SCHOOL OF ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING

SEEE 1223
DIGITAL ELECTRONICS

CHAPTER 8: COUNTERS AND SHIFT REGISTERS

DR. MOHD SAIFUL AZIMI BIN MAHMUD

P19a-04-03-30

School of Electrical Engineering

Faculty of Engineering

Universiti Teknologi Malaysia

019-7112948

azimi@utm.my

innovative • entrepreneurial • global

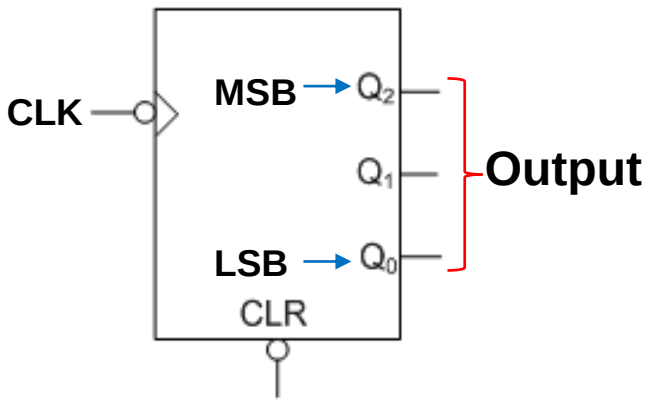
COUNTERS

COUNTERS

INTRODUCTION

- **Counter** is a group of flip flops that are connected together to perform **counting operations**.
- The number of flip-flops used and the way which they are connected will determine:
  i. The number of states
  ii. The specific sequence of states the counter goes through during each complete cycle.

Example: 3-bit binary counter



3-bit Binary Counter
Logic Symbol

IF $CLR = 0$

$Q_2Q_1Q_0 = 000$

ELSE IF $CLR = 1$

IF $CLK = \downarrow$

$Q_2Q_1Q_0 = Q_2Q_1Q_0 + 001$

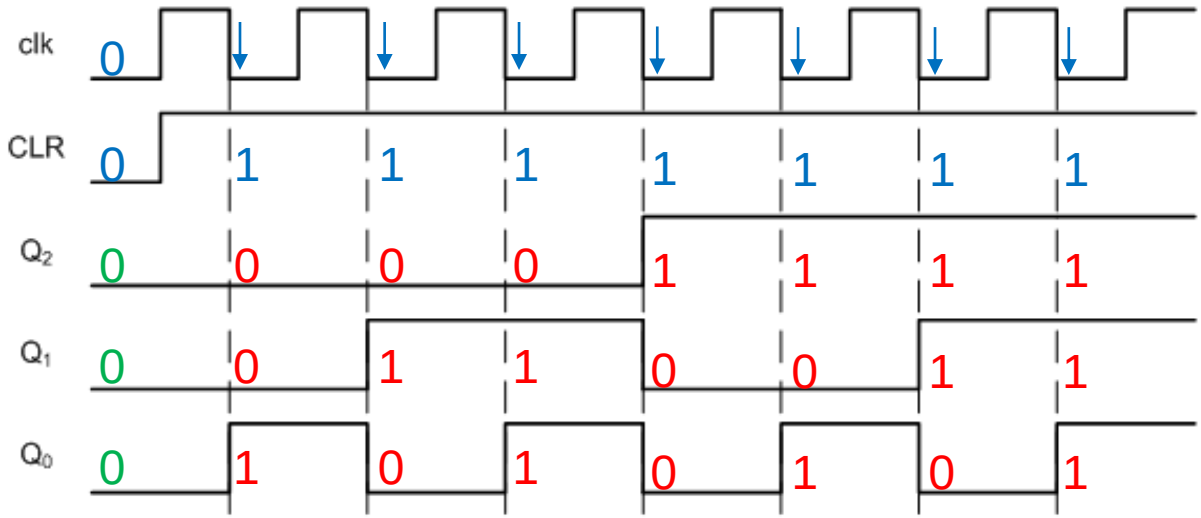
ELSE

$Q_2Q_1Q_0 = Q_2Q_1Q_0$

COUNTERS

INTRODUCTION

- Thus the output waveforms for Q_2, Q_1 and Q_0 are:

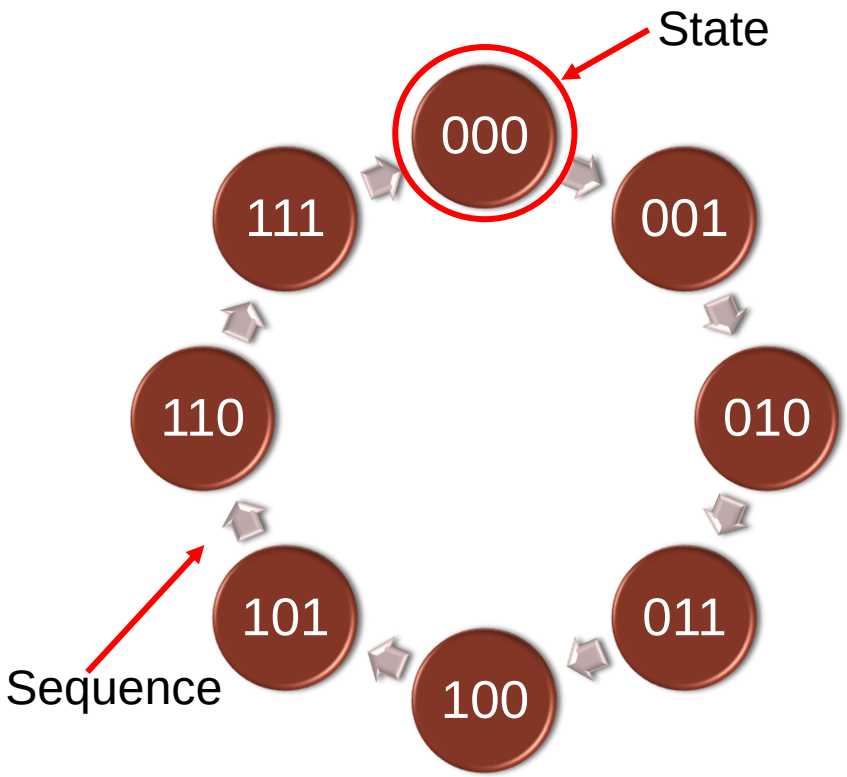


Timing diagram

COUNTERS

INTRODUCTION

- Thus the output waveforms for Q_2, Q_1 and Q_0 are:



Sequence diagram

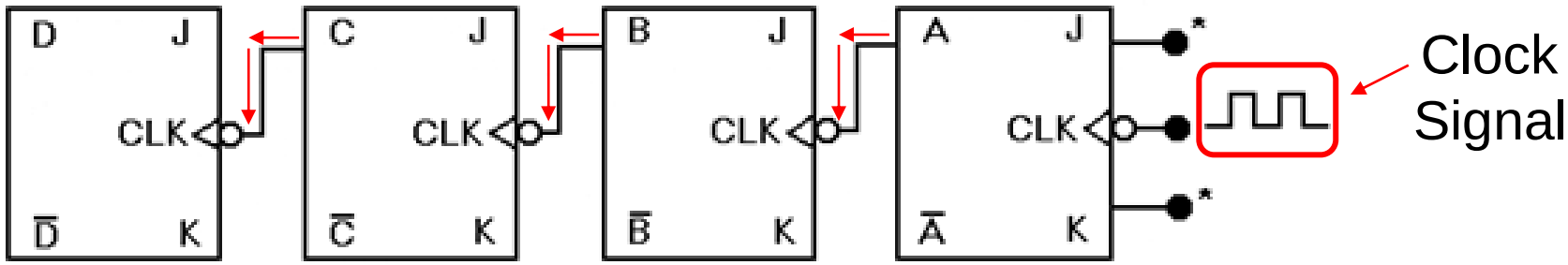
COUNTERS

INTRODUCTION

- Counters are classified into two broad categories according to the way they are clocked. Those categories are:

Asynchronous Counter

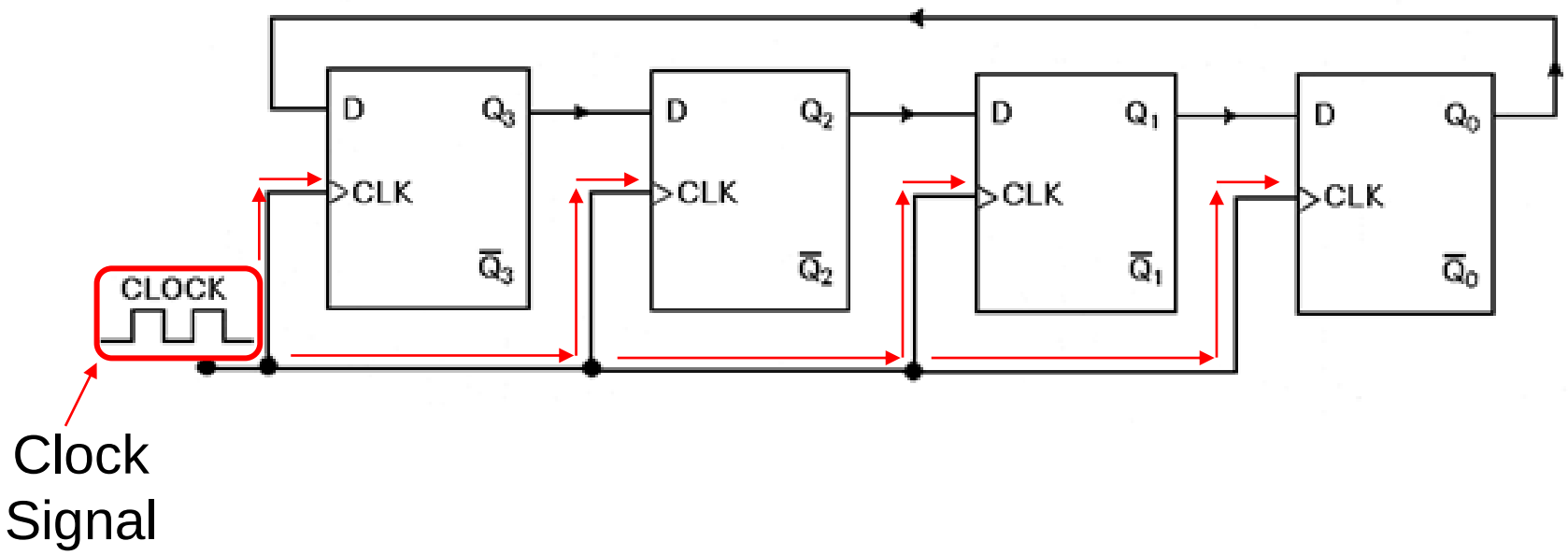
Known as ripple counter, the first flip-flop is clocked by the external clock and each of successive flip-flop is clocked by the output of the preceding flip-flop.



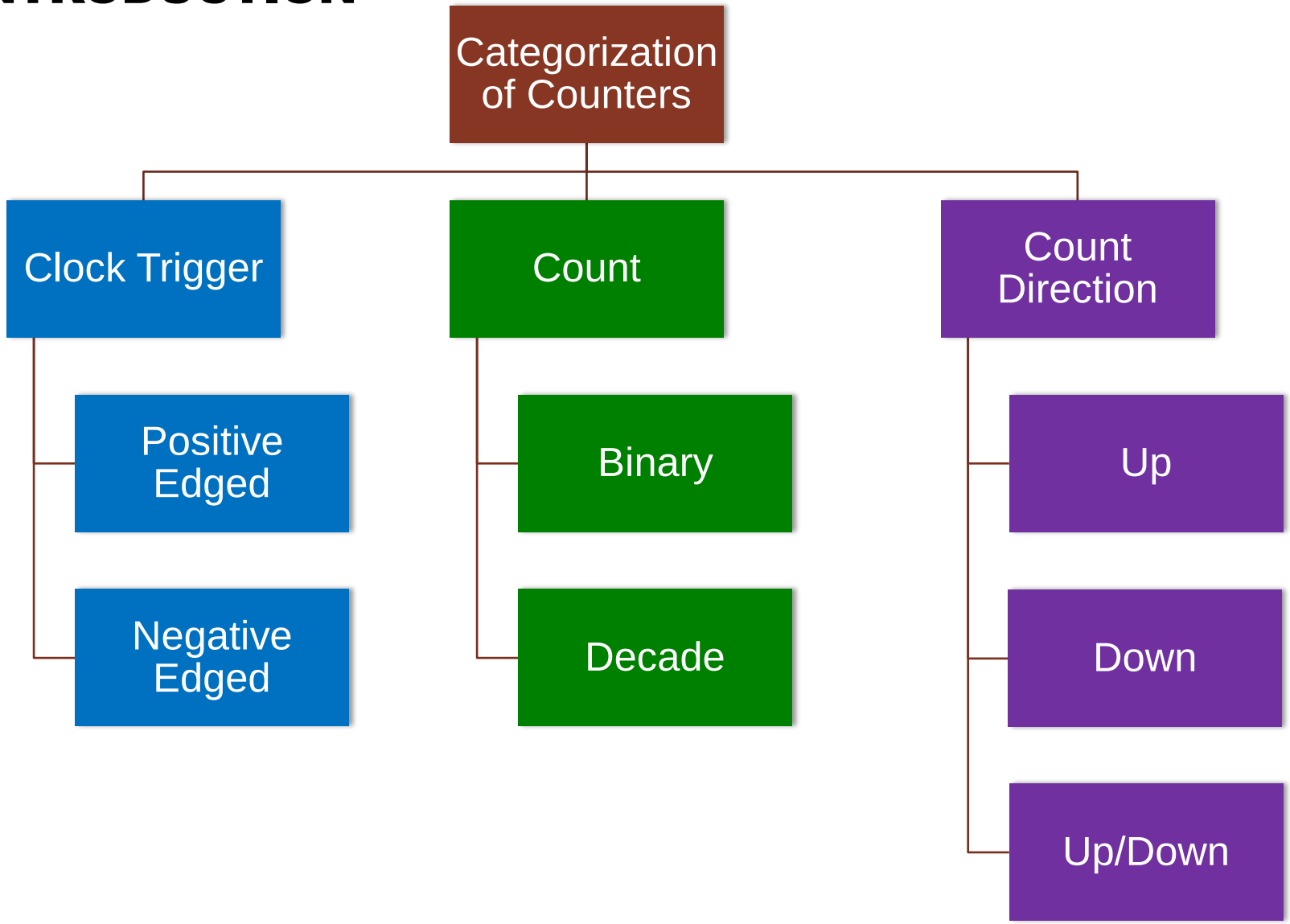
COUNTERS
INTRODUCTION

Synchronous Counter

The clock input is connected to all the flip-flops so they are clocked simultaneously.



COUNTERS
INTRODUCTION



COUNTERS
INTRODUCTION

Count the number of time that the certain event take place

To generate clock of different frequencies

Functions of Counter

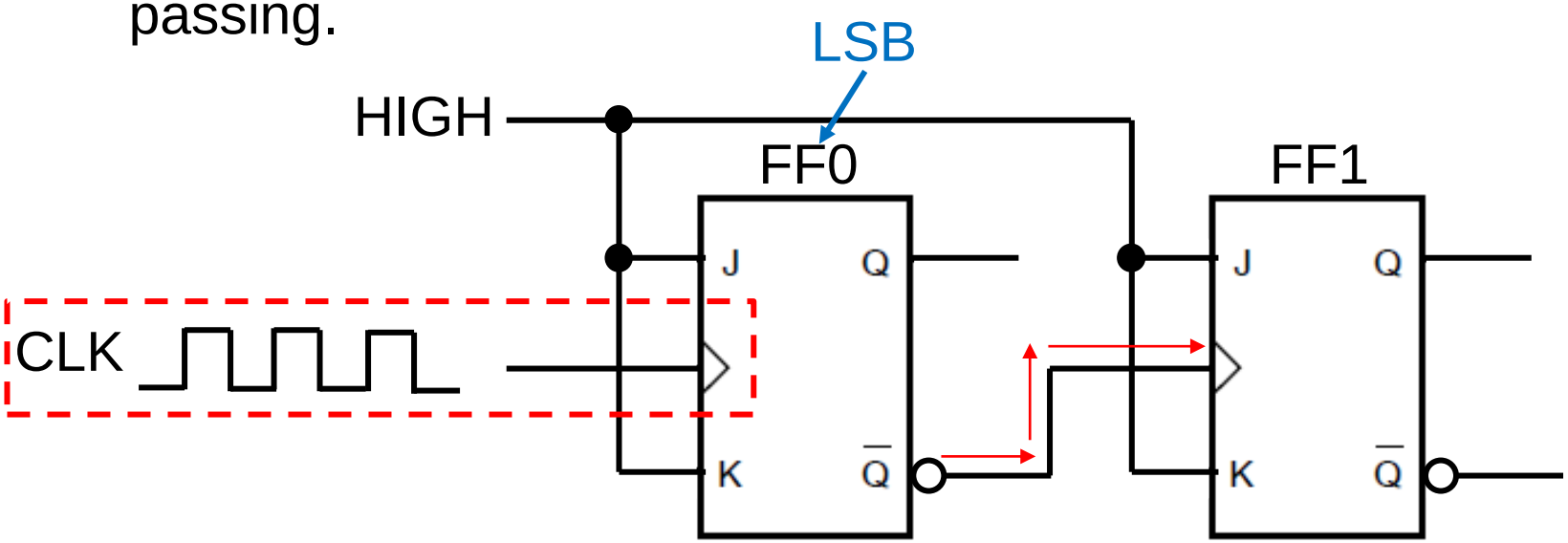
To control a fixed sequence of actions in digital system

To generate timing signals

COUNTERS

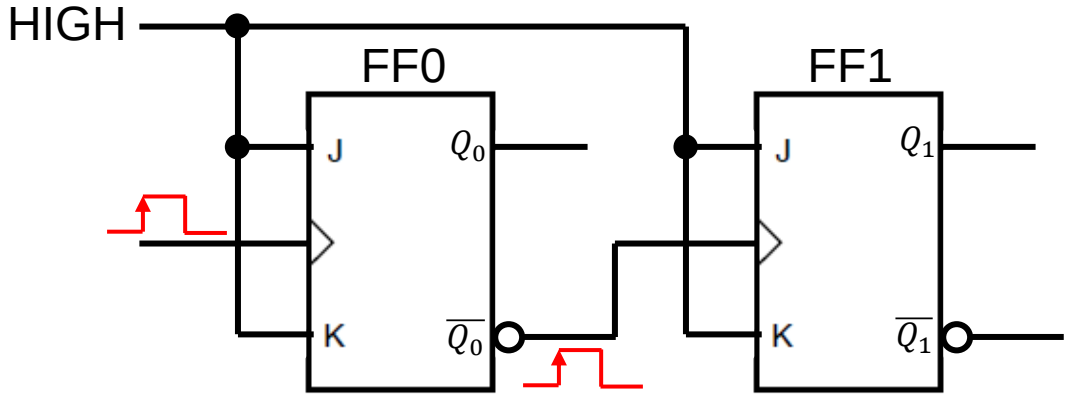
ASYNCHRONOUS COUNTER: INTRODUCTION

- **Asynchronous counter** is one in which the Flip-Flops (FF) within the counter **do not change states** at **exactly the same time** because they do not have a common clock pulse.
- The clock input is always connected only at the **LSB** of FF.
- The clock for the next FF comes from output of the previous FF.
- It is also known as a **ripple counter** – message (signal) passing.



