

# Digital Electronics and Computer Organization

## Digital Design

### Lecture 23: Synchronous Counters



**Birla Institute of Technology & Science, Pilani**  
Hyderabad Campus

10/29/2020



# Binary Synchronous Counter

3 bit binary counter

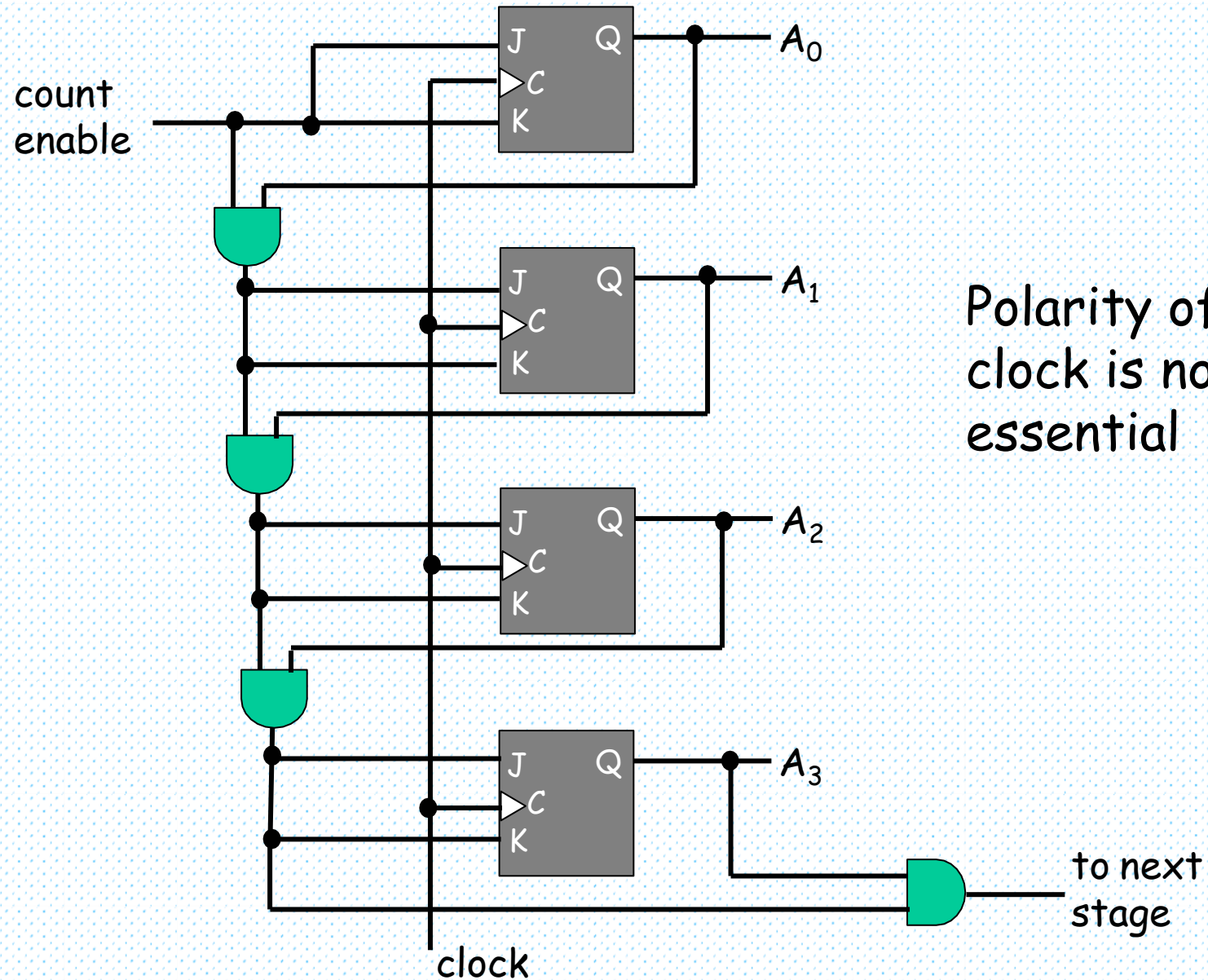
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
0	0	0	0

- **Idea:**
  - to use same clock for all flip-flops

# Synchronous Counters

- There is a common clock
  - that triggers all flip-flops simultaneously
  - If  $T = 0$  or  $J = K = 0$  the flip-flop does not change state.
  - If  $T = 1$  or  $J = K = 1$  the flip-flop does change state.
- Design procedure is so simple
  - no need for going through sequential logic design process
  - $A_0$  is always complemented
  - $A_1$  is complemented when  $A_0 = 1$
  - $A_2$  is complemented when  $A_0 = 1$  and  $A_1 = 1$
  - so on

