UNIT 4 LOGIC CIRCUITS – SEQUENTIAL CIRCUITS

Structure **Page Nos.**

4.0 INTRODUCTION

The first Unit of this Block explained the basic structure and process of instruction execution. Unit 2 provided a detailed description of data representation and Unit 3 presented the concepts of basic functional unit of a computer, viz. the logic gates and combinational circuits. In this unit, you will be introduced to one of the most fundamental circuit that can store one bit of data called flip flops. The unit also explains how flip-flops and additional logic circuit can be used to make registers, counters, sequential circuits etc. Finally, the Unit also introduces you to simple design of a sequential circuit.

4.1 OBJECTIVES

After going through this unit you will be able to:

- explain the functioning of flip-flops;
- determine the behaviour of various latches;
- construct excitation table of a flip-flop;
- explain circuits of a computer system like registers, counters etc.

4.2 SEQUENTIAL CIRCUITS: THE DEFINITION

A sequential circuit is an interconnection of combinational circuits and storage elements. The storage elements, called flip-flops, store binary information that indicates the state of sequential circuit at that point of time.

Figure 4.1 highlights that a sequential circuit may involve combinational circuits (which were discussed in Unit 3) the flip-flops (which are discussed in this unit) and a system clock, which is a useful timing device of a computer system.

Figure 4.1: Block Diagram of sequential circuits.
(Ref: M. Morris Mano, Charles R. Kime: Logic and compute design fundamental, 2nd Edition, Pearson Education)

The sequential circuits are time dependent. The present state of a combinational circuit is identified by the present output of flip-flop. This output may change over a passage of time and can also be used as one of the input. This change in state can occur either in synchronous or asynchronous manner with respect to system clock.

Synchronous circuits use flip-flops and their state can change only at discrete intervals. Asynchronous sequential circuits are regarded as combinational circuit with feedback path. Such circuits may unstable at times, when the propagation delays of output to input are small. Thus, complex asynchronous circuits are difficult to design.

Clock Pulse and sequential circuits

A sequential circuit uses clock pulse generator, which gives continuous clock pulse to synchronize change in the state of the circuit. Figure 4.2 shows the form of a clock pulse.

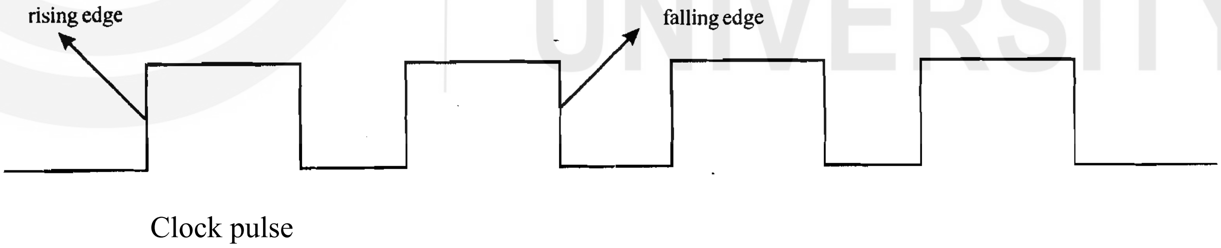


Figure 4.2: Clock signals of clock pulse generator

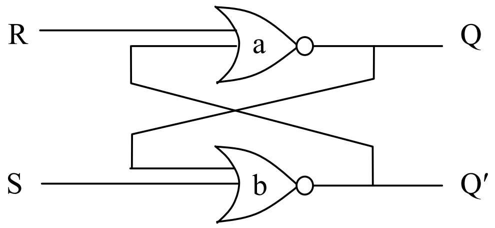
A clock pulse can have two states, viz. 0 or 1, which are also called disabled or active state. Flip-flops are allowed to change their states, in general, with the rising or falling edge of the clock pulse, so as to make stable changes in states of the flip-flops.

4.3 LATCHES AND FLIP-FLOPS

A ***latch*** is a storage element with basic logic circuit, which can store 1-bit of data. It itself is a sequential circuit. Flip-flops are constructed using latches and have more complex timing sequences than latches. Therefore, in order to learn flip-flops, learning the basic concept of latches in very useful, which is discussed next.

4.3.1 Latches

A basic latch can be constructed using either two NOR or two NAND gates. Figure 4.3 (a) shows logic diagram for S-R latch using NOR gates. This latch has two inputs viz. S and R for *S*et and *R*eset respectively; and one output Q. Please note Q′ output is complement of the output Q. This flip flop exhibits two states called SET state (when the flip-flop output Q is1, that is Q′=0) and RESET state or clear state (Q=0; Q′=1).



S	R	Q	Q′	Comment
0	0	0/1	0/1	No Change in State
0	1	0	1	Reset State
1	0	1	0	Set State
1	1	-		Undefined Input

(a) Logic Diagram **(b) Truth Table**

Figure 4.3: SR Latch using NOR gates

The following table shows the truth table for NOR gates using in the S-R latch of Figure 4.3 (a).

The Truth tables for NOR gates of Figure 4.3							
NOR gate Marked 'a'				***NOR gate Marked 'b'***			
	Input		Output		Input		Output
	R	**Q′**	**Q**		S	Q	Q′
0	0	0	1	0	0	0	1
1	0	1	0	1	0	1	0
2	1	0	0	2	1	0	0
3	1	1	0	3	1	1	0

Let us examine the latch in more details. Assume that initially latch is in clear state, i.e. Q=0 and Q′=1; also assume that both S and R input are 0. The states of the latch will be as follows (refer to the NOR gate truth table given above):

Gate 'a'
Input R Q′ :: 0 1 ⇒ Output (Q) 0
Gate 'b' Output of latch stays in CLEAR state
Input S Q :: 0 0 ⇒ Output (Q′) 1

(i) *Setting the latch:*

Now assume that S is changed to 1and R remains 0 during this time, then the output of Gate 'b' will change first:

S Q :: 1 0 ⇒ Q′ will become 0
Gate 'a' now has the following input:
R Q′ :: 0 0 ⇒ Q will be set to 1.
Gate 'b' now has the following input SET state
S Q :: 1 1 ⇒ Q′ will stay at 0.

Thus, Flip-flop will be in SET state.
Finally, after some time S will become 0;
At that time, gate 'a'
R Q′ :: 0 0 Q stays at 1
Gate 'b' Latch will stay in SET State
S Q :: 0 1 Q′ stay at 0

(ii) *Reset the latch:*

Now assume that input S remains at 0 and input R is changed to 1, also assume that at this time the latch is in Set state (Q = 1 & Q′ = 0), then the output of Gate ‘a’ will change as

Gate ‘a’

R Q′:: 1 0 ⇒ Q will become 0.

Gate ‘b’ Latch is in Reset state.