COUNTERS

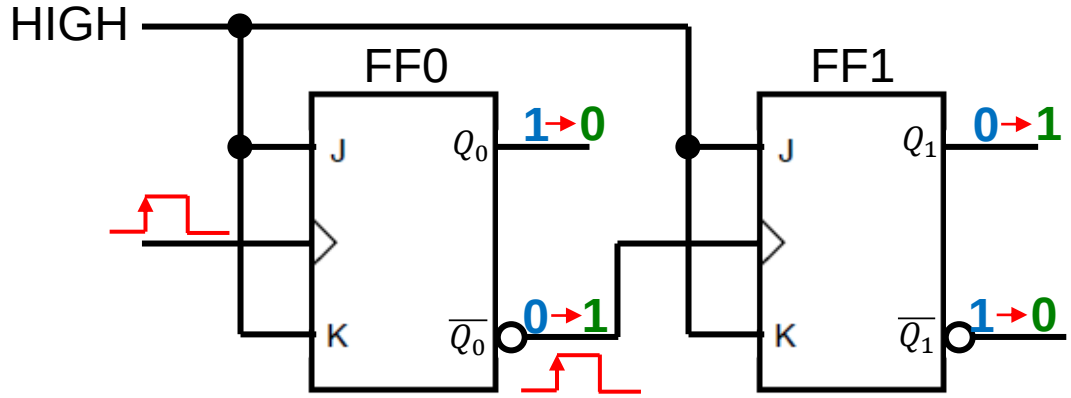
ASYNCHRONOUS COUNTER: INTRODUCTION



- **FF0** changes state at the **positive edge** of each clock pulse, but **FF1** changes only when **triggered by the positive edge of the $\overline{Q}$ output of FF0**.
- Because of the inherent propagation delay through a flip-flop, the transition of the input clock pulse and a transition of the Q output of FF0 can **never occur at exactly on the same time**.
- Therefore, the flip-flops **cannot be triggered simultaneously**, producing an **asynchronous** operation.

COUNTERS

2-BIT ASYNCHRONOUS BINARY COUNTER



COUNTER: 01

↓

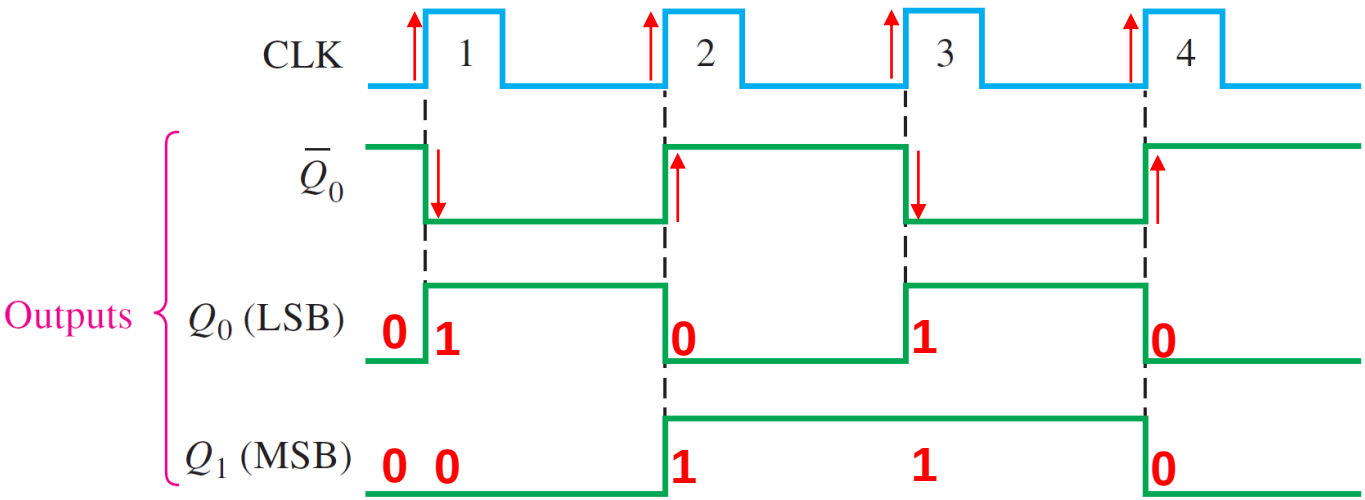
COUNTER: 10

- Input JK for each FF is tied up together. Therefore $\boldsymbol{J = K = 1\ (HIGH)}$. In other word, the JK FF function in a **toggle mode** i.e. every clock pulse Q will be complemented.
- Every positive edge of the clock will toggle $\boldsymbol{Q_0}$ for FF0.
- For FF1, the clock depend on $\overline{\boldsymbol{Q_0}}$ and the output $\boldsymbol{Q_1}$ will only toggle on the positive edge of $\overline{\boldsymbol{Q_0}}$.
- The output of the circuit counter is read $\boldsymbol{Q_1 Q_0}$.

COUNTERS

2-BIT ASYNCHRONOUS BINARY COUNTER

- The timing diagram for 2-bit asynchronous binary counter:



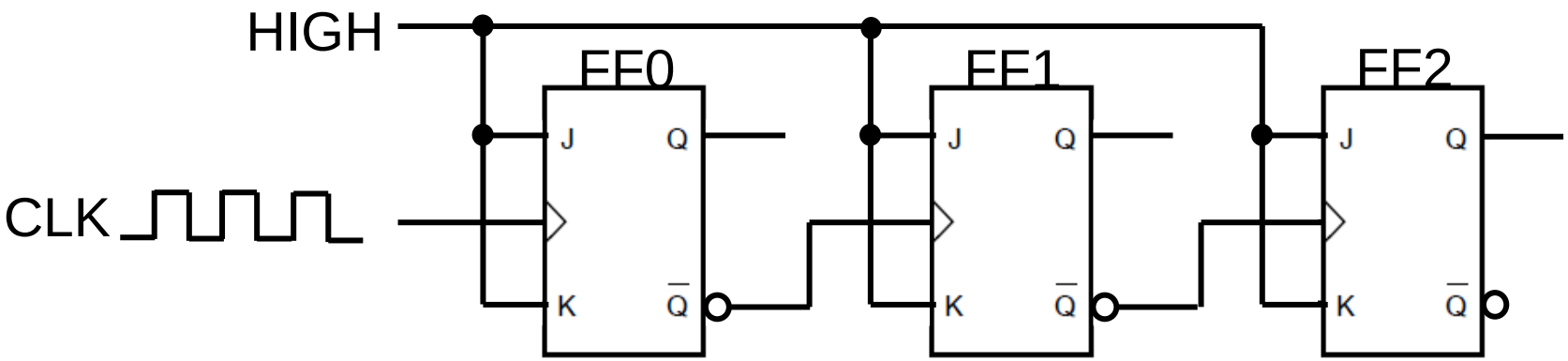
Binary state sequence for the counter

CLOCK PULSE	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

- The 2-bit ripple counter has **four different states**, each one corresponding to a count value.
- A counter with n flip-flops can have **2^n states**.

COUNTERS

3-BIT ASYNCHRONOUS BINARY COUNTER

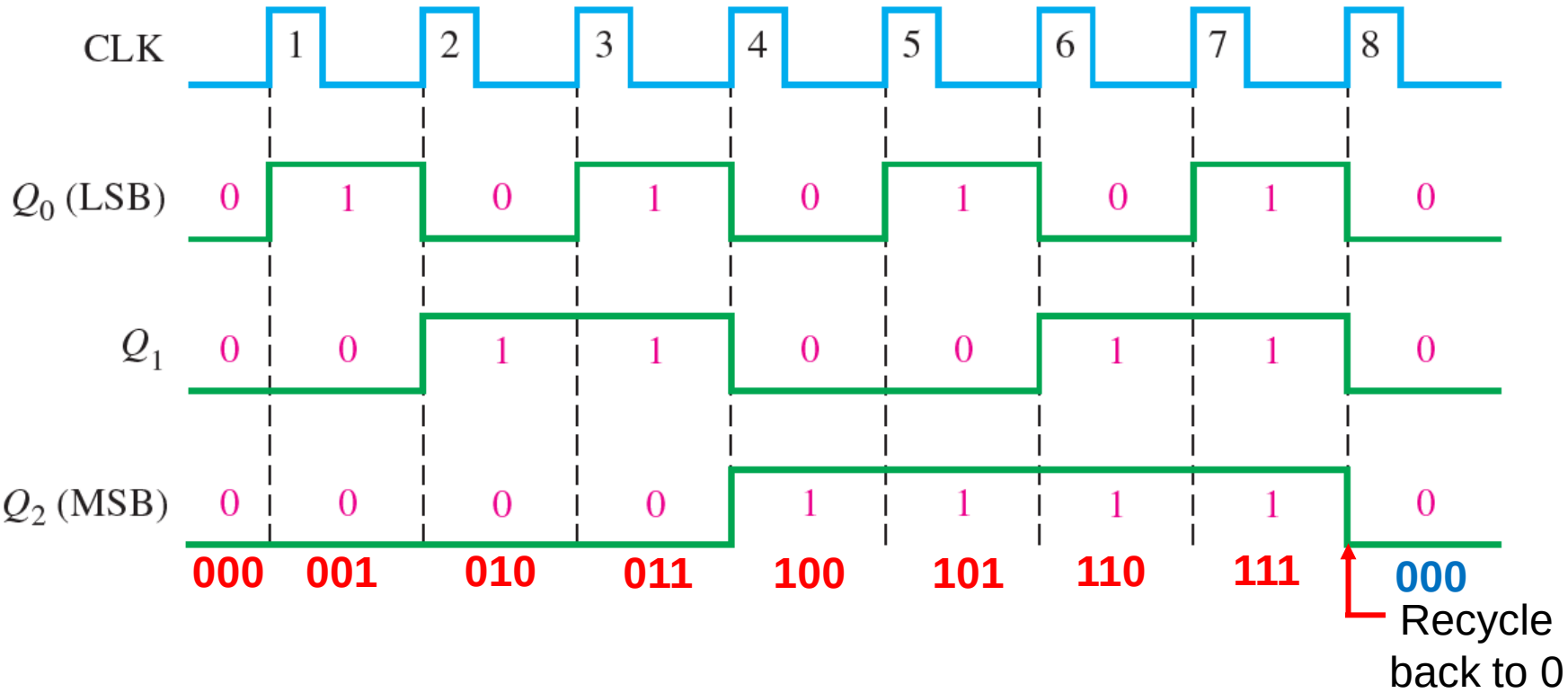


- The circuit connection is the same as 2-bit asynchronous counter, for a 3 bit counter just **add another flip flop**.

COUNTERS

3-BIT ASYNCHRONOUS BINARY COUNTER

- The timing diagram for 3-bit asynchronous binary counter:



- The counting changes at every positive edge of the clock.
- The counting sequence is 0, 1, 2, 3, 4, 5, 6, 7 and recycle back to 0.

COUNTERS

3-BIT ASYNCHRONOUS BINARY COUNTER

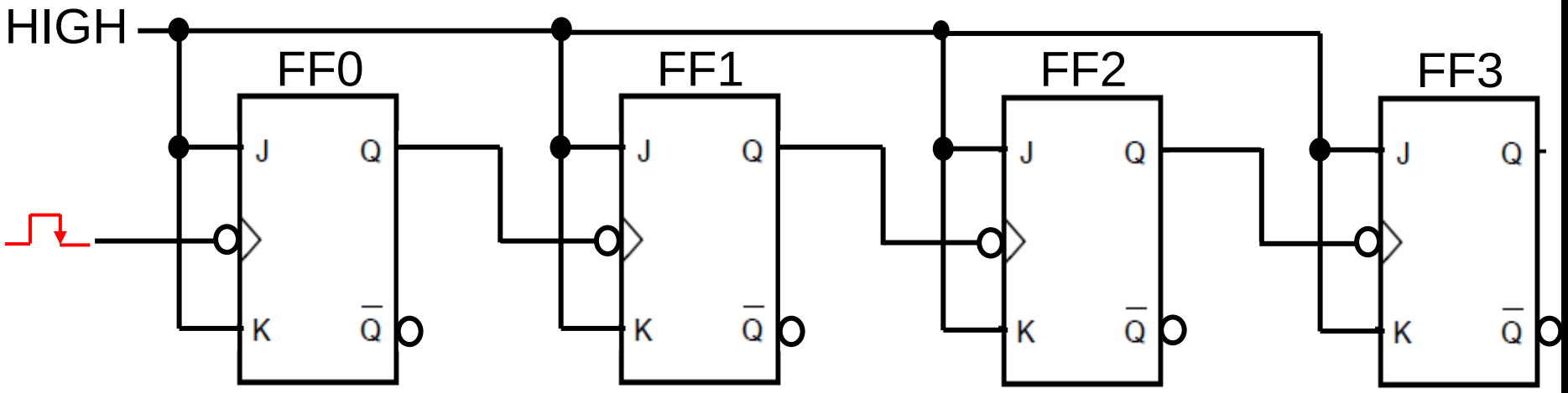
Binary state sequence for the counter

8 States (2^3)

CLOCK PULSE	Q_2	Q_1	Q_0
Initially	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
8 (recycles)	0	0	0

COUNTERS

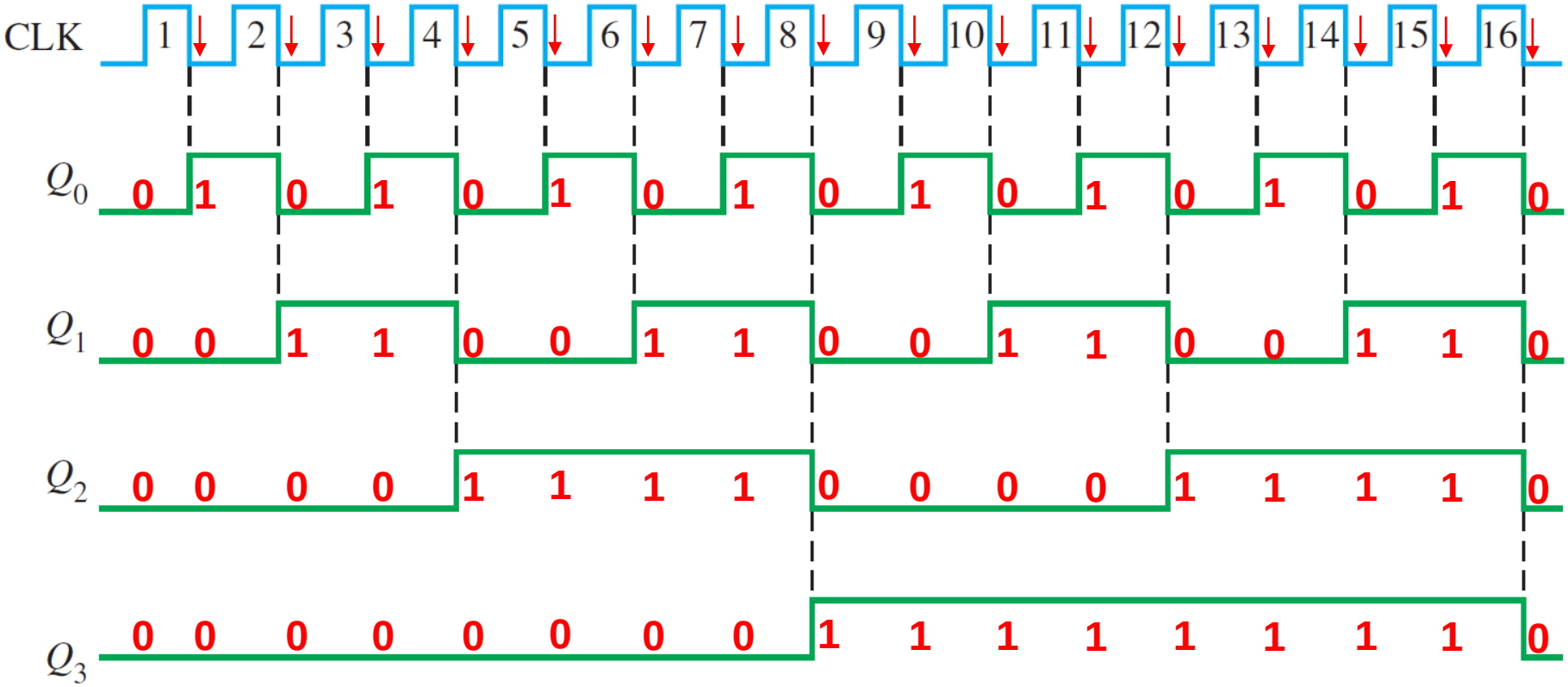
4-BIT ASYNCHRONOUS BINARY COUNTER



- The circuit connection is the same as 3-bit asynchronous counter, for a 4 bit counter just **add another flip flop**.
- In the example, the counter using a negative edge JK flip-flop. But it still **count up** because clock is connected to Q.

COUNTERS

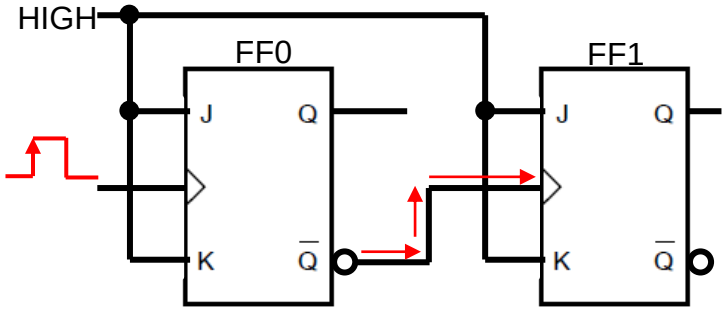
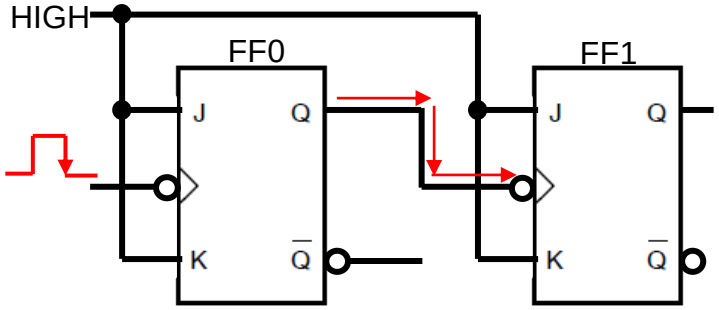
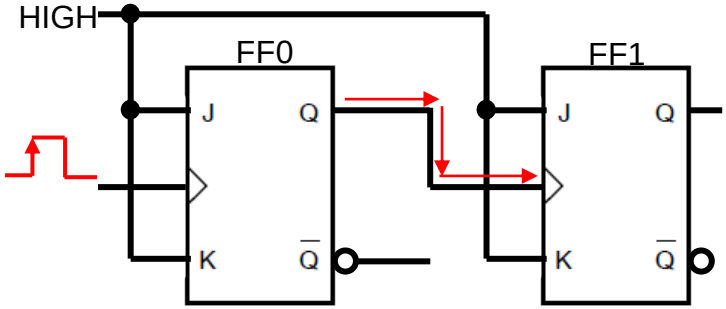
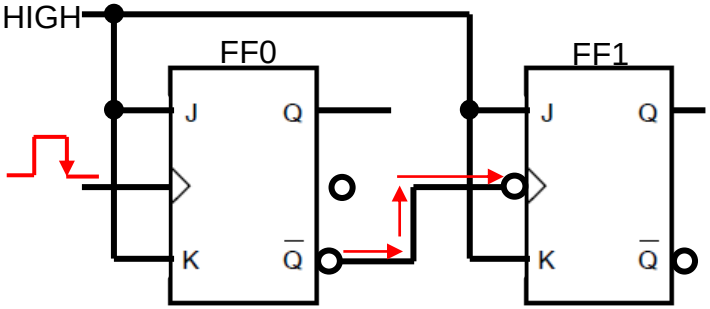
4-BIT ASYNCHRONOUS BINARY COUNTER



- In the example, the counter using a negative edge JK flip-flop. But it still **count up** because clock is connected to Q.

COUNTERS

ASYNCHRONOUS BINARY COUNTER SUMMARY

	Positive Edge	Negative Edge
Count Up		
Count Down		

COUNTERS

ASYNCHRONOUS BINARY COUNTER SUMMARY

1. Basically, the circuit connection for Asynchronous Counter is almost the same where the **external clock** connected to the **LSB Flip-Flop (FF)**.
2. Clock for each FF come from the previous FF, **except for the first FF** (i.e. FF0).
3. Every FF operates in **Toggle mode** (i.e. next output is a complement of the previous output).
   - To operate in Toggle mode, **for SR:** $S = Q$ and $R = \bar{Q}$, **for JK:** $J = K = 1$, **for D:** $D = \bar{Q}$, **for T:** $T = 1$.
4. The design connection is the same, if we want **more bits** just **add more flip-flops**.
5. The **difference in connection** will determine whether we want to count **UP** or **DOWN** and by using which type of flip-flop: positive or negative edge.

