

Digital Systems Design and Laboratory

[12. Registers and Counters]

Chung-Wei Lin

cwlin@csie.ntu.edu.tw

CSIE Department

National Taiwan University

Sequential Logic Design

❑ Unit 11: Latches and Flip-Flops

- Basic unit

❑ Unit 12: Registers and Counters

- Simple sequential circuit

❑ Units 13--15: Finite State Machines

- Complex sequential circuit

❑ Unit 16: Summary

- Put it all together

Outline

☐ **Registers and Register Transfers**

☐ Shift Registers

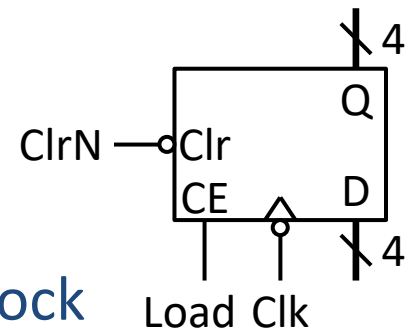
☐ Design of Binary Counters

☐ Counters for Other Sequences

☐ Counter Design Using S-R and J-K Flip-Flops

☐ Derivation of Flip-Flop Input Equations

Registers (1/2)



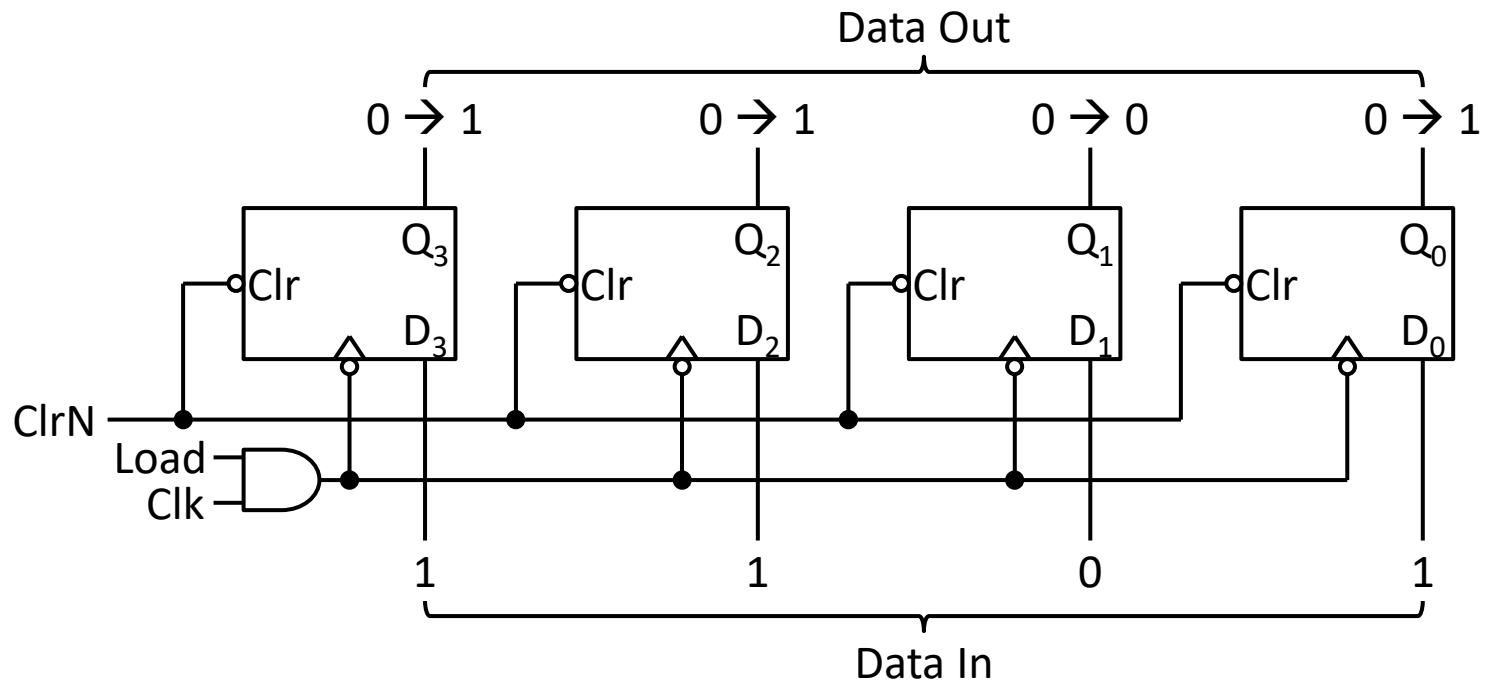
❑ Register: a group of D flip-flops with a common clock

❑ Example

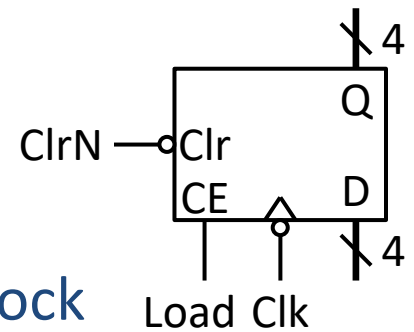
➤ 4-bit D flip-flop registers with Data, Load, Clear (ClrN), Clock (Clk)

❑ First Implementation: gated clock

➤ When Load = 1, load data at D to Q at Clk falling



Registers (2/2)



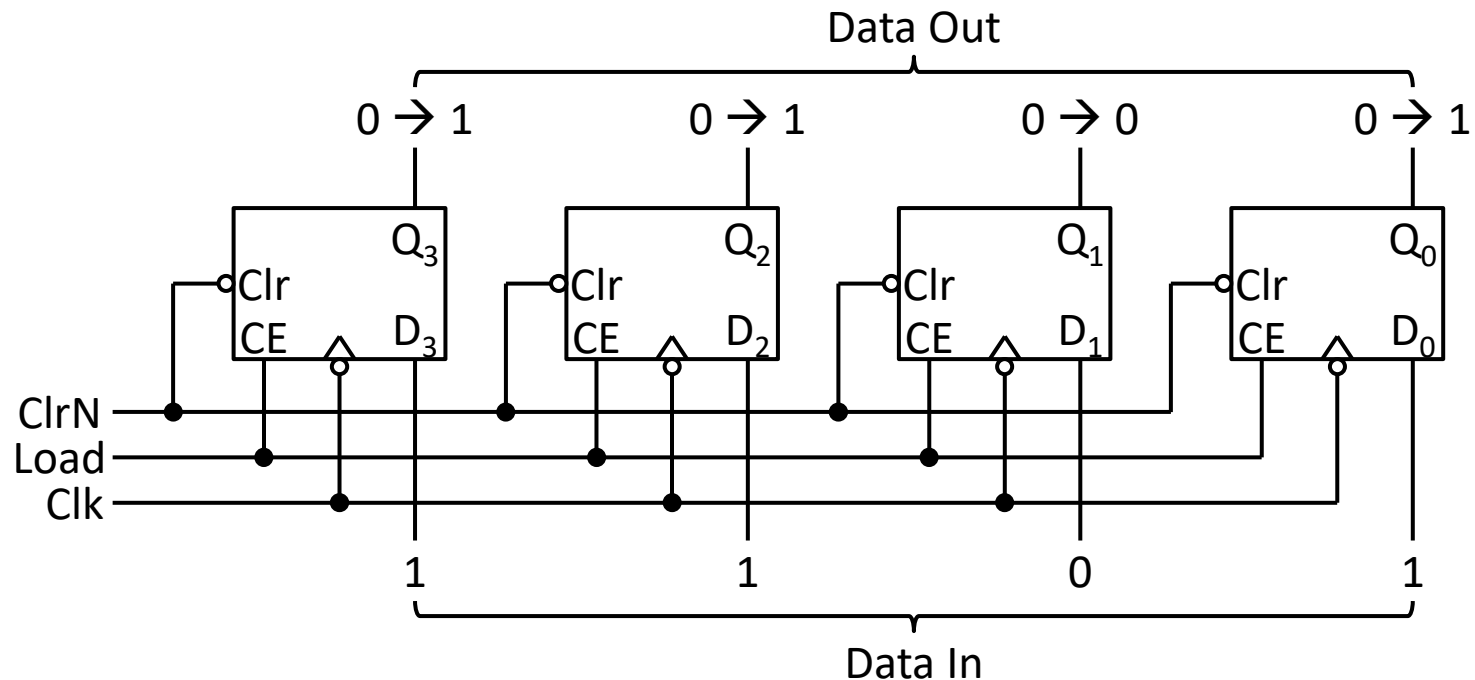
❑ Register: a group of D flip-flops with a common clock