# 4-bit Binary Synchronous Counter



Polarity of the clock is not essential

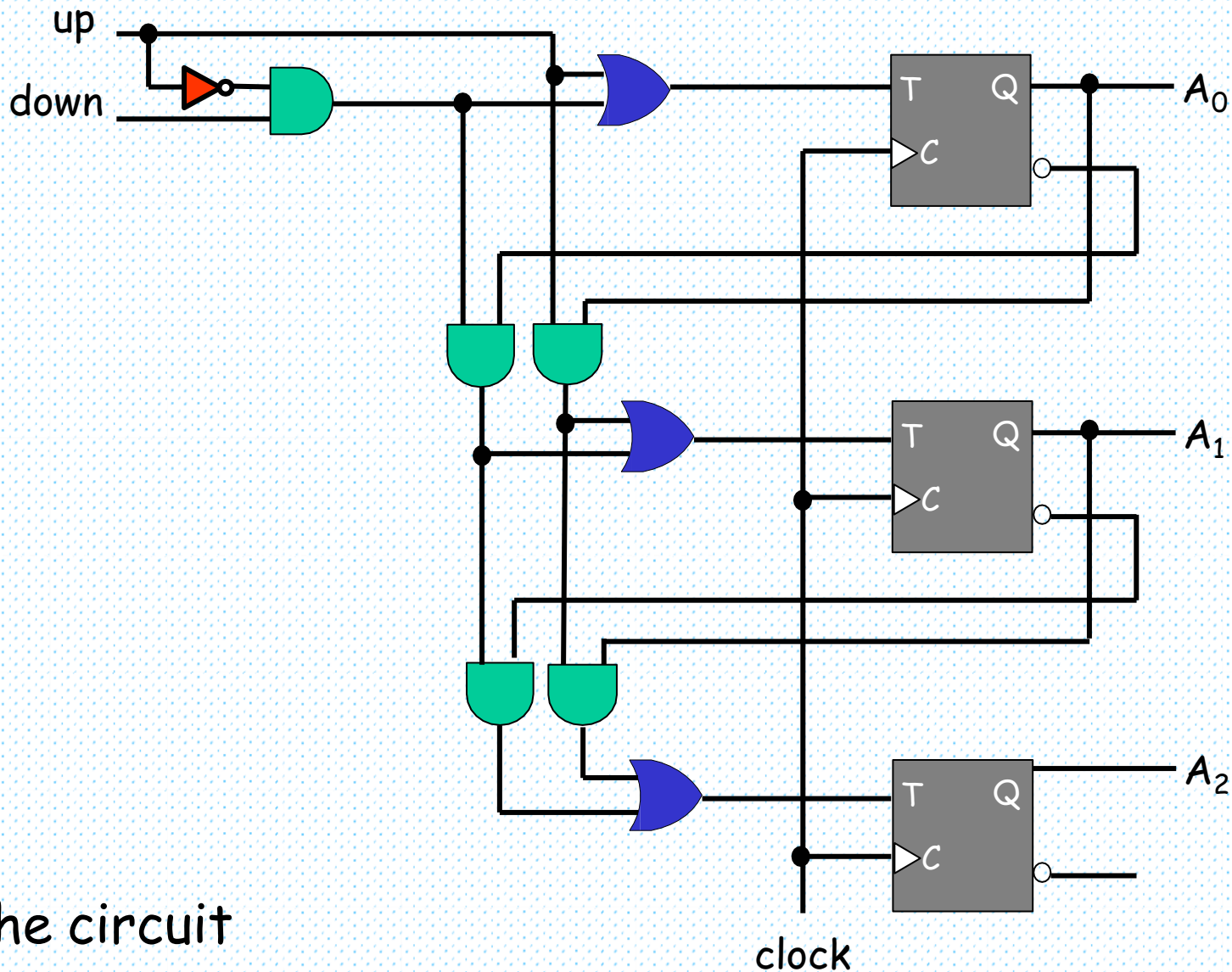
# Up-Down Binary Counter

- When counting downward
  - the least significant bit is always complemented (with each clock pulse)
  - A bit in any other position is complemented if all lower significant bits are equal to 0.
  - For example: 0100
    - Next state: 0011
  - For example: 1100
    - Next state: 1011

