

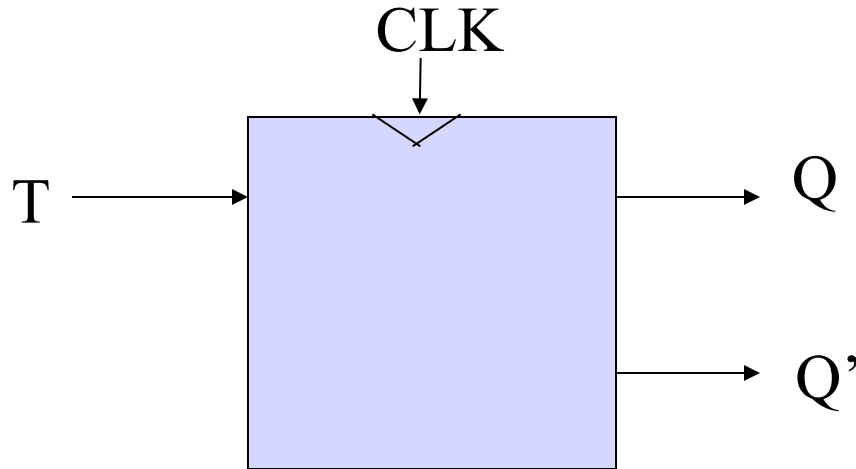
Lecture 13:

Sequential Networks – Flip flops and Finite State Machines

CSE 140: Components and Design Techniques for
Digital Systems

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

T Flip-Flop (Toggle)



State table

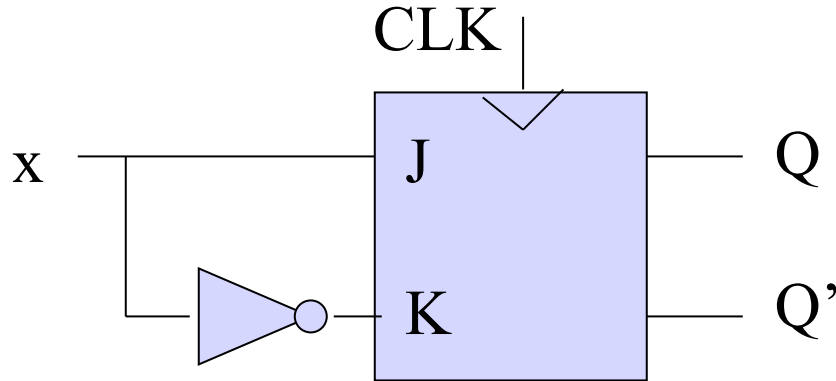
PS \ T	0	1
0	0	1
1	1	0

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

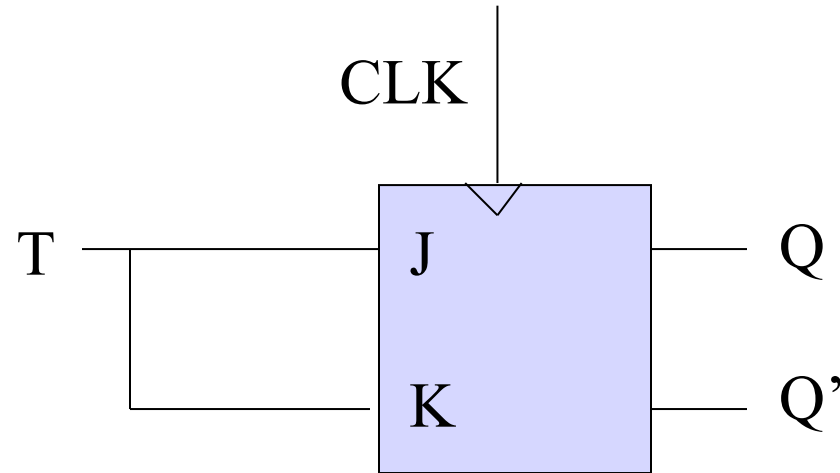


iClicker

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

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

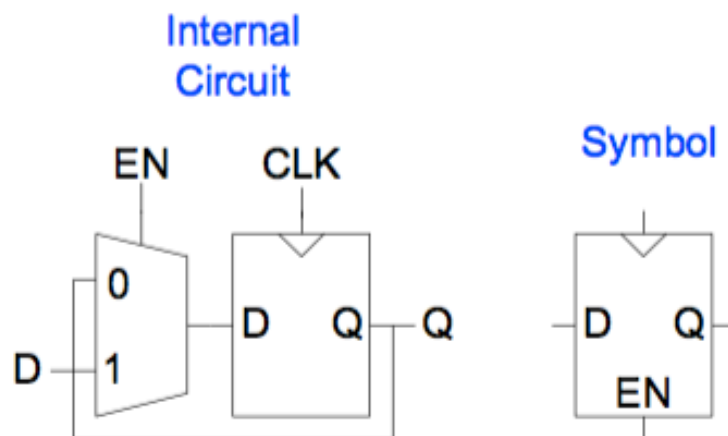
Using a JK F-F to implement a T F-F



T flip flop

Enabled Flip-Flops

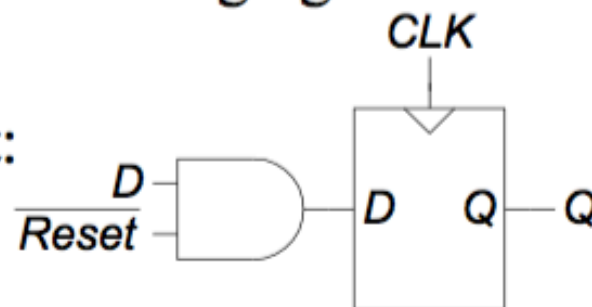
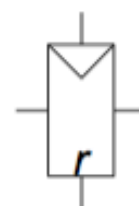
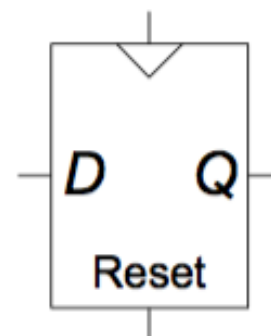
- **Inputs:** CLK , D , EN
 - The enable input (EN) controls when new data (D) is stored
- **Function**
 - $EN = 1$: D passes through to Q on the clock edge
 - $EN = 0$: the flip-flop retains its previous state



