Chapter-6 : Sequential Logic

State table: Truth table are used to describe combina-circuit, but state tables are used to describe sequential circuits.

Representation of state machine using relationsheep between inputs present state, next state, and the outputs in the tabular format is known as the state table.

State diagram: The state diagram is the graphical re-presentation of the behaviour of sequential circuit. It clearly shows the transition of states from the input present state to the next and output for a corresponding input. In this diagram, each present state is represent-ed inside a circle.

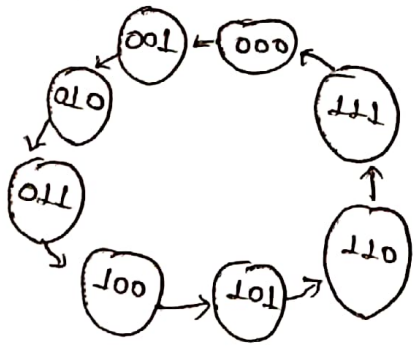

Figure-1: state diagram of a 3-bit binary counter

state equation: A state equation also known as an application equation. It is an algebric expression. that specifies the conditions for a flip-flop state transition. The left side of the equation denotes the next state of a flip flop and the right side denotes a boolean function that specifies the present state conditions that make the next state equal to 1.

The state equation is derived directly from a state table.

Flip-flop excitation table: During the design process we usually know the transition from present state to next state and wish to find the flip-flop input conditions that will cause the required transition. For this reason, we need a table that lists the required inputs for a given change of state. such list are called excitation table.

Table-1: Flip-flop excitation table

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(a) RS flip-flop

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(b) JK 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

Table-1 presents the excitation tables for the four flip flops. Each table consists of two columns, Q(t) and Q(t+1), and a column for each input to show how the required transition is achieved. There are four possibl

transitions from present state to next state. The required input conditions for each of four transitions are derived from the information available in the characteristic table. The symbol 'x' in the table represents a don't care condition, i.e. it does not matter whether the input is 1 or 0.

Design of Counters: A sequential circuit that goes to through a prescribed sequence of states upon the application of input pulse is called a counter. The input are called count pulses.

In a counter the sequence of states may follow a binary count or any other sequence of states. They are used for counting number of occurances of an event and useful for generating timing sequences to control operations in a digital system.

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 ~~form~~ from 0 to 2^n-1. As an example the state diagram of a three bit counter give bellow:

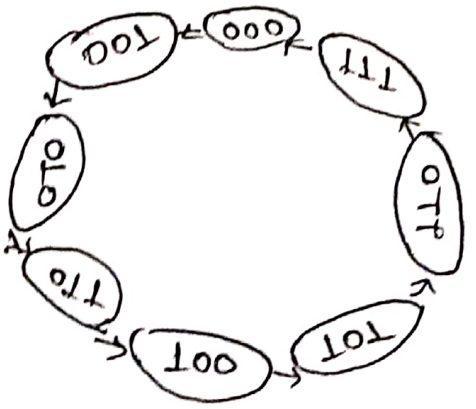

Figure-2: State diagram of a 3 bit binary counter

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 counting sequence of a 3-bit binary counter is give bellow:

Table-2: Excitation table for a 3-bit binary counter

Count sequence			Flip-flop inputs		
A_2	A_1	A_0	TA_2	TA_1	TA_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	1

The three flip-flops are given variable designations A_2, A_1, A_0. Binary counters are most efficiently constructed with T flip-flops.

As an illustration, consider the flip-flop input entries for now 001. The present state here is 001 and the next stage is 010, which is the next count in the sequence. Comparing these two counts we note that A_2 goes from 0 to 0. So TA_2 marked with a 0 because flip-flop A_2 must remain unchanged when

clock pulse occurs. A_1 goes from 0 to 1 so TA_1 must be 1. Similarly A_0 goes from 1 to 0. So TA_0 is marked with a 1. The last row 111 compared with 000 which is its next stage, all the values are complemented here.

