MEMORIES / UNIT-5 /

Sandeep Chaudhary
Electronics IInd Yr.

MEMORIES

Semiconductor memories are used to store large quantity of data and can be built by using a collection of flip-flops.

Memories are classified into two categories:

1). Static Memories.
2). Dynamic Memories.

Static memories use flip-flops as their basic constructional element whereas the dynamic memories use parasitic capacitances as their storage elements. Main considerations for the design of memory are rate of data access, high density, and low power dissipation.

CLASSIFICATION OF MEMORIES :-

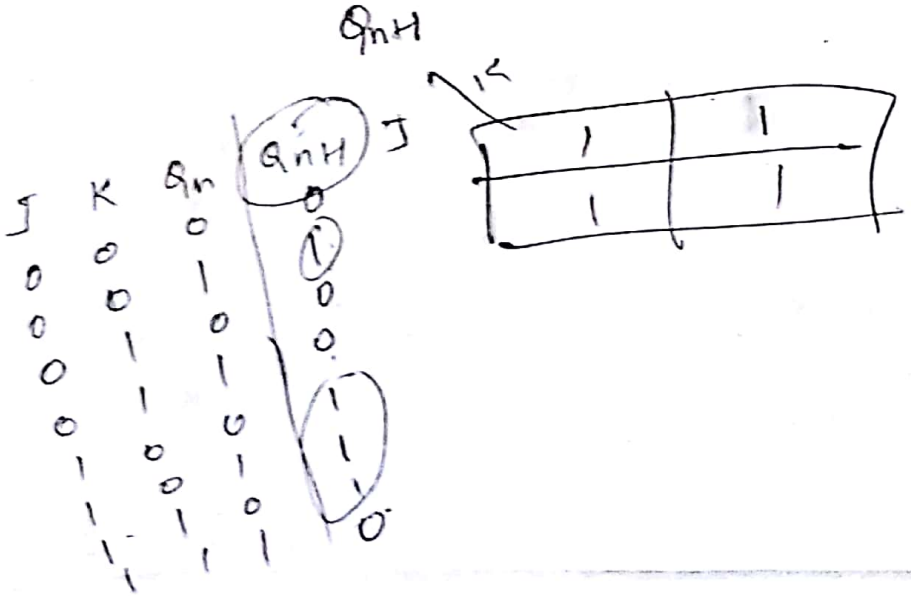
On the basis of working, memories can be classified into following categories:

1). Register memory.
2). Static and Dynamic memory.
3). Volatile and non-volatile memory.
4). Magnetic and semiconductor memory.

Register Memories :→

Register memory uses registers as their storage element. The register memories are further divided into two categories:

i). Main memory.
ii). Secondary or Auxillary memory.



MEMORIES

UNIT-(5)

MEMORIES

Sandeep Chaudhary
Electronics IInd Yr.

Semiconductor memories are used to store large quantity of data and can be built by using a collection of flip-flops.

Memories are classified into two categories:

1). Static Memories.
2). Dynamic Memories.

static memories use flip-flops as their basic constructional element whereas the dynamic memories use parasitic capacitances as their storage elements.

Main considerations for the design of memory are rate of data access, high density, and low power dissipation.

CLASSIFICATION OF MEMORIES :-

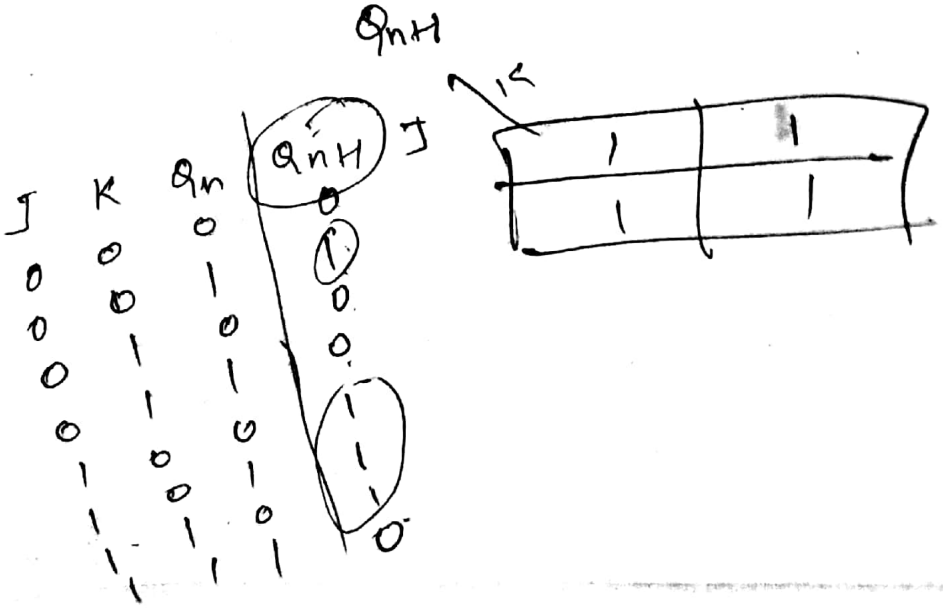
On the basis of working, memories can be classified into following categories:

1). Register memory.
2). Static and Dynamic memory.
3). Volatile and non-volatile memory
4). Magnetic and semiconductor memory.

Register Memories :→

Register memory uses registers as their storage element. The register memories are further divided into two categories:

i). Main memory.
ii). Secondary or Auxillary memory.



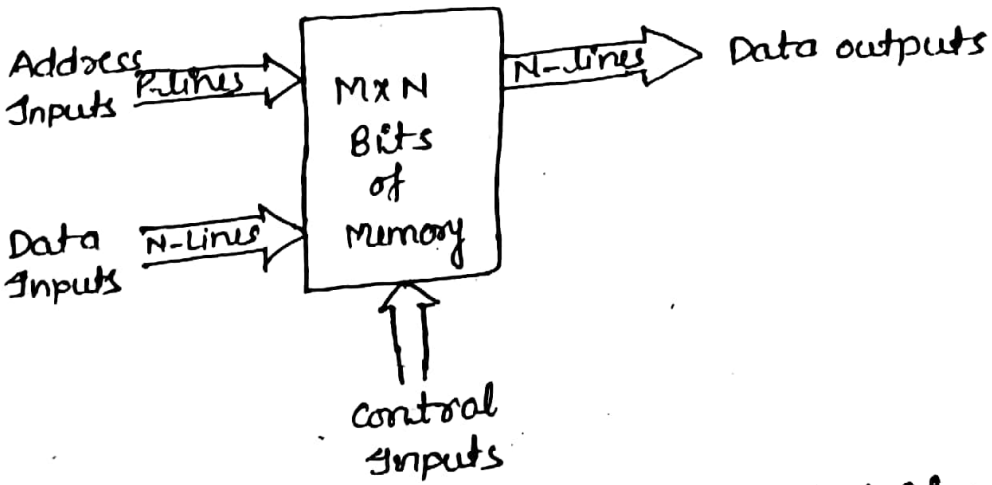
MEMORIES

Memory:-

system of a digital processing system, which provides data storage facility, is called memory. Initially magnetic memories were used, but with developements in the semiconductor technology, the semiconductor memories have become very popular.

The basic element of a semiconductor memory is a flip-flop. There are a number of locations in a memory chip, each location being meant for one word of digital information. The size of a memory chip is specified by two numbers M and N as M x N bits. The number 'M' specifies the number of locations available in the memory and the number 'N' specifies the number of bits at each location.

The block diagram of a memory device is shown in figure:



Each of the M locations of the memory is defined by a unique address and for accessing any one of the M location P inputs are required, where $2^P = M$. This set of lines is referred to as address inputs or address bus.

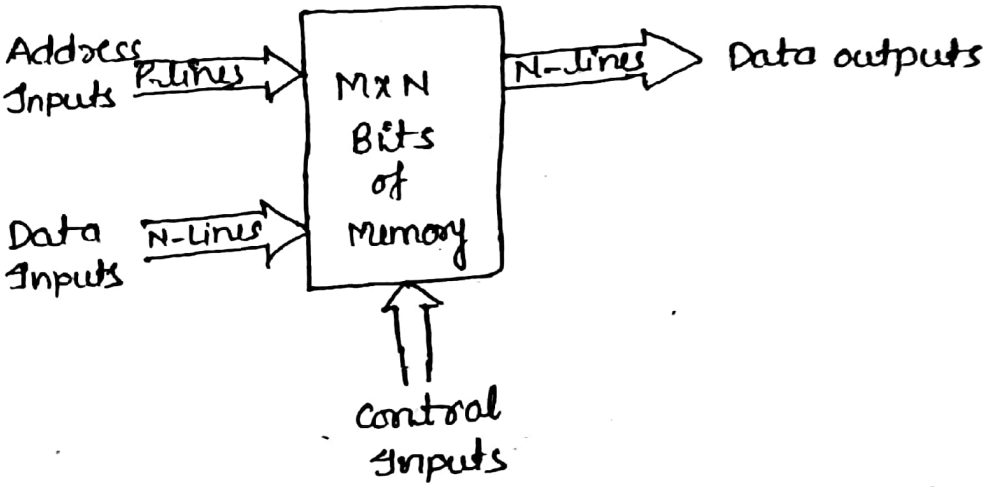
For example, consider a memory 16 words:

Since M=16, therefore, $2^P = M$ gives P=4, i.e. for selecting one out of 16 words, a 4-bit address is required. The address is specified as $A_3 A_2 A_1 A_0$. The address of each location is given in the table below:

system of a digital processing system, which provides data storage facility, is called memory. Initially magnetic memories were used, but with developements in the semiconductor technology, the semiconductor memories have become very popular.

The basic element of a semiconductor memory is a flip-flop. There are a number of locations in a memory chip, each location being meant for one word of digital information. The size of a memory chip is specified by two numbers M and N as M x N bits. The number 'M' specifies the number of locations available in the memory and the number 'N' specifies the number of bits at each location.

The block diagram of a memory device is shown in figure:

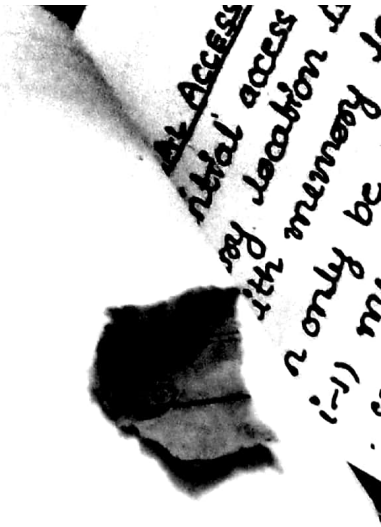


Each of the M locations of the memory is defined by a unique address and for accessing any one of the M location P inputs are required, where $2^P = M$. This set of lines is referred to as address inputs or address bus.

For example, consider a memory 16 words:

Since M=16, therefore, $2^P = M$ gives P=4, i.e. for selecting one out of 16 words, a 4-bit address is required. The address is specified as $A_3 A_2 A_1 A_0$. The address of each location is given in the table below:

Word Number	Binary Address			
	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1



Word Number	Binary Address			
	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Write Operation:

To write a word into the selected memory location requires

...TIAL ACCESS MEMORY :→

...ential access memory, is one in which a particular ...ory location is accessed sequentially. For example ...th memory location is to be accessed then it ... only be accessed after sequencing through previous ...-1) memory locations. Therefore the access time of ... sequential memory varies depending on the location ... be accessed.

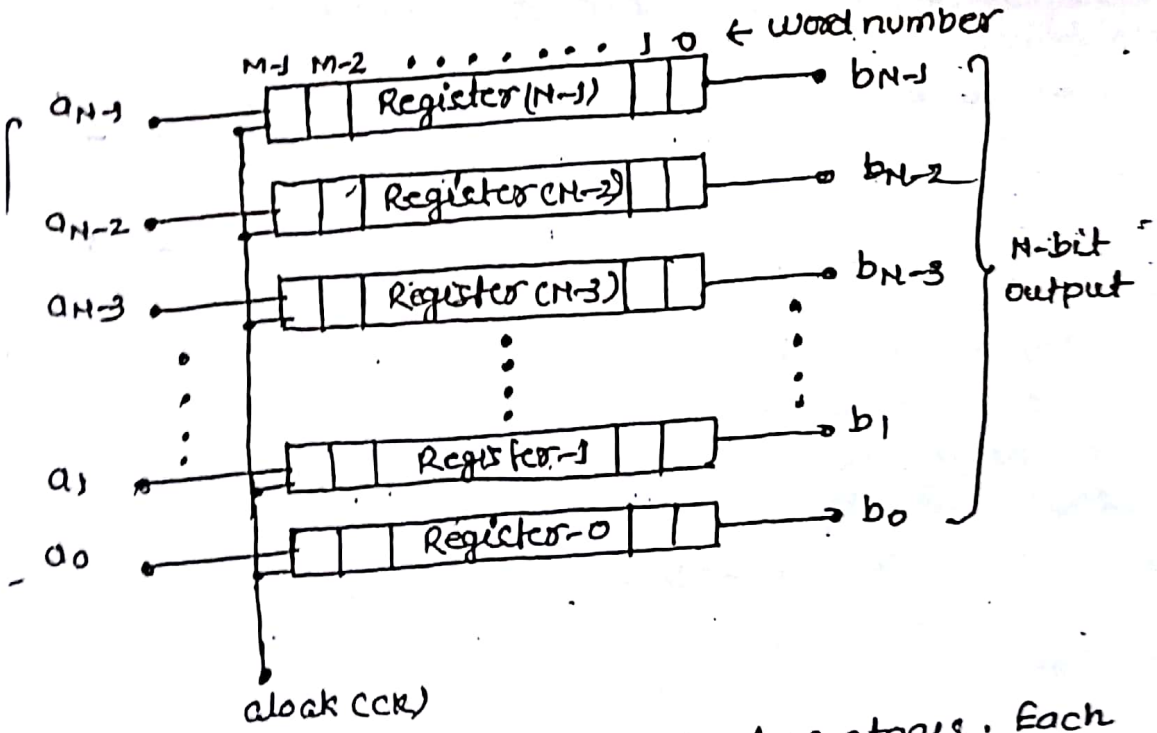
In sequential access memory, the access to a particular memory location is obtained by waiting until the desired location is reached.

There are two types of sequentially accessed memories:

(i). Shift Registers.

(ii). Charge coupled devices.

A sequential memory of size MXN is shown in figure;



It requires N shift registers, each of M stages. Each register holds one of the N-bits of each of the M-words. With each clock cycle, the bits in each register will advance towards the right by one-bit position, and the stored words will appear sequentially at the outputs of the registers.

This configuration is referred to as first-in-first-out (FIFO) sequential memory system, since the word

which is entered first will be the one read out first. The shift registers used in this configuration, must be capable of shifting the bits in either direction.

Shift registers can be either static or dynamic. In static registers, the contents of the memory location do not change with time as long as power is ON. On the other hand, in dynamic registers, it has to be refreshed at regular intervals since the information is stored in MOS capacitors.

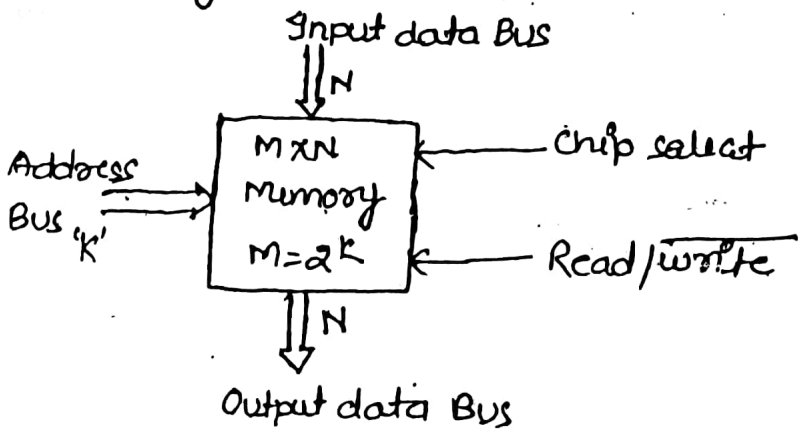
Example of sequential access memory is magnetic tape in which, the data is stored one word after another, in sequence.

RANDOM ACCESS MEMORY :→

A random access memory is one in which any location can be accessed in a random manner and thus has equal access time for all memory locations. It is a type of volatile memory in which both read and write operations can be performed. RAM is also called Read-write memory.

For reading from or writing to the cells, the data width or word size of the memory is N bits. The size, or capacity, of the memory refers to M words, so that the memory storage capacity is MN bits. To access each of the M words uniquely, an address size of k-bits ($2^k = M$) is required.

