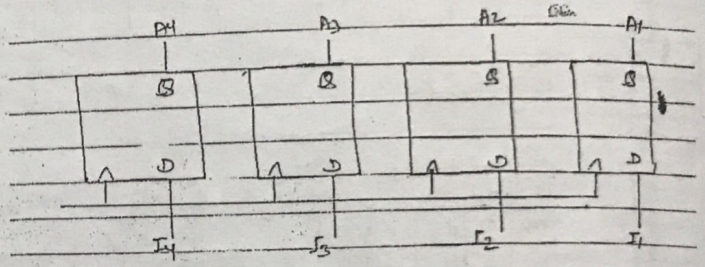
Ch:8
Registers, Counters and Memory Unit.

Registers:

A register is a group of binary storage cells suitable for holding binary information. A group of flip-flops constitutes a register, since each flf is a binary cell capable of storing one bit of information. A n-bit register has a group of n flfs and is capable of storing any binary information containing n bits. In addition to the flfs a register may have combinational gate that performs certain data processing tasks. The flfs holds binary information and the gates control when and how new information is transferred into the register.

The simplest possible register is one that consists of only flfs without any external inputs. Figure below shows a register constructed with 4 D type flfs and a common CP inputs. The CP enables all flfs so that information presently available at 4 I/Ps can be transferred into 4 bit register.



The information present at the data D input is transferred to the Q output when the enable (CP) is 1 and the Q output 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 transition is retained at the Q output. In other words, flfs are sensitive to the pulse duration & the register is enabled for as long as CP=1.

Shift Register:

A register capable of shifting binary information either to the left or right is called a shift register. The logical configuration of shift register consists of chain of flfs connected in cascade, with the output of one flf connected to the input of the next flf. All flfs receive a common CP which causes the shift from one stage to the next.

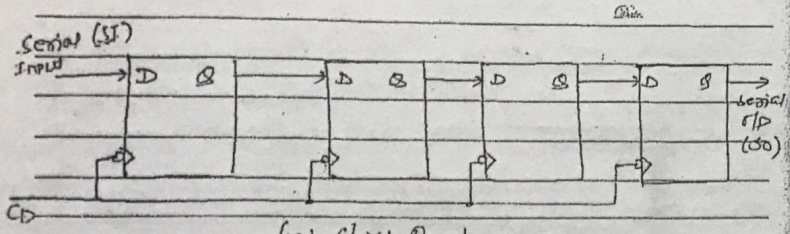
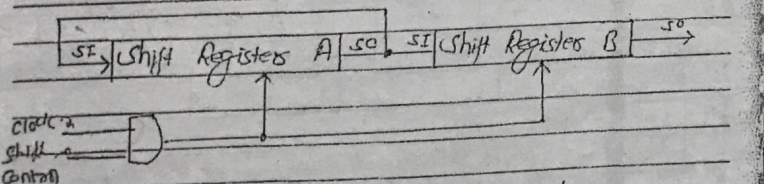


fig: Shift Register

→ The Q o/p of the given flf is connected to the D input of the flf at its right.
→ The SI determines what goes into the leftmost flf during the shift.
→ The SO is taken from the o/p of the rightmost flf prior to the application of a pulse.
→ The register in above figure shifts its contents with every CP during negative edge of the pulse transition (This is indicated by a small circle).

Serial Transfer:



→ The shift control input determines when and how many times the registers are shifted. This is done by the AND gate that allows clock pulses to pass into the CP terminals only when the shift control is 1.

→ Assume that the binary content of A before the shift is 1011 and that of B is 0010.
→ The serial transfer from A to B occurs in 4 steps as shown below.

→ After the 1st pulse, the rightmost bit of A is shifted into the leftmost bit of B and at the same time, the bit is circulated into the leftmost position of A. The other bits of A & B are shifted once to the right. The previous '50' from B is lost and its value changes from 0 to 1. The next 3 pulse performs identical operation shifting the bits of A into B, one at a time.
→ After the 4th shift, the shift control goes to zero and both registers A & B have the value 1011. Thus, the content of A is transferred to the content of B while the content of A remains unchanged.

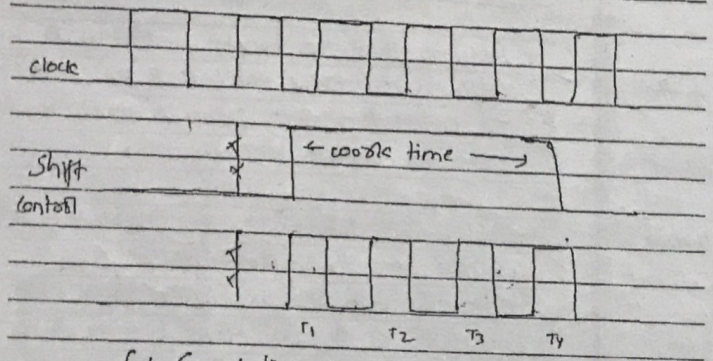


fig: Serial transfer from reg A to B

Worktime: Time required to shift the entire content of the shift register

Timing pulse	Shift Register A	Shift Reg B	00 of B
Initial value	1 0 1 1	0 0 1 0	→ 0
T1	1 1 0 1	1 0 0 1	→ 1
T2	1 1 1 0	1 1 0 0	→ 0
T3	0 1 1 1	0 1 1 0	→ 0
T4	→ 1 0 1 1	1 0 1 1	→ 4