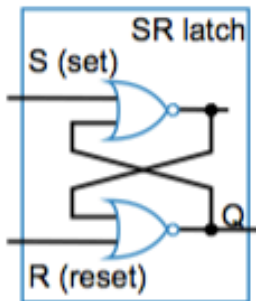
Resettable Flip-Flops

Symbols

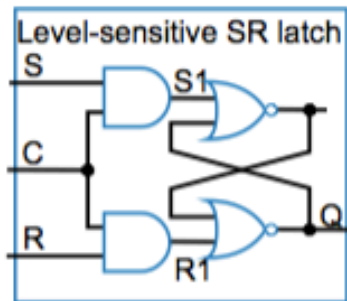
- **Inputs:** CLK , D , $Reset$
- **Function:**
 - **$Reset = 1$:** Q is forced to 0
 - **$Reset = 0$:** flip-flop behaves as ordinary D flip-flop
- Two types:
 - **Synchronous:** resets at the clock edge only
 - **Asynchronous:** resets immediately when $Reset = 1$
- Asynchronously resettable flip-flop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



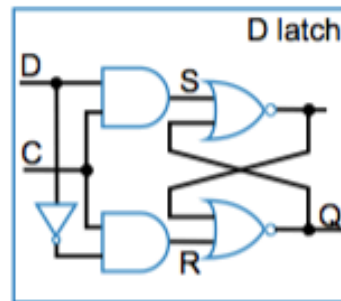
Bit Storage Overview



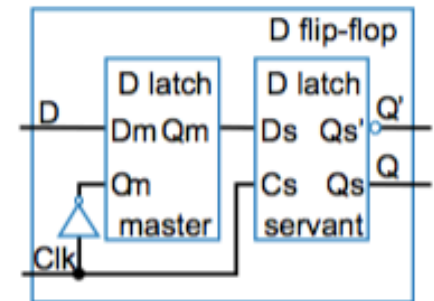
Feature: $S=1$ sets Q to 1, $R=1$ resets Q to 0. Problem: $SR=11$ yield undefined Q .



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.

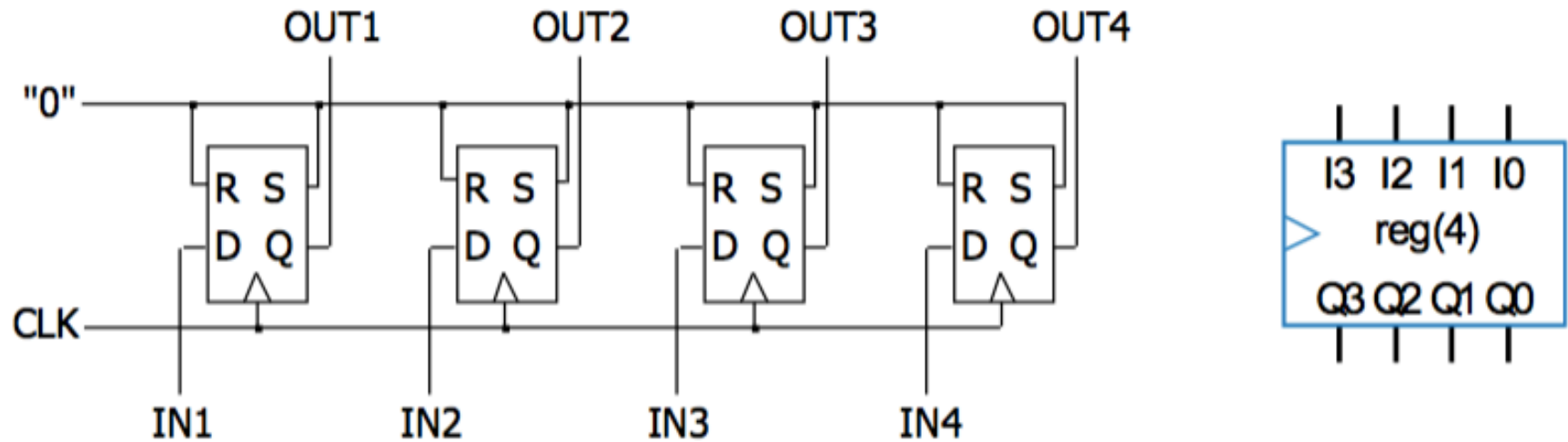


Feature: SR can't be 11 if D is stable before and while $C=1$, and will be 11 for only a brief glitch even if D changes while $C=1$. Problem: $C=1$ too long propagates new values through too many latches: too short may not enable a store.



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.

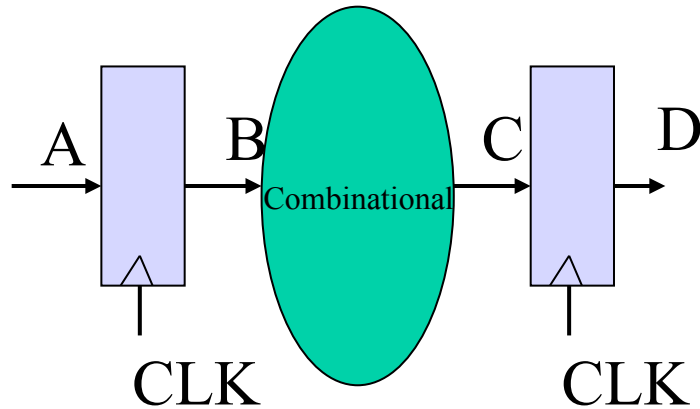
Building blocks with FFs: Basic Register



