Digital Logic and Circuit
Paper Code: CS-102

Outline

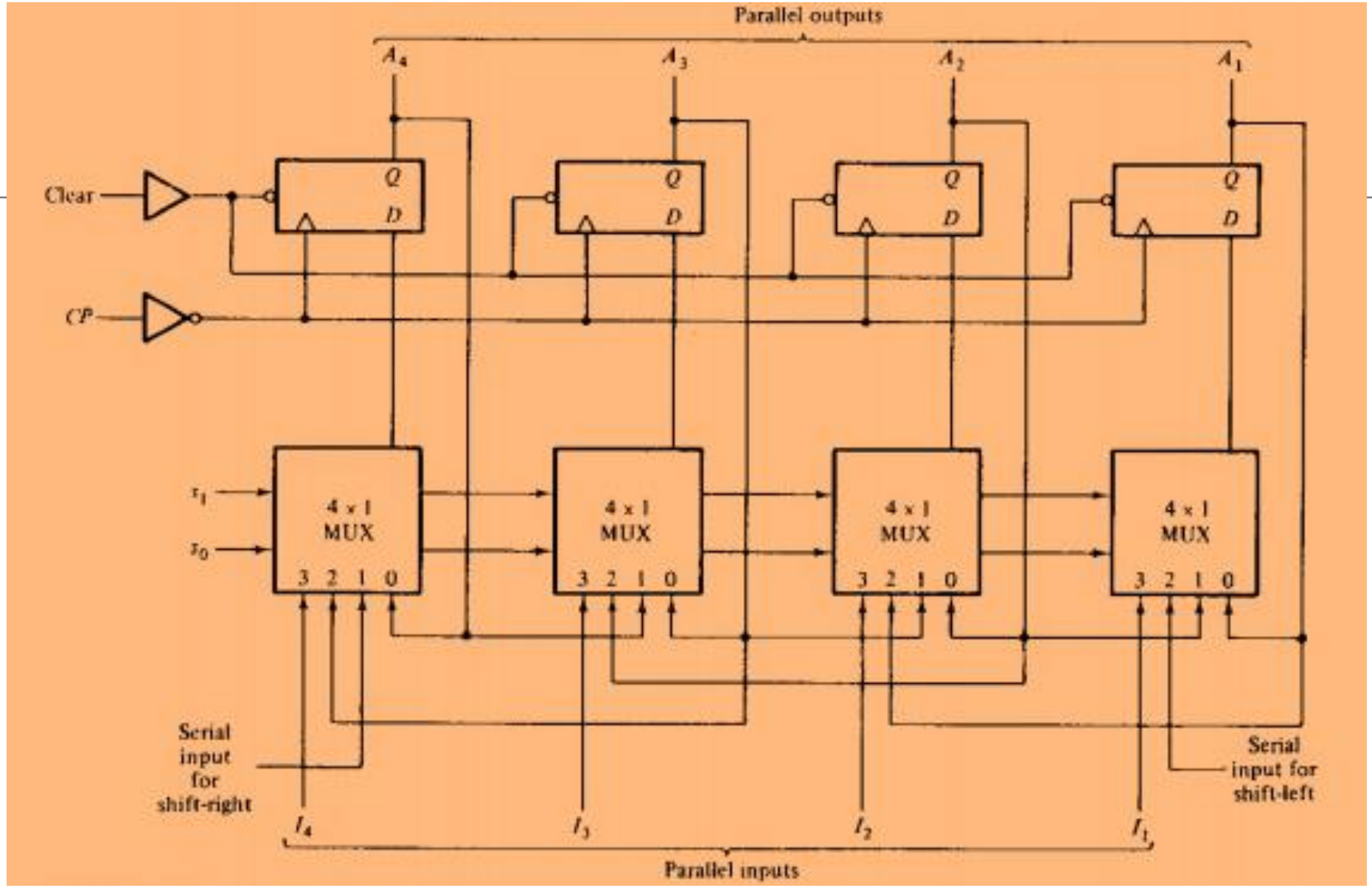
- **Sequential Circuit**
  - **Universal Shift Register**
  - **Counter**
    - **Synchronous**
    - **Asynchronous**

Universal Shift Register

- A register capable of shifting both right and left is called a bidirectional shift register.
- One that can shift in only one direction is called a unidirectional shift register.
- If the register has both shift and parallel-load capabilities, it is called a shift register with parallel load.
- If the register has left shift, right shift and parallel-load capabilities, it is called a Universal Shift Register.

The most general shift register has all the capabilities listed below. Others may have only some of these functions, with at least one shift operation.

1. A clear control to clear the register to 0.
2. A CP input for clock pulses to synchronize all operations.
3. A shift-right control to enable the shift-right operation and the serial input and output lines associated with the shift right.
4. A shift-left control to enable the shift-left operation and the serial input and output lines associated with the shift left.
5. A parallel- load control to enable a parallel transfer and the n input lines associated with the parallel transfer.
6. n parallel output lines.
7. A control state that leaves the information in the register unchanged even though clock pulses are continuously applied.


Parallel outputs
A_4
A_3
A_2
A_1
Clear
CP
Q
D
s_1
s_0
4 x 1
MUX
3 2 1 0
Serial
input
for
shift-right
Serial
input for
shift-left
I_4
I_3
I_2
I_1
Parallel inputs

- The diagram of a shift register that has all the capabilities listed above is shown in figure.
- It consists of four D flip-flops, although RS flip-flops could be used provided an inverter is inserted between the S and R terminals.
- The four multiplexers (MUX) are part of the register and are drawn here in block diagram form. The four multiplexers have two common selection variables, s1 and s0 .
- Input 0 in each MUX is selected when s1s0 = 00, input 1 is selected when s1so = 01, and similarly for the other two inputs to the multiplexers.