S Q :: 0 0 ⇒ Q′ will become 1

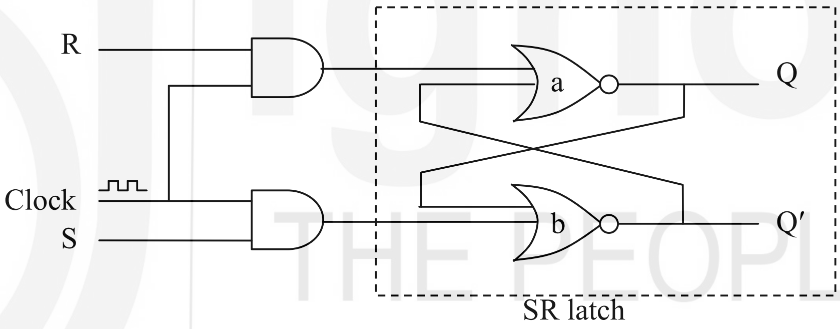
Once again, when S and R both input will become 0, latch will remain in RESET state.

iii) *When both S and R become 1 simultaneously, then?*

A basic S-R latch, in general, changes state at any time, which may result in asynchronous changes in Q output, which can make system unstable. Therefore, latches are constructed with controlled input using clock. This is explained next.

SR latch with Clock

The following diagram shows an SR latch which changes its data only with the occurrence of a clock pulse.



(a) Logic Diagram

Clock(c)	S	R	Present State Q_t before the clock pulse	Next State/Q_{t+1} after occurrence of clock pulse.	Comments
0	Any	Any	0/1	0/1	No change in state
1	0	0	0/1	0/1	No change in state
1	0	1	0/1	0	Reset the latch
1	1	0	0/1	1	Set the flip-flop
1	1	1	0/1	-	Not defined.

(b) Characteristic Table

Figure 4.4: R-S latch with clock.

Operations on this clocked SR latch are given below:

1) If no clock signal i.e. clock=0 ⇒ No change in state of latch.

2) Presence of clock signal

(i) if S=0 and R=0, No change in state/output stays same as earlier state.

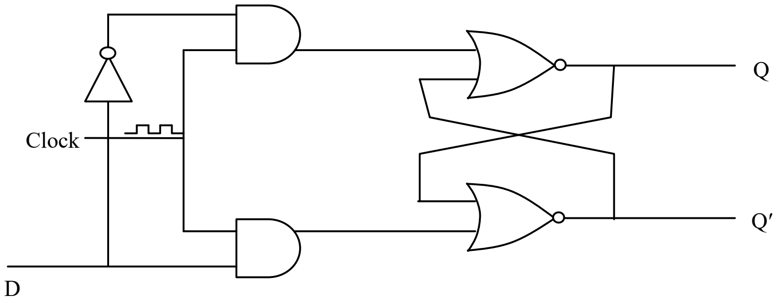
(ii) if S=1, R=0, then next state is the SET state Q=1 & Q′ =0

(iii) if R=1 S=0, then next state is the RESET state Q=0 & Q′=1

(iv) if both S and R become 1, then next state/output is not defined.

D Latch

The D (data) latch is modification of RS latch. D latch only uses one input named D, it stores the value of D in the latch, e.g. if the D input is 1, then the next state of latch will also be 1. Figure 4.4 shows the clocked D latch.



D	Q_t	Q_{t+1}	When clock pulse occurs
0	0/1	0	Clear the latch
1	0/1	1	Set the latch

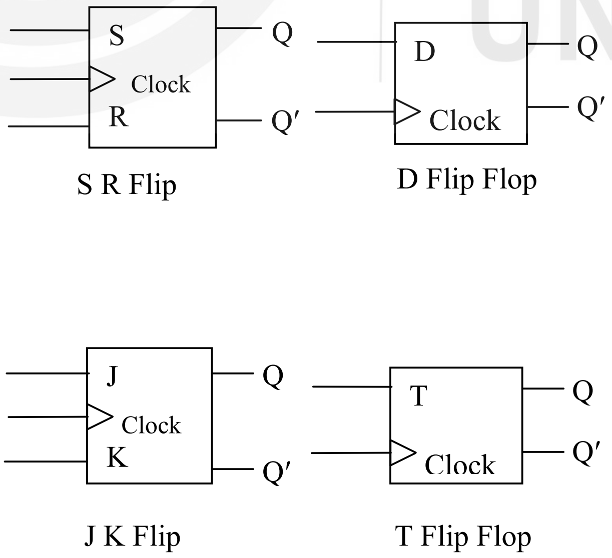
(a) Logic Diagram I **(b) Characteristic Table**

Figure 4.5: D latch with clock

You may please go through the circuit and identify various changes in Q, Q′ with D as shown for SR latch.

4.3.2 Flip-Flops

Latches suffer from the problem due to frequent changes of output, e.g. the output of latch may change depending on the value of R and S input, which may change from 1 to 0 or vice-versa during a single clock pulse. Therefore, they are less suitable for sequential circuits. Flip-flops add more circuitry in latches so that changes in states occur during the rising or falling edge of clock pulse (these are called edge triggered flip-flop). R-S latch with clock can be used with additional circuits to make R-S flip-flop. The flip-flops can also be represented using a block diagram. Figure 4.6 shows the block diagram of basic flip-flops. Please note that in the block diagram the arrow head in front of the clock signal represents that the flip-flop will respond to input during the leading or rising edge (when transition from 0 to 1 takes place) of the clock



S R Flip

D Flip Flop

J K Flip

T Flip Flop

(Edge trigger is shown by arrow in clock)

Figure 4.6: Graphical Symbols of basic Flip-Flops

JK flip is almost identical to SR flip-flop, except the last combination of J = 1 and K = 1 is used to complement the current state of the flip-flop. T-flip-flop is obtained by joining the J and K input, thus, it shows just two input values. When T = 0, there is no change of state and at T = 1, the current state is complemented. The following figure shows the characteristics table for the basic flip-flops shown in Figure 4.7

SR Flip-flop			
S	R	Q_{t+1}	Comments
0	0	Q_t	No Change in state
0	1	0	Clear state
1	0	1	Set state
1	1	-	Not Defined

JK Flip-Flop			
J	K	Q_{t+1}	Comments
0	0	Q_t	No Change in state
0	1	0	Clear state
1	0	1	Set state
1	1	Q'_t	Complement of Q_t

D Flip-flop		
D	Q_{t+1}	Comments
0	0	Clear State
1	1	Set State

T Flip-flop		
T	Q_{t+1}	Comments
0	Q_t	No Change in state
1	Q'_t	Complement of Q_t

Figure 4.7: Characteristic Table for flip-flops

4.3.3 Excitation Tables

The characteristic tables of flip-flops as shown in Figure 4.7 show how the state of flip-flop will change to the next state based on the present state and input values. The characteristic tables are used for analysis of the sequential circuits. While designing sequential circuits, you need to consider for what transition what possible input combinations would be required. This is done with the help of excitation table. Figure 4.8 shows excitation tables for the basic flip-flops.

Q_t	**Q_{t+1}**	**J**	**K**
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(a) JK Flip flop

Q_t	**Q_{t+1}**	**S**	**R**
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(b) SR Flip flop

Q_t	**Q_{t+1}**	**D**
0	0	0
0	1	1
1	0	0
1	1	1

(c) D Flip flop

Q_t	**Q_{t+1}**	**T**
0	0	0
0	1	1
1	0	1
1	1	0

(d) T Flip flop

X-denotes *DONOT CARE* condition.