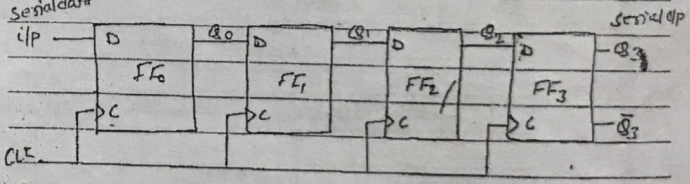
Types of shifting:

According to the types of shiftings, registers may be 4 types.

(i) Serial-In-Serial Out
(ii) Serial In - Parallel Out
(iii) Parallel In - Serial Out
(iv) Parallel In - Parallel out.

(i) Serial-In-Serial Out Shift Register:

The serial IN/Serial OUT shift register accepts data serially i.e one bit at a time on a single line. It produces the stored information on its output also in serial form.



1101

Figure shows the entry of the 4 bits 1010 into the register begining with the rightmost bit. The register is initially clear. The 0 input on the data input, making D=0 for FF0, when the 1st CP is applies, FF0 is reset thus storing the 0. Next, the 2nd bit which is 1, is applied to the data input, making D=1 for FF0 and D=0 for FF1 because the D input

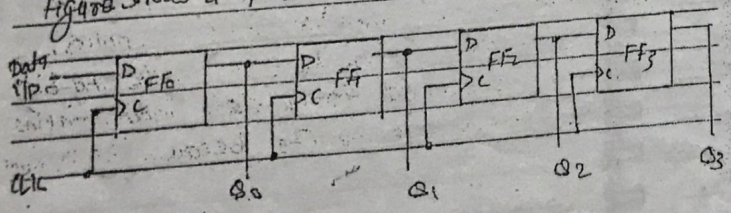
of FF1 is connected to the Q_0 output when second CP occurs, the 1 in the data is shifted into FF0, causing FF0 to set and then 0 that was in FF0 is shifted into FF1.

Similarly this process completes the serial entry of the 4 bits into the shift register, where they can be stored for any length of time as long as the flip have dc power. Thus we see that that 4 bit serial data, one bit at a time is transferred.

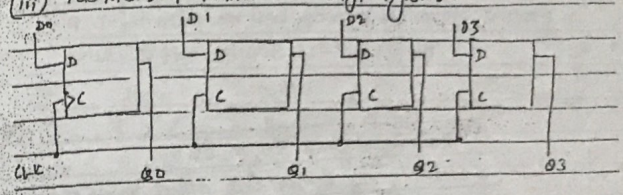
(ii) Serial IN/Parallel OUT Shift Registers:

Data bits are registered serially (right-most bit first) into this type of register in the same manner as discussed in serial in/serial out shift register. The difference is the way in which the data bits are taken out of the register in the parallel output register, the output of each stage is available. Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously, rather than on a bit-by-bit basis as with the serial output.

Figure shows a 4-bit serial in/Parallel out shift register.

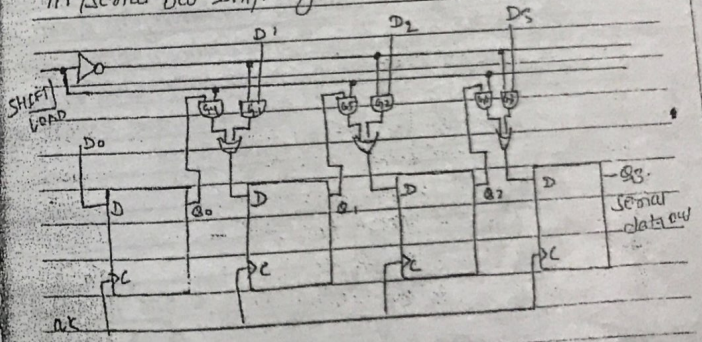


(iii) Parallel IN/Parallel OUT Shift Register:



(iv) Parallel IN/Serial OUT Shift Register:

For a register with parallel data inputs the bits are entered simultaneously into their respective stages on parallel lines rather than on a bit-by-bit basis on on line as with serial data inputs. The serial output is the same as described as previous types, once the data are completely stored in the register. Fig below shows 4 bit parallel in/serial out shift register.



→ 4 Data input lines D_0, D_1, D_2, D_3

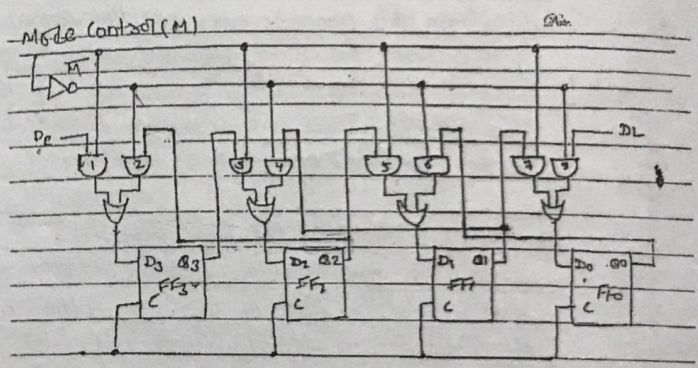
→ SHIFT/$\overline{LOAD}$ input line allows 4 bit of data to load in parallel into the registers. When SHIFT/$\overline{LOAD}$ is low, gates G1 through G3 are enabled allowing each data bit to be applied to the D input of its respective FF.