Flip flops

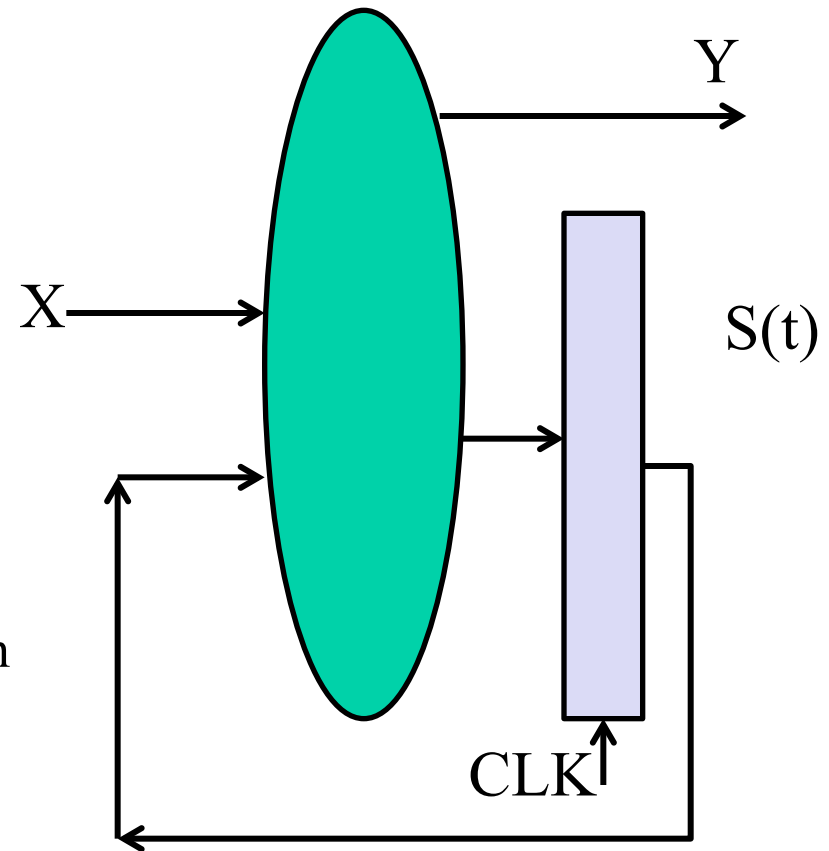
- Write the state-table of the following flip-flops
 - D
 - JK
 - T

Sequential Networks



RTL: Register-Transfer Level Description

1. Components F-Fs
2. Specification
3. Implementation: Excitation Table



Specification

- Combinational Logic
 - Truth Table
 - Boolean Expression
 - Logic Diagram (No feedback loops)
- Sequential Networks: **State Diagram (Memory)**
 - State Table and Excitation Table
 - Characteristic Expression
 - Logic Diagram (FFs and feedback loops)

What we will learn:

1. Describe the desired behavior of a sequential circuit over time (FSMs)
2. Given the behavior of a sequential circuit, implement the circuit

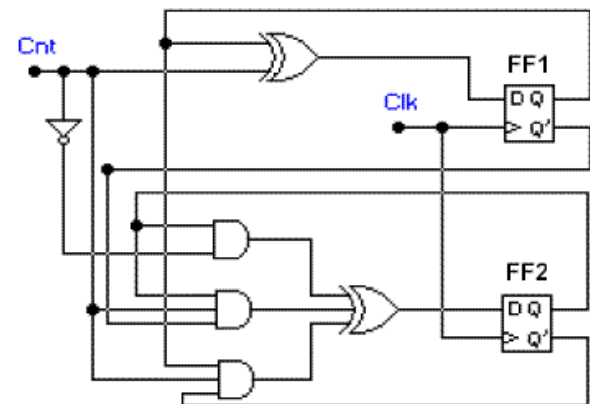
Wall-E is a Finite State Machine



Describing Wall-E

Active

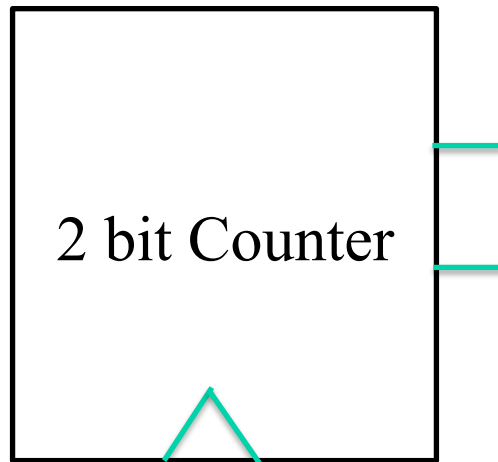
Inactive



Implementing Wall-E

Finite State Machines: Describing circuit behavior over time

Symbol/ Circuit

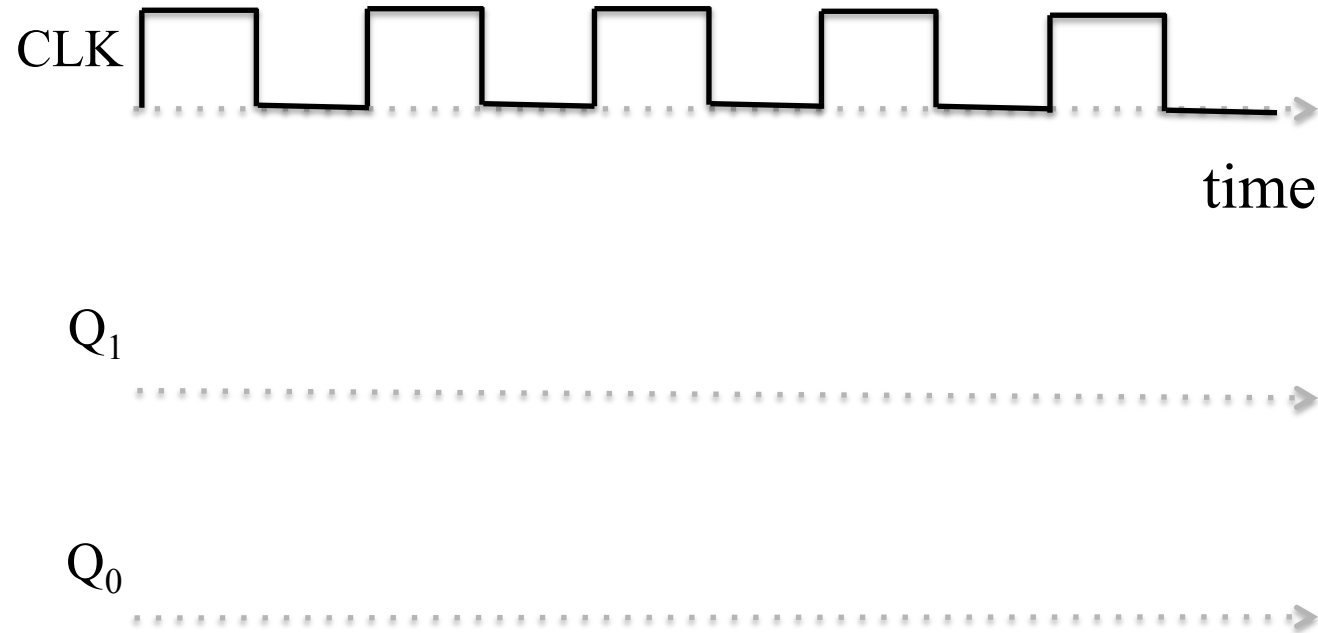


Finite State Machines: Describing circuit behavior over time

Symbol/ Circuit



Output over time



What is the expected output of the counter over time?