COUNTERS

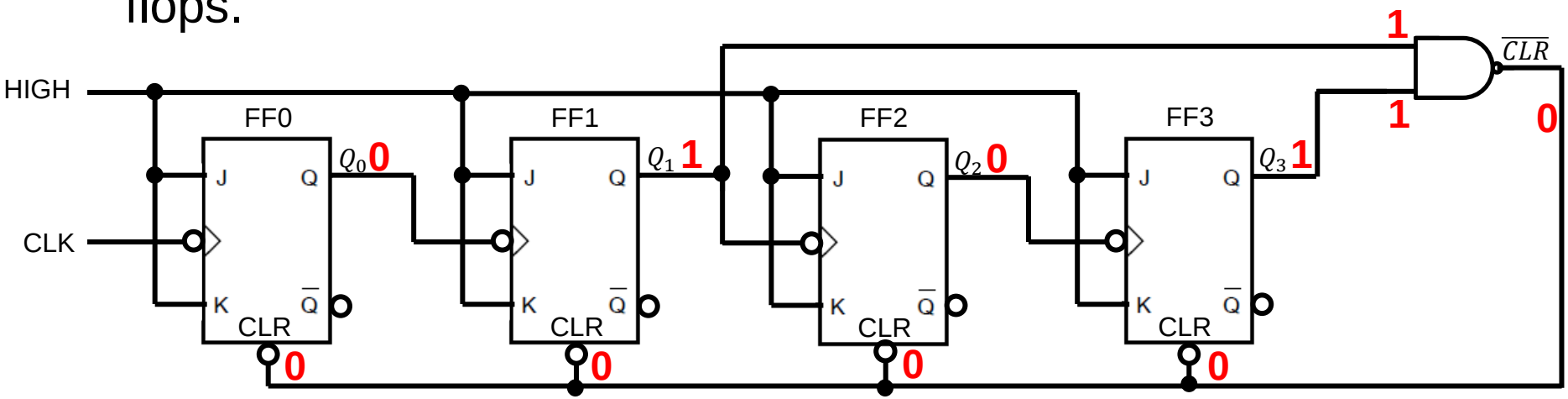
ASYNCHRONOUS DECADE COUNTER

- The **modulus** of a counter is the number of unique states through which counter will sequence.
- Previously, the possible number of state (maximum modulus) of counter is $\mathbf{2^n}$.
- Counter can be designed to have number of state in their sequence that is **less than $\mathbf{2^n}$**.
- This type of sequence is called a **truncated sequence**.
- One common modulus for counters with truncated sequence is ten that is called as **decade counter**.
- This counter has sequence count of zero (0000) through nine (1001) and is referred as a **BCD decade counter** because the ten state sequence is in BCD code (4 bits).

COUNTERS

ASYNCHRONOUS DECADE COUNTER

- To obtain a truncated sequence, it is necessary to **force the counter** to cycle before going through all of its normal states.
- For example, the BCD decade counter must **recycle back** to the state **0000 after state 1001**. Note that a decade counter will require 4 flip-flops (three flip-flops are insufficient as $2^3 = 8$).
- To achieve recycling after the count of nine (1001), the counter has to **decode ten (1010)** with a **NAND gate** and connect the output of the NAND gate to the **clear (CLR)** inputs of all the flip flops.



COUNTERS

ASYNCHRONOUS DECADE COUNTER

- Notice that only $\boldsymbol{Q_1}$ and $\boldsymbol{Q_3}$ are connected to the NAND gate inputs.
- This arrangement is an example of **partial decoding**, in which two unique states ($Q_1 = 1$ and $Q_3 = 1$) are enough to decode the count of 10.
- This is because **none of other states** (zero through nine) have $\boldsymbol{Q_1}$ and $\boldsymbol{Q_3}$ **HIGH** at the same time.
- Consequently, when the counter goes into **count ten (1010)**, the decoding gate output goes **LOW** and asynchronously **RESETS** all of the flip-flops.

COUNTERS

ASYNCHRONOUS DECADE COUNTER

Binary state sequence for Asynchronous Decade Counter

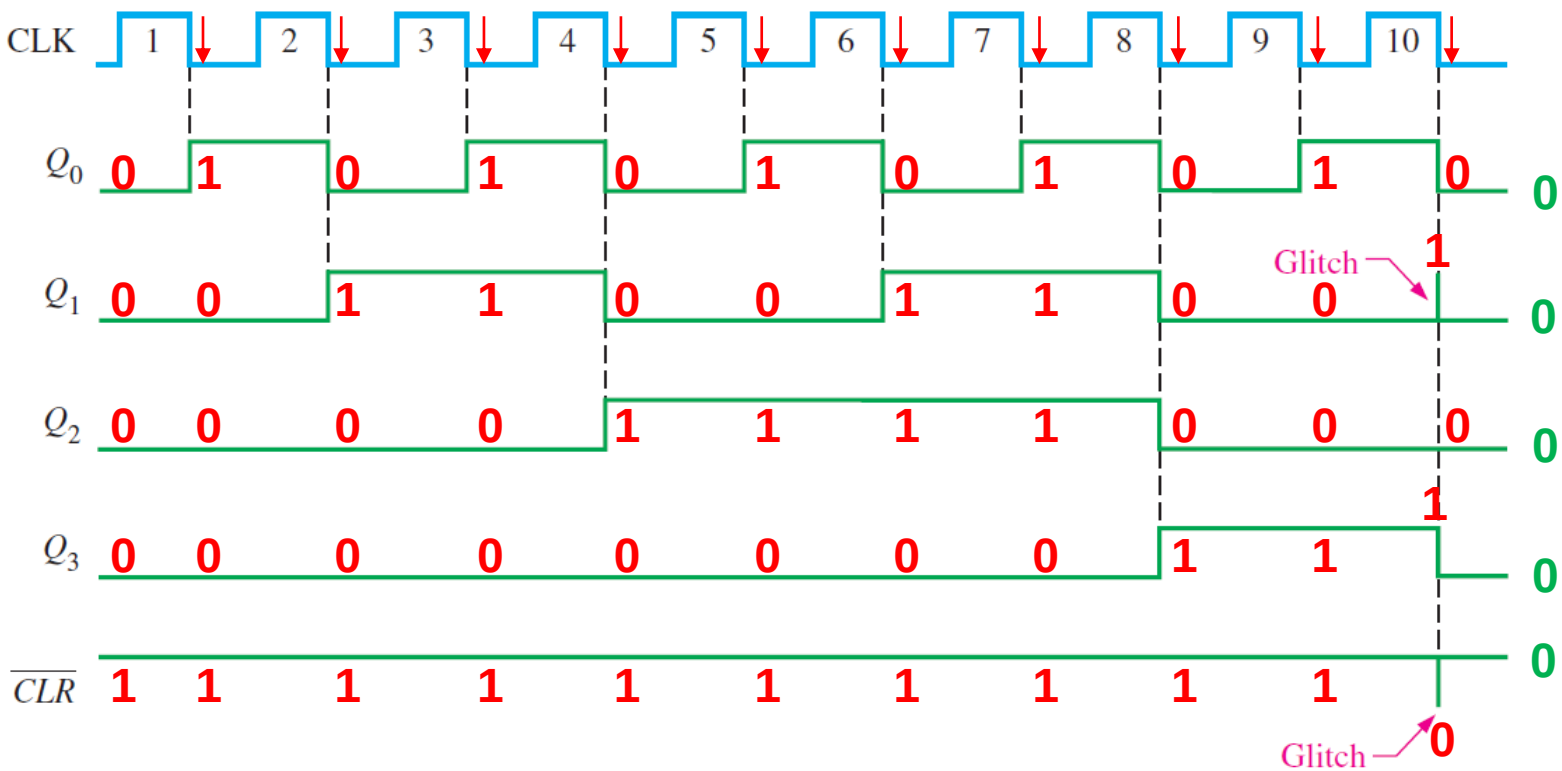
Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1

Q_3	Q_2	Q_1	Q_0
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**

Not present in Asynchronous Decade Counter

COUNTERS

ASYNCHRONOUS DECADE COUNTER



- Notice that, there is a **glitch** in Q1 and $\overline{CLR}$. The reason of this glitch is that Q1 must first go HIGH before the count 10 can be decoded. Several nanoseconds after the decoding gate goes low.

COUNTERS

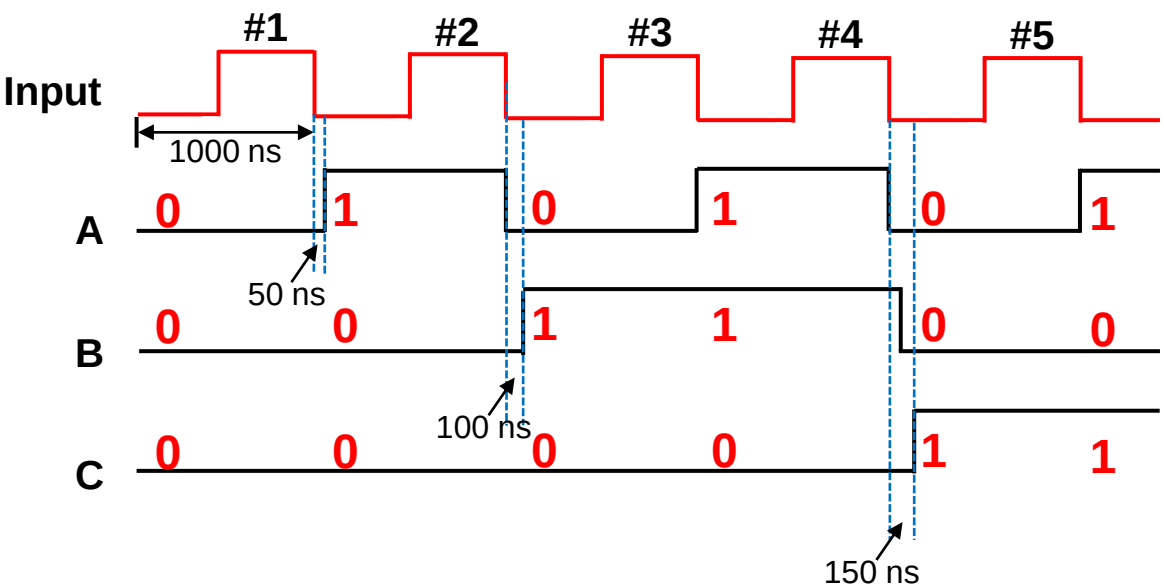
ASYNC COUNTER: ADVANTAGE & DISADVANTAGE

Advantage

- The design step is simple (easy)

Disadvantage

- Propagated delay accumulate as in ripple binary adder which may cause a missing counting state especially at high frequency operations.

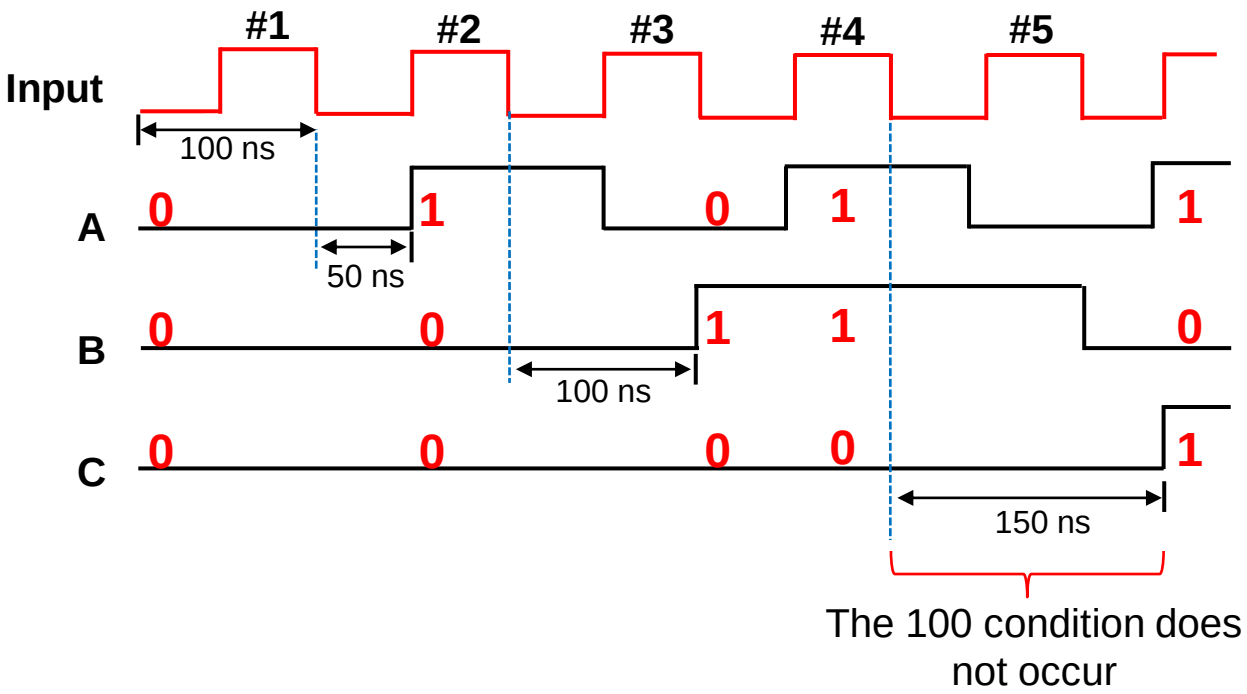


Case 1: Operating at low frequency of 1 MHz

For each stage there is a **50 ns** delay and it accumulates up to **150 ns** at the **3rd stage**. The accumulated delay is still **lower** than the **period of the signal** which is 1000 ns, therefore there is **no effect** to the counting sequence.

COUNTERS

ASYNC COUNTER: ADVANTAGE & DISADVANTAGE



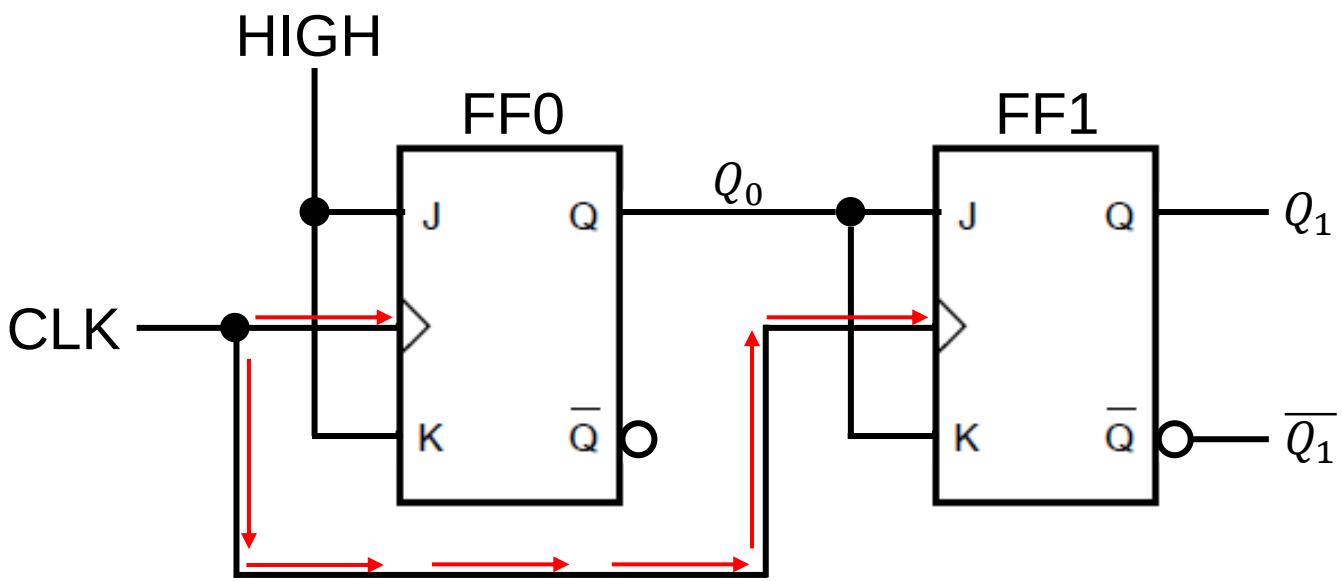
Case 2: Operating at higher frequency of 10 MHz

The clock period is only **100 ns**. For a 3 stage flip-flop, the **accumulated delay** is the same as before which is **150 ns**. But, the current accumulated delay is **more than the clock period**, therefore there is a **missing count**, 100 does not exist!

COUNTERS

SYNCHRONOUS COUNTER: INTRODUCTION

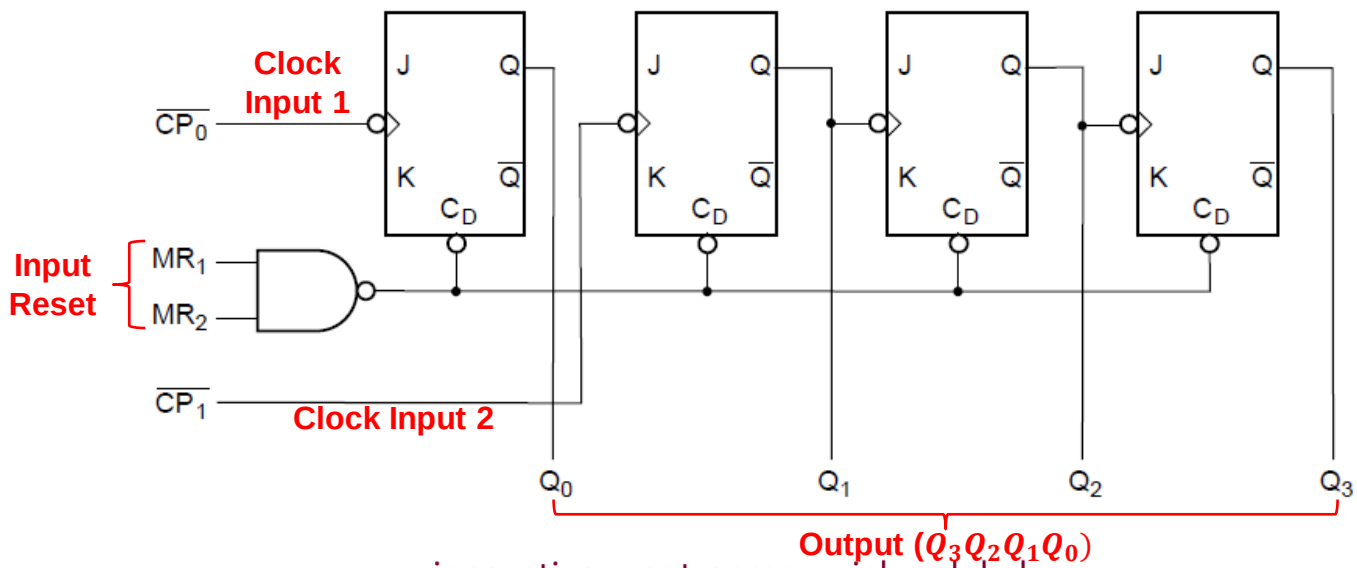
- A **synchronous counter** is one in which all flip flops in the counter are clocked at the same time by a common clock pulse.



COUNTERS

COUNTER IC: ASYNCHRONOUS COUNTER IC (74x293)

- 74x293 is an asynchronous counter IC that has **four JK flip-flops** with outputs Q_3, Q_2, Q_1 and Q_0.
- Two flip-flops have their own CP (clock pulse) input. Clock input to FF_1 and FF_0 labelled as $\overline{CP_1}$ and $\overline{CP_0}$ while FF_2 and FF_3 comes from output Q_1 and Q_2 respectively.
- Each flip-flop has C_D input to RESET (that are connected to two input NAND gate from MR_1 and MR_2. (IC RESET when MR_1 and MR_2 is **HIGH**).



