Semiconductor Memories

Introduction

Basically memory is a means of storing data or information in the form of binary words. The information usually consists of programs ie set of instructions that a computer executes to achieve a desired result. Memory used to store data is called data memory and memory used to store programs is called program memory.

computers which store programs in their memory are called stored-program type computers. All modern computers are of the stored-program type. In these computers, programs are stored as a set of machine Language instructions, in binary codes. Each memory location is identified by an address. The number of storage locations can vary from a few in some memories to millions in others.

Each storage location can accommodate one or more bits. Generally the total number of bits a memory can store is its capacity. Sometimes the capacity is specified in terms of bytes. Memories are made up of storage elements (FFs or capacitors in semiconductor memories and magnetic domains in magnetic memories), each of which stores one bit of data. A storage element is called a cell.

These memories have become very popular due to their small size, Low cost, high speed, high reliability & ease of expansion of the memory size. Therefore it is necessary for a designer of digital processors to know thoroughly the principles of operation and Limitations of various semiconductor memory devices.

Memory Organization and Operation

All memory, regardless of its type or use, consists of locations for storing binary information or bits. Each location is identified by an address. A word is the fundamental group of bits used to represent one entity of information such as one numerical value. The word size - the number of bits in a word - varies among computer systems and may range from 4 to 64 or more bits. The word size is usually expressed as a certain number of bytes. For example, a 16-bit word is 2 bytes.

A memory location is thus a set of devices capable of storing one word. For example, each memory location in an 8-bit microcomputer one that uses 8-bit words might consist of eight latches. Each latch stores one bit of a word, and is referred to as a cell. The capacity or size of a memory is the total number of bits or bytes or words that it can store.

For convenience, the size of a memory is expressed as a multiple of $2^{10} = 1024$, which is abbreviated as K. For example, a memory of size $2^{11} = 2048$ is said to be 2K. A memory of size 2^{14} (16,384) is 16K and a memory of size 2^{16} (65,536) is 64K.

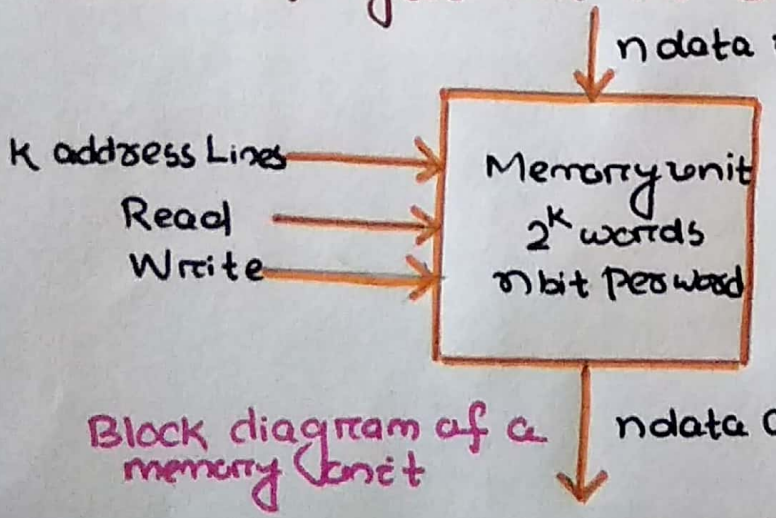
Each memory system requires several different types of input and output lines to perform the following functions.

1. Select the address in memory that is to be accessed for a read or write operation.
2. Select either a read or a write operation to be performed.
3. Supply the input data to be stored in memory during a write operation.
4. Hold the output data coming from memory during a read operation
5. Enable or disable the memory, so that it will or will not respond to the address inputs and read/write command.

Memory Unit

A memory unit is a collection of storage cells, together with associated circuits needed to transfer information into or out of a device The architecture of memory is such that information can be selectively retrieved from any of its internal locations.

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. A memory word is a group of 1's and 0's and may represent a number, an instruction, one or more alphanumeric characters, or any binary coded information A group of 8 bits is called a byte. Most computer memories use words that are multiples of 8 bit length. Thus, a 16-bit word contains two bytes, and a 32-bit word is made up of four bits. The capacity of a memory unit is usually stated as the total number of bytes that the unit can store.



Block diagram of a memory unit

The n data input lines provide the information to be stored in the memory, and the n data output lines supply information coming out of memory. The K address lines specify the particular word chosen among the many available. The Write input causes binary data to be transferred into the memory and the Read input causes binary data to be transferred out of memory

The address lines select one particular word. Each word in memory is assigned an identification number, called an address, starting from 0 up to $2^K - 1$, where K is the number of address lines.

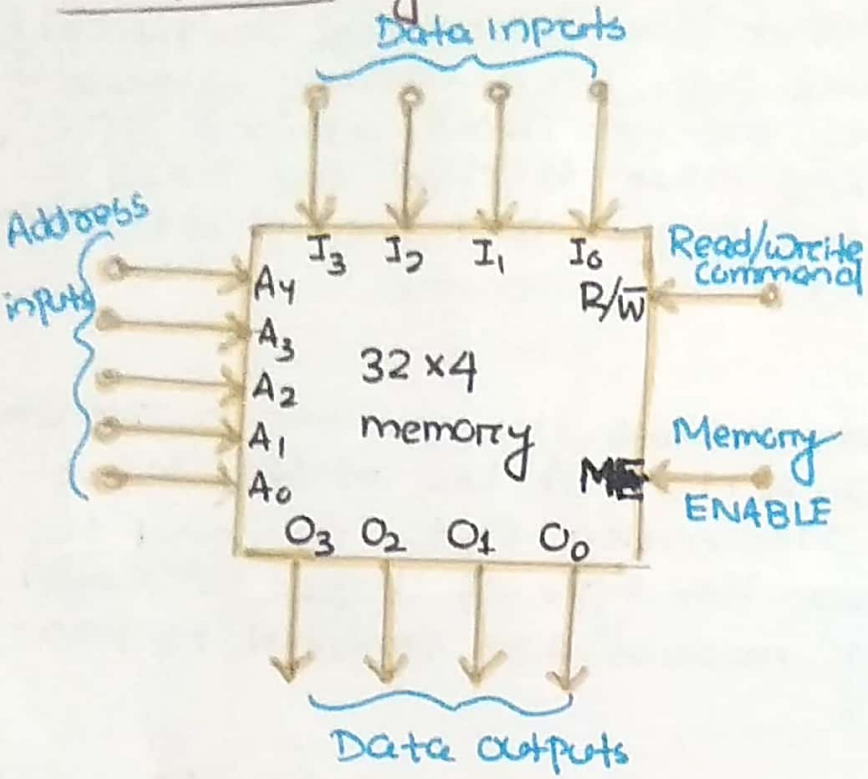
Q Consider a memory of size 16 words. Find the binary address of each location.

Since $M = 16$, therefore $M = 2^K$ gives $K = 4$.
So for selecting one out of 16 words, a 4-bit address is required.

The address is specified as $A_3 A_2 A_1 A_0$, where A_3 is the most-significant bit (MSB) and A_0 represents the Least-significant bit (LSB) of the address. The address of each Location is given in the table.

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

32 x 4 memory



Memory cells				Addresses
0	1	1	0	00000
1	0	0	1	00001
1	1	1	1	00010
1	0	0	0	00011
0	0	0	1	00100
0	0	0	0	00101
				⋮
1	1	0	1	11101
1	1	0	1	11110
1	1	1	1	11111

Diagram of a 32x4 memory and arrangement of memory cells.

A 32x4 memory stores 32 4-bit words. Since the word size of 4 bits, there are four input lines I_0 to I_3 and four data output lines O_0 to O_3. During a write operation, the data to be stored in memory have to be applied to the data input lines. During a read operation, a word being read from memory appears at the data output lines.

Address inputs

Since this memory stores 32 words, it has 32 different storage locations and therefore, 32 different binary addresses ranging from 00000 to 11111 (0 to 31 decimal). Thus there are five address inputs A_0 to A_4. To access one of the memory locations for a read or write operation, the 5-bit address code for that particular location is applied to the address inputs. In general, N address inputs are required for a memory that has a capacity of 2^N words.

Each address location contains four memory cells that hold 1s and 0s to make up the data word stored at that location.

The R/$\bar{W}$ input

The read/write (R/$\bar{W}$) input line determines the memory operation that would take place. Some memory systems use two separate inputs, one for read and one for write. When a single R/$\bar{W}$ input is used, the read operation takes place for R/$\bar{W}$ = 1 and the write operation takes place for R/$\bar{W}$ = 0.

Memory ENABLE

An active-HIGH input enables the memory to operate normally when it is kept HIGH. A Low on this input disables the memory, preventing it to respond to address and R/$\bar{W}$ inputs. This type of input is useful when several memory modules are combined to form Larger memory.

Q A certain memory has a capacity of 8K x 16.

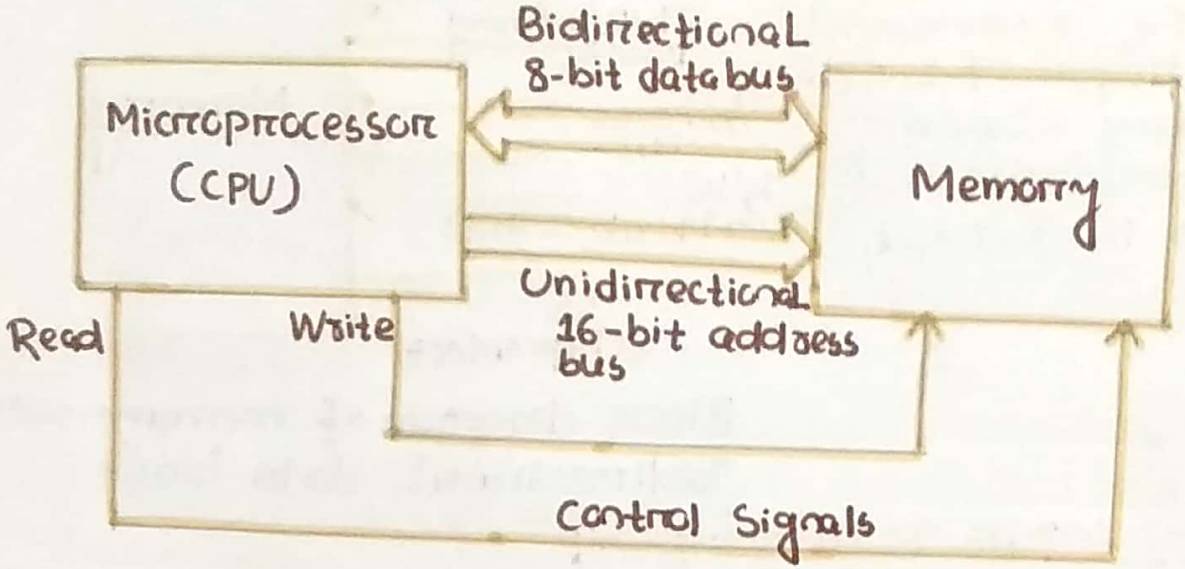
a) How many data input and data output lines does it have ?

b) How many address lines does it have ?

c) What is its capacity in bytes ?

Write and Read Operation

The process of storing data in memory is called writing in memory. Retrieving data from memory is called reading memory



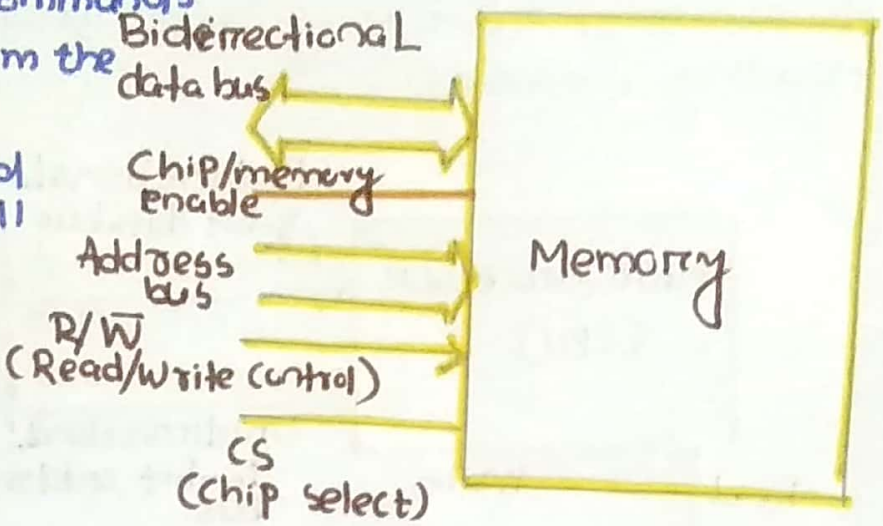
Microcomputer System

The microprocessor serves as the central processing unit (CPU) for the computer. It contains an arithmetic/Logic unit (ALU), registers array and control unit, that is uses to perform read and write operations as well as to execute programs. The control signals are Labelled as read and write. The CPU activates these when a read or a write operation is to be performed The 8-bit data bus consisting of eight lines on which data bits D_0 through D_7 are transmitted. It is called a bidirectional data bus, because words can be transmitted from the CPU to memory (write) or from memory to the CPU (read). The unidirectional address bus is the set of lines over which the CPU transmits the address bits corresponding to the memory address to be read or written into.

The address bus is a 16-bit bus (A_0 through A_{15}) meaning that the CPU can access (read or write into) up to $2^{16} = 65,536$ different memory addresses.

A number of control inputs are required to give commands to the device to perform the desired operation.

For example, a command signal is required to tell the memory whether a read or write $R/\bar{W}$ operation is desired.



Block diagram of memory with bidirectional data bus

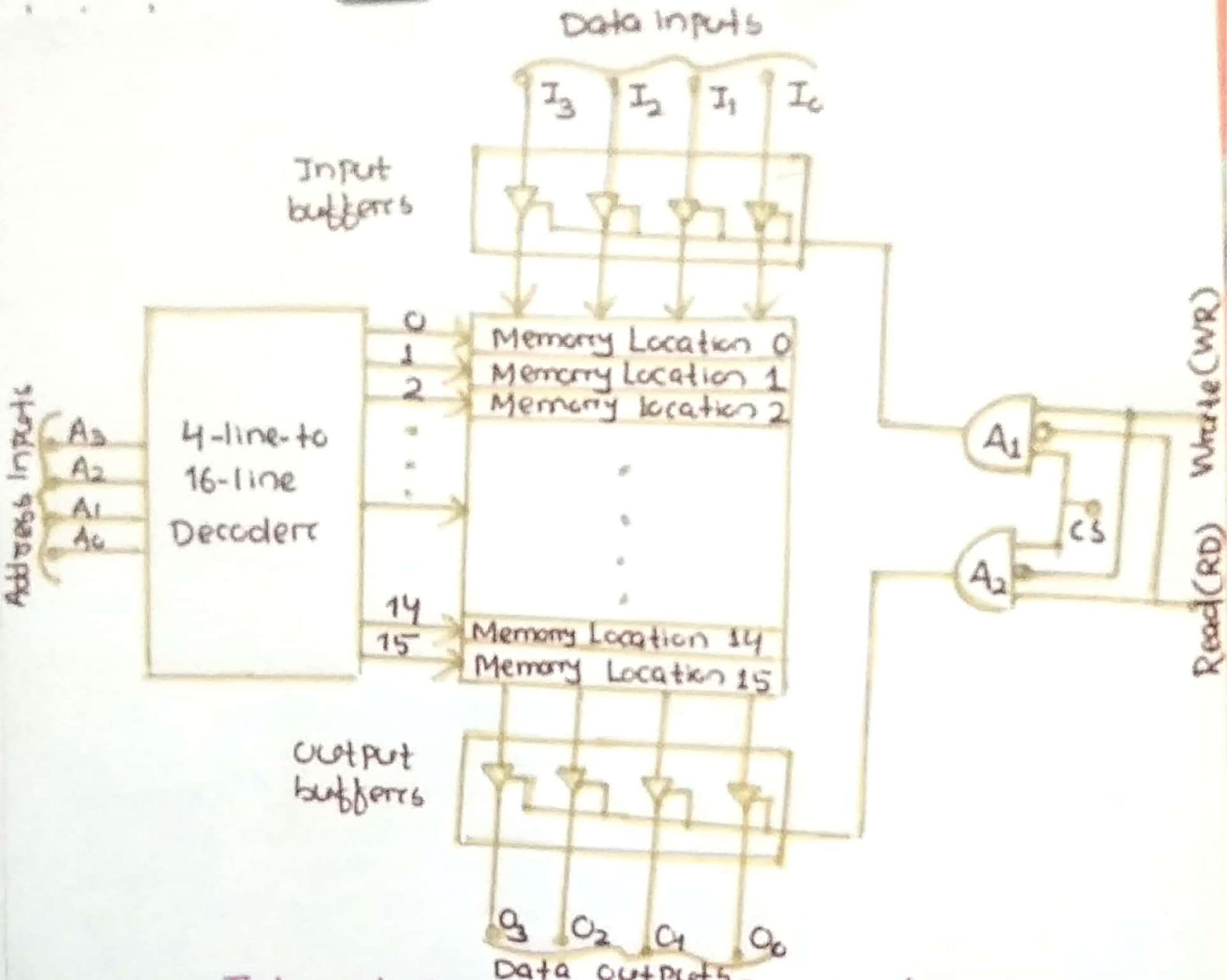
Control inputs to memory chip

Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

When $R/\bar{W}$ is HIGH, the databus will be used for reading the memory whereas when $R/\bar{W}$ is LOW, the data on the bus will go into the memory. Other command include inputs memory or chip enable, Chip Select (CS) etc.

In addition to the above mentioned functional pins, a minimum of two pins are required for power supply and ground.

The internal organization of a 16x4 memory chip is illustrated in the below figure.