A_2 \ A_1A_0	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$TA_2 = A_1A_0$

A_2 \ A_1A_0	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$TA_1 = A_0$

A_2 \ A_1A_0	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$TA_0 = 1$

Figure: Maps for a 3 bit counter

The flip-flop input functions from the excitation tables are simplified in the map figure: 3. The boolean function listed under each map specify the combinational circuit part of a counter. Indicating these with three flip-flops, we obtain the logic diagram of the counter shown in bellow:

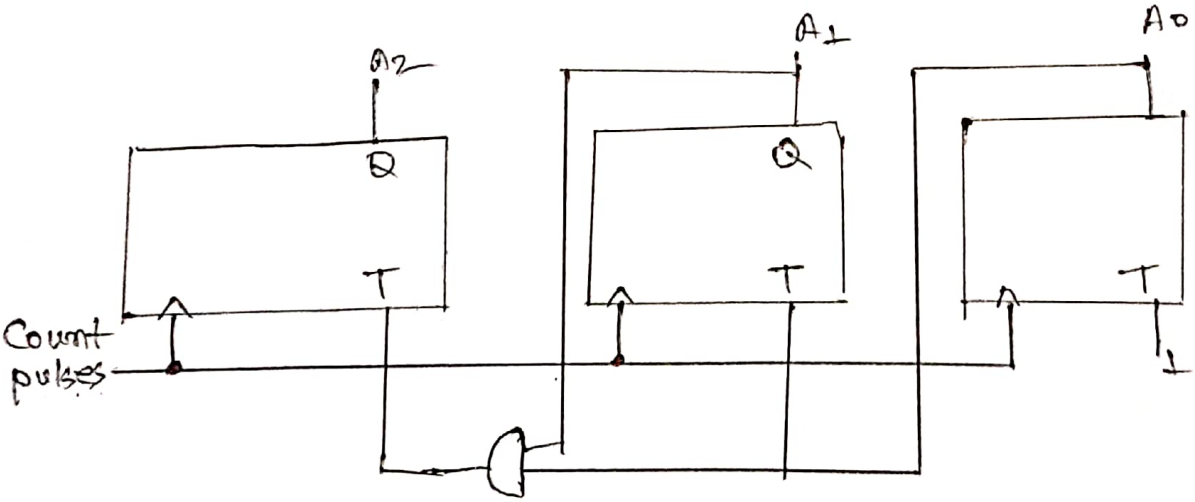

Figure-4: Logic diagram of a 3-bit binary counter

A BCD counter counts the binary sequence from 0000 to 1001 and returns to 0000 to repeat the sequence.

Chapter - 7

Registers, Counters and the memory unit

Introduction: A clocked sequential circuit consists of a group of flip-flops and combinational gates connected to form a feedback path. A circuit with only flip flop is considered a sequential circuit even in the absence of combinational gates.

An MSI circuit that contains storage cells within it is, by definition, a sequential circuit. MSI circuit are classified in one of three categories

(i) registers (ii) counters (iii) Random-access memory

Register: A register is a group of binary storage cells suitable for holding binary information. A group of flipflops constitutes a register, since each flip-flop is a binary cell capable of storing one bit of information. An n bit register has a group of n flip flop and is capable of storing any binary information containing n bits.

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.

Random access memory: A memory unit is a collection of storage cells together with associated circuits needed to transfer information in and out of storage. A random access memory (RAM) differs from a read only memory (ROM) in that a RAM can transfer the stored information out (read)

and it also capable of recieving new information in for stonage (write). A more appropriate name for such a memory would be read write memory.

Registers, counters, memories are extensively used in the design of digital systems in general and digital computers in particular. Register can also be used to faciliate the design of sequential circuits. Counters are useful for generatings timing variables to sequence and control the operating in a digital system.

7-2 Registers: The simplest possible register is one that consists of only flip-flops without any external gates.