→ When a CP is applied, the FF with D=1 will SET and those with D=0 will reset, thereby storing all 4 bits simultaneously.

→ When SHIFT/$\overline{LOAD}$ is HIGH, gate G1 through G3 are disabled and gates G4 through G6 are enable, allowing the data bits to shift right from one stage to the next. The OR gates allows either the normal shifting operation or the parallel data entry operation depending on which AND gates are enabled by the level on the SHIFT/$\overline{LOAD}$ input.

Bidirectional Shift Register:

A bidirectional shift register is one in which the data can be shifted either left or right. It can be implemented by using gate logic that enables the transfer of data bit from one stage to the next stage to the right or to the left depending on the level of mode control line (M).



M = 1

Shift Right Operation

M = 0

Shift left Operation

$(11)_2 \rightarrow (3)_{10}$

Shift left

$(110)_2 \rightarrow (6)_{10}$

i.e $(3)_{10} \xrightarrow{\times 2} (6)_{10}$

Shift Right

$(6)_{10} \xrightarrow{\div 2} (3)$

Hence, shift left $\rightarrow \times 2$

shift right $\rightarrow \div 2$

8) Design a combinational ckt using ROM. The ckt accepts a 3-bit number and generates an o/p binary no. equal to the square of the i/p no.

Soln

Input	Output
x y z	B5 B4 B3 B2 B1 B0
0 0 0	0 0 0 0 0 0
0 0 1	0 0 0 0 0 1
0 1 0	0 0 0 1 0 0
0 1 1	0 0 1 0 0 1
1 0 0	0 1 0 0 0 0
1 0 1	0 1 1 0 0 1
1 1 0	1 0 0 1 0 0
1 1 1	1 1 0 0 0 1

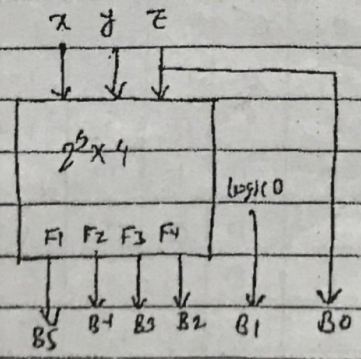
connected always 0

i/p = 3

So, no. of combⁿ = $2^3 = 8$

We don't need to show B1 & B0 from ROM. So programming is done only for B2, B3, B4, B5 & hence o/p of ROM is 4bit

∴ Size of ROM = $2^3 * 4 = 8 * 4$



Asynchronous Counter:-

1) Binary Ripple Counter:

A binary ripple counter consists of a series connection of complementary FFs (T or JK type), with the output of each flf connected to the CP input of the next higher flfs.

The flf holding the LSB receives the incoming count pulse. The diagram of a 4-bit binary ripple counter is shown in fig below. All J & K inputs are equal to 1. The small circle in the CP indicates that the flf complements during a negative-going transition ~~that the flf complements to 0's~~ when the output to which it is connected goes from 1 to 0.

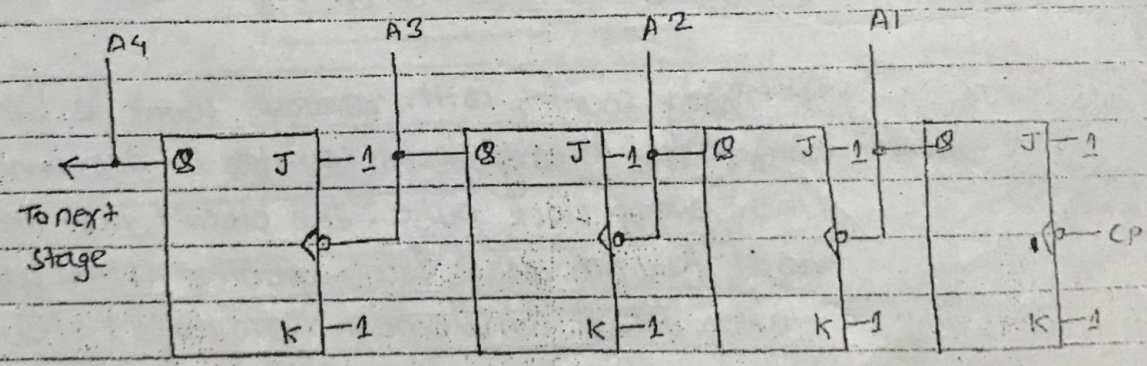
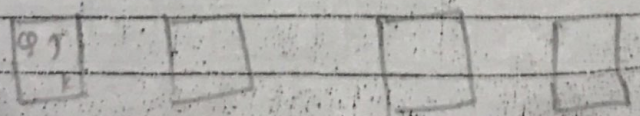


fig: 4-bit binary ripple counter



Operation

Count Sequence				Conditions for complementing flfs
A4	A3	A2	A1	
0	0	0	0	Complement A1
0	0	0	1	" A2, A2 will go from 1 to 0 & comp. A2
0	0	1	0	" A1
0	0	1	1	" A1, A1 will go from 1 to 0 & comp A2; A2 " " " 1 to 0 & " A3
0	1	0	0	complement A1
0	1	0	1	" A1; A1 will go from 1 to 0 & comp. A2
0	1	1	0	" A1
0	1	1	1	" A1 ; A1 will go from 1 to 0 & com A2 ; A2 will go from 1 to 0 & com A3 ; A3 " " " 1 to 0 & com A4
1	0	0	0	and so on----