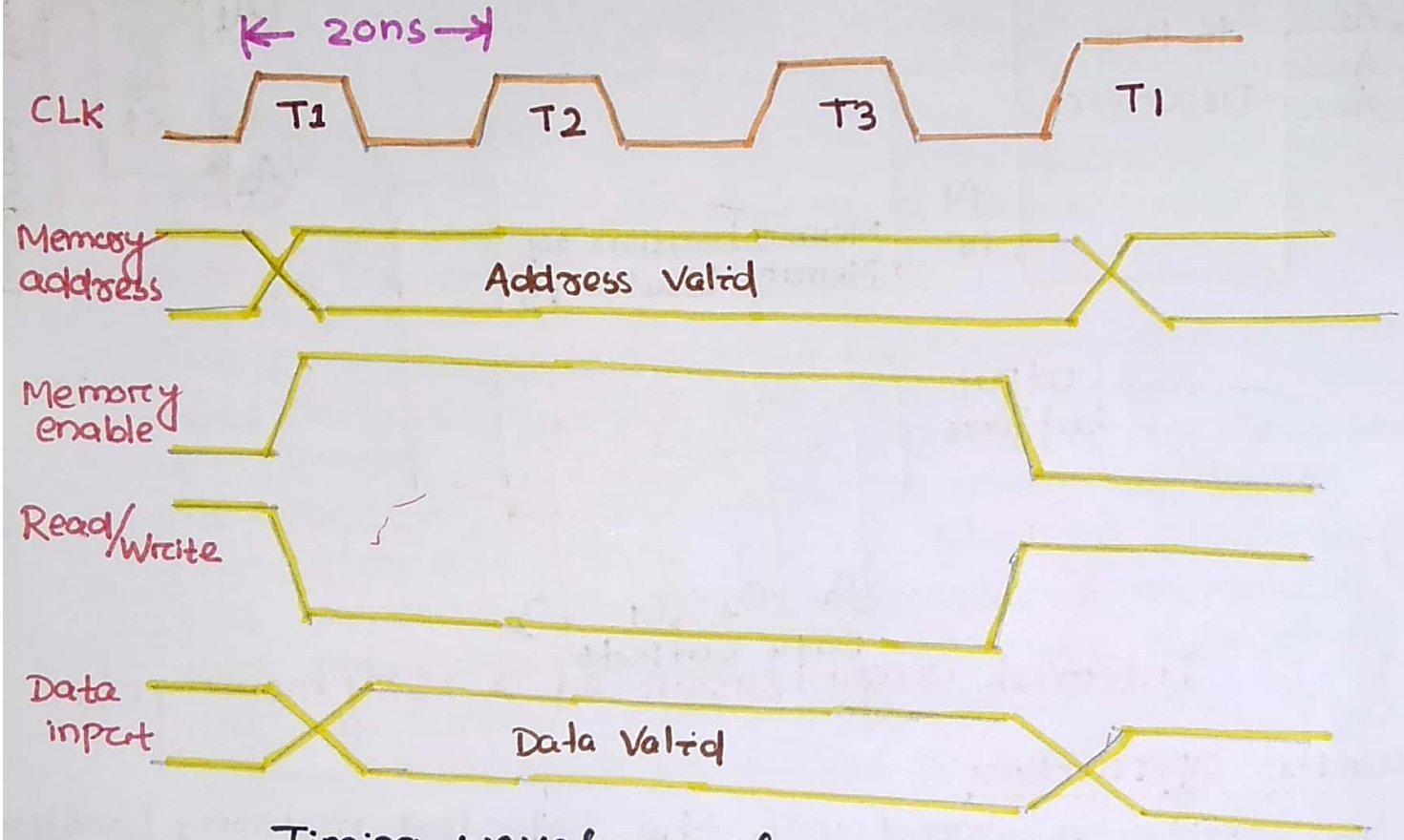
Internal organization of a 16x4 memory chip

Write operation

To write a word into the selected memory Location requires Logic 1 voltage to be applied to CS (chip select) and Write inputs and Logic 0 voltage to Read (RD) input. This combination of inputs give outputs of AND gates A_1 and A_2 1 and 0 respectively. A 1 at the output of A_1 enables the input buffers so that the 4-bit word applied to the data inputs will be Loaded (entered) into the selected memory Location. A 0 at the output of A_2 disables the output buffers so that data outputs are not available.

For writing a word into a particular memory location, the following sequence of operations is to be performed.

1. The chip select signal is applied to the CS terminal.
2. The word to be stored is applied to the data input terminals.
3. The address of the desired memory location is applied to the address-input terminals.
4. A write command is applied to the write-control input terminal with RD=0.



Timing waveforms of Write cycle

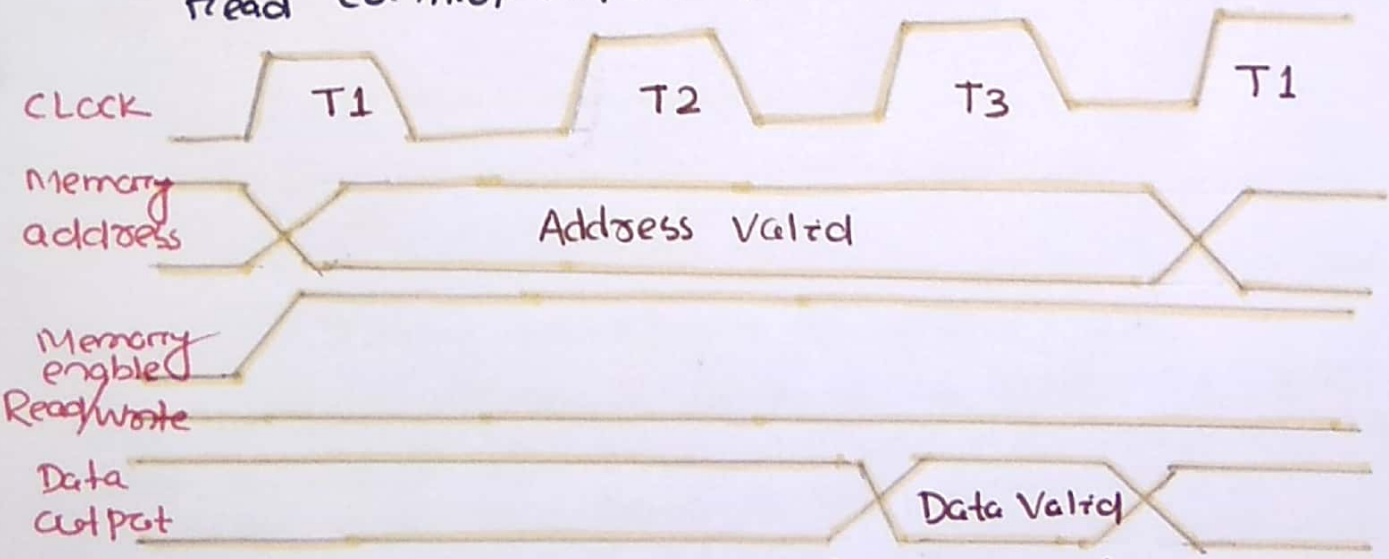
The write cycle shows three 20-ns cycles: T1, T2 and T3. For a write operation, the CPU must provide the address and input data to the memory. This is done at the beginning of T1. The memory enable and the read/write signals must be activated after the signals in the signals in the address lines are stable. The memory enable and read/write signals must switches to the high level and the low level respectively to indicate a write operation.

Read operation

In order to read the contents of a selected memory Location, the Read (RD) and the chip select (CS) inputs must be at Logic 1 level and WR at logic 0 level. This gives output of $A_2 = 1$ which enables the output buffers so that the contents of the selected memory location will appear at the data output.

To read (or retrieve) a data word, known to be stored at a particular address, the following sequence of operations is required to be performed.

1. The chip select signal is applied to the CS terminal.
2. The address of the desired memory location is applied to the address-input terminals.
3. A read-command signal is applied to the read control-input terminal.



Timing waveforms of read cycle

The memory-enable and read/write signals must be in their high level for a read operation. The memory places the data of the word selected by the address into the output data line. The CPU can transfer the data into one of its internal registers during the negative transition of T3

Expanding Memory Size

In many memory applications the required memory capacity i.e. the number of words or word size cannot be satisfied by a single available memory IC chip. Therefore, several similar chips have to be combined suitably to provide the desired number of words or word size.

Expanding Word Size

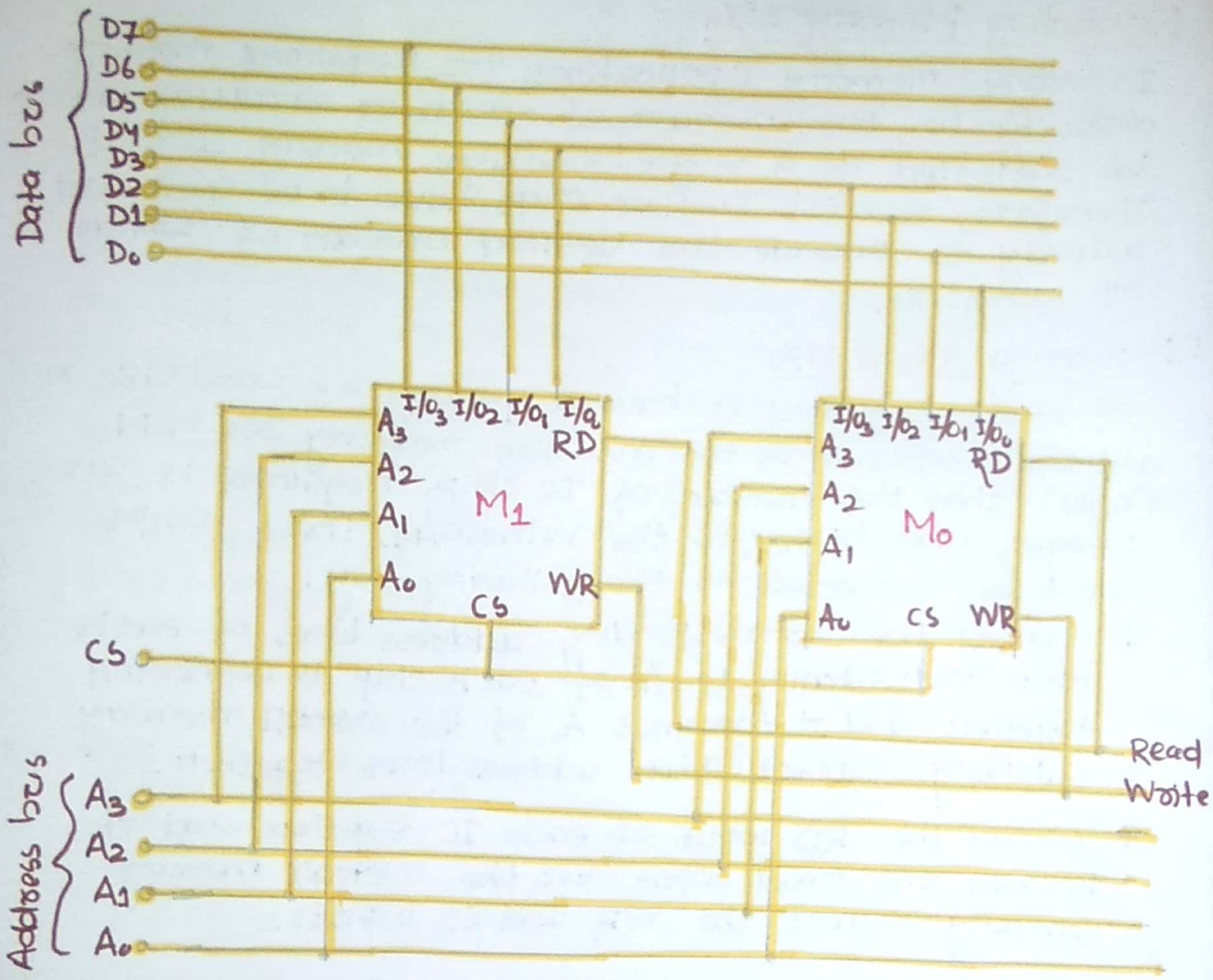
If it is required to have a memory of word size n and the word size of the available memory ICs is N (n>N), then the number of IC chips required is an integer, next higher to the value n/N. These chips are to be connected in the following way:

1. Connect the corresponding address lines of each chip individually ie. A_0 of each chip is connected together and it becomes A_0 of the overall memory. Similarly, connect other address lines together.
2. Connect the RD input of each IC together and it becomes the read input for the overall memory. Similarly, connect the WR and CS inputs.

Ex Obtain a 16x8 memory using 16x4 memory ICs.

Since the word size required is n=8 and the word size of available IC is N=4, therefore n/N = 2 chips are required to obtain the desired memory.

Since each chip can store 16 4-bit words and we want to store 16 8-bit words, each chip is required to store half of each word. Here, we have assumed bidirectional input/output (I/O) lines which is common in many available memory chips. In this 16x8 memory, the higher order four bits (D_7, D_6, D_5, D_4) of each 8-bit word are located in memory M_1 and the lower order four bits (D_3, D_2, D_1, D_0) are located in memory M_0



16x8 memory obtained by combining two 16x4 memory chips.

ii) Expanding Word capacity

Memory chips can be combined together to produce a memory with the desired number of locations. To obtain a memory of capacity m words, using the memory chips with M words each, the number of chips required is an integer next higher to the value m/M. These chips are connected in the following way.

1. Connect the corresponding address lines of each chip individually.
2. connect the RD input of each chip together. Similarly connect the WR inputs.

3. Use a decoder of proper size and connect each of its outputs to one of the CS terminals of memory chips. For example, if eight chips are to be connected, a 3-line-to-8 line decoder is required to select one out of eight chips at any one time.

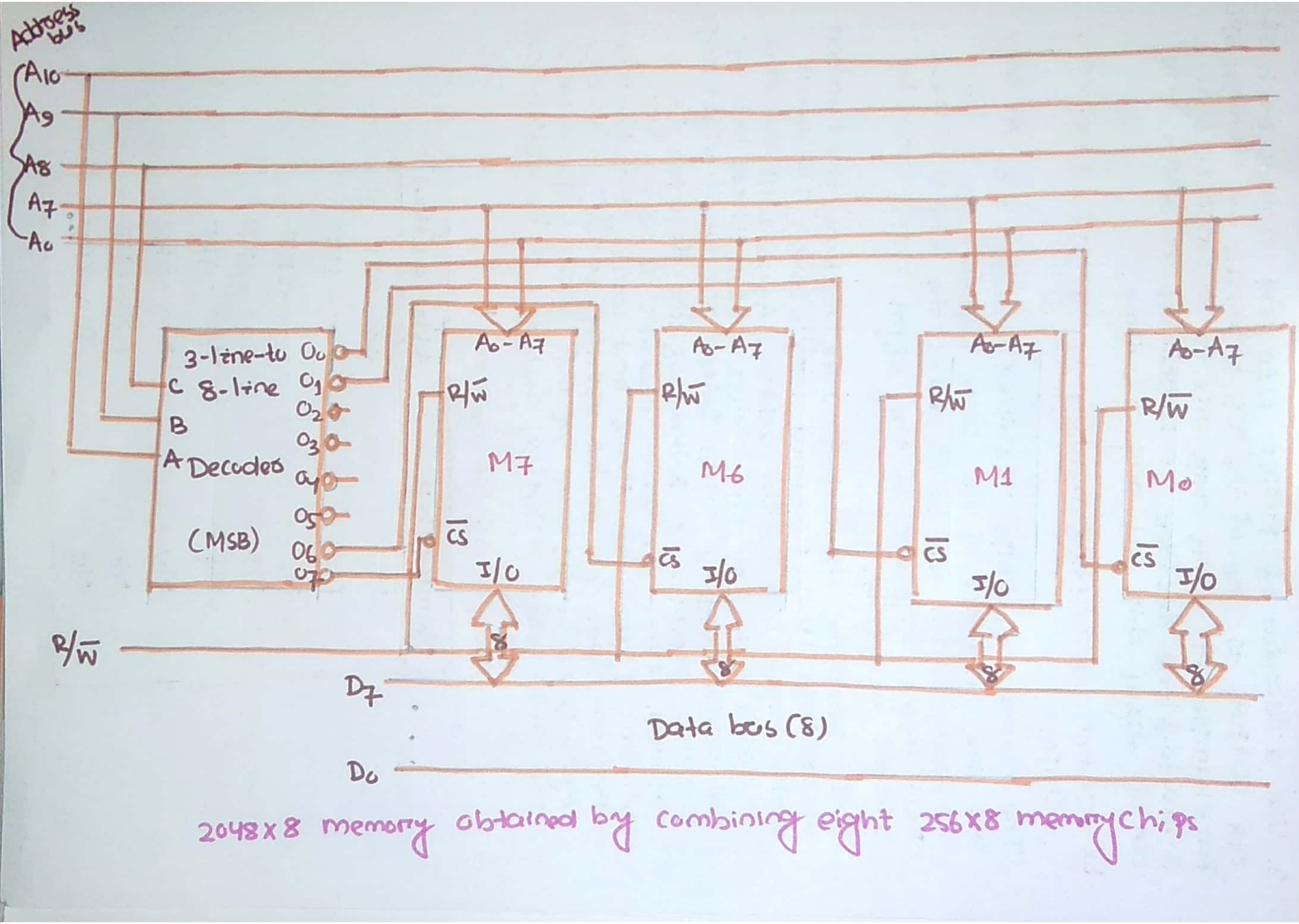
Ex Obtain a 2048×8 memory using 256×8 memory chips.

The number of chips required is $\frac{2048}{256} = 8$. At any one time, only one of the 2048 Locations is to be accessed, which will be in one of the eight chips. That means only one of the eight chips must be selected at a time.