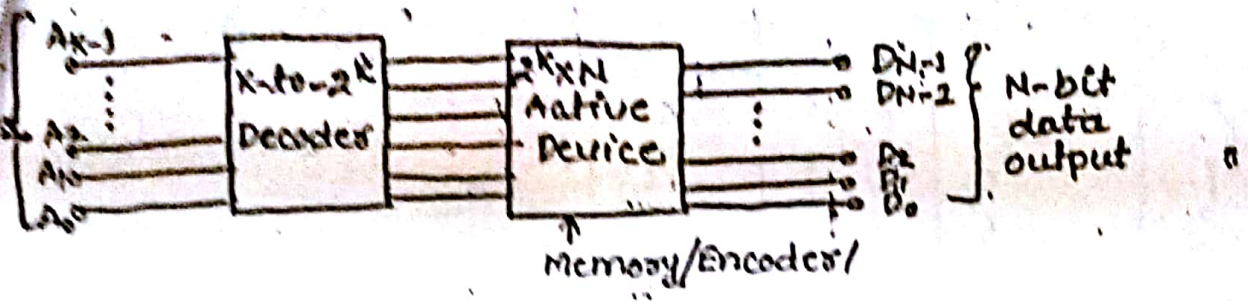
The signals and buses in a semiconductor RAM chip are shown in figure:



NLY MEMORY :→

ly memory is a type of memory that stores information that can only be read. The information is during manufacturing and requires various algorithms to alter the information stored. ROMs are used implementation of combinational circuits and for ing monitor programs.

ic ROM structure is shown in figure:



capacity of ROM is denoted by $2^k \times N$ bits, where 'k' denotes the number of address bits and 'N' denotes the number of data bits. A single k-to-2^k decoder is used for decoding the k-bit address. Each output bit represents a unique combination of k-input bits.

CLASSIFICATION :→

Read only memory can be classified into following categories:

1). Programmable ROM (PROM).
2). Erasable Programmable ROM (EPROM).
3). Electrically Erasable Programmable ROM (EEPROM).

PROM :→

Information stored in simple read only memory cannot be altered. In order provide alteration facility programmable ROMs are introduced.

PROMs can be programmed by users according to their requirements. In PROMs fusible links are included

combination of K-input bits.

CLASSIFICATION :→

Read only memory can be classified into following categories :

1). Programmable ROM (PROM).
2). Erasable Programmable ROM (EPROM).
3). Electrically Erasable Programmable ROM (EEPROM).

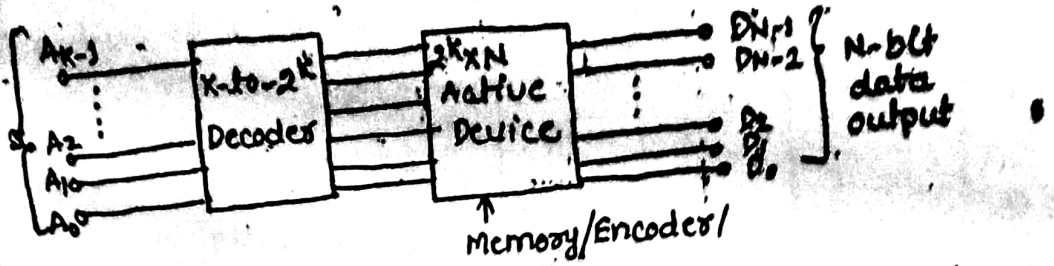
PROM :→

Information stored in simple read only memory cannot be altered. In order provide alteration facility programmable ROMs are introduced.

PROMs can be programmed by users according to their requirements. In PROMs fusible links are included to connect the Rows to the columns. Where no connection is needed, the link is burned out or open circuited by passing large current during programming. Once burned, the patterns stay permanent.

ly MEMORY :→

ly memory is a type of memory that stores
nation that can only be read. The information is
during manufacturing and requires various algori-
to alter the information stored. ROMs are used
implementation of combinational circuits and for
ning monitor programs.
asic ROM structure is shown in figure:



Capacity of ROM is denoted by $2^K \times N$ bits, where 'K' denotes the number of address bits and 'N' denotes the number of data bits. A single K-to-2^K decoder is used for decoding the K-bit address. Each output bit represents a unique combination of K-input bits.

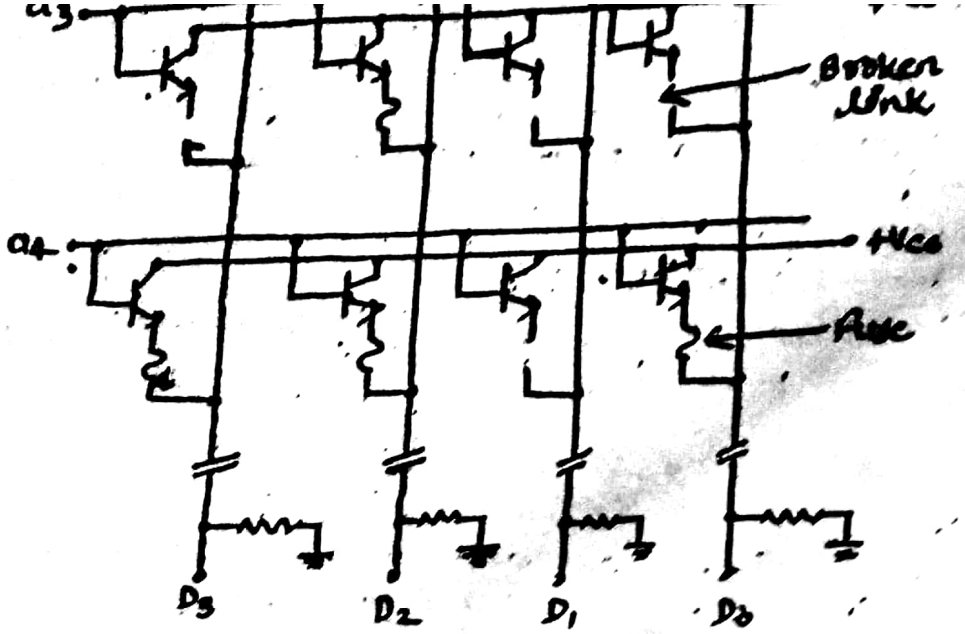
CLASSIFICATION :→

Read only memory can be classified into following categories:

1). Programmable ROM (PROM).
2). Erasable Programmable ROM (EPROM).
3). Electrically Erasable Programmable ROM (EEPROM).

PROM :→

Information stored in simple read only memory can-
not be altered. In order provide alteration facility
programmable ROMs are introduced.
PROMs can be programmed by users according to their
requirements. In PROMs fusible links are included
to connect the Rows to the columns. Where no
connection is needed, the link is burned out or
open circuited by passing large current during
programming. Once burned, the patterns stay permanent.



Reprogramming is achieved by burning out the polysilicon fusible links by passing current pulses in the range of 20 mA to 30 mA.

Advantages of PROM :→

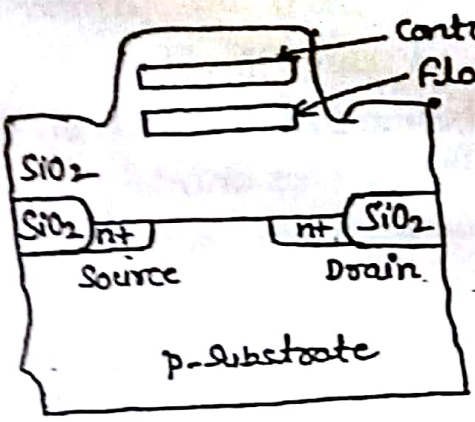
1). PROM uses polysilicon fuses which provides low resistance and hence less voltage drop at low operating currents.
2). PROMs, as the name suggests, can be programmed by users according to their requirements.
3). PROMs can be used in fast arithmatic operations.

Disadvantages of PROMs :→

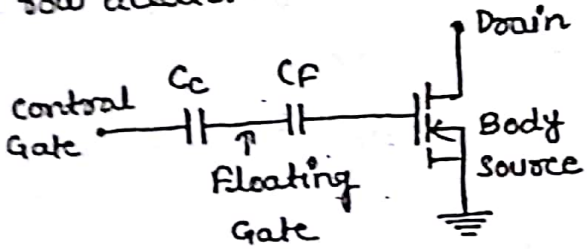
1). Consume large area.
2). Power requirements are higher.
3). PROMs are only Programmable once.

is are MOSFET devices that can be reprogrammed many times. Reprogramming can be done by illuminating storage cells with a strong ultraviolet light. To store different data patterns, all cells must first be returned to their unprogrammed state.

Each storage cell in an EPROM is a special MOSFET consisting of two gates. One gate is called the control or select gate and other is called the floating gate Structure of a floating gate MOSFET is shown in fig:



The first above the channel region is the floating gate and the second gate is normal control gate Floating gate is left unconnected whereas the control gate is connected to the row decoder.



In order to program the cell, the floating gate must be charged. Once the floating gate is charged, the charge remains trapped even after the programming voltages are removed because there is no discharge path available as the gate is surrounded by insulating oxide.

Advantages :-

1). EPROMs can be reprogrammed as many times as required.
2). EPROMs have higher reliability.
3). Low Power dissipation.

4). Higher density.
5). Lower cost.

Disadvantages :→

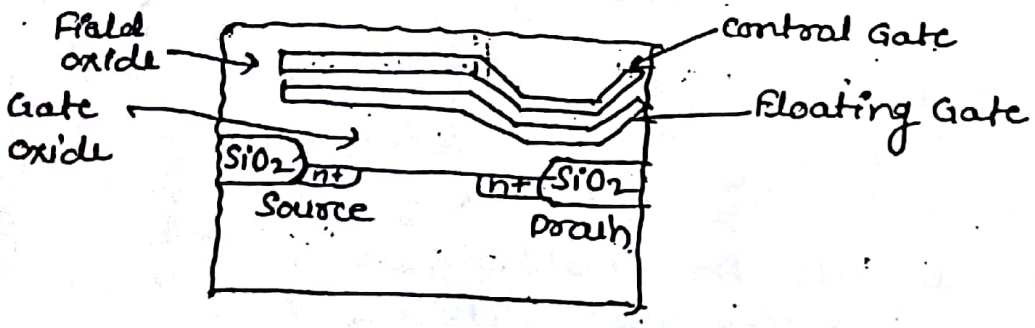
1). Ultraviolet radiations erase all the cells at once. Therefore, in order to reprogram a cell, all cells must be reprogrammed.
2). Lower speed of operation.
3). Before reprogramming, the EPROM must be removed from the circuit.

EEPROM :→

In EPROM, changes in the selected memory locations can only be made by erasing the entire memory before reprogramming. This drawback can be removed by using EEPROM.

EEPROM can be erased and programmed by the application of controlled electric pulses to the IC in the circuit, and thereby changes can be made in the selected memory locations without disturbing the data in other memory locations.

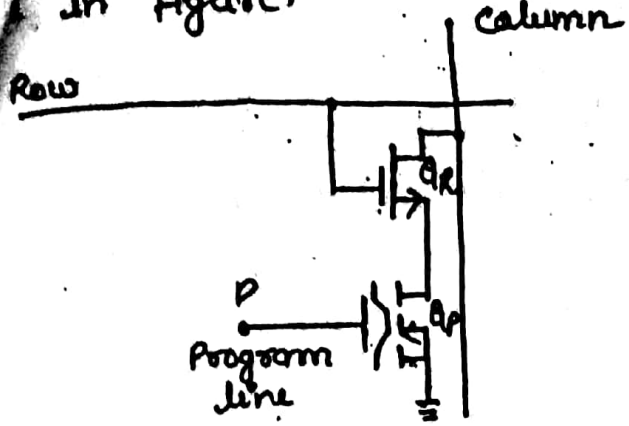
Structure of EEPROM MOSFET is shown below:



In order to program, the floating gate is charged by applying voltage to the control gate.

Programming of EEPROM is similar as of EPROM, the only difference is, in EEPROM, the floating gate is placed very close to the drain. The thickness of insulating oxide between the drain and the floating gate is extremely small such that a high electric field can discharge the floating gate.

tion example for EEPROM in ROM cell is in figure.



ring Read process, the gates of both Q_R and Q_p are logic high. If Q_p has a stored '1', the device ill be off and the column line will be logic igh through Q_R. For a stored '0', Q_p conducts and e column line will be logic low through Q_p.

Advantages :→

All the advantages and disadvantages of EPROM.

MEMORY EXPANSION (MEMORY BANKS) :→

Memory banks are used to increase the capacity of the memory chips when the capacity of the single chip is not sufficient to incorporate all the functions.

Capacity of memory chips can be increased by connecting more than one chips together.

If a particular memory chip is capable of storing M words with each word having N bits in it, then the size of the memory chip can be expanded in two ways!

i). By Expanding the word size (Increasing N)

ii). By Expanding the word capacity (Increasing M)

Expanding Word Size :→

A memory chip has the capacity 'M x N'. In order to increase the capacity the word size (N) can be increased. For example a 16 x 4 memory can be used

to obtain 16×8 by using word size expansion as explained below.

Available memory size = 16×4

Required memory size = 16×8

STEP(1): Determine the number of memory chips required as:

If the required word size = n

and Available word size = N

Then, the number of memory chips is an integer 'x' that satisfies

$x \geq n/N$, Here $n=8, N=4 \Rightarrow x=2$

Therefore, the number of memory chips of size 16×4 to obtain 16×8 chip are 2.

STEP(2): Determine the number of address lines required.

For this example number of locations = 16

Therefore the number of address lines is given by 2^n relation.

$(2^4 = 16) \Rightarrow n = 4$

The number of address lines = 4

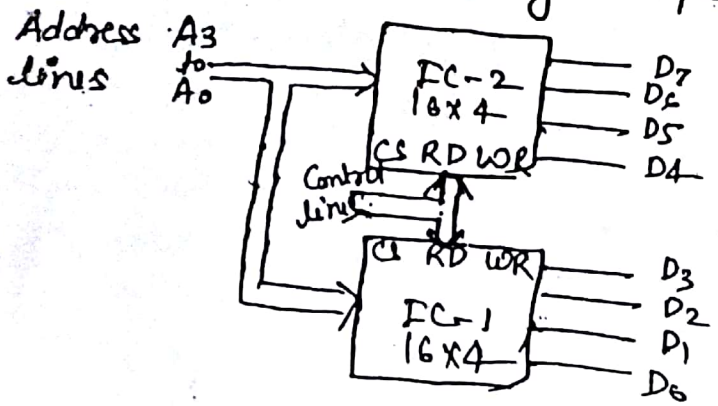
All the four address lines will be common to both the memory chips.

STEP(3): Determine the number of data lines.

Memory size to be obtained = 16×8

⇒ Required data lines = 8

STEP(4): Make connections by connecting the Read (RD), write (WR) and chip select inputs of both the 16×4 memory chips.



to obtain 16×8 by using word size expansion as explained below.

Available memory size = 16×4

Required memory size = 16×8

STEP(1): Determine the number of memory chips required as:

If the required word size = n

and Available word size = N

Then, the number of memory chips is an integer 'X' that satisfies

$X \geq n/N$, Here $n=8, N=4 \Rightarrow X=2$

Therefore, the number of memory chips of size 16×4 to obtain 16×8 chip are 2.

STEP(2): Determine the number of address lines required.

For this example number of locations = 16

Therefore the number of address lines is given by 2^n relation.

$(2^4 = 16) \Rightarrow n = 4$

The number of address lines = 4

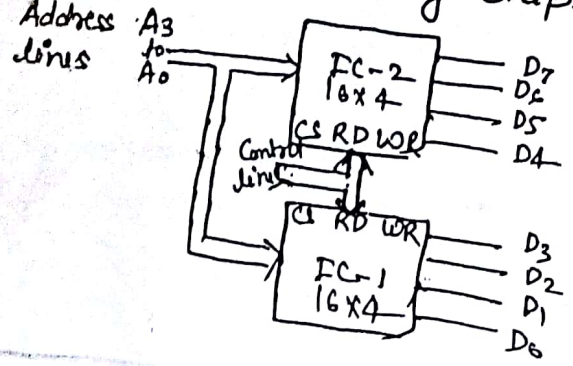
All the four address lines will be common to both the memory chips.

STEP(3): Determine the number of data lines.

Memory size to be obtained = 16×8

⇒ Required data lines = 8

STEP(4): Make connections by connecting the Read (RD), write (WR) and chip select inputs of both the 16×4 memory chips.



ing the word capacity :-

rder to increase the capacity of the memory chip
f size (M x N), its word capacity 'M' can also be
increased keeping the word size 'N' unchanged.

or example, a 16x4 memory can be used to obtain 32x4
y using word capacity expansion as explained below-

TEP(1) : Determine the number of memory chips required.

Required word capacity = 32 = m (say)
Available word capacity = 16 = M (say)

Then the number of memory chips is an integer 'x'
that satisfies:

$x \geq m/M$

Therefore, x=2 chips of capacity 16x4 are required to
obtain 32x4 memory chip.

TEP(2) : Determine the number of address lines required:

Here, number of locations = 32

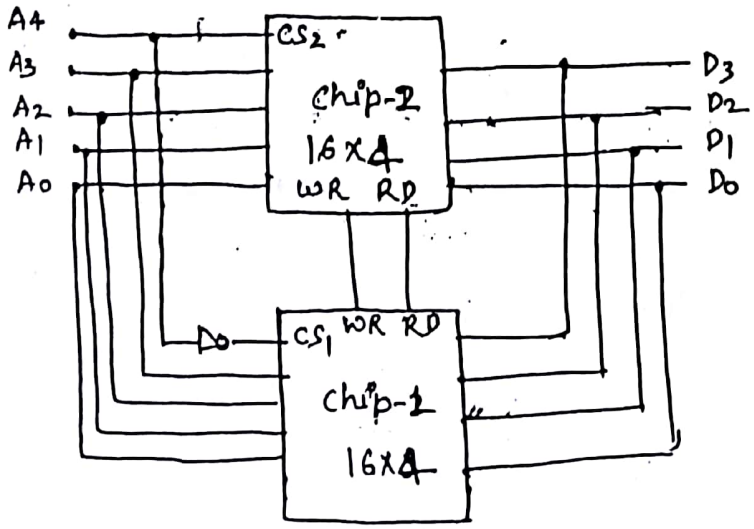
Therefore the no. of address lines = 5 ($2^5 = 32$)

Out of 5 address lines, the 4 LSB will be same for each
chip and the MSB will be used for the selection of
chips.