❑ Example

➤ 4-bit D flip-flop registers with Data, Load, Clear (ClrN), Clock (Clk)

❑ Second implementation: clock enable

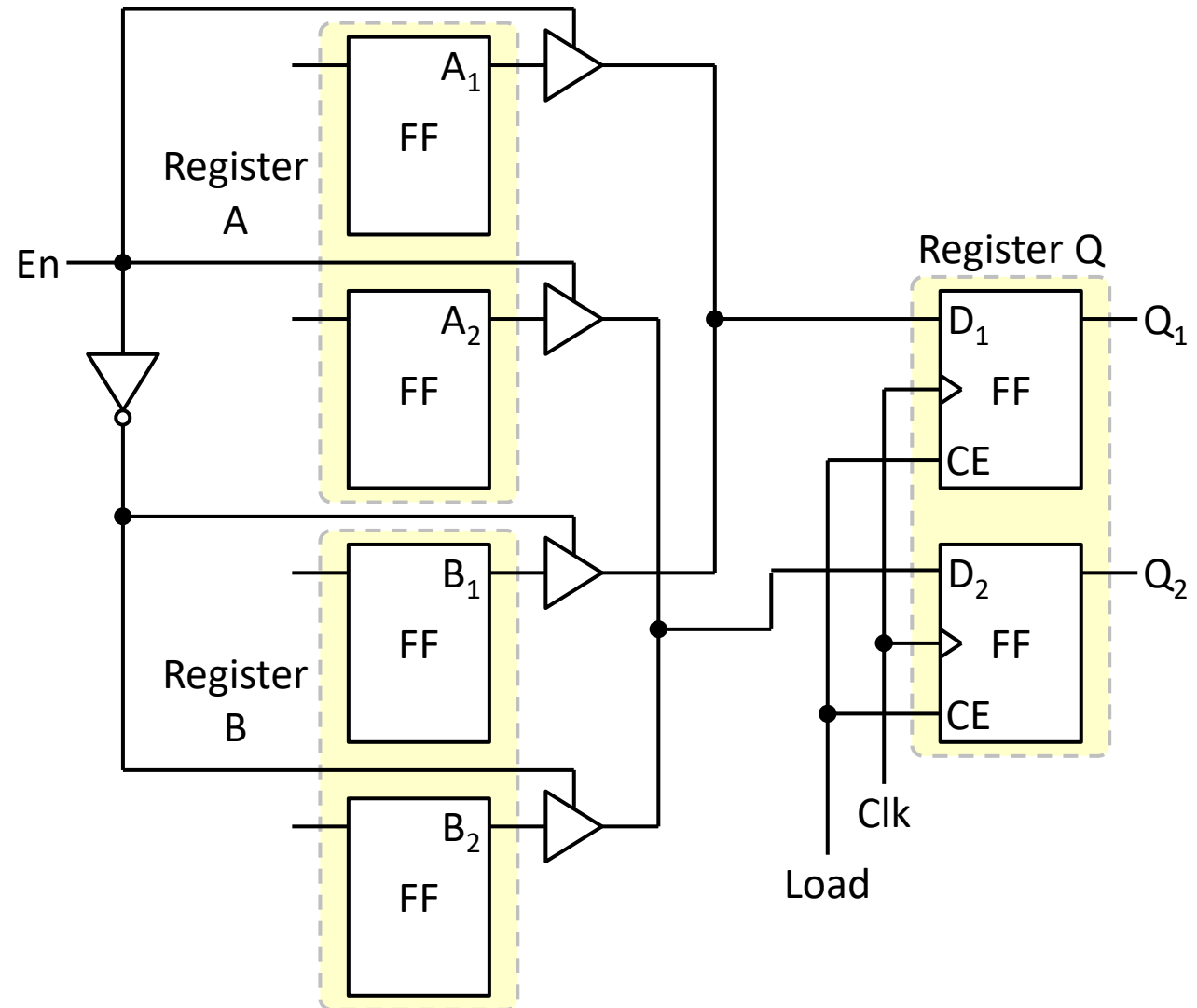
➤ When Load = 1, load data at D to Q at Clk falling



Data Transfer between Registers

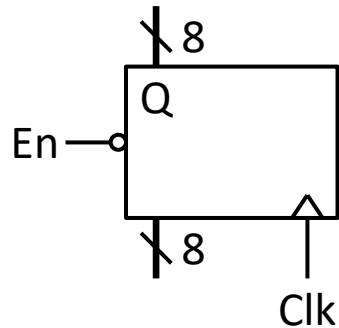
2-to-1 MUX

- If $En = 1$, $Q = A$
- If $En = 0$, $Q = B$

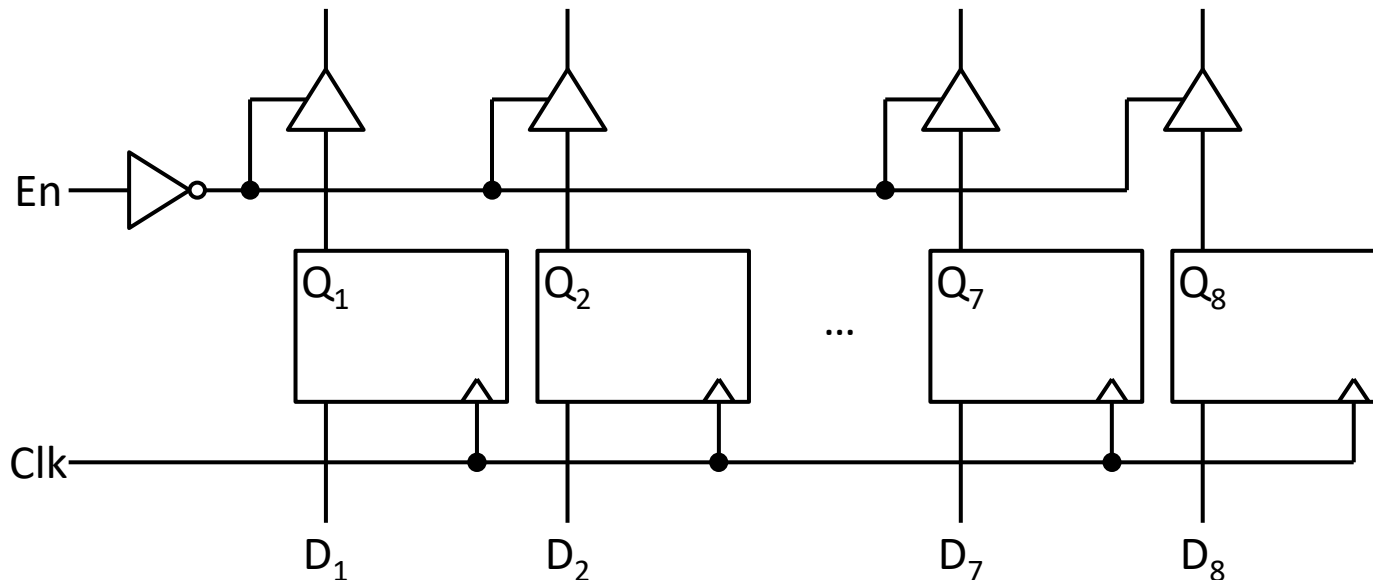


8-Bit Register with Tri-State Output (1/2)