For selecting one out of 2048 Locations, the number of address lines required is $n(2^n = 2048)$. The lower order eight bits of the address A_7-A_0 will be same for each chip and the higher order three bits of the address A_{10}-A_8 must select one out of eight chips. For this purpose a 3-line-to-8-line decoder is required. Here we have assumed a common terminal ($R/\overline{W}$) for read and write. For the read operation, Logic 1 is to be applied to $R/\overline{W}$, whereas Logic 0 is to be applied for the write operation. The chip-select input is assumed to be active-Low.

Addresses of the memory chips

Memory chip	Addresses(Hex.)
M_0	000-0FF
M_1	100-1FF
M_2	200-2FF
M_3	300-3FF
M_4	400-4FF
M_5	500-5FF
M_6	600-6FF
M_7	700-7FF



2048x8 memory obtained by combining eight 256x8 memory chips

Classification and characteristics of Memories

Various memory devices can be classified on the basis of their principle of operation, physical characteristics modes of access, technology used for fabrication etc.

Principle of operation

Memories can be classified according to their principle of operation. The most commonly used memories are

- Sequential memories
- Read and Write memories (RWM or RAM)
- Read-only memories (ROM)
- Content addressable memories (CAM)

Sequential Memories

In the sequential memories, the memory Locations are organised in sequence i.e. one after other, such as a magnetic tap audio/video cassette. Writing into or reading from it is in sequential fashion. Therefore, the time required for accessing a memory Location for writing or reading is different for different Locations. There are two types of semiconductor sequential memories. These are

- shift registers and
- charge coupled devices (CCD)

Shift registers can be either static or dynamic. In a static memory, the contents of the memory Location do not change with time as long as power is on. On the other hand, in dynamic memories, the information is stored in MOS capacitors which changes with time and therefore, it has to be refreshed at regular intervals. The dynamic memories are simpler, less expensive, require less power, have high packing density in comparison to static memories and are therefore widely used in digital systems. However, the cost of the additional circuitary required for refreshing may increase the system cost.

The charge coupled devices are implemented using MOS technology. These devices have high density & Low cost.

Random Access Memories (RWM or RAM)

In this type of memories, the memory locations are organised in such a way so that any memory location requires equal time for writing or reading. This type of memory is also known as read-and-write memory (RWM) or RAM. RAMs can be static or dynamic and are fabricated using bipolar or unipolar technologies.

Read-Only Memories (ROM)

Read-only-memory (ROM), as the name suggests, is meant only for reading the information from it. This does not mean that information is not written into it, because unless some information is stored into it, there cannot be anything to read from. Actually, the process of entering information into this type of memory is much more complicated than for RAM and is done outside the system where it is used. Therefore, it is called as read-only memory. It is used to store information which is fixed, such as tables for various functions, fixed data and instructions. The ROMs are also organised so that every memory location requires equal time for reading the data already stored in that location.

The various types of read-only memories are further sub-classified on the basis of technique employed for storing information into memory (referred to as programming) or their erasability properties. These are:

Read-only Memory

It is programmed at the time of manufacturing as the last process of fabrication, according to the information specified by user. It is referred to as custom programmed or mask programmed. The data stored cannot be changed after fabrication. Since this process is quite costly, therefore this type of ROM is suitable only for bulk requirements in order of millions of chips.

Programmable Read-only Memory (PROM)

This type of ROM is programmed by the user using a special circuit - a PROM programmer. A PROM can be programmed only once after which its contents are permanently fixed as in a ROM. This type of ROM is suitable for storage of data which is of permanent nature. The chip is available without any data stored from the vendor.

Erasable and Programmable ROM

This type of ROMs are reprogrammable ie. it can be programmed again and again. It is referred to as erasable and programmable ROM.

There are two techniques used for erasing; in one technique the chip is exposed to ultraviolet radiation, and in the other technique the contents are altered electrically. The erasable programmable ROM using ultraviolet radiation for erasing is known as EPROM, whereas the device using electrical voltage for erasing is known as electrically alterable ROM (EAROM)

Content Addressable Memories (CAM)

It is a special purpose random access memory which performs association operation in addition to read/write operations.

ii) Physical Characteristics

Memories can be classified according to their physical characteristics, such as:

1. Erasable or non-erasable, and
2. Volatile or non-volatile

Erasable or Non-erasable Memories

A memory in which the information stored can be erased and new information stored is called erasable memory. On the other hand, the information stored in the non-erasable memory cannot be erased, for example ROM is non erasable.

Volatile or Non-Volatile Memories

If the information stored in a memory is lost when electrical power is switched off, the memory is referred to as a volatile memory. For example, the RAM is a volatile memory. On the other hand, in a non-volatile memory, the information once stored remains intact until changed. All the type of ROMs are non-volatile memories.

iii) Mode of Access

Mode of access refers to the manner in which a memory Location is accessed for reading or writing. In case of ROMs only reading is possible. There are two modes of access. These are

- Sequential access
- Random access

Sequential memories are referred to as sequentially accessed memories, whereas RAM, ROM, and CAM are random access memories. In random-access memories any memory location requires equal time for accessing (access time), where as the access time is different for different Locations in case of sequentially accessed memory

iv) Fabrication Technology

Memories can be classified on the basis of the fabrication technology used. The two broad categories of memories based on fabrication technology used are:

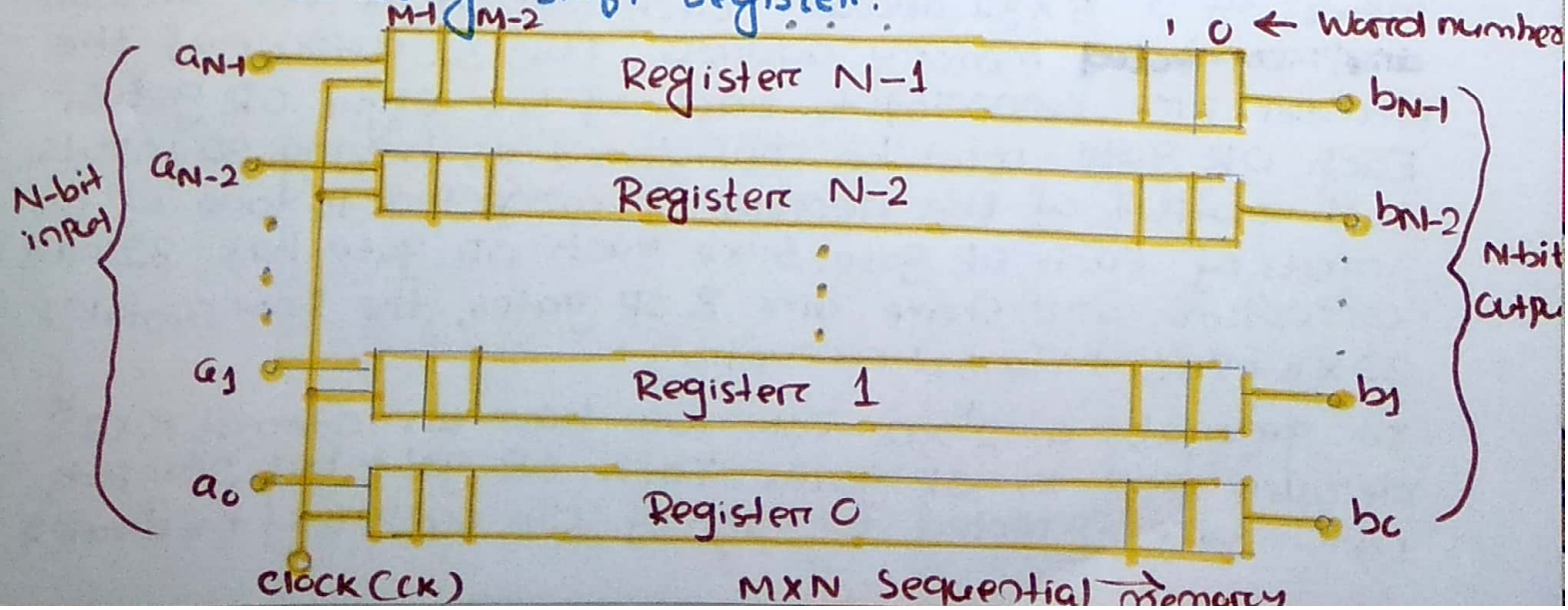
- Bipolar and
- Unipolar (MOS)

Static RAM, ROM and PROM can be fabricated using either bipolar technology (TTL, ECL etc) or MOS technology, whereas dynamic RAM, EPROM and EAROM can be fabricated only using unipolar devices (MOSFETs).

Sequential Memory

In a sequential memory, the words are stored-in and read-out in sequence. For example, if the pth location is being accessed at a particular time, then the (P+q)th location is not accessible, unless all the intermediate locations have been accessed one-by-one in sequence. In other words, the access to a particular location is obtained by waiting until the desired location is reached. This means that the access time is not same for all the locations. Shift registers are examples of sequential memories. Another kind of shift registers gaining popularity in recent years are charge-coupled devices (CCDs) and bubble memories.

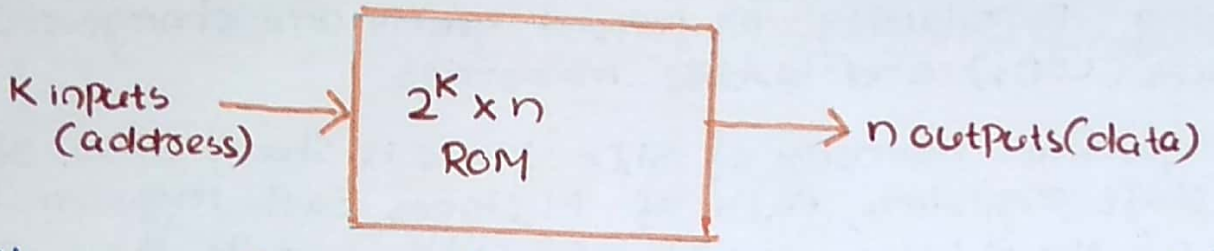
A sequential memory of size M x N is shown below. It requires N shift registers, each of M stages. Each register holds one of the N bits of each of the M words. With each clock cycle, the bits in each register will advance towards the right by one position and stored words will appear sequentially at 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. In contrast, if the outputs are taken from the (M-1)th stage instead of 0th stage, the resulting configuration is called as last-in-first-out (LIFO). The shift registers used in this configuration must have a provision of shifting the bits in either direction. In the FIFO sequential memory system, the word being read may be transferred back to the left-most register position. This is known as circulating shift register.



Read only Memory (ROM)

A read-only memory (ROM) is essentially a memory device in which permanent binary information is stored. The binary information must be specified by the designer and is then embedded in the unit to form the required interconnection pattern. Once the pattern is established, it stays within the unit even when power is turned off and on again.

A block diagram of a ROM consisting of k inputs and n outputs is shown below



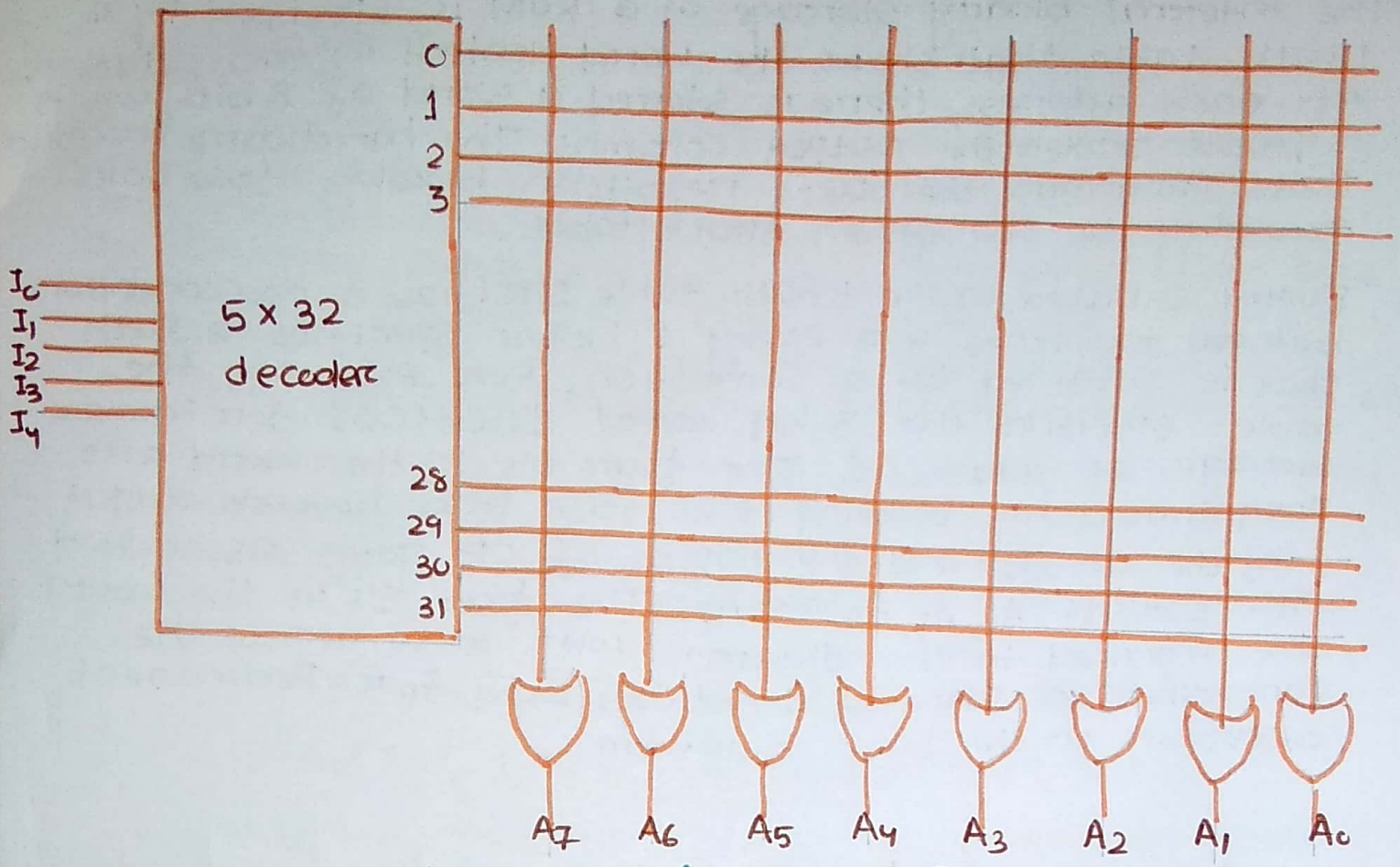
The inputs provide the address for memory, and the outputs give the data bits of the stored word that is selected by the address. The number of words in a ROM is determined from the fact that k address input lines are needed to specify 2^k words. Note that ROM does not have data inputs, because it does not have a write operation.

General Structure of ROM

Consider the example of a 32x8 ROM. The unit consists of 32 words of 8 bits each. There are five input lines that form the binary numbers from 0 through 31 for the address.

The five inputs are decoded into 32 distinct outputs by means of a 5x32 decoder. Each output of the decoder represents a memory address. The 32 outputs of the decoder are connected to each of the eight OR gates. Each OR gate must be considered as having 32 inputs. Each output of the decoder is connected to one of the inputs of each OR gate. Since each OR gate has 32 input connections and there are 8 OR gates, the ROM contains 32x8 = 256 internal connections.

In general, a 2^kxn ROM will have an internal $k \times 2^k$ decoder and n OR gates. Each OR gate has 2^k inputs, which are connected to each of the outputs of the decoder.



Structure of a 32x8 ROM

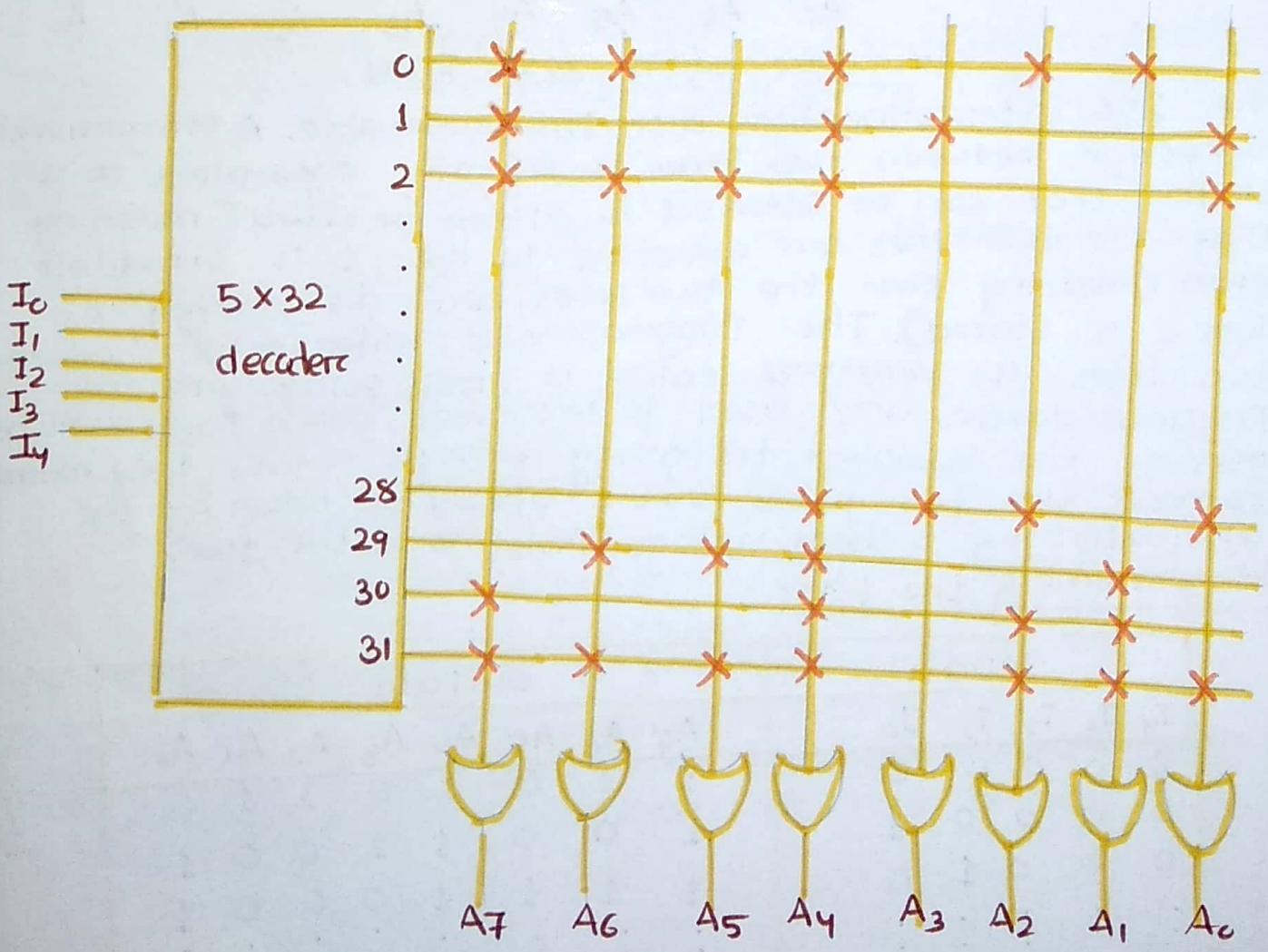
The 256 interconnections are programmable. A programmable connection between two lines is logically equivalent to a switch that can be altered to either be closed (meaning that the two lines are connected ie logic 1 is stored) or open (meaning that the two lines are disconnected ie. logic 0 is stored). The programmable intersection between two lines is sometimes called a cross point. Various physical devices are used to implement cross point switches one of the simplest technology employs a fuse that normally connects the two points, but is opened or blown by the application of a high-voltage pulse onto the fuse.