STATE TABLE

COUNT	Q1	Q0
3	1	1
2	1	0
1	0	1
0	0	0

# Up-Down Binary Counter



- The circuit

# Synchronous BCD Counter

- Better to apply the sequential circuit design procedure

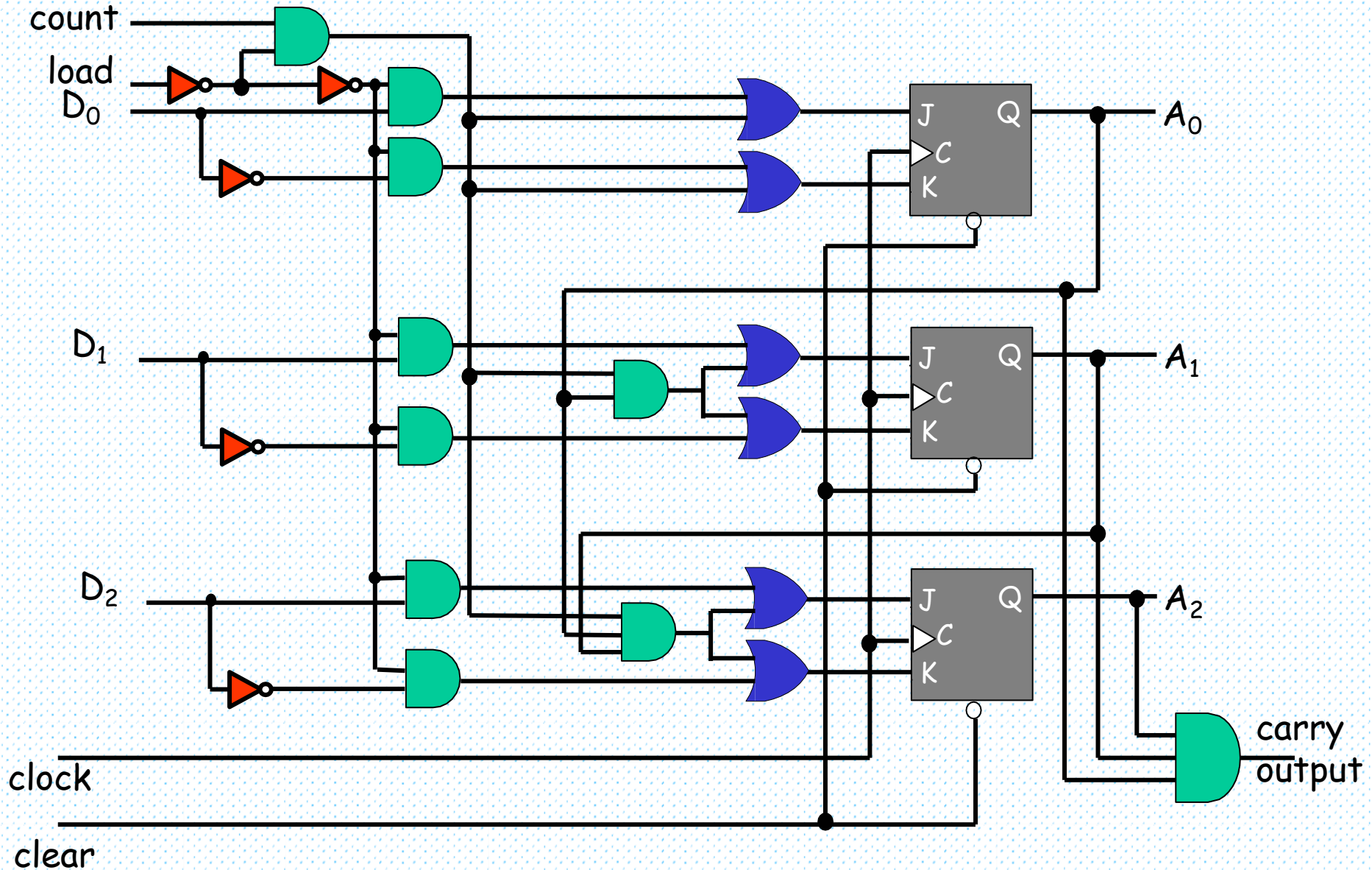
Present state				Next state				output	Flip-Flop inputs			
$A_8$	$A_4$	$A_2$	$A_1$	$A_8$	$A_4$	$A_2$	$A_1$	$y$	$T_8$	$T_4$	$T_2$	$T_1$
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

# Synchronous BCD Counter

- The flip-flop input equations
  - $T_1 = 1$
  - $T_2 = A_8' A_1$
  - $T_4 = A_2 A_1$
  - $T_8 = A_8 A_1 + A_4 A_2 A_1$
- Output equation
  - $y = A_8 A_1$
- Cost
  - Four T flip-flops
  - four 2-input AND gates
  - one OR gate
  - one inverter