□ Symbol

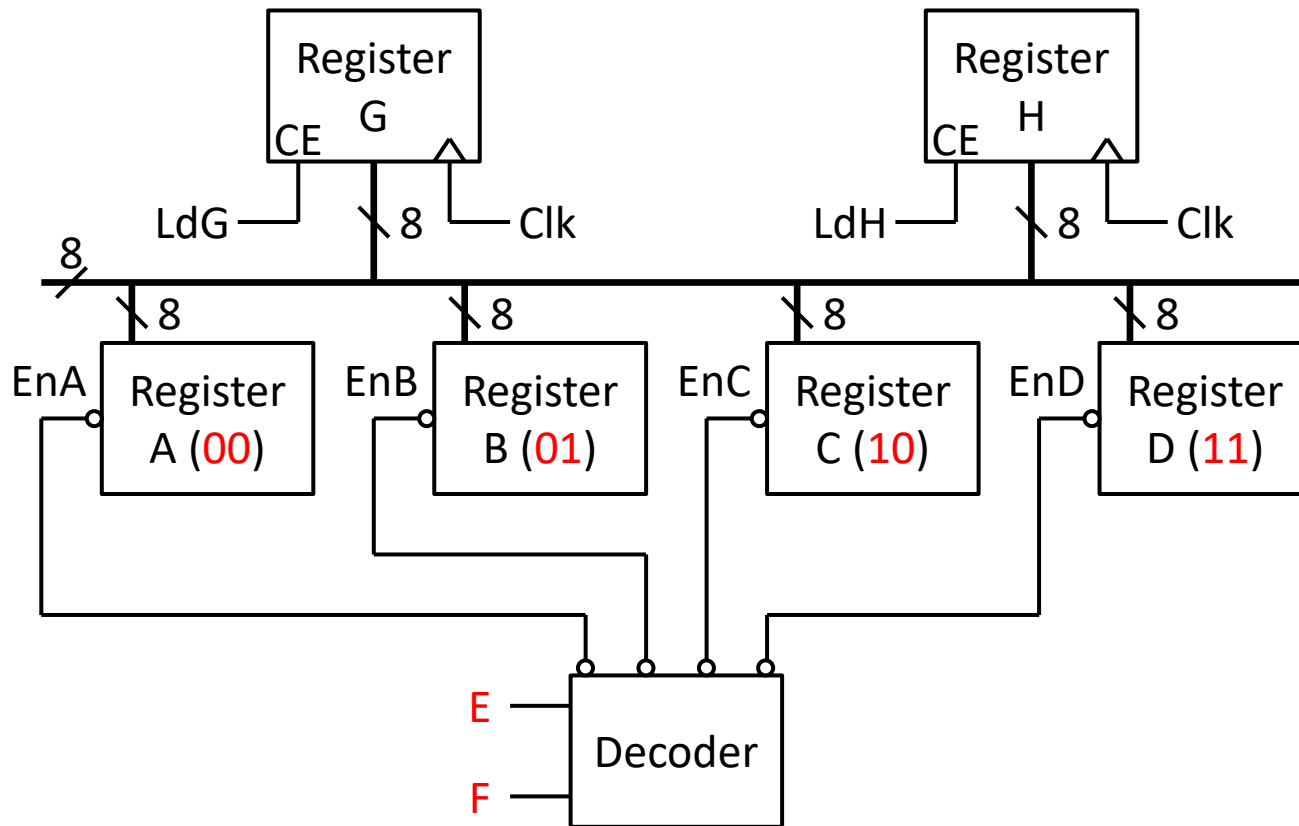


□ Logic diagram



8-Bit Register with Tri-State Output (2/2)

□ Data transfer

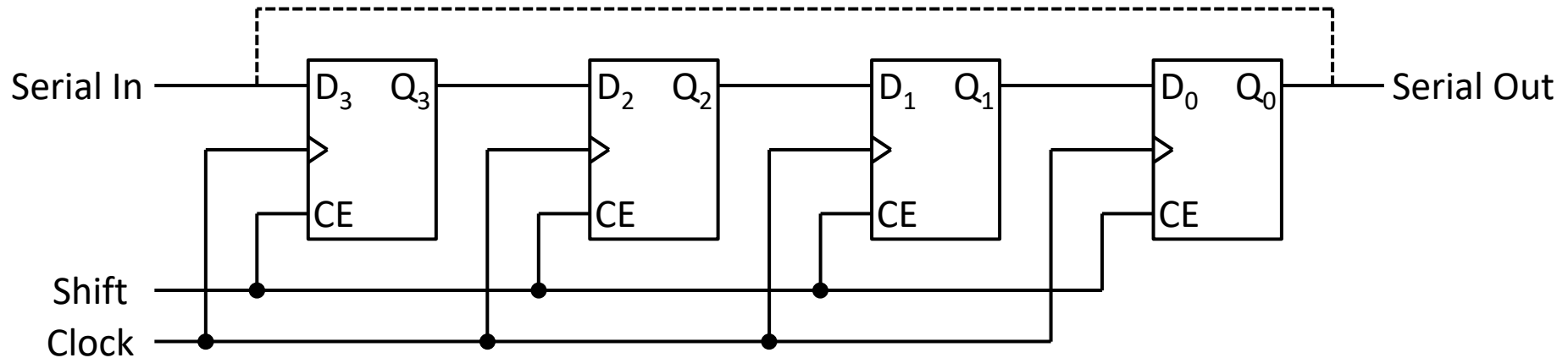


Outline

- ❑ Registers and Register Transfers
- ❑ **Shift Registers**
- ❑ Design of Binary Counters
- ❑ Counters for Other Sequences
- ❑ Counter Design Using S-R and J-K Flip-Flops
- ❑ Derivation of Flip-Flop Input Equations

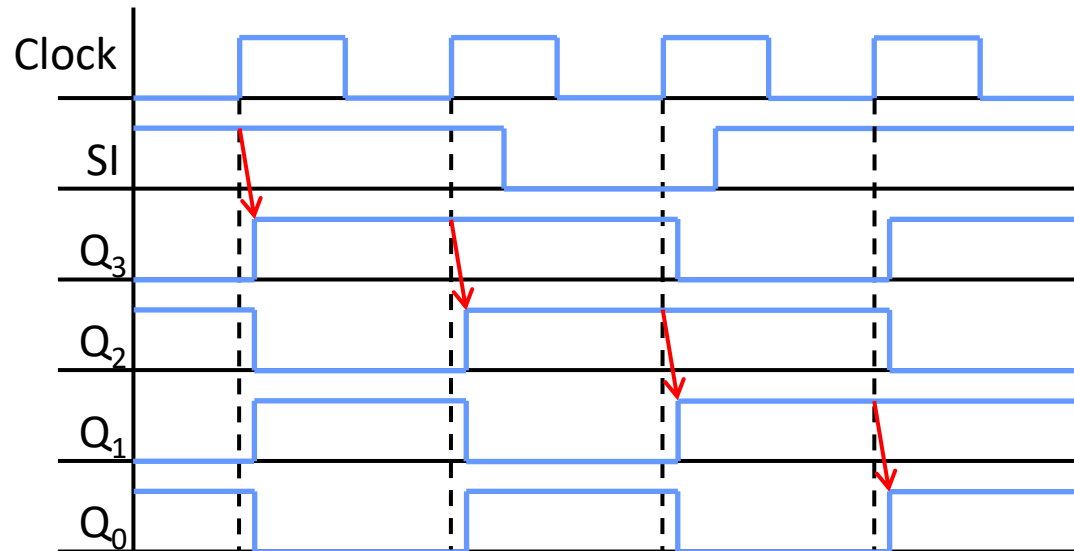
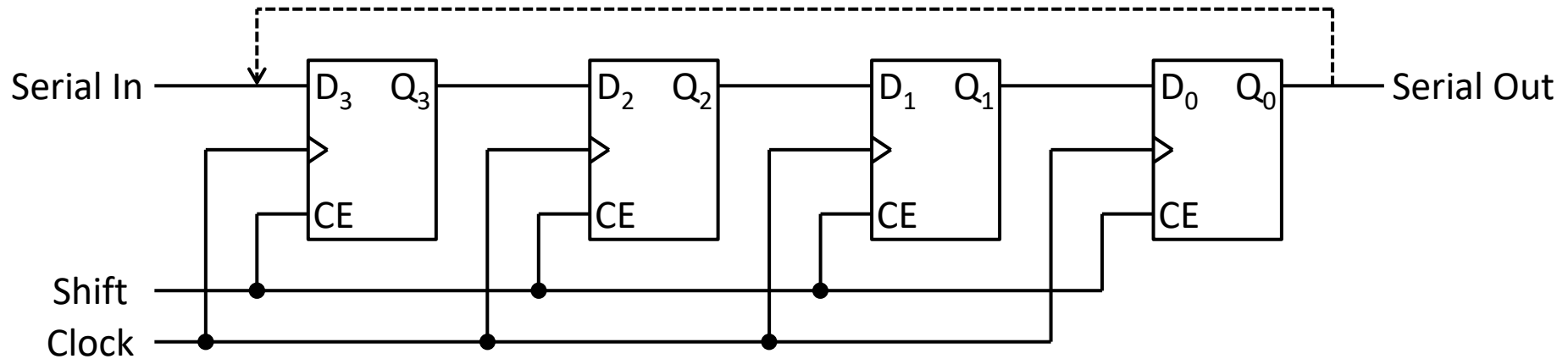
Shift Registers (1/2)