Programming the ROM

Inputs					Outputs							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	0	0	0	0	1	1	0	1	0	1	1	0
0	0	0	0	1	1	0	0	1	1	0	0	1
0	0	0	1	0	1	1	1	1	0	0	0	1
:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:
1	1	1	0	0	0	0	0	1	1	1	0	1
1	1	1	0	1	0	1	1	1	0	0	1	0
1	1	1	1	0	1	0	0	1	0	1	1	0
1	1	1	1	1	1	0	1	1	0	1	1	1

The internal binary storage of a ROM is specified by a truth table that shows the word content in each address. At each address, there is stored a word of 8 bits, which is listed under the output columns. The hardware procedure that programs the ROM results in blowing fuse links according to the given truth table.

Every 0 Listed in the truth table specifies a no connection between two lines and every 1 Listed specifies a path that is obtained by a connection. For example, the table specifies the 8-bit word 10011001 for permanent storage at address 1. The four 0's in the word are programmed by blowing the fuse links between output 1 of the decoder and the inputs of OR gates associated with outputs A_6, A_5, A_2 and A_1. The four 1's in the word are marked in the diagram with x to denote the connection in place of a dot (x) used for permanent connection in the Logic diagram.



Programmed ROM for the truth table.

Combinational Circuit Implementation

A decoder generates the 2^k minterms of the K input variables. By inserting OR gates to sum the minterms of boolean functions, we generate any desired combination circuit. A ROM is essentially a device that includes both decoder and the OR gates within a single device. By choosing connections of those minterms that are included in the function the ROM output can be programmed to represent the Boolean functions of the output variables in a combinational circuit.

For example, the programmed ROM may be considered as a combinational circuit with eight outputs, each being a function of the five input variables. output A_7, A_6 can be expressed in sum of minterms.

$$A_7(I_4, I_3, I_2, I_1, I_0) = \sum (0, 1, 2, \ldots 30, 31)$$

$$A_6(I_4, I_3, I_2, I_1, I_0) = \sum (0, 2, \ldots, 29, 31)$$

A connection marked with x in the figure produces a minterm for the sum. All other cross points are not connected and are not include in the sum.

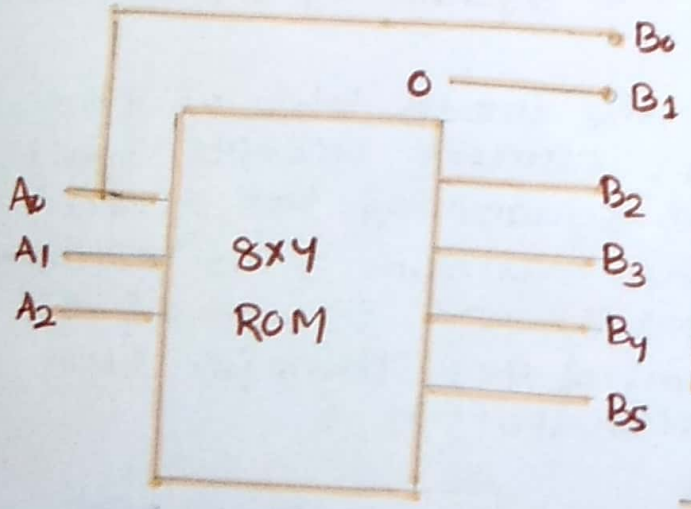
Design a combinational circuit using a ROM. The circuit accepts a 3-bit number and generates an output binary number equal to the square of the input number.

The first step is to derive the truth table of the combinational circuit. This circuit accepts 3-bit binary and produces the square of the given 3-bit binary number. The maximum 3-bit number is 111 (decimal equivalent 7) and square of 111 is 11001 (decimal equivalent of 49). Therefore the required input is 3 and output is 6.

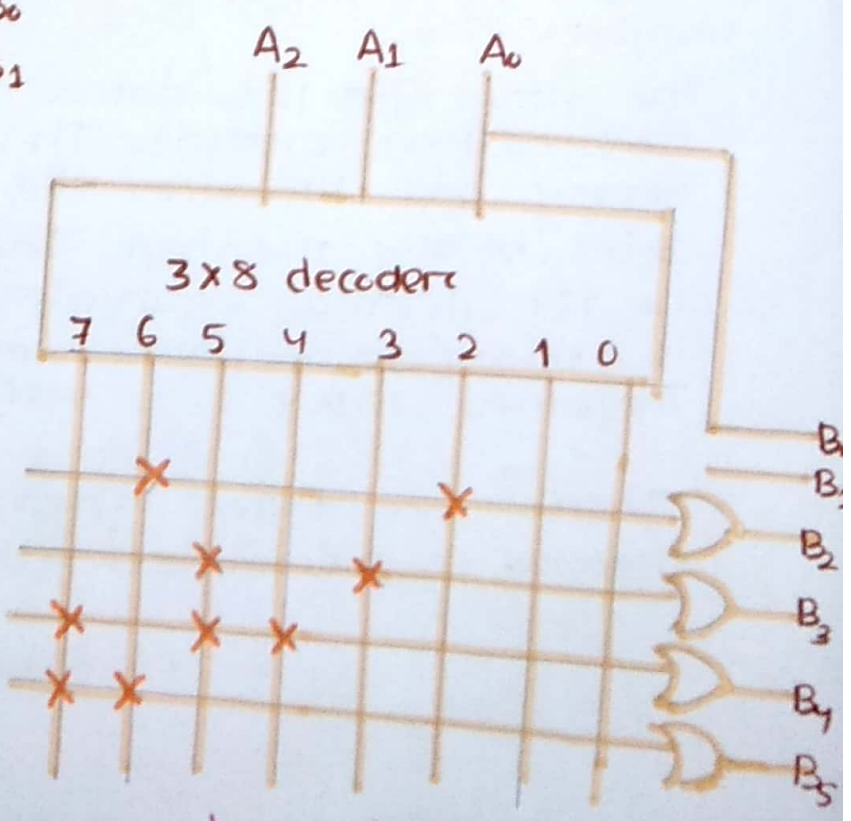
There are three inputs and six outputs are needed to accomodate all possible binary numbers.

Inputs				Outputs						
A_2	A_1	A_0	Decimal equivalent	B_5	B_4	B_3	B_2	B_1	B_0	Decimal equivalent
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	1	1
0	1	0	2	0	0	0	1	0	0	4
0	1	1	3	0	0	1	0	0	1	9
1	0	0	4	0	1	0	0	0	0	16
1	0	1	5	0	1	1	0	0	1	25
1	1	0	6	1	0	0	1	0	0	36
1	1	1	7	1	1	0	0	0	1	49

We note that output B_0 is always equal to input A_0. So there is no need to generate B_0 with a ROM since it is equal to an input variable. Moreover, the output B_1 is always 0. So this output is known as constant. Therefore, we need to generate only four outputs with ROM. The minimum size ROM needed must have three inputs and four outputs. Three inputs specify eight words, so the ROM must be of size 8 x 4.



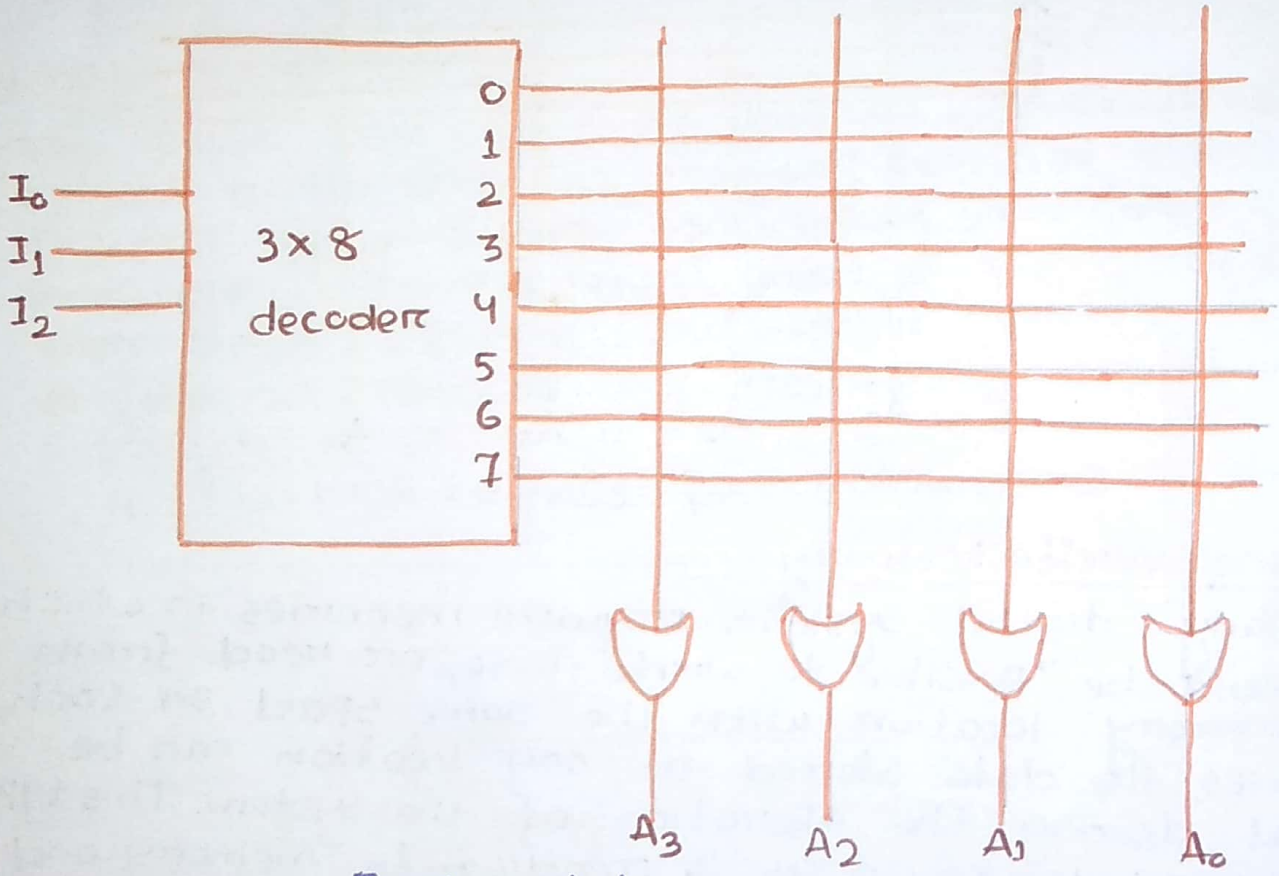
Block diagram



Logic diagram

Q Give the Logic implementation of a 8x4 bit ROM using a decoder of a suitable size.

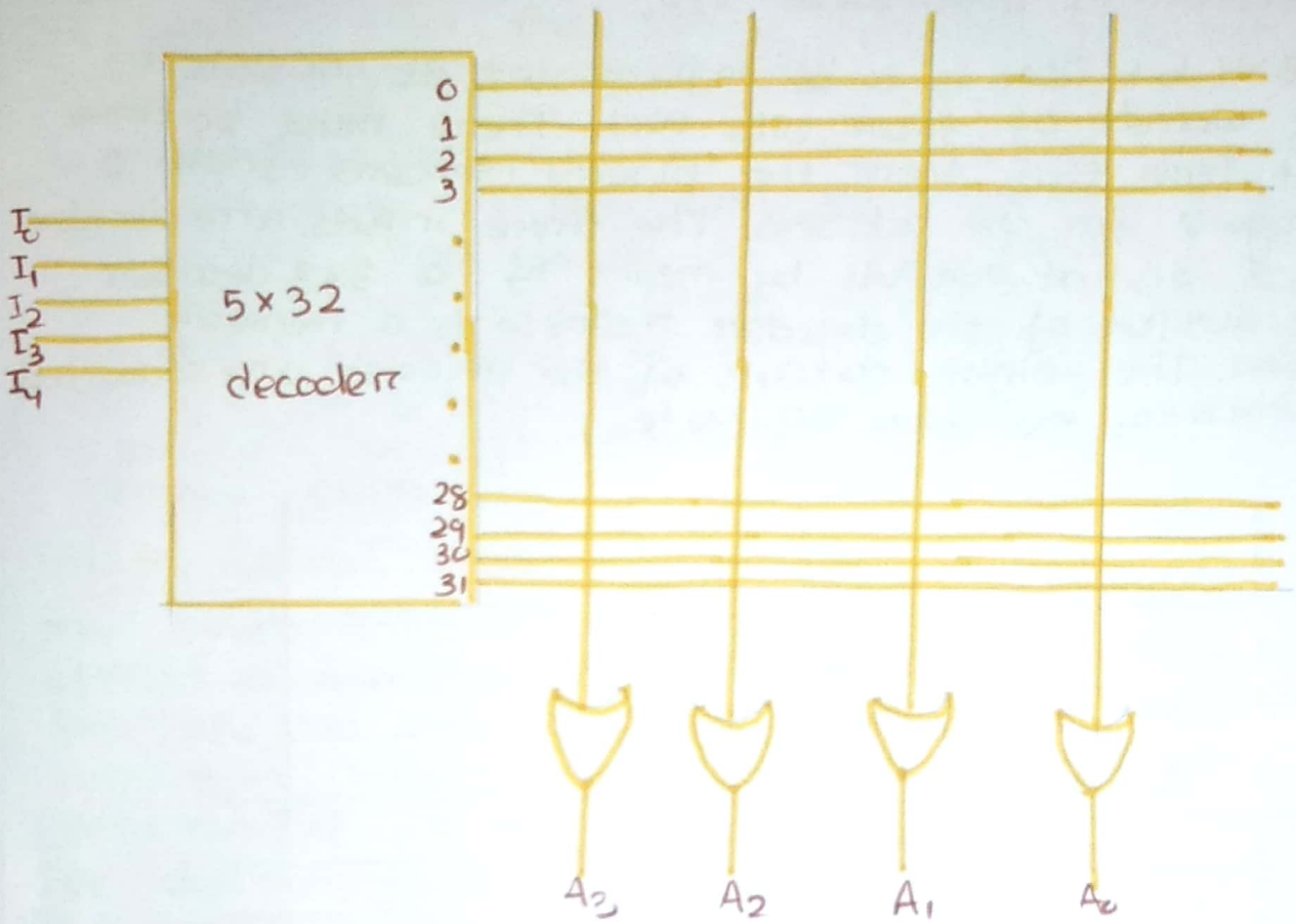
An 8x4 bit ROM is to be implemented. It consists of eight words of four bits each. There must be three input lines that form the binary numbers from 0 through 7 for the address. The three inputs are decoded into 8 distinct outputs by means of a 3x8 decoder. Each output of the decoder represents a memory address. The eight outputs of the decoder are connected to each of the four OR gates.



Implementation of 8x4 bit ROM

Q Give the logic implementation of a 32x4 bit ROM using a decoder of a suitable size.

A 32x4 bit ROM is to be implemented. It consists of 32 words of four bits each. There must be five input lines that form the binary numbers from 0 through 31 for the address. The five inputs are decoded into 32 distinct outputs by means of a 5x32 decoder. Each output of the decoder represents a memory address. The 32 outputs of the decoder are connected to each of the four OR gates.



Implementation of 32x4 bit ROM

Read and Write Memory

Many digital systems require memories in which it should be possible to write into, or read from any memory location with the same speed. In such memories, the data stored at any location can be changed during the operation of the system. This type of memory is known as a read/write memory and is usually referred to as RAM (random-access memory).

The RAMs are used in computers for the temporary storage of programs and data. The contents of many RAM address locations are read from and written to as the computer executes a program. This requires fast read and write cycle times for the RAM so as not to slow down the computer operation. A major disadvantage of RAMs is that they are volatile and lose all stored information if power is interrupted and turned off. Some CMOS RAMs, however, use such small amounts of power in the standby mode (when no read and write operations take place)

that they can be powered from batteries whenever the main power is interrupted. of course, the main advantage of RAMs is that they can be written into or read from rapidly with equal ease.