COUNTERS

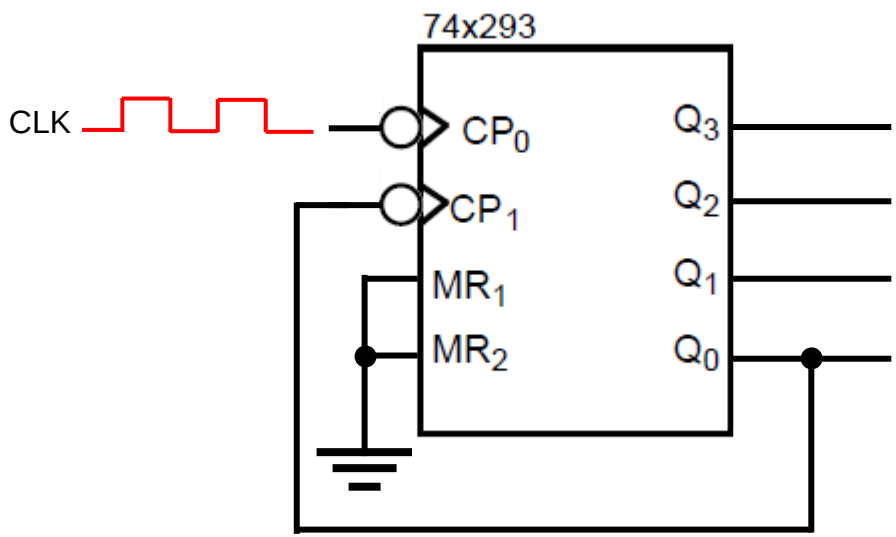
COUNTER IC: ASYNCHRONOUS COUNTER IC (74x293)

Example

Show how the 74x293 is connected to become a mod-16 counter.

Solution $No.of\ States = 2^n \quad 2^4 = 16\ states$

Since mod-16 required 4 flip-flops, all flip flops in 74x293 are used. Therefore, Q_0 is connected to CP_1 while CP_0 is connected to external clock.



COUNTERS

COUNTER IC: ASYNCHRONOUS COUNTER IC (74x293)

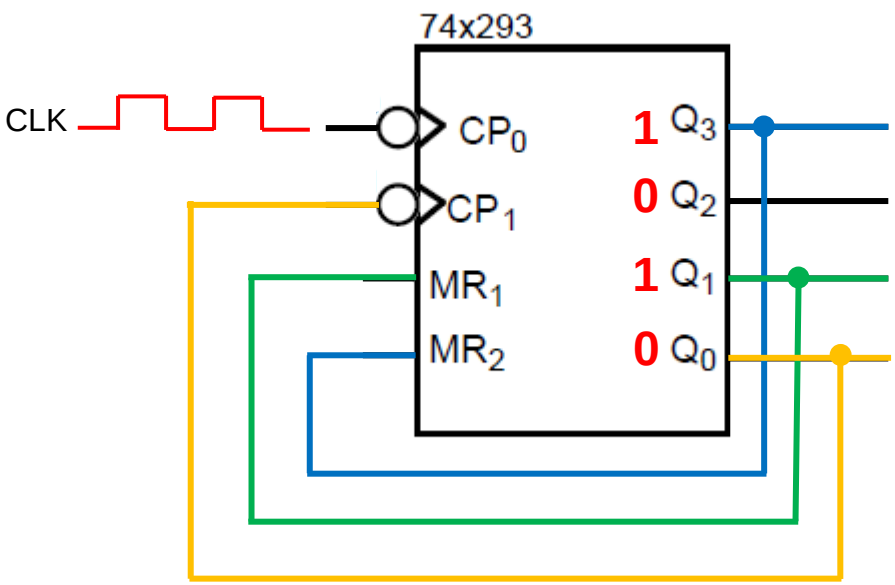
Example

Show how the 74x293 is connected to become a mod-10 counter.

Solution

$$No.\,of\ States = 2^n \qquad 2^4 = 16\ states > 10$$

Since mod-10 required 4 flip-flops, all flip flops in 74x293 are used. Therefore, Q_0 is connected to CP_1 while CP_0 is connected to external clock. Meanwhile, since the counter count from 0000 (0) to 1001 (9), so the counter must be cleared (0000) when reach to 1010. Thus Q_3 and Q_1 are connected to MR_2 and MR_1, respectively.



COUNTERS

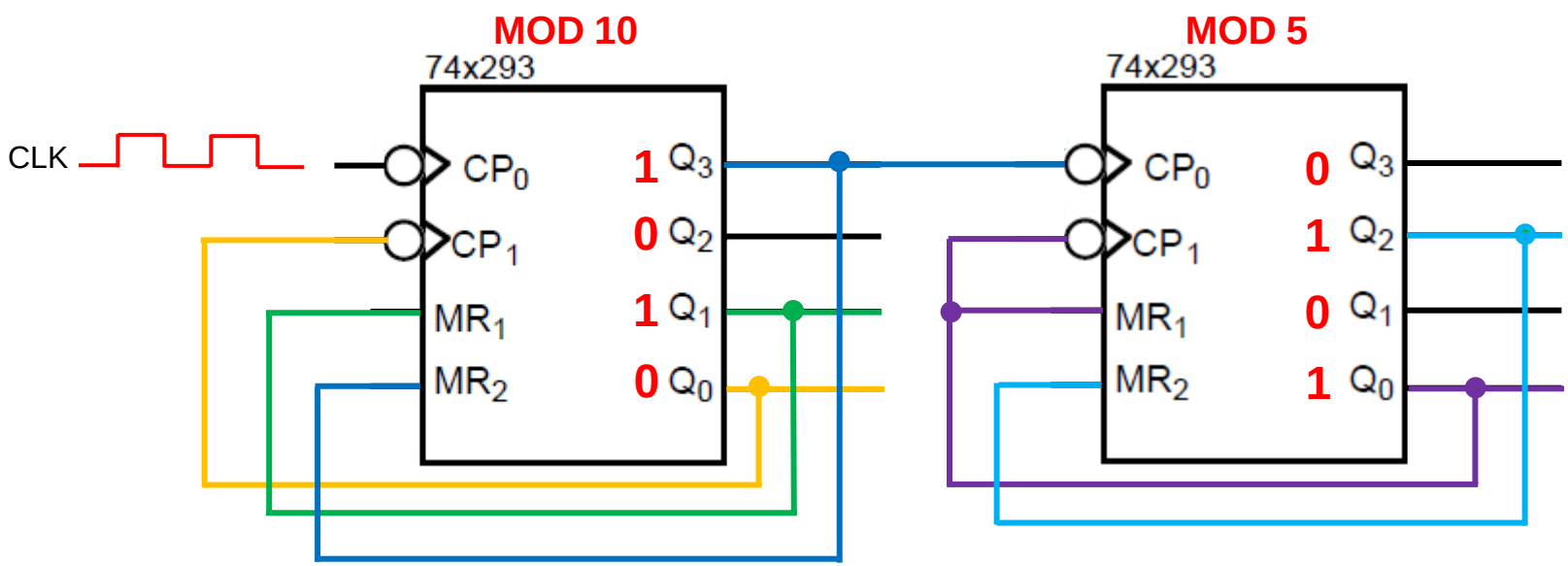
COUNTER IC: ASYNCHRONOUS COUNTER IC (74x293)

Example

Show how the 74x293 is connected to become a mod-50 counter.

Solution $No.of\ States = 2^n \quad 2^4 = 16\ states < 50$

Since mod-50 required two 74x293s that are cascaded together, the first counter as function mod-10. The output is fed up to the CP_0 at second counter which function as mod-5.



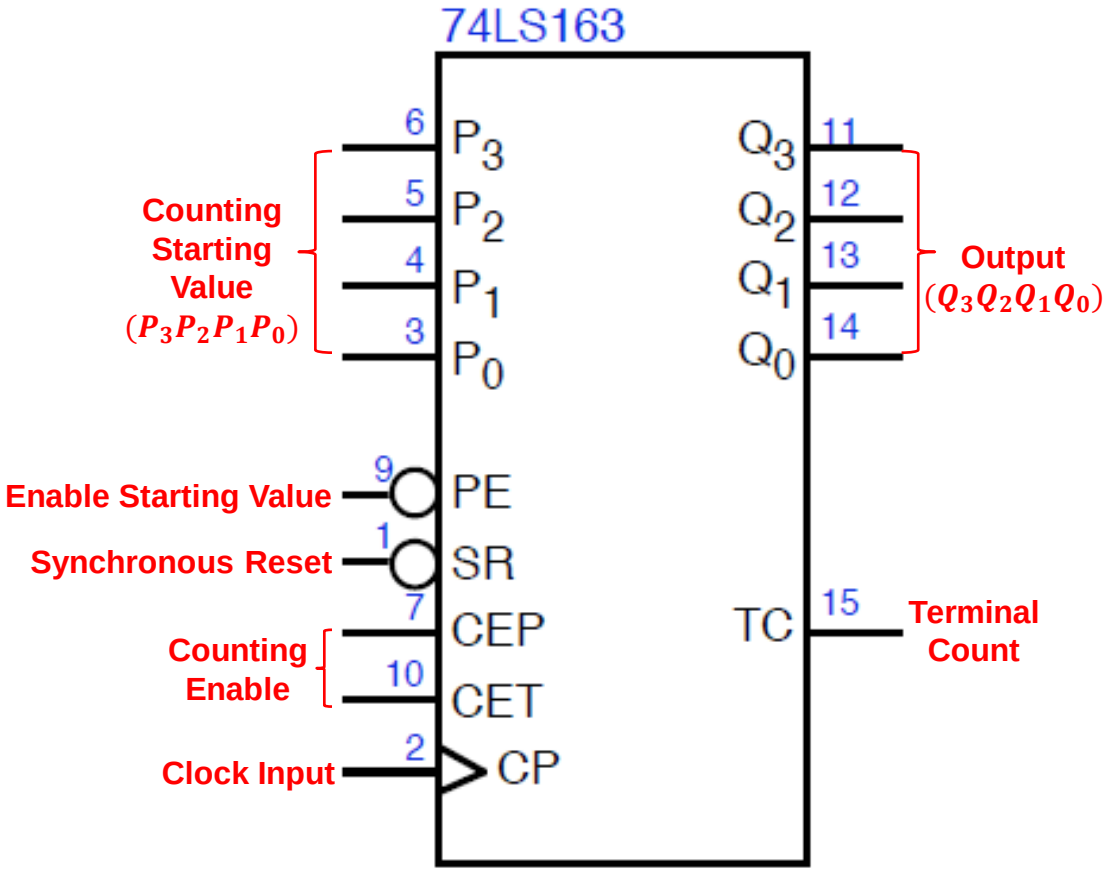
COUNTERS

COUNTER IC: SYNCHRONOUS COUNTER IC (74x163)

- 74x163 is a popular synchronous counter IC.
- It can be initialized by any starting value by giving a low pulse at the **PE input**, and any required starting value at $P_3P_2P_1P_0$.
- To enable counting, both **CEP** and **CET** must be **HIGH**.
- The input **SR** (Synchronous RESET) only valid at the positive trigger edge.
- The output **TC** (terminal count) becomes HIGH when counter reach 1111- that is useful to enable other counters in cascade configuration.

COUNTERS

COUNTER IC: SYNCHRONOUS COUNTER IC (74x163)



Logic Symbol of 74x293 Asynchronous Counter

COUNTERS

COUNTER IC: SYNCHRONOUS COUNTER IC (74x163)

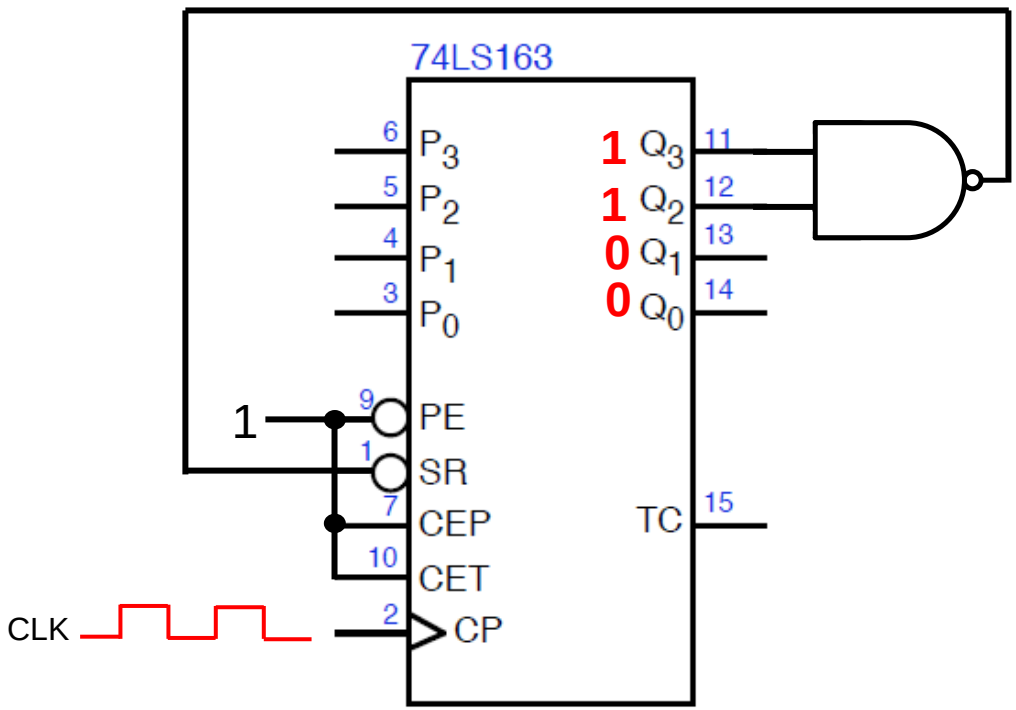
Example

Show how the 74x163 is connected to implement a mod-13 counter.

Solution

$$No.of\ States = 2^n \quad 2^4 = 16\ states > 13$$

The mod-13 counts from 0 to 12. Since counter clear synchronously (after a clock pulse arrives), in order to detect number 12, only need to detect high value at Q3 and Q2 to NAND gate which will activated when count reaches 1100.



COUNTERS

COUNTER IC: SYNCHRONOUS COUNTER IC (74x163)

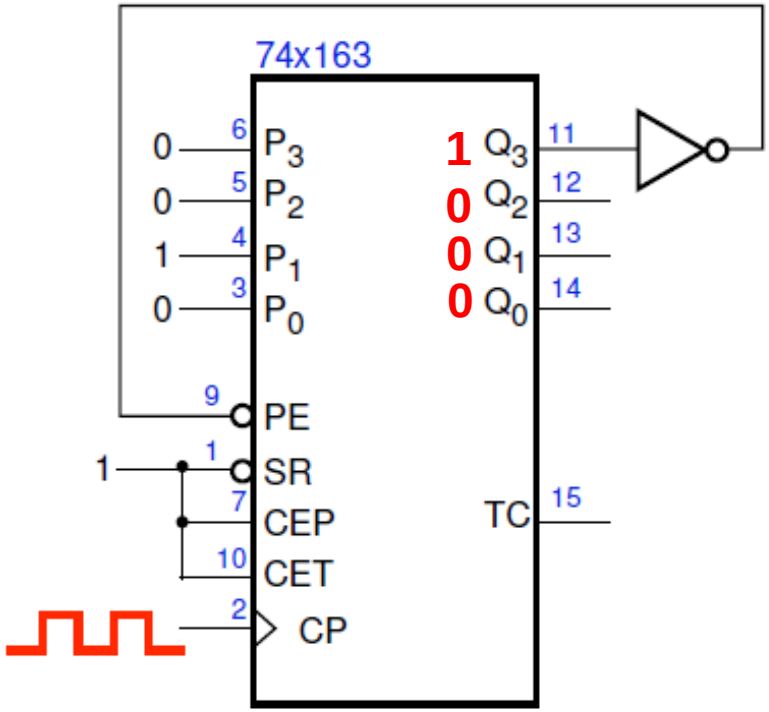
Example

Show how the 74x163 is connected to make it count from 2 to 8.

Solution

$$No.of\ States = 2^n \qquad 2^4 = 16\ states > 7$$

To make counting from 2 (0010) to 8 (1000), the counter must stop counting at 8 and starting counting from 2. To do that, the detection of logic 1 at Q3 is enough (1000) to make the counter stop at 8. connect the Q3 at PE with 0010 (2) as parallel input.



COUNTERS

COUNTER IC: SYNCHRONOUS COUNTER IC (74x163)

Example

Show how two 74x163s are connected to become a mod-256 counter

Solution $No.of\ States = 2^n \quad 2^4 = 16\ states < 256$

Since mod-256 required two 74x163s that are cascaded together, the left device is LSB counter while the right is MSB counter. Both counter must be enabled thus PE and SR are HIGH. The left counter counts continuously, that's why CEP and CET fixed at 1. The right counter only start counting when the left counter overflow, CEP and CET connected to TC.