# Binary Counter with Parallel Load



# Binary Counter with Parallel Load

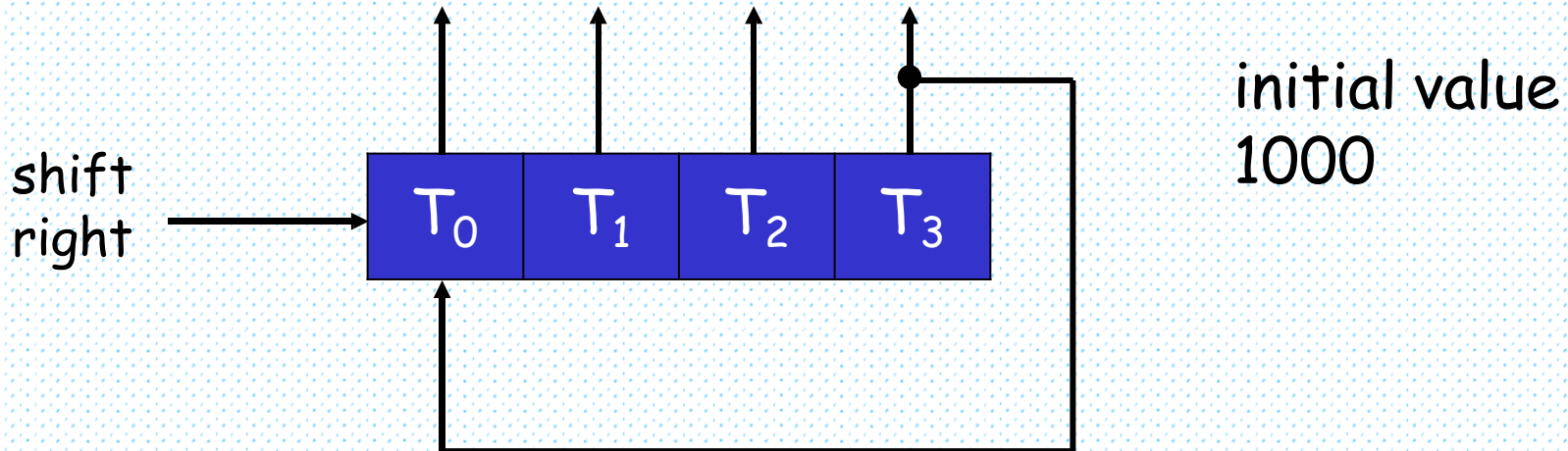
## Function Table

clear	clock	load	Count	Function
0	X	X	X	clear to 0
1	↑	1	X	load inputs
1	↑	0	1	count up
1	↑	0	0	no change