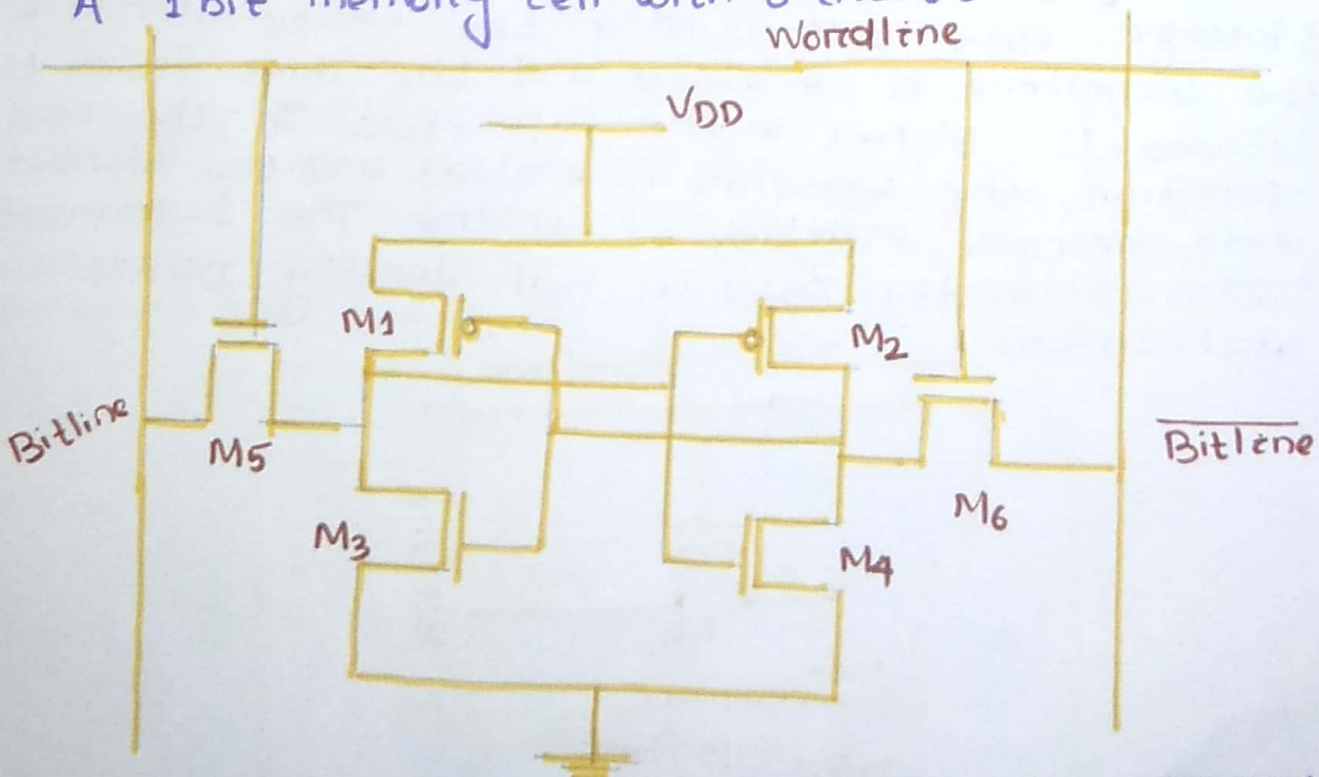
Like the ROM, the RAM can also be thought of as consisting of a number of registers, each storing a single data word and having a unique address. The RAMs typically come with word capacities of 1K, 4K, 8K, 16K, 64K, 128K, 256K and 1024K, and the word size of 1, 4, or 8-bits. The word capacity and word size can be expanded by combining several memory chips.

i) Static RAMS (SRAMs)

The static RAM can store data as long as power is applied to the chip. Its memory cells are essentially flip-flops that will stay in a given state (store a bit) indefinitely, provided that power to the circuit is not interrupted. SRAM is also called Volatile memory. It does not require any refreshing operation.

6-transistor SRAM Cell

A 1 bit memory cell with 6 transistors is shown below



Two CMOS inverters are connected back to back. Transistor M_1 & M_3 form one inverter and transistors M_2 & M_4 form another inverter. The transistor M_5 & M_6 are used as switch controlled by wordline

operation

i) Read operation

To read data from the cell wordline is enabled. This makes transistors M_5 and M_6 ON. Hence stored data is available in both the true and complemented form in the Bitline and $\overline{\text{Bitline}}$ respectively

ii) Write operation

To write data into the cell, again wordline is enabled. The data to be written is made available in the bitlines. The data is stored in the Latch.

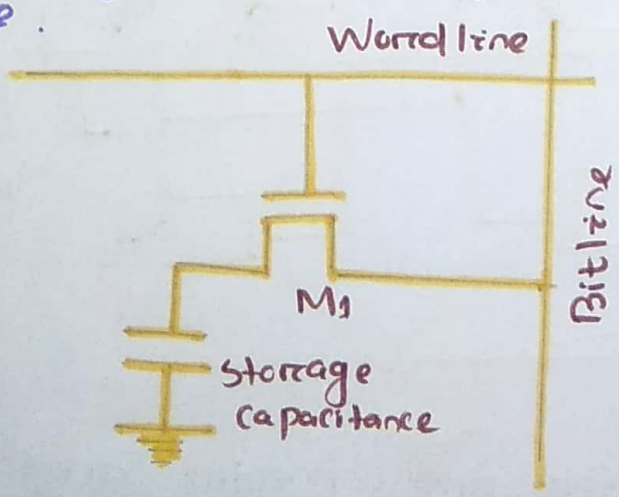
ii) Dynamic RAMs (DRAMs)

In DRAM, the binary data is stored in the parasitic capacitance which discharges with time. In order to retain the stored data, the capacitor must be charged periodically This phenomenon is called dynamic refreshing of DRAM.

DRAM can be implemented using four, three and one transistors (s).

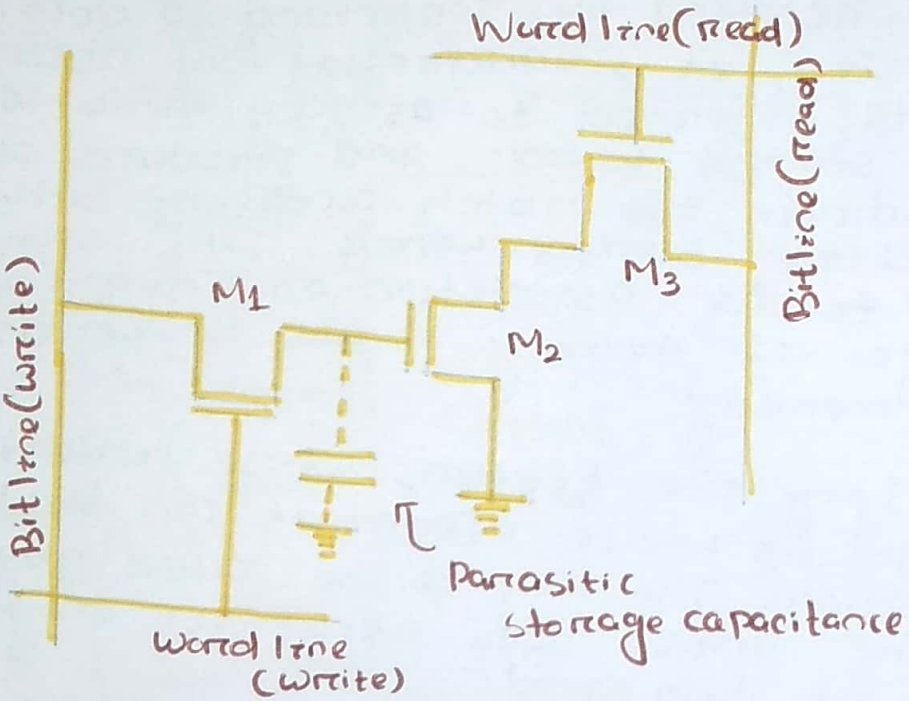
1-transistor DRAM

The 1-transistor DRAM cell uses a dedicated storage capacitor. During the write operation, the wordline is enabled and the data from the bitline is stored in the capacitor. In the read operation, the wordline is enabled and the stored data becomes available at bitline. The 1-transistor DRAM is mostly used in high density DRAM architecture.



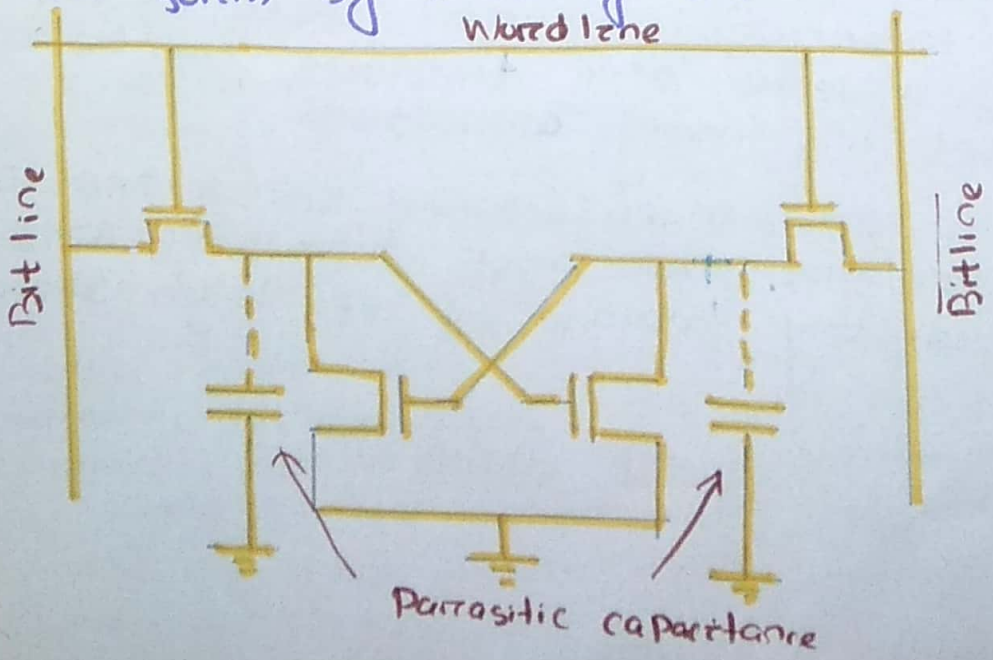
3-transistor DRAM

In 3-transistor DRAM architecture, there are three nMOS transistors. Only M_2 is storing the binary data in association with its parasitic gate capacitance. In the write mode, the write word line is enabled, the logic from the write bit line is passed to the parasitic storage capacitance. In the read mode, read word line is enabled, and the complement of the stored data becomes available in the read bit line.



4-transistor DRAM

The 4-transistor DRAM structure has four nMOS transistors. The binary data is stored in the parasitic capacitances. The data is written in true and complemented form by enabling the word line.



As there is no restoring path from V_{DD} to these capacitances, the stored charge is lost. So to retain the logic level, the capacitance must be refreshed periodically. During the read operation, the wordline is enabled and stored data becomes available at the bitlines both in true and complemented form.

Content Addressable Memory (CAM)

The content addressable memory (CAM) is a special purpose random access memory device that can be accessed by searching of data content. For this purpose, it is addressed by associating the input data, referred to as key, simultaneously with all the stored words and produces output signals to indicate the match conditions between the key and the stored words. This operation is referred to as association or interrogation and this type of memory is also known as associative memory.

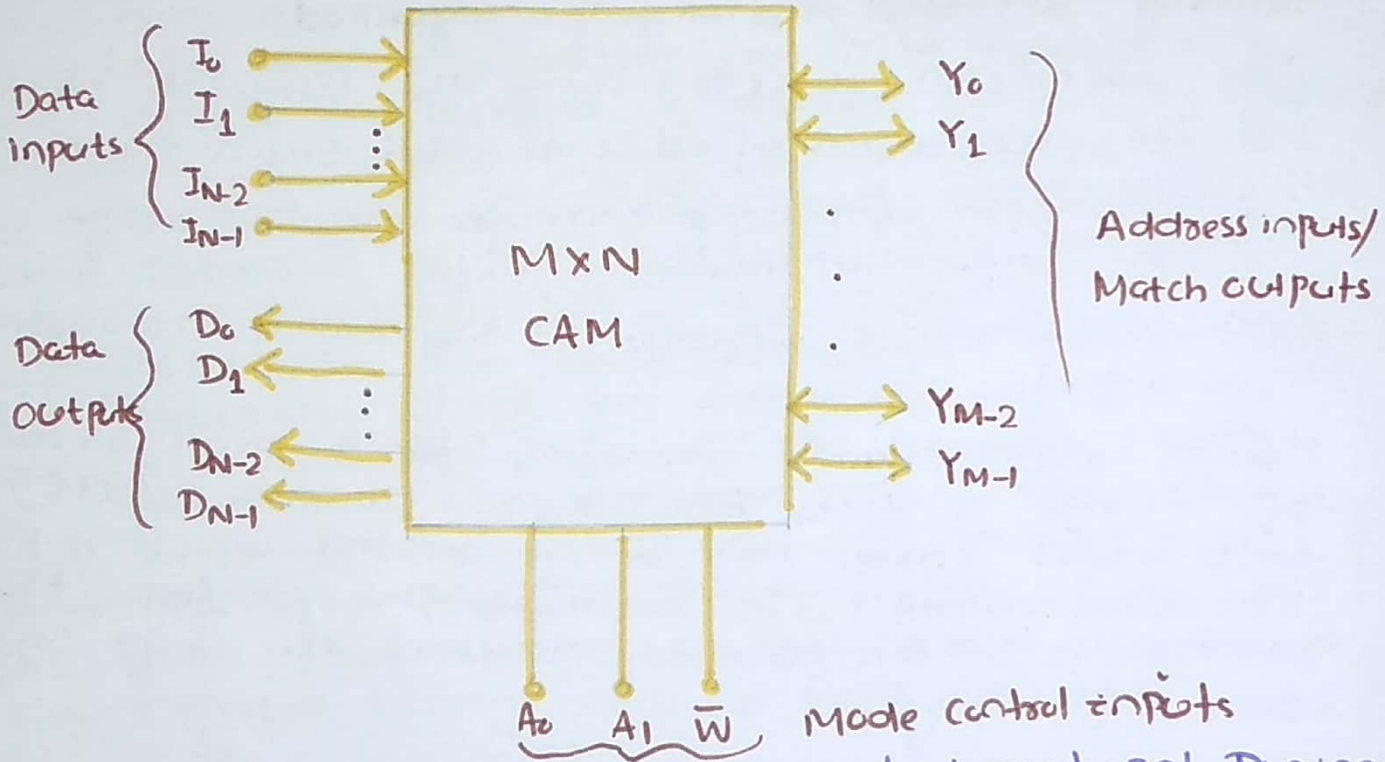
After identifying the locations whose contents match the key, read or write operations can be performed to these locations. The key to be used may either consist of the entire data word or only some specific bits of data word.

A CAM differs from the conventional memory organization in that addressing of a location in latter has no relation to the memory content. A CAM has the ability to search or interrogate stored data on the basis of its contents and therefore, can be a powerful asset in many applications.

CAMs are manufactured using MOS, CMOS or bipolar technologies. The most popular CAMs use ECL circuitary because of its high speed operation.

Operation of CAM

A CAM can perform three basic operations: read, write and associate. Figure shows the block diagram of a CAM. Its storage capacity is M x N bits and is organized as M words of N bits each. It has N data input and N data output lines (one line for each bit of a word). The data input lines I_0 through I_{N-1} are used to input data to be written into the memory and for key word in case of associate operation. Data are read out of the CAM at the data output lines D_0 through D_{N-1}



The Y lines (Y_0 through Y_{M-1}) are bidirectional. During a read or write operation, these lines are used to select the storage location. There is one address input line for each word in the CAM. For example, Y_0 is the address line for memory location 0, Y_1, for memory location 1 and so on.

The Y lines serve as match output lines one for each memory location, when an association operation is performed. For example, if the key word matches with the word stored in memory location 5 & 8, lines Y_5 and Y_8 will become HIGH to indicate the match condition.

The mode control inputs are used to select the required operation. The read and write operations are performed in a manner similar to that used for RAM. However, during write operation, the input data also appears at the data outputs.

Charge coupled device memory

The charge coupled device (CCD), a new concept for storage of digital information, was announced in early 1970 by Bell Telephone Laboratories of U.S.A. It is a array of MOS capacitors operating as a dynamic shift register. CCDs are simple, versatile, and low cost devices and can be used wherever a serially accessed memory is required.