Function Table of Universal Register

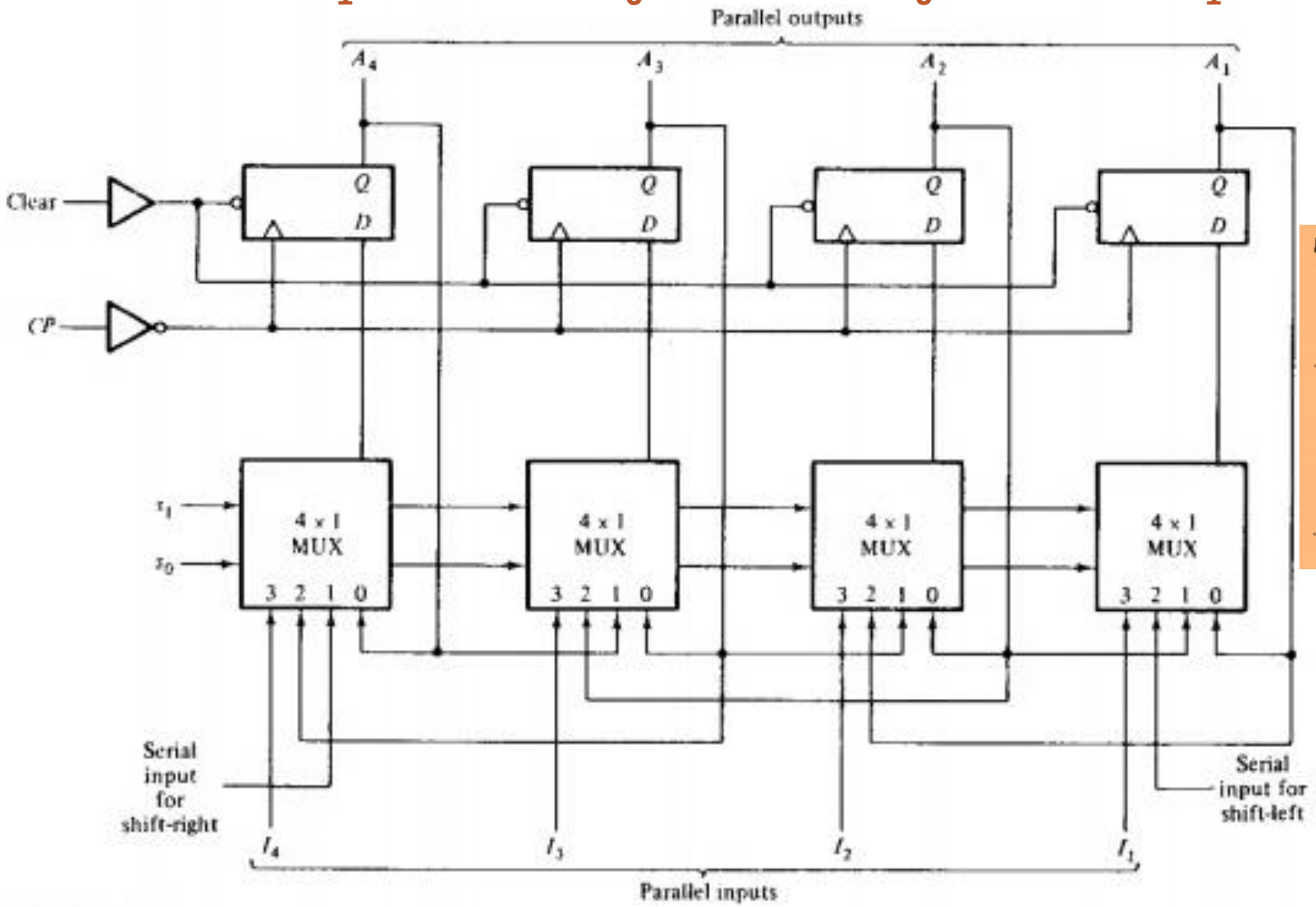
Function Table for the Register

Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

- When s1so = 00, the present value of the register is applied to the D inputs of the flip-flops.
- This condition forms a path from the output of each flip-flop into the input of the same flip-flop.
- The next clock pulse transfers into each flip-flop the binary value it held previously, and no change of state occurs.
- When s1s0 = 01, terminals 1 of the multiplexer inputs have a path to the D inputs of the flipflops.
- This causes a shift- right operation, with the serial input transferred into flip-flop A4 .
- When s1so = 10, a shift-left operation results, with the other serial input going into flip-flop A1 .
- Finally, when s1s0 = 11, the binary information on the parallel input lines is transferred into the register simultaneously during the next clock pulse.