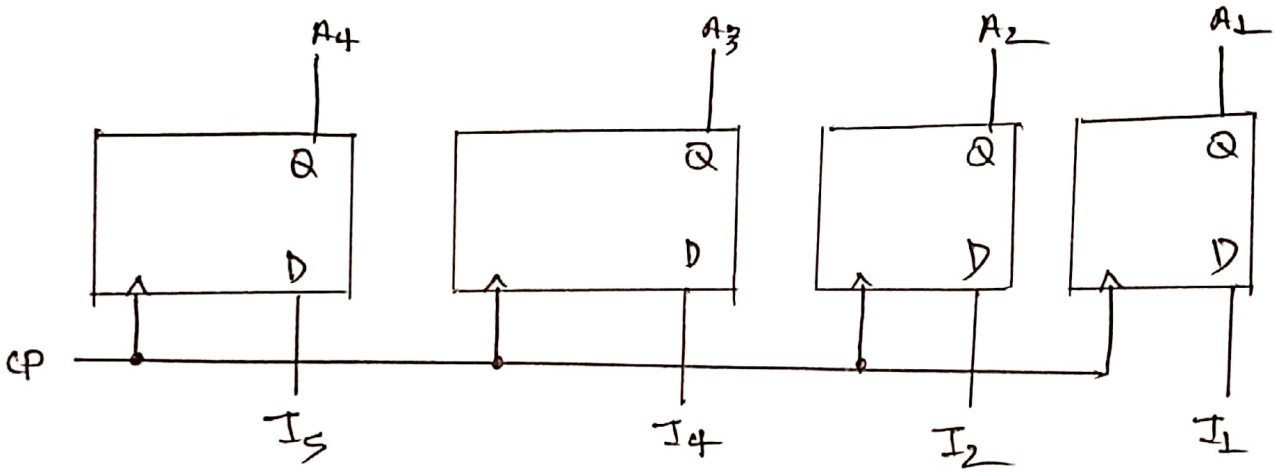

Figure-1: 4-bit register

Figure-1: Shows such a register constructed with four D-type flipflops and a common clock pulse input The clock pulse input, cp, enables all flip-flops so that information presently available at four inputs can be transferred into the 4-bit register. If the flip-flops are constructed with gated D type Catches then information present at a data (D) input is

transferred to the Q output when the enable CP is 1. and the Q output when the enable follows the input data as long as the CP signal remains 1. When CP goes to 0, the information that was present at the data input just before the transistion is retained at the Q output.

Register with parallel Load: The transfer of new information into a register is reffered to as loading the register. If all the bits of the register are loaded simultaneously with a single clock pulse, we say that the loading is done in parallel.

A 4-bit register with a load control input using RS flip-flops is shown in figure-2; The CP input of the registers recieves contineously synchronized pulses which are applied to all flip-flops. The inverter in the CP path causes all flip-flops to be triggered by the negative edge of the incoming pulses. The purpose of the inverter is to reduce the loading of the master-clock generator. This is because the CP input is connected to only one gate (the inverter) instead of the four gate inputs that would have been required if the connections were made directly into the flip-flop clock inputs (marked with small triangle.