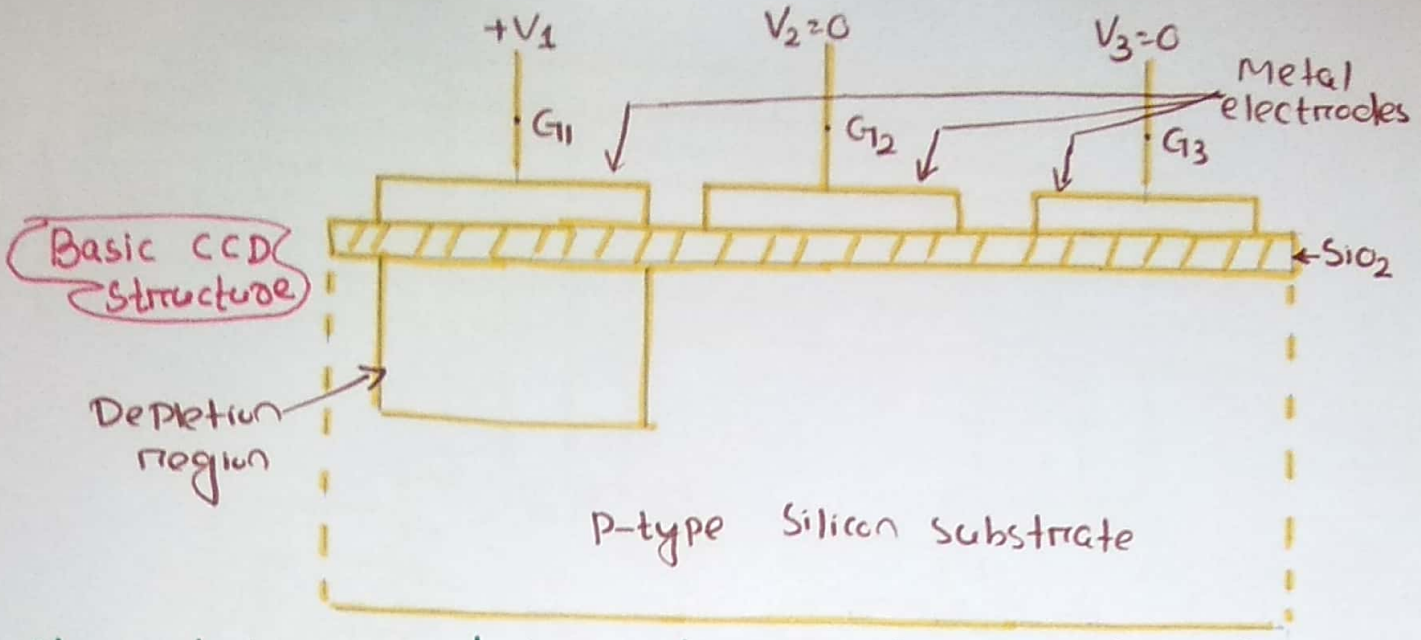
The operation of CCDs involve the following steps

1. The conversion of digital input signal into charge
2. Transfer of charge through various stages in sequential manner and
3. conversion of charge at the output into a digital signal.

During each charge transfer step, a small amount of charge is lost. Also, due to thermal effects, undesirable charge may be generated which is known as dark current. To overcome these defects, the charge is recirculated around the shift register for refreshing.

Basic concept of CCD

Consider a P-type Silicon substrate covered with a thin oxide Layer and closely spaced metallic electrodes. Each metallic electrode (gate) and the substrate form a MOS capacitor which can store charge. If a positive voltage is applied at a gate electrode, a depletion region is formed in the substrate immediately under the metallic electrode. This happens due to the repulsion of free holes in the substrate because of the positive voltage at the gate electrode.



These holes are driven downward away from the oxide Layer and consequently immobile negetive ions are exposed and a depletion region comes into existence. In figure, a positive voltage V_1 is applied at the G_1 gate and the other two gates are held at the same potential as the substrate. The depletion region is indicated below the gate G_1. This plot also represents the potential-energy barrier (well) for electrons, which are the minority charge carriers. Now, if a packet of negetive charge is injected into the depletion region these charges can move freely within the well, but cannot penetrate the potential-energy walls of the well. This means that as long as the voltage V_1 is present, the negetive charge will be held (trapped) there.

In a conventional capacitor, the charges are held on conducting plates, whereas in a MOS device the charges are held on a conductor and in the depletion region under the conductor. The stored charge can be moved from left to right down the channel by applying voltages at the gates in a proper sequence. We assume that a Logic 1 is stored when a negetive charge is held in the depletion region and a logic 0 is stored when the depletion region is empty. This type of operation makes it possible to make long shift registers using these devices

Programmable Logic device (PLD)

A Programmable Logic device is an IC that is user configurable and is capable of implementing Logic functions. In this Logic device the Logic function is programmed by the user and in some cases, can be reprogrammed many times. It is programmed by the user to perform a function required for application.

Fixed-function ICs are those ICs in which each one performs a specific, fixed function. For example, ICs Performing basic digital operation and functions such as multiplexers, demultiplexers, adders, comparators, code converters, shift registers, counters etc are fixed-function ICs. These devices are designed by the manufacturers and are manufactured in Large quantities to meet the needs of a wide variety of applications & are readily available.

PLDs have the following advantages over fixed-function ICs

i) Reduction in board space requirements
ii) Reduction in power requirements
iii) Design security
iv) Compact circuitry
v) Higher switching speed

The Application specific integrated circuits (ASICs) are designed by the users to meet the specific requirements of a circuit and are produced by an IC manufacturer as per the specifications supplied by the user. Usually the designs are too complex to be implemented using fixed-function ICs.

PLDs have many of the advantages of ASICs as given below

i) Higher densities
ii) Lower quantity production costs
iii) Design security
iv) Reduced power requirement
v) Reduced space requirement

Types of PLDs

The three major types of programmable logic devices are SPLD, CPLD and FPGA.

SPLDs (simple programmable Logic devices)

SPLDs are the least complex form of PLDs. An SPLD can typically replace several fixed function SSI or MSI devices and their interconnections. The SPLD was the first type of programmable logic available. A few categories of SPLDs are

- PAL (Programmable array logic)
- GAL (generic array logic)
- PLA (Programmable logic array)
- PROM (programmable read-only memory)

CPLDs (complex programmable logic devices)

CPLDs have a much higher capacity than SPLDs, permitting more complex logic circuits to be programmed into them. A typical CPLD is the equivalent of from two to sixty-four SPLDs. There are several forms of CPLD, which vary in complexity and programming capability. CPLDs typically come in 44-pin to 160-pin packages depending on the complexity

FPGAs (field-programmable gate arrays)

FPGAs are different from SPLDs and CPLDs in their internal organisation and have the greatest logic capacity. FPGAs consist of an array of any where from sixty-four to thousands of logic-gate groups that are sometimes called logic blocks. FPGAs come in packages ranging up to 1000 pins or more.

PLD Programming

A logic circuit design for a PLD is entered using one of the two basic methods: schematic entry or text-based entry. Some times, a combination of both methods is used.

In the schematic entry method, the software allows the user to enter a logic design using logic components (e.g. logic gates, flip-flops) and to interconnect them on the computer screen to form a schematic diagram.

In the text-based entry method, also known as language-based entry, the software allows the user to enter a logic design in the form of text using a hardware description Language (HDL). Several HDLs are available, such as VHDL and Verilog HDL developed for programming PLDs and are widely used.

An HDL that is becoming widely used, especially for programming CPLDs and FPGAs is VHDL, a standard developed by the Department of Defense and adopted by the IEEE (Institute of Electrical and Electronics Engineers). The latest version of VHDL is IEEE Std 1076-1993. Verilog is another popular HDL for programming CPLDs and FPGAs

Simple programmable Logic Devices (SPLDs)

Programmable logic devices (PLDs) are used in many applications to replace fixed-function circuits. They save space and reduce the actual number and cost of devices in a given design.

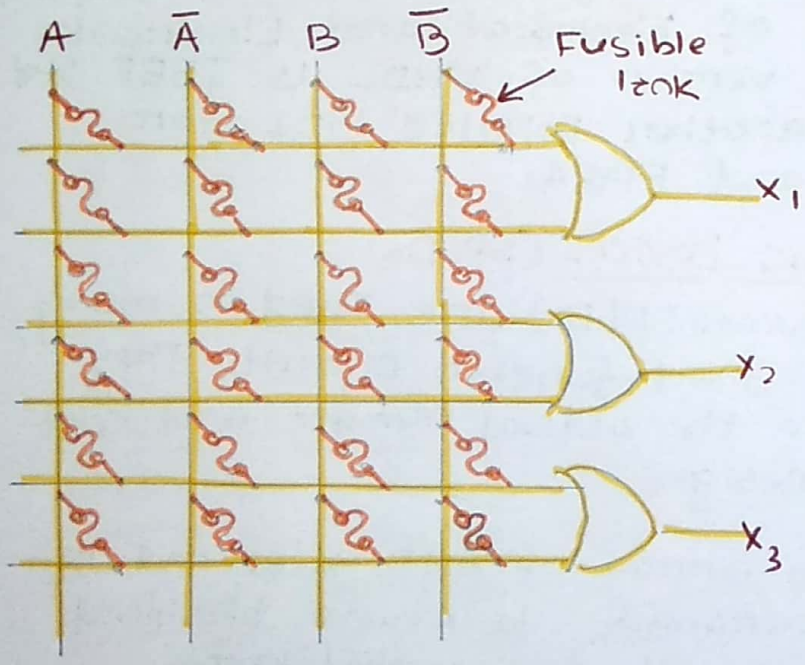
An SPLD consists of an array of AND gates and OR gates that can be programmed to achieve specified logic functions. Four types of devices that are classified as SPLDs are the programmable array Logic (PAL), the generic array Logic (GAL), the programmable Logic array (PLA) and the programmable read-only memory (PROM).

Programmable Arrays

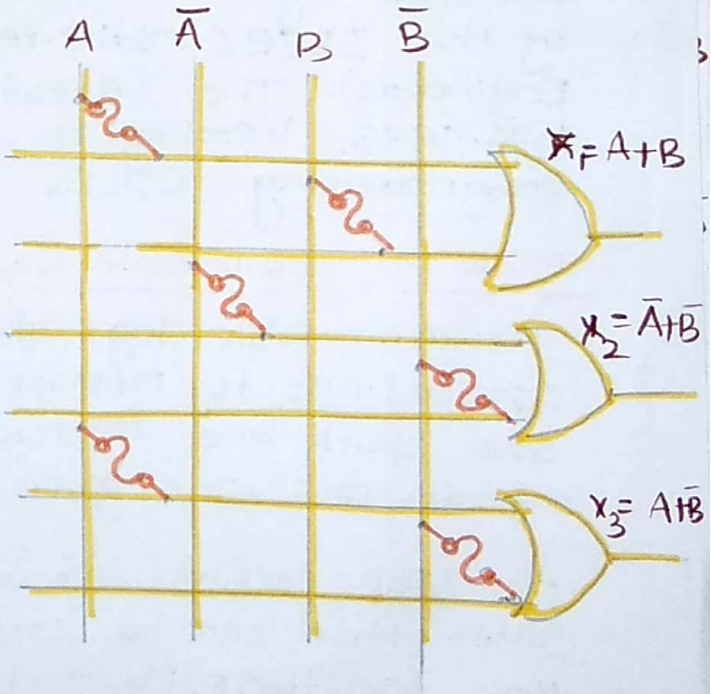
All PLDs consist of programmable arrays. A programmable array is essentially a grid of conductors that form rows and columns with a fusible link at each cross point. Arrays can be either fixed or programmable. The earliest type of programmable array back in 1960s, was a matrix with a diode at each cross point of the matrix.

The OR Array

The original diode array evolved into the integrated OR array, which consists of an array of OR gates connected to a programmable matrix with fusible links at each cross point of a row and column. The array can be programmed by blowing fuses to eliminate selected variables from the output functions. For each input to an OR gate, only one fuse is left intact in order to connect the desired variable to the gate input. Once a fuse is blown, it cannot be reconnected.



Unprogrammed

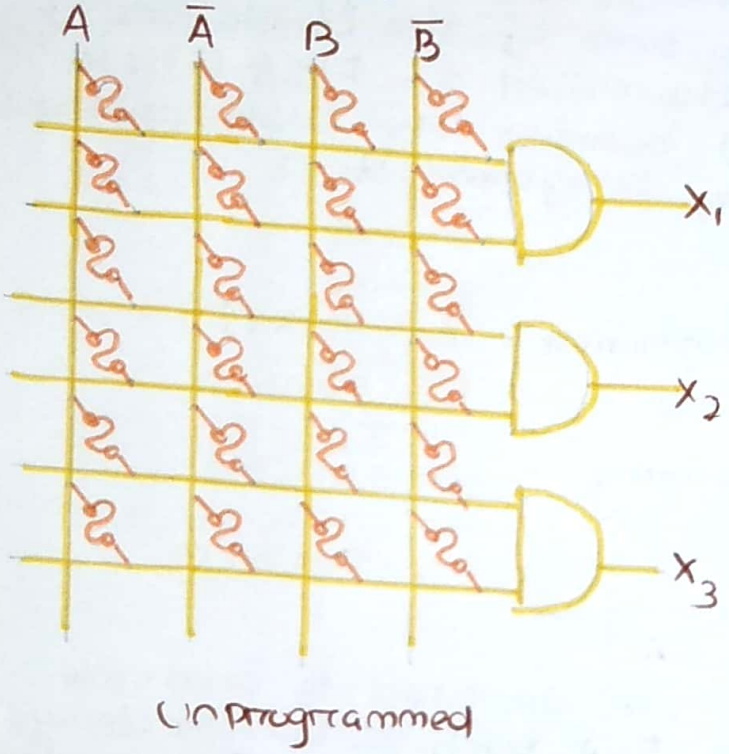


Programmed

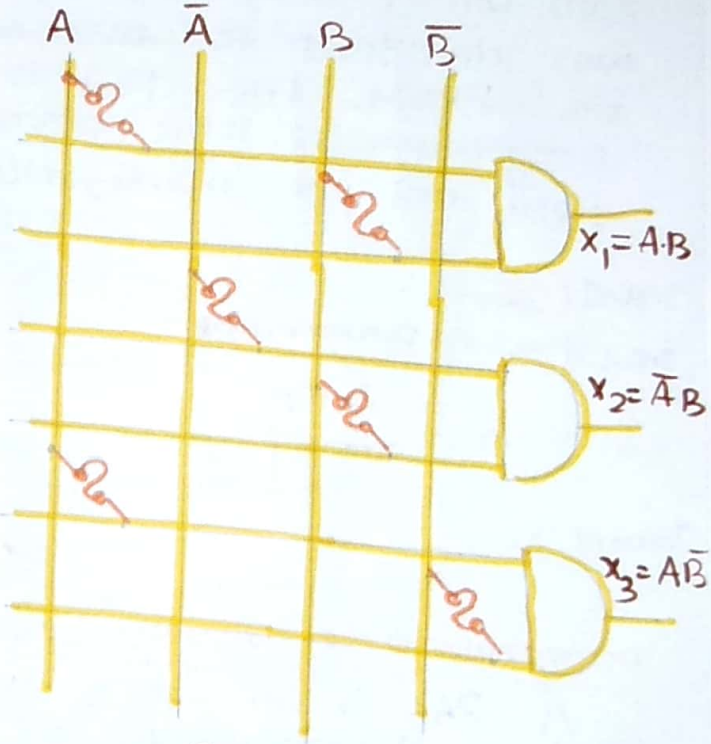
Another method of programming a PLD is the antifuse, which is the opposite of the fuse. Instead of a fusible link being broken or opened to program a variable, a normally open contact is shorted by melting the antifuse material to form a connection.

The AND Array

This type of array consists of AND gates connected to a programmable matrix with fusible links at each cross point. Like the OR array, the AND array can be programmed by blowing fuses to eliminate variables from the output function. For each input to an AND gate, only one fuse is left intact in order to connect the desired variable to the gate input. Also, like the OR array, the AND array with fusible links or with antifuses is one-time programmable.



Unprogrammed



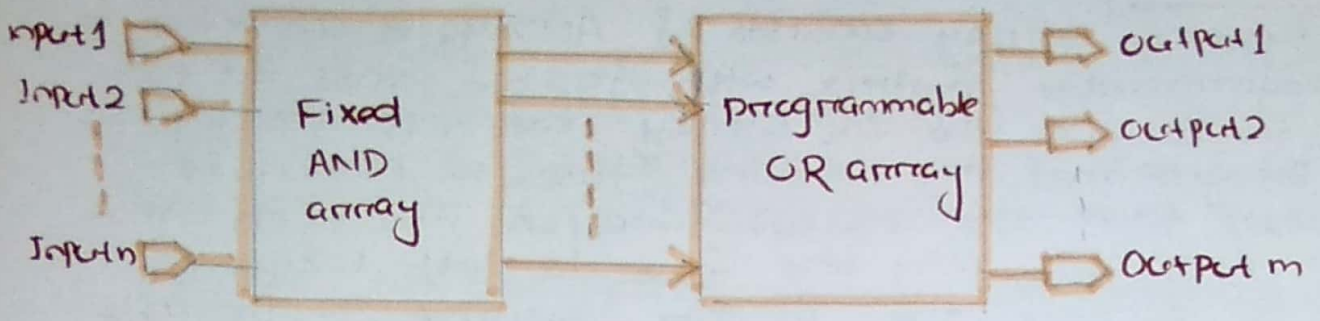
Programmed

Classification of SPLDs

SPLDs are classified according to their architecture, which is basically the internal functional arrangement of the elements that give a device its particular operating characteristic.

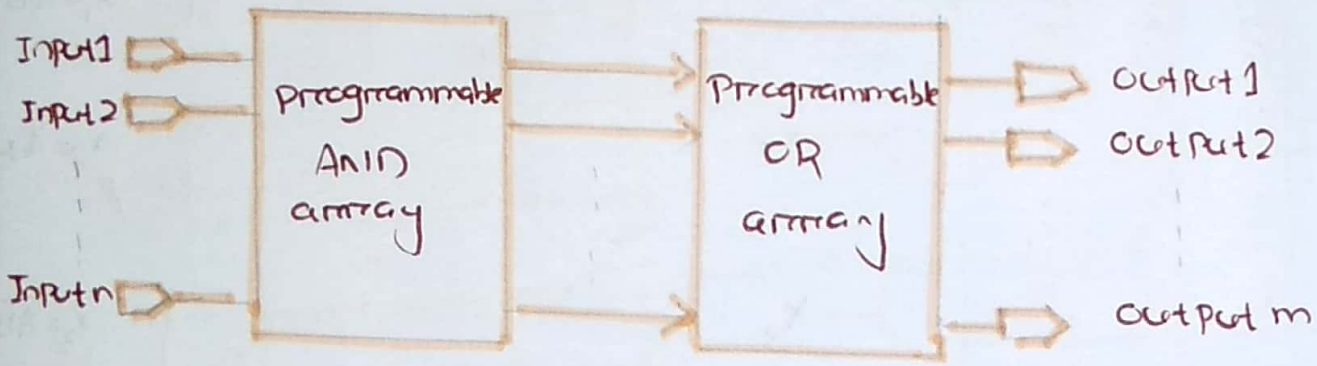
Programmable Read-only Memory (PROM)

A PROM consists of a set of fixed (nonprogrammable) AND gates connected as a decoder and a programmable OR array. The PROM is used primarily as an addressable memory and not as a logic device because of limitations imposed by the fixed AND array.