COUNTERS

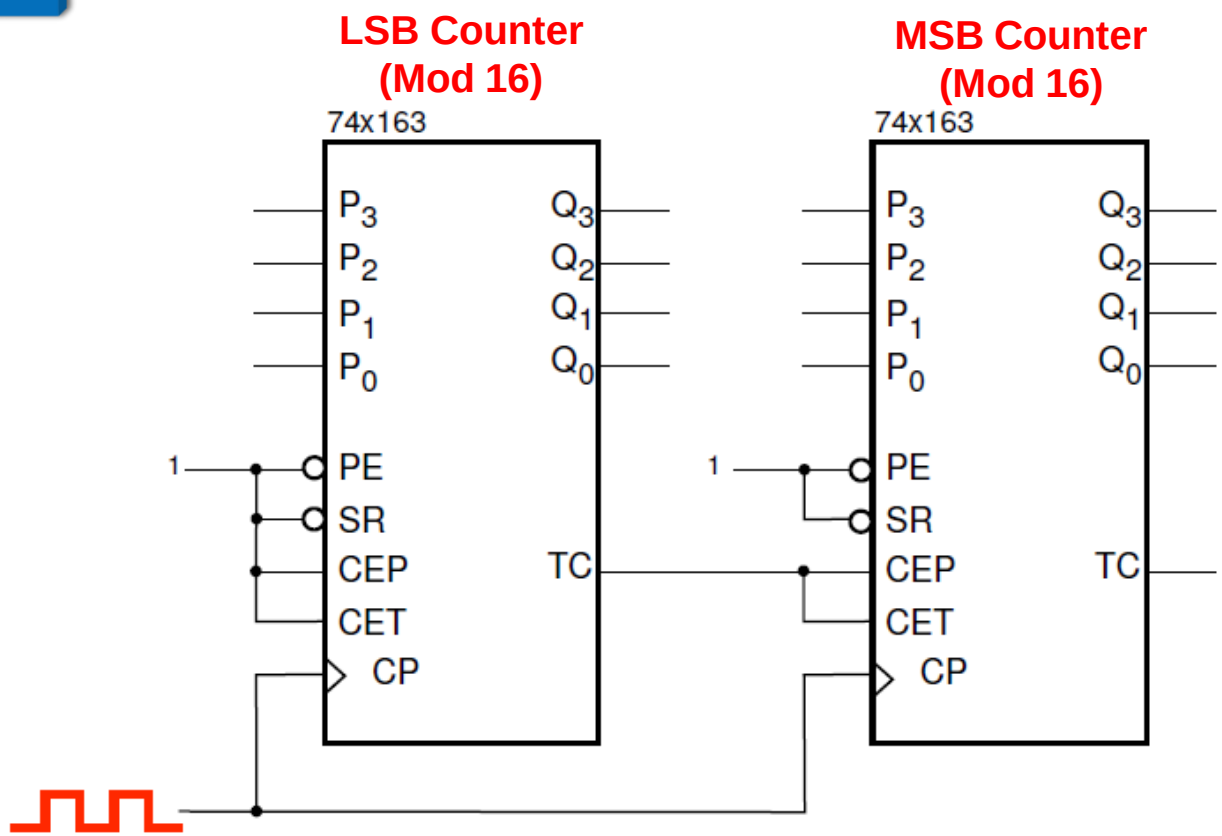
COUNTER IC: SYNCHRONOUS COUNTER IC (74x163)

Example

Show how two 74x163s are connected to become a mod-256 counter

Solution

$$No.of\ States = 2^n \quad 2^4 = 16\ states < 256$$

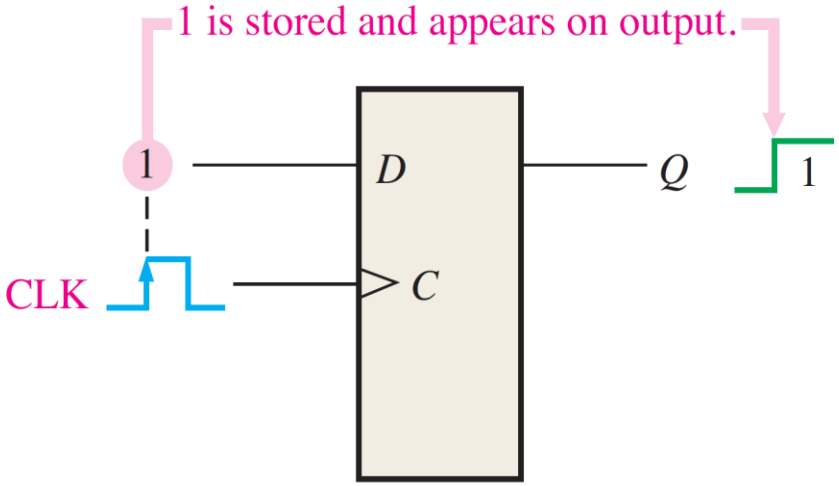


SHIFT REGISTERS

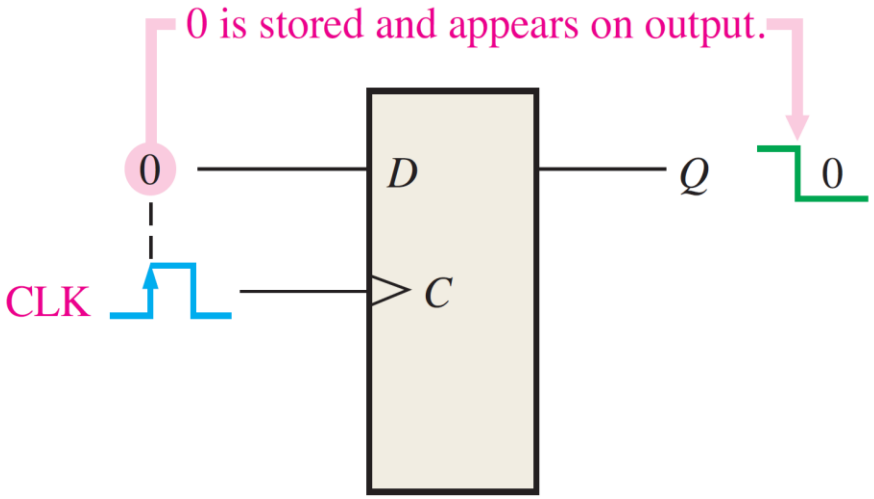
SHIFT REGISTERS

INTRODUCTION

- **Shift registers** consists of arrangements of flip-flops in which it used to:
  i. Store data
  ii. Shift data
- The storage capability of a register makes it important type of memory device.



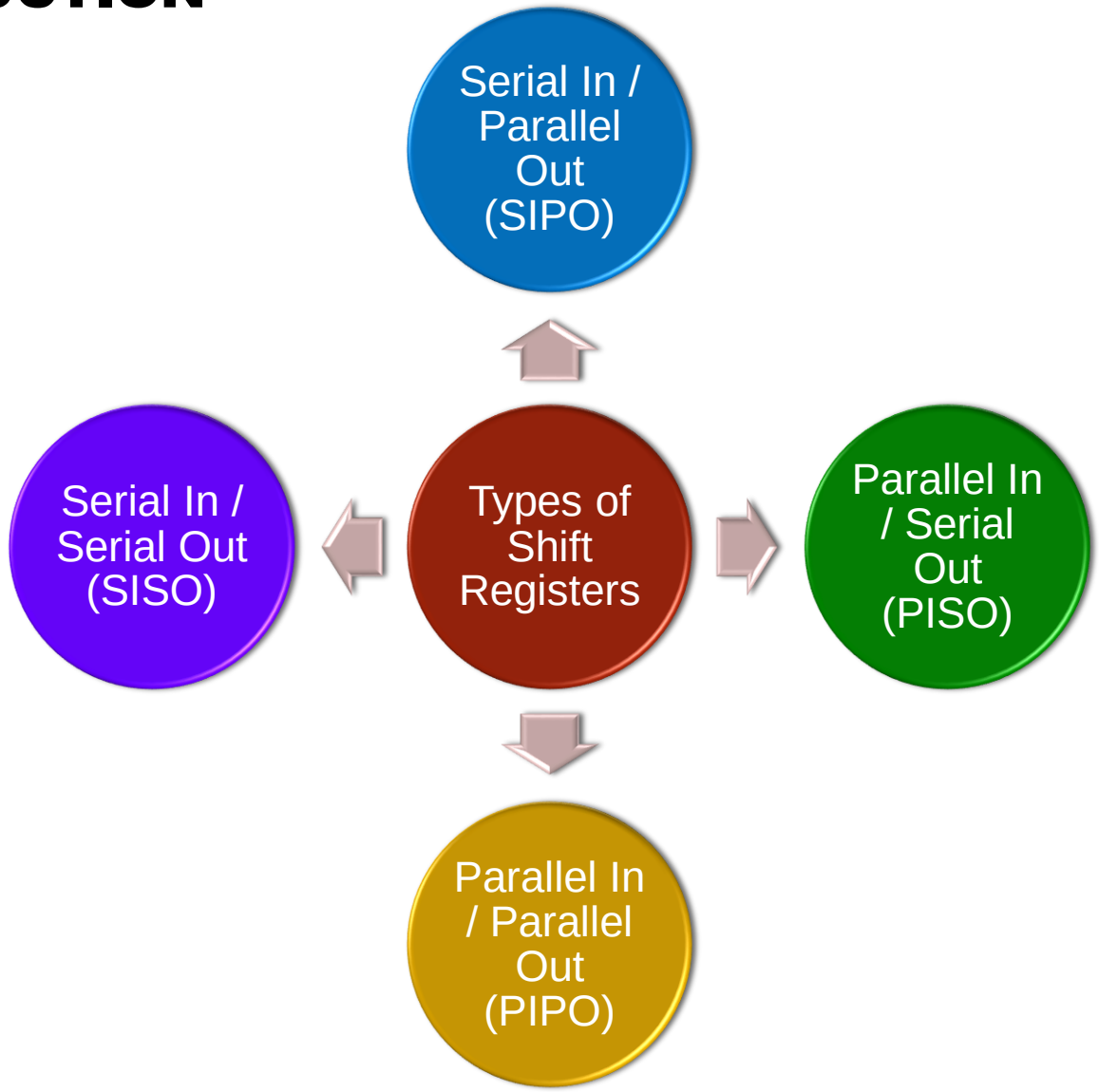
Concept of storing 1 for D flip-flop



Concept of storing 0 for D flip-flop

SHIFT REGISTERS

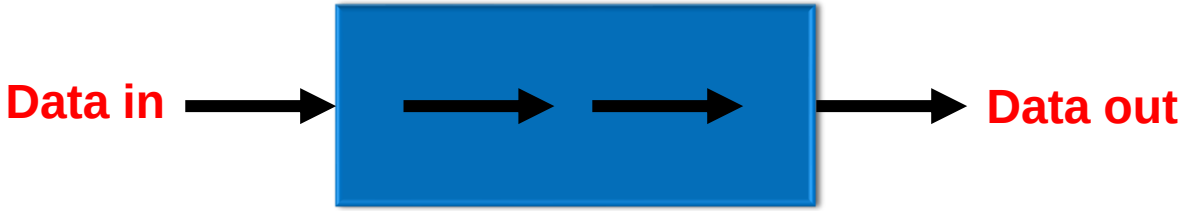
INTRODUCTION



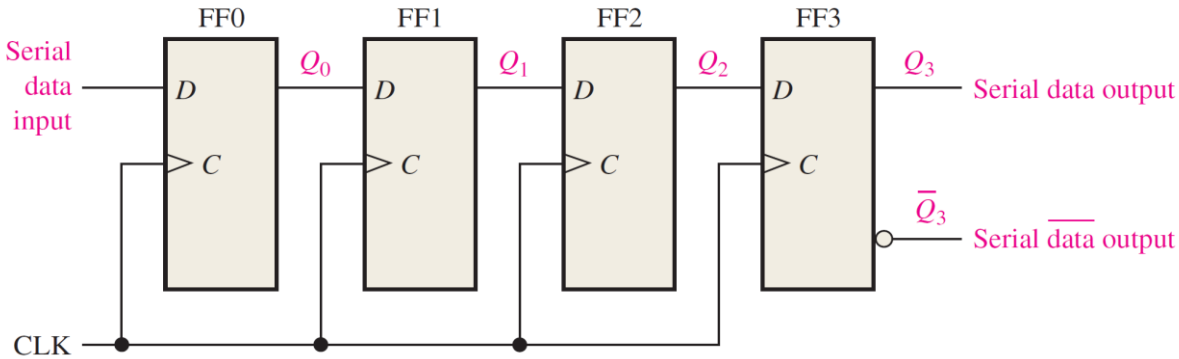
SHIFT REGISTERS

SISO

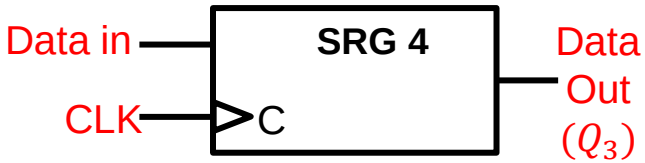
- The **SISO** accept data serially: one bit at a time on a single line.



- Every clock pulse, one bit of **Data in** will enter the shift register and at the same time one bit will be shifted to **Data out**.



Example of 4-bit SISO implemented using D flip-flops



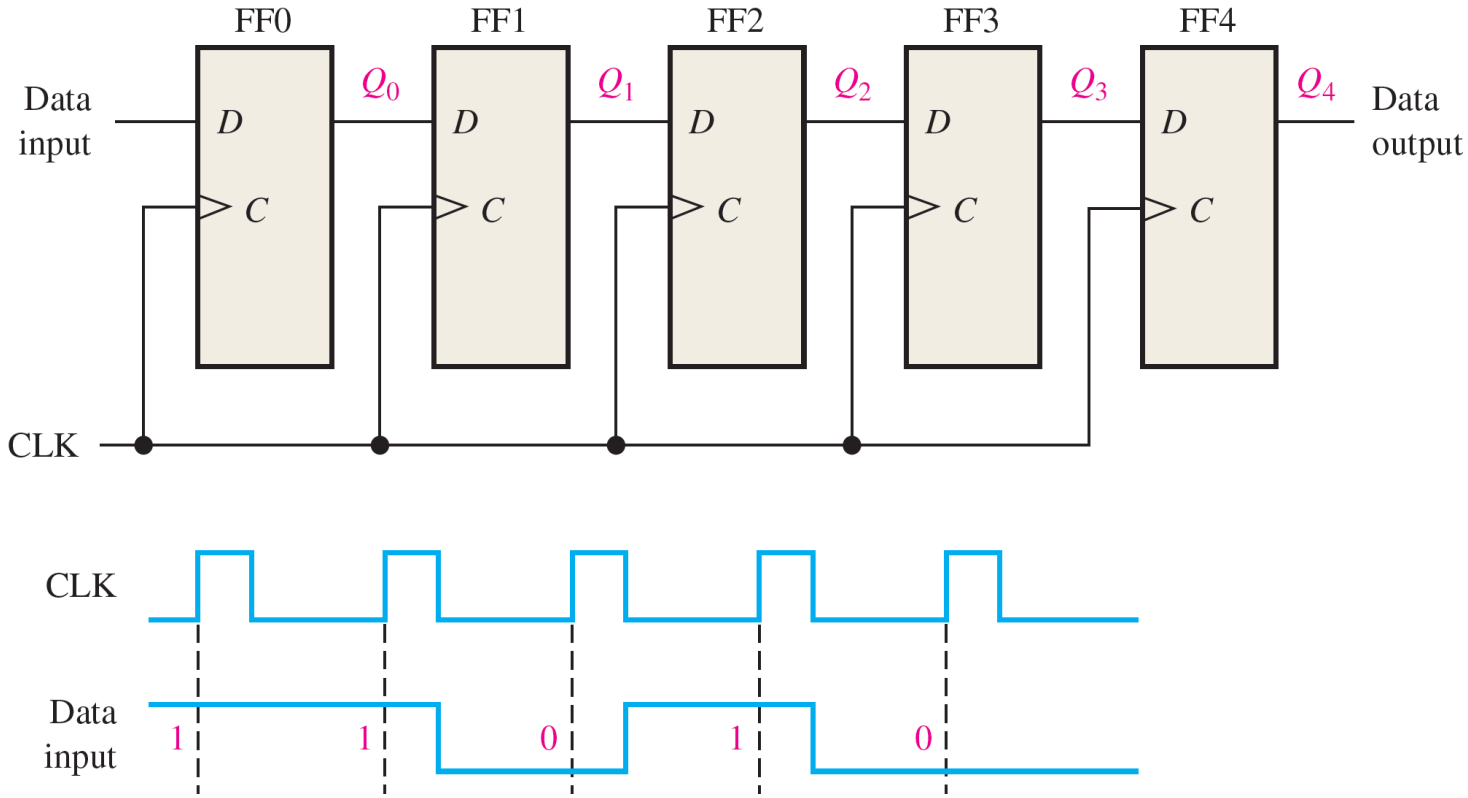
Logic symbol for 4-bits SISO

SHIFT REGISTERS

SISO

Example

Shows the state of 5-bit register in figure below for the specified data input and waveforms. Assume the register is initially CLEAR (all 0s)

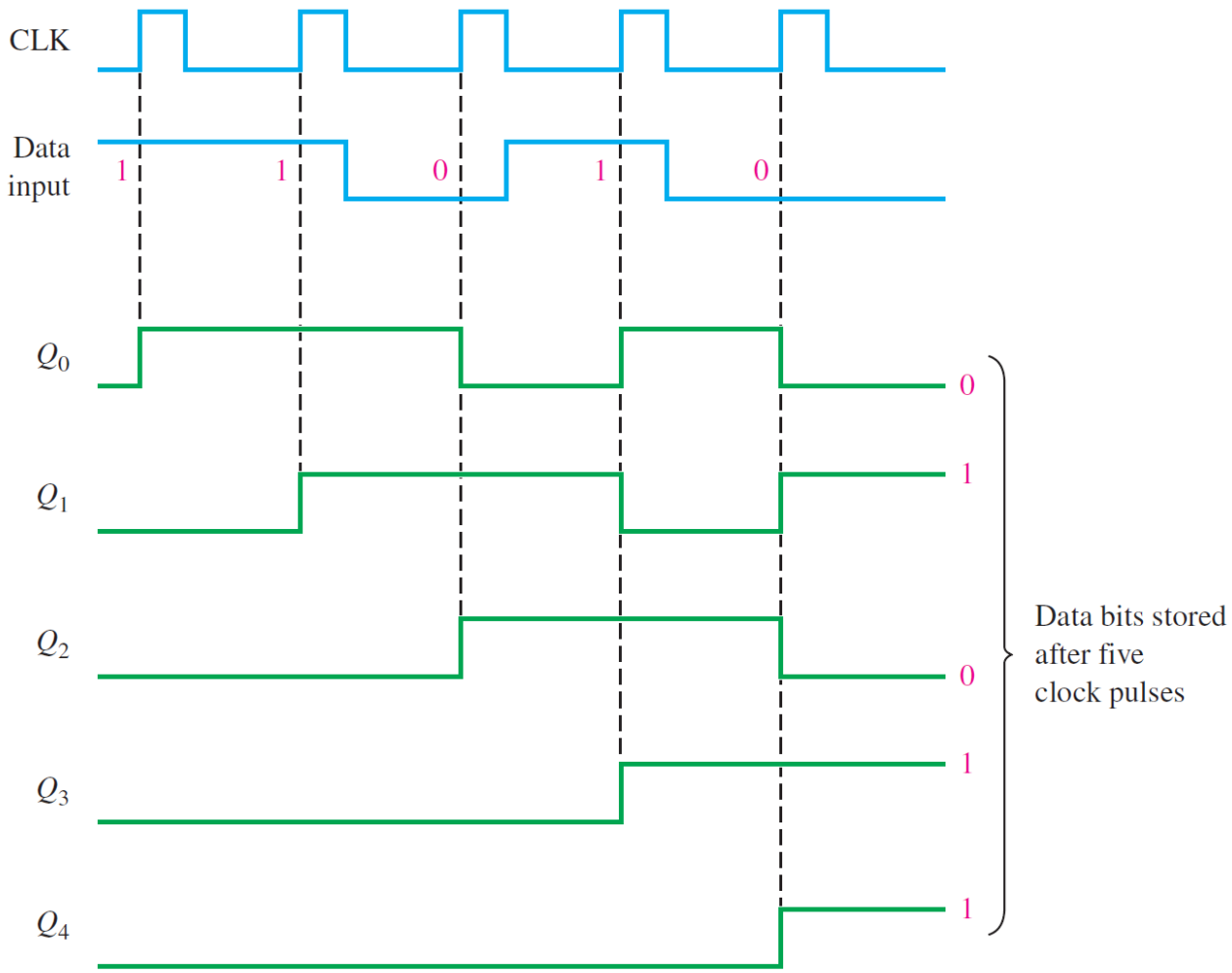


Example of 5-bit SISO implemented using D flip-flops

SHIFT REGISTERS

SISO

Solution



SHIFT REGISTERS

SISO

INPUT: 11010

Solution

	CLK	FF0	FF1	FF2	FF3	FF4
0 1 0 1 1	Initial	0	0	0	0	0
0 1 0 1 1 →	1	→ **1**	0	0	0	0
0 1 0 1 →	2	→ **1**	1	0	0	0
0 1 0 →	3	→ **0**	1	1	0	0
0 1 →	4	→ **1**	0	1	1	0
0 →	5	→ **0**	1	0	1	1
	6	→ **0**	0	1	0	1
	7	→ **0**	→ **0**	0	1	0
	8	→ **0**	→ **0**	→ **0**	0	1
	9	→ **0**	→ **0**	→ **0**	→ **0**	0
	10	→ **0**	→ **0**	→ **0**	→ **0**	→ **0**

Data bits **stored** after 5 cycles

Data bits **shifted out** after another 5 cycles

Note: Therefore for n bit SISO;

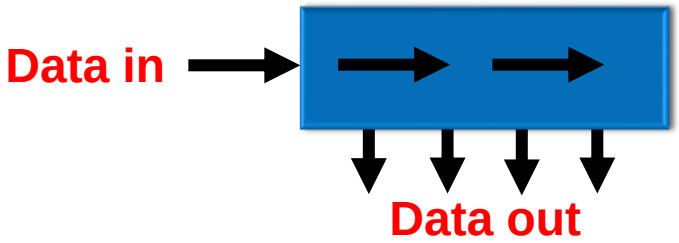
- It requires n clock cycle to shift in all the data set.
- It requires another n clock cycle to shifted out all the data set.

Output **read one bit at a time** until 9 cycles.