Finite State Machines: Describing circuit behavior over time

Symbol/ Circuit

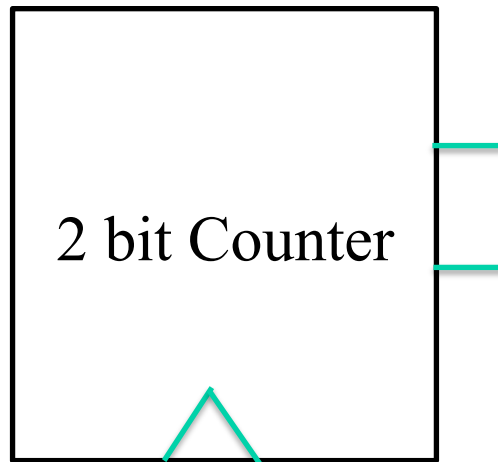
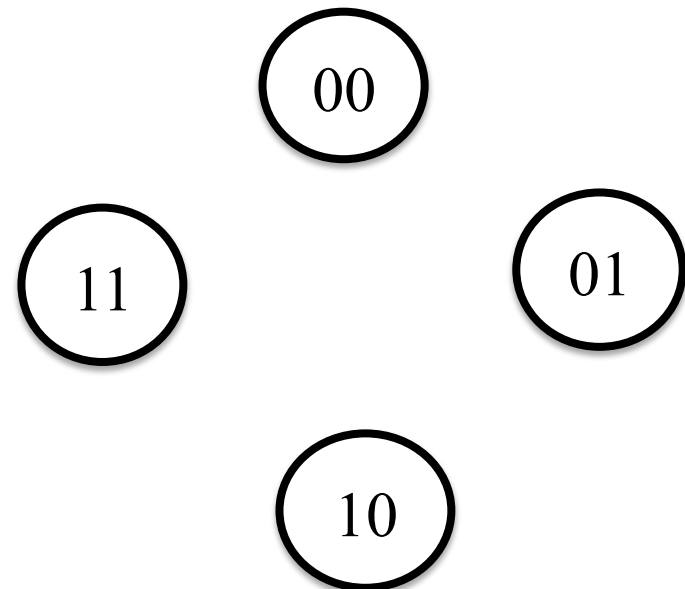
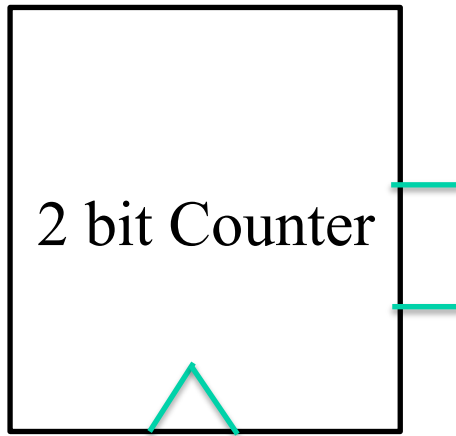


Diagram that depicts behavior over time

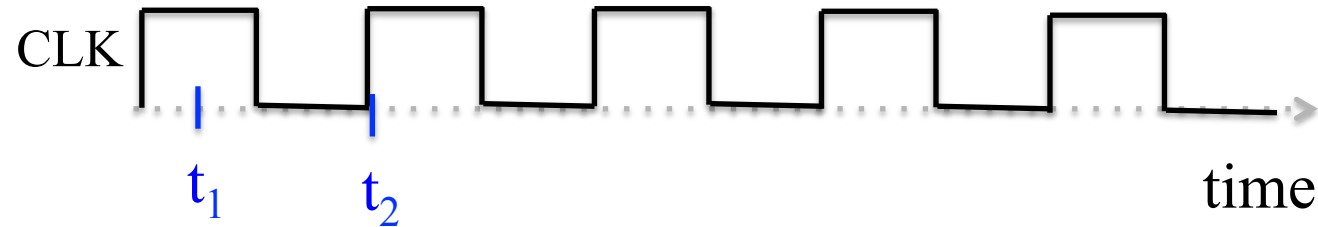


State: What is it ? Why do we need it?

Symbol/ Circuit



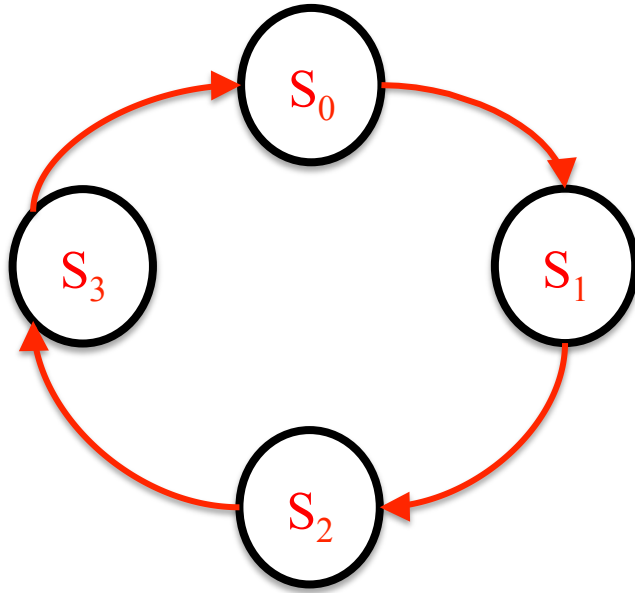
Behavior over time



PI Q: At time t_1 , what information is needed to produce the output of the counter at the next rising edge of the clock (i.e t_2)?