Universal Shift Register

1 0 0 1



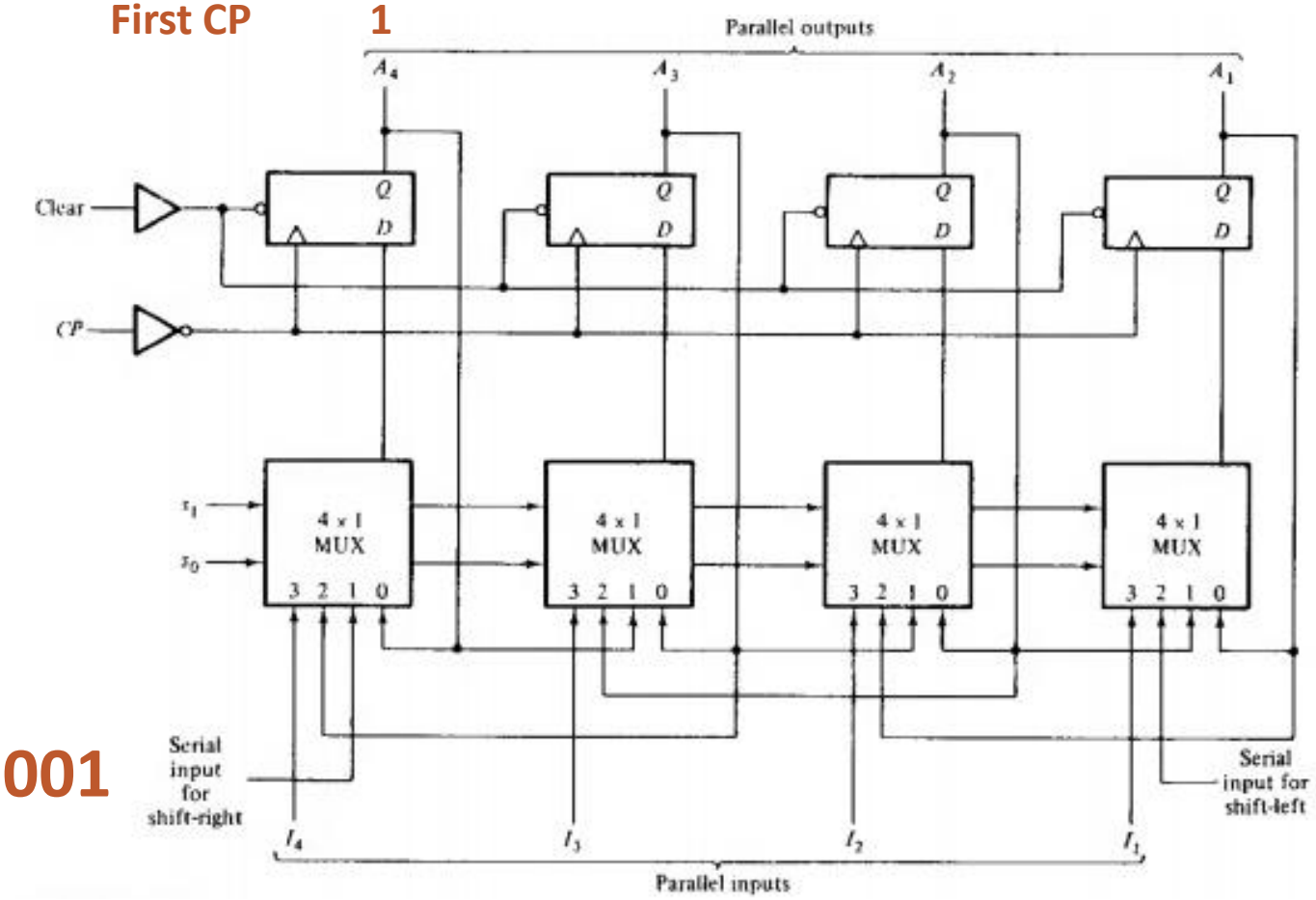
CP	D	Q(t+1)
0	X	memory
1	0	0
1	1	1

Function Table for the Register

Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Universal Shift Register: Shift Right operation

Fourth CP	**1**	**0**	**0**	**1**
Third CP	**0**	**0**	**1**	
Second CP	**0**	**1**		
First CP	**1**			