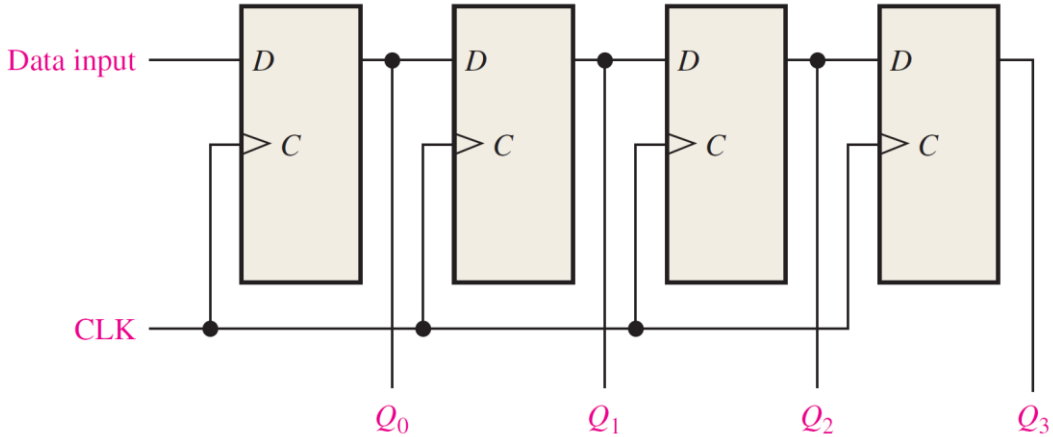
SHIFT REGISTERS

SIPO

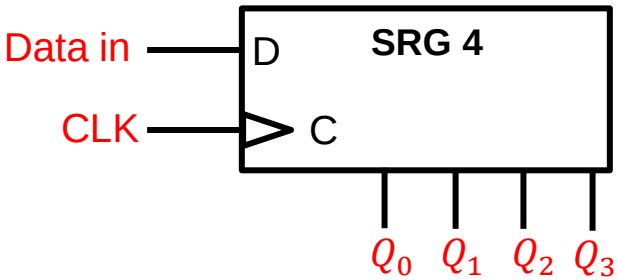
- The **SIPO**: Data bits entered serially at a time but the output is parallel; all bits are available simultaneously but only after N clock cycle for N bit SIPO.



- For SISO, output only at Q_3. But the output for SIPO is at Q_3, Q_2, Q_1 and Q_0.



Example of 4-bit SIPO implemented using D flip-flops

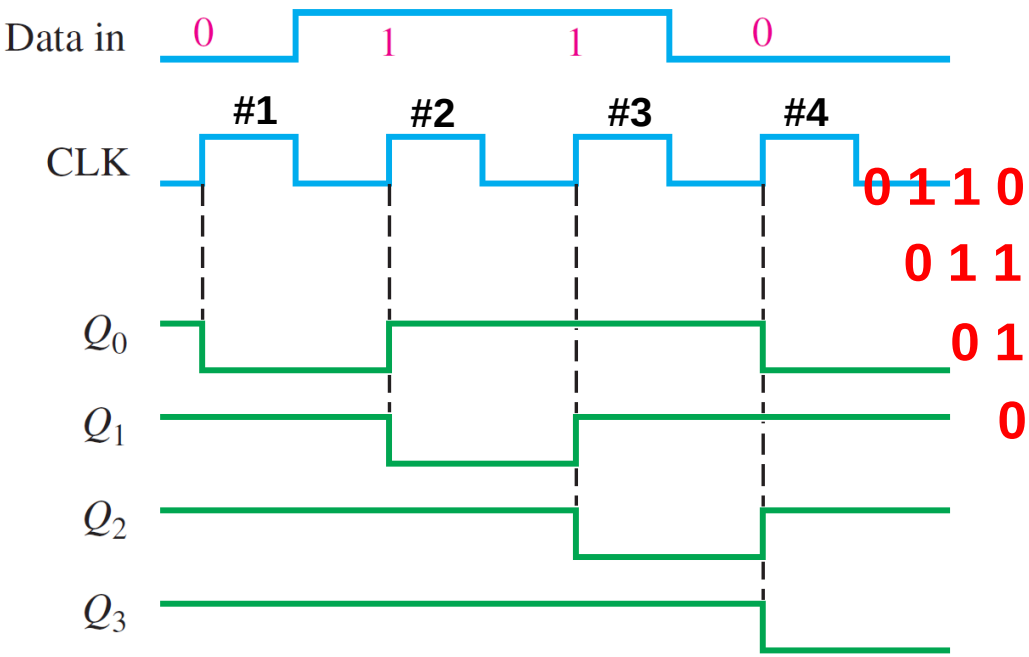


Logic symbol for 4-bits SIPO

SHIFT REGISTERS

SIPO

- The timing shown the waveform diagram when ‘0110’ is shifted through the register.
- Assume, the register is initially SET (all 1’s)



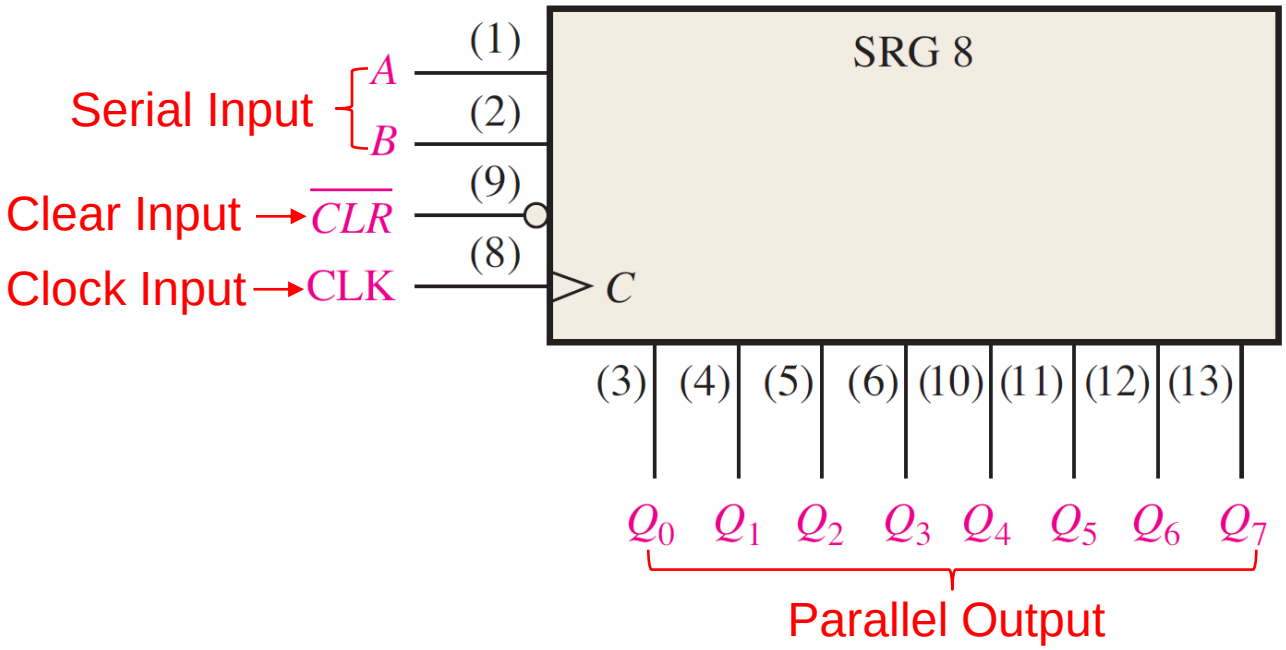
CLK	FF0	FF1	FF2	FF3
Initial	1	1	1	1
→ 1 →	0	1	1	1
→ 2 →	1	0	1	1
→ 3 →	1	1	0	1
→ 4 →	0 →	1	1	0
	Q0	Q1	Q2	Q3

Outputs **stored & read simultaneously** after 4 cycles

SHIFT REGISTERS

SIPO IC: 74x164

- The 74x164 is an IC shift register having SIPO operation.
- The device has two gates serial input, A and B, and a clear input ($\overline{CLR}$). The parallel outputs are Q_0 through Q_7.

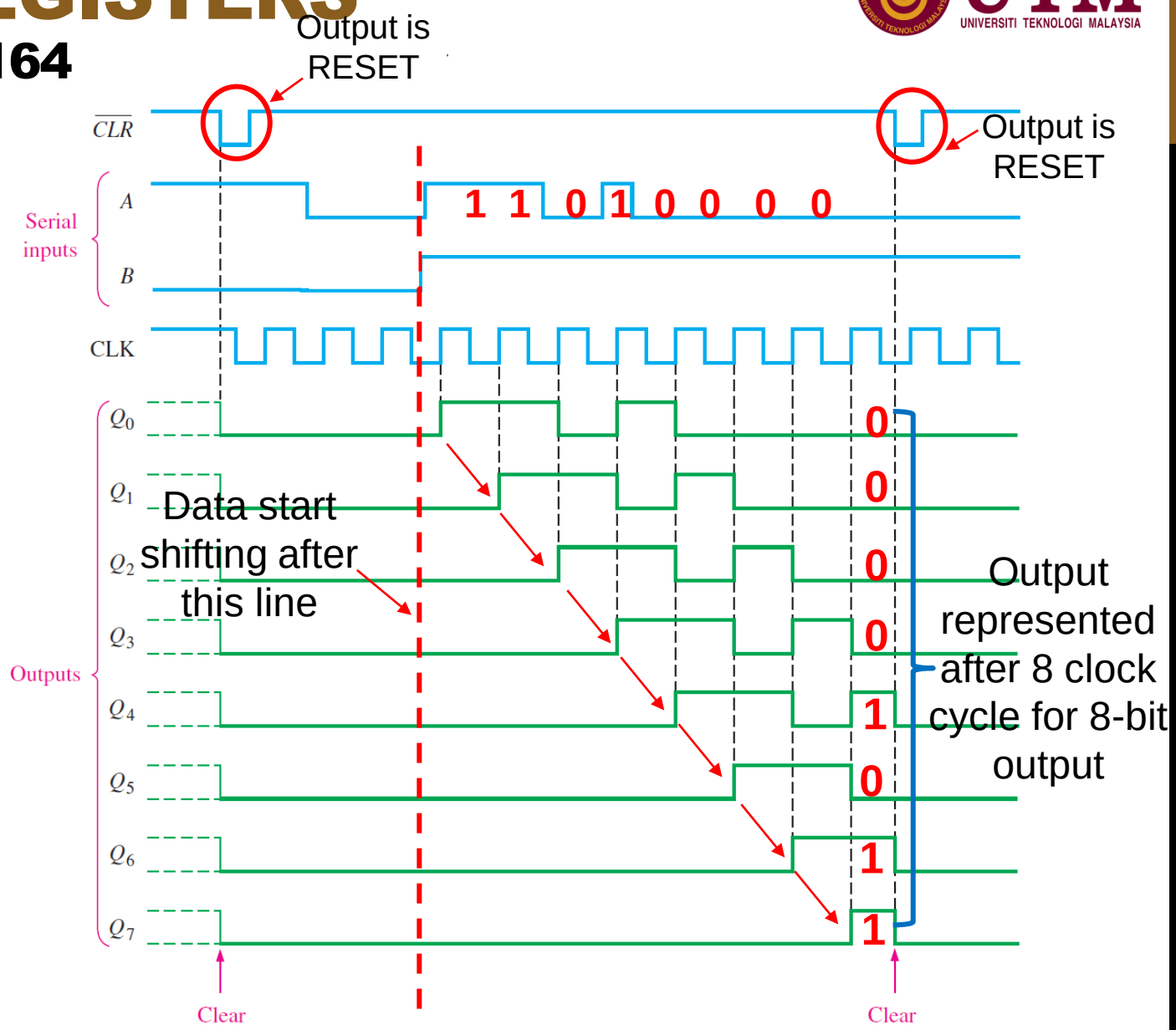


Logic Symbol of 74x164 SIPO IC

SHIFT REGISTERS

SIPO IC: 74x164

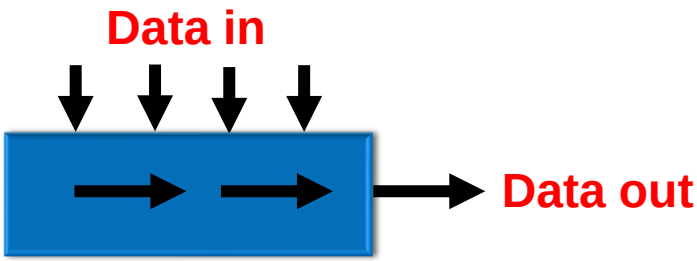
- Sample timing diagram for 74x164: Notice that, the serial input data on A only shifted into and through the register after input B goes HIGH.

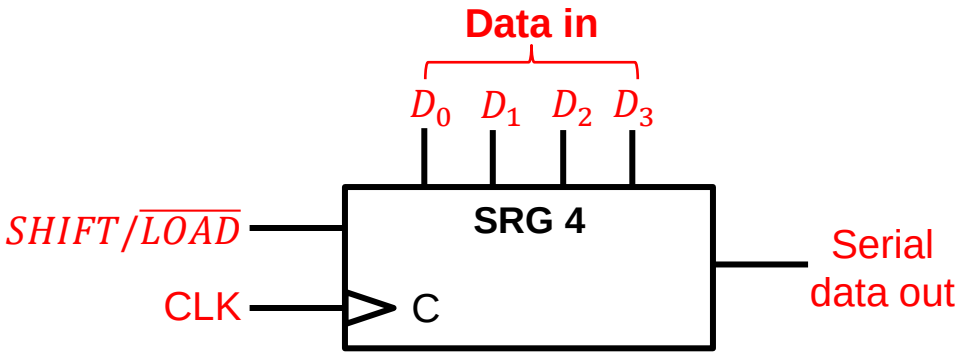


SHIFT REGISTERS

PISO

- The **PISO**: Data bits entered simultaneously into respective stage on parallel lines. While outputs are one bit at a time (same in SISO).
- For PISO, there are SHIFT/ $\overline{LOAD}$ input:
  - 0 = LOAD the input values
  - 1 = SHIFT them out at clock cycles

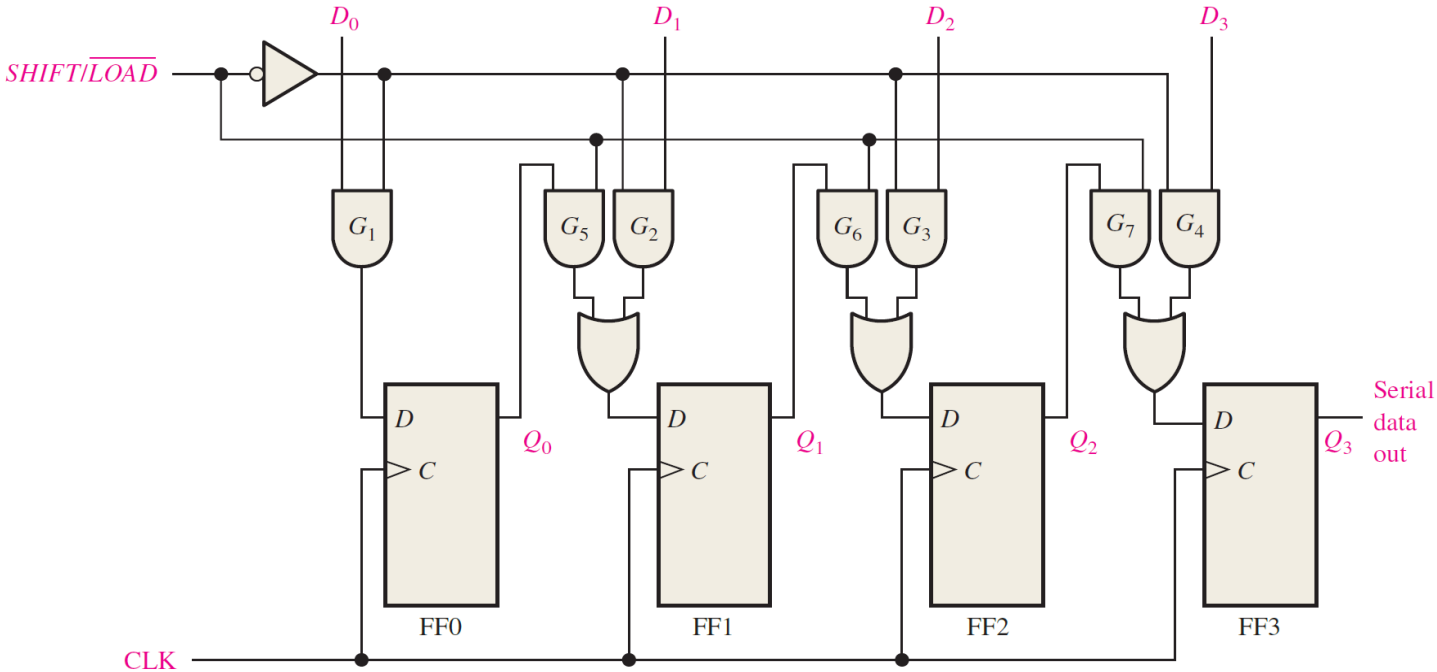




Logic symbol for 4-bits PISO

SHIFT REGISTERS

PISO



Logic diagram of 4-bit PISO using D flip-flops

$$DFF0 = D_0$$
$$DFF1 = Q_0\, SHIFT + \overline{LOAD} D_1$$
$$DFF2 = Q_1\, SHIFT + \overline{LOAD} D_2$$
$$DFF3 = Q_2\, SHIFT + \overline{LOAD} D_3$$

SHIFT REGISTERS

PISO

- LOAD mode:
  - Input must be available at parallel input in: D_3, D_2, D_1 and D_0.
  - When SHIFT/$\overline{LOAD} = 0$, the data loaded to the 4 flip-flop inside the shift register at the positive edge of the clock.
  - The MSB of the data will appear at Data out.
- SHIFT mode:
  - When SHIFT/$\overline{LOAD} = 1$, data will be shifted on every positive edge of the clock cycle.
  - After 4 clock cycle, the LSB data will appear at Data out.
  - At the 5th clock cycle, all the 4 bit data set will be lost and replaced by a new set of 4 bit data.