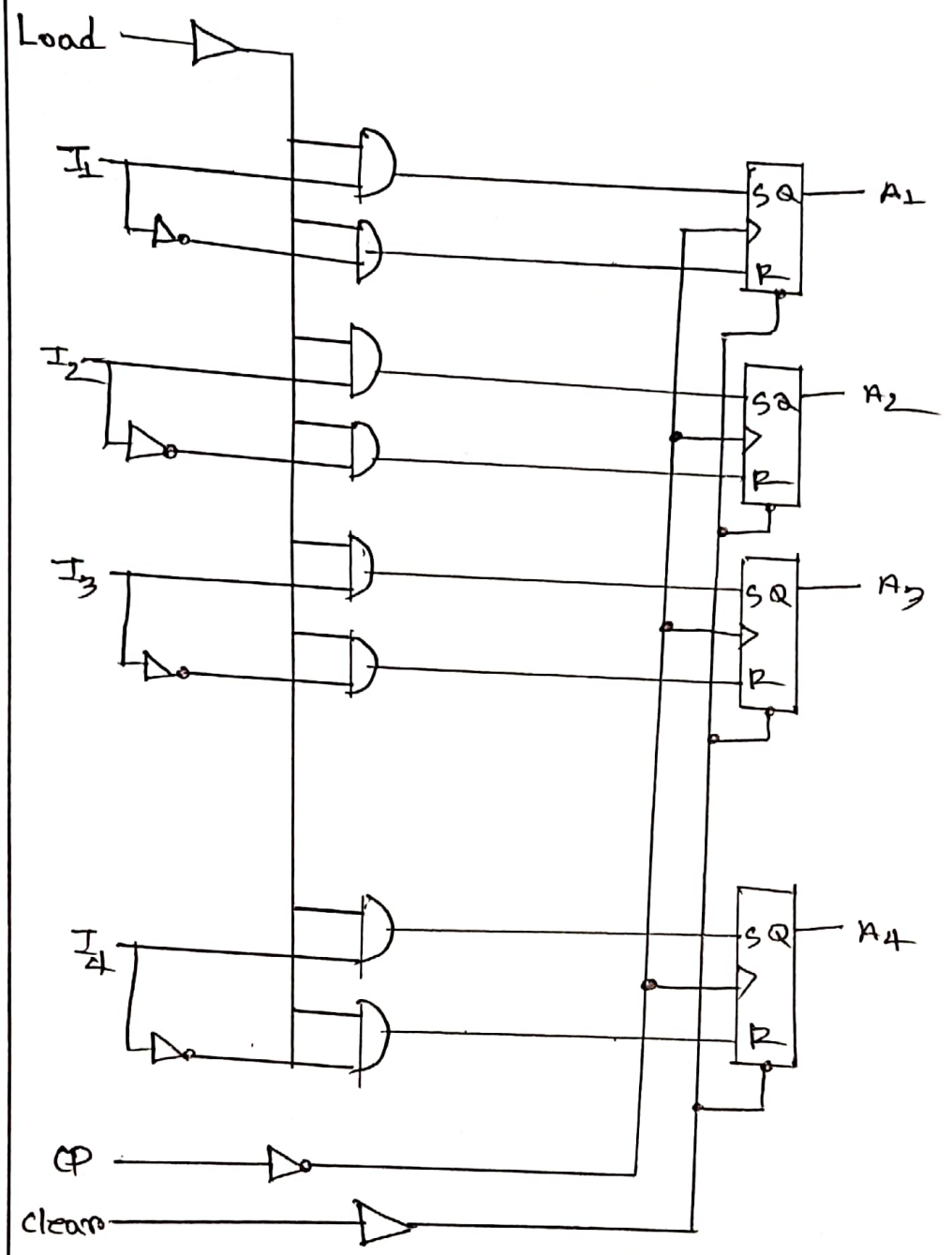

Figure-2: 4-bit register with parallel load

The clear input goes to a special terminal in each flip flop through a noninverting buffer gate. When terminal goes to 0, the flip-flop is cleared asynchronously. The clear input is useful for clearing the

register to all 0's prior to its clocked operation. The load input goes through a buffer gate (to reduce loading) and through a series of AND gates to the R and S inputs of each flip-flop. The two AND gates and an inverter associated with each input I determine the values of R and S. When the load control goes to 1, inputs I_1 through I_4 specify what binary information is loaded into the register on the next clock pulse. If the buffer gate associated with the load input is changed to an inverter gate, then the register is loaded when the load input is 0 and inhibited when the load input is 1.

Sequential Logic implementation: A clocked sequential circuit consists of A group of flip-flops and combinational gate. Since Registers are redily available as MSI circuits, it is a part of the sequential circuit. According to figure-3 the present state of the register and extendrnal inputs determine the next state of the register and the values of external outputs. part of the combinational circuit determines the next state and the other part generates the outputs.