Figure 4.8: Excitation Tables for basic flip-flops

Q_t and Q_{t+1} indicate present and next state of a flip flop, respectively. Symbol X in the table means do not care condition i.e. it does not matter whether the input is 0 or 1.

How these excitation tables are created? This is explained with an example of creation of excitation table of JK Flip flop.

a) The state transition from $Q_t = 0$ to $Q_{t+1} = 0$

(i) As both Q_t and Q_{t+1} are 0 it means that there is no change in the state of flip flop, which can be achieved by J=0, K=0;

(ii) Using the input, J=0, K=1, the flip flop can be RESET, i.e. $Q_{t+1} = 0$.

b) The state transition from $Q_t = 0$ to $Q_{t+1} = 1$

(a) Using the input, J=1, K=0, the flip flop is SET, i.e. $Q_{t+1 =}$ 1

(b) Using the input, J=1, K=1, the flip flop is complemented from Q_t having a value 0 to $Q_{t+1} = 1$

c) State transition from $Q_t = 1$ to $Q_{t+1} = 0$

(a) Using the input, J=0, K=1, flip flop is RESET, i.e. $Q_{t+1} = 0$

(b) Using the input, J=1, K=1, the flip flop is complemented from Q_t having a value 1 to $Q_{t+1} = 0$

d) For state transition from $Q_t = 1$ to $Q_{t+1} = 1$

(a) Using the input, J=0, K=0, no change in flip flop so $Q_{t+1} = 1$

(b) Using the input, J=1, K=0, flip flop is SET, i.e. $Q_{t+1} = 1$

These entire set of input for various transitions can be summarized in the table below:

Present State (Q_t)	**Next State** (Q_{t+1})	**Input J and K**	**Input using DONOT CARE**
0	0	(i) J=0, K=0 (ii) J=0, K=1	**J=0, K=X**
0	1	(i) J=1, K=0 (ii) J=1, K=1	**J=1, K=X**
1	0	(i) J=0, K=1 (ii) J=1, K=1	**J=X, K=1**
1	1	(i) J=0, K=0 (ii) J=1, K=0	**J=X, K=0**

The excitation table has been derived for J-K flip-flop as above. You may draw the excitation table for all other flip-flops using the same method.

Check Your Progress 1

1. What is a sequential circuit? How are sequential circuits different from combinational circuits?

....

………………………………………………………………………………………

………………………………………………………………………………………

2. What is a latch? How is different from a flip-flop?

....

………………………………………………………………………………………

………………………………………………………………………………………

3. What is an excitation table? Draw the excitation table for SR, D and T flip-flops.

....

………………………………………………………………………………………

4.3.4 Master-Slave Flip-Flop

The master slave flip-flop is constructed using two or more latches. Figure 4.9 shows how two S-R flip-flops can be used to construct a master-slave flip-flop.

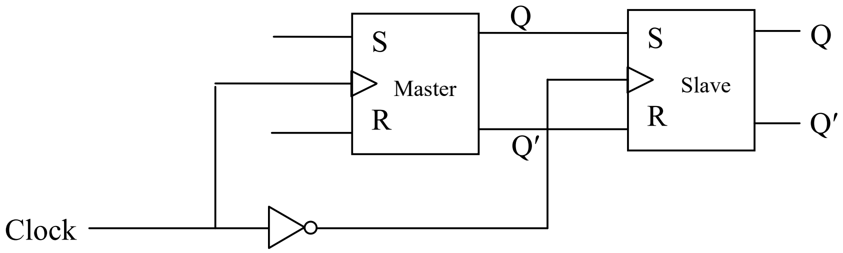


Figure 4.9: Master – Slave flip- flop

You may please note that you can construct a master-slave flip-flop using D or JK flip-flop also. This flip-flop consists of master which changes state when clock pulse occurs. The slave flip flop goes to the state of master flip-flop when the clock signal is 0. (Refer to figure 4.9) This is explained below:

The flip-flop operates is two steps:

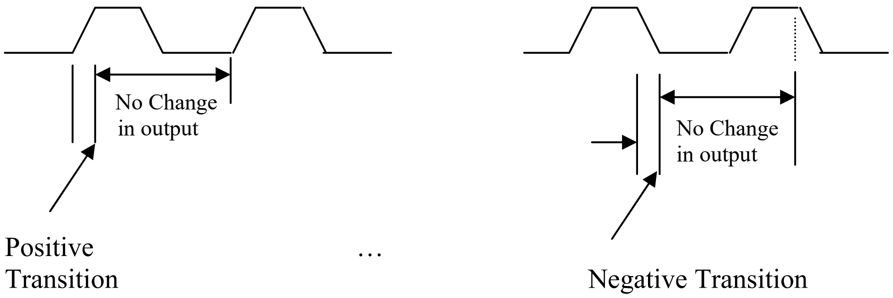
(i) When a clock pulse input is 1: As this time the Master flip-flop, based on the value of S and R, goes to Set or Clear state as the case may be. At this time the slave flip-flop cannot change its state as it receives the inverse of clock pulse. Thus, on the occurrence of clock pulse 'Master' flip-flop goes to the next state (Q_{t+1}), whereas the output from slave flip-flop is the present state (Q_t).

(ii) When the clock pulse input is 0: In this time the input to Master flip-flop will not have any effect on the Master flip-flop output, which has been put in the next state (Q_{t+1}) in the previous step. However, now this Q_{t+1} output of master flip-flop will be applied on the slave flip, which will result in transition of state of slave flip flop to Q_{t+1}. Thus, on completion of a clock cycle master and slave flip-flops both will be in Q_{t+1}. Please note that for slave flip flop only following transitions are possible:

Master output ≡ Slave input				Slave Output		
Q	Q′	S	R	Q	Q′	
1	0	1	0	1	0	(Set)
0	1	0	1	0	1	(Reset)

4.3.5 Edge-Triggered flip-flops

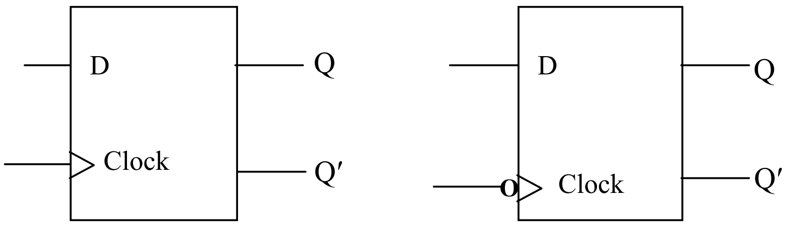
An edge-triggered flip-flop triggers the change either during the rising edge or positive transition (0 to 1 transition) or the falling edge or negative transition of the clock (1to 0 transition). Fig 4.10 shows the clock pulse signal in positive & negative edge-triggered flip-flops.



(a) Positive edge-triggering **(b) Negative edge triggering**

Figure 4.10: Clock Pulse Signal

The following figure shows the block diagram of edge triggered D flip-flop.



