Science and Engineering
University of California, San Diego

D Flip-Flop vs. D Latch



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
 $Q(t+1) = D(t)$

What does the equation mean?

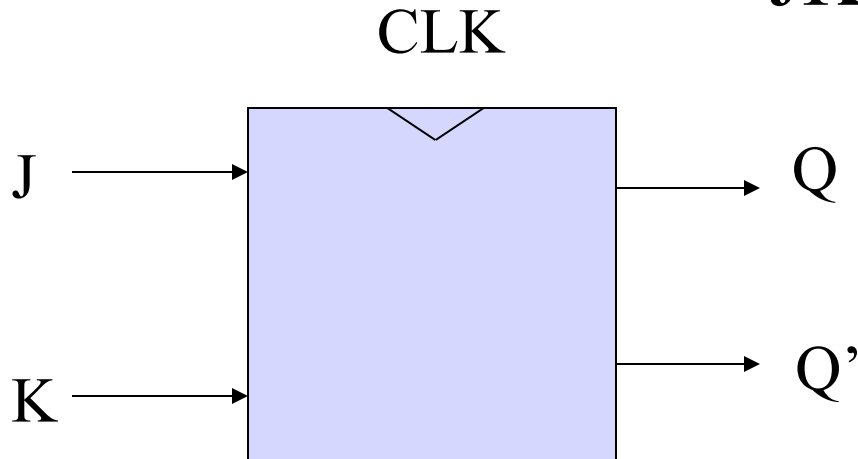


iClicker

How long does a D-flip flop store a bit before its output can potentially change?

- A. Half a clock cycle
- B. One clock cycle
- C. Two clock cycles
- D. There is no minimum time

JK F-F



State table

PS \ JK		JK			
		00	01	11	10
0	0	0	0	1	1
	1	1	0	0	1

Q(t+1)

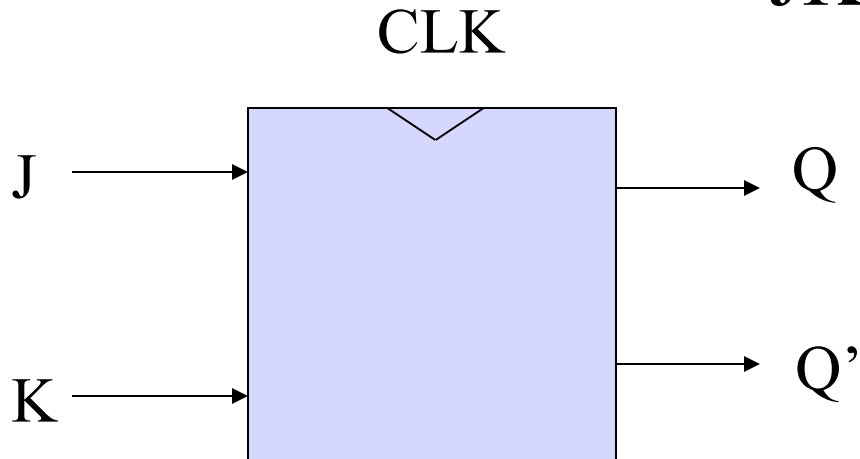
Sounds a lot like a
latch I know...



Characteristic Expression
 $Q(t+1) = Q(t)K'(t) + Q'(t)J(t)$

JK F-F

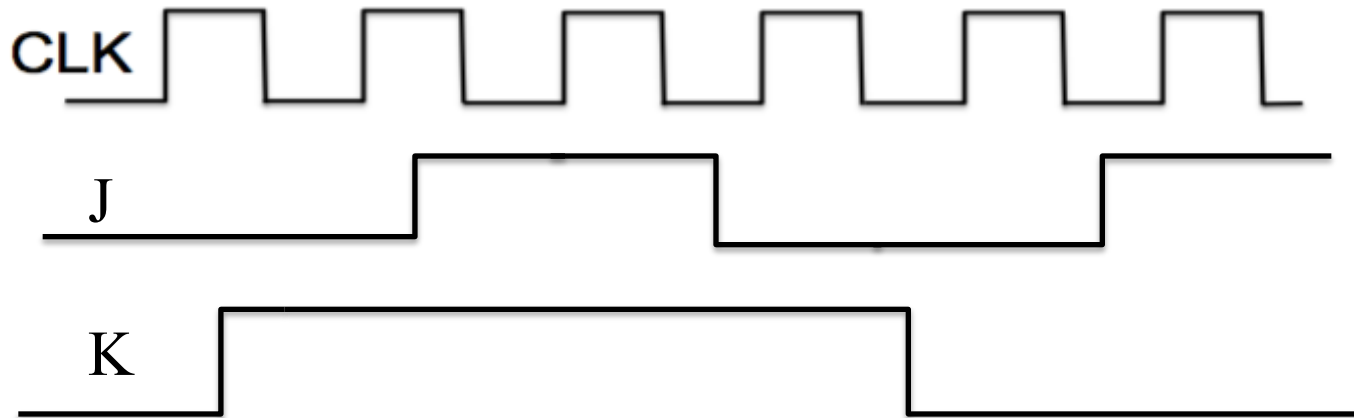
Characteristic Expression
 $Q(t+1) = Q(t)K'(t) + Q'(t)J(t)$