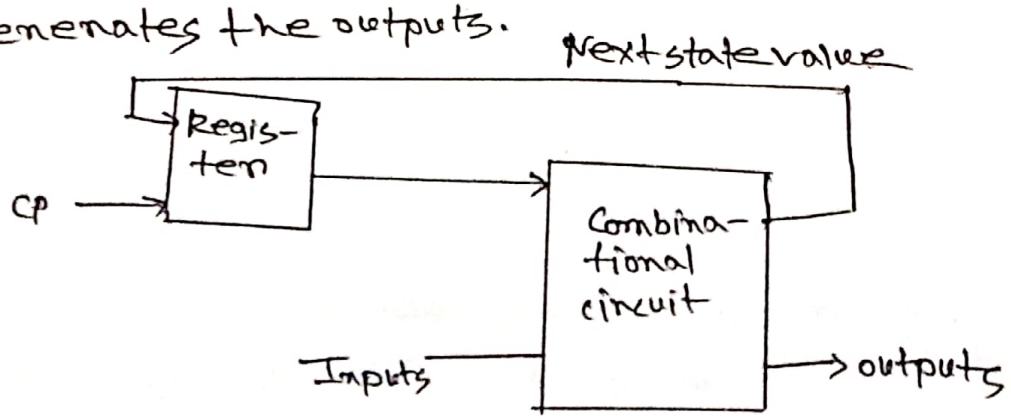

Figure-3 : Block diagram of sequential circuit

The next state value from the combinational circuit is loaded into the register with a clock pulse. If the register has a load input, it must be set to 1. otherwise, if the register has no load input, the next state value will be transferred automatically every clock pulse. By using a register, it is possible to reduce the design of a sequential circuit to that of a combinational circuit connected to a register.

7-3 shift register: A register capable of shifting its binary information either to the right or the left is called a shift register. The logical configuration of a shift register consists of a chain of flip-flop connected in cascade with the output of one flip-flop connected to the input of the next flipflop. All flipflops recieve a common clock pulse which cause the shift from one stage to the next.

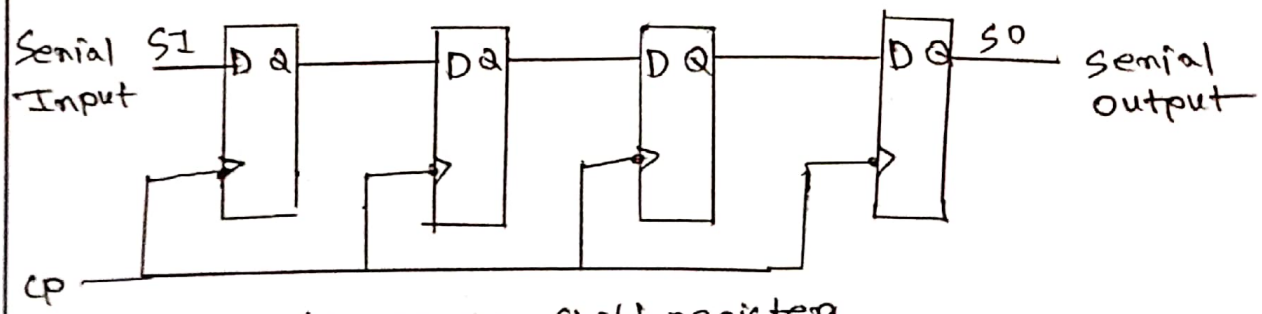

Figure-4: Shift register

The simplest possible shift register is one that uses only flipflops, as ~~shown~~ shown in figure-4. The Q output of a given flip-flop is connected to the D

~~inp~~ input of the flip-flop at its right. Each clock pulse shift the contents of the register one bit position to the right. The serial input determines what goes into the leftmost flip-flop during the shift. The serial output is taken from the output of the right most flip-flop prior to the application of a pulse.