- A. All the outputs of the counter until t_1
- B. The initial output of the counter at time $t=0$
- C. The output of the counter at current time t_1
- D. We cannot determine the output of the counter at t_2 prior to t_2

Implementing the 2 bit counter



State Diagram

Current state	Next State
S_0	S_1
S_1	S_2
S_2	S_3
S_3	S_0

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$

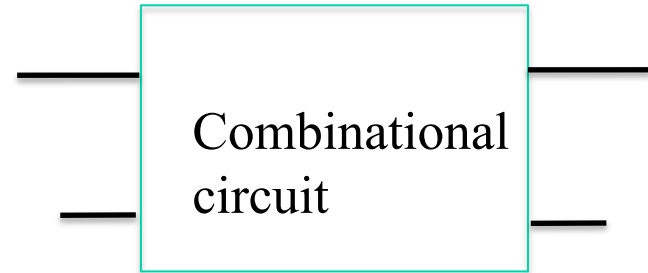
State Table

State Table

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

PI Q: Which of the following is the likely structure of the circuit realization of the counter

A.



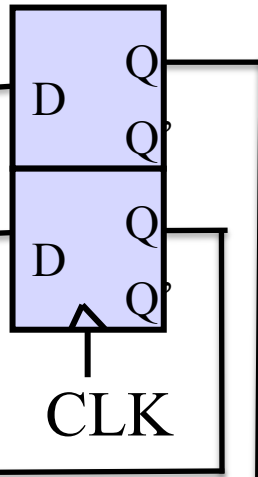
B.

$Q_0(t)$

Combinational circuit

$Q_1(t)$

CLK



Circuit with 2 flip flops

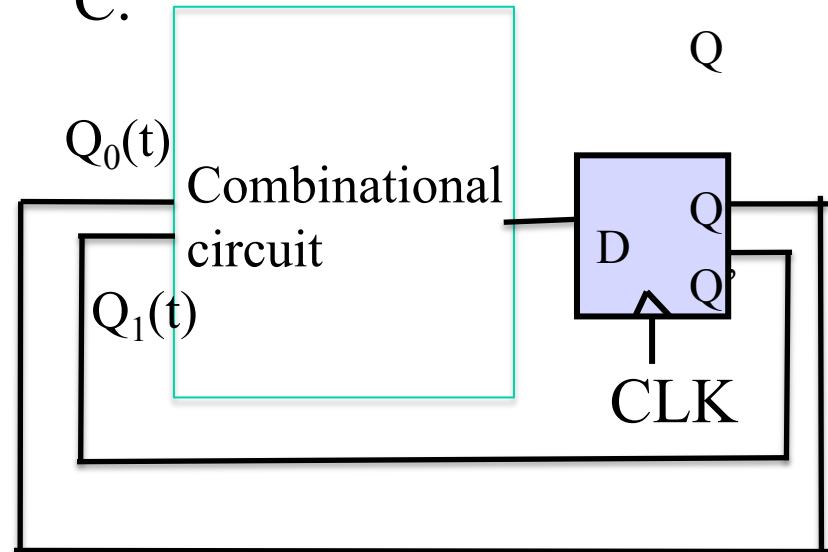
C.