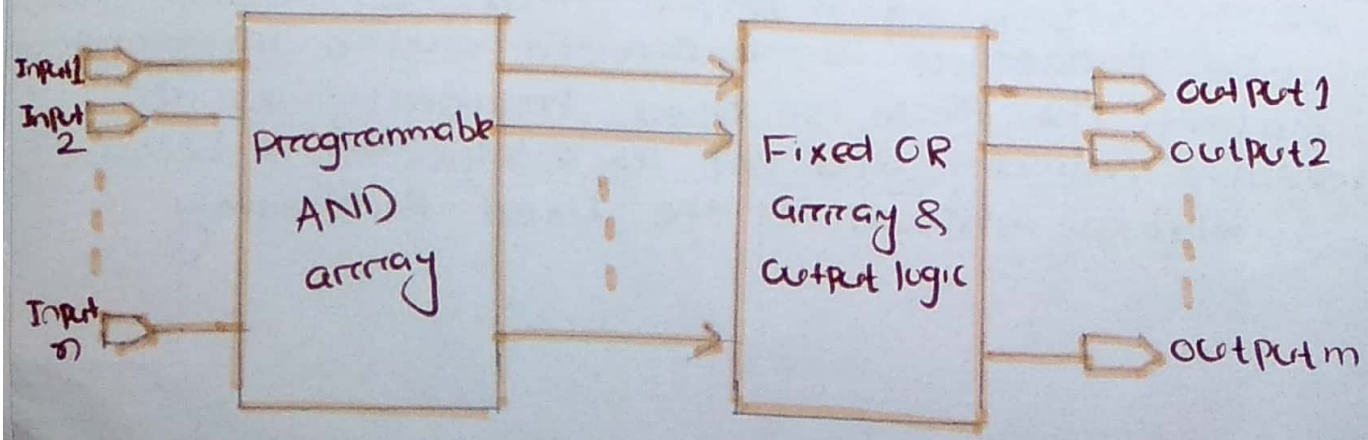
Programmable Logic Array (PLA)

A PLA is an SPLD that consists of a programmable AND array and a programmable OR array. The PLA was developed to overcome some of the limitations of the PROM. The PLA is also called an FPLA (field programmable logic array) because the user in the field, not the manufacturer, programs it.



Programmable Array Logic (PAL)

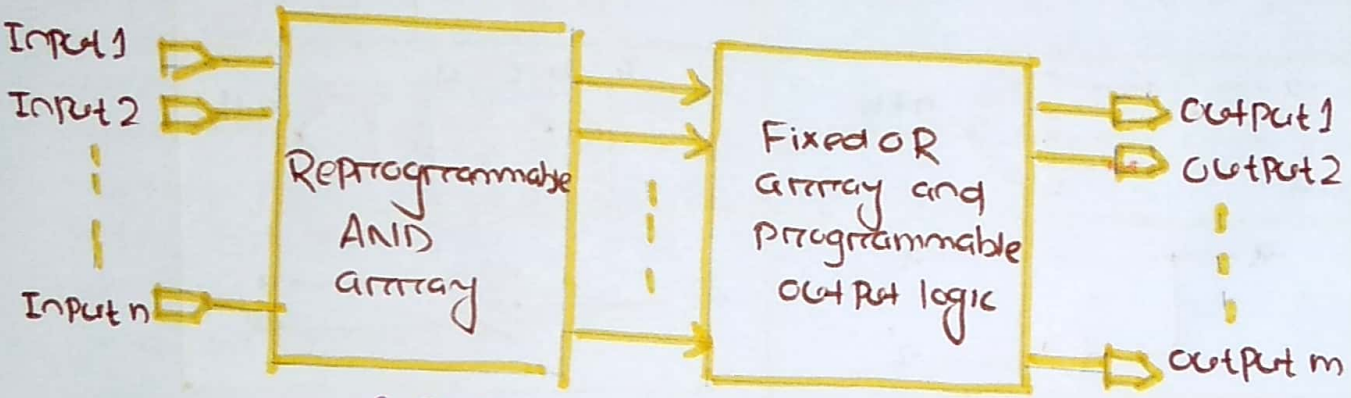
A PAL is an SPLD that was developed to overcome certain disadvantages of the PLA, such as longer delays due to additional fusible links that result from using two programmable arrays and more circuit complexity. The basic PAL consists of a programmable AND array and a fixed OR array with output logic. The PAL is the most common one-time programmable (OTP) logic device and is implemented with bipolar technology (TTL or ECL).



Generic Array Logic (GAL)

The GAL has a reprogrammable AND array and a fixed OR array with programmable output Logic. The two main differences between GAL and PAL devices are a) the GAL is reprogrammable and b) the GAL has programmable output configuration.

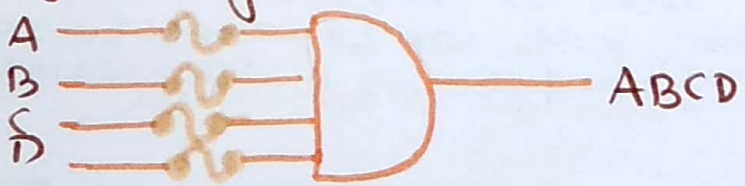
The GAL can be reprogrammed again and again because it uses E^2CMOS(electrically erasable CMOS) technology instead of bipolar technology and fusible links.



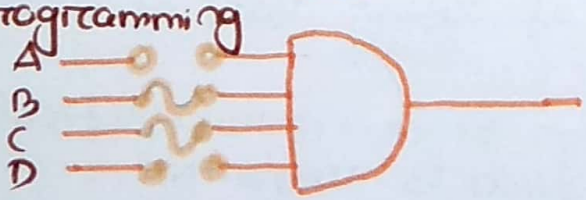
Symbology of PLD

In a programmable array the connections to each gate can be modified by symbol representation.

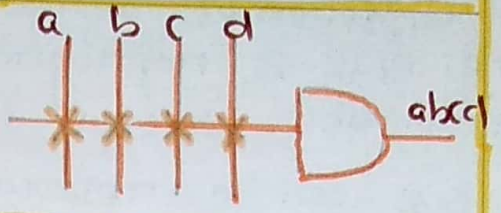
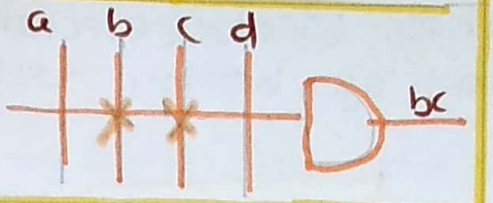
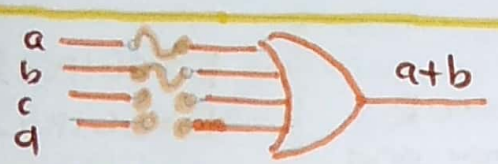
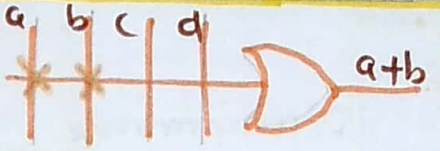
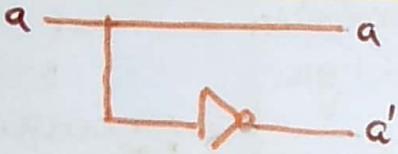
Before programming



After programming



In this figure, the AND gate realizes the product term abcd. Assume the product term BC to be generated. To do this, the gate is programmed by removing the fuse link of a and d.

Ordinary representation	Symbol representation
	
	
	
	

ROM as a PLD

A read-only memory is basically a combinational circuit and can be used to implement a logic function. Since programmable ROMs can be used for logic design, therefore it is also referred to as a programmable logic device (PLD).

The advantages of using ROM as a programmable logic device are

i) Ease of design since no simplification or minimization of logic function is required.

ii) Designs can be changed, modified rapidly.

iii) It is usually faster than discrete SSI/MSI circuit.

iv) Cost is reduced.

There are few disadvantages also of ROM-based circuits such as non-utilization of complete circuit, increased power requirement and increase in size with increase in number of input variables.

Programmable Logic Array (PLA)

A PLA consists of two-level AND-OR circuits on a single chip. The number of AND and OR gates and their inputs are fixed for a given PLA chip. The AND gates provide the product terms and the OR gates logically sum these product terms and thereby generate a SOP expression. The PLAs can be used to implement combinational and sequential logic circuits.

Program table

Before programming the PLA we first form the program table based on the specification. Program table has three parts; product term, inputs and outputs. The first column gives the number of product terms numerically, the second column specify the required paths between inputs and AND gates. The third column specifies the required paths between the AND gates and OR gates.

For example

$$F_1 = AB + A\bar{B}$$

$$F_2 = \bar{A}\bar{B} + AB$$

The above functions have only three minterms, the program table is formed as follows

Product term	Inputs		Outputs	
	A	B	F_1	F_2
AB (1)	1	1	1	1
$A\bar{B}$ (2)	1	0	1	-
$\bar{A}\bar{B}$ (3)	0	0	-	1

Q A combinational circuit is defined by the function

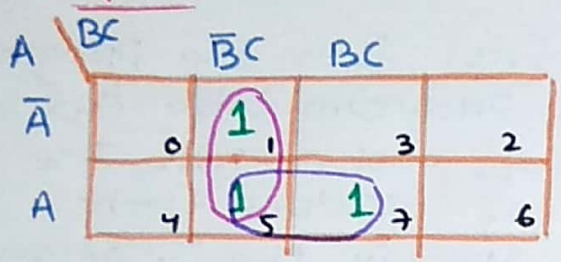
$F_1 = \sum m(1,5,7)$

$F_2 = \sum m(5,6,7)$

Implement the circuit with a PLA.

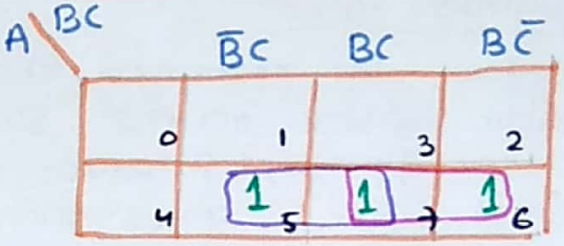
The simplified Boolean expression can be determined by using 3 variable K-map.

Expression for F_1



$F_1 = AC + \bar{B}C$

Expression for F_2

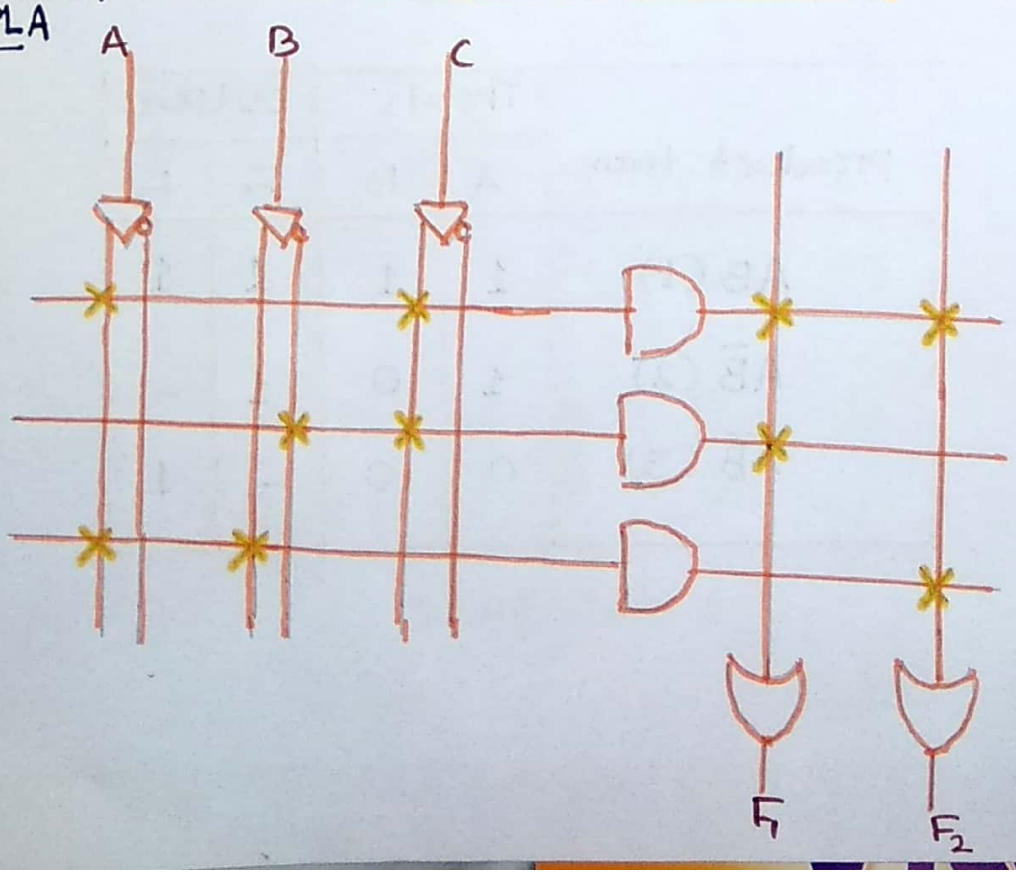


$F_2 = AC + AB$

Next form the program table to implement combinational Logic circuit into PLA. We have only three product terms.

Product term	Inputs			Outputs	
	A	B	C	F_1	F_2
AC (1)	1	-	1	1	1
$\bar{B}C$ (2)	-	0	1	1	-
AB (3)	1	1	-	-	1

Programmed PLA



A combinational logic circuit is defined by the function

$$F_1 = A\bar{B} + AC + \bar{A}B\bar{C}$$

$$F_2 = \overline{AC + BC}$$

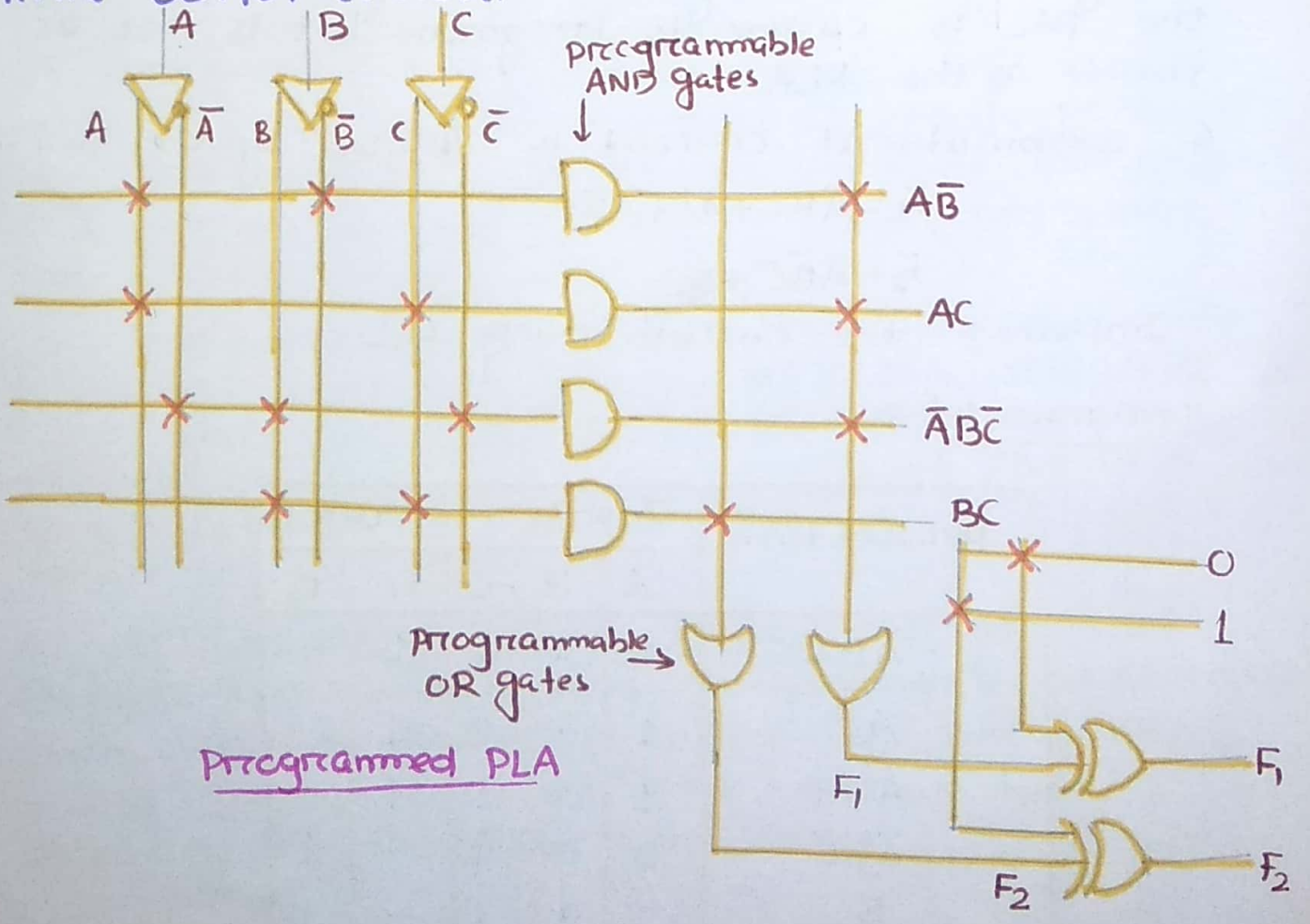
Implement the circuit with a PLA.