- ❑ Shift register: a group of flip-flops where binary data can be stored and shifted left or right when a shift signal is applied
- ❑ Example: 4-bit right-shift register



Shift Registers (2/2)

□ Timing diagram of a 4-bit right-shift register



Initial $Q_3Q_2Q_1Q_0 = 0101$

SI = 1, 1, 0, 1

$Q_3Q_2Q_1Q_0 =$ 0101

1010

1101

0110

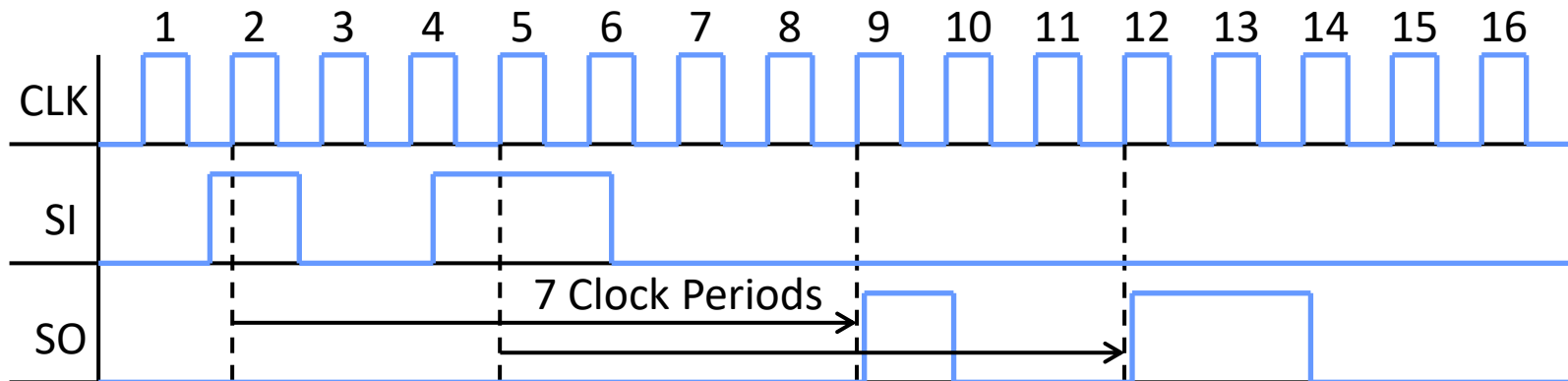
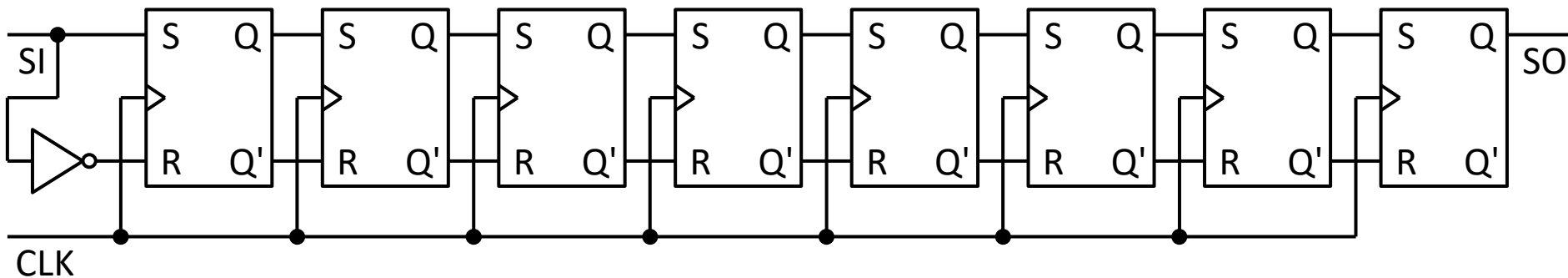
1011

N-bit Serial-In Serial-Out Shift Registers

❑ Take $(n-1)$ cycles to output data

➤ SI: Serial In

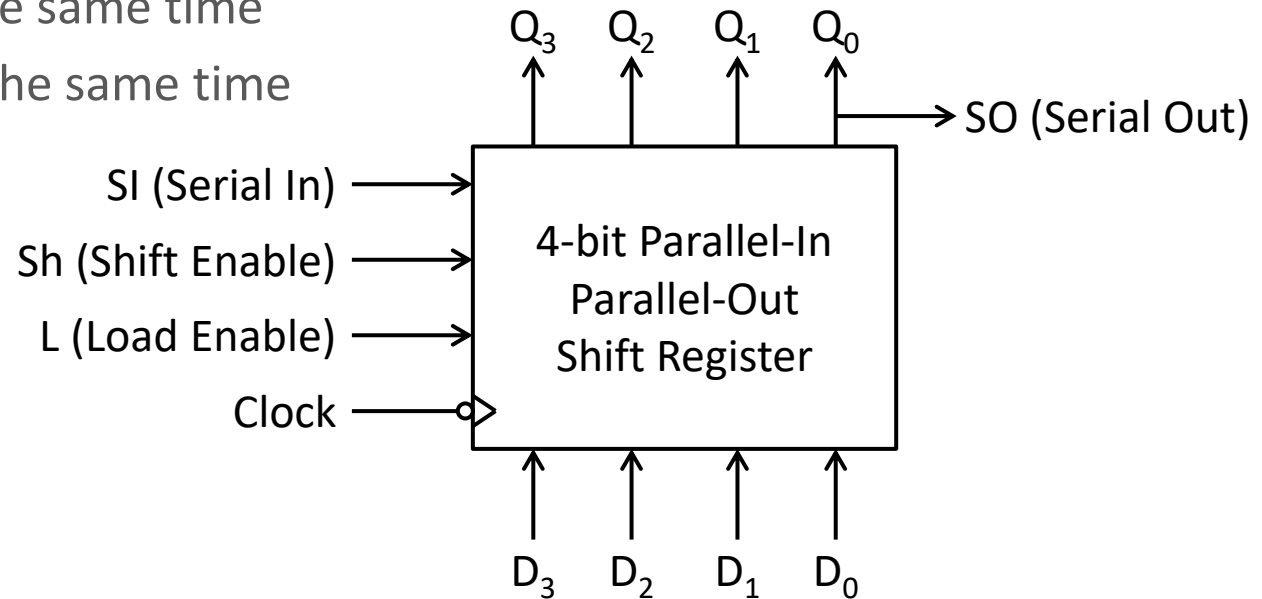
➤ SO: Serial Out



Parallel-In Parallel-Out Right Shift Register (1/2)

□ Parallel-in parallel-out (PIPO)

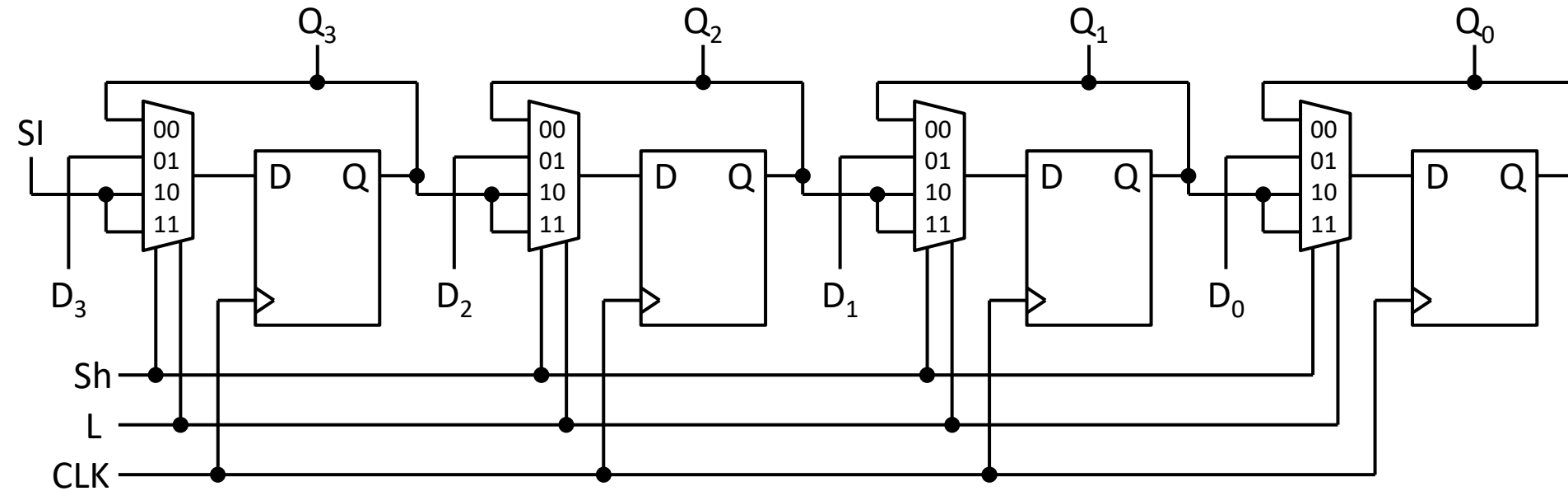
- Load all data at the same time
- Read out data at the same time