STEP(3) : Determine the number of data lines.

Since the word size is unchange therefore, the number
of data lines required = 8

STEP(4) : Make the connections as shown below:



...ing the word capacity :-

...order to increase the capacity of the memory chip of size (M×N), its word capacity 'M' can also be increased keeping the word size 'N' unchanged.

For example, a 16×4 memory can be used to obtain 32×4 by using word capacity expansion as explained below-

STEP<1> : Determine the number of memory chips required.

Required word capacity = 32 = m (say)

Available word capacity = 16 = M (say)

Then the number of memory chips is an integer 'x' that satisfies:

$x \geq m/M$

Therefore, x = 2 chips of capacity 16×4 are required to obtain 32×4 memory chip.

STEP<2> : Determine the number of address lines required :-

Here, number of locations = 32

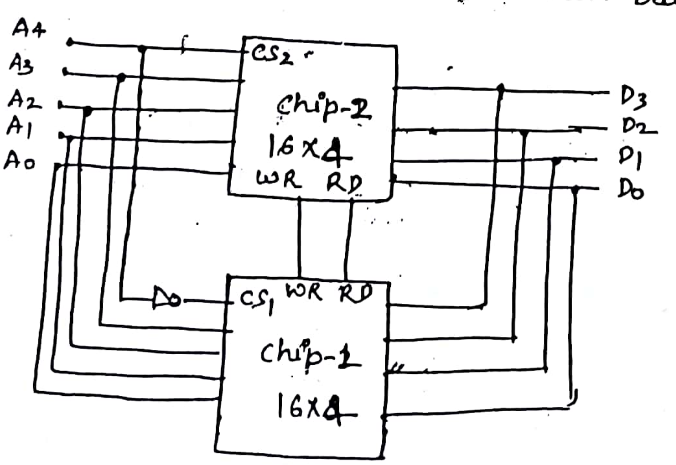
Therefore the no. of address lines = 5 ($2^5 = 32$)

Out of 5 address lines, the 4 LSB will be same for each chip and the MSB will be used for the selection of chips.

STEP<3> : Determine the number of data lines.

Since the word size is unchange therefore, the number of data lines required = 8

STEP<4> : Make the connections as shown below:



Example: How many 256x8 memory chips are required to obtain 2048x8 memory chip? Show connections.

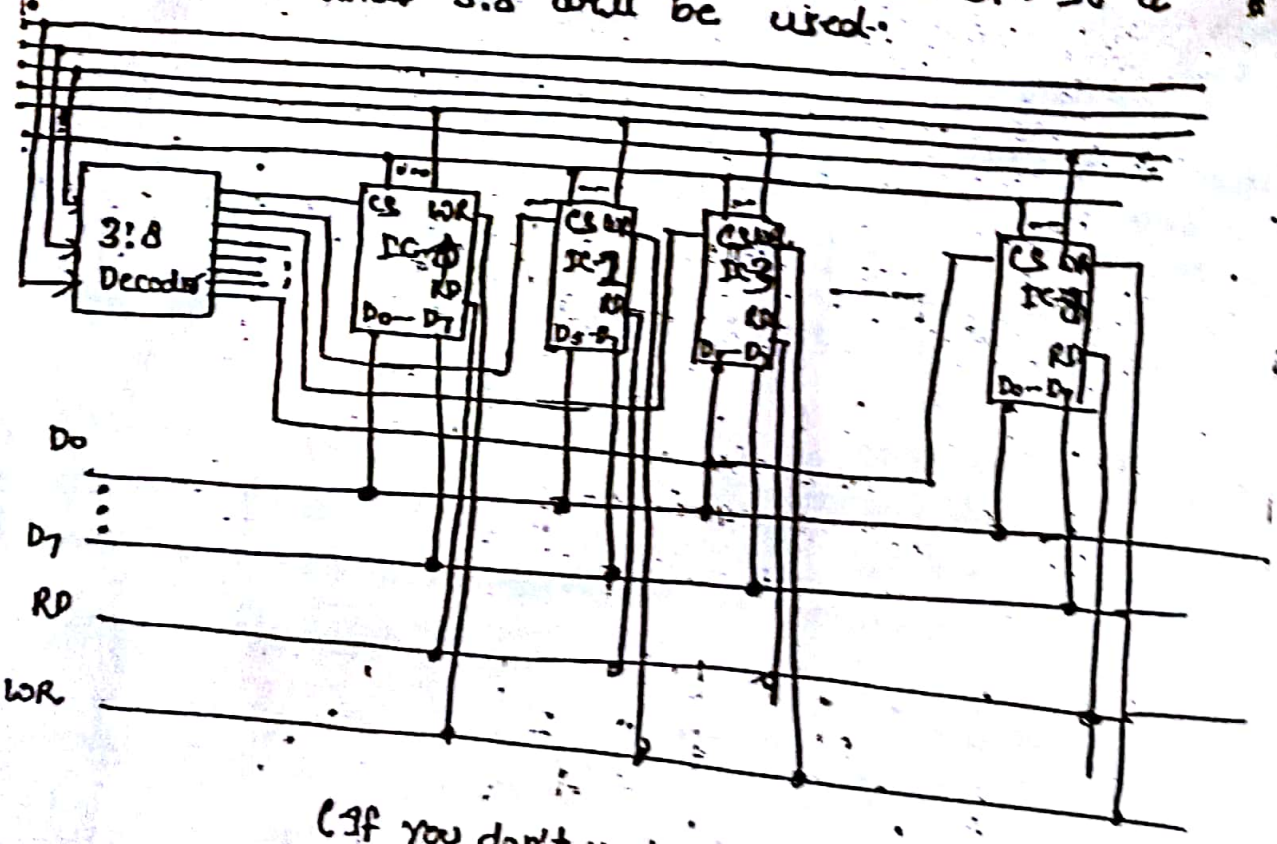
Hint: chips required = 2048/256 = 8

No. of address lines = 11. (2^{11} = 2048)

No of data lines = 8.

Connections: → Since all 11 address lines are required, if we write combinations of 11 inputs it will be 2048. Out of these 2048 combinations, only 3 MSB (i.e. A_{10}, A_9, A_7) will be different and rest are same. ——— (Keep this in mind) ———

→ Since 8 chips are required, so there will be 8 chip select inputs. 3 MSB's will produce 8 combinations and these will be used for chip select. So a 3:8 decoder that 3:8 will be used.

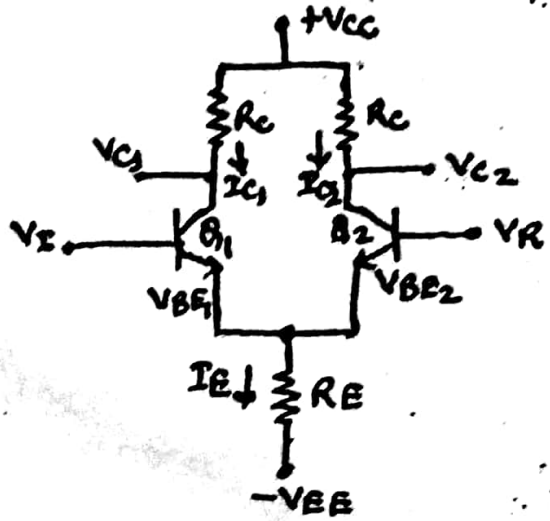


(If you don't understand this come anytime but before coming to my cabin do give me a call)

EMITTER COUPLED LOGIC :→

It is a current mode logic family. In this, the logic functions are realized by steering current from one part of the ckt to another part. ECL family has the fastest switching speed and dissipates high power.

Basic circuit of ECL logic family is the emitter coupled differential amplifier circuit as shown below:



Case(I): When $V_I = V_R$

$\Rightarrow V_{BE_1} = V_{BE_2} \Rightarrow \Delta V = 0$

Therefore from equation (6)

$$I_{C_1} \simeq I_{C_2}$$

This state is called operating state.

Case(II): When $V_I > V_R$

$\Rightarrow \Delta V$ will be positive.

$\Rightarrow I_{C_1} > I_{C_2} \Rightarrow V_{C_1} < V_{C_2}$

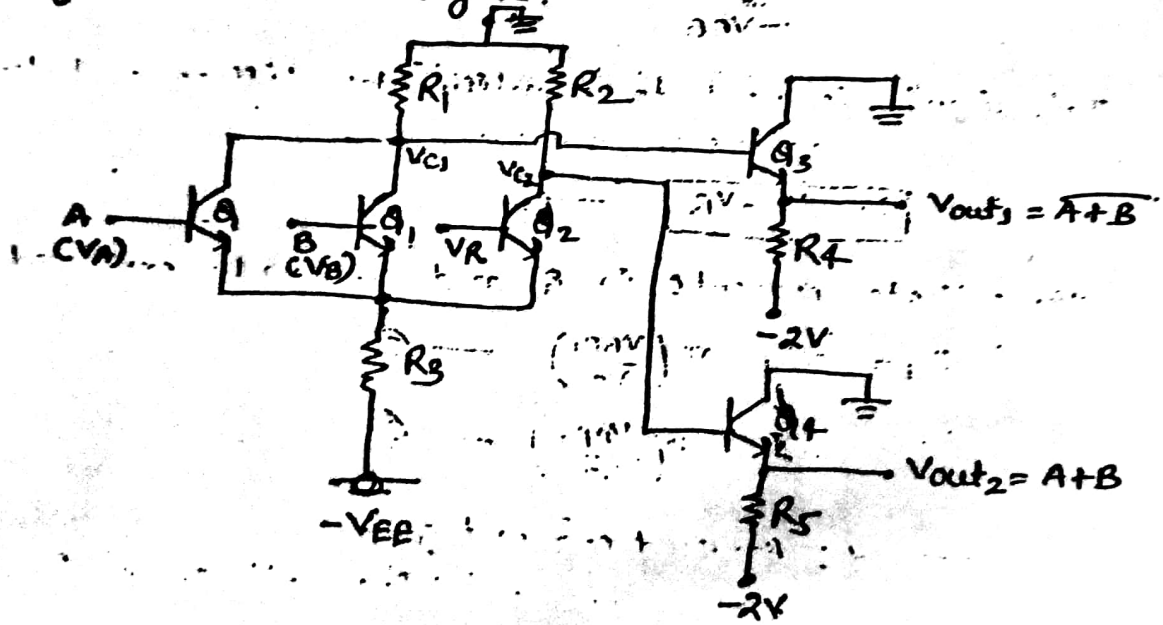
Case(III): When $V_I < V_R$

$\Rightarrow \Delta V$ is negative.

$\Rightarrow I_{C_1} < I_{C_2} \Rightarrow V_{C_1} > V_{C_2}$

ECL OR/NOR Gate :→

The ECL circuit that realizes the logic function of OR/NOR gate is shown in figure:



TRUTH TABLE :→

I/Ps		Transistors					outputs	
A	B	Q	Q_1	Q_2	Q_3	Q_4	V_{out_1}	V_{out_2}
0	0	OFF	OFF	ON	ON	OFF	1	0
0	1	OFF	ON	OFF	OFF	ON	0	1
1	0	ON	OFF	OFF	OFF	ON	0	1
1	1	ON	ON	OFF	OFF	ON	0	1

Operation :→

If V_A or V_B is above V_R (reference voltage), Q_1 or Q_2 conducts and brings voltage V_{C_1} below zero (Logic 'low'), and therefore Q_3 goes into "off state" and hence $V_{out_1} = 0$. Since reference voltage V_R is lesser than input voltages, Q_2 is cut off and this brings $V_{C_2} = V_{CC}$ (Logic High). High V_{C_2} drives Q_4 into ON state and therefore $V_{out_2} = 1$.

If both the input are lower than the reference voltage, Q_1 and Q_2 are turned off and Q_2 conducts. V_{C_1}, therefore, is at V_{CC} (Logic '1') and V_{C_2} is at logic '0'. High V_{C_1} drives Q_3 into ON state and therefore $V_{out_1} = 1$. Low V_{C_2} cuts Q_4 off and hence $V_{out_2} = 0$.

Therefore if output is taken at V_{out_1}, the circuit behaves as a NOR gate. Whereas, if it is taken at V_{out_2}, the circuit behaves as OR gate.

CHARACTERISTICS OF ECL :-

1). Very high switching speed.
2). Less reliable because of less noise margin.
3). No need for inverters.
4). Small fan out.
5). High power dissipation.
6). No noise spikes.

Imp

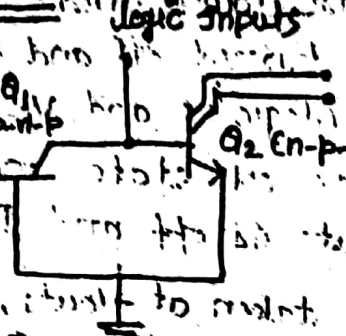
Integrated Injection logic (IIL) :→

Integrated injection logic is a type of current mode logic family in which realization of logic functions is done by steering the current from one part of the circuit to another part.

IIL dissipates lowest power among logic families.

It uses both the n-p-n and p-n-p transistors to implement logic functions.

IIL Inverter :→



operation :→

In IIL inverter, the base of Q_1 and emitter of Q_2 are internally merged and collector of Q_1 and base of Q_2 are merged.

The p-n-p transistor acts as a current source and the n-p-n transistor acts as an inverter. Most of the current leaving the emitter is injected into the base of Q_2, and hence the emitter of Q_1 is called injector.

Case (I): When input is logic low (0):

Logic low input provides negligible current at the base of transistor Q_2 and, hence, it goes into cut off state. Q_1 injects no current into the base of Q_2. Therefore, the output is logic high as no path is available towards ground.

Case (II):- When input is at logic High (1):

Logic high input at the base of Q_2 allows the injector current to hold Q_2 ON and hence its output goes logic Low.

Imp

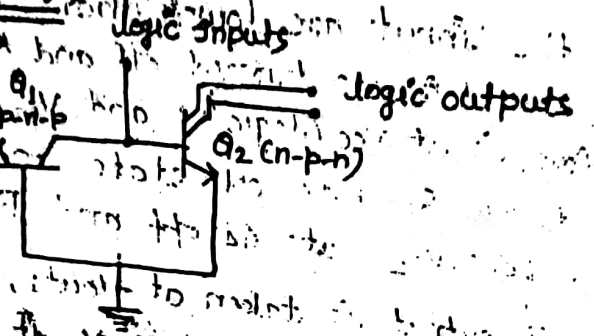
Integrated Injection logic (IIL) :→

Integrated injection logic is a type of current mode logic family in which realization of logic functions is done by steering the current from one part of the circuit to another part.

IIL dissipates lowest power among logic families.

It uses both the n-p-n and p-n-p transistors to implement logic functions.

IIL Inverter :→



Operation :→

In IIL inverter, the base of Q_1 and emitter of Q_2 are internally merged and collector of Q_1 and base of Q_2 are merged.

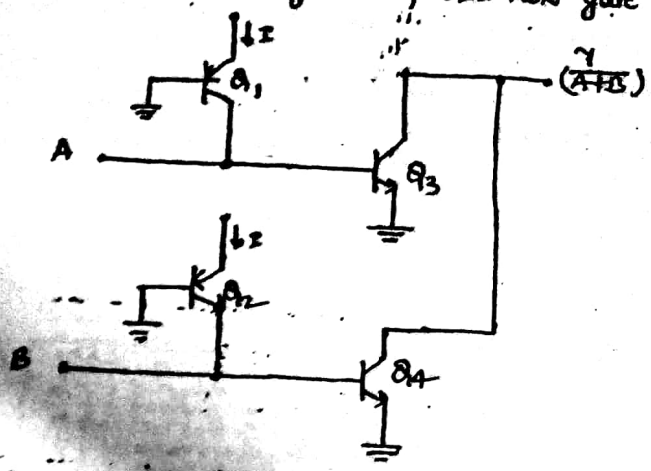
The p-n-p transistor acts as a current source and the n-p-n transistor acts as an inverter. Most of the current leaving the emitter is injected into the base of Q_2, and hence the emitter of Q_1 is called injector.

Case (I): When input is logic low (0):
Logic low input provides negligible current at the base of transistor Q_2 and, hence, it goes into cut off state. Q_1 injects no current into the base of Q_2. Therefore, the output is logic high as no path is available towards ground.

Case (II): When input is at logic High (1):
Logic high input at the base of Q_2 allows the injector current to hold Q_2 ON and hence its output goes logic Low.

IIL NOR Gate :→

The circuit diagram of IIL NOR gate is shown below



Operation :→

When either of both the inputs are high, one or both Q_3 and Q_4 are ON and, therefore, output Y goes LOW.