(a) Positive edge-triggered D flip-flop **(b) Negative edge–triggered D flip-flop**

Figure 4.11: Edge triggered and master slave D flip-flop

More detailed discussion on these flip-flops are beyond the scope of this unit. You may refer to further readings for the same.

Check Your Progress 2

1. List the advantages of master- slave flip-flop.

   ....

   ....

2. How edge- triggered flip-flops are different to master-slave flip-flops?

   ....

   ....

4.4 SEQUENTIAL CIRCUIT DESIGN

A sequential circuit not only consists of external input and external output, but also an internal state which is characterised by the state of flip flops internal to the circuit. The state of sequential circuit changes as per its design based on some control signal like the clock control. Therefore, design of a sequential circuit is required to address the changes in the internal state of itself. Therefore, in addition to the logic circuit, sequential circuit design requires the information about the changes in state of flip-flops. The process of design of a sequential circuit is explained with the help of an example of design of a 2-bit counter circuit given below.

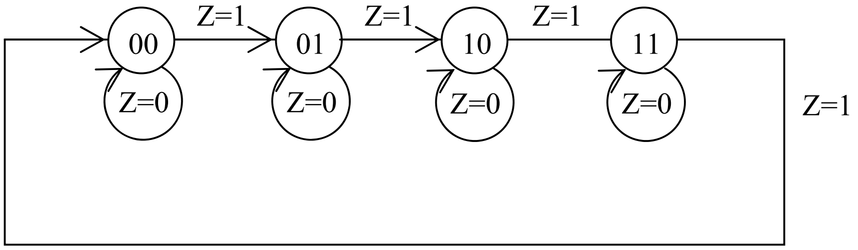
Example: Design a 2-bit counter circuit.

Solution: A counter is a special circuit which counts the timing sequences. A 2-bit counter will require two flip-flops. The state sequence of 2-bit counter would be 00, 01, 10, 11, 00 and so on. Thus, using a 2-bit counter, you can have 4 distinct internal states of the circuit and counter should move in each transition from one state to next as:

$$00 \longrightarrow 01 \longrightarrow 10 \longrightarrow 11 \longrightarrow 00$$

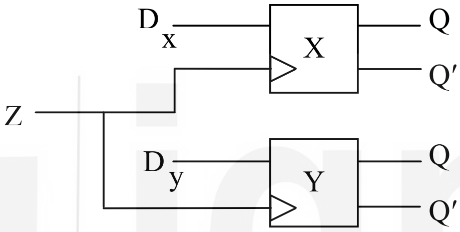
Assuming that these transitions are triggered by a control signal, say *Z* which can be a clock signal or any other signal generated for this purpose, a state change sequence can be presented as shown in the diagram given below:

State



This circuit uses two bits to store the state, therefore, requires two flip-flops. The state of the circuit changes to next state, when Z=1, else it stays in the same state. Thus, in this sequential circuit, you require 2 flip-flops and one control signal Z. But, what would be other input and output to this sequential circuit. Well! The other input will be the current states of flip-flops which will govern the next states of flip-flops.

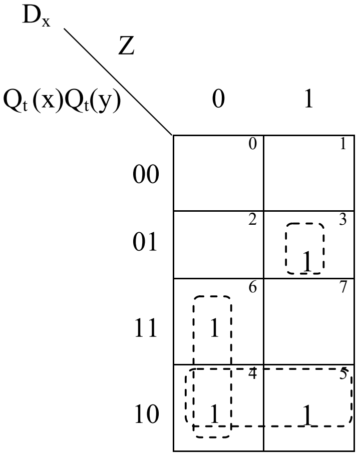
Next, you may take D flip-flop to design the circuit then a Rough design of the circuit would be:

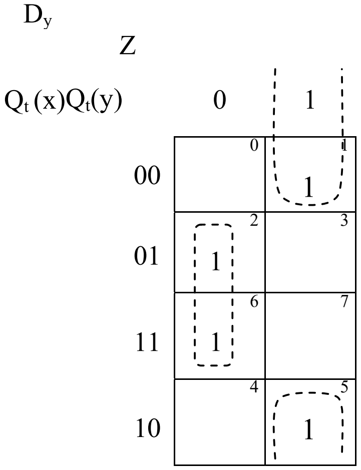


In order to design the logic circuit, which generates the signal D_x and $D_{y,}$ let us first draw a truth table for flip-flop's X and Y. This truth table is shown in the following table:

	Present States of Flip-Flops			Next State of Flip-Flops		Required value of D_x for transition of X and D_y for the transition of Y	
	Flip-flops		Input	Flip-flops			
	Q_t of X	Q_t of Y	Z	Q_{t+1} of X	Q_{t+1} of Y	D_x	D_y
0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1
2	0	1	0	0	1	0	1
3	0	1	1	1	0	1	0
4	1	0	0	1	0	1	0
5	1	0	1	1	1	1	1
6	1	1	0	1	1	1	1
7	1	1	1	0	0	0	0

Interestingly, it is the D_x and D_y input that should be generated from the present state and Z input, so that the Next state (Q_{t+1}) of the flip-flops can be derived from the present state of the flip-flop (Q_t). Thus, for the design of counter circuit, you can draw K-map for the design of D_x and D_y with input Q_t (X), Q_t (Y) and Z. TheK-maps for D_x and D_y can be drawn as:





D_x = Terms of (Adjacency of 4,5 + Adjacency of 4,6 + Cell 3)

$D_x = Q_t(X) \,.\, Q_t'(Y) + Q_t(X) \,.\, Z' + Q_t'(X) \,.\, Q_t(Y) \,.\, Z$

D_y = Terms of (Adjacency of 2,6 + Adjacency of 1,5)