# Other Counters

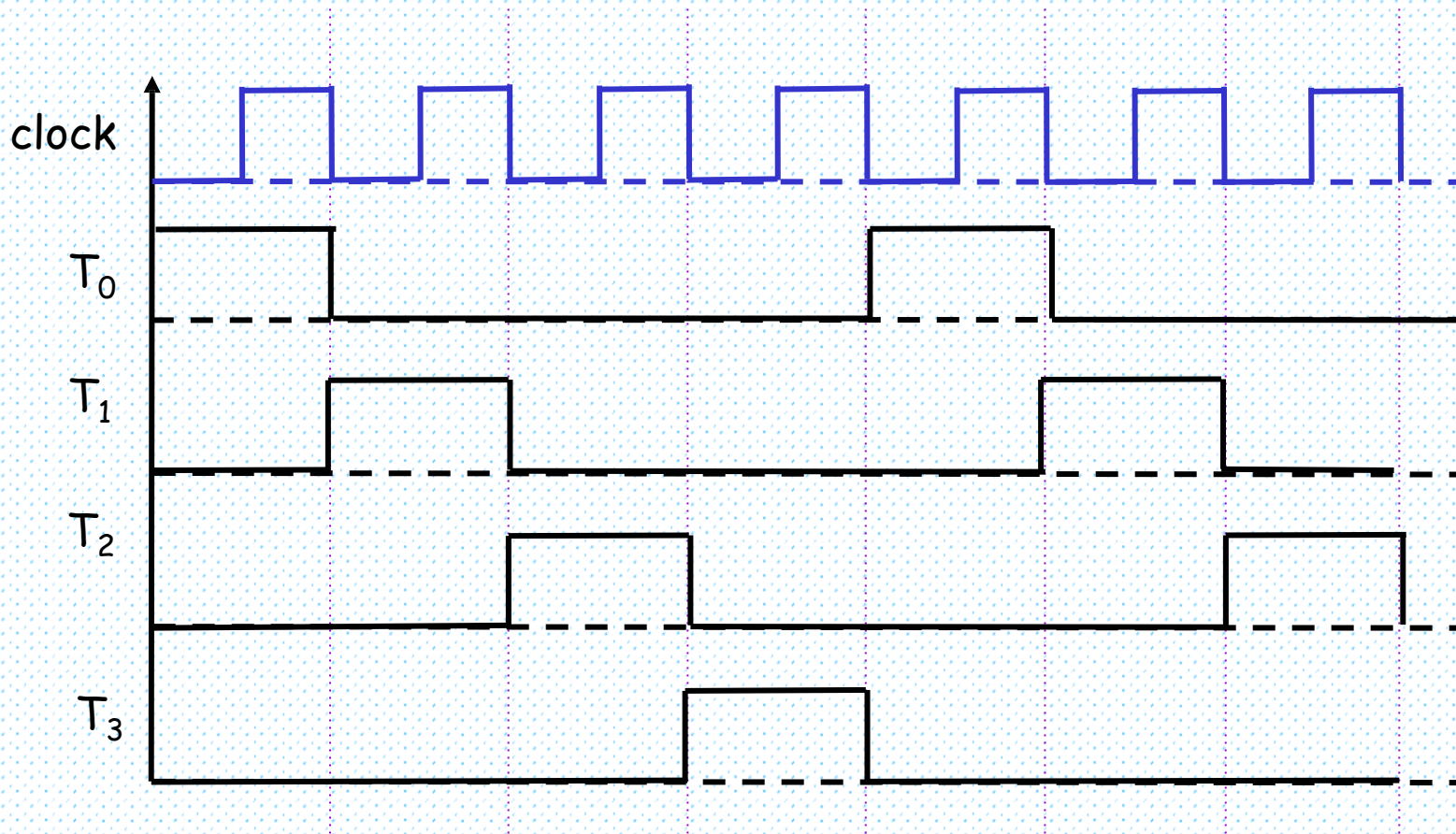
- Ring Counter

- Timing signals control the sequence of operations in a digital system
- A ring counter is a circular shift register with only one flip-flop being set at any particular time, all others are cleared.



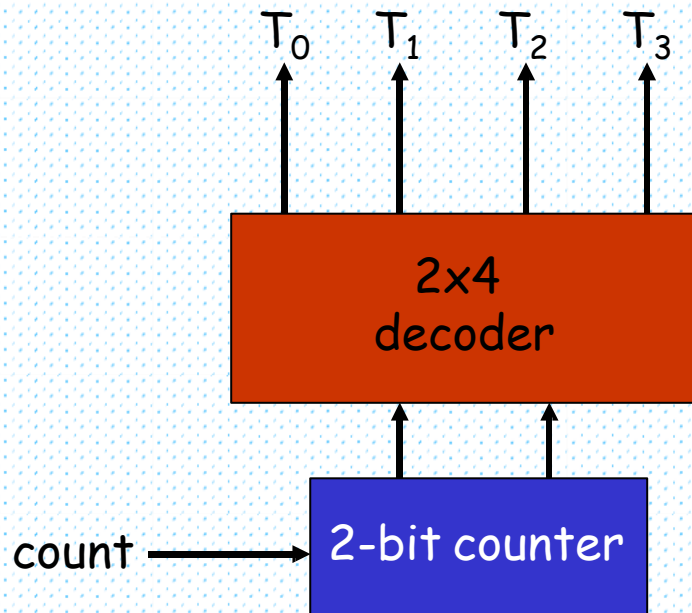
# Ring Counter

- Sequence of timing signals



# Ring Counter

- To generate  $2^n$  timing signals,
  - we need a shift register with  $2^n$  flip-flops
- or, we can construct the ring counter with a binary counter and a decoder