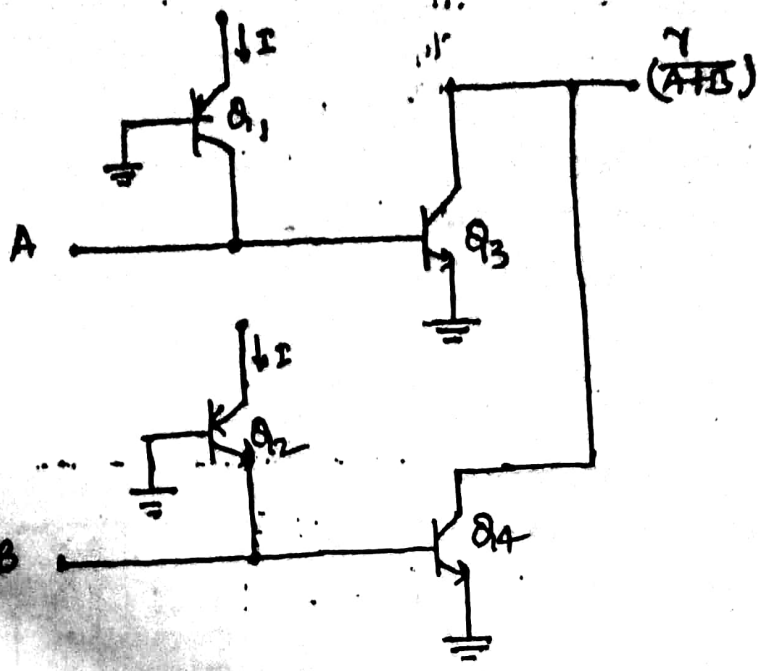
When both the inputs are low, both Q_3 and Q_4 are off and, therefore, output Y goes HIGH.

CHARACTERISTICS OF IIL :-

1). Simple circuit configuration.
2). Several switching transistors are combined.
3). Lowest power dissipation.
4). High speed.
5). High Packaging density.
6). Requires less chip area.

IIL NOR Gate :→

The circuit diagram of IIL NOR gate is shown below



Operation :→

When either of both the inputs are high, one or both Q_3 and Q_4 are ON and, therefore, output Y goes LOW.

When both the inputs are low, both Q_3 and Q_4 are off and, therefore, output Y goes HIGH.

CHARACTERISTICS OF IIL :-

1). Simple circuit configuration.
2). Several switching transistors are combined.
3). Lowest power dissipation.
4). High speed.
5). High Packaging density.
6). Requires less chip area.

Logic Family	Characteristic				
	Propagation Delay (ns)	Average Power (mW) Dissipation	Noise Margin (Volt)	Cost	Fan out
TTL	10	10 mW	0.4	High	12
ECL	2	25 mW	0.25	Highest	16
I^2L	0.7 to 20	0.1 to 100	0.4	lowest	12
NMOS	1	0.1 mW	2	High	12
CMOS	1	0.002 mW	2.5	High	12
BiCMOS	1	0.002 mW	2.5	High	12

organized as a 64,536 × 1 chip. The operation of the chip is quite similar to the 41[illegible] except that it requires only a single 5 Vdc power supply. Note that these, or comparable, chips are available from a number of different manufacturers, but the part numbers may differ even though the chips are compatible. For instance, the 64K DRAM (65,536 × 1) is available under the following part numbers: Advanced Micro Devices, AM9064; Fairchild, F4164; Intel, 2164; Mostek, MK4564; Motorola, MCM6665; and Texas Instruments, TMS4164.

12-8 MEMORY CELLS

A memory cell is the basic unit for storing a single bit of information in a memory. The cells used to construct semiconductor memories are usually flip-flops designed by using either bipolar or MOS transistors or a charge storage circuit that uses MOS transistors. The intent here is not to study these memory cells with the detail required to design these circuits. (Such subjects are more appropriately covered elsewhere.) Rather, we are interested primarily in understanding at least one example that illustrates exactly how a memory cell operates. Secondly, such an in-depth understanding leads to a clear concept of the circuit requirements, as well as an appreciation for the operating limitations imposed on a memory chip. It is not possible to discuss all the different circuit designs, but the basic circuits covered here illustrate the principles used in most memory cells. For each circuit discussed, we assume the signal voltage levels to be high = $+V_{CC}$ and low = 0.0 Vdc.

A memory cell used in a TTL static RAM along with a sense amplifier and a write amplifier, are shown in Fig. 12–21. Two bipolar junction transistors (Q_1 and Q_2), cross-coupled to form a simple latch and capable of storing 1 bit of information, serve as the memory element. The only unusual aspect is that each transistor has three emitter leads. Power supply connections necessary for the circuits are $+V_{CC}$ and ground (GND).

If either the ROW or COLUMN select line is low (GND), the cell is disabled (deselected). In order to select a cell, the ROW and COLUMN lines must both be high ($+V_{cc}$). When selected, a data bit can be stored in the cell (write), or the contents of the cell can be sensed (read out). Here's how it works.

When the ROW and COLUMN select lines are high, the emitters connected to these lines are reverse-biased, and the cell behaves as a simple latch. So, one of the transistors is on, and one is off. The transistor that is on conducts current out through its emitter, down through a sense resistor R, and into the base of the sense amplifier transistor, turning it on. In the cell, there is no current coming out of the emitter of the transistor that is off, and thus the sense amplifier transistor on that side of the cell is off. The net result is that the sense amplifier transistors "mimic" the transistors in the memory cell. Therefore, when selected, the contents of a cell is immediately available at the sense amplifier data outputs. For instance, if a 1 is stored in the cell, Q_1 is on and Q_2 is off. When selected, DATA OUT will be high (a 1), and its complement, $\overline{\text{DATA OUT}}$, will be low (a 0).

The WRITE amplifier is used to store information in the cell when its W input is held high. When W is low, the amplifier is disabled and its outputs are disconnected from the sense resistors R_s. There are numerous configurations, but the basic requirements are that WRITE 1 goes high and WRITE 0 goes low whenever DATA IN is high; conversely, WRITE 0 goes high and WRITE 1 goes low whenever DATA IN is low. To store a bit in the cell, it is first selected by taking both ROW and COLUMN high. Then a high at the

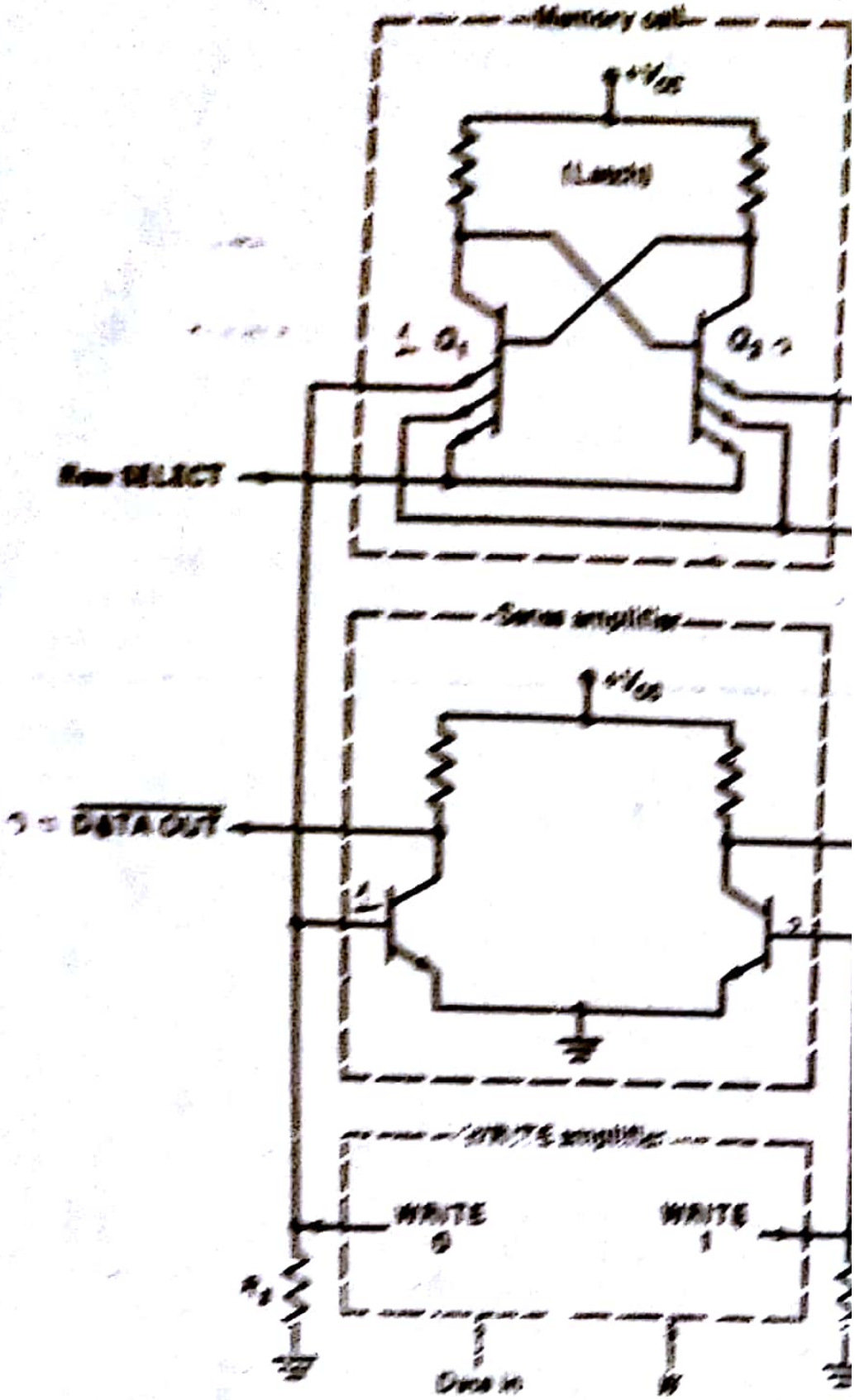


FIG. 12–21

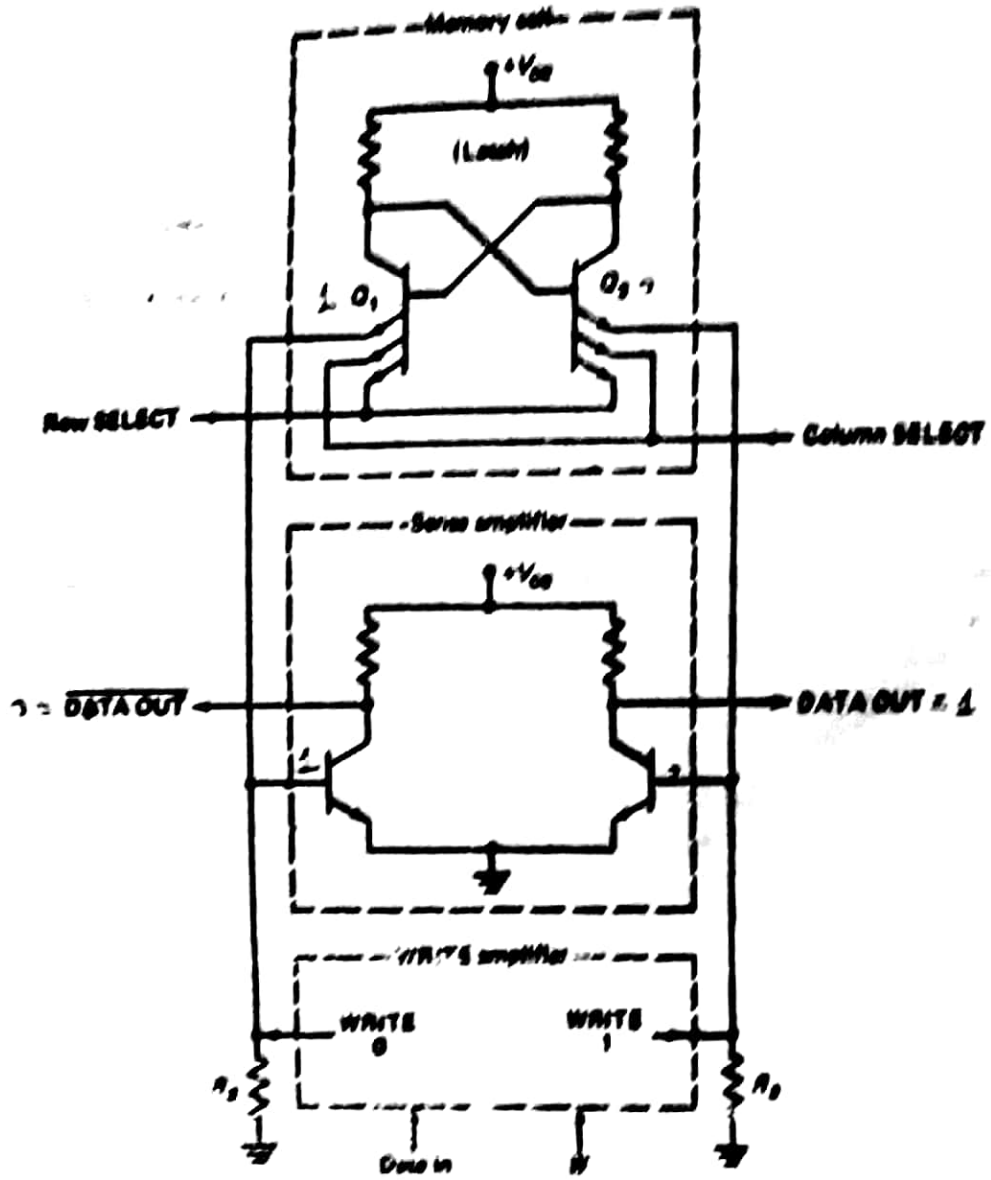


FIG. 12–21

DATA IN of the write amplifier will cause WRITE 1 to go high and WRITE 0 to go low. Thus all three of the emitters of Q_2 will be high, forcing Q_2 off and Q_1 on, and the latch will store a 1. Conversely, a 0 at the DATA IN will take all three of the emitters of Q_1 high, forcing Q_1 OFF and Q_2 on, storing a 0 in the latch. This is the write cycle.

In a static n-type-channel metal oxide semiconductor (NMOS) memory, the memory cell is constructed by using two NMOS inverters, cross-coupled to form a simple latch. In

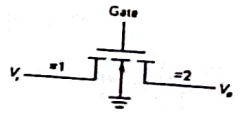


FIG. 12–22 NMOS transmission gate

the drain leads may be interchanged with no difference in performance. Since this is the case, let's simply call these two leads 1 and 2. Let lead 1 be the input and lead 2 the output. The GATE lead is the control. The voltage signals are either 0.0 Vdc or $+V_{CC}$. Here's how it works.

If the GATE is held low (at 0.0 Vdc), the transistor is off, and the input and output terminals are isolated from one another. There is no transmission of signal through the gate.

If the GATE is high ($+V_{CC}$), and the voltage at lead 1 is low, the transistor is ON, and current will flow through the transistor until $V_o = V_i = 0.0$ Vdc.

If the GATE is high and the voltage at lead 1 is also high, the transistor will be on if V_o is low, and current will flow through the transistor until $V_o = V_i = +V_{CC}$. Or, if V_o is also high, the transistor will remain off since $V_o = V_i = +V_{CC}$.

To summarize, the transmission gate is disabled whenever the GATE is low, and the input is isolated from the output; thus it acts like an open switch. The transmission gate is enabled whenever the GATE is high, and the output will equal the input, $V_o = V_i$; thus it acts like a closed switch.

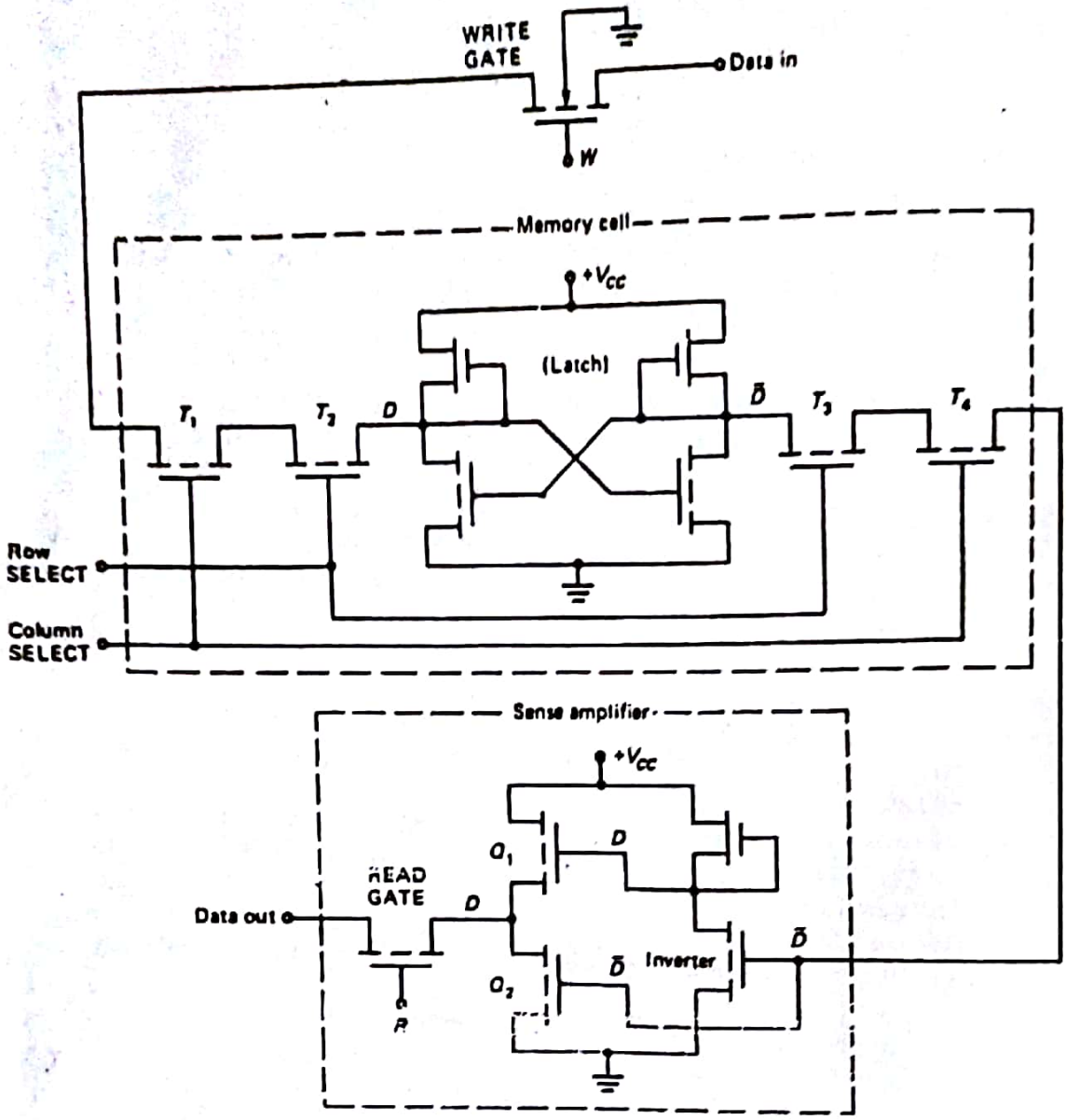
The circuit in Fig. 12–23 is typical of the memory cell and sense amplifier used in an NMOS static RAM. The memory cell consists of two NMOS inverters, cross-coupled to form a latch. There are four transmission gates, T_1, T_2, T_3, and T_4, used to select the cell. When both ROW and COLUMN are high, the cell is selected and its contents can be read out by the sense amplifier, or a data bit can be stored in the cell by using the WRITE gate (W).