$D_y = Q_t(Y) \,.\, Z' + Q'(Y) \,.\, Z$

Thus, the final 2-bit counter circuit will be drawn as shown in Figure 4.12

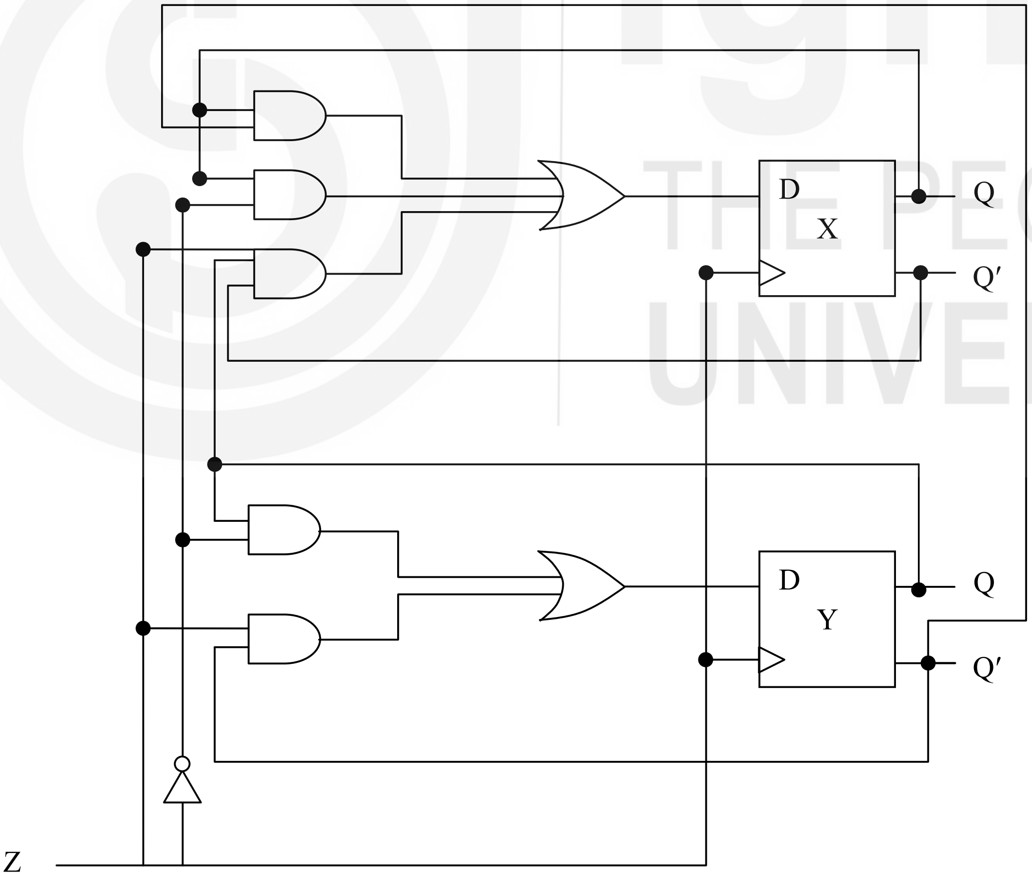


Figure 4.12: 2-bit counter

4.5 EXAMPLES OF SEQUENTIAL CIRCUITS

Let us now explain the basic function of some of the useful examples of sequential circuits like registers, counters etc.

4.5.1 Registers

Registers are the basic storage unit of a computer. Since register temporarily stores certain values, therefore, it requires flip-flops. The size of registers is computed using number of bits it stores. One bit storage requires, at least, one flip-flop. Thus, in general, an n bit register would use n flip-flops. Two common operations on register are:

- To load all bits of a register simultaneously or parallel load.
- Shifting of bits, of register, towards left or right

Figure 4.13 shows a parallel load register..

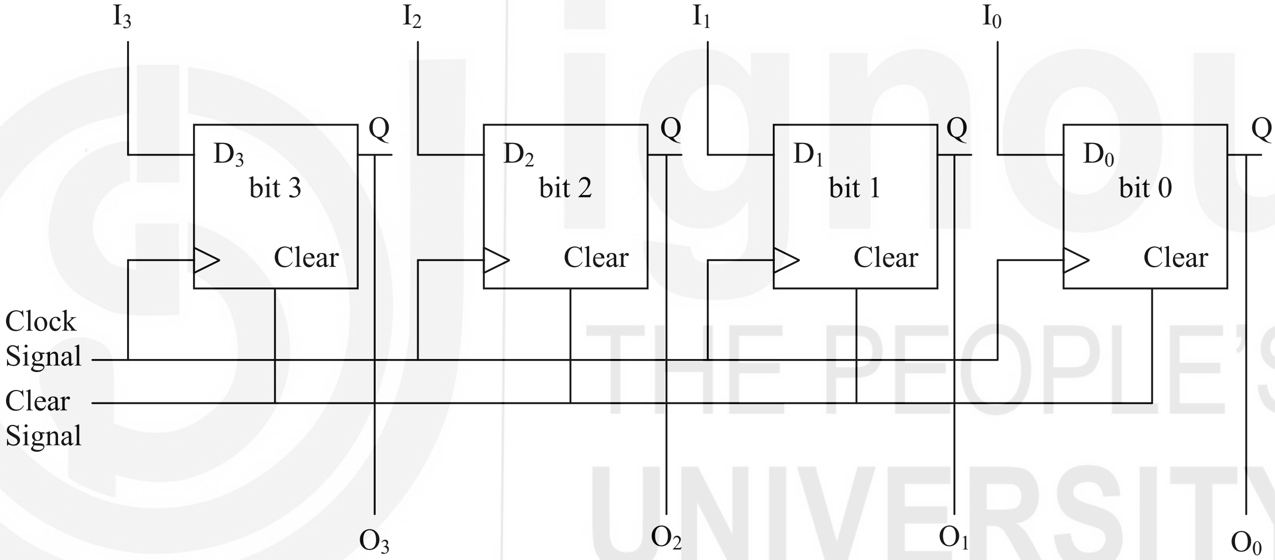


Figure: 4.13 A register with parallel load.

Please note the following point about the register circuit as above:

(1) The 4-bit register is made up of 4 D flip-flops.

(2) Clock signal is applied to all flip-flops simultaneously; therefore, loading operation will load the values I_3, I_2, I_1, and I_0 respectively into the four flip-flops, simultaneously.

(3) Special clear signal is used, which can clear all the bits of the register simultaneously, if needed.

(4) The output of register O_3, O_2, O_1, O_0 can be used for any arithmetic operations. Please note that registers output changes synchronously.

Shift register: Shift operation is very special operation for a computer ALU. A shift register is capable of shifting the content of a register either to left or to the right by one bit at a time. The following figure shows a right shift register, however, you can construct a left shift register in a similar manner.

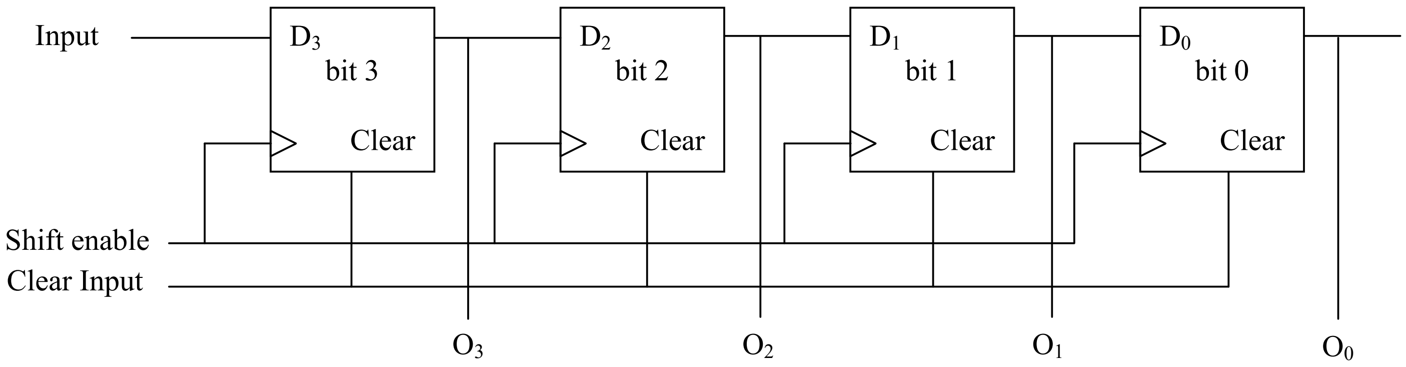


Figure 4.14: 4-bit Right Shift Register

Please note the following points.