## Cost:

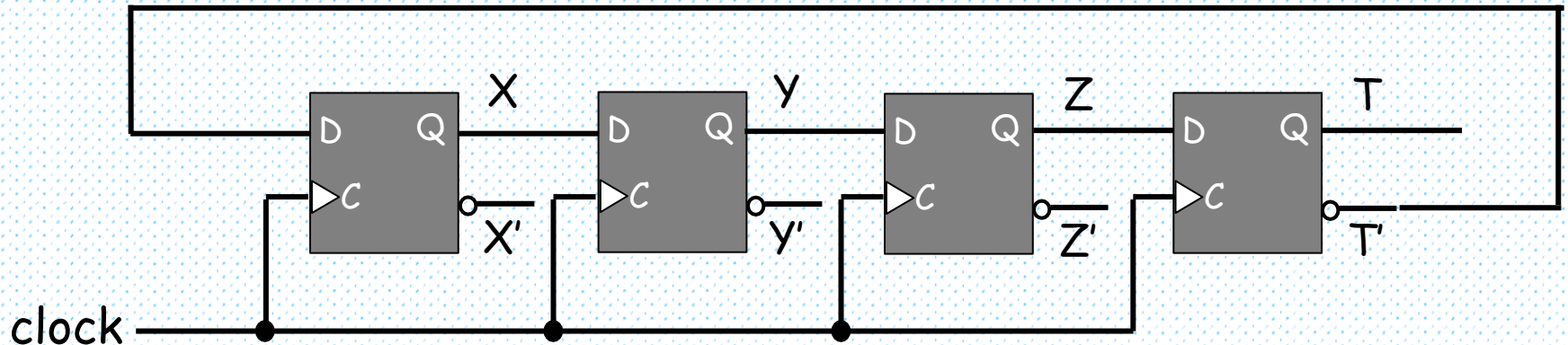
- 2 flip-flop
- 2-to-4 line decoder

## Cost in general case:

- $n$  flip-flops
- $n$ -to- $2^n$  line decoder

# Johnson Counter

- A k-bit ring counter can generate k distinguishable states
- The number of states can be doubled if the shift register is connected as a switch-tail ring counter