Let's assume the cell is selected by holding ROW and COLUMN both high. Then the two transmission gates T_1 and T_2 are both enabled. (They act like closed switches.) If W is held high, the WRITE transmission gate is also enabled and a data bit at the DATA IN (either a high or a low) will be connected directly to the latch at D. If DATA IN is high, a 1 will be stored, since the latch will stabilize with D high; conversely, a low at DATA IN will store a 0, since the latch will stabilize with D low. This is the WRITE cycle.

Again, let's assume that the cell is selected with both ROW and COLUMN high. The two transmission gates T_3 and T_4 are enabled. (They act like closed switches.) Whatever data bit is stored in the latch at D, its complement appears at $\overline{D}$ and is coupled by the transmission gates T_3 and T_4 out to the sense amplifier. If the READ gate is enabled (R is high), the data bit stored in the latch D will appear at the DATA OUT. This is a READ cycle.

The first two transistors in the sense amplifier form an inverter whose output is D. The two transistors at the sense amplifier output (Q_1 and Q_2) have input signals of D and $\overline{D}$ with the result that one of them is always on and the other one is always off. For instance, if the data bit stored in the latch is a 1, then D = high. So, Q_1 will be on and Q_2

22



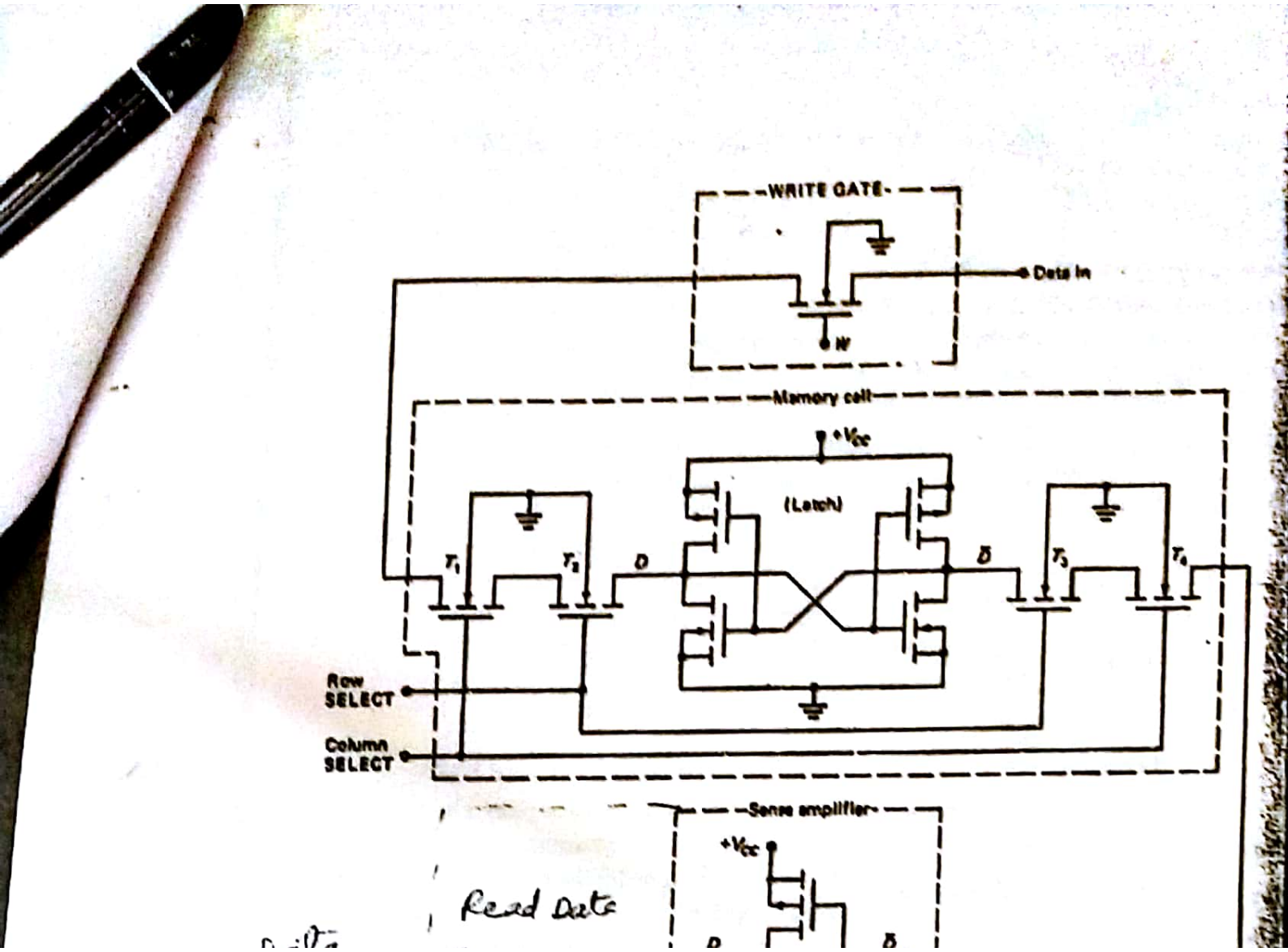
Note: All transistors are NMOS with their substrate leads connected to ground (substrate leads not shown)

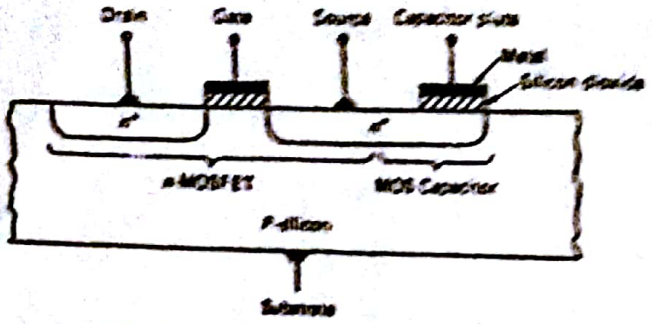
FIG. 12–23 Memory cell for an NMOS static RAM

will be off, and the DATA OUT will be high (1). Conversely, if a 0 is stored in the latch, D = low. So, Q_1 will be off, Q_2 will be on, and the DATA OUT will be low (0).

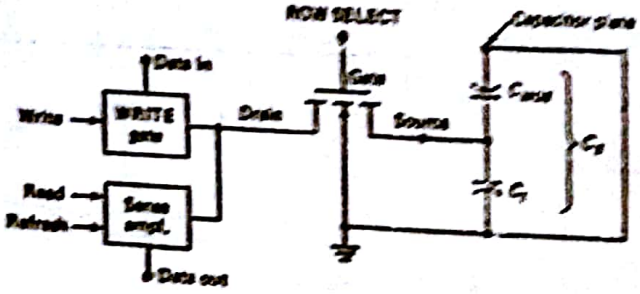
The circuit in Fig. 12–24 is typical of the memory cell and sense amplifier used in a CMOS static RAM. The memory cell consists of two CMOS inverters, cross-coupled to form a simple latch. There are four transmission gates, T_1, T_2, T_3, and T_4, used to select the cell. When both ROW and COLUMN are high, the cell is selected and its contents can be read by using the sense amplifier, or a data bit can be stored in the cell by use of the WRITE gate (W).

The WRITE cycle for this circuit is exactly like the WRITE cycle for the NMOS circuit described previously. The cell is selected by holding both ROW and COLUMN





(a) Fabrication of a one-transistor dynamic memory cell



(b) Circuit diagram

FIG. 12–25 One transistor dynamic memory cell

A data bit stored in a dynamic memory cell is in the form of a charge stored on a capacitor. Since this charge will leak off with time, the capacitor must periodically be recharged. Furthermore, the process of sensing, or reading out, the contents of the dynamic cell may dissipate the stored charge, resulting in a loss of the stored data bit. This is referred to as *destructive readout*, a problem not encountered with static memory cells. Even though dynamic memories require additional circuitry and timing signals to deal with refreshing as well as the destructive readout problem, the advantage of greater memory capacity on a single chip is generally worth the effort.

Let's take a look at a one-transistor MOS dynamic memory cell as shown in Fig. 12–25. The cell consists of an NMOS transistor and a MOS capacitor fabricated on a chip as shown. If the capacitor PLATE lead and the substrate are connected to ground, the storage capacitance C_s used to store the charge for a bit consists of the MOS capacitor C_{MOS} in parallel with the transistor junction capacitance C_t. A 1 is stored in the cell if the capacitor C_s is charged, and a 0 is stored if C_s is discharged.

The transistor itself behaves essentially like a transmission gate. The ROW select will enable all the transistors in an entire row in the memory. Then on a READ cycle, the sense amplifier must detect the contents of the cell and at the same time recharge the storage capacitance C_s. Notice that the sense amplifier thus solves the destructive readout problem and at the same time will be used for the refresh operation.

During a write cycle, the write gate is used to store a 1 or 0 in a cell — charge or discharge respectively.

12-3 MEMORY ADDRESSING

CELL SELECTION

Addressing is the process of selecting one of the cells in a memory to be written into or to be read from. In order to facilitate selection, memories are generally arranged by placing cells in a rectangular arrangement of rows and columns as shown in Fig. 12-9*a*. In this particular case, there are *m* rows and *n* columns, for a total of $n \times m$ cells in the memory.

The control circuitry that accompanies the basic memory array is designed such that one and only one row line is activated and one and only one column line is activated, the memory cell at the intersection of these two lines is selected. For instance, in Fig. 12-9*a*, if row *A* is activated and column *B* is activated, the cell at the intersection of this row and column is selected—that is, it can be read from or written into. For convenience, this cell is then called *AB*, corresponding to the row and the column selected. This designation is defined as the *address* of the cell. The activation of a line (row or column) is achieved by placing a logic 1 (or perhaps a logic 0) on it.

MATRIX ADDRESSING

Let's take a little time to consider the various possible configurations for a rectangular array of memory cells. The different rectangular arrays of 16 cells are shown in Fig. 12-10. In each of the five cases given, there are exactly 16 cells. The 16×1 and the 1×16 arrangements in Fig. 12-10*a* are really equivalent; likewise, the 8×2 and the 2×8 are essentially the same. So, there are really only three different configurations, each of which contain the exact same number of cells.

For any of the three configurations, the selection of a single cell still requires a single row and a single column to define a unique address. In Fig. 12-10*a*, a total of 17 address lines must be used—16 rows and 1 column, or 1 row and 16 columns. The minimum requirement in either case is really only 16 lines. However, either arrangement in Fig. 12-10*b* requires only 10 address lines—8 rows and 2 columns, or 2 rows and 8 columns. Clearly the best arrangement is given in Fig. 12-10*c*, since this configuration only requires 8 address lines—4 rows and 4 columns!

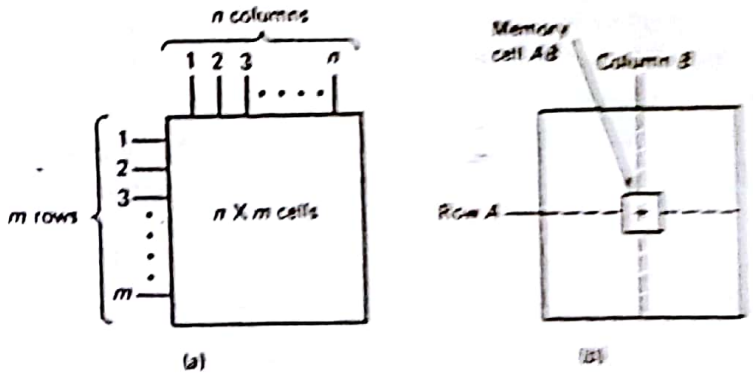


Fig. 12-9 (*a*) A rectangular array of $m \times n$ cells. (*b*) Selecting the cell at memory address *AB*.

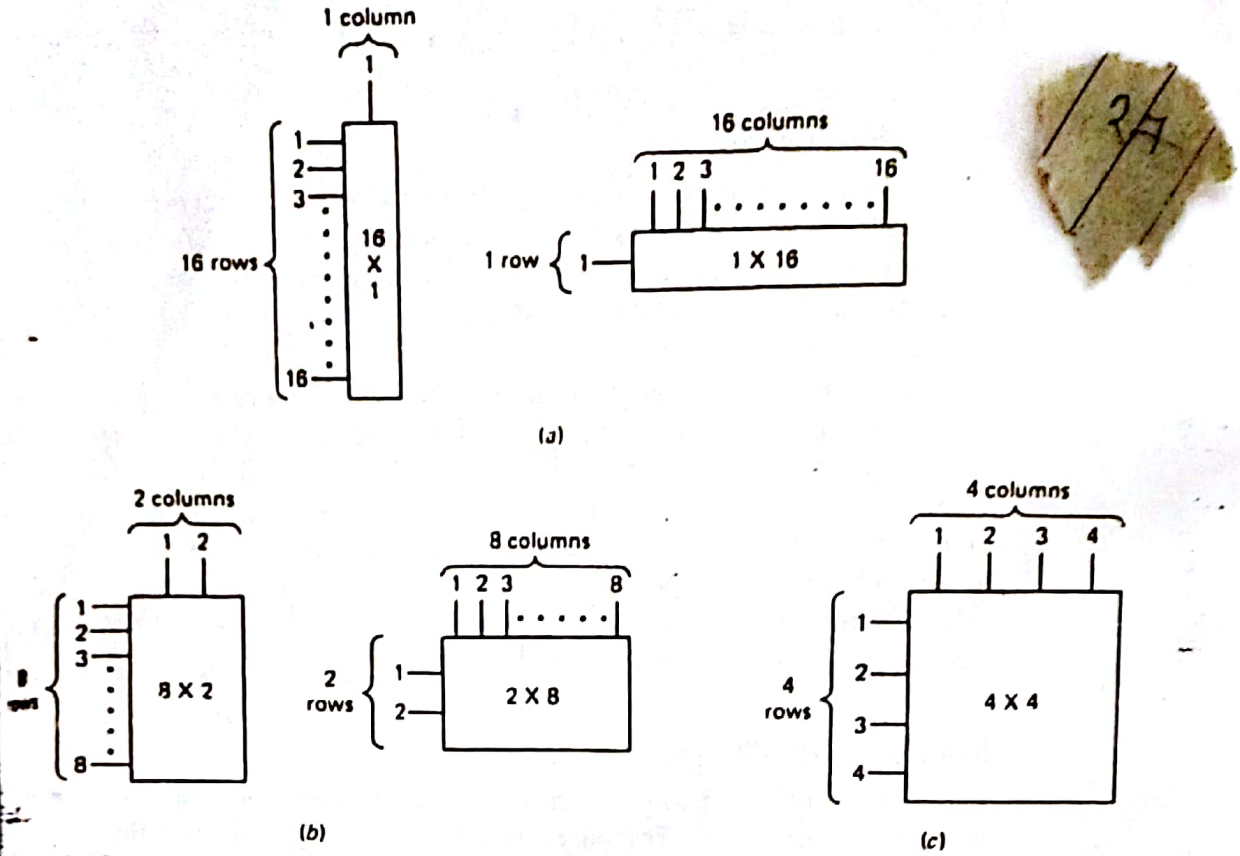


Fig. 12-10

In general, the arrangement that requires the fewest address lines is a square array of rows and n columns for a total memory capacity of $n \times n = n^2$ cells. It is exactly for this reason that the square configuration is so widely used in industry. This arrangement of n rows and n columns is frequently referred to as *matrix addressing*. In contrast, a single column that has n rows (such as the 16 × 1 array of cells) is frequently called *linear addressing*, since selection of a cell simply means selection of the corresponding row, and the column is always used.

matrix addressi

linear addressi

For instance, a 74S201 is a 256-bit bipolar RAM, arranged in a 256 × 1 array. The IEEE symbol for the 74S201 ('S201) is given in Fig. 12-11 on the next page. Eight address lines ($A_0, A_1, \ldots, A_7$) are required to select one of the 256 cells. There are three chip select lines ($\overline{S}_1$, $\overline{S}_2$ and $\overline{S}_3$), all of which must be low in order to activate (select) the chip. When the $R/\overline{W}$ line is high, the data bit at input D is stored at the selected address. When the $R/\overline{W}$ line is low, the *complement* of the bit at the selected address appears at the $\overline{Q}$ output. The small triangle ($\bigtriangledown$) at the $\overline{Q}$ output means that the output is three-state (tri-state). We'll use this chip in Sec. 12-5.