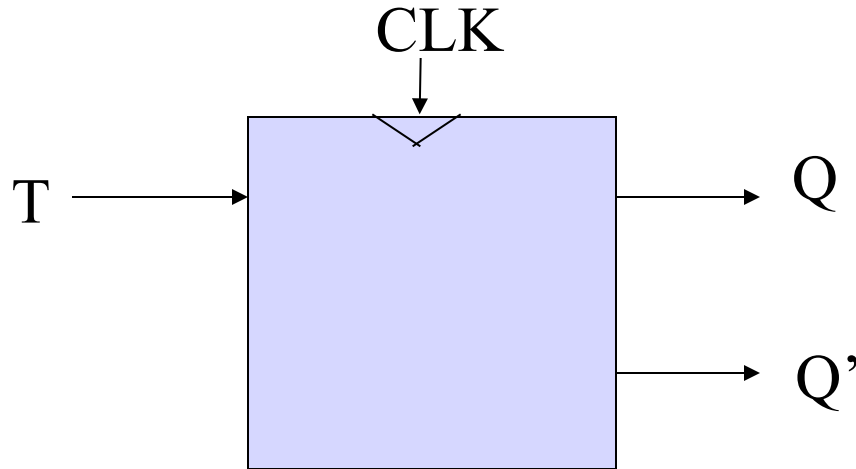
State table

PS \ JK		JK			
		00	01	11	10
0	0	0	0	1	1
	1	1	0	0	1

$Q(t+1)$



T Flip-Flop (Toggle)



State table

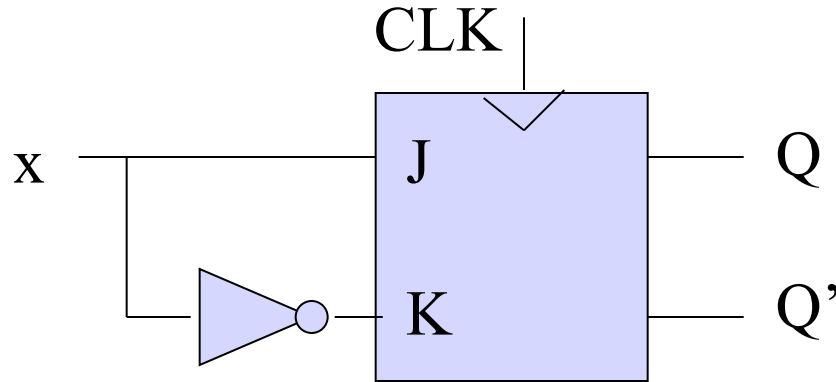
PS \ T	0	1
0	0	1
1	1	0

$Q(t+1)$

Characteristic Expression

$$Q(t+1) = Q'(t)T(t) + Q(t)T'(t)$$

Using a JK F-F to implement a D and T F-F

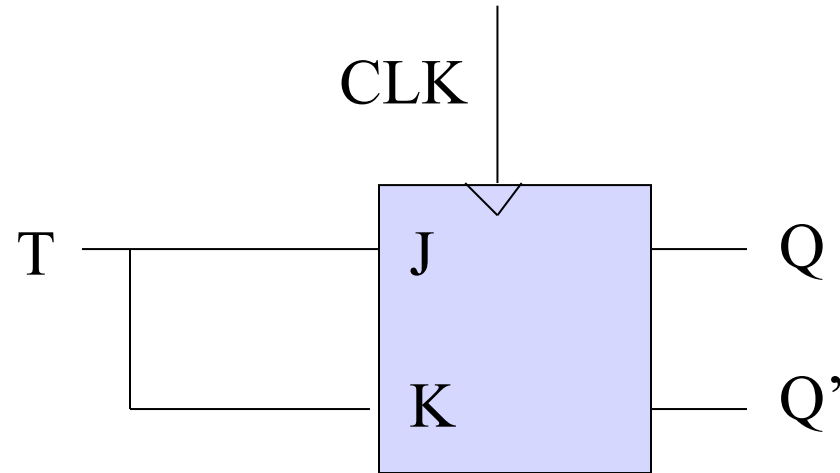


iClicker

The above circuit behaves as which of the following flip flops?