Sh (Shift)	L (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	Action
0	0	Q_3	Q_2	Q_1	Q_0	No Change
0	1	D_3	D_2	D_1	D_0	Load
1	X	SI	Q_3	Q_2	Q_1	Right Shift

Parallel-In Parallel-Out Right Shift Register (2/2)

□ Implement using flip-flops and MUXes

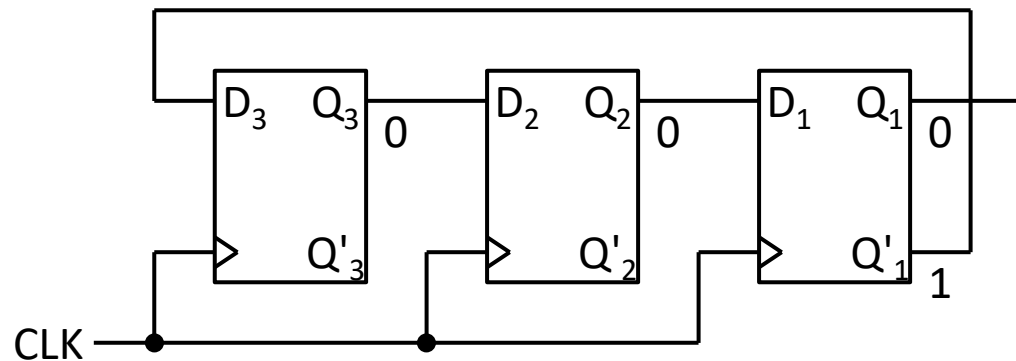


Sh (Shift)	L (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	Action
0	0	Q_3	Q_2	Q_1	Q_0	No Change
0	1	D_3	D_2	D_1	D_0	Load
1	X	SI	Q_3	Q_2	Q_1	Right Shift

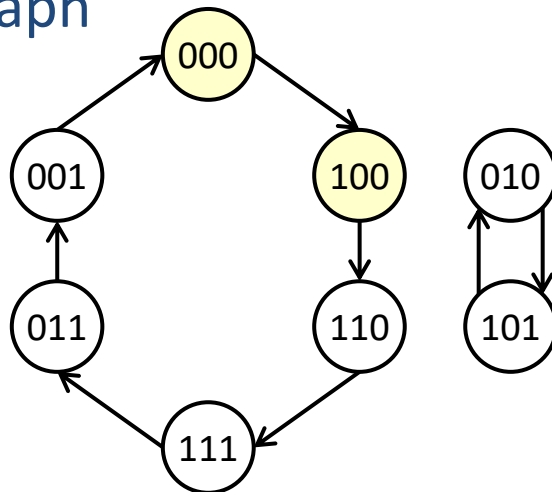
Shift Register with Inverted Feedback

❑ Johnson counter: a shift register with inverted feedback

➤ Counter: a circuit that cycles through a fixed sequence of states



❑ State graph



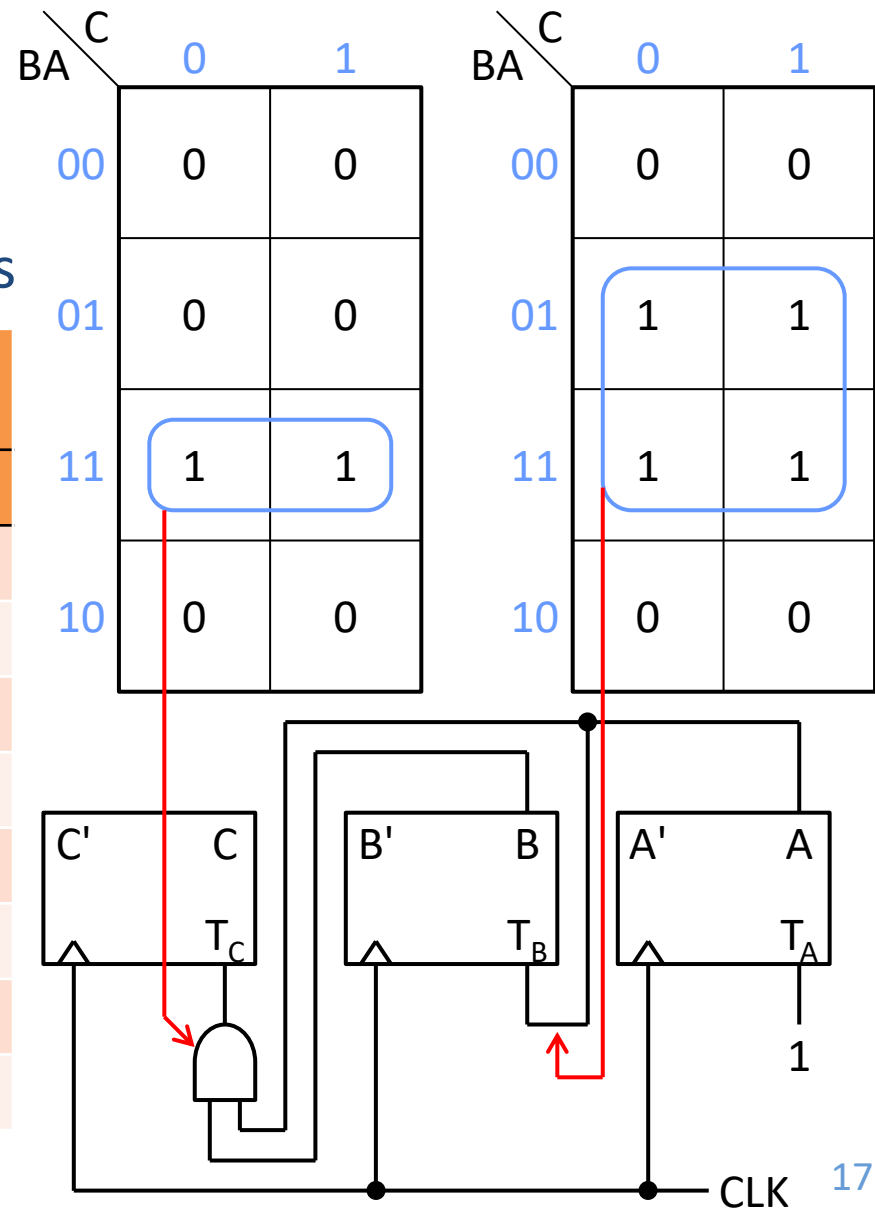
Outline

- ❑ Registers and Register Transfers
- ❑ Shift Registers
- ❑ **Design of Binary Counters**
- ❑ Counters for Other Sequences
- ❑ Counter Design Using S-R and J-K Flip-Flops
- ❑ Derivation of Flip-Flop Input Equations

Counting 0--7 (1/2)

- ❑ **Synchronous** counter: flip-flops are synchronized by a clock
- ❑ First implementation: T flip-flops