A binary counter with reverse count is binary down counter. The binary down counter is decremented by 1 with every clock pulse. The circuit just down above will function as a down counter if the outputs are taken from complement terminals i.e $\bar{Q}$ of all flfs.

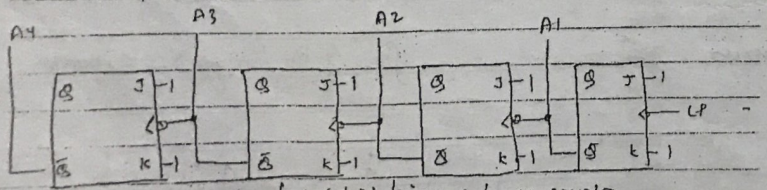
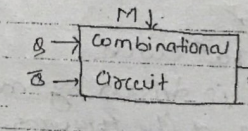


fig: 4 bit binary down counter.

A4	A3	A2	A1
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	0
0	1	1	1

3 bit UP/DOWN Ripple Counter:



M=0 → Upcounting ⇒ Q connected to
M=1 → Down Counting ⇒ $\bar{Q}$ "

M	Q	$\bar{Q}$	Y	
0	0	0	0	UP counting
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	0	DOWN counting
1	0	1	1	
1	1	0	0	
1	1	1	1	

K-map for Y

M \ Q$\bar{Q}$	00	01	11	10
0	0	0	1	1
1	0	1	0	0

$Y = M\bar{Q} + \bar{M}Q$

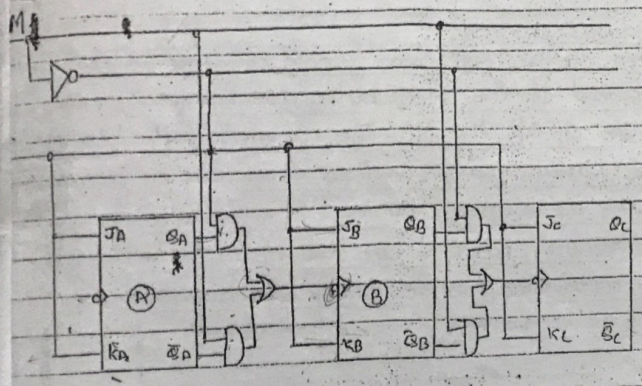


fig: 3 bit updown counter

CD Ripple Counter:

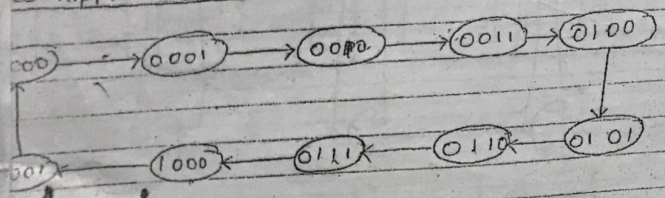


fig: State diagram of BCD ripple counter

→ follows ten states
→ returns to 0 after the count of 9.
→ Similar to binary counter except that the state after 1001 is 0000.

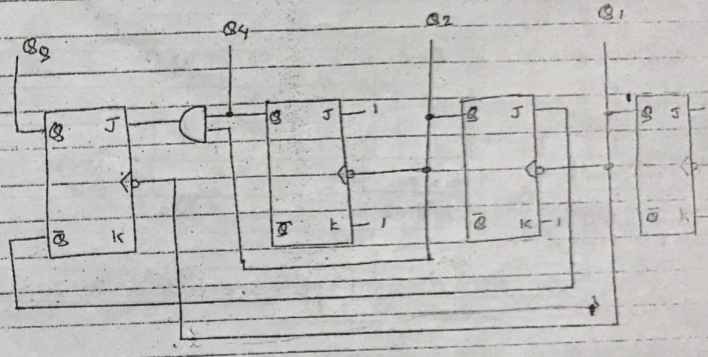


fig: Logic diagram of a BCD ripple counter.

→ FFs triggers on -ve edge i.e when CP signal goes from 1 to 0.
→ When CP goes from 1 to 0, if FF is set if J=1, and cleared if K=1, is complemented if J=K=1 and is uncharged if J=K=0. The following are the conditions for each flip-flop state transition.
1. Q_1 is complemented on the negative edge of every count pulse.
2. Q_2 is complemented if Q_1 goes from 1 to 0.

Q_4 complemented when Q_2 goes from 1 to 0

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 is 0 and Q_1 goes from 1 to 0.