- A. D F-F
- B. T F-F
- C. None of the above

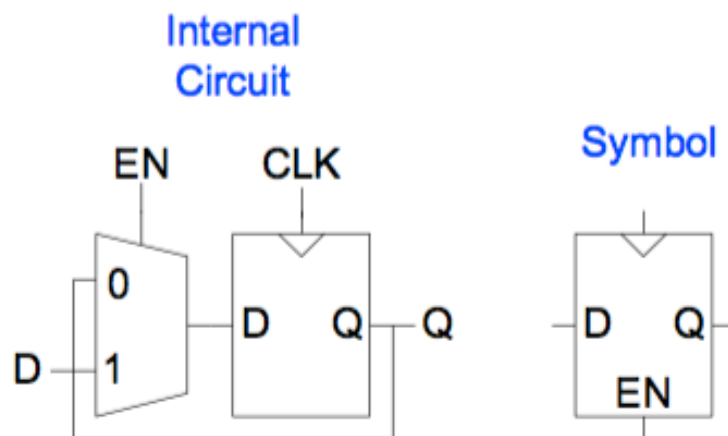
Using a JK F-F to implement a D and T F-F



T flip flop

Enabled Flip-Flops

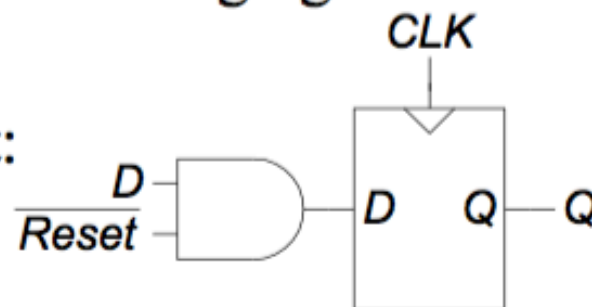
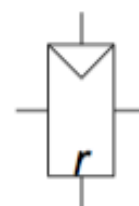
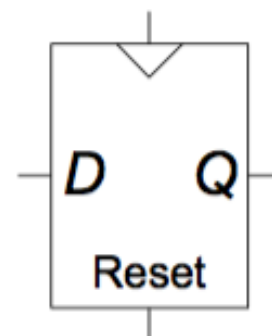
- **Inputs:** CLK , D , EN
 - The enable input (EN) controls when new data (D) is stored
- **Function**
 - $EN = 1$: D passes through to Q on the clock edge
 - $EN = 0$: the flip-flop retains its previous state



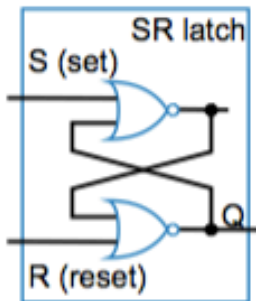
Resettable Flip-Flops

Symbols

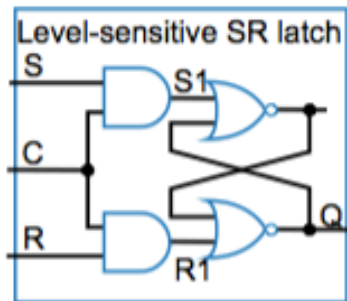
- **Inputs:** CLK , D , $Reset$
- **Function:**
 - **$Reset = 1$:** Q is forced to 0
 - **$Reset = 0$:** flip-flop behaves as ordinary D flip-flop
- Two types:
 - **Synchronous:** resets at the clock edge only
 - **Asynchronous:** resets immediately when $Reset = 1$
- Asynchronously resettable flip-flop requires changing the internal circuitry of the flip-flop
- Synchronously resettable flip-flop circuit:
- There are also synch/asynch settable FFs



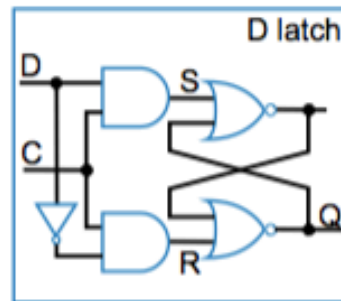
Bit Storage Overview



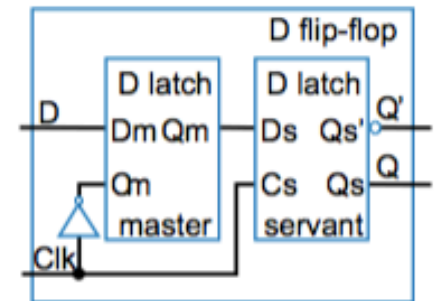
Feature: $S=1$ sets Q to 1, $R=1$ resets Q to 0. Problem: $SR=11$ yield undefined Q .