ADDRESS DECODING

Take another look at the 4 × 4 memory in Fig. 12-10*c*. To select a single cell, we must activate one and only one row, and one and only one column. This suggests the use of two 1 of 4 binary to decimal decoders as shown in Fig. 12-12 on the next page. Consider the selection of the cell at address 43 (row 4 and column 3). If $A_4 = 1$ and $A_3 = 1$, the

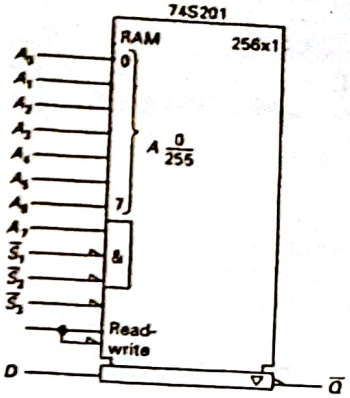


Fig. 12-11

decoder will hold the row 4 line high while all other row lines will be low. Similarly, if $A_2 = 1$ and $A_1 = 0$, the decoder will hold column 3 high and all other column lines low. Thus an input $A_4A_3A_2A_1 = 1110$ will select cell 43. We can consider A_4A_3 as a row address of 2 bits and A_2A_1 as a column address of 2 bits. Taken together, any cell in the array can be uniquely specified by the 4-bit address $A_4A_3A_2A_1$. As another example, the address $A_4A_3A_2A_1 = 0110$ selects the cell at row 2 and column 3 (address 23).

The address decoders shown in Fig. 12-12 further reduce the number of address lines needed to uniquely locate a memory cell, and they are almost always included on the memory chip. Recall that a binary-to-decimal decoder having n binary inputs will select one of 2^n output lines. For instance, a decoder that has 3 binary inputs will have $2^3 = 8$ outputs, or a decoder having 4 inputs will have 16 outputs, and so on.

In general, an address of B bits can be used to define a square memory of 2^B cells, where there are $B/2$ bits for the rows and $B/2$ bits for the columns, as shown in Fig. 12-13. Notice that the total number of address bits B must be an even integer (2, 4, 6, 8, . . .).

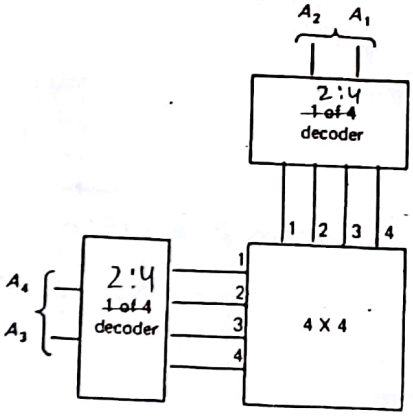


Fig. 12-12

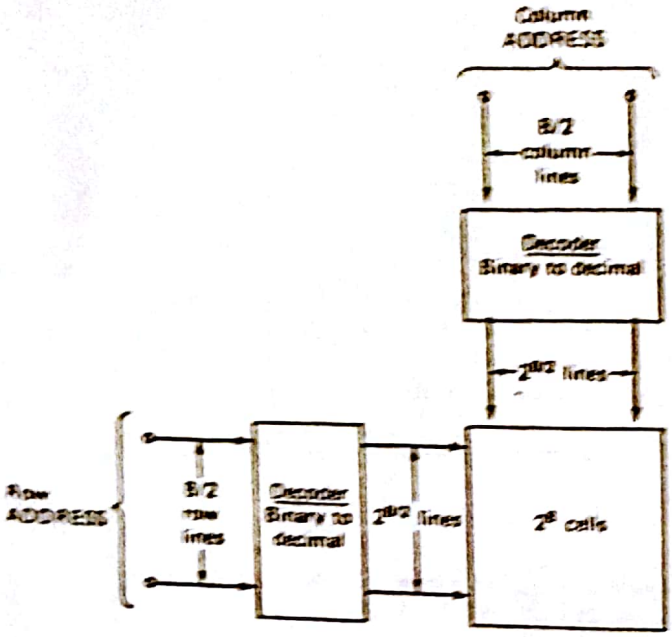


Fig. 12-13

Since the input to each decoder is $B/2$ bits, the output of each decoder must be $2^{B/2}$ lines. So the capacity of the memory must be $2^{B/2} \times 2^{B/2} = 2^B$. For instance, an address of 12 bits can be used in this way for a memory that has $2^{12} = 4096$ bits. There will be 6 address bits providing $2^6 = 64$ rows and likewise 6 address bits providing 64 columns. The memory will then be arranged as a square array of $64 \times 64 = 4096$ memory cells.

You may have noticed that most commercially available memories have capacities like 1024, 2048, 4096, 16,384, and so on. The reason for this is now clear—all of these numbers are clearly integer powers of 2! Incidentally, a memory having 1024 bits is usually referred to as a 1K memory (1000 bits) simply for convenience. Similarly, a memory advertised as 16K really has 16,384 bits, 4K is really 4096, and so on.

Example 12-2

What would be the structure of the binary address for a memory system having a capacity of 1024 bits?

Solution

Since $2^{10} = 1024$, there would have to be 10 bits in the address word. The first 5 bits could be used to designate one of the required 32 rows, and the second 5 bits could be used to designate one of the required 32 columns. Notice that $32 \times 32 = 1024$.

Example 12-3

For the memory system described in the previous example, what is the decimal address for the binary address 10110 01101? What is the address in hexadecimal?

Solution

The first 5 bits are the row address. Thus row = 10110 = 22. The second 5 bits are the column address. So, column = 13. The decimal address is thus 22 13. In hexadecimal this same address is 16 0D.

EXPANDABLE MEMORY

So far, we have only discussed memories that provide access to a single cell or bit at a time. It is often advantageous to access groups of bits—particularly groups of 4 bits (a nibble) and groups of 8 bits (a byte). It is not difficult to extend our discussion here to accommodate such requirements. There are at least two popular methods. The first simply accesses groups of cells on the same memory chip, and we discuss this idea next. The second connects memory chips in parallel, and we consider this technique in a following section.

The logic diagram for a 64-bit (16 × 4) bipolar memory is given in Fig. 12-14. There are 16 rows of cells with four cells in each row; thus the description (16 × 4). Each cell is a bipolar junction transistor flip-flop. The address decoder has 4 address bits and thus 16 select lines—one for each row. In this case, each select line is connected to all four of the cells in a row. So, each select line will now select four cells at a time. Therefore, each select line will select a 4-bit word (a nibble), rather than a single cell.

You might think of this arrangement as a "stack" of sixteen 4-bit registers. This is really a form of linear addressing, since the 4 address bits, when decoded, select one of

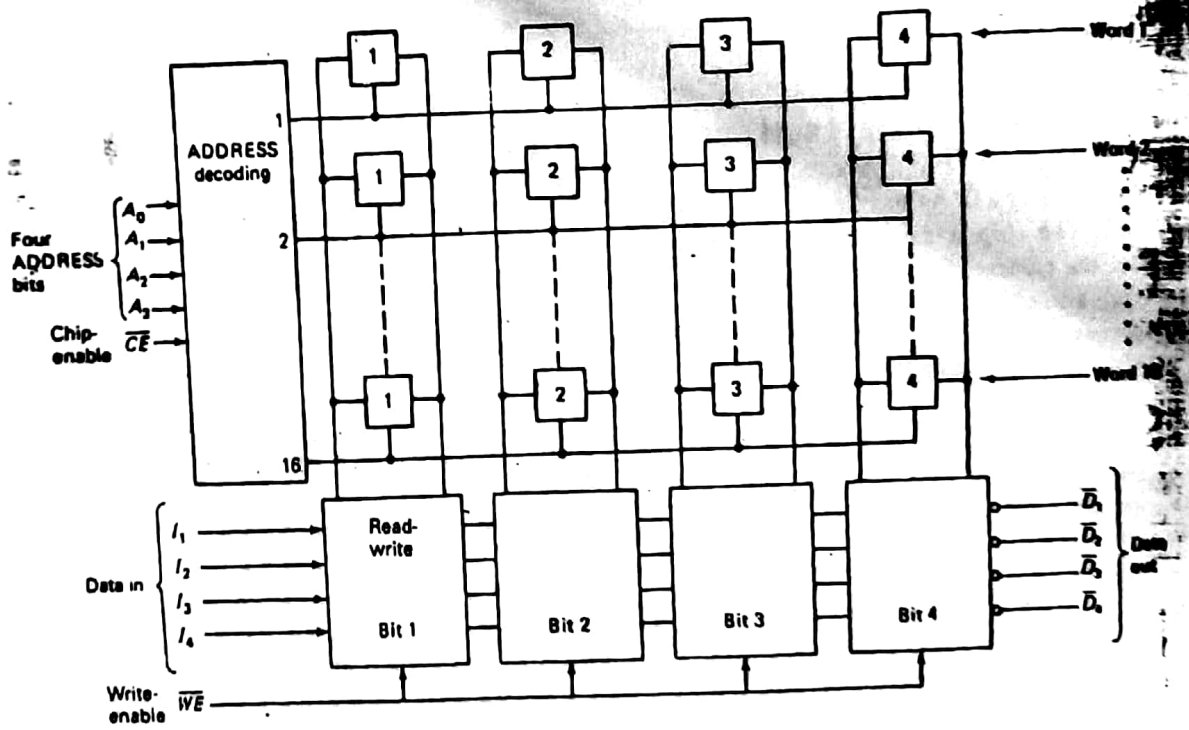


Fig. 12-14 64-bit (16 × 4) memory.

the sixteen 4-bit registers. In any case, when data is read from this memory it appears at the four data output lines $\bar{D}_1$, $\bar{D}_2$, $\bar{D}_3$, and $\bar{D}_4$ as a 4-bit data word. Similarly, data is presented to the memory for storage as a 4-bit data word at input lines I_1, I_2, I_3, and I_4. The 74S89 and the 74LS189 both are 64-bit (16 × 4) bipolar scratch pad memories arranged in exactly this configuration (look ahead in Fig. 12-20). The idea is easily extended to memories that access a word of 8 bits (a byte) at a time—for instance, the TBP18S030 ROM discussed in the next section.

mask-pro... bit chip

field-program... bit chip

SELF-TEST

7. What binary address will select cell 145 (decimal) in the 74S201 in Fig. 12-11?
8. The address applied in Fig. 12-12 is $A_4A_3A_2A_1 = 1010$. What cell is being accessed?

12-4 ROMs, PROMs, AND EPROMs

Having gained an understanding of memory addressing, let's turn our attention to the operation of a ROM. The term ROM is generally reserved for memory chips that are programmed by the manufacturer. Such a chip is said to be *mask-programmable*, in contrast to a PROM, which is said to be *field-programmable*—that is, it can be programmed by the user. EPROMs can be programmed, erased, and programmed again; they are clearly much more versatile chips than PROMs.

PROGRAMMING

What exactly does programming a ROM, PROM, or EPROM involve? It simply involves writing, or storing, a desired pattern of 0s and 1s (data). Each cell in the memory chip can store either a 1 or a 0. As supplied from the manufacturer, most chips have a 0 stored in each cell. The chip is then programmed by entering 1s in the appropriate cells. For instance, the content of every 4-bit word in a 64 × 4 chip is initially 0000. If the desired content of a word is to be 0110, then the two inner bit positions will be altered to 1s during programming.

In the case of a ROM, you must supply the manufacturer with the exact memory contents to be stored in the ROM. The Texas Instruments TMS4732 is a ROM having 4096 eight-bit words (a 4096 × 8 ROM). The logic diagram is given in Fig. 12-15 on the next page. The 8-bit word length makes this NMOS (*n*-channel MOS) chip ideal for microprocessor applications. Texas Instruments will store user-specified data during manufacturing. The user must supply data storage requirements in accordance with detailed instructions given on the TMS4732 data sheet.

The Texas Instruments TBP18S030 is a bipolar memory chip arranged as thirty-two 8-bit words (256 bits). The logic diagram for this user-programmable PROM chip is given in Fig. 12-16 on the next page. Basically, the programming is done by applying a current pulse to each output terminal where a logic 1 must appear (be stored). The current pulse will destroy an existing *fuse link*. When the fuse link is present, the transistor circuit in

we press and release 5. The digit 3 is shifted one place to the left and 5 appears on the extreme right. As we press and release 6, the digit 3 and 5 shift to the left and 6 appears in the extreme right position. This simple example illustrates two characteristics of a shift register.

1. It is a temporary memory and holds the number displayed.
2. When we press a new digit on the keyboard, the earlier number is shifted to the left. Thus, shift register has memory and shifting characteristics.

The basic shift operations are :

(*a*) serial shift left, then out

(*b*) serial shift right, then out

(*c*) Parallel shift in

(*d*) parallel shift out

(*e*) rotate left

(*f*) rotate right.

These operations are shown in Fig. 8.4.

8.4. Shift Left Register

Fig 8.5 shows, a shift left register. It uses *D* flip flops. The circuit shown has positive edge triggering. It is a 4 bit register using 4 flip flops FF_0, FF_1, FF_2, FF_3. D_{in} is the input to FF_0. Output of FF_0 is Q_0 and is fed to D_1. Similarly, Q_1 is fed to D_2 and Q_2 to D_3. All the flip flops are clocked together by the clock pulse.

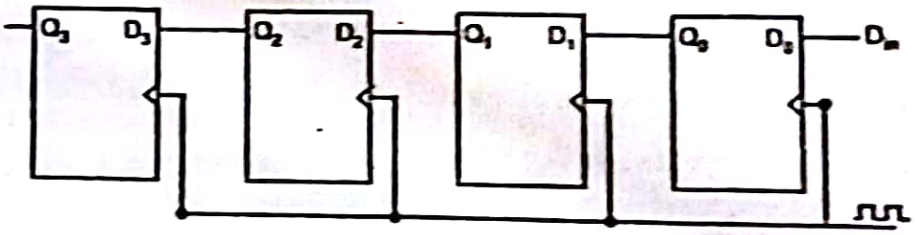


Fig. 8.5. Shift left register

Initially $Q_3\ Q_2\ Q_1\ Q_0 = 0000$

At the first positive edge of clock pulse FF_0 is set and then $Q_3\ Q_2\ Q_1\ Q_0 = 0001$

At the second positive edge of clock pulse FF_1 is set and then $Q_3\ Q_2\ Q_1\ Q_0 = 0011$

The positive edge of the third pulse sets FF_2 and then $Q_3\ Q_2\ Q_1\ Q_0 = 0111$

Finally, the positive edge of fourth clock pulse sets FF_3 and then $Q_3\ Q_2\ Q_1\ Q_0 = 1111$

As long as $D_{in} = 1$, the stored word cannot change further. Now let D_{in} be changed to 0. Then

8.5. Shift Right Register

Fig. 8.6 shows, a shift right register using D flip flops. As the name suggests the stored contents are shifted right on each clock pulse. The Q output is connected to the D input of preceding flip flop. At the arrival of each positive edge of clock shift right operation occurs. Let $D_{in} = 1$ and

$$Q_3\ Q_2\ Q_1\ Q_0 = 0$$

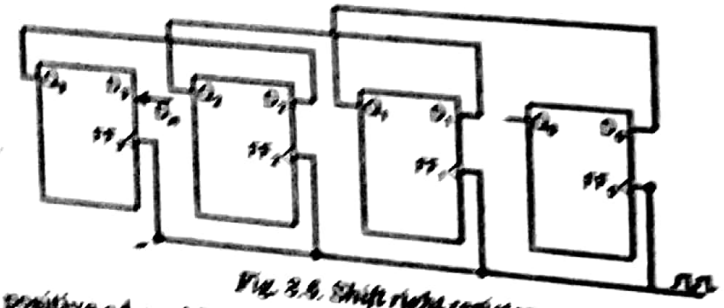


Fig. 8.6. Shift right register

The positive edge of first clock pulse sets up flip flop FF_3 and

$$Q_3\ Q_2\ Q_1\ Q_0 = 1000$$

The positive edge of second clock pulse makes the stored contents as

$$Q_3\ Q_2\ Q_1\ Q_0 = 1100$$

The positive edge of third clock pulse gives

$$Q_3\ Q_2\ Q_1\ Q_0 = 1110$$

and the positive edge of fourth clock pulse gives

$$Q_3\ Q_2\ Q_1\ Q_0 = 1111$$

After this the stored contents remain the same till $D_{in} = 1$. Let D_{in} be changed to 0 now. The successive clock pulses make the stored contents as under.

Clock pulse	Stored contents
1	0111
2	0011
3	0001
4	0000

As long as $D_{in} = 0$, the subsequent clock pulse do not cause any further change in stored contents.