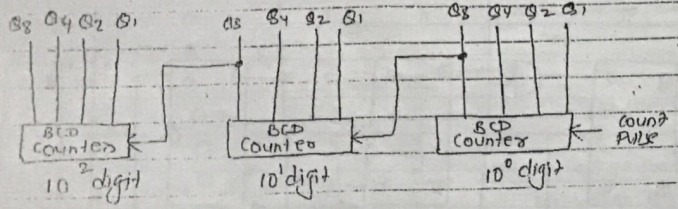


fig: Block diagram of a 3-decade decimal BCD counter

Q_8	Q_4	Q_2	Q_1
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
1	0	0	0
1	0	0	1

Synchronous Counters:

They are distinguished from ripple counter in that CP are applied to the clock pulse are applied to the clock i/ps of all the flfs. The common pulse trigger all flfs simultaneously rather than one at a time in succession as in ripple counter.

Synchronous counter can be distinguished using the procedures for designing synchronous sequential circuits.

1.) Binary Counter:

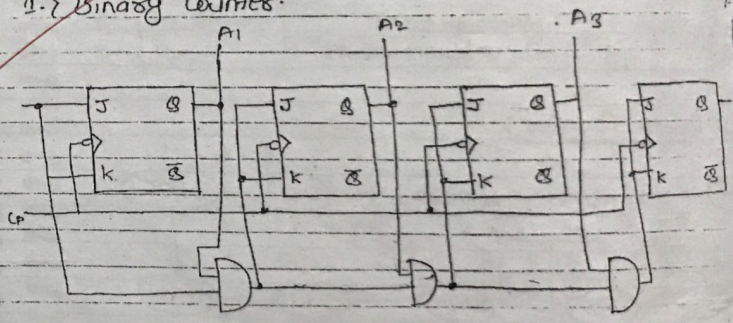
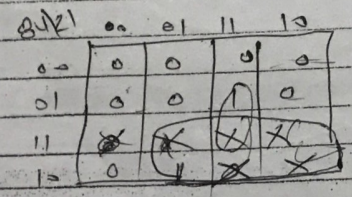


fig: 4-bit binary counter

A4	A3	A2	A1
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
1	0	0	0

and so on.

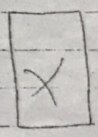
lowest bit A1 is complemented on every clock pulse
A2 is complemented if A1 is 1
A3 is complemented if both A1 & A2 are 1
A4 is complemented if all A1, A2 & A3 are 1.



2) BCD Counter

Present state				Next state				Excitation Table			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	T_A	T_B	T_C	T_D
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	1
0	1	1	1	1	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	1

$T_A = Q_8Q_1 + Q_4Q_2Q_1$; $T_B = Q_2Q_1$; $T_C = \overline{Q_8}Q_1$; $T_D = 1$ } from K-map



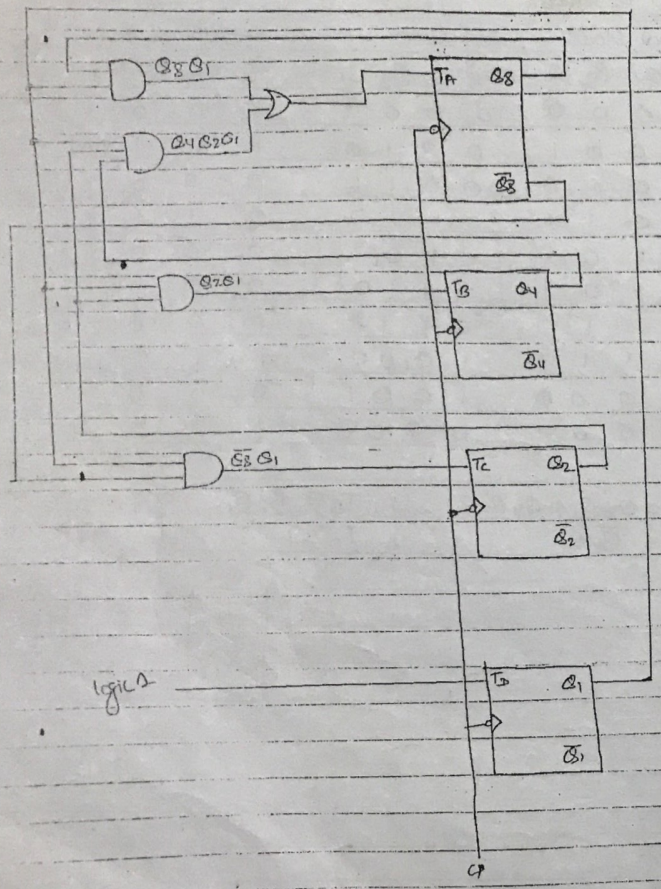
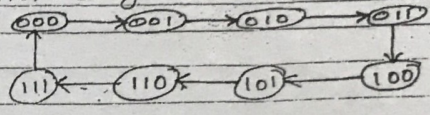


fig: Logic Diagram of BCD counter

Design a 3-bit synchronous up counter using T-flipflop



ring Counter:

t ring counter is a circular shift ~~any~~ register with only one flf being set at any particular time, ll others are cleared. The single bit is shifted one flf to the other to produce the sequence of timing signal.