The circuit function is defined by boolean function, so we can construct the program table directly.

Program table

Product term	Inputs			Outputs	
	A	B	C	F_1	F_2
$A\bar{B}$	1	0	–	1	–
AC	1	–	1	1	1
$\bar{A}B\bar{C}$	0	1	0	1	–
BC	–	1	1	–	1

The function F_2 is inverted, hence we do the modification in the output circuit.



Programmed PLA

The outputs of the AND gates are connected to the inputs of each OR gate. The output of the OR gate goes to XOR gate where the other input can be programmed to receive a signal equal to either logic 1 or 0. The output is inverted when the XOR input is connected to 1. The output does not change when the XOR input is connected to 0

Programmable Array Logic (PAL)

Programmable array Logic is a particular family of programmable logic devices (PLDs) that is widely used and available from a number of manufactures. The PAL circuit consist of a set of AND gates whose inputs can be programmed and whose outputs are connected to an OR gate. PAL is a PLD with a fixed OR array and a programmable AND array. Because only the AND gates are programmable, the PAL is easier to program but is not as flexible as the PLA.

A combinational circuit is defined by the function

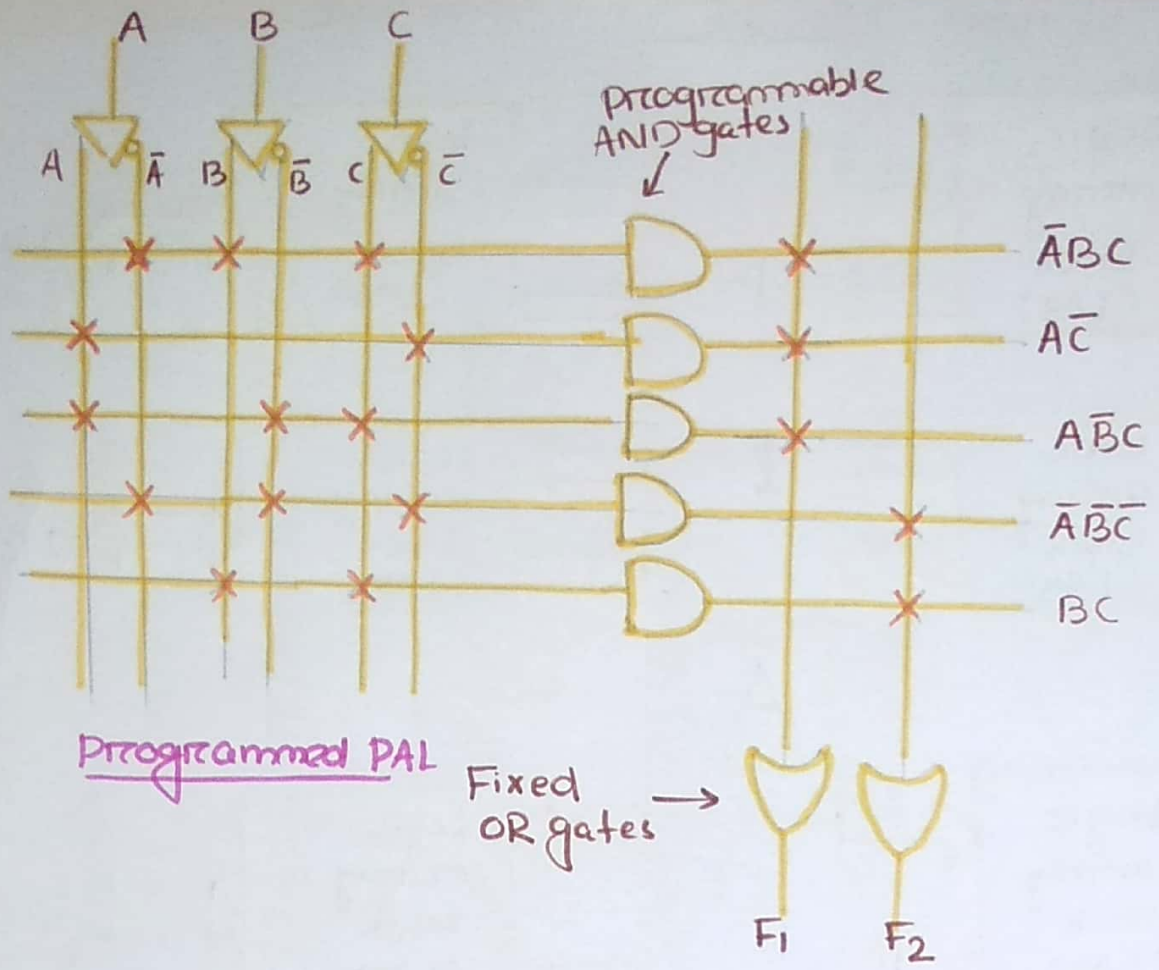
$$F_1 = \bar{A}BC + A\bar{C} + A\bar{B}C$$

$$F_2 = \bar{A}\bar{B}\bar{C} + BC$$

Implement the circuit using PAL.

Program table

Product term	Inputs			Outputs	
	A	B	C	F_1	F_2
$\bar{A}BC$	0	1	1	1	-
$A\bar{C}$	1	-	0	1	-
$A\bar{B}C$	1	0	1	1	-
$\bar{A}\bar{B}\bar{C}$	0	0	0	-	1
BC	-	1	1	-	1

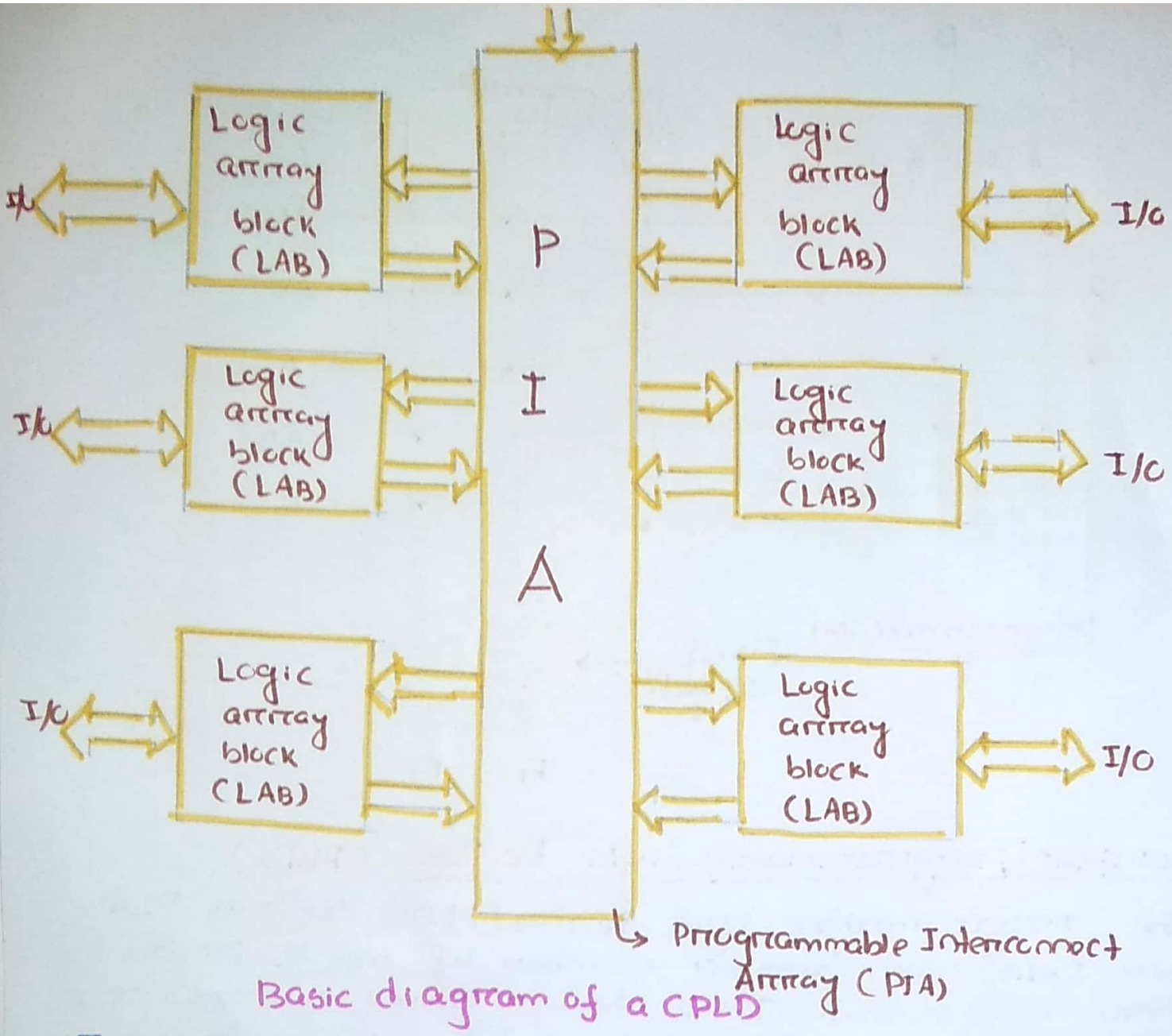


Complex Programmable Logic Devices (CPLDs)

The programmable logic devices (PLDs) such as PLAs and PALs have limited number of inputs, product terms and outputs. These devices, therefore, can support up to about 32 total number of inputs and outputs only.

For implementation of circuits that require more inputs and outputs than that are available in a single SPLD chip, either multiple SPLD chips can be employed or more sophisticated type of chip, referred to as complex programmable logic device (CPLD) can be used.

A CPLD basically consists of multiple groups of PAL/GAL-like arrays with programmable interconnections. Each PAL/GAL group is called Logic array block (LAB), function block or some similar term depending on the particular device.



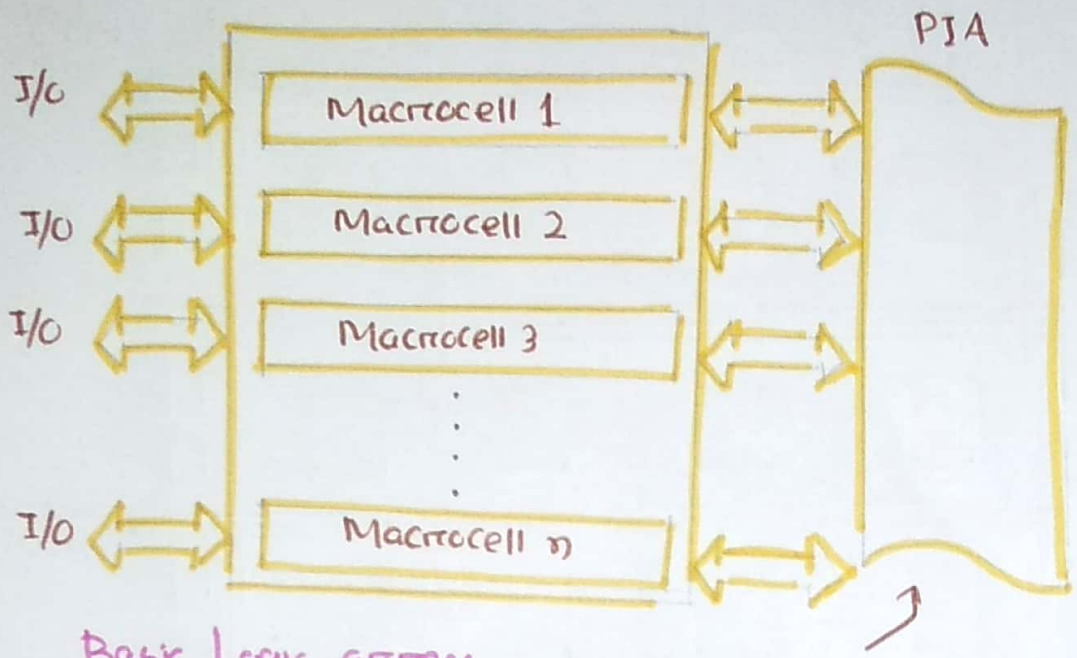
Basic diagram of a CPLD

Each logic array block contains several PAL/GAL-like arrays called macrocells. Each LAB can be interconnected with other LABs or to other I/Os (input/output) using the programmable interconnect array (PIA) to form large complex logic functions. Like the PAL or GAL the CPLD is based on a sum-of-products (SOP) architecture.

Macrocells

Each Logic array block in a CPLD contains several macrocells. CPLD architecture varies from one manufacturer to another, but generally there are from 32 to several hundred macrocells in one LAB. For circuits requiring a very large number of gates, CPLDs having large number of macrocells can implement circuits of up to about 10,000 equivalent gates. A typical macrocell has an AND array, a product term matrix,

an OR gate, and a programmable register section.

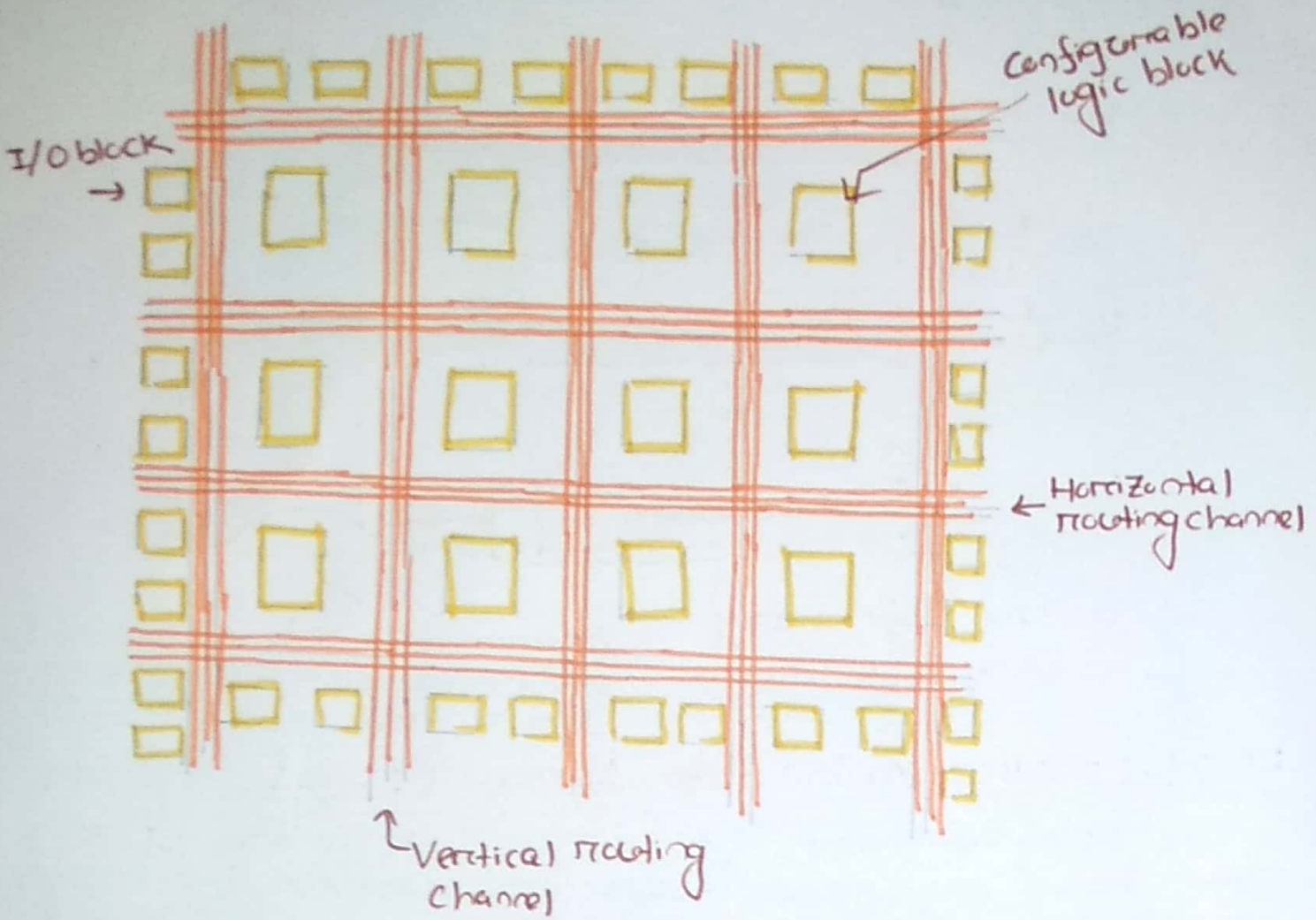


Basic Logic array block in a CPLD

Programmable interconnections to other Logic array blocks and I/Os in the CPLD

Field-Programmable Gate Array (FPGA)

The programmable logic devices (SPLDs and CPLDs) are based on similar basic architecture. the programmable array Logic. Over the years, programmable arrays have increased in size and complexity and higher configurable output macrocells have been added to enhance their flexibility and expandability. To increase the effective size and to add more functionality in a single programmable device, alternate architectures have been developed which are known as field-programmable gate arrays (FPGAs). The logic densities of FPGAs are much higher than those of CPLDs. They range in size from 10,000 to a few hundreds of thousands equivalent gates. FPGA devices support implementation of relatively Large Logic circuits.



FPGA Architecture

The FPGA basically consists of an array of logic blocks with programmable interconning channels ie horizontal & Vertical routing channel surrounded by programmable I/O blocks. Each logic block in FPGA contains several Logic elements The programmable logic blocks of FPGAs are called configurable logic blocks (CLBs). It consists of a large number of programmable logic blocks surrounded by programmable I/O block. The programmable logic blocks of FPGA are smaller and less capable than a PLD, but an FPGA chip contains a lot more logic blocks to make it more capable. The logic blocks are distributed across the entire chip. These logic blocks can be interconnected with programmable interconnections.