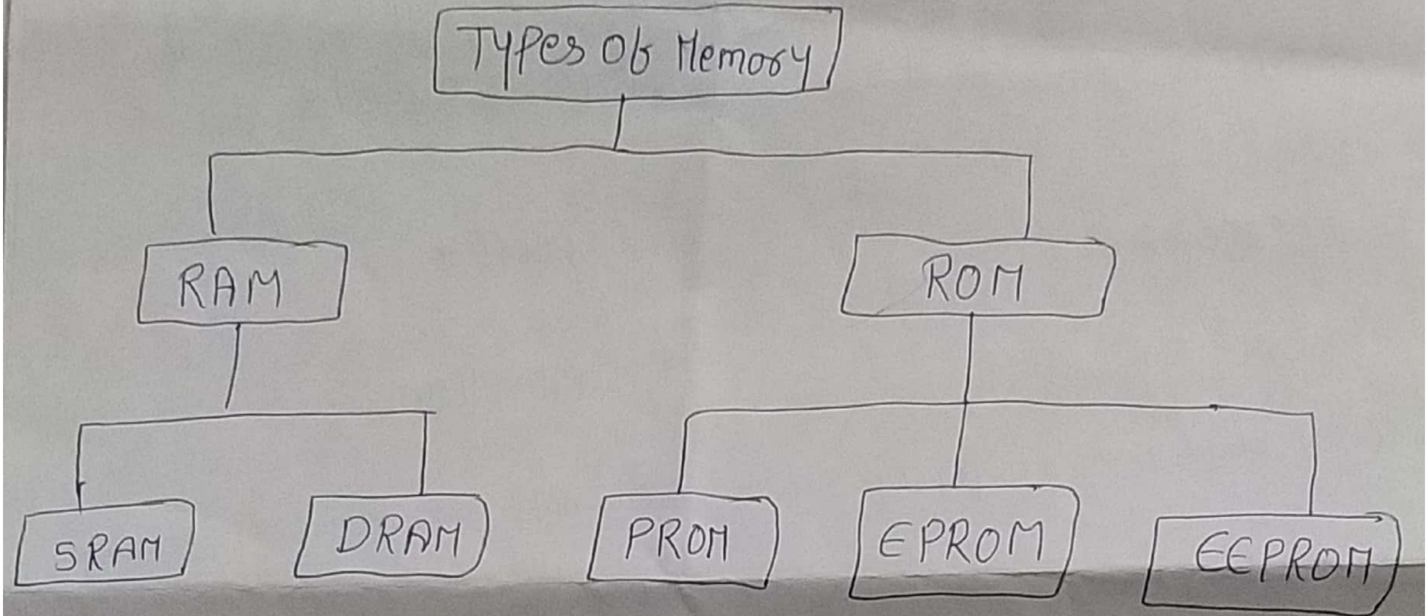
Memory :- Computer memory is the storage space in the computer where data is to be processed and instructions required for processing and storage. The memory is divided into large number of small parts called cells.



RAM: Random Access Memory

→ It is also called as read write memory (or) the main memory or the primary memory

→ The programs and data that the CPU requires during execution of a program are stored in this memory

→ It is a volatile memory as the data loses when the power is turned off.

→ RAM is further classified in to two types

→ SRAM - Static Random Access Memory

→ DRAM - Dynamic Random Access Memory

DRAM	SRAM
1) Constructed of tiny capacitors that leak electricity	1. Constructed of circuits similar to D-Flip Flop.
2 Requires a recharge every few milliseconds to maintain its data	2 Holds its contents as long as Power is available
3 In expensive	3 Expencive
4 slower than SRAM	4 Faster than
5- Can store many bits per chip	5 Can not store many bits per chip
6 Uses less Power	6 uses more Power
7 Generates less heat	7 Generates more heat
8 used for main memory	8 used for cache.

Read only Memory:-

Stores crucial information essential to operate the system, like the Program essential to boot the computer.

→ It is not volatile.

→ Always retains its data

→ used in embedded systems or where the programming needs no change.

→ used in calculators an peripheral devices.

→ RoM is Further Classibied into 4 types: ROM, PROM, EPROM, and EE PROM

Types ob Read only Memory :-

1. PROM (Programmable read-only memory) :-

It Can be Programmed by user. once Programmed, The data and instructions in it Cannot be Changed

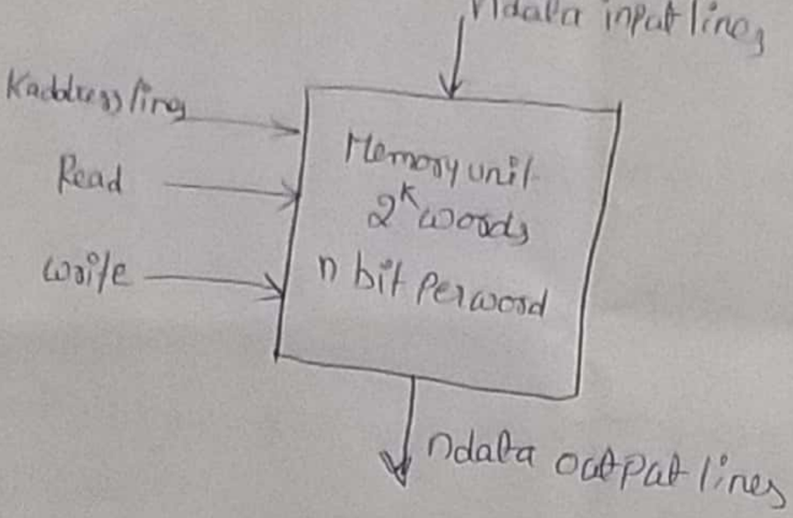
2. EPROM (Erasable Programmable read only memory) :-

It Can be reprogrammed To erase data from it, expose it to ultra violet light To reprogram it, erase all The previous data.

3. EEPROM (Electoically erasable programmable read only Memory) :-

The data Can be erased by applying electric bield, no need ob ultra violet light. we Can erase only portions of The Chip

RAM	ROM
1. Temporary Storage	1. Permanent storage
2. Store data in MB's	2. Store data in GB's
3 volatile	3. Non-volatile
4. Used in normal operations	4. Used bor Startup process of Computer
5. writing data is faster	5. writing data is slower.

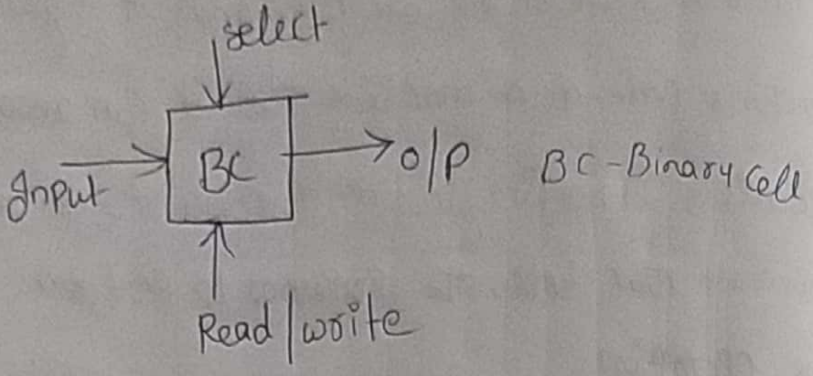


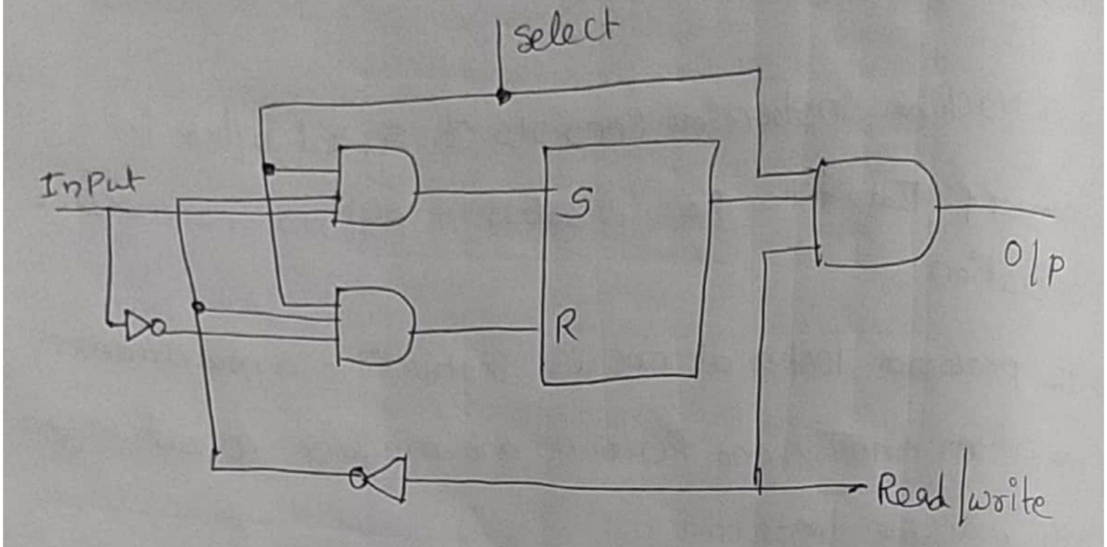
Block diagram of Memory unit

Unit-V

Memory Decoding

In a memory unit There is a need for decoding circuits to select the memory word specified by the Input ~~Registers~~ addresses





Logic diagram

The Internal constoruction of a RAM of m words and n bits Per word Consists of mxn binary storage cells and associated decoding Ciraits for selecting Individual words the Binary storage cell is The basic building block of a memory unit. The equivalent logic of a binary cell That stores one bit of information

The Cell is an electronic circuit with four to6 transistor It is possible and Convenient to model it in terms of logic symbols. A binary storage Cell must be very small in order to be able to pack as many Cells as possible in The small area Available in The integrated circuit chip The binary Cell stores one bit in its internal latch The select input enables The Cell for reading or writing

The Logical Construction of small RAM :

The RAM Consist of four words of four bits each and has a total of 16 binary Cells. The small blocks Labeled BC represents binary Cell with its. Three inputs and one output. as specified Labeled BC Represents The binary Cell with its Three inputs and one o/p. A memory with four words need two address lines. The two address inputs go Through a 2×4 decoder to select one of The four words.

The decoder is enabled with The memory enable input when The memory enable is 0, all o/p of The decoder are 0 and none of The memory words selected

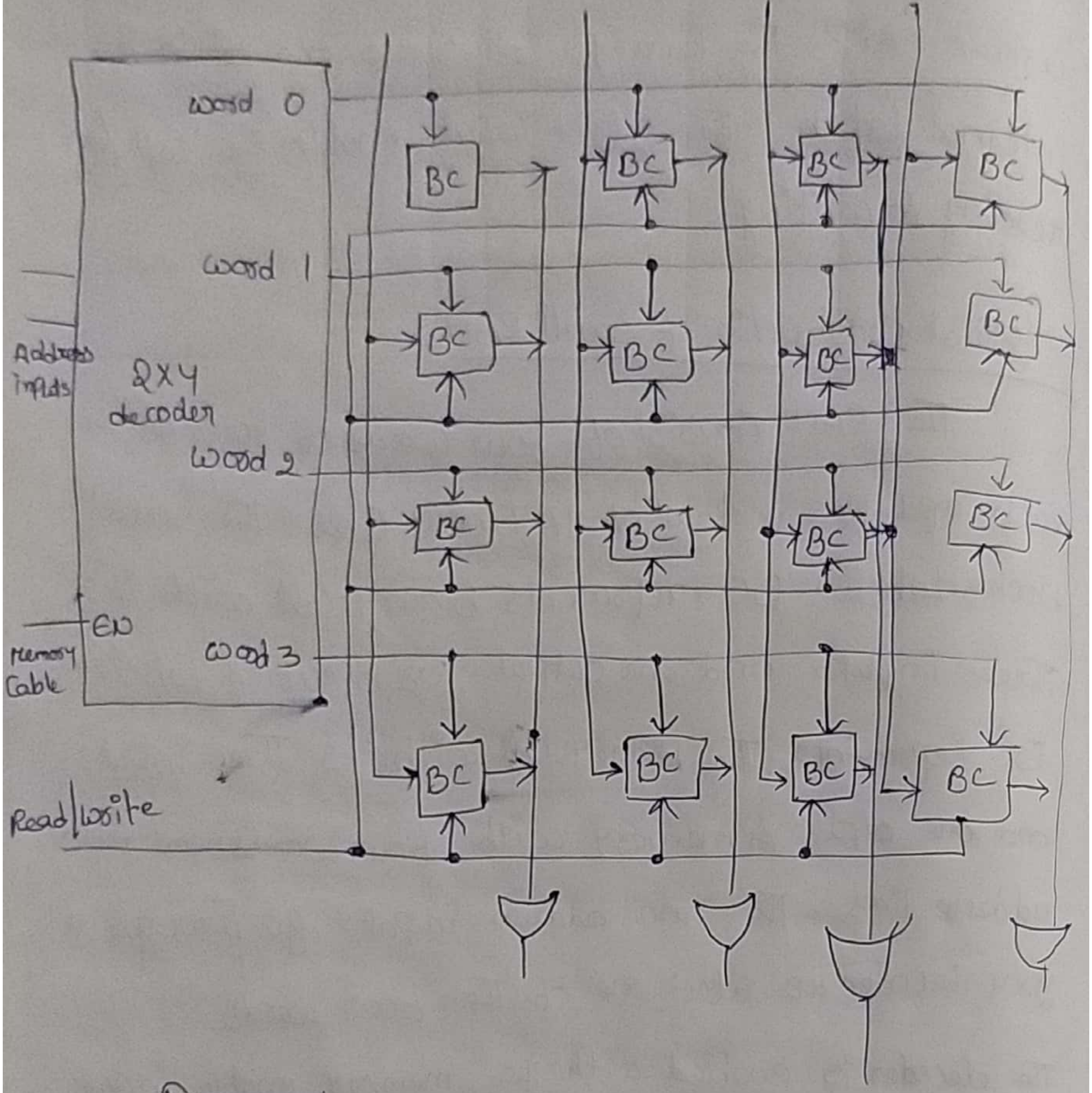
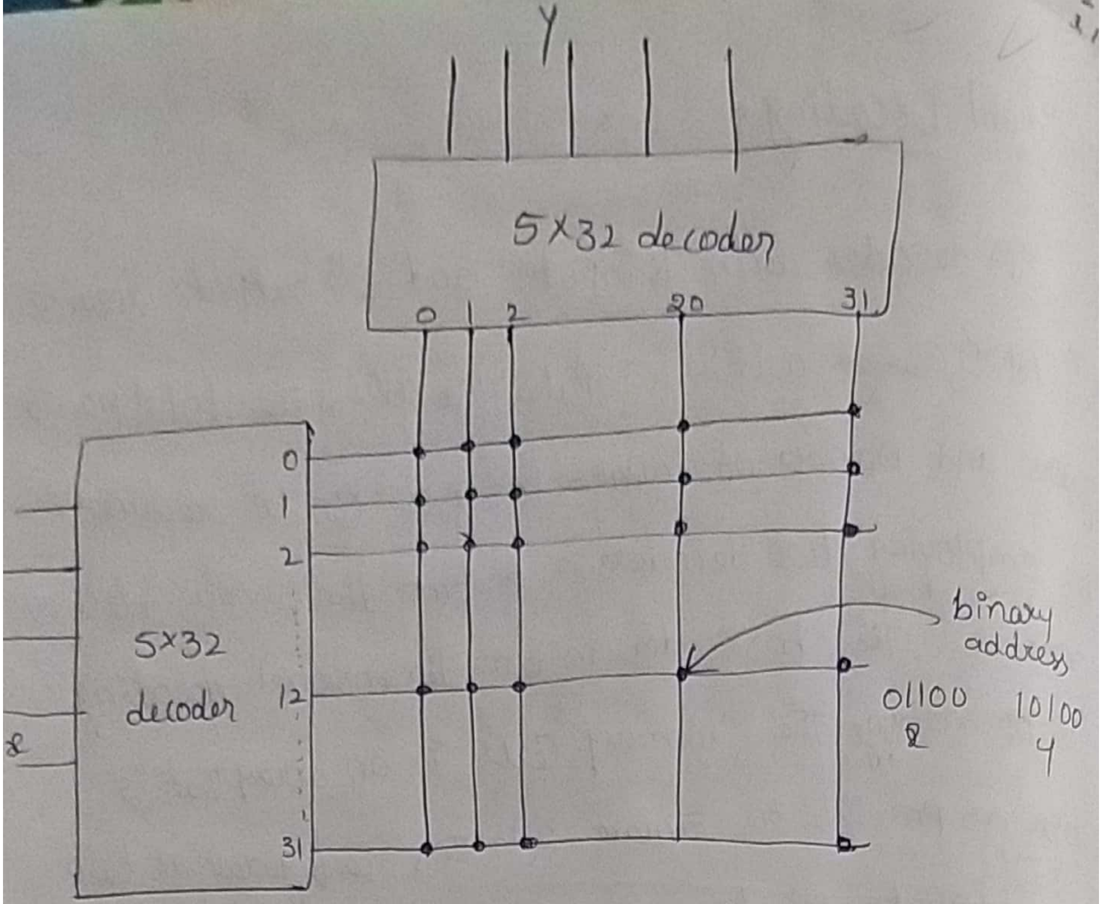


Diagram of 4X4 RAM

Coincident Decoding:

A decoder with k inputs and 2^k outputs require 2^k AND Gates with k inputs per gate The total no of gates and number of inputs per gate Can be reduced by employing two decoders in a two dimensional selection scheme. The basic idea in two-dimensional decoding is to arrange The memory Cells in an array That is close as Possible to square. In this Configuration two $K/2$ - inputs decoders are used insted of one k-input decoder. one decoder performs The row selection and The other The Column selection in a twodimensional matrix Configuration.

The two dimensional Selection Pattern is demonstrated for a 1k-word memory. Insted of using a single 10X1024 decoder, we use two 5x32 decoders. with The single decoder, we would need 1024 AND gates with 10 inputs in each. In The two decoder Case we need 64 AND gates with 5 inputs in each.



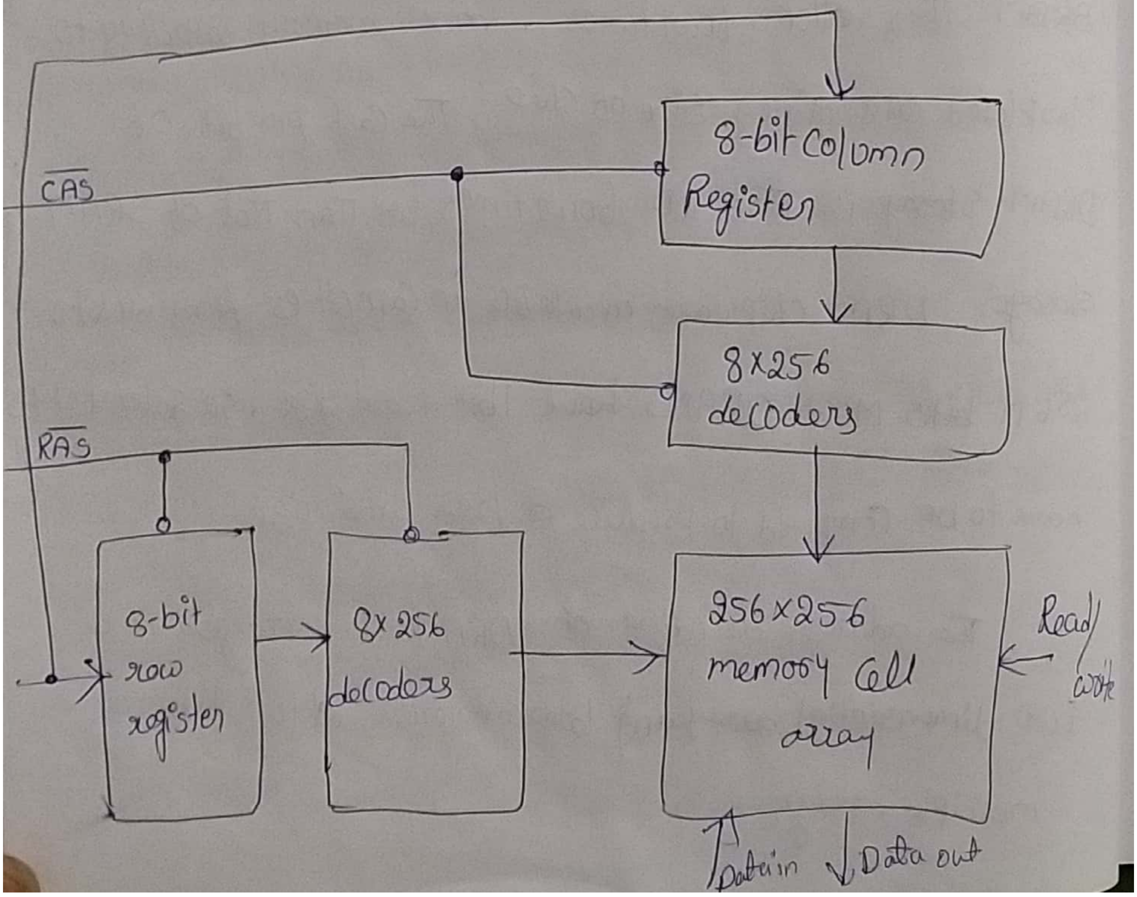
Two -Dimensional decoding structure for a 1k-word Memory

Consider The word whose address is 404. The 10-bit binary equivalent of 404 is 0100 10100 This makes $X = \underset{8+4}{01100}$ (binary 12) and $Y = \underset{16+4}{10100}$ (binary 20) The n-bit word That is Selected lies in The x-decoder output number 12 and The y decoder out put number 20. All The bits of The word are selected for reading or writing.

Address multiplexing:

Address multiplexing Permits you to use one tag (multiplex tag) to call multiple memory locations in the contoollers address area you can have read and write access to the multiple memory locations without having to define a tag for each individual address. This is a very efficient method of processing large volume data. The SRAM memory cell modeled typically contains six transistors In order to build memories with higher density it is necessary to reduce the number of transistors in a cell. The DRAM cell contains a single MOS transistor and a capacitor. DRAMS typically have four times the density of SRAM. This allows four times as much memory capacityy to be placed on a given size of chip. The cost per bit of DRAM storage is three to four times less than that of SRAM storage. DRAM chips are available in capacities from 64 K to 256 M bits Most DRAMS have 1 bit word size, so several chips have to be combined to produce a large word size.

The address decoding of DRAMS is arranged in a two dimensional array and large memories often have a multiple arrays we will use a 64k-word memory to illustoate The address multiplexing idea. The memory Consists of a two dimensional array of Cells arranged into 256 rows by 256 Columns. for a total of $2^8 \times 2^8 = 2^{16} = 64k$ words. There is a single data input line, a single data output line and a read/write control, as well as an eight bit address input and two address strobes, The latter included for enabling The row and Column address into Their respective registers. The ROW address Stoobe (R, enables The 8 bit row Register and The Column address stoobe (CAS) enables 8-bit Column Register. The bar on name of The stoobe symbol indicates That The register are enable on zero level of The signal.



The 16-bit address is applied to the DRAM in two steps using RAS & CAS. the both latches are in the 1 state The 8bit row address is applied to the address inputs and RAS is changed to 0 This loads the row address into the row address register. RAS also enable the Row decoder so that it can decode the row address and select one row of the array. The 8-bit column address is then applied to the address inputs and CAS is driven to the 0 state This transfer the column address into the column register and enable the column decoder

Error Detection and Error Correction

A memory unit may cause occasional errors in storing and retaining the binary information. The reliability of a memory unit may be improved by employing error detecting & correcting codes. The most common error detecting scheme is parity bit. A parity bit is generated and stored along with the data word in memory.

The data word is accepted if the parity of the bits read out is correct. If the parity checked result in an inversion an error is detected, but it cannot be corrected.

Hamming Code:

One of The most Common error Correcting Codes used in RAMs was devised by R.W Hamming. In a hamming code K parity bits are added to an n-bit data word, forming a new word of n+k bits. The bit positions are numbered in sequence from 1 to n+K. Those positions numbered as a power of 2 are reserved for The Parity bits. The remaining bits are the data bits. The Code Can be used with words of any length.

for example The 8-bit data word 11000100 we include 4 parity bits with The 8 bit word and arrange The 12 bits as follows

Bit Positions:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

The 4 parity bits P1, P2, P4 and P8 are in positions 1,2,4 respectively. The 8 bits of The data word are in The remai Positions. Each parity bit Calculated like

P_1 = XOR of bits (3,5,7,9,11)

P_2 = XOR of bits (3,6,7,10,11)

P_4 = XOR of bits (5,6,7,12)

P_8 = XOR of bits (9,10,11,12

$P_1 = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$

$P_2 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$

$P_4 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$

$P_8 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$

The 8bit data word is stored in memory together with the 4 Parity bits as a 12-bit Composite word substaining the 4P bits in Their Proper positions, we obtain the 12 bit Composite word stored in memory

0 0 1 1 1 0 0 1 0 1 0 0
1 2 3 4 5 6 7 8 9 10 11 12 → Bit Position

When the 12 bits are read from memory, They are checked again for errors. The Parity is checked over the same combination of bits, including parity bit, The 4 bits are evaluated.

C_1 = XOR of bits(1, 3, 5, 7, 9, 11) = $0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus \underline{0}$

C_2 = XOR of bits (2, 3, 6, 7, 10, 11) = $0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$

C_4 = XOR of bits (4, 5, 6, 7, 12) = $1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$

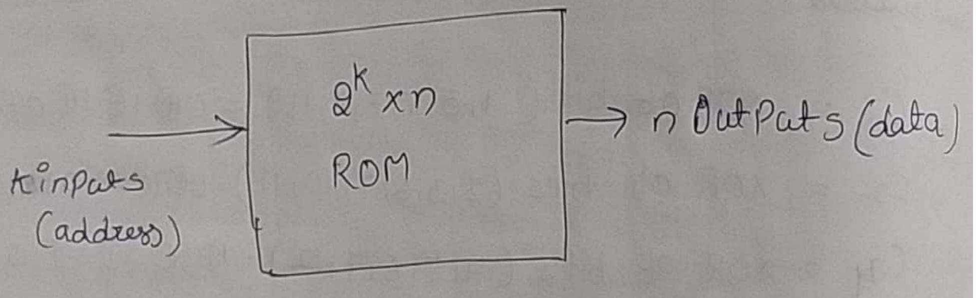
C_8 = XOR of bits (8, 9, 10, 11, 12) = $1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$

A 0 check bit designates even parity over the checked bits and a 1 designated odd parity. since the bits were stored with even parity the result $C = C_8 C_4 C_2 C_1 = 0000$ indicates that no error occured.

Read only Memory :

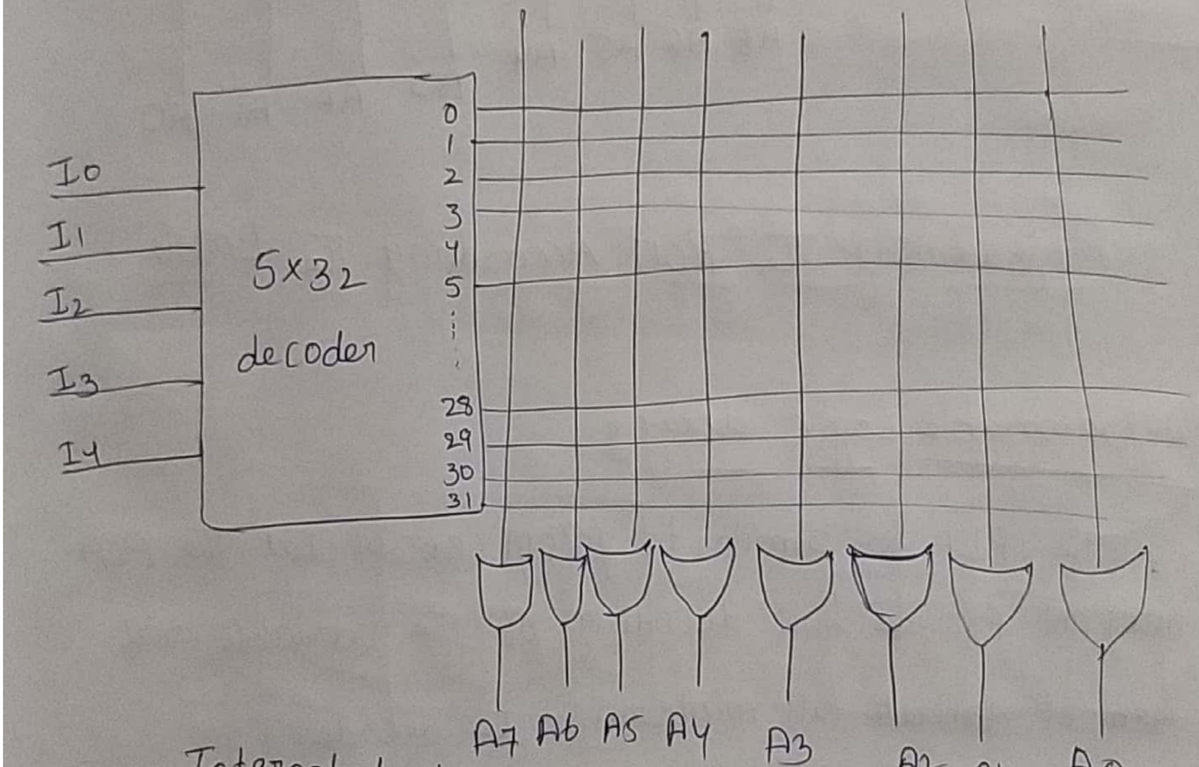
A ROM is essentially a memory device in whic Permanent Binary information is stored. The binary inform must be specified by the designer and is then embedded in the unit to form the required interconnection Pattern

A block diagram of a ROM Consisting of k inpu and n outputs. The inputs Provide the address for memo and o/p gives the databits 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. ROM does not have data inputs, because it does not have a write operation.



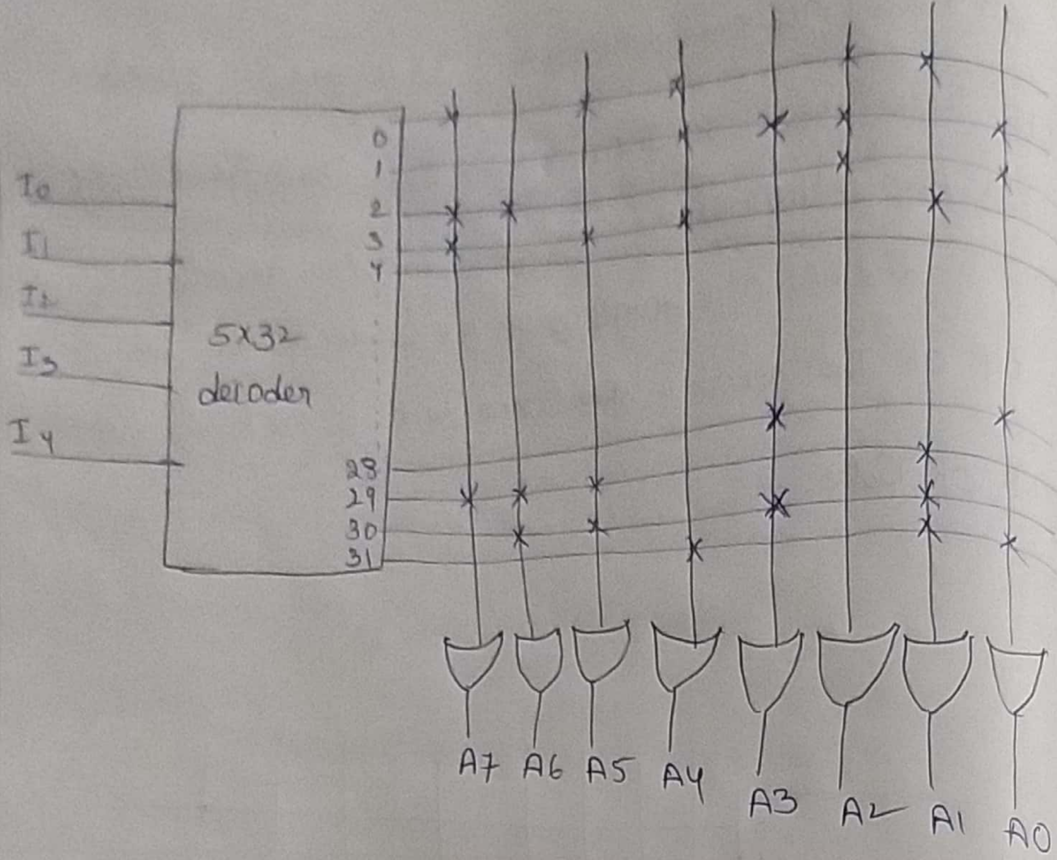
Integrated circuit ROM Chips have one or more enable inputs and sometimes Come with Three state o/P to facilitate the Construction of Large arrays of ROM.

A 32×8 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. The five inputs decoded into 32 distinct o/p by means of 5×22 decoder. The 32 o/p of The decoder connected to each of The eight OR Gates.



Internal Logic of ~~32~~ X32 ROM

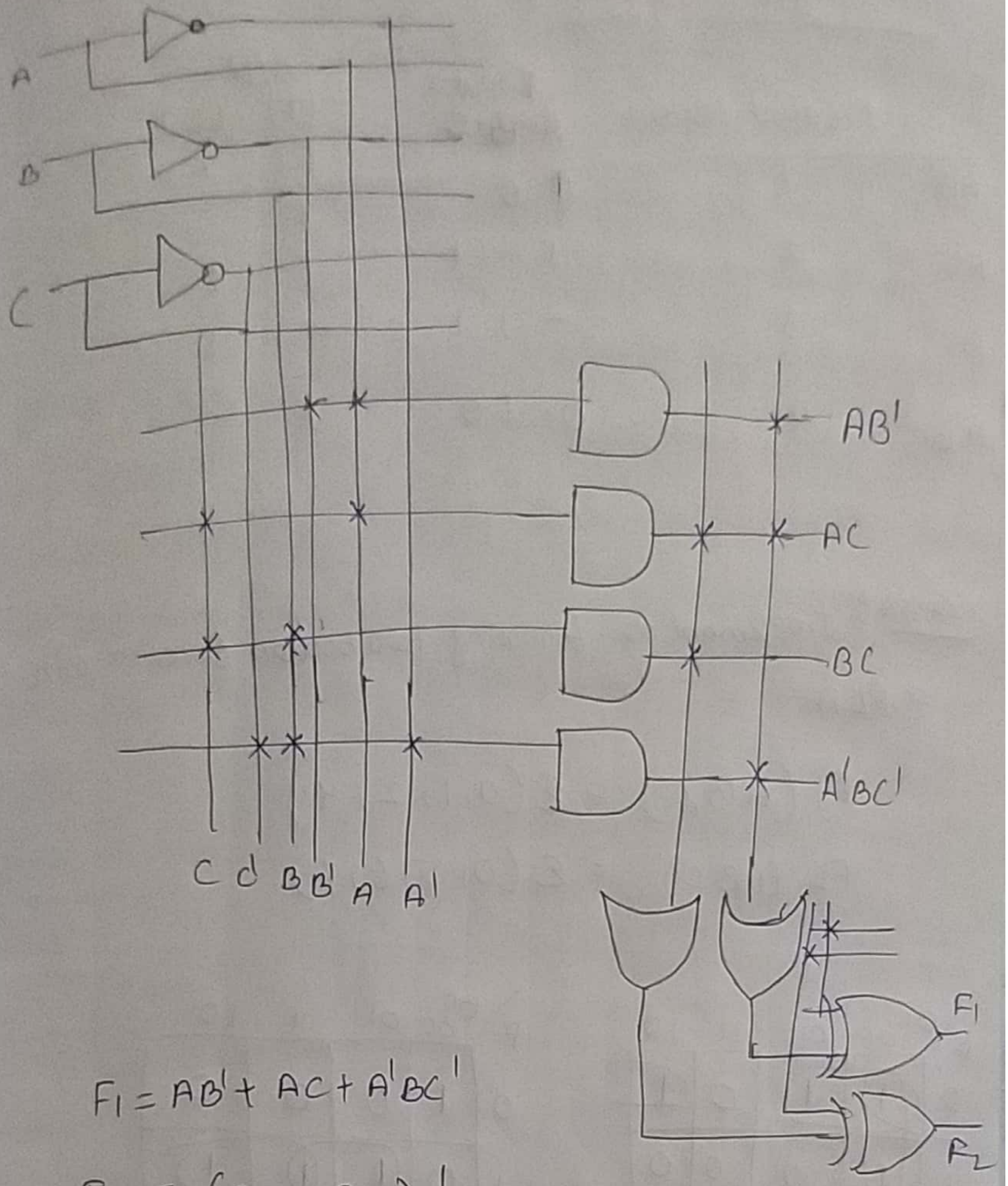
Inputs					Outputs							
I4	I3	I2	I1	I0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	6	1	1	0	0	1	0
		⋮						⋮				
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	ɸ	1



Programming The ROM According The table

Programmable Logic array:

The PLA is similor to PROM, except That The PL does not provide full decoding of The variable and does not generate all minterms. The decoder is Replaced by an array of AND gates That Can be programm to generat any Product term of The input Variables The Product terms are Then Connected to OR Gates to provide The Sum of products for The required boolean function.



$F_1 = AB' + AC + A'BC'$

$F_2 = (AC + BC)'$

PLA with 3inputs, four product terms & 2 outputs

PLA Programming table :

	Product term	Inputs			O/P	
					(T)	(C)
		A	B	C	F1	F2
AB'	1	1	0	–	1	–
AC	2	1	–	1	1	1
BC	3	–	1	1	–	1
A'BC'	4	0	1	0	1	–

Example :- Implement the following two boolean function with a PLA

$$F_1(A,B,C) = \Sigma(0,1,2,4)$$

$$F_2(A,B,C) = \Sigma(0,5,6,7)$$

F1

A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	0	0

A \ BC	00	01	11	10
0	1	0	0	0
1	0	1	1	1

$A'B' + A'C' + B'C'$ $\qquad$ $A'B'C' + AC + AB$

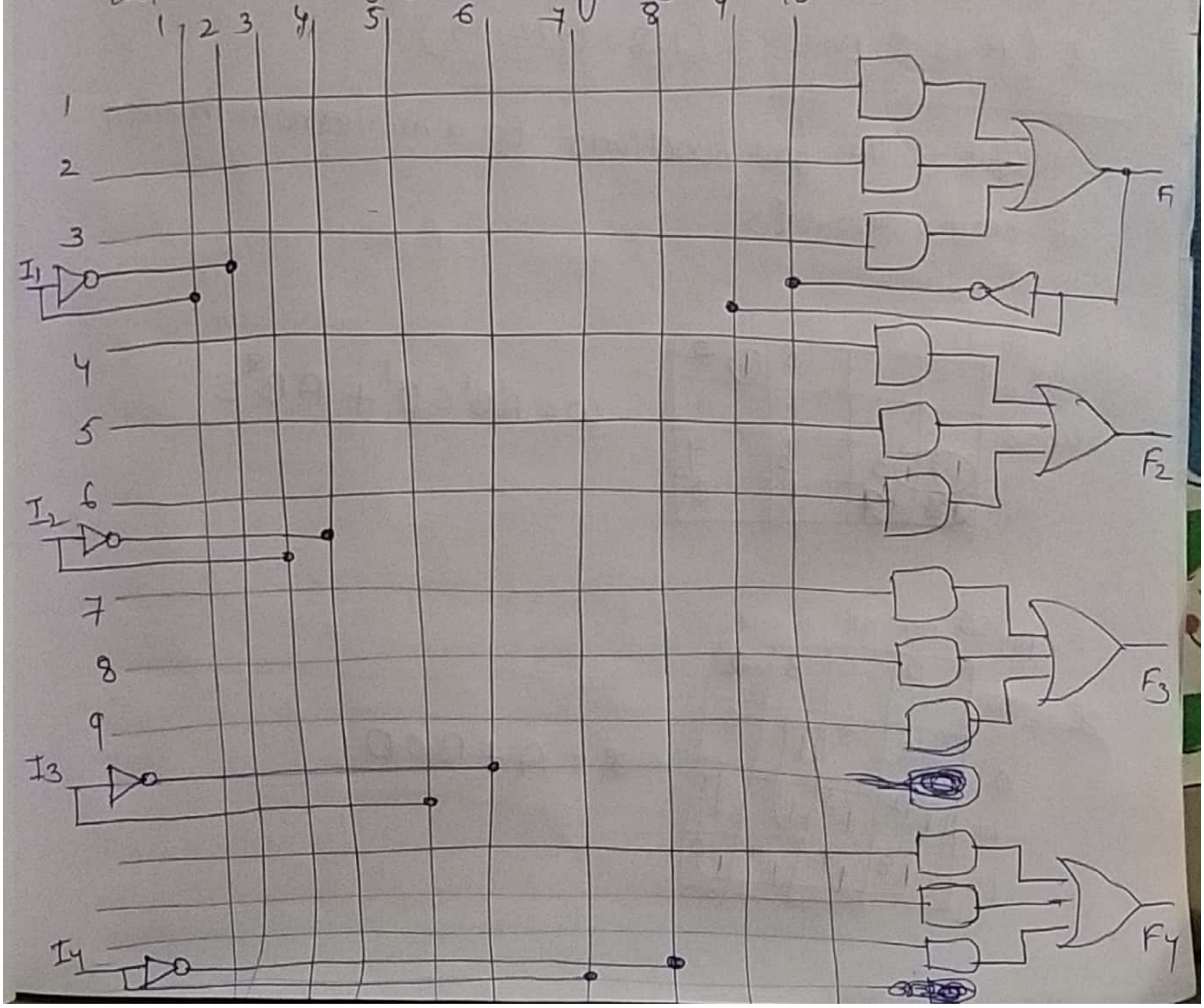
$F_1 = (AB + AC + BC)'$ $\qquad$ $F_2 = AB + AC + A'B'C'$

PLA Programming table

	Product term	Inputs			O/P	
		A	B	C	F1	F2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
A'B'C'	4	0	0	0	–	1

Programmable Logic array Logic :

The PAL is a Programmable array Logic device with a Fixed OR array and a Programmable AND array. Because only The AND Gates are programmable, The PAL is easier to program Than but is not as flexible as The PLA The Logic Configuration of a typical PAL with four inputs and four outputs Each input has a buffer inverted gate, and each output is generated by a fixed OR gate. There ar four sections in The unit, each Composed of an AND-OR array that is three wide, The term used to indicate that there are Three programmable AND gates in each section and one fixed OR gates.



In designing with a PAL, The Boolean function must be sim
to fit into each section each function Can be simplif
by itself, with out regard to common product term
The number of product terms in each section is fixed
and if The number of terms in The function is too
large, it may be neccessory to use two sections to
implement Boolean functions.

an example of using a PAL in The design of a
Combinational circuit

$w(A,B,C,D) = \Sigma(2,12,13)$
$x(A,B,C,D) = \Sigma(7,8,9,10,11,12,13,14,15)$
$y(A,B,C,D) = \Sigma(0,2,3,4,5,6,7,8,10,11,15)$
$z(A,B,C,D) = \Sigma(1,2,8,12,13)$

simplifying The four functions to a minimum number
of terms results

w =

AB \ CD	00	01	11	10
00	0	1	3	1 2
01	4	5	7	6
11	1 12	1 13	15	14
10	8	9	11	10

$w = A'B'CD' + AB'C'$

x =

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	1 7	6
11	1 12	1 13	1 15	1 14
10	1 8	1 9	1 11	1 10

$x = A + BCD$

y =

AB \ CD	00	01	11	10
00	0 1	1	3 1	2 1
01	4 1	5 1	7 1	6 1
11	12	13	15 1	14
10	8 1	9	11 1	10 1

$y = A'B + CD + B'D'$

Z =

AB \ CD	00	01	11	10
00	0	1 1	~~3~~	2 1
01	4	5	7	6
11	12 1	13 1	15	14
10	8 1	9	11	10

$$Z = A'C'D' + ABC' + A'B'C'D + A'B'CD'$$

$$Z = \underbrace{A'B'CD' + ABC'}_{\omega} + AC'D' + A'B'C'D$$

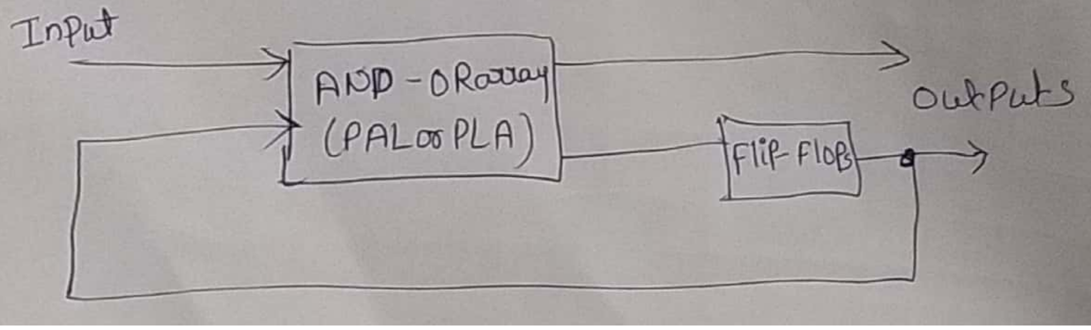
$$Z = \omega + AC'D' + A'B'C'D$$

2^4

Product term	AND inputs					Outputs
	A	B	C	D	w	
1	1	1	0	—	—	$w = ABC' + A'B'CD'$
2	0	0	1	0	—	
3	—	—	—	—	—	
4	1	—	—	—	—	$x = A + BCD$
5	—	1	1	1	—	
6	—	—	—	—	—	
7	0	1	—	—	—	$y = A'B + CD + B'D$
8	—	—	1	1	—	
9	—	0	—	0	—	
10	—	—	—	—	1	$z = w + AC'D' + A'B'C'D$
11	1	—	0	0	—	
12	0	0	0	1	—	

Sequential Programmable Devices:

Digital systems are designed with flip-flop and gates since The Combinational PLD consists of only gates, it is neccessary to include external flip-flops when they are used in the design. sequential programable devices include both gates and flip flops.



A Programable logic device (PLD) is an electronic Component used to build reConfigurable digital circuits unlike integrated circuits (IC) which consist of logic gates and have a fixed function, a PLD has an undefined function at The time of manufacture

Two major Categories of user-programable logic logic are simple Programmable logic device) and CPLD (complex PLD), an FPGA (field Programmable gate array).