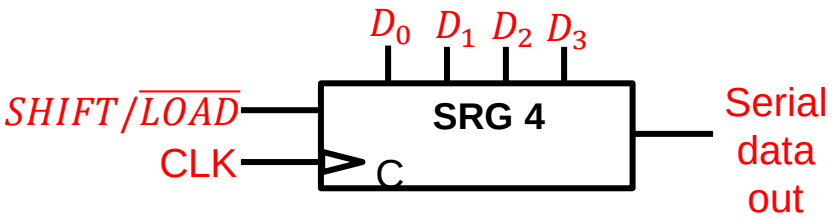
Logic symbol of 4-bit PISO

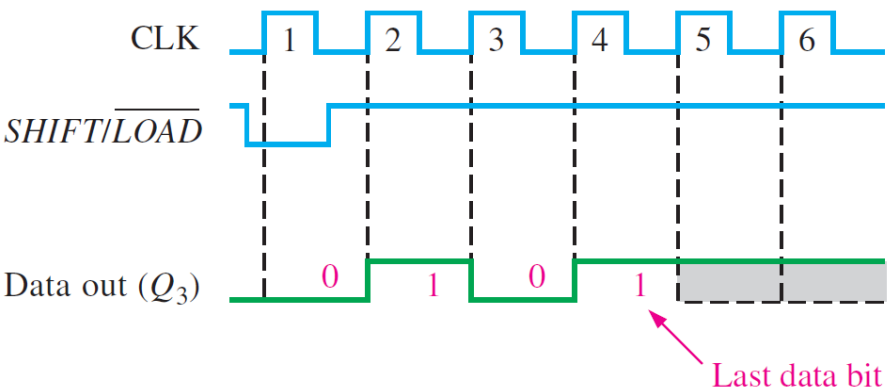
SHIFT REGISTERS

PISO

Example

Show the data output waveform for a 4-bit register with the parallel input data and SHIFT/$\overline{LOAD}$ waveform as follow:





LOAD

CLK	$S/\bar{L}$	FF0	FF1	FF2	FF3
1	0	1	0	1	**0**
2	1	X	1	0	**1**
3	1	X	X	1	**0**
4	1	X	X	X	**1**
5	1	X	X	X	X

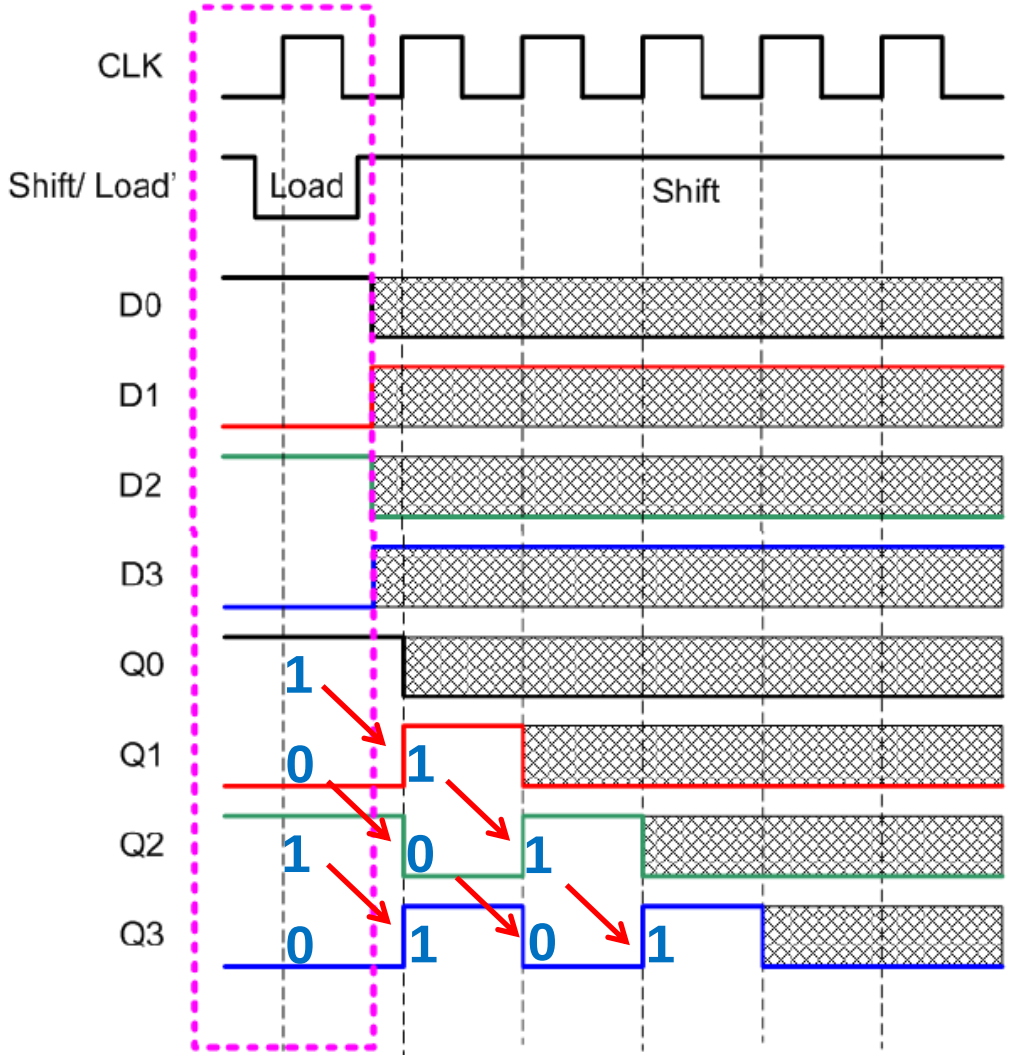
Data bits **shifted out** after another 5 cycles

Output **read one bit at a time** until 4 cycles

SHIFT REGISTERS

PISO

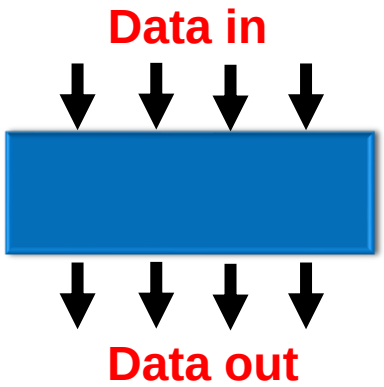
Solution

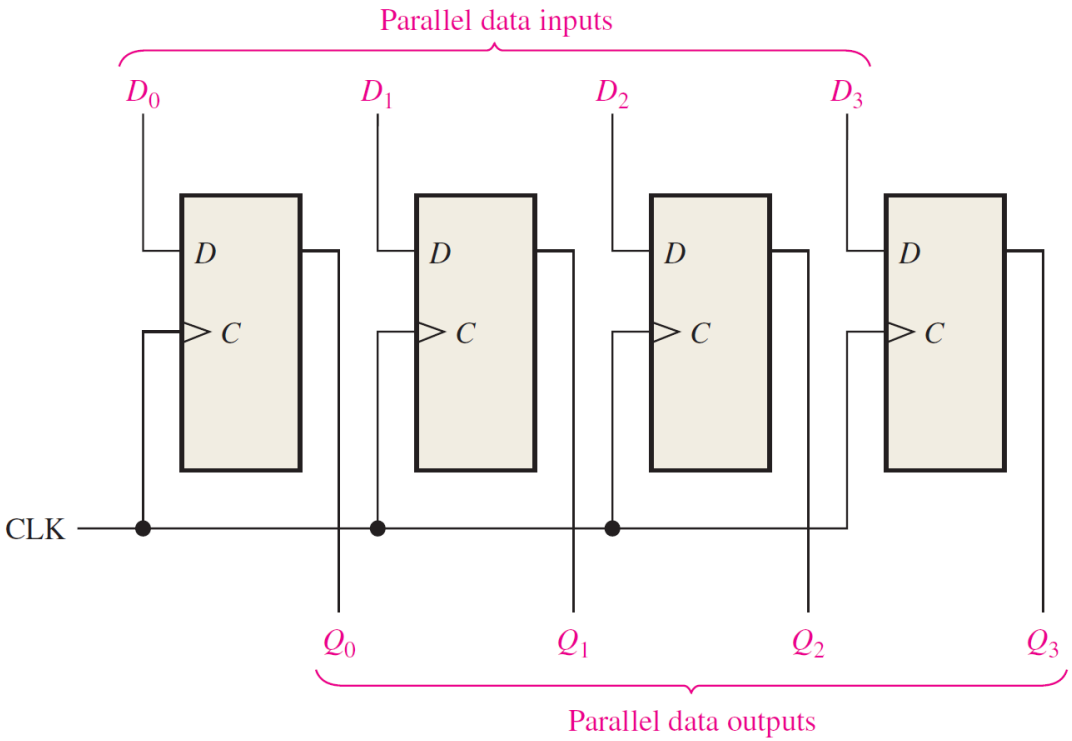


SHIFT REGISTERS

PIPO

- The **PIPO**: Input and output done in parallel.
- When enter the inputs, bits appear on the parallel outputs.





Logic diagram of 4-bit PIPO using D flip-flops

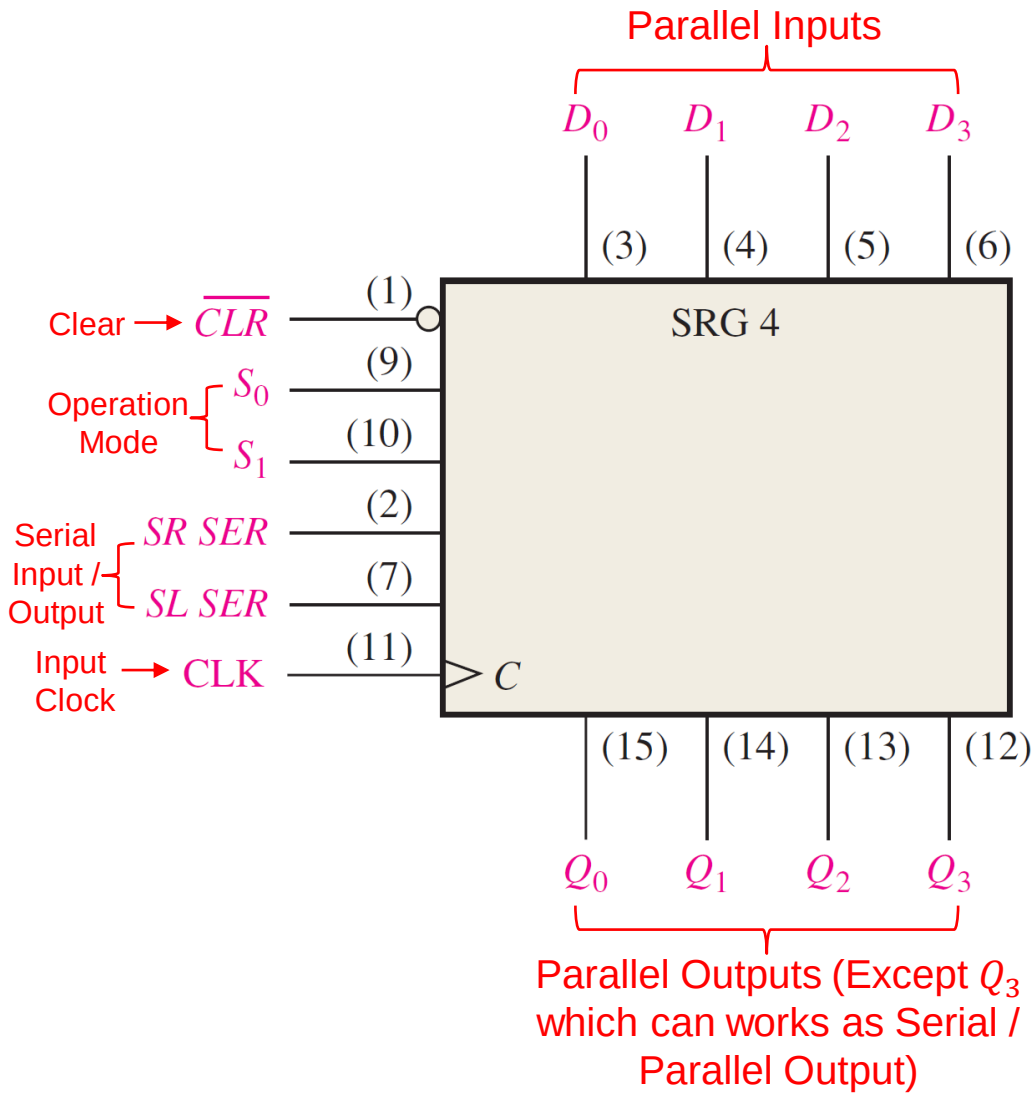
SHIFT REGISTERS

UNIVERSAL BIDIRECTIONAL IC: 74x194

- The **74x194** is a universal bidirectional shift register.
- It has both parallel and serial input/output capability. Can implement:
  i. SISO- input: **SR SER**, output: Q_3 for right shift. OR input: **SL SER**, output: Q_3 for left shift.
  ii. SIPO- input: either **SR SER** or **SL SER**, output: Q_3, Q_2, Q_1, Q_0.
  iii. PISO- input: D_3, D_2, D_1, and D_0, output: either **SR SER** or **SL SER**.
  iv. PIPO- input: D_3, D_2, D_1 and D_0, output: Q_3, Q_2, Q_1 and Q_0.

SHIFT REGISTERS

UNIVERSAL BIDIRECTIONAL IC: 74x194



Logic Symbol of 74x194 Universal Bidirectional IC

SHIFT REGISTERS

UNIVERSAL BIDIRECTIONAL IC: 74x194

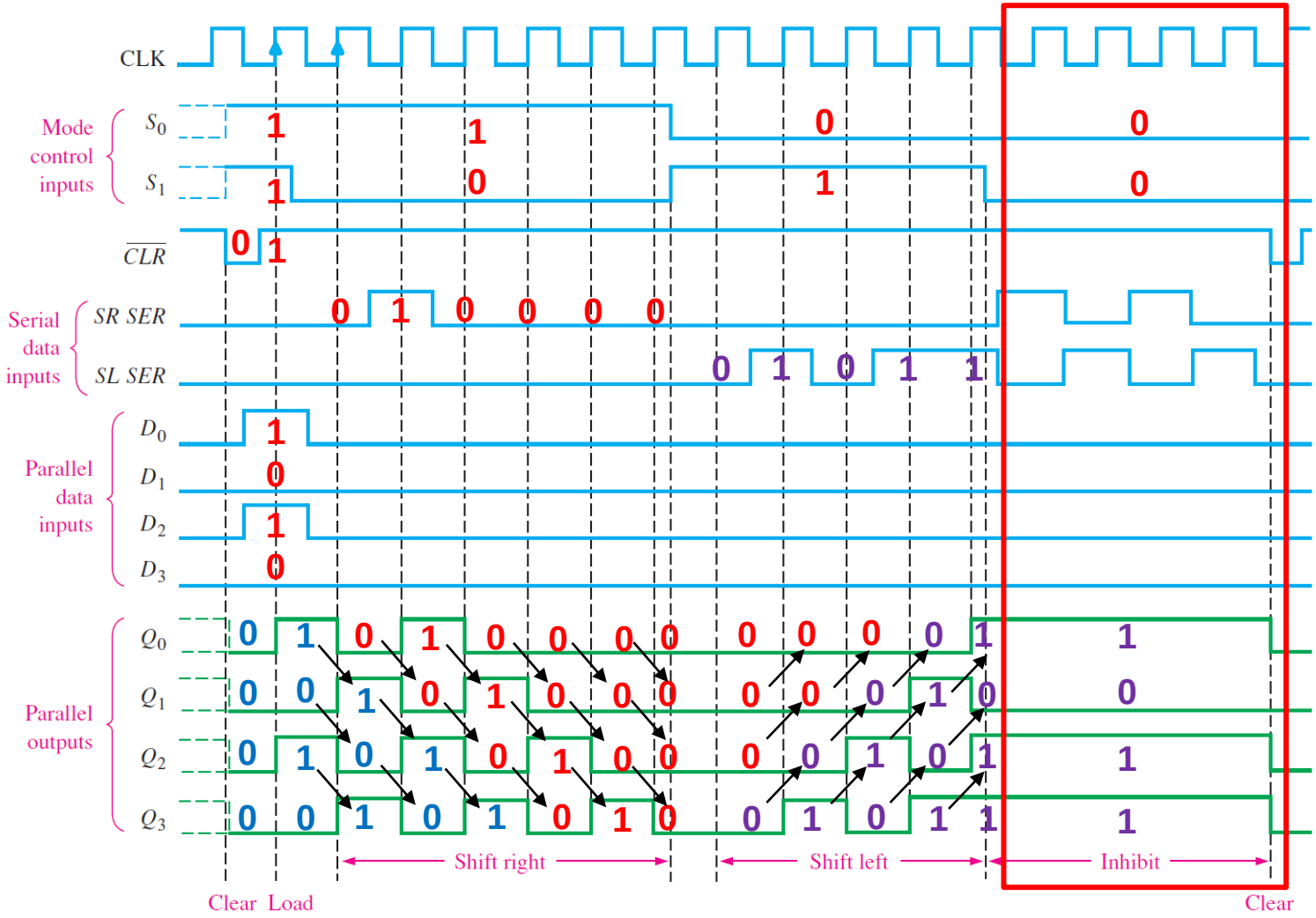
- Mode operation is controlled by S1S0.

Function	Input		Output			
	S_1	S_0	Q_{0+}	Q_{1+}	Q_{2+}	Q_{3+}
Hold	0	0	Q_0	Q_1	Q_2	Q_3
Shift Right	0	1	SR SER	Q_0	Q_1	Q_2
Shift Left	1	0	Q_1	Q_2	Q_3	SL SER
Parallel Load	1	1	D_0	D_1	D_2	D_3

SHIFT REGISTERS

UNIVERSAL BIDIRECTIONAL IC: 74x194

- Timing diagram of 74x194 shift register.

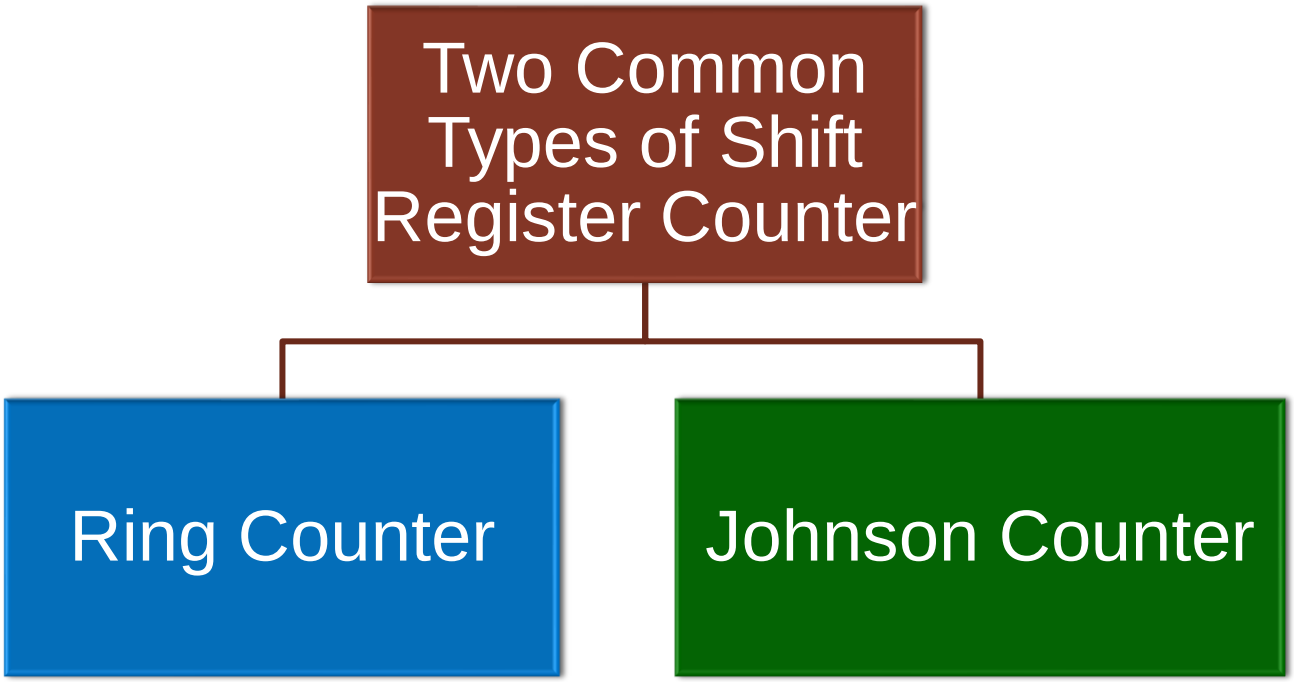


SHIFT REGISTER COUNTERS

SHIFT REGISTER COUNTERS

INTRODUCTION

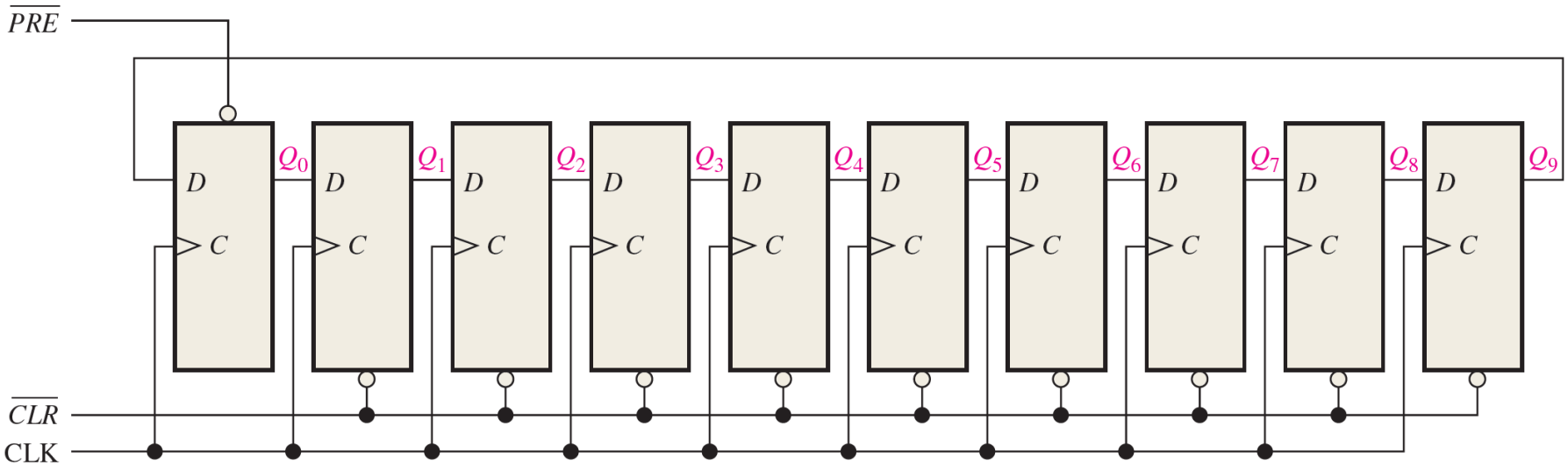
- **Shift register counter** is a shift register with serial output connected back to serial input to produce special sequences.