$Q_0(t)$

Combinational circuit

$Q_1(t)$

CLK



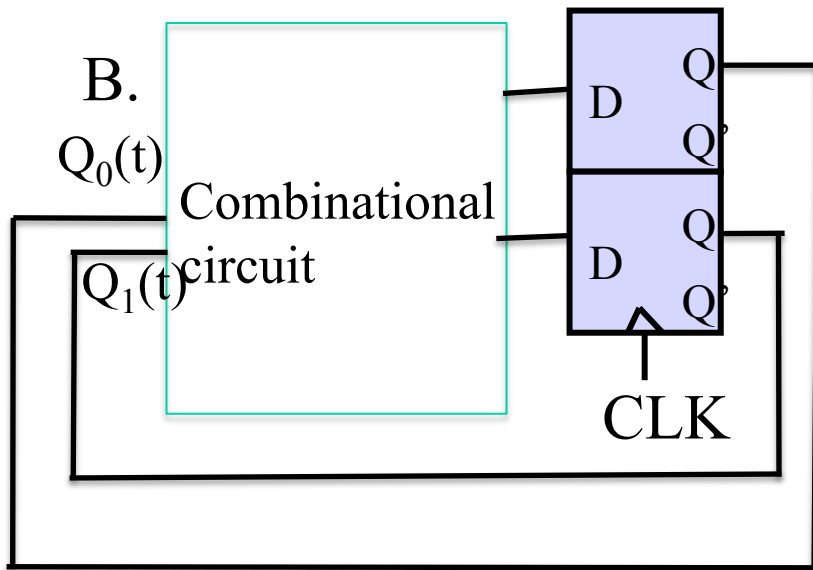
Circuit with one flip flop

State Table

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

$$D_0(t) = Q_0(t)'$$

$$D_1(t) = Q_0(t) Q_1(t)' + Q_0(t)' Q_1(t)$$



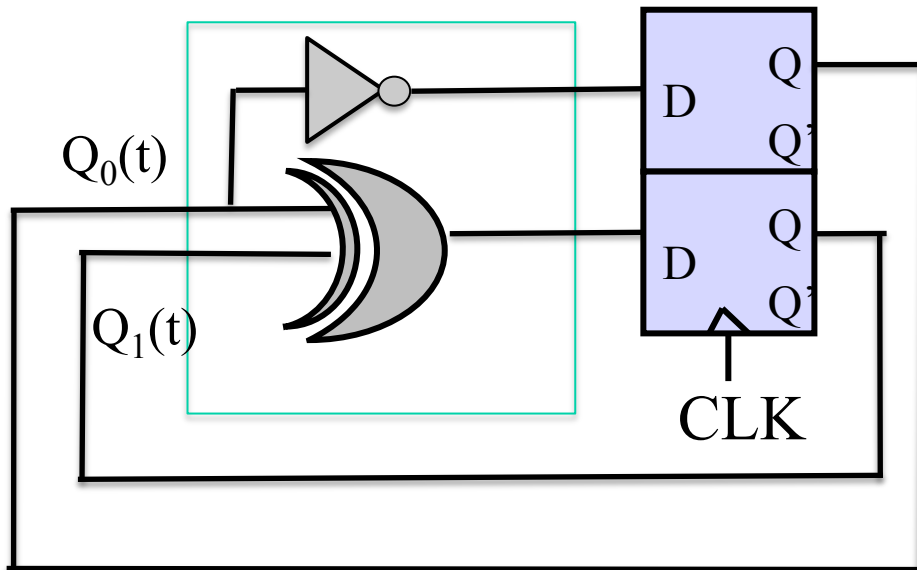
Circuit with 2 flip flops

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

State Table

$$Q_0(t+1) = Q_0(t)'$$

$$Q_1(t+1) = Q_0(t) Q_1(t)' + Q_0(t)' Q_1(t)$$

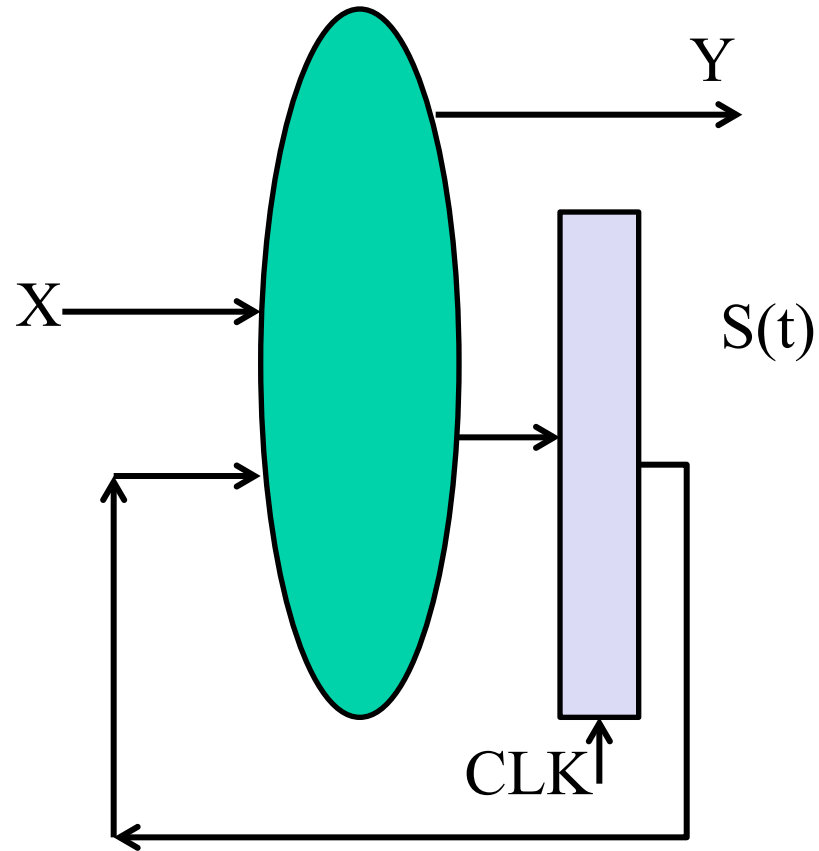


Implementation of 2-bit counter

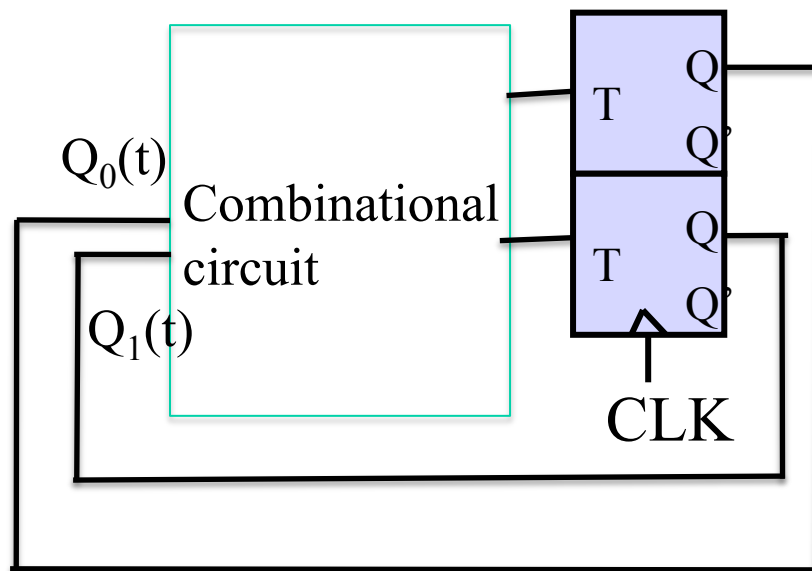
We store the current state using D-flip flops so that:

- Inputs to the combinational circuit don't change while the next output is being computed
- The transition to the next state only occurs at the rising edge of the clock

Generalized Model of Sequential Circuits



Let's implement our free running 2-bit counter using T-flip flops

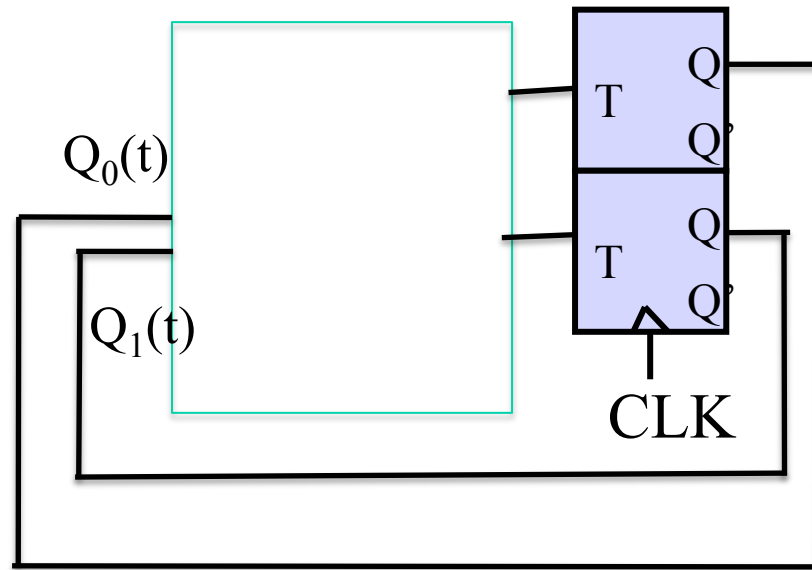


Circuit with 2 T-flip flops

Excitation table

id	$Q_1(t)$	$Q_0(t)$	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0			0	1
1	0	1			1	0
2	1	0			1	1
3	1	1			0	0