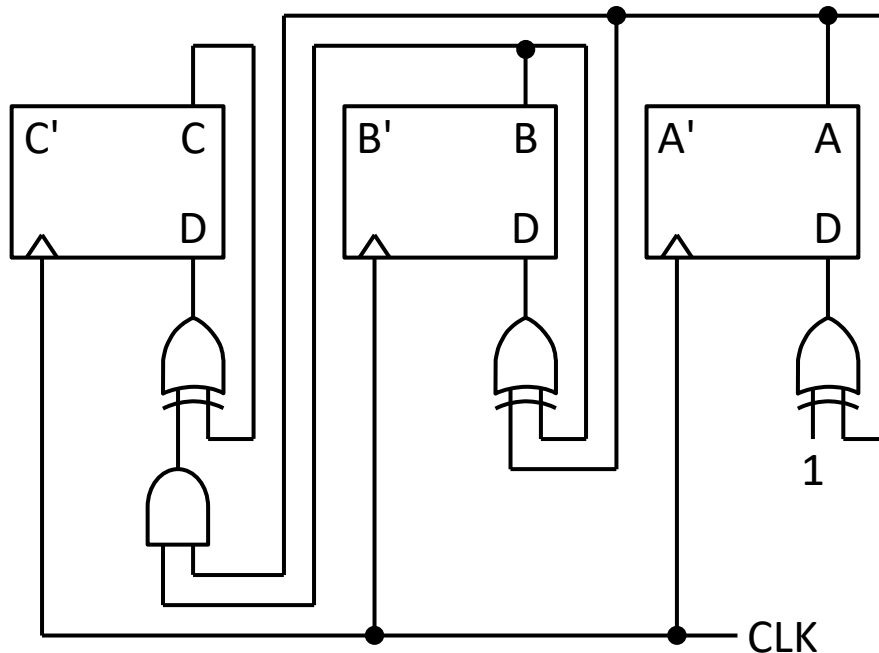
Present State			Next State			Flip-Flop Inputs (By Observation)		
C	B	A	C ⁺	B ⁺	A ⁺	T _C	T _B	T _A
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1



Counting 0--7 (2/2)

❑ Second implementation: D flip-flops

- $D_A = A^+ = A'$
- $D_B = B^+ = BA' + B'A = B \oplus A$
 - B changes when $A = 1$
- $D_C = C^+ = C'BA + CB' + CA' = C \oplus BA$
 - C changes when $A = B = 1$

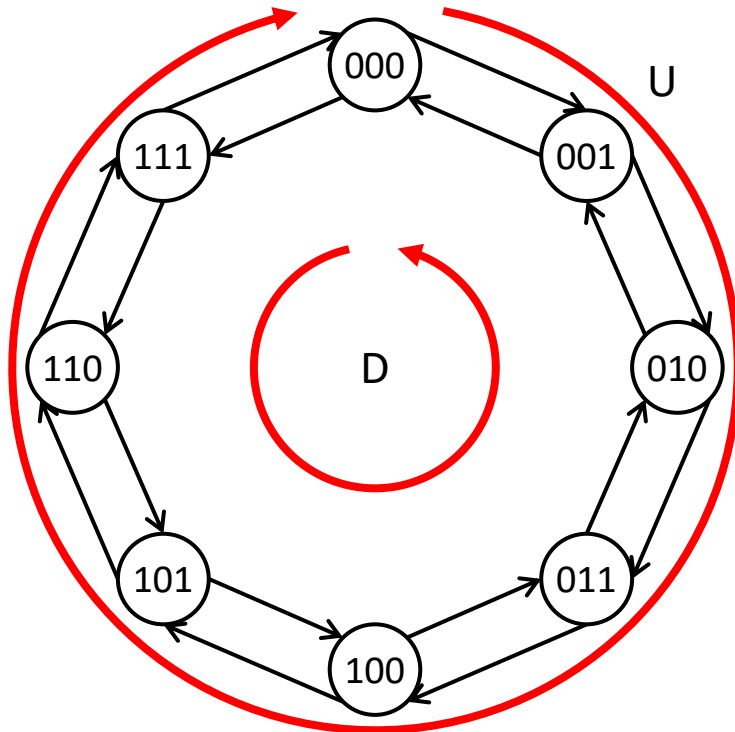


Present State			Next State		
C	B	A	C ⁺	B ⁺	A ⁺
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Up-Down Counter

□ U and D control "up" and "down"

- Do not allow $U = D = 1$
- $D_A = A^+ = A \oplus (U + D)$
- $D_B = B^+ = B \oplus (UA + DA')$
- $D_C = C^+ = C \oplus (UBA + DB'A')$



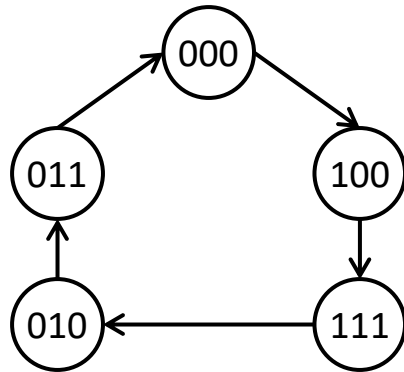
CBA	C ⁺ B ⁺ A ⁺	
	U	D
000	001	111
001	010	000
010	011	001
011	100	010
100	101	011
101	110	100
110	111	101
111	000	110

Outline

- ❑ Registers and Register Transfers
- ❑ Shift Registers
- ❑ Design of Binary Counters
- ❑ **Counters for Other Sequences**
- ❑ Counter Design Using S-R and J-K Flip-Flops
- ❑ Derivation of Flip-Flop Input Equations

State Diagram of Counter

❑ What if the sequence is not in straight binary order?



Present State			Next State		
C	B	A	C ⁺	B ⁺	A ⁺
0	0	0	1	0	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	0	1	0

K-Map Derivation

□ Next states

BA \ C		0	1
00	00	1	1
01	01	X	X
11	11	0	0
10	10	0	X

C^+

BA \ C		0	1
00	00	0	1
01	01	X	X
11	11	0	1
10	10	1	X

B^+

BA \ C		0	1
00	00	0	1
01	01	X	X
11	11	0	0
10	10	1	X

A^+

C	B	A	C^+	B^+	A^+
0	0	0	1	0	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	0	1	0

Implementation: T Flip-Flops (1/2)

□ T inputs: $T = Q \oplus Q^+$

BA \ C		0	1
		0	1
00	1	0	
01	X	X	
11	0	1	
10	0	X	

$$T_C = C'B' + CB$$

BA \ C		0	1
		0	1
00	0	1	
01	X	X	
11	1	0	
10	0	X	

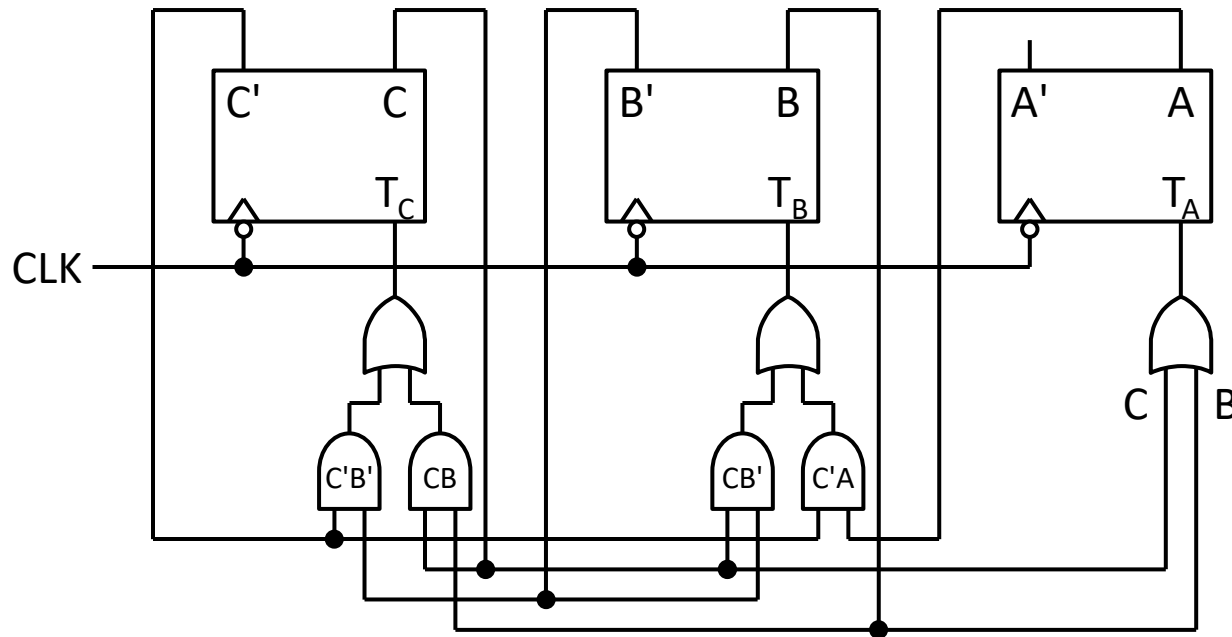
$$T_B = C'A + CB'$$

BA \ C		0	1
		0	1
00	0	1	
01	X	X	
11	1	1	
10	1	X	

$$T_A = C + B$$

C	B	A	C ⁺	B ⁺	A ⁺
0	0	0	1	0	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	0	1	0