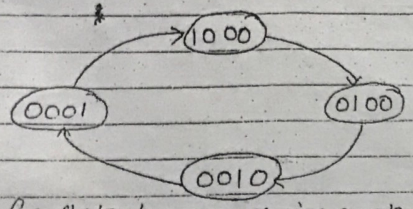
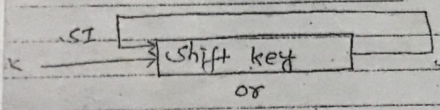


fig: State diagram of ring counter



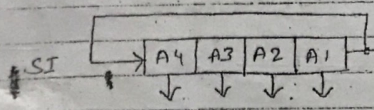


fig: Ring Counter implemented using shift register

~~It~~ is a k-bit ring counter that circulates a single bit among the flipflop to provide k-distinguished state.

→ No. of distinct states is equal to the no. of flipflop used in the counter. And n-bit ring counter can count only n-bits whereas the same ripple can count upto 2^n bits.

→ Advantage of ring counter is decoding circuit is not required, simple and fast operation.

Switch-Tail Ring Counter:

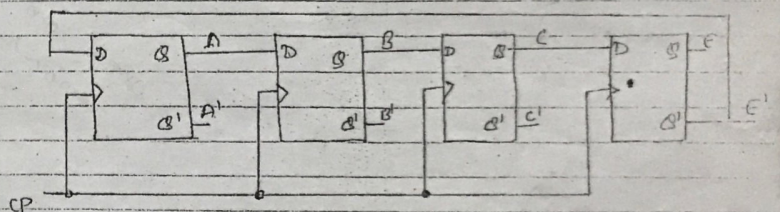


fig: Switch-tail ring counter

→ A k-bit switch tail ring counter generates 2k timing sequences follows a regular pattern.

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

Johnson Counter:

The combination of k-bit switch-tail ring counter and 2k decoding gates to provide o/p for 2k timing signals is Johnson Counter.

→ It is more cost effective

Sequence No.	FF o/p A B C E	AND gate required for o/p
1.	0 0 0 0	AE'
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	$C'E$

0 state - complement 2 extreme o/p
1 state - normal 2 extreme o/p

→ All 0's state is decoded by taking the complement of the two extreme flfs o/p.

→ All 1's state is decoded by taking the normal o/p of the two extreme flfs.

→ All other states are decoded from an adjacent 1 0 and 0 1 pattern in the sequence.

Disadvantage: Once it moves to an unused state, it will continue to move from one invalid state to next invalid state and never finds its way to a valid state.

Random Access Memory (RAM):

A memory unit is a collection of storage cells together with associated circuits needed to transfer information in and out of the device.

A memory unit stores binary information in groups of bits called words. A word in memory is an entity of bits that move in and out of storage as a unit.

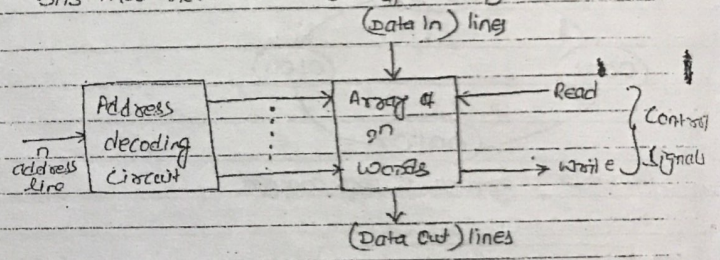


fig: Block Diagram of RAM

Types of RAM:

① Static RAM (SRAM)

Static RAMs use flfs as storage elements and can transfer therefore, store data indefinetly as long as dc power is applied.

② Dynamic RAM (DRAM)

Dynamic RAMs use capacitor as storage element and cannot retain data very long without the capacitors being recharged by a process called refreshing.

both SRAM & DRAM will lose stored data when dc power is removed and therefore are classified as volatile memories.

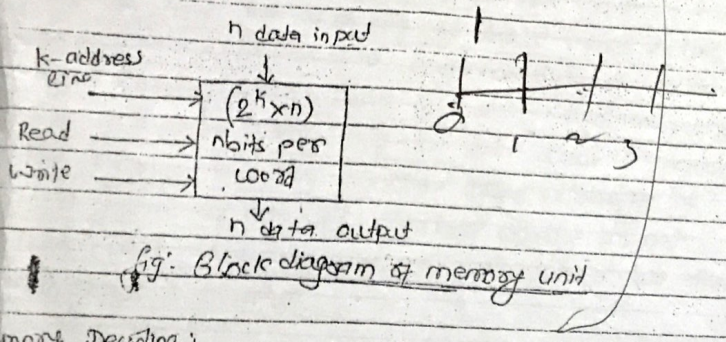


fig: Block diagram of memory unit

Memory Decoding:

In addition to storage components in a memory unit, there is a need for decoding circuits to select the memory word specified by the input address.

Internal Construction:

The internal construction of random-access memory of m words with n bits per word consists of mxn binary storage cells and associated decoding circuits for selecting individual words. The Binary Cell (BC) is the basic building block of a memory unit.