Let's implement our free running 2-bit counter using T-flip flops

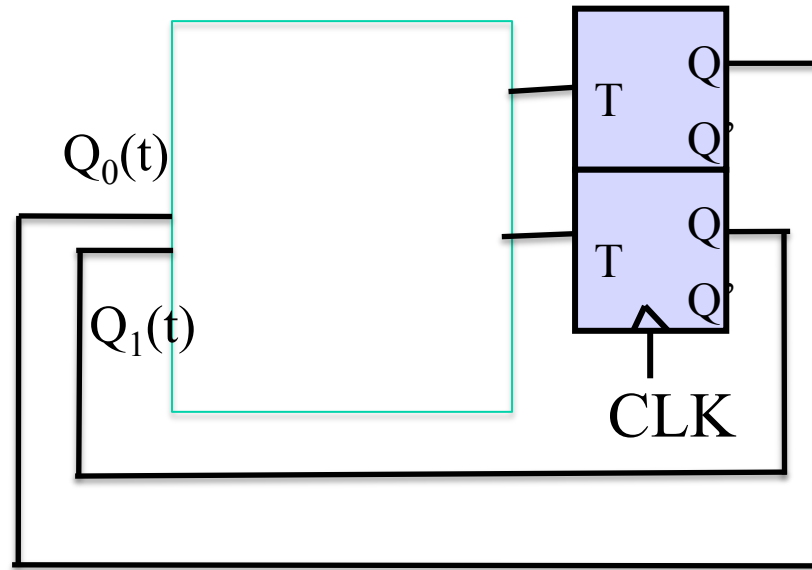


Circuit with 2 T-flip flops

Excitation table

id	$Q_1(t)$	$Q_0(t)$	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	1	0	1
1	0	1	1	1	1	0
2	1	0	0	1	1	1
3	1	1	1	1	0	0

Let's implement our free running 2-bit counter using T-flip flops



Circuit with 2 T-flip flops

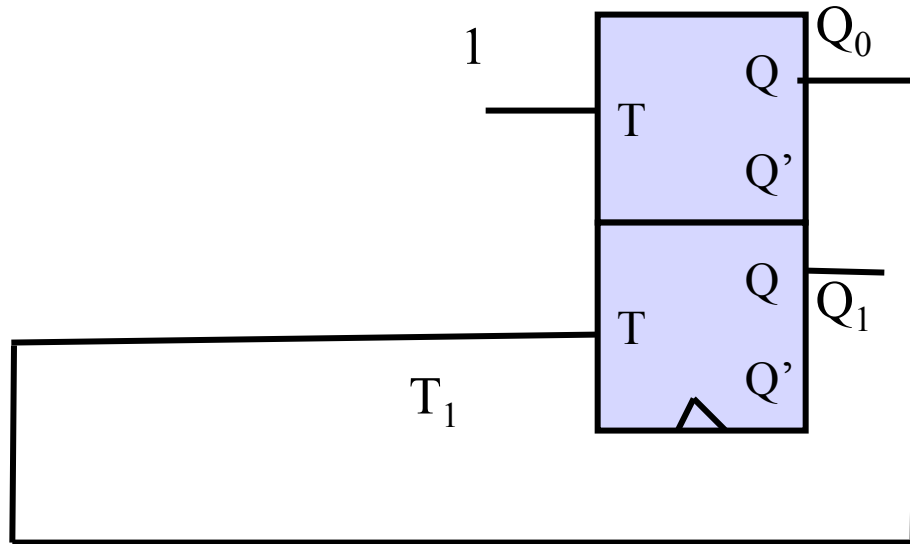
Excitation table

id	$Q_1(t)$	$Q_0(t)$	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	1	0	1
1	0	1	1	1	1	0
2	1	0	0	1	1	1
3	1	1	1	1	0	0