SHIFT REGISTER COUNTERS

RING COUNTER

- **Ring counter** is a SISO type shift register with the final output feed back to the first input.



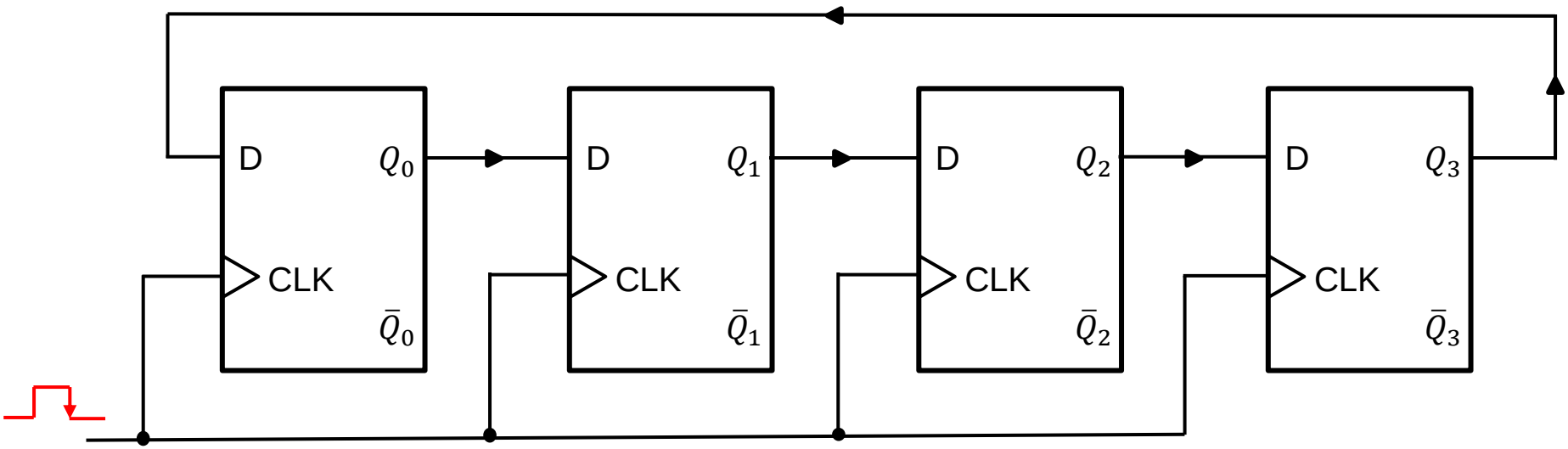
- At first, the FFO (first FF) is preset to **1** by making $\overline{PRE} = 0$, momentarily. All of other FFs are cleared, **0**, by making $\overline{CLR} = 0$.
- Then, the **1** is shifted to round the ring in the next cycle with $Q1 = 1$, while other FF outputs are **0**, thus represent as state 1.
- The **1** then keep circulates in the shift register.
- For n bit ring counter, the number of state it has is n (MOD n).

SHIFT REGISTER COUNTERS

RING COUNTER

Example

For 4-bit ring counter, there are 4 FFs which make it Mod-4 counter.

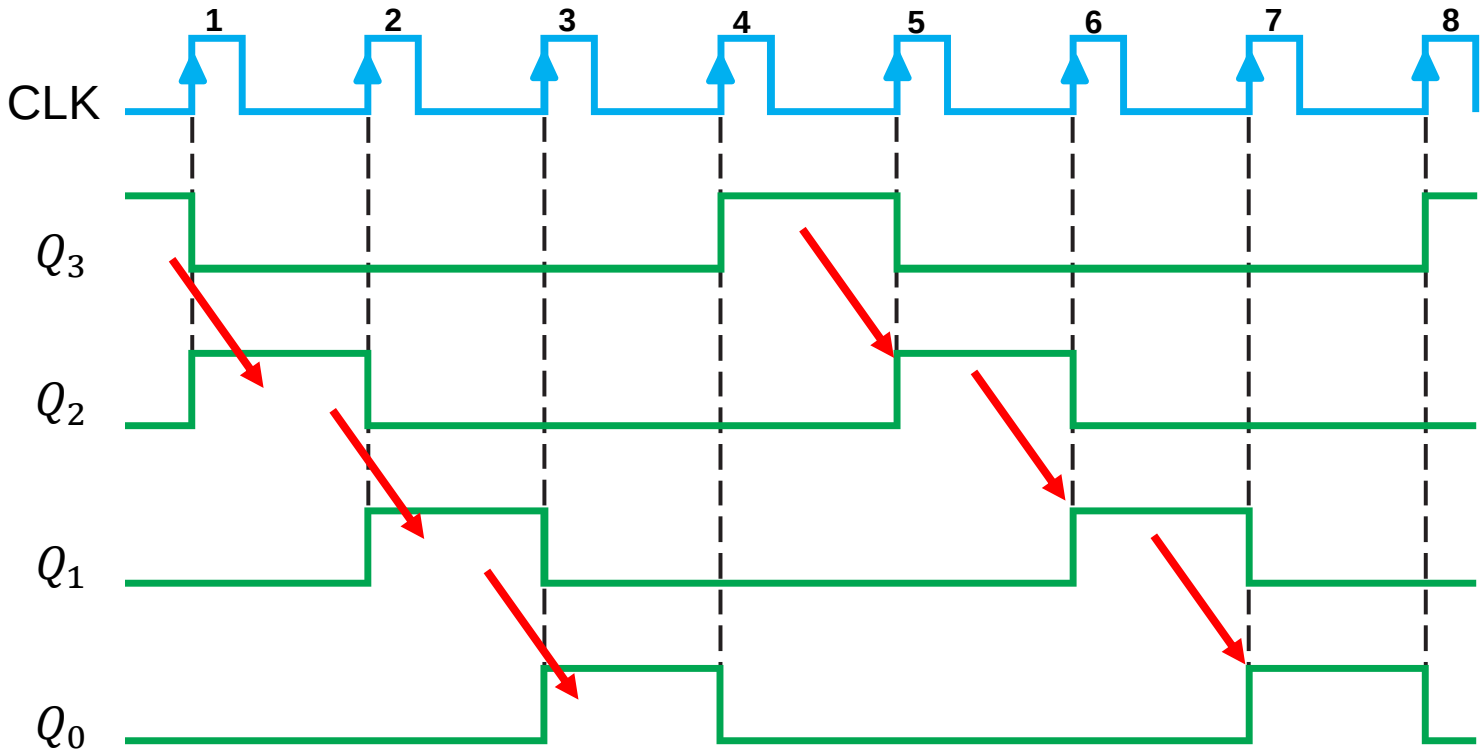


SHIFT REGISTER COUNTERS

RING COUNTER

Example

In Ring Counter, only FF is HIGH while others are LOW, thus every state has unique **1** base on the position of **1**.



SHIFT REGISTER COUNTERS

RING COUNTER

Example

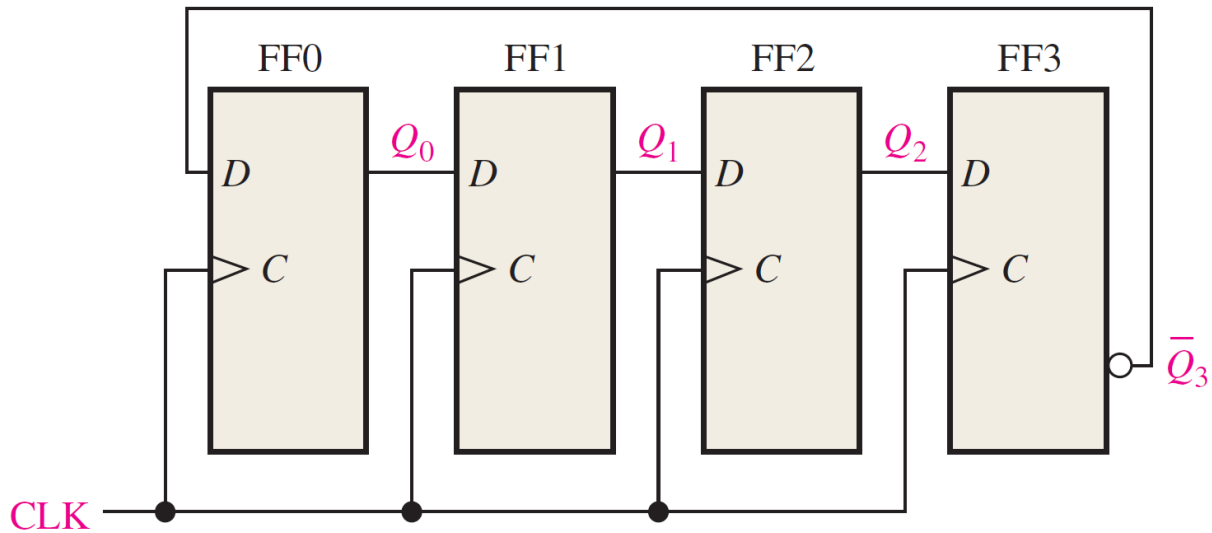
In Ring Counter, only FF is HIGH while others are LOW, thus every state has unique **1** base on the position of **1**.

Q_3	Q_2	Q_1	Q_0	Clock Pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7

SHIFT REGISTER COUNTERS

JOHNSON COUNTER

- **Johnson counter** is also known as Moebius counter or twisted ring counter.



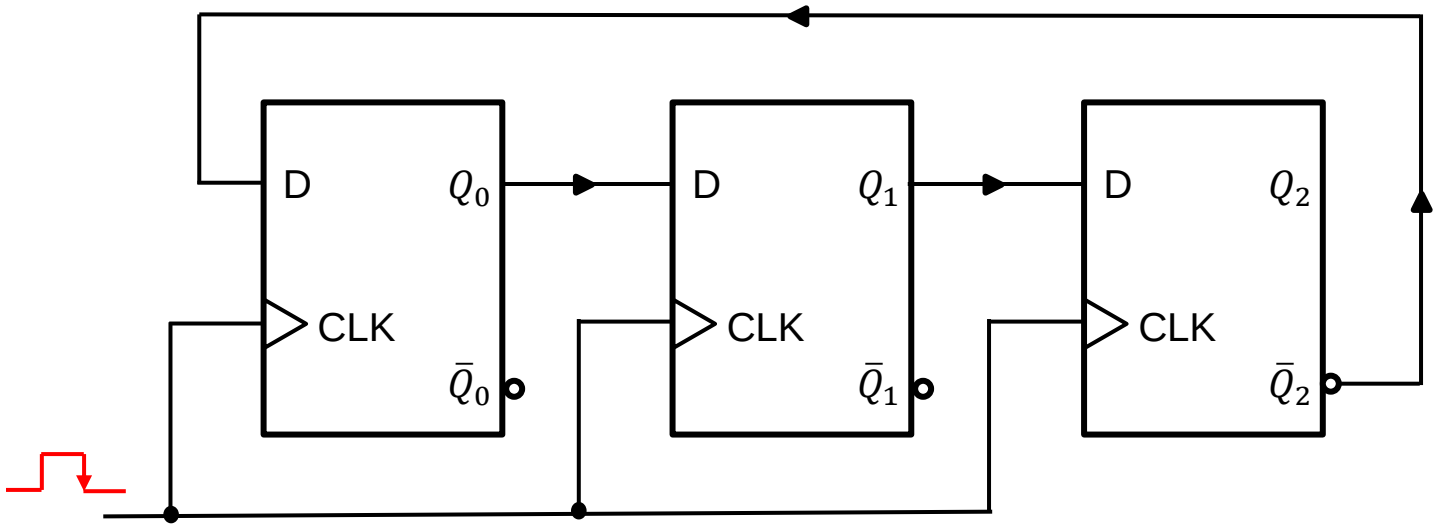
- In Johnson counter, the last complemented output is fed back as an input to the FF.
- Number of unique states are 2 times the number of bits (FF).
  - 4 bits 2×4 = 8 states
  - 5 bits 2×5 = 10 states

SHIFT REGISTER COUNTERS

JOHNSON COUNTER

Example

For 3-bit Johnson counter, there are 3 FFs which make it Mod 6 counter. The $\overline{Q_2}$ is connected to D0.



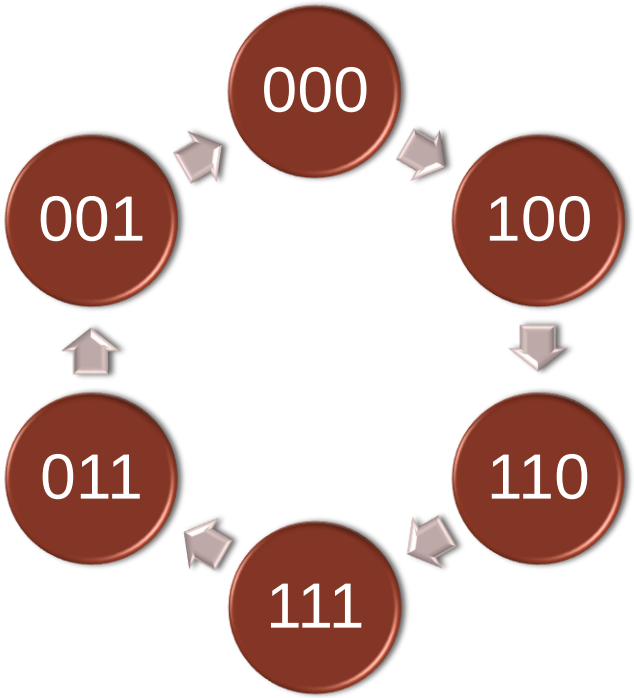
SHIFT REGISTER COUNTERS

JOHNSON COUNTER

Example

After the 6th cycles, the pattern repeat again. Thus, 6 unique states Mod 6.

CLK	FF0	FF1	FF2	$\overline{Q_2}$
Initial	0	0	0	1
1	1	0	0	1
2	1	1	0	1
3	1	1	1	0
4	0	1	1	0
5	0	0	1	0
6	0	0	0	1
7	1	0	0	1
8	1	1	0	1

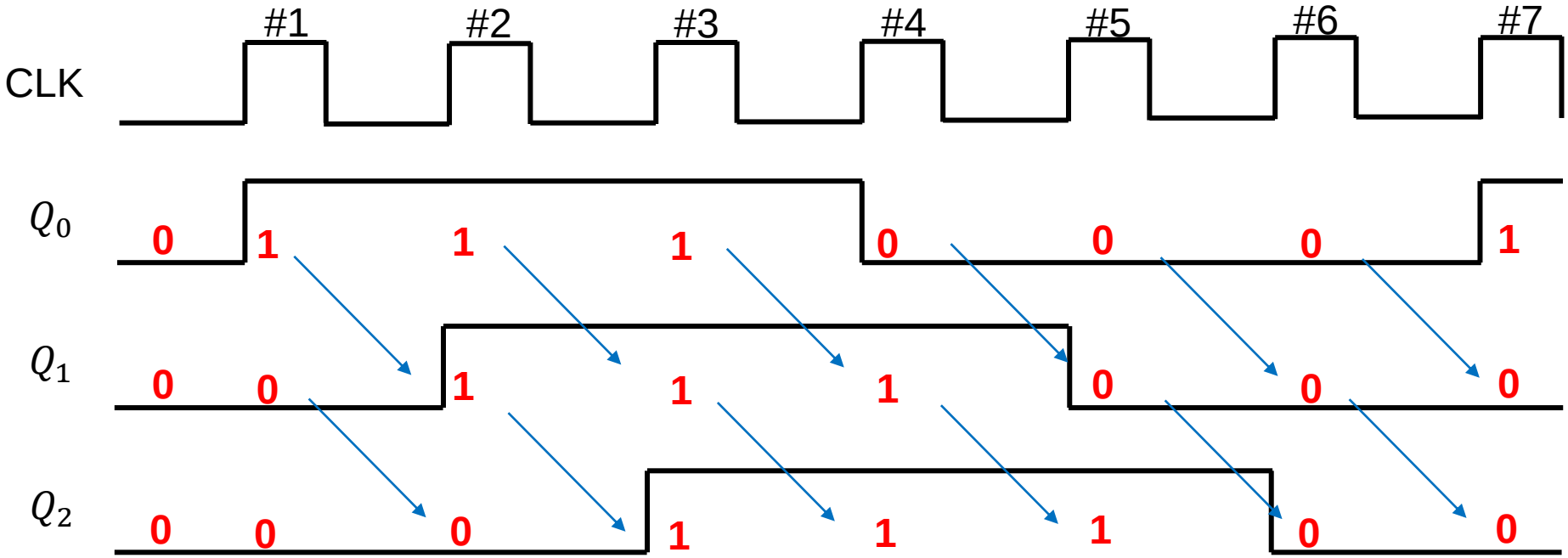


SHIFT REGISTER COUNTERS

JOHNSON COUNTER

Example

After the 6th cycles, the pattern repeat again. Thus, 6 unique states Mod 6.



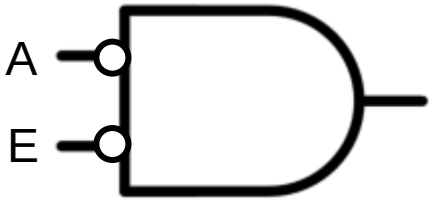
SHIFT REGISTER COUNTERS

JOHNSON COUNTER CHARACTERISTICS

- For n bit FF the MOD is $2n$.
- For Johnson counter, there are always exists 2 bit that are unique compared to other state.
  - ❖ It only requires 2 input decoder to decode a state.

Sequence Number	Flip-flip outputs				AND gate required for output
	A	B	C	D	
1	0	0	0	0	$\bar{A}\bar{E}$
2	1	0	0	0	$A\bar{B}$
3	1	1	0	0	$B\bar{C}$
4	1	1	1	0	$C\bar{E}$
5	1	1	1	1	AE
6	0	1	1	1	$\bar{A}B$
7	0	0	1	1	$\bar{B}C$
8	0	0	0	1	$\bar{C}E$

- Table shows 4 bit Johnson counter with 8 states.
- Each state has unique 2 input variable.
- The unique state simplify the decoder because a decoder with two inputs are required to decode any state.



Decoder for State 1

SHIFT REGISTER COUNTERS

COMPARISON OF COUNTER

- For a given n FF, the binary counter can produce the most state and the ring is the worst.
- In term of decoding, Ring counter is the best because there is no need for decoder to decode each state, binary counter is the worst because need more complex decoder.
- Johnson is always in the middle when comparing those features.

	MOD for n flip-flop	Decoder Input
Ring	n	No need for a decoder
Johnson	$2n$	2 input decoder
Binary	2^n	Usually >2 input decoder