8.6. Shift Register Operations

One method to describe the operation of shift register is the method of loading in and reading from the storage bits. There could be 4 such operations :

(a) **Serial in-Serial out :** The data is loaded into and read from the shift register serially. [Fig. 8.7(a)]

(b) **Serial in-Parallel out :** The data is loaded into the register serially but read in parallel (*i.e.*, data is available from all bits simultaneously. [Fig. 8.7(b)]

(c) **Parallel in-Serial out :** The data is loaded in parallel, *i.e.*, the bits are entered simultaneously in their respective stages and read serially. [Fig. 8.7(c)]

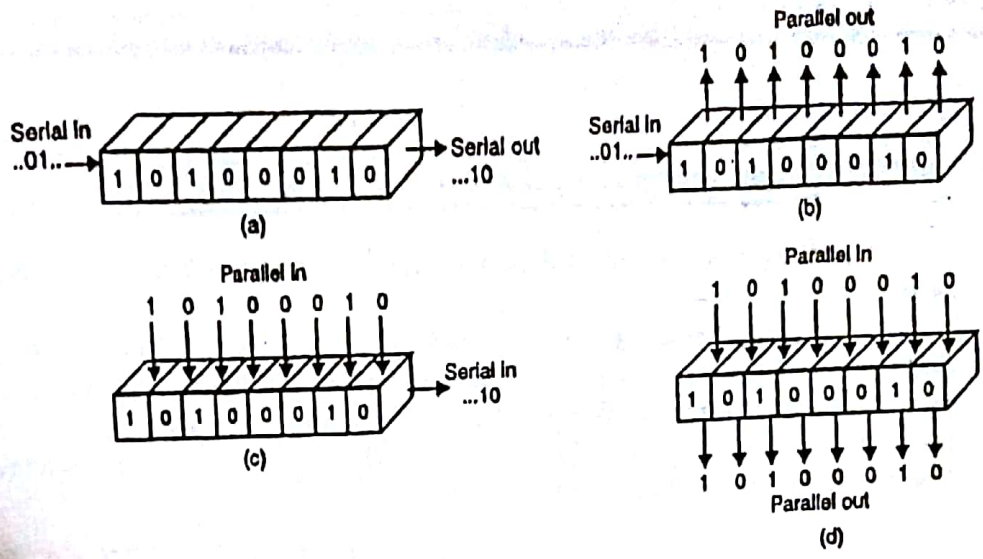


Fig. 8.7. Shift register operations (a) serial in - serial out (SISO) (b) serial in- parallel out SIPO) (c) parallel in - serial out (PISO) (d) parallel in - parallel out (PIPO)

8.7. Serial In- Serial out Shift Register

Figure 8.8 shows, a 4 bits serial in-serial out shift register consisting of four D flip flops FF_0, FF_1, FF_2 and FF_3. As shown it is a positive edge triggered device. We study the working of this register for the data 1010 in the following steps :

1. Bit 0 is entered into data input line. $D_0 = 0$, first clock pulse is applied, FF_0 is reset and stores 0.
2. Next bit 1 is entered. $Q_0 = 0$, since Q_0 is connected to D_1, D_1 becomes 0.
3. Second clock pulse is applied, The 1 on the input line is shifted into FF_0 because FF_0 sets. The 0 which was stored in FF_0 is shifted into FF_1.

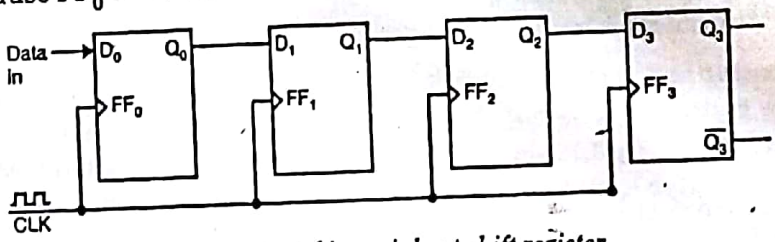


Fig. 8.8. Serial in-serial out shift register

4. Next bit 0 is entered and third clock pulse applied 0 is entered into FF_0, 1 stored in FF_0 is shifted to FF_1 and 0 stored in FF_1 is shifted to FF_2.
5. Last bit 1 is entered and 4th clock pulse applied. 1 is entered into FF_0, 0 stored in FF_0 is shifted to FF_1, 1 stored in FF_1 is shifted to FF_2 and 0 stored in FF_2 is shifted to FF_3. This completes the serial entry of 4 bit data into the register. Now the LSB 0 is on the output Q_3.
6. Clock pulse 5 is applied.LSB 0 is shifted out. The next bit 1 appears on Q_3 output.
7. Clock pulse 6 is applied. The 1 on Q_3 is shifted out and 0 appears on Q_3 output.
8. Clock pulse 7 is applied. 0 on Q_3 is shifted out. Now 1 appears on Q_3 output.
9. Clock pulse 8 is applied. 1 on Q_3 is shifted out.

When the bits are being shifted out (on CLK pulse 5 to 8) more data bits can be entered in.

IC 7491A is a serial in-serial out shift register. It uses eight clocked SR flip flops. Thus it is an 8 bit register. It is designated as SRG8 (shift register 8 bits). A and B are two input lines. When data is entered into A, B must be High and *vice versa*. The output is Q_H and $\overline{Q}_H$ is complement of Q_H. Its logic symbol is shown in Fig. 8.9. The pin numbers are as shown.

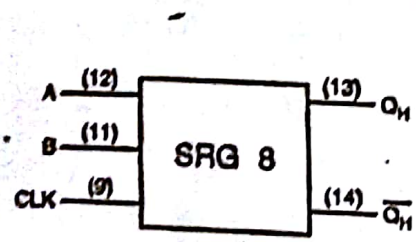


Fig. 8.9. Logic symbol of 8 bit serial in serial out IC shift register 7491 A

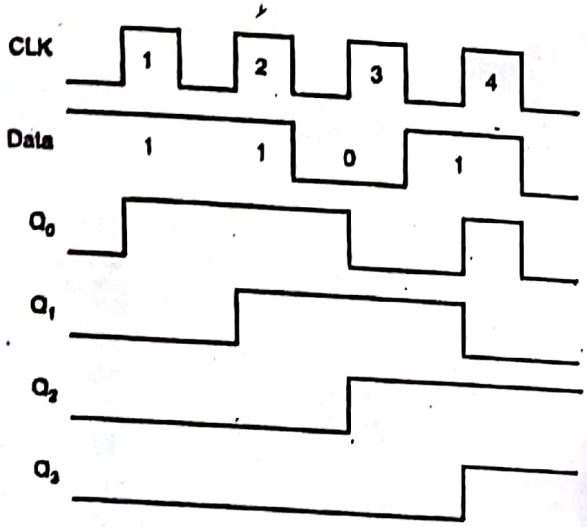


Fig. 8.10.

Example 8.1. *Data 1101 is fed into 4 bit serial in - serial out shift register. Show the status of register at various clock pulses.*

Solution : Fig 8.10 shows the diagram depicting the status of register.

8.8. Serial In-parallel Out Shift Register

The data bits are entered serially but they are available at their respective positions simultaneously. Fig. 8.11 show the circuit of 4 bit serial in-parallel out shift register using D flip flops. Q_0, Q_1 Q_2 and Q_3 are the output terminals and data is available at all of them together.

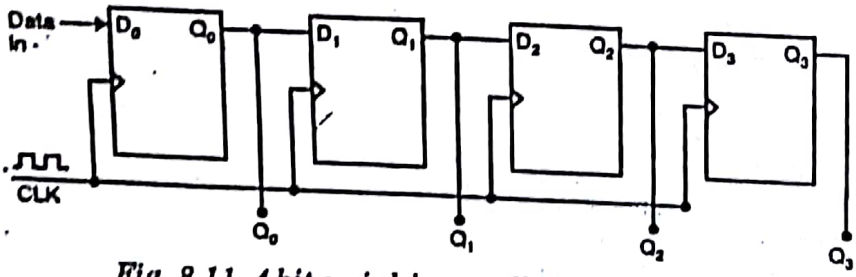


Fig. 8.11. 4 bit serial-in parallel out shift register

IC 74164 is an 8 bit serial in-parallel out shift register. It has two serial inputs A and B, active Low clear $\overline{CLR}$ and parallel outputs Q_A to Q_H. It uses SR flip flops. Fig. 8.12 shows its logical its logical symbol and pins.

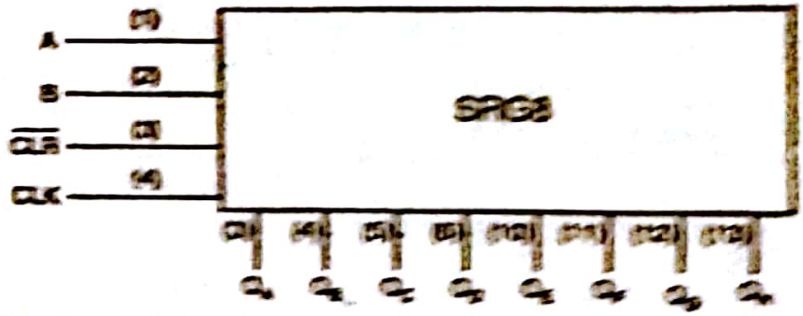
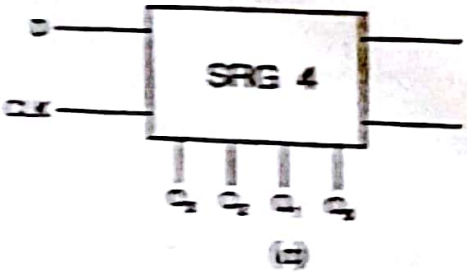
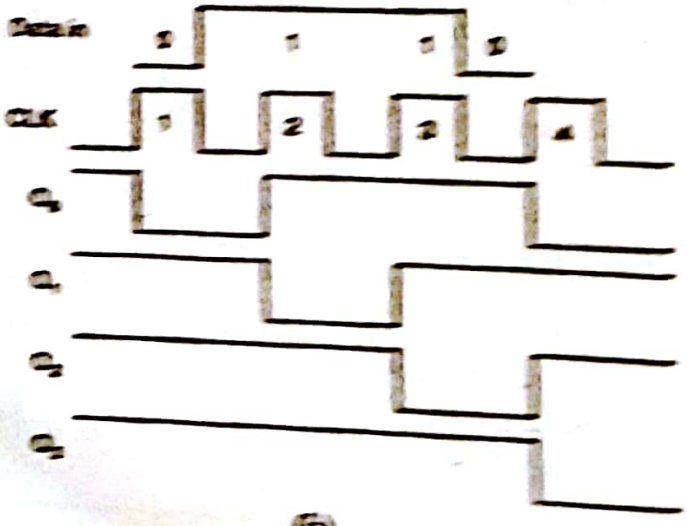


Fig. 8.12. 8 bit serial in-parallel out shift register IC 74164

Example 8.2. *Fig. 8.13 (a) shows a 4 bit serial in-parallel out shift register. Draw the status of 4 registers for the data 0110.*



(a)



(b)

Fig. 8.13.

Solution : Fig. 8.13 (b) shows the diagram depicting the status of register.

8.9. Parallel in-Serial Out Shift Register

The data bits are entered simultaneously and taken out serially. Fig. 8.14 shows such a shift register with 4 bits. It uses D flip flops and 4 data input lines A, B, C, D. Moreover, it has a shift/Load input which allows 4 bits of data to be entered

into the register simultaneously. When shift/$\overline{\text{Load}}$ is Low, gates G_1, G_2, G_3 are enabled and data can be entered into the D input of the four flip flops. When clock pulse is applied, the flip flop with data bit 1 will SET and flip flops with data bits 0 will reset and thus the 4 bit data will be stored.

When shift/$\overline{\text{Load}}$ is High, gates G_1, G_2, G_3 are disabled while gates G_4, G_5, G_6 are enabled. Thus data bits can shift right one stage to the next. The OR gates allow parallel data entry operation or shifting opeartion depending on which AND gates are enabled by the level on shift/$\overline{\text{Load}}$ input.

IC 74165 is an 8 bit parallel in-serial out shift register SH/$\overline{\text{LD}}$ is the shift/$\overline{\text{Load}}$ terminal. *ABCDEFGH* are the terminals for 8 bit data input. An additional terminal SER is provided for serial data input. The clock can be inhibited any time by a High on CLKINH input. The serial data output is Q_H and its complement is $\overline{Q_H}$. Fig. 8.15 shows its symbol.

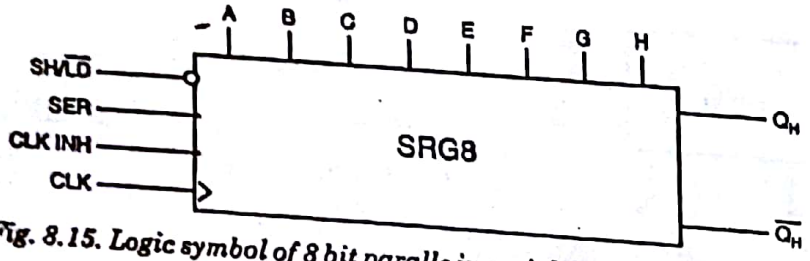


Fig. 8.15. Logic symbol of 8 bit paralle in-serial out IC 74165 shift register

Example 8.3. *Fig. 8.16 (a) shows the symbol of a 4 bit parallel in-serial out shift regisrter. Data 1010 is entered in it. Show the data and waveform along with clock and* shift/$\overline{\text{Load}}$ *waveforms.*

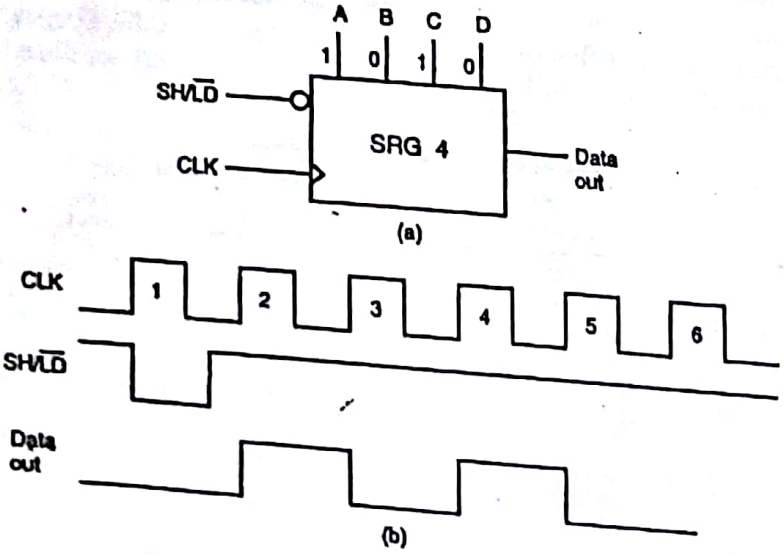


Fig. 8.16.

Solution : The clock, shift/$\overline{\text{Load}}$ and data (1010) waveforms are shown in Fig. 8.16(*b*).

8.10. Parallel in - parallel out Shift Register

Fig. 8.17 (*a*) shows a 4 bit parallel in-parallel out shift register. *ABCD* are parallel data bits and Q_A Q_B Q_C Q_D are parallel data outputs.

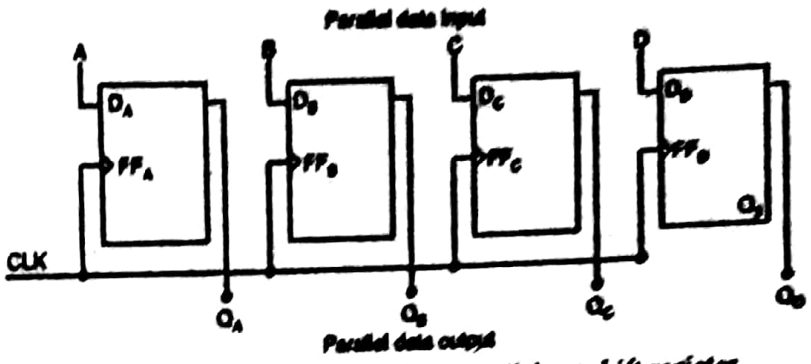


Fig. 8.17.(a) Parallel in - parallel out shift register

IC 74195 is a 4 bit parallel in - parallel out shift register. Its symbol is shown in Fig. 8.17 (*b*). When SH/$\overline{\text{LD}}$ input is Low, data on parallel lines *A, B, C, D* can be entered synchronously on the positive edge of clock pulse. When SH/$\overline{\text{LD}}$ input is High, stored data shifts to right (*i.e.*, Q_A to Q_D) synchronously with the clock. This IC can also be used for serial input-serial output. *J* and $\overline{K}$ are serial data inputs to the first stage of IC. Q_D can be used for serial data output. The active Low clear is asynchronous.

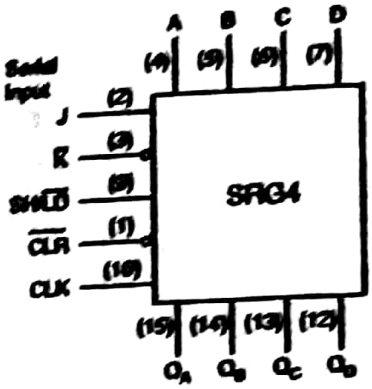
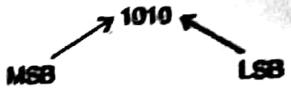


Fig. 8.17(b) logic symbol of IC 74195 4 bit shift register

Example 8.4. In fig. 8.17 (a), A = 0, B = 1, C = 0 and D = 1. What are data outputs after 2 clock pulses.