$$T_0(t) =$$

$$T_1(t) =$$

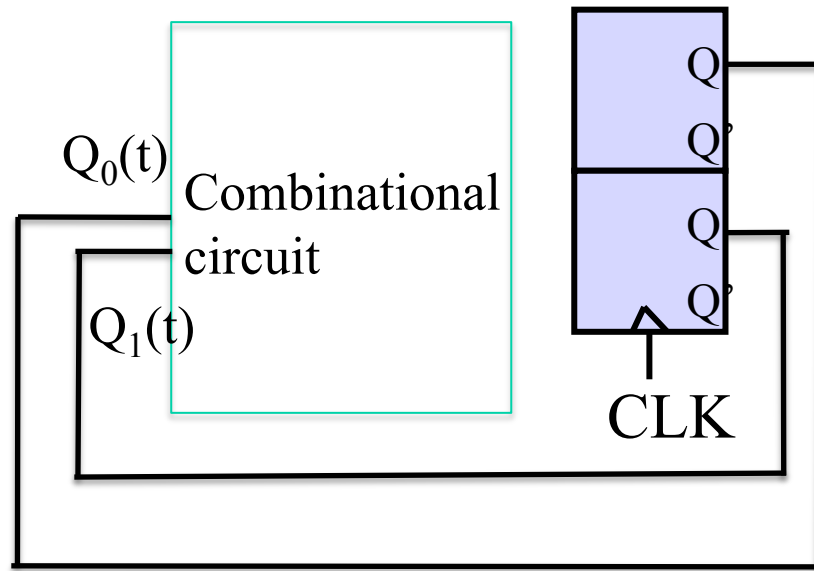
Free running counter with T flip flops



$$T_0(t) = 1$$

$$T_1(t) = Q_0(t)$$

Let's implement our free running 2-bit counter using JK-flip flops

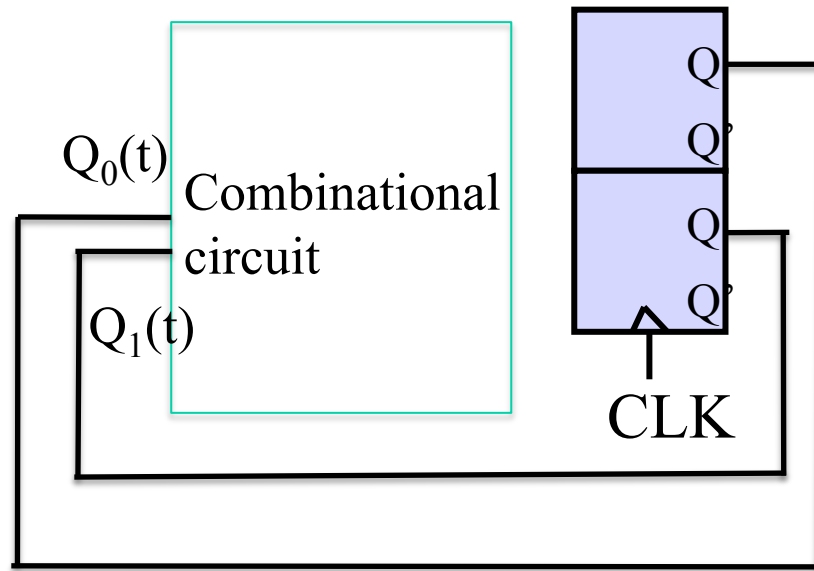


Circuit with 2 JK-flip flops

Excitation table

id	$Q_1(t)$	$Q_0(t)$	$J_1(t)$	$K_1(t)$	$J_0(t)$	$K_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0					0	1
1	0	1					1	0
2	1	0					1	1
3	1	1					0	0

Let's implement our free running 2-bit counter using JK-flip flops

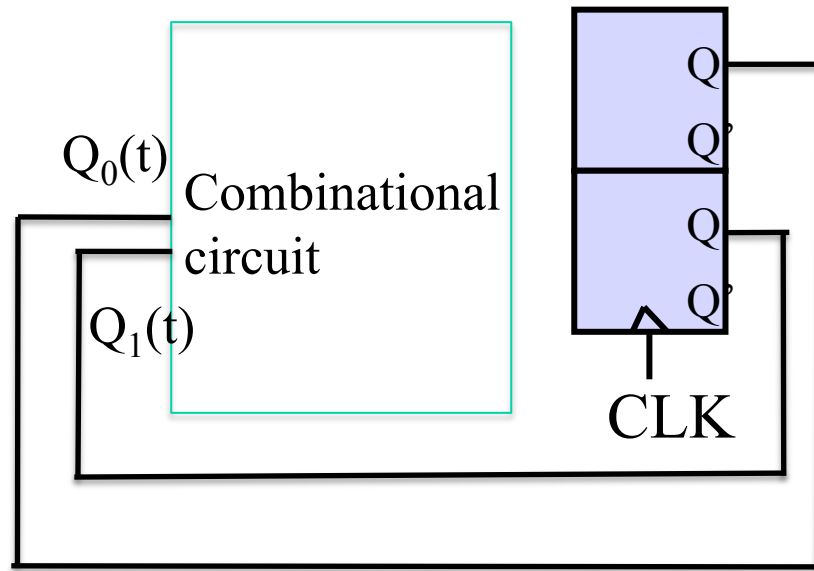


Circuit with 2 JK-flip flops

Excitation table

id	$Q_1(t)$	$Q_0(t)$	$J_1(t)$	$K_1(t)$	$J_0(t)$	$K_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	X			0	1
1	0	1	1	X			1	0
2	1	0	X	0			1	1
3	1	1	X	1			0	0

Let's implement our free running 2-bit counter using JK-flip flops

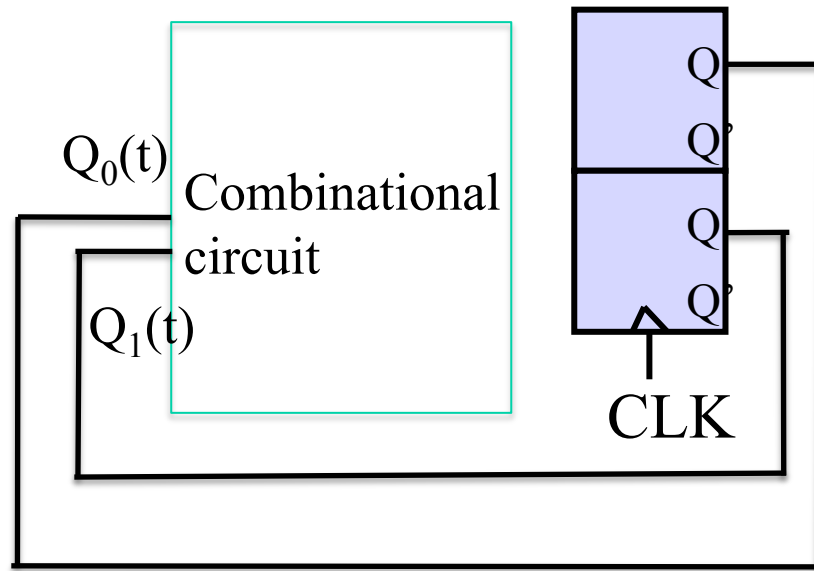


Circuit with 2 JK-flip flops

Excitation table

id	$Q_1(t)$	$Q_0(t)$	$J_1(t)$	$K_1(t)$	$J_0(t)$	$K_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	X	1	X	0	1
1	0	1	1	X	X	1	1	0
2	1	0	X	0	1	X	1	1
3	1	1	X	1	X	1	0	0

Let's implement our free running 2-bit counter using JK-flip flops



$$\begin{aligned} J_1(t) &= Q_0(t) \\ K_1(t) &= Q_0(t) \\ J_0(t) &= 1 \\ K_0(t) &= 1 \end{aligned}$$

Circuit with 2 JK-flip flops

Excitation table

id	$Q_1(t)$	$Q_0(t)$	$J_1(t)$	$K_1(t)$	$J_0(t)$	$K_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	X	1	X	0	1
1	0	1	1	X	X	1	1	0
2	1	0	X	0	1	X	1	1
3	1	1	X	1	X	1	0	0