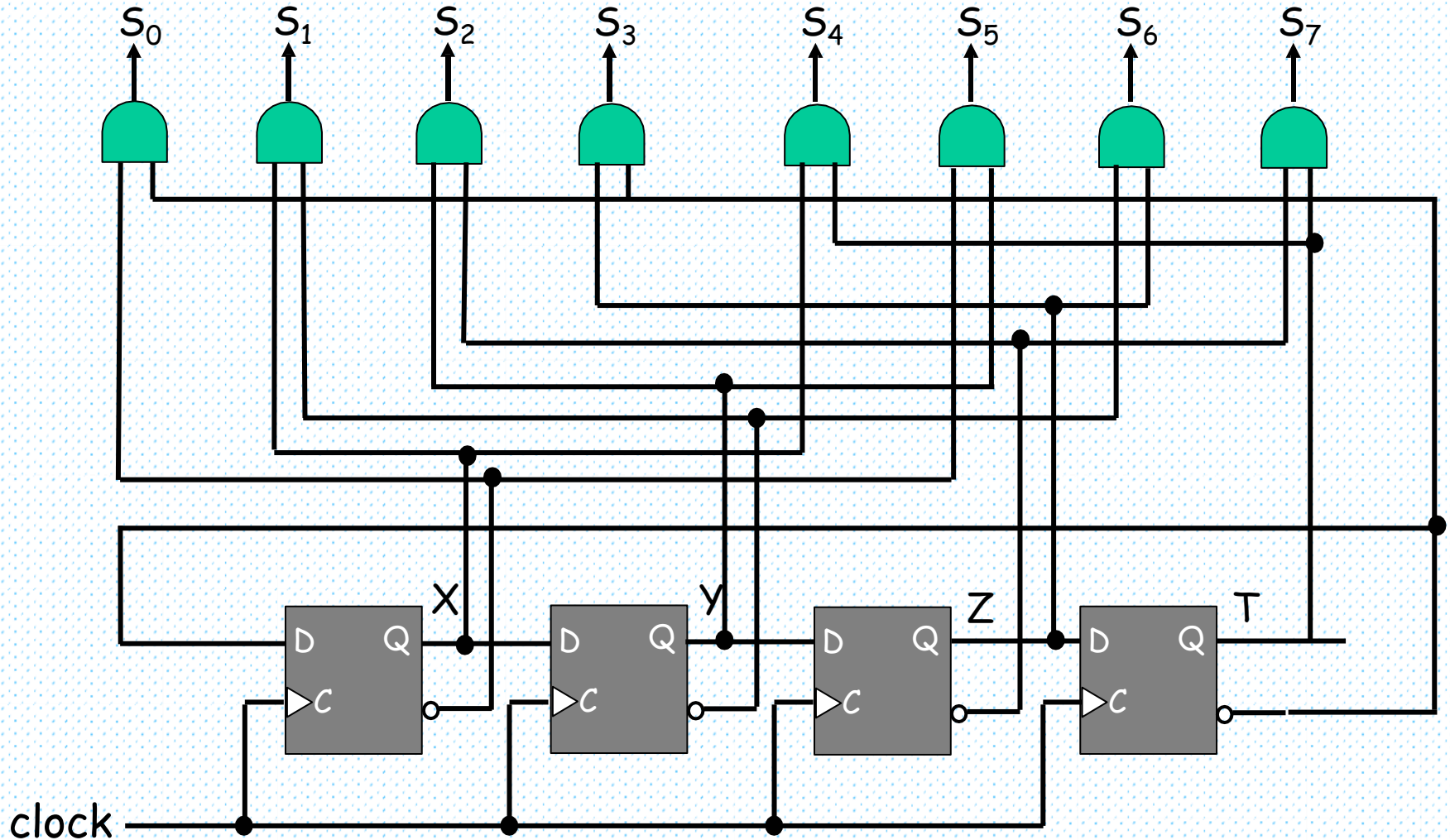
# Johnson Counter

- Count sequence and required decoding

sequence number	Flip-flop outputs				Output
	X	Y	Z	T	
1	0	0	0	0	$X'T'$
2	1	0	0	0	$XY'$
3	1	1	0	0	$YZ'$
4	1	1	1	0	$ZT'$
5	1	1	1	1	$XT$
6	0	1	1	1	$X'Y$
7	0	0	1	1	$Y'Z$
8	0	0	0	1	$Z'T$

# Johnson Counter

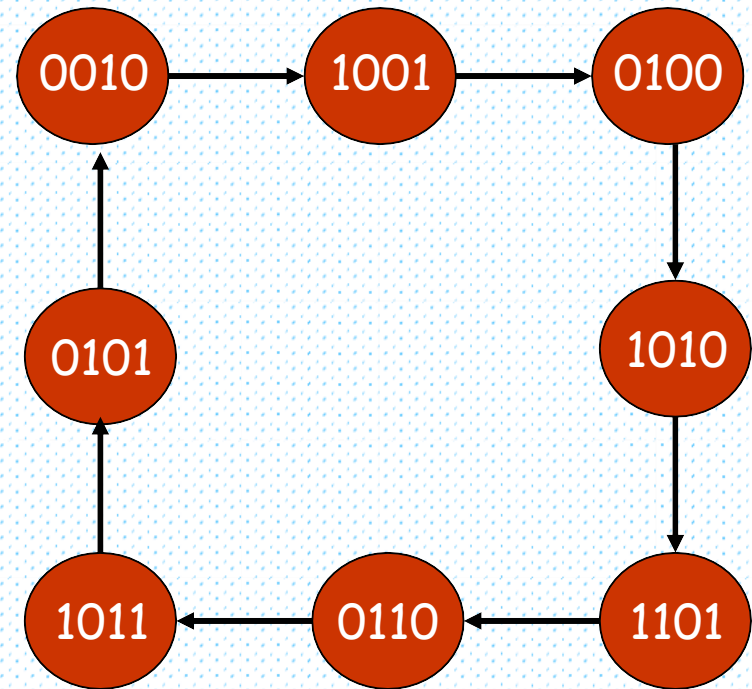
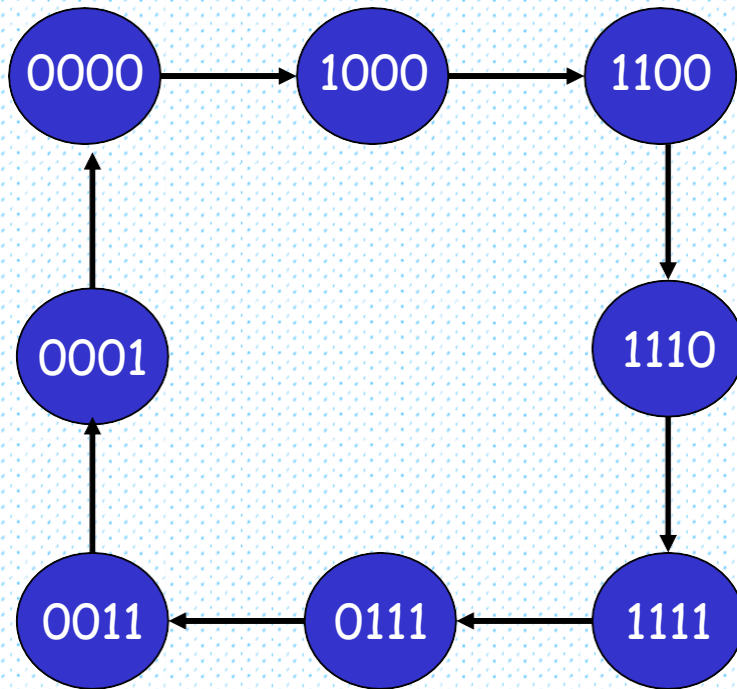
- Decoding circuit