Solution : The data output remains the same as input, *i.e.*, 1010



8.11. Universal Shift register

A universal shift register (also known as bidirectional shift register) can shift data in both directions, *i.e.*, left as well as right. The logic gates are arranged in such a way that data bits can be transferred from one stage to the next in either direction depending on the control line input. Fig. 8.18 (*a*) shows, the circuit of such a register using *D* flip flops. When Right/$\overline{\text{Left}}$ control input is High, it acts as a shift right register. This occurs because gates G_1, G_2, G_3, G_4 are enabled and the *Q* output of one stage goes to *D* input of next stage. At positive edge of each clock, data bits shift one place to the right. When Right/$\overline{\text{Left}}$ control inputs is Low, it acts as a shift left register. In this case gates G_5, G_6, G_7, G_8 are enabled and *Q* output of each stage goes to *D* input of preceding stage. At the positive edge of each clock data bits shift one place to the left.

IC 74194 is a 4 bit universal shift register. Its symbol is shown in Fig. 8.18(*b*). *ABCD* are the inputs and Q_A, Q_B Q_C, Q_D are outputs. Parallel loading is achieved by applying 4 bits of data simultaneously to the inputs *ABCD* and High to S_0, S_1 inputs. At positive edge of clock, parallel data is loaded into the register.

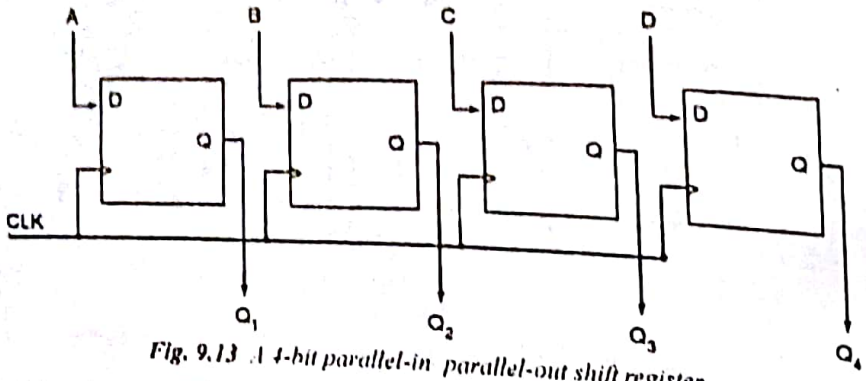


Fig. 9.13 A 4-bit parallel-in parallel-out shift register

When the SH / $\overline{LD}$ input is LOW, the data on the parallel inputs, i.e. A, B, C and D, are entered synchronously on the positive transition of the clock. When SH/ $\overline{LD}$ is HIGH, stored data will shift right (Q_A to Q_D) synchronously with the clock. Let J and $\overline{K}$ be the serial data inputs to the first stage of the register (Q_A); Q_D can be used for getting a serial output data. The active-LOW clear is asynchronous.

There are a number of 4-bit, parallel-input–parallel-output shift registers available since they can be conveniently packaged in a 16-pin DIP. An 8-bit register can be created by connecting two 4-bit registers in series. ICs 74174, 74178, 74198 and 7495 are parallel-in–parallel-out registers.

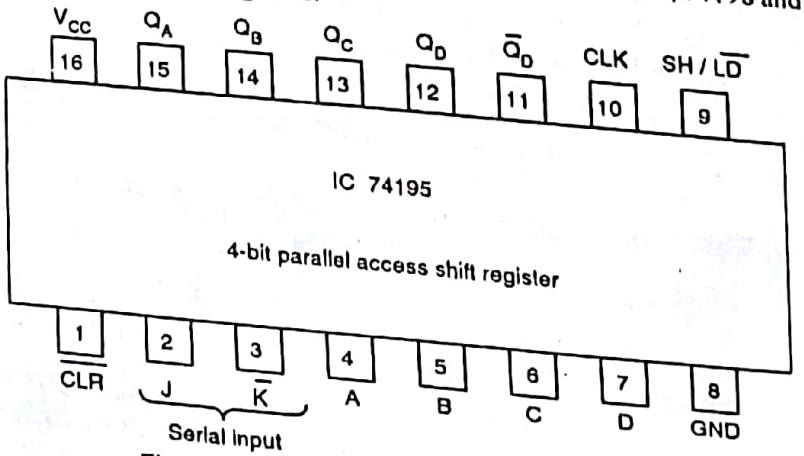


Fig. 9.14 Pinout diagram of IC 74195 shift register

9.3 UNIVERSAL SHIFT REGISTERS

A register which is capable of shifting data both to the right and left is called a *bi-directional shift register*. A register that can shift in only one direction is called a *uni-directional shift register*. If the register has shift and parallel load capabilities, then it is called a *shift register with parallel load or Universal Shift Register*.

Shift registers can be used for converting serial data to parallel data, and vice-versa. If a parallel load capability is added to a shift register, then data entered in parallel can be taken out in serial fashion by shifting the data stored in the register.

Some shift registers have necessary input and output terminals and also have both shift-right and shift-left capabilities. The most general shift register has 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 CLK input for clock pulses to synchronise all operations.
3. A *shift-right* control to enable the shift-right operation and *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 line that leaves the information in the register unchanged even though clock pulses are continuously applied.

The diagram of a universal shift register that has all the capabilities listed above is shown in Fig. 9.15. It consists of four D flip-flops and four 4-input multiplexers (MUX). Let S_0 and S_1 be the two selection inputs connected to all the four multiplexers. These two selection inputs are used to select one of the four inputs of each multiplexer. Input 0 in each MUX is selected when $S_1S_0 = 00$ and input 1 is selected when $S_1S_0 = 01$. Similarly, inputs 2 and 3 are selected when $S_1S_0 = 10$ and $S_1S_0 = 11$ respectively. The inputs S_1 and S_0 control the mode of operation of the register. When $S_1S_0 = 00$, the present value of the register is applied to the D inputs of the flip-flops. This is done by connecting the output of each flip-flop to the 0 input of the respective multiplexer. The next clock pulse transfers into each flip-flops, the binary value it held previously, and hence no change of state occurs. When $S_1S_0 = 01$, terminal 1 of the multiplexer inputs has a path to the D inputs of the flip-flops. This causes a shift-right operation, with the left serial input transferred into flip-flop A_3. When $S_1S_0 = 10$, a shift-left operation results, with the right serial input going into flip-flop A_0. Finally, when $S_1S_0 = 11$, the binary information on the parallel input lines (I_0, I_1, I_2 and I_3) are transferred into the register simultaneously during the next clock pulse. The function table of bidirectional shift register with parallel inputs and parallel outputs is shown in Table 9.6.

Table 9.6 Function table of bidirectional shift register

Mode control		Register operation
S_1	S_0	
0	0	No change
0	1	Shift-right
1	0	Shift-left
1	1	Parallel Load

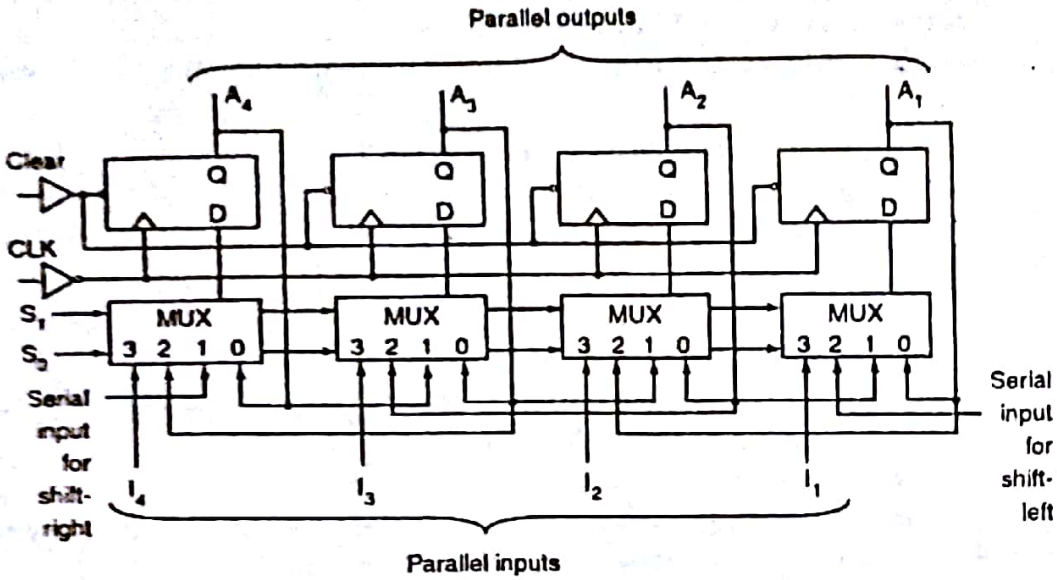


Fig. 9.15 A 4-bit Universal shift register

A 4-bit bidirectional shift register is shown in Fig. 9.16. When mode control $M = 1$, the AND gates G_1 through G_4 are enabled and the data at D_R is shifted to the right when the clock pulses are applied, and thus it acts as a shift-right register. When $M = 0$, the AND gates G_5 through G_8 are enabled allowing the data at D_L to be shifted to the left, and thus it acts as a shift-left register. M should be changed only when CLK = 0, otherwise the data stored in the register may be altered.

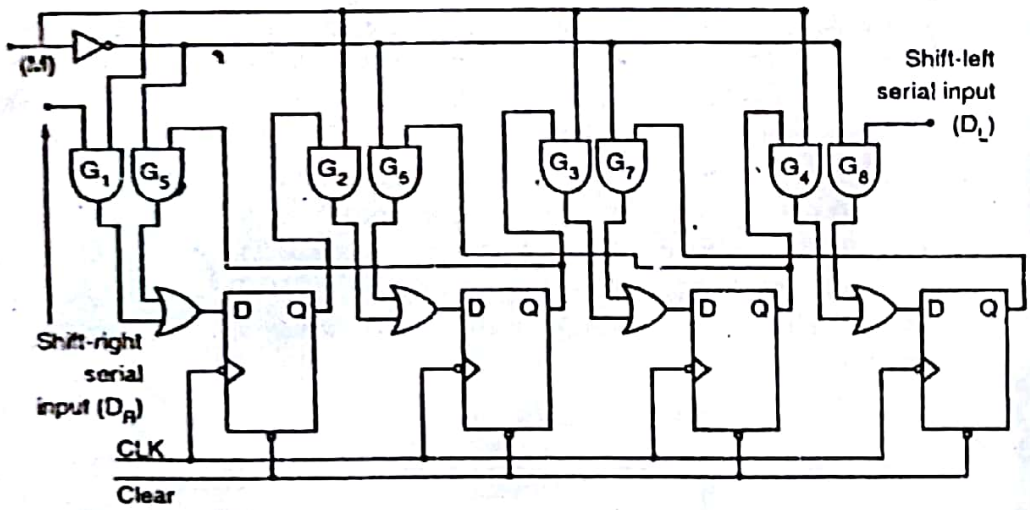


Fig. 9.16 A 4-bit bidirectional shift register

9.3.1 IC 74194—4-bit Bidirectional Shift Register

IC 74194 is a four-bit bidirectional shift register. The pinout diagram of IC 74194 is shown in Fig. 9.17. Parallel loading, which is synchronous with a positive transition of the clock, is accomplished by applying the four bits of data to the parallel inputs and a HIGH to the S_0 and S_1 inputs.

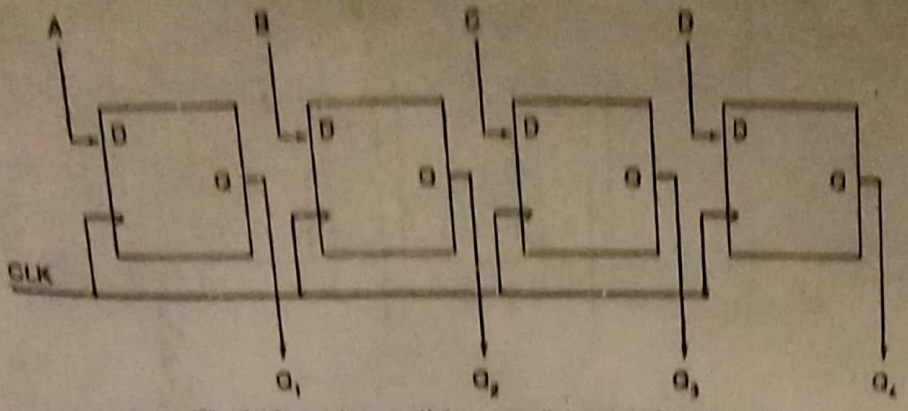
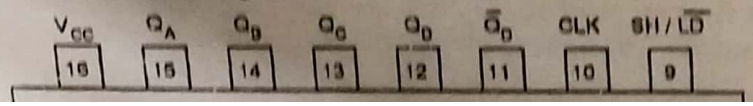


Fig. 9.13 A 4-bit parallel-in parallel-out shift register

When the SH/$\overline{LD}$ input is LOW, the data on the parallel inputs, i.e. *A*, *B*, *C* and *D*, are entered synchronously on the positive transition of the clock. When SH/$\overline{LD}$ is HIGH, stored data will shift right (Q_A to Q_D) synchronously with the clock. Let *J* and $\overline{K}$ be the serial data inputs to the first stage of the register (Q_A); Q_D can be used for getting a serial output data. The active-LOW clear is asynchronous.

There are a number of 4-bit, parallel-input–parallel-output shift registers available since they can be conveniently packaged in a 16-pin DIP. An 8-bit register can be created by connecting two 4-bit registers in series. ICs 74174, 74175, 74198 and 7495 are parallel-in–parallel-out registers.



Some shi
shift-right and
ties listed belo
operation.

1. A cle
2. A Cl
3. A *shi*
outpu
4. A *shi*
outpu
5. A *par*
associ
6. *n* para
7. A con
though

The diagra
is shown in Fig
(MUX). Let S_0 a
ers. These two s
plexer. Input 0 is
$S_1S_0 = 01$. Simila
tively. The inpu
$S_1S_0 = 00$, the pr
This is done by c

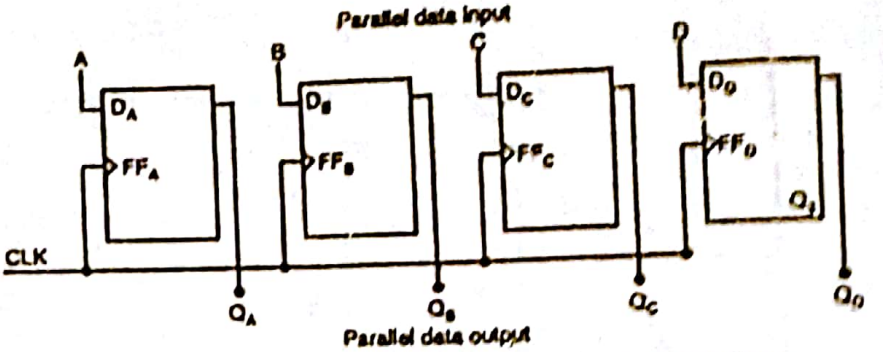


Fig. 8.17.(a) Parallel in - parallel out shift register

IC 74195 is a 4 bit parallel in - parallel out shift register. Its symbol is shown in Fig. 8.17 (*b*). When SH/$\overline{LD}$ input is Low, data on parallel lines *A*, *B*, *C*, *D* can be entered synchronously on the positive edge of clock pulse. When SH/$\overline{LD}$ input is High, stored data shifts to right (*i.e.*, Q_A to Q_D) synchronously with the clock. This IC can also be used for serial input-serial output. *J* and $\overline{K}$ are serial data inputs to the first stage of IC. Q_D can be used for serial data output. The active Low clear is asynchronous.

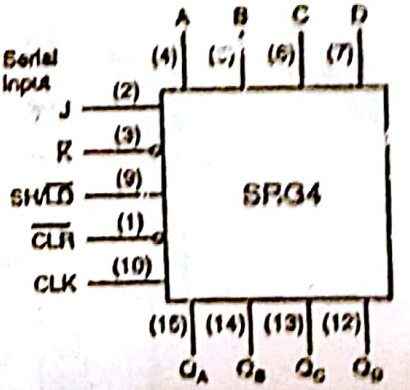
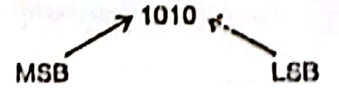


Fig. 8.17(b) logic symbol of IC 74195 4 bit shift register

Example 8.4. In fig. 8.17 (a), A = 0, B = 1, C = 0 and D = 1. What are data outputs after 2 clock pulses.

Solution : The data output remains the same as input, *i.e.*,



8.11. Universal Shift register

A universal shift register (also known as bidirectional shift register) can shift data in both directions, *i.e.*, left as well as right. The logic gates are arranged in such a way that data bits can be transferred from one stage to the next in either direction depending on the control line input. Fig. 8.18 (*a*) shows, the circuit of such a register using *D* flip flops. When Right/$\overline{Left}$ control input is High, it acts as a shift right register. This occurs because gates G_1, G_2, G_3, G_4 are enabled and the *Q* output of one stage goes to *D* input of next stage. At positive edge of each clock, data bits shift one place to the right. When Right/$\overline{Left}$ control inputs is Low, it acts as a shift left register. In this case gates G_5, G_6, G_7, G_8 are enabled and *Q* output of each stage goes to *D* input of preceding stage. At the positive edge of each clock data bits shift one place to the left.

IC 74194 is a 4 bit universal shift register. Its symbol is shown in Fig. 8.18(*b*). *ABCD* are the inputs and Q_A, Q_B Q_C, Q_D are outputs. Parallel loading is achieved by applying 4 bits of data simultaneously to the inputs *ABCD* and High to S_0, S_1 inputs. At positive edge of clock, parallel data is loaded into the register.