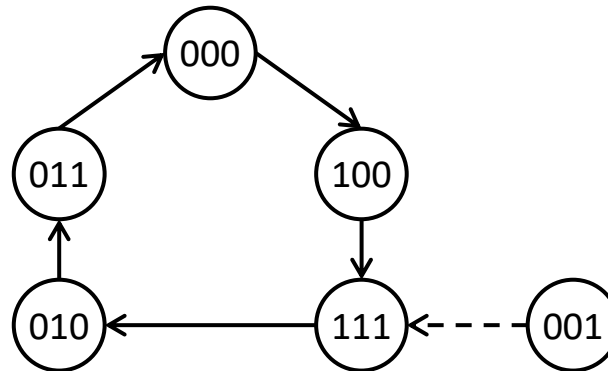
Implementation: T Flip-Flops (2/2)



Don't Care States

❑ If flip-flops are initially set to CBA = 001

- Tracking signals through the network shows that $T_C = T_B = 1$ and $T_A = 0$, so the state changes to 111



❑ When the power is turned on, the initial states of all flip-flops are unpredictable!!

- Don't care states should be checked to make sure that they eventually lead into the main counting sequence
- Or use power-up reset

Implementation: D Flip-Flops (1/2)

□ Next states

		C	
		0	1
BA	00	1	1
	01	X	X
11	11	0	0
	10	0	X

$$D_C = B'$$

		C	
		0	1
BA	00	0	1
	01	X	X
11	11	0	1
	10	1	X

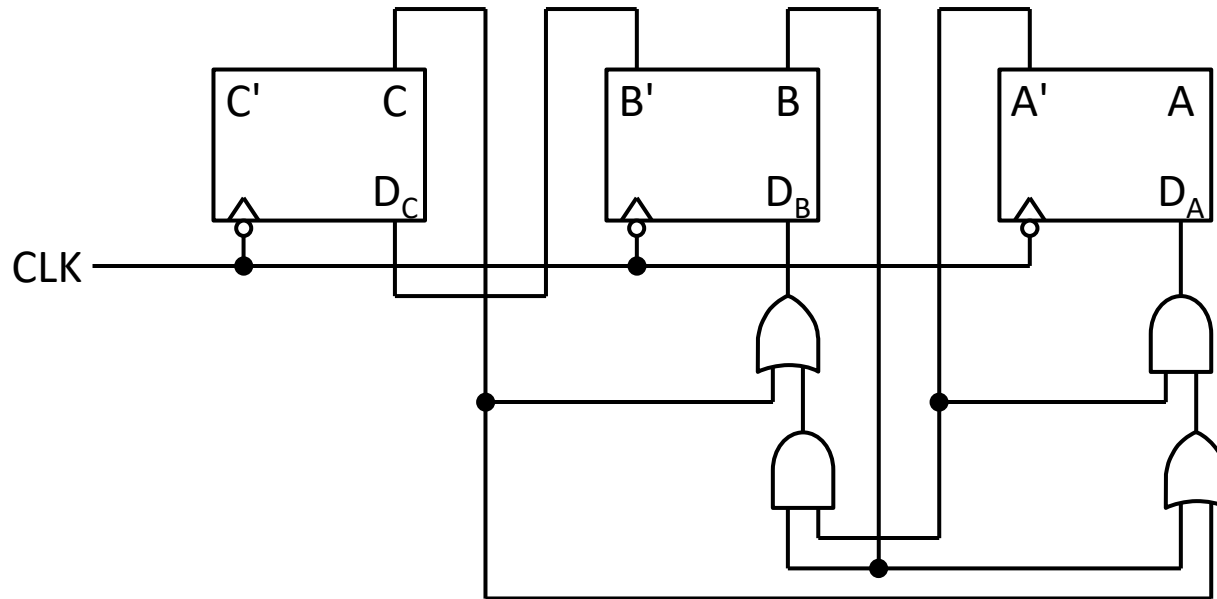
$$D_B = C + BA'$$

		C	
		0	1
BA	00	0	1
	01	X	X
11	11	0	0
	10	1	X

$$D_A = CA' + BA'$$

C	B	A	C ⁺	B ⁺	A ⁺
0	0	0	1	0	0
0	0	1	-	-	-
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	1	1
1	0	1	-	-	-
1	1	0	-	-	-
1	1	1	0	1	0

Implementation: D Flip-Flops (2/2)



Outline

- ❑ Registers and Register Transfers
- ❑ Shift Registers
- ❑ Design of Binary Counters
- ❑ Counters for Other Sequences
- ❑ **Counter Design Using S-R and J-K Flip-Flops**
- ❑ Derivation of Flip-Flop Input Equations

Recap: S-R Flip-Flops

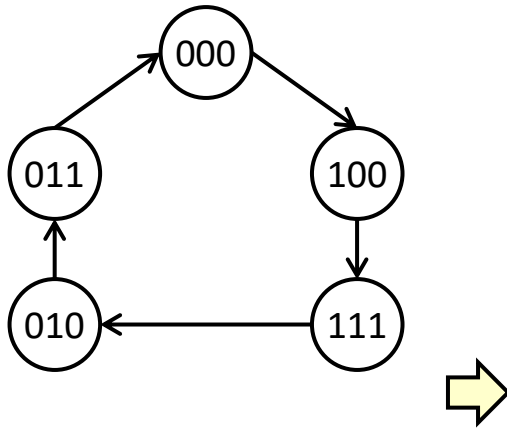
□ What is the relation between S, R and Q, Q⁺?

➤ We do it reversely from Q and Q⁺ to S and R

S	R	Q	Q ⁺		Q	Q ⁺	S	R		Q	Q ⁺	S	R
0	0	0	0	} Unchanged	0	0	0	0	➡	0	0	0	X
0	0	1	1		0	1	0	1		0	1	0	1
0	1	0	0	} Reset to 0	0	1	1	0		0	1	1	0
0	1	1	0		0	1	1	0		0	1	1	0
1	0	0	1	} Set to 1	1	0	0	1		1	0	0	1
1	0	1	1		1	0	0	1		1	0	0	1
1	1	0	X	} Inputs Not Allowed	1	1	0	0		1	1	X	0
1	1	1	X		1	1	1	0		1	1	1	X
					Excitation Table								

Using S-R Flip-Flops (1/2)

□ Derive S-R flip-flop inputs from the excitation table



Q	Q ⁺	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

C	B	A	C ⁺	B ⁺	A ⁺	S _C	R _C	S _B	R _B	S _A	R _A
0	0	0	1	0	0	1	0	0	X	0	X
0	0	1	-	-	-	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	0
0	1	1	0	0	0	0	X	0	1	0	1
1	0	0	1	1	1	X	0	1	0	1	0
1	0	1	-	-	-	X	X	X	X	X	X
1	1	0	-	-	-	X	X	X	X	X	X
1	1	1	0	1	0	0	1	X	0	0	1

By Karnaugh maps

$$S_C = B', R_C = A, S_B = C, R_B = C'A, S_A = CA' + BA', R_A = A$$

Using S-R Flip-Flops (2/2)

- Alternative: derive S-R flip-flop inputs with K-maps (faster?)

Q	Q ⁺	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

		C	
		0	1
BA	00	1	1
	01	X	X
	11	0	0
	10	0	X
		C ⁺	



		C	
		0	1
BA	00	1	X
	01	X	X
	11	0	0
	10	0	X
		S _C = B'	

		C	
		0	1
BA	00	0	0
	01	X	X
	11	X	1
	10	X	X
		R _C = A	

Recap: J-K Flip-Flops

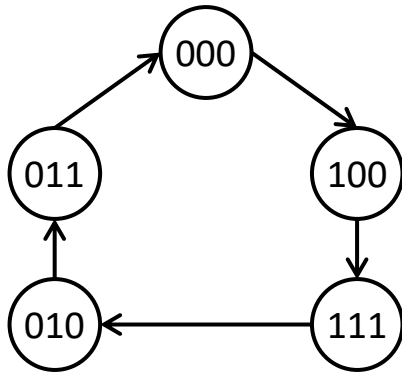
❑ What is the relation between J, K and Q, Q⁺?