- The external input is applied to D_3. The output of D_3 is applied to D_2, and so on.
- The shift enable is applied as clock input. It enables the shift operation.
- For example, assume the shift register had state 1 0 0 1 and input bit is 1, then after the right shift operation the output will change as:

	I	O_3	O_2	O_1	O_0
Before Shift	1	1	0	0	1
After Shift	1	1	1	0	0

A single registers can be included with the facility of left shift, right shift and parallel load. Such a register is called bi-directional shift register with parallel load. You may create its block diagram as an exercise.

4.5.2 Counters Circuit

Counters are sequential circuits, which produce output in a sequence on the occurrence of a transition signal. The counters may be used in keeping sequence such as steps of execution of a single instruction. There are two types of counters- asynchronous and synchronous.

In synchronous counter the flip-flop change their state one by one, while in synchronous counter all flip-flops may change the state simultaneously

Asynchronous Counter: An asynchronous counter is also called a ripple counter as the changes in the state of flip-flop is done one by one like a wave. Figure 4.17 shows a 3-bit ripple counter using T-flip flop (Please note the earlier 2-bit counter was designed using D flip-flop). These counters also have all clear input but for simplicity it is omitted in the figure.

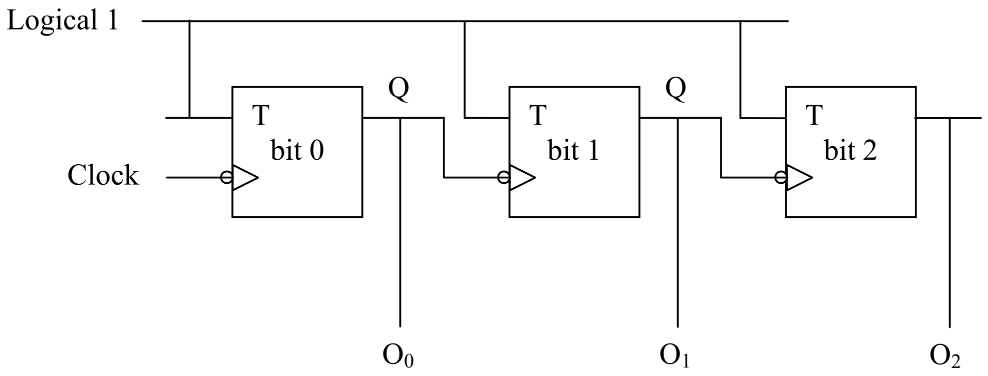


Figure: 4.15: 3-bit ripple counter

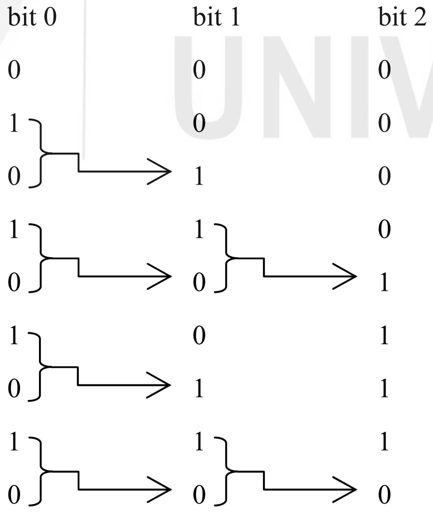
Please note the following points in the figure.

(i) The bit 0 will complemented each time a clock pulse occurs as it is connected to clock.

(ii) For bit 1 flip flop the transition will be triggered, if Q_t of bit 0 flip-flop is 1. In that case Q_{t+1} of bit 1 will be complemented. This occurs because the transition signal of (bit 1) flip-flop is connected to Q_t output of bit 0 flip-flop. Similarly, the transition of bit 2 flip flop will occur, if Q_t of bit 1 flip-flop was in state 1.

(iii) The transition is expected to occur with the falling edge (indicated by o before the clock input).

(iv) Please note change in states would be as follows. Assuming initial state to be 0 0 0



bit 0	bit 1	bit 2
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1
1	0	1
0	1	1
1	1	1
0	0	0

} indicates the falling edge.

4.5.3 Synchronous Counter:

The flip-flops of the synchronous counter can change their state simultaneously. A 3-bit synchronous counter with rising edge of clock signal is shown in figure 4.16

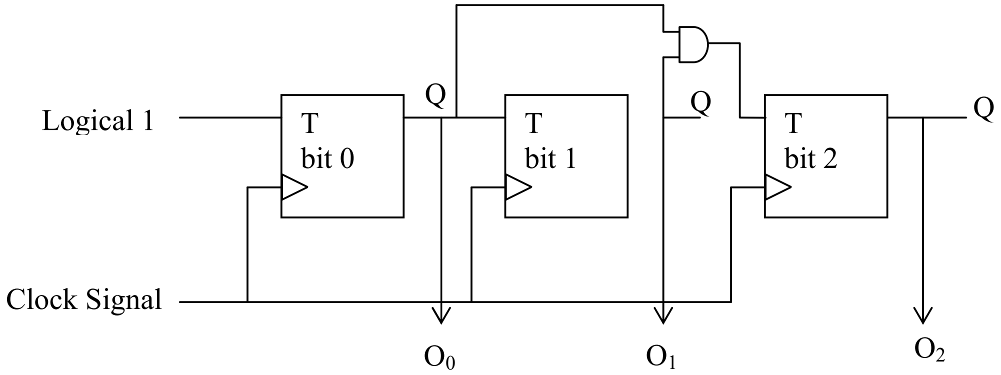


Figure 4.16: Synchronous Counter

Please note in the figure above

(i) bit 0 is complement on occurrence of clock pulse

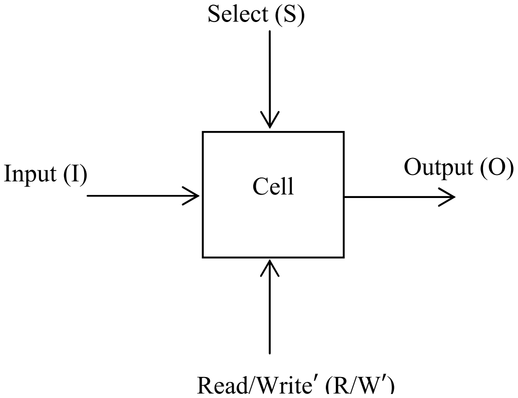
(ii) bit 1 is complemented, if (Q_t of bit 0) was 1.

(iii) bit 2 is complemented, if Q_t of bit 0 and Q_t of bit 1, both are 1.