Thus undirected shift register can function either as a shift right or as a shift left ~~rgi~~ register.

The register in figure-4 shifts its contents with every clock pulse during the negetive edge of the pulse transition. If we want to control the shift so that it occurs only with certain pulses but not with others, we ~~most~~ must control the cp input of the register.

7-4 Ripple counters: MSI counters come in two categories ripple counters and synchronous counters.

In a ripple counter, the flip-flop output transistion serves as a source for triggering other flip-flops.

In a synchronous counter, the input pulses are applied to all cp inputs of all flip-flops. The change of a state of a particular flip-flop is dependent on the present state of other flip-flops.

Binary Ripple Counter: A binary ripple counter consists of a series connection of complementing flip-flops (T or JK type) with the output of each flip-flop connected to the cp input of the next higher order flip-flop.

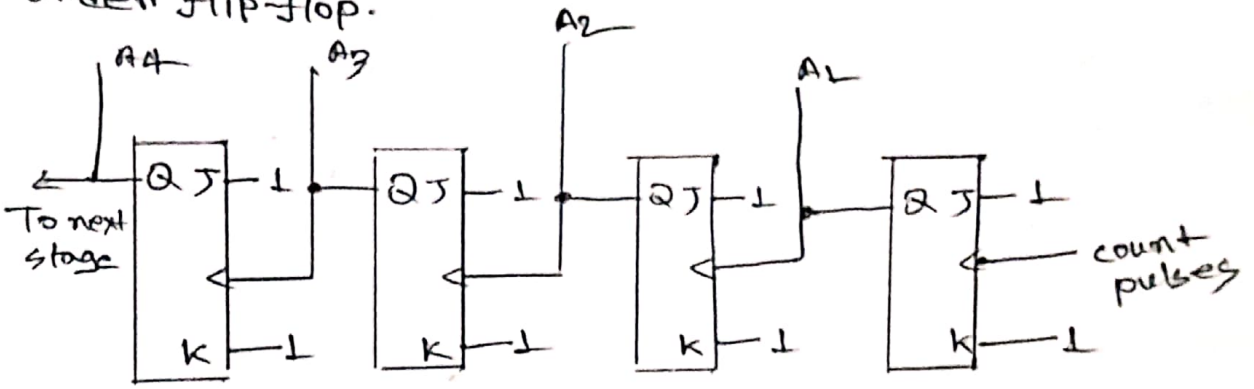

Figure-5: 4-bit binary ripple counter

In figure-5, All J and K inputs are equal to 1. The small circle in the cp input indicates that the flip-flop complements during a negative going transition or when the output to which it is connected goes from 1 to 0. In table-1, we can see the lowest order bit A_1 must be complemented with each other count pulse. Everytime A_1 goes from 1 to 0, it complements A_3.

Table-1: Count sequence for a binary ripple counter

Count Sequence A_4 A_3 A_2 A_1	Condition for complementing flipflops.
0 0 0 0	Complement A_1
0 0 0 1	Complement A_1, A_1 will go from 1 to 0 and complement A_2
0 0 1 0	Complement A_1
0 0 1 1	Complement A_1, A_1 will go from 1 to 0 and complement A_2, A_2 will go from 1 to 0 and complement A_3.
0 1 0 0	complement A_1
0 1 0 1	Complement A_1, A_1 will go from 1 to 0 and complement A_2.
0 1 1 0	Complement A_1
0 1 1 1	Complement A_1, A_1 will go from 1 to 0 and complement A_2, A_2 will go from 1 to 0 and complement A_3, A_3 will go from 1 to 0 and complement A_4.
1 0 0 0	and so on.....

and so on. For example, take the transition from count 0111 to 1000. The arrows in the table emphasize the transitions in this case. A_1 is complemented with the count pulse. Since A_1 goes ~~to~~ from 1 to 0, it triggers A_2 and complemented it. As a result A_2 goes from 1 to 0 which in turn complemented A_3. A_3 goes from 1 to 0. which complements A_4. The output transition of A_4, if connected to next stage, will not trigger the next flipflop since it goes from 0 to 1. The flip flops

Change one at a time in rapid succession, and the signal propagates through the counter in a ripple fasion. Ripple counters are sometimes called asynchronous counters.