1001

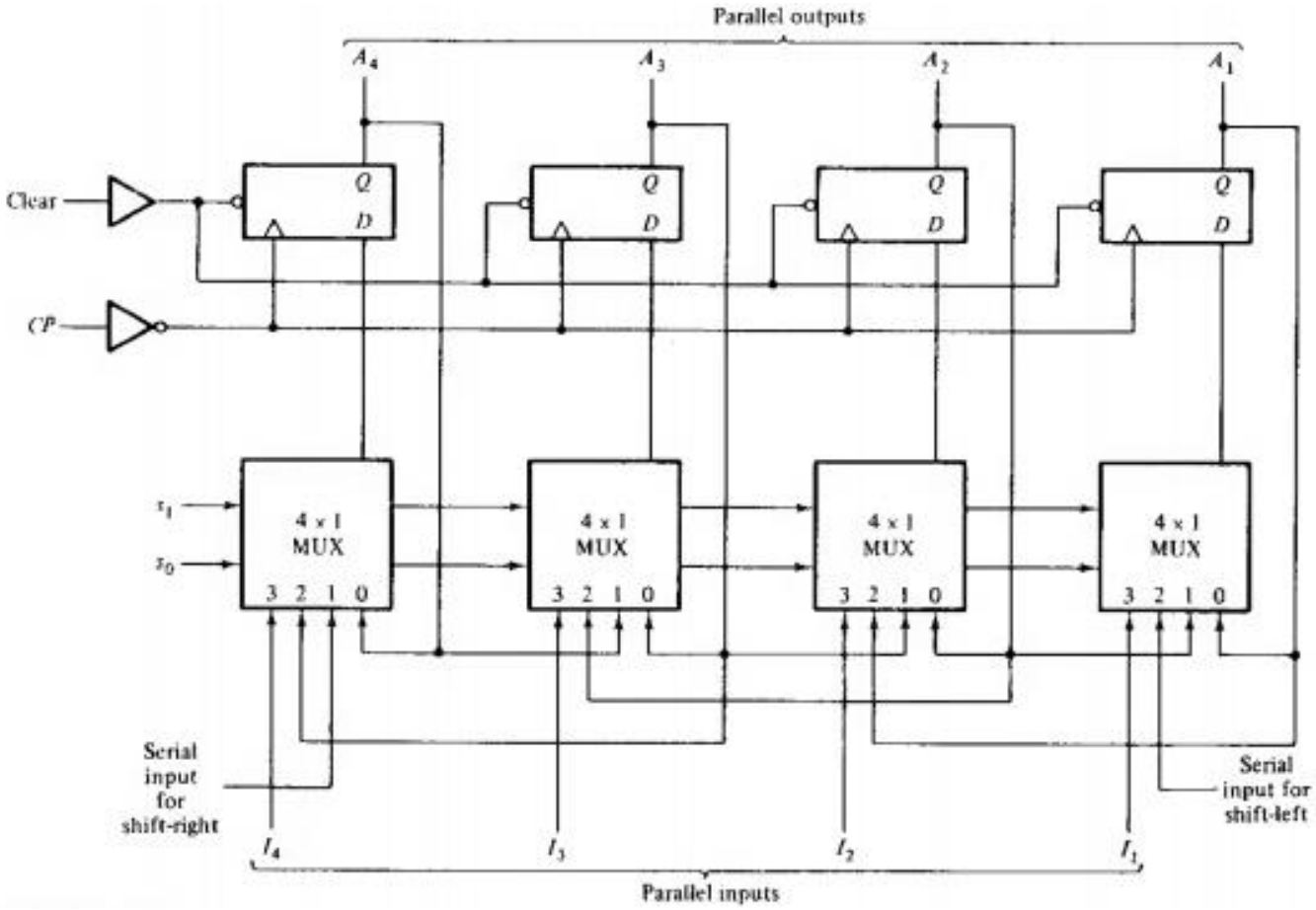
CP	D	Q(t+1)
0	X	memory
1	0	0
1	1	1

Function Table for the Register

Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Universal Shift Register: Shift Left operation

Fourth CP	**1**	**0**	**0**	**1**
Third CP		**1**	**0**	**0**
Second CP			**1**	**0**
First CP				**1**



CP	D	Q(t+1)
0	X	memory
1	0	0
1	1	1

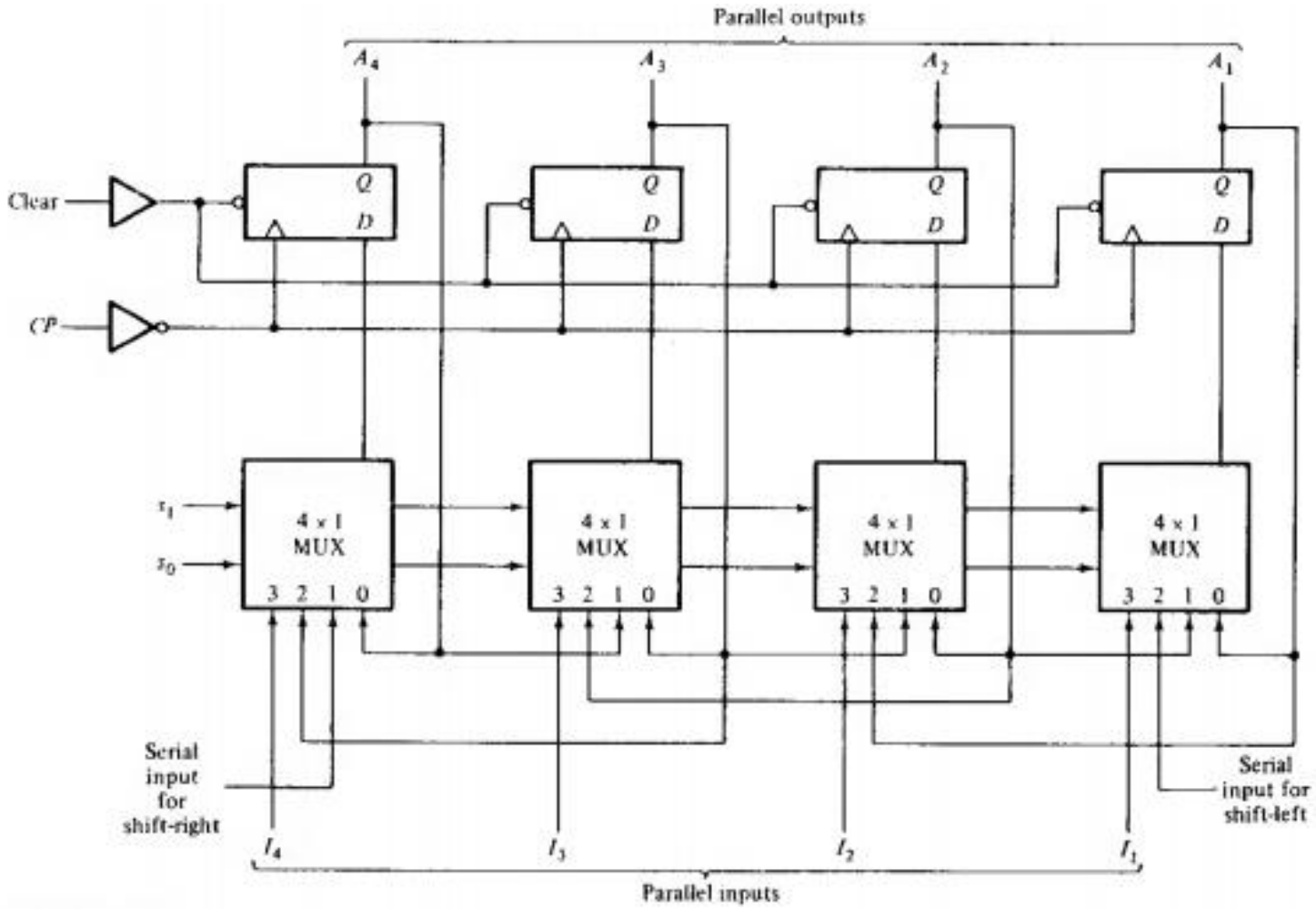
Function Table for the Register

Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

1001

Universal Shift Register: Parallel Load operation

1 0 0 1



1 0 0 1

CP	D	Q(t+1)
0	X	memory
1	0	0
1	1	1

Function Table for the Register

Mode Control		
S_1	S_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

Counters

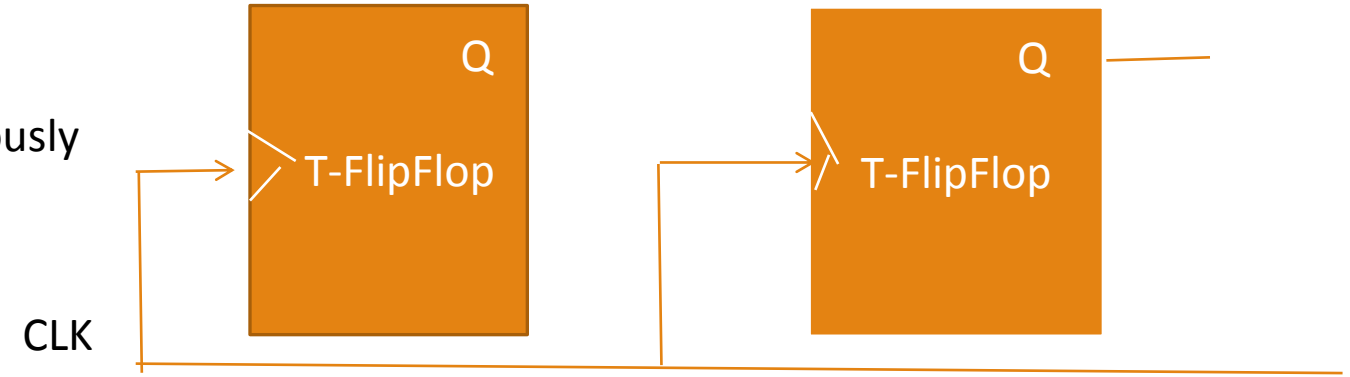
Counter

- **A sequential circuit that goes through a prescribed sequence of states upon the application of input pulses is called a counter .**
- The input pulses, called count pulses , may be clock pulses or they may originate from an external source and may occur at prescribed intervals of time or at random.
- In a counter, the sequence of states may follow a binary count or any other sequence of states.
- Counters are found in almost all equipment containing digital logic.
- **They are used for counting the number of occurrences of an event and are useful for generating timing sequences to control operations in a digital system.**

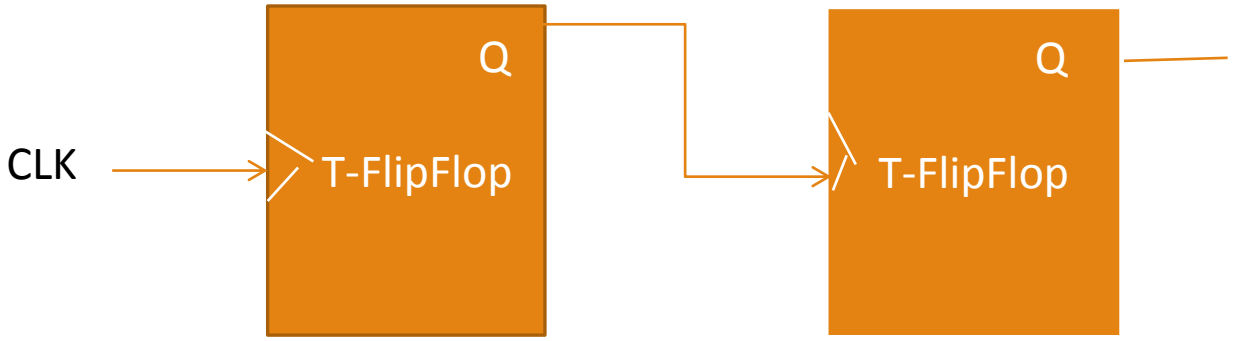
Types of Counters

Synchronous Counter:

Flipflops are clocked simultaneously



Asynchronous/Ripple Counter:



- Flipflops are not clocked simultaneously
- Output of one flipflop drives the clock of next flipflop

A counter that follows the binary sequence is called a **binary counter** .

An n-bit binary counter consists of n flip-flops and can count in binary from 0 to 2^n — 1.

Steps to design Synchronous counter

1. Decide the type and number of flipflop
2. Draw excitation table of flipflop.
3. Draw state diagram and circuit excitation table.
4. Obtain simplified expression using K-Map.
5. Draw the logic diagram.

Example: 3 bit binary Counter

The state diagram of a 3-bit counter is shown in Figure. 3 bit mean 3 flipflops are required.

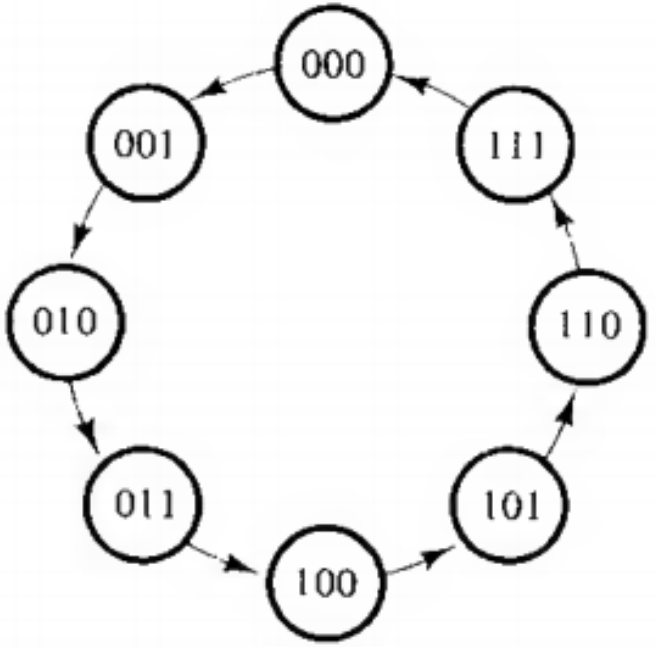
As seen from the binary states indicated inside the circles, the flip-flop outputs repeat the binary count sequence with a return to 000 after 111.

The directed lines between circles are not marked with input-output values as in other state diagrams.

Remember that state transitions in clocked sequential circuits occur during a clock pulse.

The flip-flops remain in their present states if no pulse occurs.

For this reason, the clock-pulse variable CP does not appear explicitly as an input variable in a state diagram or state table.



- The next state of a counter depends entirely on its present state, and the state transition occurs every time the pulse occurs.
- In the excitation table for the 3-bit binary counter, the three flip-flops are given variable designations A2, A1, and A0

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0
	T	

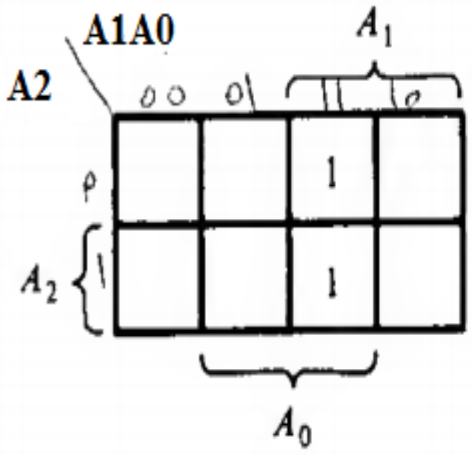
Excitation table for T Flipflop

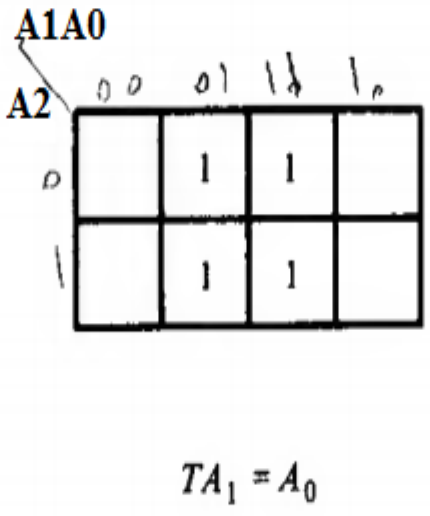
Excitation Table for 3-Bit Counter

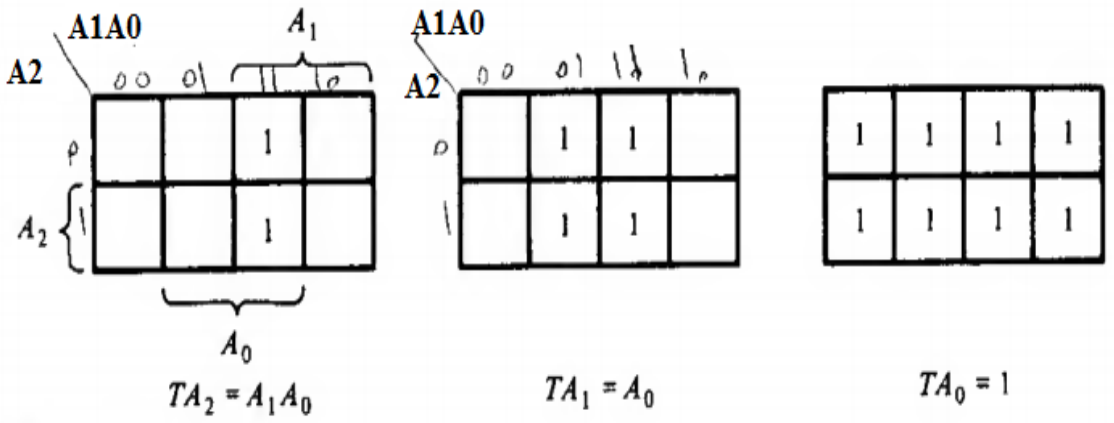
Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	TA_2	TA_1	TA_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

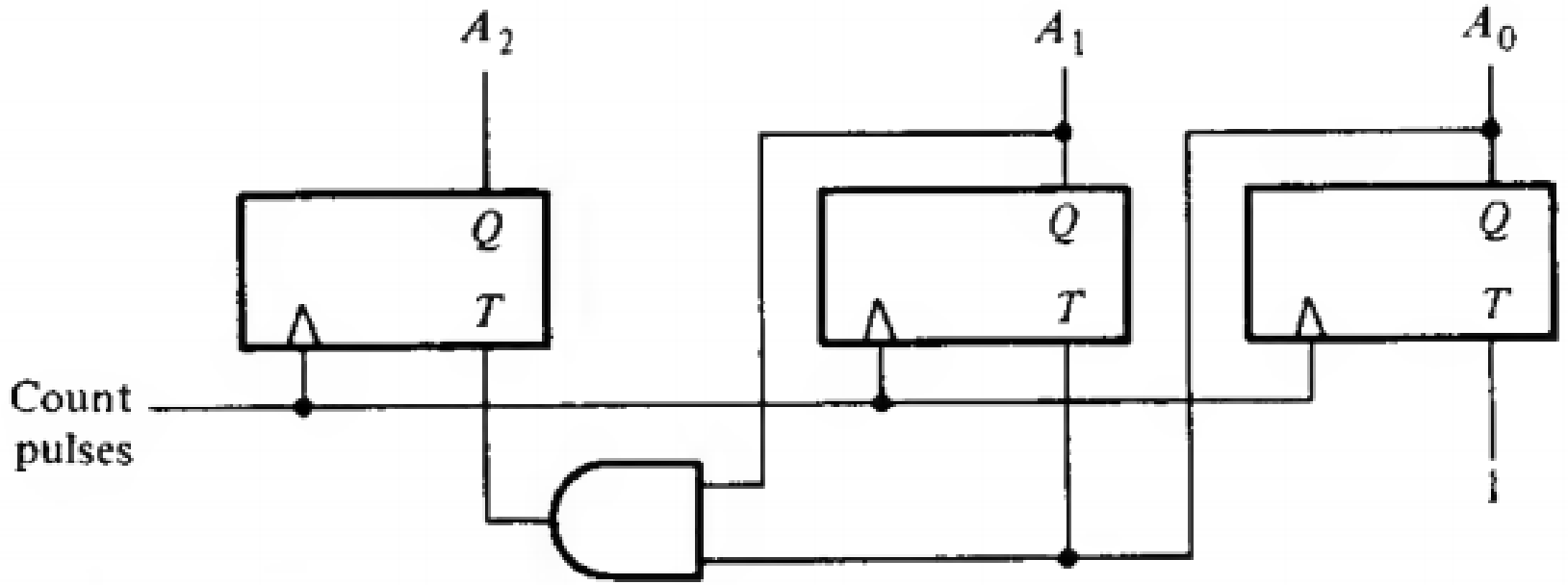
Excitation Table for 3-Bit Counter

Present State			Next State			Flip-Flop Inputs		
A_2	A_1	A_0	A_2	A_1	A_0	TA_2	TA_1	TA_0
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









Logic diagram for 3-bit Counter

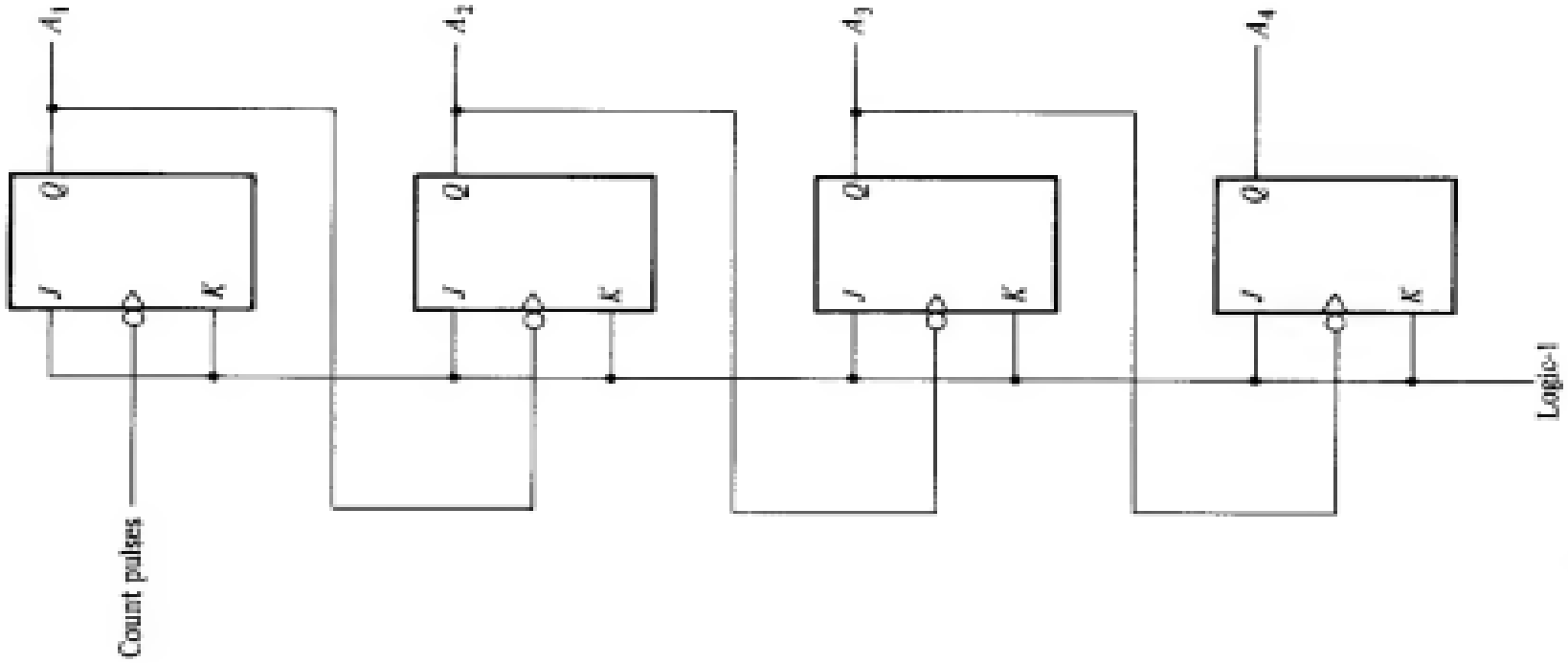
Asynchronous counter

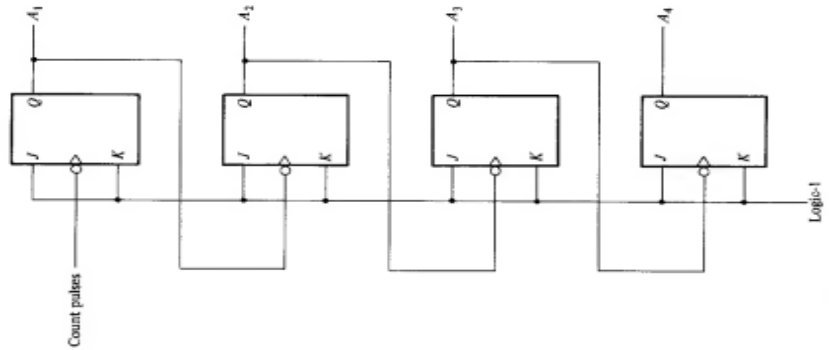
Asynchronous/Ripple Counter

- In a ripple counter, the flip-flop output transition serves as a source for triggering other flipflops.
- In other words, the CP inputs of all flip-flops (except the first) are triggered not by the incoming pulses, but rather by the transition that occurs in other flip-flops.

- A binary counter with a reverse count is called a binary down- counter .
- In a down counter, the binary count is decremented by 1 with every input count pulse.
- The count of a 4-bit down-counter starts from binary 15 and continues to binary counts 14, 13, 12, . . . , 0 and then back to 15.

4 bit Ripple UP Counter




A_1
A_2
A_3
A_4
Q
J
K
Count pulses
Logic-1

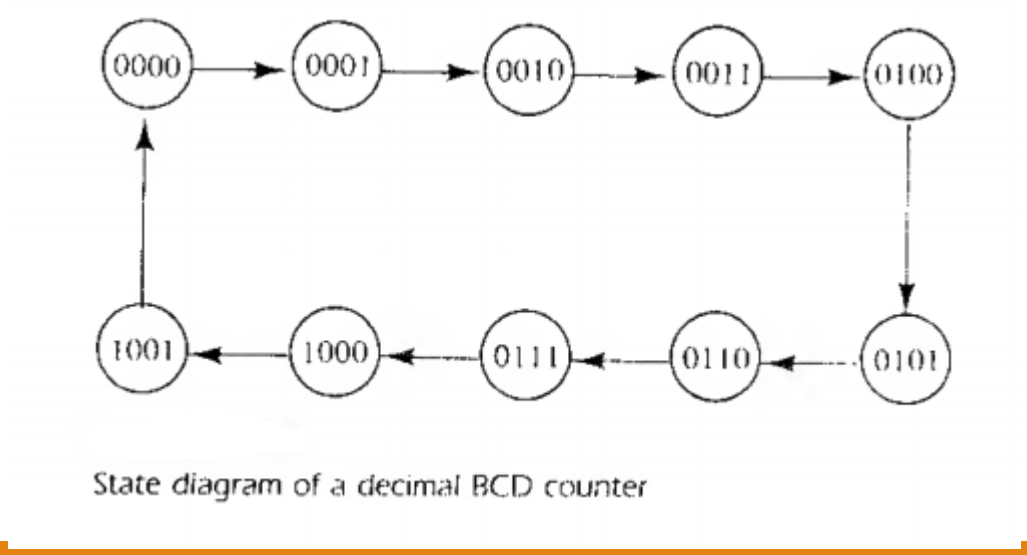

CP

CP	A4	A3	A2	A1
NO CP	0	0	0	0
1st	0	0	0	1
2nd	0	0	1	0
3rd	0	0	1	1
4th	0	1	0	0
5th	0	1	0	1
6th	0	1	1	0
7th	0	1	1	1
8th	1	0	0	0
9th	1	0	0	1
10th	1	0	1	0
11th	1	0	1	1
12th	1	1	0	0
13th	1	1	0	1
14th	1	1	1	0
15th	1	1	1	1
16th	0	0	0	0

Design a 3 bit Ripple down Counter

Design a BCD Ripple Counter

- A decimal counter follows a sequence of ten states and returns to 0 after the count of 9.
- Such a counter must have at least four flip-flops to represent each decimal digit, since a decimal digit is represented by a binary code with at least four bits.
- The sequence of states in a decimal counter is dictated by the binary code used to represent a decimal digit.



State diagram of a decimal BCD counter

Suggested Reading

- M. Morris Mano, Digital Logic and Computer Design, PHI.

Thank you