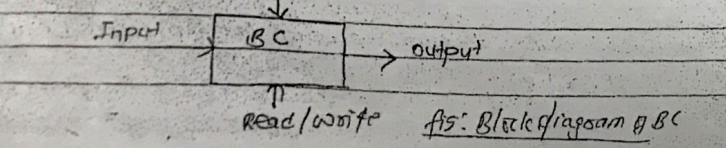


fig: Block diagram of BC

Random Access Memory (RAM):

A collection of storage cells together with additional information

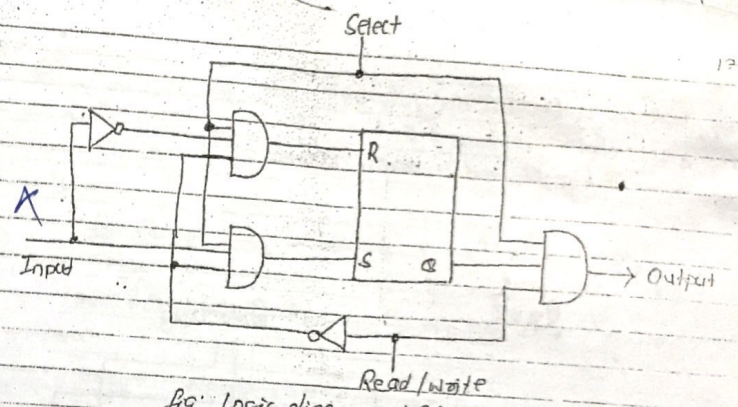


fig: Logic diagram of BC

Construction of 4x3 RAM:

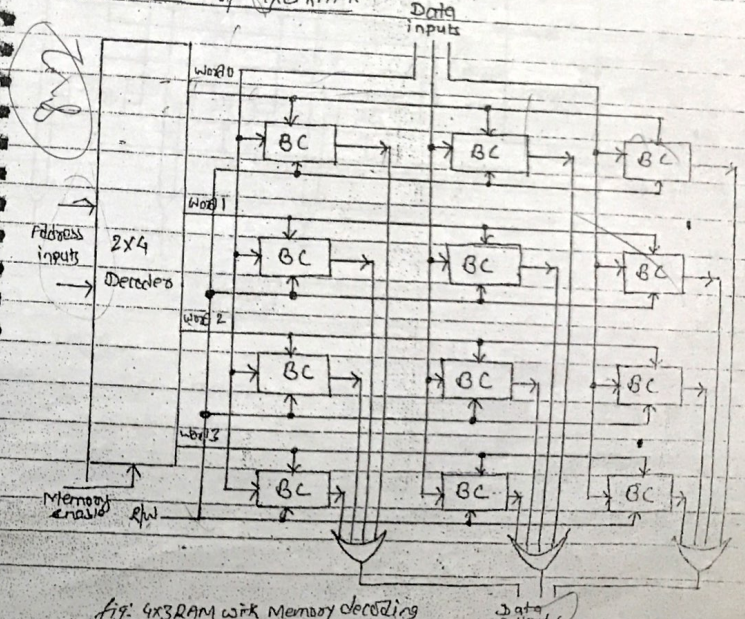


fig: 4x3 RAM with Memory decoding

Q. Design a RAM with following data
No. of address = 2
Data stored in each address = 3

Soln

Here $2^K = 2^1$

$\therefore K = 1$

$\& n = 3$

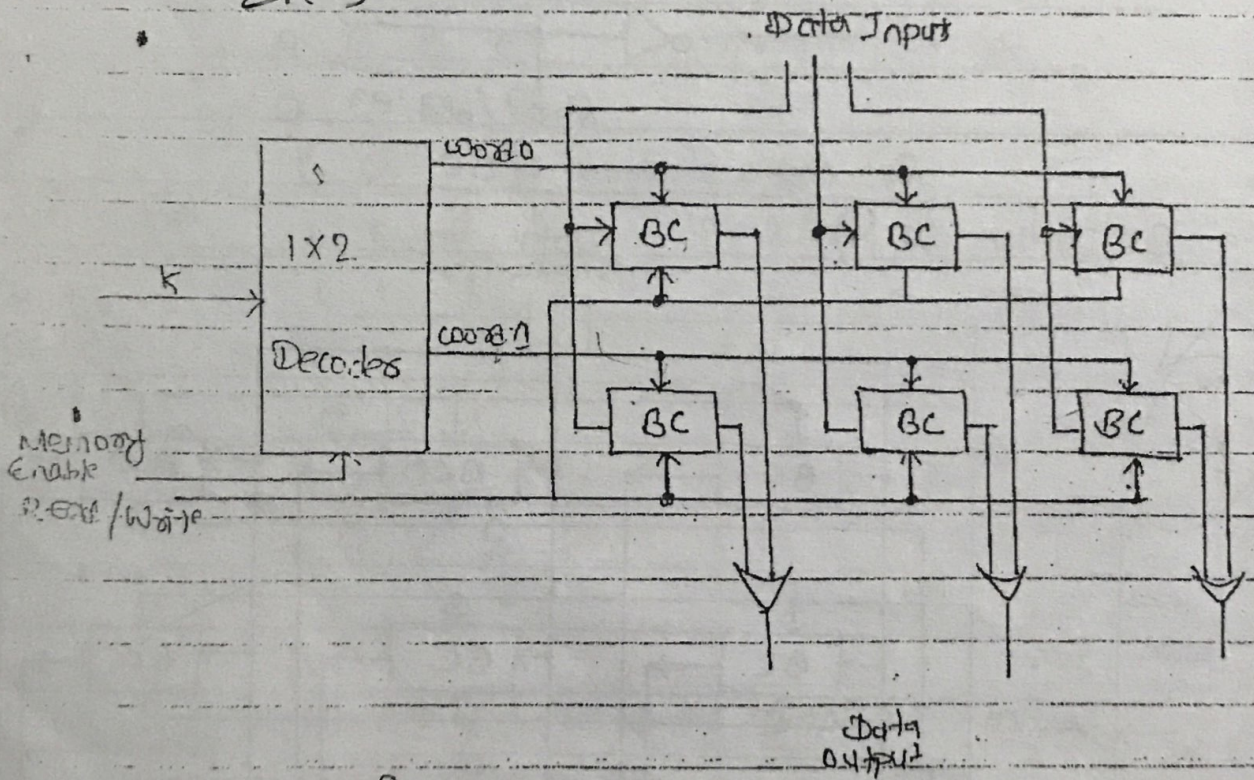


fig: 2X3 RAM

Error detection and correction:-

Due to complexity of memory, when the data is retrieved from the memory, sometimes error in bit occurs. To detect and correct those errors in bits, some error detection and correction codes are used. Those error detection ~~detec~~ code adds additional parity bits and store along with the data word in memory. The parity bits are checked after retriving the data from memory. If checked parity do not match with the parity present in the stored data then the error is detected.

For error correction, multiple parity bits are stored in the memory along with the data. On retriving the data, check bit is generated. Check bits are the parity over a group of bit in data words. The check bits are compared with stored parity bit. If they do not match, they generate a unique pattern called syndrome. That can be used to identify the bit in errors.
Eg: Error detecting code is Hamming code.

mming code:

It provides for the detection of bit errors.
It also identifies which bit is in error so that it can be corrected.
It is constructed for single error-correction as follows.

Q) A (7,4) hamming code is received as 1010111. Determine the correct code when even parity is there.

So,

Bit Position : 1 2 3 4 5 6 7

Msg : 1 0 1 0 1 1 1

Parity & Msg bits : P_1 P_2 M_1 P_3 M_2 M_3 M_4 [2^k → Parity bits]

Now

error = No. of Parity bits

i.e

E_3 = 4567 = 0111, for even parity, $E_3 = 1$

E_2 = 2367 = 0111, " " " , $E_2 = 1$

E_1 = 1357 = 1111, " " " , $E_1 = 0$

∴ Error position = $(110)_2 \rightarrow (6)_{10}$

i.e 6th position

Hence

Corrected msg = 1010101

Note:

If error position = 0, it means the received msg is error free.

Show how error is detected in parity system with an example.

Ans Let transmitted msg = 0010010

& Received " = 0110010

Now,

Bit Position :	1	2	3	4	5	6	7
	C	1	1	0	0	1	0
Parity & Msg :	P_1	P_2	M_1	P_3	M_2	M_3	M_4

Error = No of Parity bits. for odd parity

i.e $E_3 = 4567 = 0010$ $E_3 = 0$

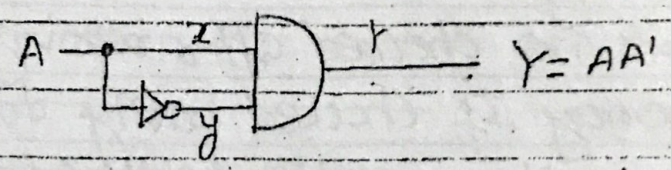
$E_2 = 2567 = 0010$, $E_2 = 1$

$E_1 = 1357 = 0100$

Output Hazard Races:

Hazards are unwanted switching transients that may appear at the output of a circuit because difference path exhibit different propagation delay.

Hazards in the combinational circuit is given below



We know that the output of circuit shown above should always be 0. Assume that A' momentarily changes from 0 to 1. In such condition, the input x receives 1 but due to the propagation delay of inverter y i/p receives 0 i/p after certain time as a result of which there is glitches in the output. This undesirable condition is called hazards.

Types of Hazards in combinational circuits are:

(i) Static Hazard:

It is the situation where when one input variable changes, the output changes momentarily before stabilizing to the correct value.

a) Static 0 Hazard: If the output goes suddenly to 1 and returns 0, where the o/p should have remained in 0 is known as static-0 hazard.

b) Static 1 Hazard: If the o/p momentarily goes to 0 where it should have remained in 1 is known as

Static-1 Hazard.

Dynamic Hazard:

It causes the output to change three or more times when it should change from 1 to 0 or 0 to 1.

Function Hazard:

It occurs when more than one input variables changes at the same time. It cannot be logically eliminated as the problem lies with the actual specification of the circuit. The only real way to avoid such problems is to restrict the changing of input variables so that only one input should change at any given time.