Feature: S and R only have effect when $C=1$. We can design outside circuit so $SR=11$ never happens when $C=1$. Problem: avoiding $SR=11$ can be a burden.

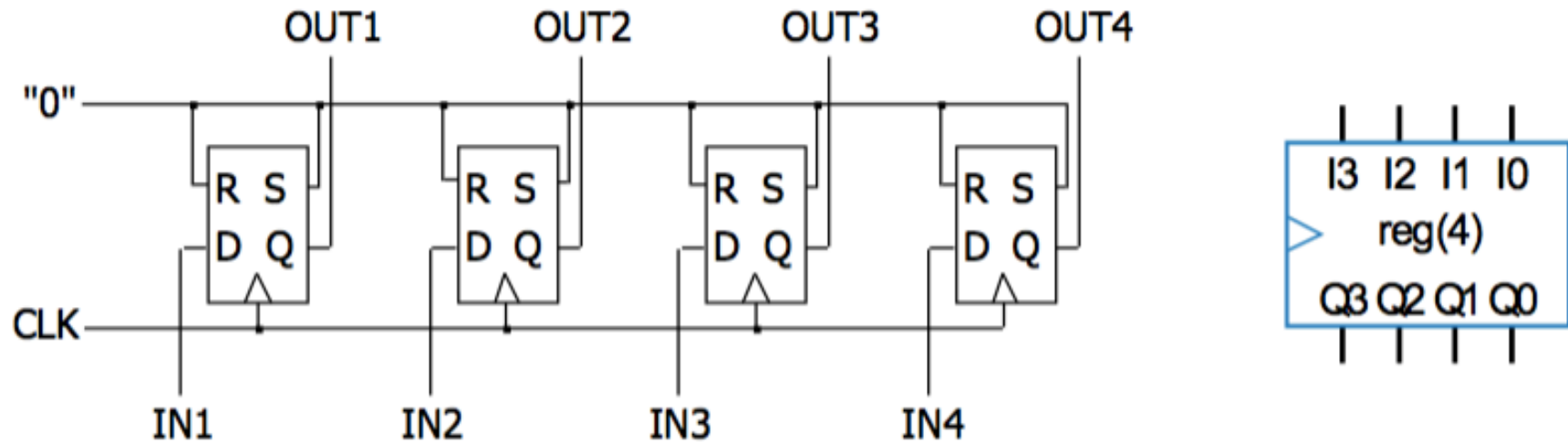


Feature: SR can't be 11 if D is stable before and while $C=1$, and will be 11 for only a brief glitch even if D changes while $C=1$. Problem: $C=1$ too long propagates new values through too many latches: too short may not enable a store.



Feature: Only loads D value present at rising clock edge, so values can't propagate to other flip-flops during same clock cycle. Tradeoff: uses more gates internally than D latch, and requires more external gates than SR – but gate count is less of an issue today.

Building blocks with FFs: Basic Register



Midterm review

True or False

1. In a K-map an implicant can consist of a group of two adjacent cells containing zeros
2. A minterm may evaluate to a one for multiple input combinations.
3. An incompletely specified function results in a minimal circuit whose output cannot be determined for some input combinations.
4. When minimizing a function to product of sum form, we always assume that the don't care terms are 1.
5. The output of combinational circuits depends only on current inputs.

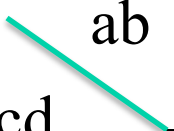
Reduce to SOP form

$$F(a,b,c,d)=\Pi M(3,4,5,6,7,11,12). \Pi D(10, 15)$$

ab \ cd	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Reduce the following to a SOP form

$$F(a,b,c,d)=\prod M(3,4,5,6,7,11,12). \prod D(10, 15)$$



ab		00	01	11	10
cd	00	0 1	4 0	12 0	8 1
	01	1 1	5 0	13 1	9 1
	11	3 0	7 0	15 X	11 0
	10	2 1	6 0	14 1	10 X

Minimize using Boolean algebra

- $F(x,y,z) = xy + y'((x+y')' + z) + xz$

Design Problem

Design a circuit that gives the absolute distance between two 2-bit numbers (e.g. $x=3$, $y=1$, $d=2$)

Design Problem

- Design a circuit that gives the absolute distance between the two numbers (e.g. $x=3$ $y=1$ $d=2$)

x_1	x_0	y_1	y_0	d_1	d_0
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