➤ We do it reversely from Q and Q⁺ to J and K

J	K	Q	Q ⁺		Q	Q ⁺	J	K		Q	Q ⁺	J	K
0	0	0	0	} Unchanged	0	0	0	0	} Unchanged	0	0	0	X
0	0	1	1		0	0	0	1		0	0	0	1
0	1	0	0	} Reset to 0	0	1	1	0	} Reset to 0	0	1	1	X
0	1	1	0		0	1	1	1		0	1	1	1
1	0	0	1	} Set to 1	1	0	0	1	} Set to 1	1	0	X	1
1	0	1	1		1	0	1	1		1	0	X	1
1	1	0	1	} Toggle	1	1	0	0	} Toggle	1	1	X	0
1	1	1	0		1	1	1	0		1	1	X	0

Using J-K Flip-Flops

- Derive J-K flip-flop inputs from the excitation table



Q	Q ⁺	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



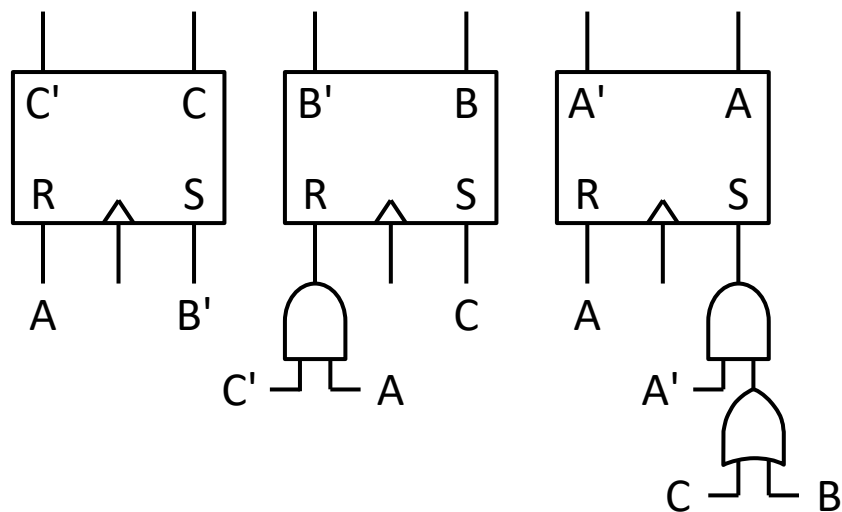
C	B	A	C ⁺	B ⁺	A ⁺	J _C	K _C	J _B	K _B	J _A	K _A
0	0	0	1	0	0	1	X	0	X	0	X
0	0	1	-	-	-	X	X	X	X	X	X
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	0	0	0	0	X	X	1	X	1
1	0	0	1	1	1	X	0	1	X	1	X
1	0	1	-	-	-	X	X	X	X	X	X
1	1	0	-	-	-	X	X	X	X	X	X
1	1	1	0	1	0	X	1	X	0	X	1

By Karnaugh maps

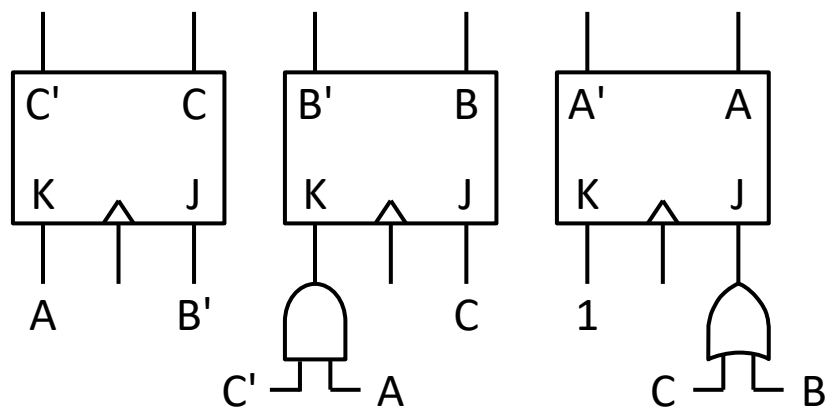
$$J_C = B', K_C = A, J_B = C, K_B = C'A, J_A = C + B, K_A = 1$$

Implementation

□ S-R flip-flops



□ J-K flip-flops



Outline

- ❑ Registers and Register Transfers
- ❑ Shift Registers
- ❑ Design of Binary Counters
- ❑ Counters for Other Sequences
- ❑ Counter Design Using S-R and J-K Flip-Flops
- ❑ **Derivation of Flip-Flop Input Equations**

Derivation of Flip-Flop Input Equations

❑ Determine the flip flop input equations from the next-state equations using K-maps

➤ Always copy X's from next state maps onto input maps first

Type of FF	Input	Q = 0		Q = 1		Rules for forming input map from next state map	
		Q ⁺ = 0	Q ⁺ = 1	Q ⁺ = 0	Q ⁺ = 1	Q = 0 Half of Map	Q = 1 Half of Map
D	D	0	1	0	1	No change	No change
T	T	0	1	1	0	No change	Complement
S-R	S	0	1	0	X	No change	Replace 1's with X's
	R	X	0	1	0	Replace 0's with X's Replace 1's with 0's	Complement
J-K	J	0	1	X	X	No change	Fill in with X's
	K	X	X	1	0	Fill in with X's	Complement

Important Tables

Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

D Flip-Flop

Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

T Flip-Flop

Q	Q ⁺	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

S-R Flip-Flop

Q	Q ⁺	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

J-K Flip-Flop

3-Variable Example (1/3)

AB \ Q		0	1
00	0	1	
01	1	0	
11	0	0	
10	1	X	

Q^+



AB \ Q		0	1
00	0	1	
01	1	0	
11	0	0	
10	1	X	

$$D = Q'A'B + QB' + AB'$$

AB \ Q		0	1
00	0	0	
01	1	1	
11	0	1	
10	1	X	

$$T = A'B + AB' + QB$$

3-Variable Example (2/3)

AB \ Q		0	1
00		0	1
01		1	0
11		0	0
10		1	X

Q^+



AB \ Q		0	1
00		0	X
01		1	0
11		0	0
10		1	X

$S = AB' + Q'A'B$

AB \ Q		0	1
00		X	0
01		0	1
11		X	1
10		0	X

$R = QB$

3-Variable Example (3/3)

AB \ Q	0	1
00	0	1
01	1	0
11	0	0
10	1	X

Q^+



AB \ Q	0	1
00	0	X
01	1	X
11	0	X
10	1	X

$J = A'B + AB'$

AB \ Q	0	1
00	X	0
01	X	1
11	X	1
10	X	X

$K = B$

4-Variable Example (1/3)

BC \ Q ₁ A		00	01	11	10
BC	00	0	1	0	1
	01	X	1	1	0
	11	1	X	X	1
	10	0	0	0	X

Q₁⁺



BC \ Q ₁ A		00	01	11	10
BC	00	0	1	1	0
	01	X	1	0	1
	11	1	X	X	0
	10	0	0	1	X

T₁

4-Variable Ex. (2/3)

AB		00	01	11	10
CQ ₂	00	1	X	1	0
	01	0	0	X	1
	11	1	0	X	1
	10	X	0	0	1

Q_2^+



AB		00	01	11	10
CQ ₂	00	1	X	1	0
	01	0	0	X	X
	11	X	0	X	X
	10	X	0	0	1

S_2

AB		00	01	11	10
CQ ₂	00	0	X	0	X
	01	1	1	X	0
	11	0	1	X	0
	10	X	X	X	0

R_2

4-Variable Ex. (3/3)

AB		00	01	11	10
Q ₃ C	00	0	0	1	X
	01	0	1	X	1
	11	X	X	0	0
	10	1	1	1	0

Q₃⁺

AB		00	01	11	10
Q ₃ C	00	0	0	1	X
	01	0	1	X	1
	11	X	X	X	X
	10	X	X	X	X

J₃



AB		00	01	11	10
Q ₃ C	00	X	X	X	X
	01	X	X	X	X
	11	X	X	1	1
	10	0	0	0	1

K₃

Q&A