BCD ripple counter: A decimal counter follows a sequence of ten states and return to 0 after the count of 9. If BCD is used, the sequence of states is as shown in the state diagram of figure-6 This similar to a binary counter, except that the state after 1001 is 0000.

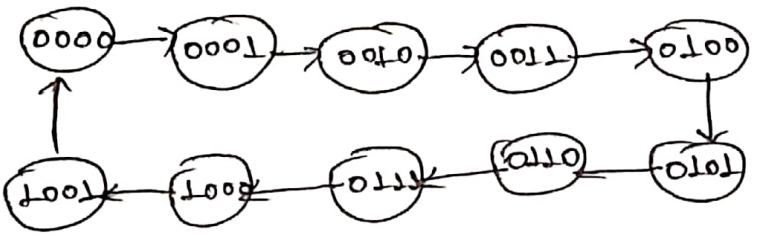

Figure-6: State diagram of a decimal BCD counter

The design of a decimal ripple counter of any ripple counter not following the binary sequence is not a straightforward procedure.

The logic diagram of a BCD ripple counter is shown in Figure-7

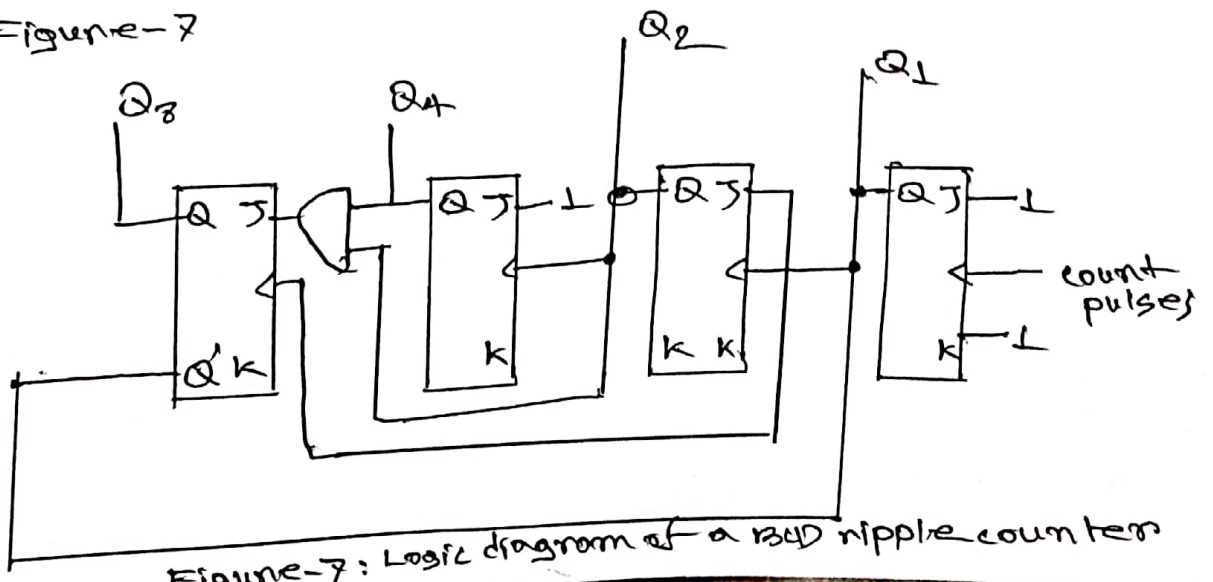

Figure-7: Logic diagram of a BCD ripple counter

The four outputs are designated by the letter symbol Q with a numeric subscript equal to the binary weight of the corresponding bit in the BCD code. The flipflops trigger on the negative edge, i.e when the cp signal goes from 1 to 0. Note that the output of Q_1 is applied to the cp inputs of both Q_2 and Q_8 and the output of Q_2 is applied to the cp input of Q_4. The JK input are connected either to a permanent 1 signal or to outputs of flip-flops as shown in the diagram. The following are the conditions for each flip-flop state transition:

1. Q_1 is the complemented on the negative edge of every count pulse.

2. Q_2 is complemented if $Q_8 = 0$ and Q_1 goes from 1 to 0.

3. Q_4 is complemented when Q_2 goes from 1 to 0.

4. Q_8 is complemented when $Q_4 Q_2 = 11$ and Q_1 goes from 1 to 0. Q_8 is cleared if either Q_4 or Q_2 and Q_1 goes from 1 to 0.

Johnson Counter: A k-bit ring counter circulates a single bit among the flip-flop to provide k distinguishable states. The number of states can be doubled if the shift register is connected as a switch-tail ring counter. A switch-tail ring counter is a circular shift register with the complement output of the last flip-flop connected to the input of the first flip-flop.

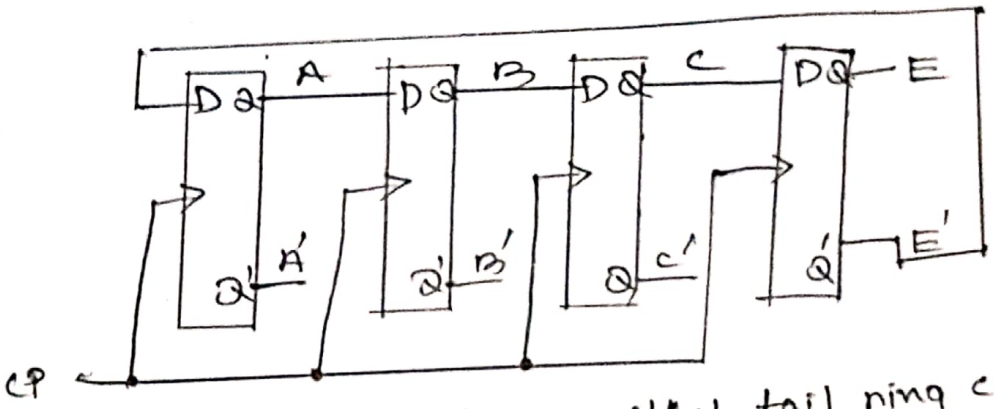

Figure-8: 4-stage switch tail ring counter

Figure-8 shows such a shift register. The circular connection is made from the complement output of the rightmost flip-flop to the input of the left most flip-flop. The register shifts its contents once to the right with every clock pulse, and at the same time, the complement value of the E flip-flop is transferred into the A flip-flop. Starting from a cleared state, the switches-tail ring counter goes through a sequence of eight state as listed in table-2. In general, a k-bit switch tail-ring will go through a sequence of 2k states.

Table-2: Count sequence and required decoding:

Sequence Number	Flip-flop output A B C E	AND gate required for output
1	0 0 0 0	$A'E'$
2	1 0 0 0	AB'
3	1 1 0 0	BC'
4	1 1 1 0	CE'
5	1 1 1 1	AE
6	0 1 1 1	$A'B$
7	0 0 1 1	$B'C$
8	0 0 0 1	$C'E$

Starting from all 0's, each shift operation inserts 1's from the left until the register is filled with all 1's. In the following sequence 0's are inserted from the left until the register is again filled with all 0's.

One disadvantage of the circuit in figure-8, is that if it find itself is an unused state, it will persist in moving from one invalid state to another and never find its way to a valid state.

One connecting procedure is to disconnect the output from flip-flop B that goes to the D inputs of flip-flop C, an instead enable the input of flipflop C by the function $DC = (A+C)B$

where DC is the flipflop input function for the D input of the flip-flop C.