# Unused States in Counters

- 4-bit Johnson counter



# Johnson Counter

Inputs				Outputs			
X	Y	Z	T	X	Y	Z	T
0	0	0	0	1	0	0	0
1	0	0	0	1	1	0	0
1	1	0	0	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1
0	0	1	1	0	0	0	1
0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	1
1	1	0	1	0	1	1	0
0	1	1	0	1	0	1	1
1	0	1	1	0	1	0	1
0	1	0	1	0	0	0	0
0	0	1	0	1	0	0	1
1	0	0	1	0	1	0	0
0	1	0	0	1	0	0	0

# K-Maps

		ZT			
		00	01	11	10
XY	00	1			1
	01	1			1
	11	1			1
	10	1			1

$$X = T'$$

		ZT			
		00	01	11	10
XY	00				
	01				
	11	1	1	1	1
	10	1	1	1	1

$$Y = X$$

		ZT			
		00	01	11	10
XY	00				
	01			1	1
	11	1	1	1	1
	10				

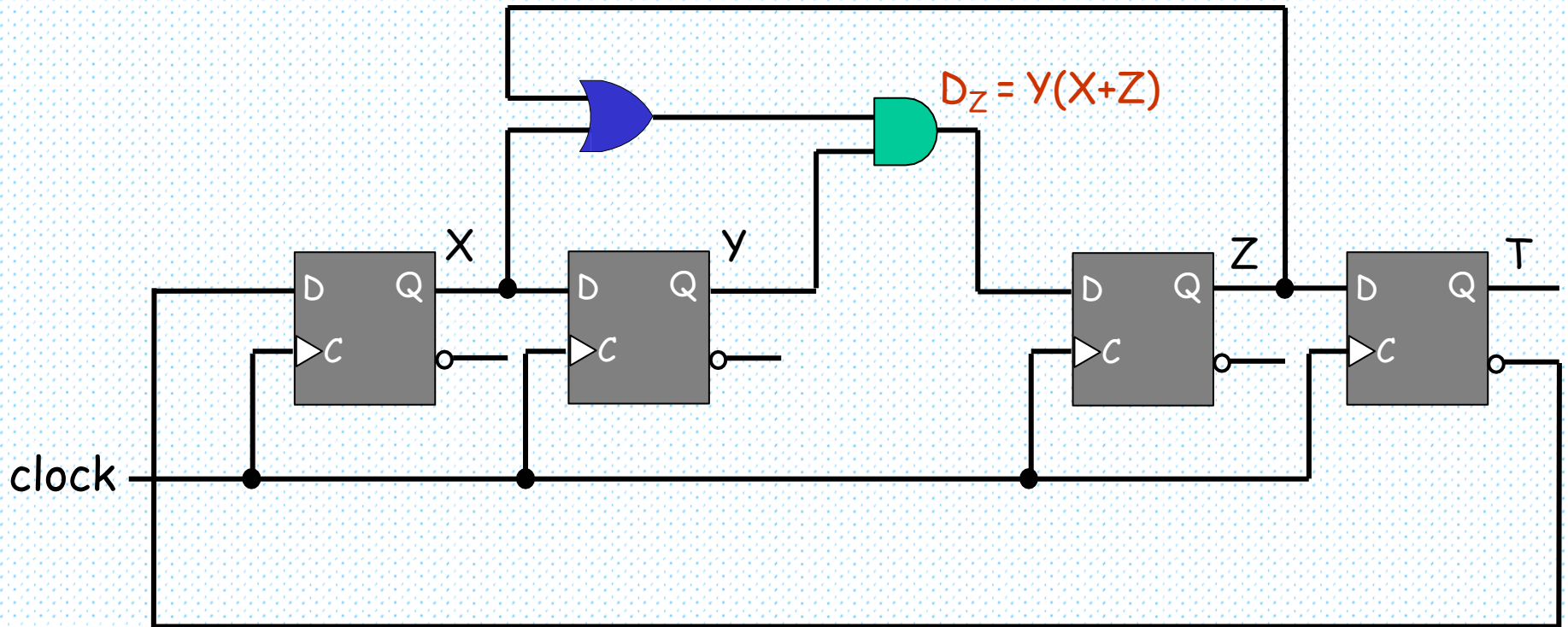
$$Z = XY + YZ$$

		ZT			
		00	01	11	10
XY	00			1	1
	01			1	1
	11			1	1
	10			1	1

$$T = Z$$

# Unused States in Counters

- Remedy



# Unused States in Counters

- State diagram