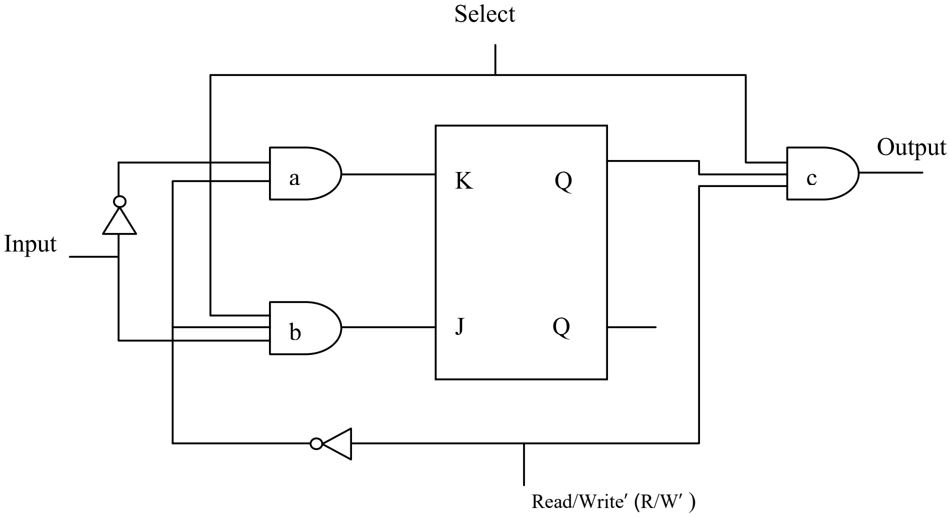
Flip-Flop		
bit 0	bit 1	bit 2
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1
1	0	1
0	1	1
1	1	1
0	0	0

4.5.4 Random Access Memory

In this section, a general configuration for flip-flop based random access memory (RAM) is proposed. A RAM essentially stores bits, therefore, it (especially DRAM technology) may be a sequential circuit. Two basic operations are performed on RAM: *Reading information from RAM,* this operation requires decoding operation, which identifies the cells or lines that are to be read; and *writing to RAM*, which in addition to identifying the cell, also requires changing the state of selected RAM flip-flops based on the input value. The figure 4.21 shows the block diagram and logic diagram and of a RAM cell, which is a single flip-flop.



(a) Block Diagram



(b) Logic Diagram

Figure 4.17: Binary Cell

A RAM cell as shown consists of one flip-flop. The behavior of this cell is exhibited in the following table.

(i) Read/Write′ bit 1 ⇒ Operation is Read

Select bit	Read bit	Q_t	Output of cell (c flip-flop)
0	1	0/1	Not activated
1	1	0/1	Q_t

(ii) Read/Write′ bit 0 ⇒ Write Operation

Assume Input bit to the circuit in Figure 4.17 (b) is I

Select Bit (S)	**Input Bit (I)**	**Q_t of Gate**		**Flip-flop Input**		**Q_{t+1}**	**Comments**
		'a'	**'b'**	**J**	**K**		
0	-	Any	Any	0	0	Q_t	Not selected
1	0	0	1	0	1	0	Clear memory flip-flop
1	1	1	0	1	0	1	Set the memory cell

The write operation as shown in the table above changes the content of memory cell to the value of Input (I), or in other words memory cell has been written into by the value of input (I).

In addition to read/write′ to memory cell, additionally a RAM is to be organized as an array of RAM cells, so as to decode the address of the cells, which is shown in Figure 4.18.

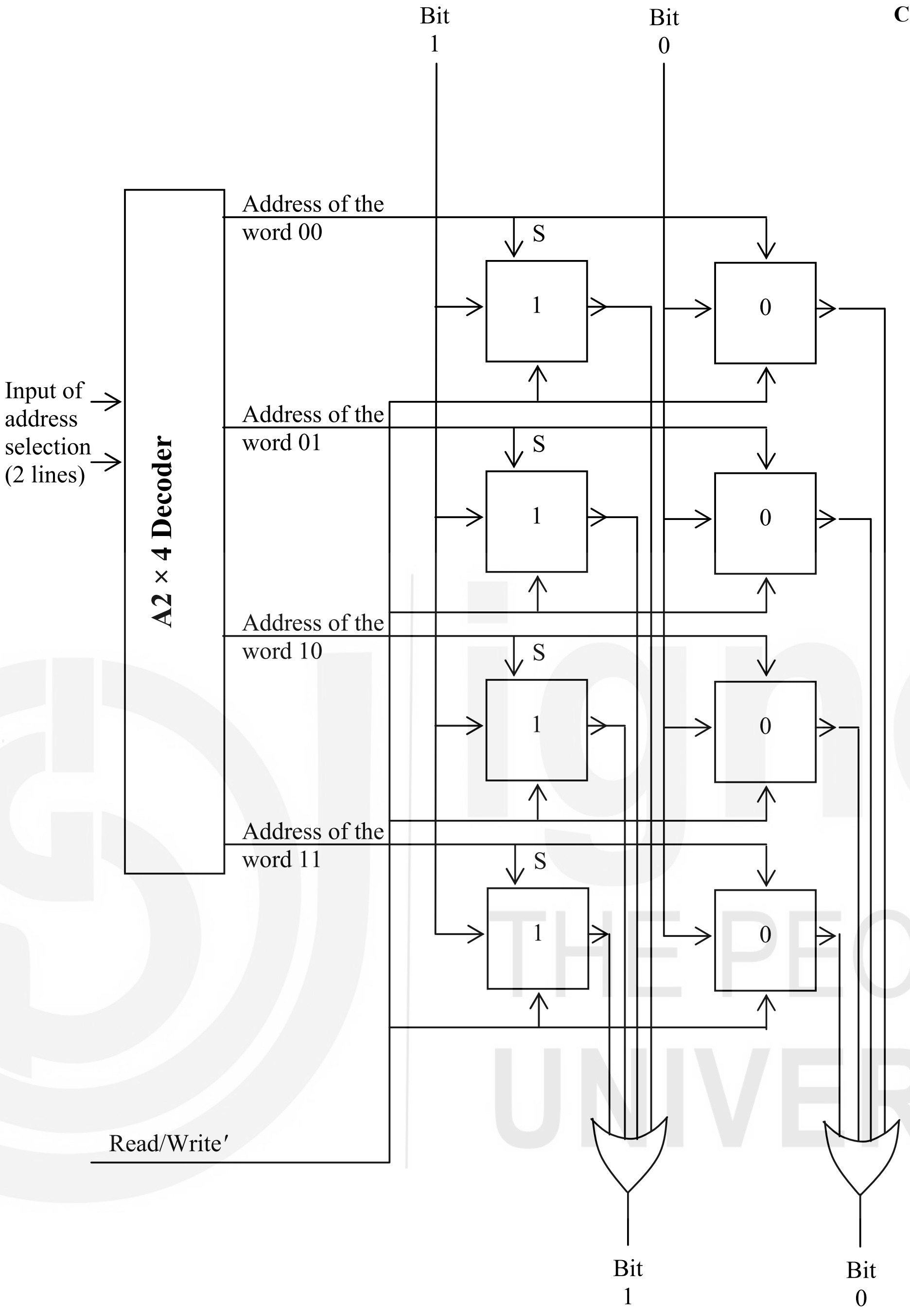


Figure 4.18: Two-dimensional Array based 4 × 2 RAM

The RAM has 4 words, which are decoded by the address decoder. Please note as there are 4 words or lines, therefore, you require 2×4 decoder. This logic can be extended, e.g. a RAM of size 1024×8, would require 10×1024 decoder as $2^{10} = 1024$. So it will have 10 address lines which will decide which word of the RAM array is to be selected.

For this implementation, the number of bits stored in each word would be 2 only, that is why every memory line will have 2 cells. Please note that for a word size of 2 bits, the RAM array would require 2 input and 2 output lines.

For this memory array, in case an address 01 is given as input of address selection bits, it will activate the Select input of cells of address 01 for read or write operation. Please note that current RAM chip design is not a 2 dimensional design as shown in Figure 4.18. It may follow a different more optimal organization, discussion on which is beyond the scope of this unit.

Check Your Progress 3

1) What are the differences between synchronous & asynchronous counters?

....

....

....

2) Is ripple counter same as shift register?

....

....

....

3) Design a two bit counter, which has the states 00, 01, 10, 00, 01, 10…..

....

....

....

4.6 SUMMARY

This unit introduces you the concepts of sequential circuits which is the foundation of digital design. Flip-flops are also a sequential circuit and the basic storage unit of a computer system. This unit also explains the working of a latch, which is the basic circuit that can be used for storing one bit of information. The sequential circuit can be formed using combinational circuits (discussed in the last unit) and flip flops. The unit also discusses the construction of some of the important sequential circuits like registers, counters, RAM. For more details, the students can refer to further reading.

4.7 SOLUTIONS / ANSWERS

Check Your Progress 1

1. A sequential circuit is designed to process and store data. Therefore, it consists of flip-flops for storing a state representing 0 or 1 and additional combinational circuit that may result in change of state depending on the combinational logic. Example of sequential circuits are - registers, counters etc. The main difference is that a sequential circuit also has a state.

2. Latch is a basic asynchronous sequential circuit designed with feedback to exhibit two different states, viz. 0 or 1. These states can be modified as per the input to latch. Example of latch is SR latch. Latches can change their state at any point of time based on input, whereas flip-flops are designed to change their states at specific time, for example on the occurrence of a clock pulse. Therefore, flip-flops have more complex circuitry than latch.

3. Excitation table are used for analysis and design of sequential circuits. They represent different combination of input to a flip-flop that may cause a specific state transition in the flip-flop. The following are the excitation tables of different flip-flops:

   SR Flip-flop

Present State (Q_t)	**Next State** (Q_{t+1})	**Input S and R**	**Input using DONOT CARE**
0	0	(i) S=0, R=0 (ii) S=0, R=1	**S=0, R=X**
0	1	S=1, R=0	**S=1, R=0**
1	0	S=0, R=1	**S=0, R=1**
1	1	(i) S=0, R=0 (ii) S=1, R=0	**S=X, R=0**

D Flip-flop

Present State (Q_t)	**Next State** (Q_{t+1})	**Input D**	**Input using DONOT CARE**
0	0	D=0	**D=0**
0	1	D=1	**D=1**
1	0	D=0	**D=0**
1	1	D=1	**D=1**

T Flip-flop

Present State (Q_t)	**Next State** (Q_{t+1})	**Input D**	**Input using DONOT CARE**
0	0	T=0	**T=0**
0	1	T=1	**T=1**
1	0	T=1	**T=1**
1	1	T=0	**T=0**

Check Your Progress 2

1. The master-slave flip is a simple structure, which changes its output during the clock pulse, when it is 0, thus, will result in synchronous state transitions of flip-flop.

2. An edge- triggered flip-flop changes it state either during the rising or falling edge of the clock pulse, thus, has a different construction than master-slave slip-flop.

Check Your Progress 3

1. All the flip-flops in the synchronous counter may change their state simultaneously, whereas in asynchronous counter, change of state of previous flip-flop may cause that effect to take place.

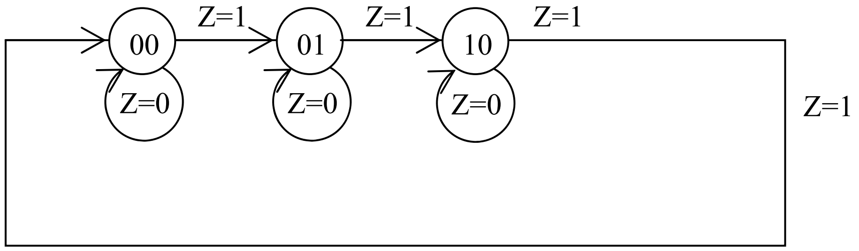
2. No, shift register causes shifting of state of a flip-flop to next flip-flop, whereas ripple counter is governed by the change of state.

3. The states 00, 01, 10, 00, 01, 10…..

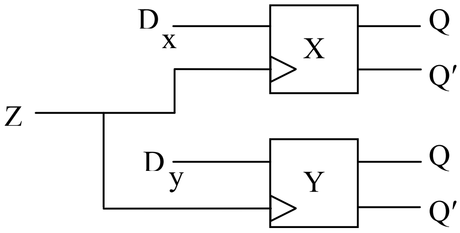
$$00 \longrightarrow 01 \longrightarrow 10 \longrightarrow 00 \longrightarrow 01$$

Assuming the control signal, say Z, state transitions are:

State



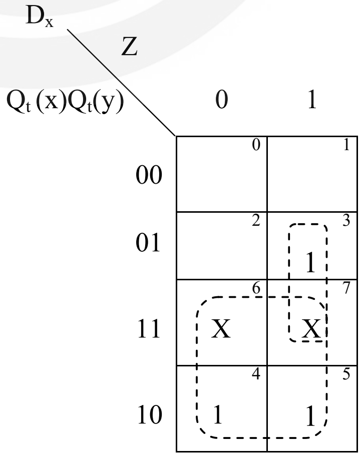
Rough design of the circuit would be:

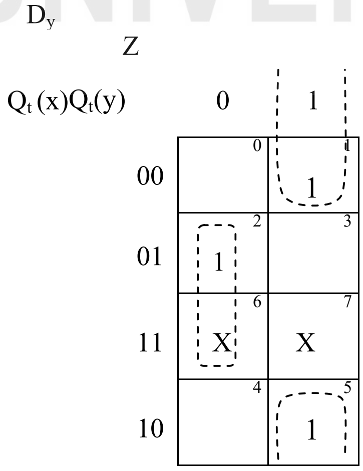


Truth table for flip-flop's X and Y:

	Present States of Flip-Flops			Next State of Flip-Flops		Required value of D_x for transition of X and D_y for the transition of Y	
	Flip-flops		Input	Flip-flops			
	Q_t of X	Q_t of Y	Z	Q_{t+1} of X	Q_{t+1} of Y	D_x	D_y
0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1
2	0	1	0	0	1	0	1
3	0	1	1	1	0	1	0
4	1	0	0	1	0	1	0
5	1	0	1	0	0	1	1
6	1	1	0	-	-	X	X
7	1	1	1	-	-	X	X

The K-maps for D_x and D_y can be drawn as:





D_x = Terms of (Adjacency of 4,5 + Adjacency of 3,7)

$D_x = Q_t(X) + Q_t(Y) . Z$

D_y = Terms of (Adjacency of 2,6 + Adjacency of 1,5)

$D_y = Q_t(Y) . Z' + Q'(Y) . Z$

Thus, the final counter circuit for the given states would be: