UNIT IV

MEMORY AND PROGRAMMABLE DEVICES

4.1Introduction

Memories are made up of registers. Each register in the memory is one storage location called memory locations. Each memory location is identified by an address. The number of storage locations can vary from a few in some memories to hundreds of thousands in others. Each can accommodate one or more bits. Generally, the total number of bits that a memory can store is its capacity. Most of the types the capacity is specified in terms of bytes.

Each register can consist of storage elements, each of which stores one bit of data. A storage element is called a cell.

The data stored in a memory by a process called writing and are retrieved from the memory by a process called reading. Figure: 4.1.1illustrates in a very simplified way the concept of write, read, and address and storage capacity for a generalized memory.

The time it takes to transfer information to or from any desired random location is always the same-hence the name random-access memory, abbreviated RAM . In contrast the time required to retrieve information that is stored n magnetic tape depends on the location of the data.

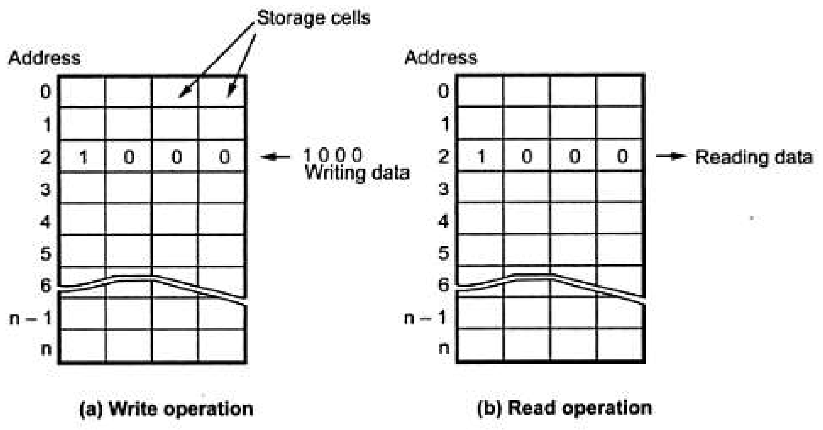


Figure: 4.1

As shown in Figure: 4.1a memory unit stores binary information in groups of bits known as words. A word in memory is an entity f bits that moves in and out of storage as a unit. A word having group of 8-bit is known as a byte. Most computer memories use words that are multiples of 8-bits in length. Thus, a 16-bit word contains 2-bytes, and a 32-bit word is made f 4-bytes.

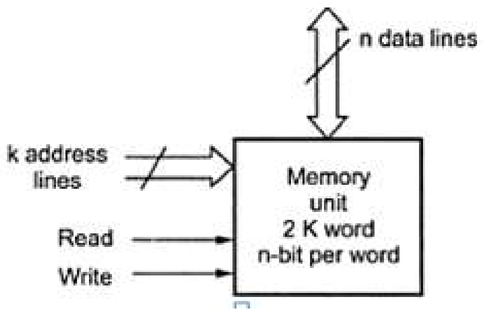
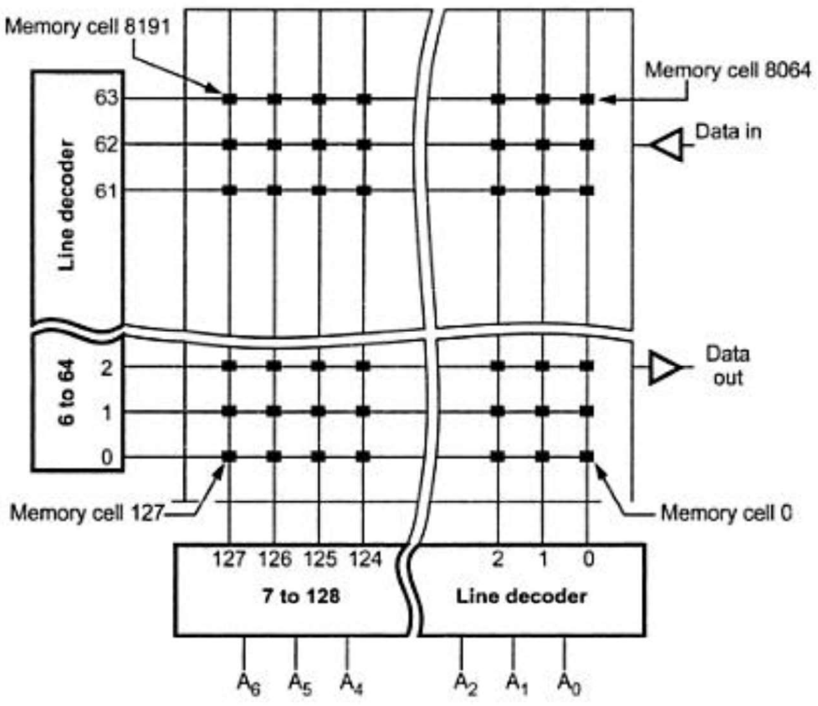


Figure: 4.2 Block diagram of memory unit



The communication between a memory and its environment is achieved through data lines, address selection and control lines that specify the direction of transfer . The fig shows the bock diagram of memory unit. Then data lines provide the information to be stored in memory and the ‘K’ address lines specify the particular word chosen among the many available. The 2 control inputs specify the direction transfer. The memory enable is used t enable the memory chip. T perform read/write pertain; memory enable signal should be active. When there are K address lines we can access 2^k memory words. For example, if K=10 we can access 2^10=1024 memory words.

4.2 ROM (READ ONLY MEMORY)

We can't write data in read only memories : It is non-volatile memory . i:e ,it can hold data even if power is turned off . Generally , ROM is used to store the binary codes for the sequence of instructions you want the computer to carry out and data such as look up tables. This is because this type of information does not change .It is important to note that although we give the name to static and dynamic read/write memory devices, that does not mean the ROMs that we are using are also not random access devices. In fact, most ROMs are accessed randomly with unique addresses. There are 4 types of ROM :Masked ROM,PROM,EPROM,EEPROM.

4.2.1PROM (Programmable ROM)

PROM is programmable by user. To provides the programming facility, each address select and dateline intersection has its own fused MOSFET or transistor. When the fuse is intact, the memory cell is configured as a logic 1 and when fuse is blown ,the memory cell is the vertical data line with a pulse of high current . The figure shows a PROM fused MOSFET memory cell.

The figure shows 4 byte PROM. It has diodes in every bit position; therefore, the output is initially a 0`s. Each diode however has a fusible ink in series with it. By addressing bit and applying proper current pulse at the corresponding output we can blow out of fuse, storing logic 1 at that bit position. The fuse material is like nichrome and polycrystalline. For blowing the fuse it is necessary to pass around 20 to 50 mA of current for period 5 to 20 s. The blowing of fuses according to the truth table is called programming of ROM. The user can program PROMs with special PROM programmer. The PROM programmer selectively burns the fuses according to the bit pattern to be stored. This process is also known as burning of PROM. The PROMs are one time programmable. Once programmed, the information stored is permanent.

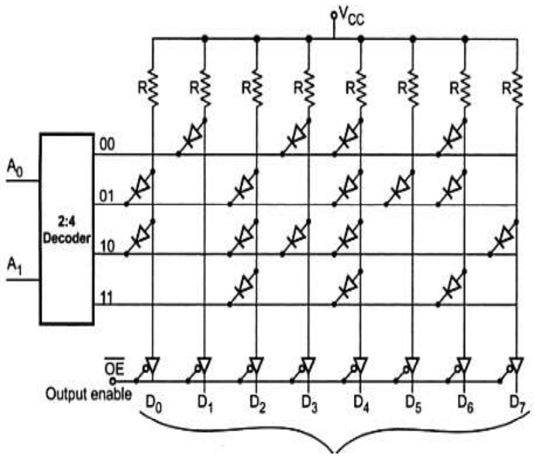


Figure 4.3: four byte PROM

4.2.2EPROM (Erasable Programmable ROM)

EPROMs use MOS circuitry .They store 1`s and 0`s as a packet of charge in a buried layer of the IC chip. EPROMs can be programmed by the user with special EPROM programmers. the important point is that we can erase the stored data in the EPROMs by exposing the chip to ultra violet light through its quartz window for 15 to 20 minutes
It is not possible to erase selective information , when erased the entire information is most. The chip can be reprogrammed. This memory is ideally suitable for product development experimental projects and college laboratories, since this chip can be reused many times , over.

EPROM Programming

When erased each once in the EPROM contains 1. Data is introduced by selectively programming 0`s into the desired bit locations. Although only 0`s will be programmed both 1`s and 0`s can be presented in the data.

During programming address and data are applied to the address and data pins of the EPROM. When address and data are stable , program pulse is applied to the program input of EPROM . The program pulse duration is around 50 ms and its amplitude depends on EPROM IC. It is typically 5:5V to 25V. In EPROM , it is possible to program any location at any time – either individually, sequentially or at random.

4.2.3EEPROM (Electrically Erasable Programmable ROM)

EEPROM s also uses MOS circuitry very similar to that of EEPROM. Data is stored as charge or no charge on an insulated layer or an insulated floating gate in the device. The insulated layer is made very thin. Therefore, a voltage as low as 20 to 25V can be used to move charges across thin barrier in either direction for programming or erasing. EEPROM allows selective erasing at the register level rather than erasing all the information since the information can be changed by using electrical signals. The EEPROM memory also has a special chip erase mode by which entire chip can be erased in 10ms. The time is quite small as compared to time required to erase EPROM and it can be erased and reprogrammed with device right in the circuit. However, EEPROMs are most expensive and the least dense ROMs.

4.3RAM (RANDOM ACCESS MEMORY)

There are 2 types
- Static RAM
- Dynamic RAM

4.3.1STATIC RAM:

Memories that consist of circuits capable of retaining their state as long as power is applied are known as static memories. These are RAM and hence combinely called Static RAM memories.

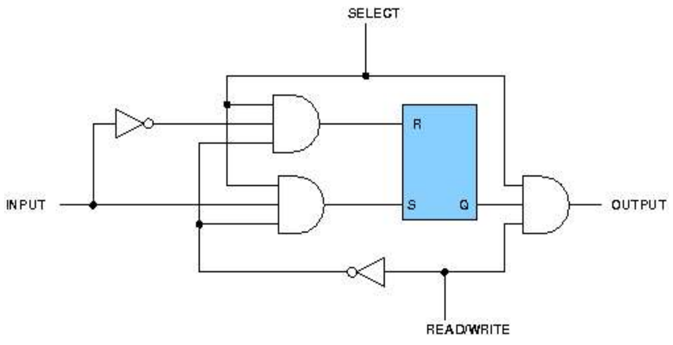


Figure 4.4: One bit binary cell

Fig shows the One bit binary cell for static RAM. The storage part of the cell is modified by an SR latch with associated gates to form a D latch The binary cell is capable of storing one bit in its internal latch. Most of the static RAMs are built using MOS technology but some are built using bipolar technology. In this section we will see both the types.

4.3.1.1Bipolar RAM CELL

Figure 4.1.5shows a simplified schematic of a bipolar memory cell. The memory cell is implemented using TTL (Transistor-Transistor Logic) multiple emitter technology. It stores 1 bit of information. It is nothing but a flip-flop. It can store either 0 or 1 as long as power is applied and it can set or reset to store either 1 or 0 respectively.

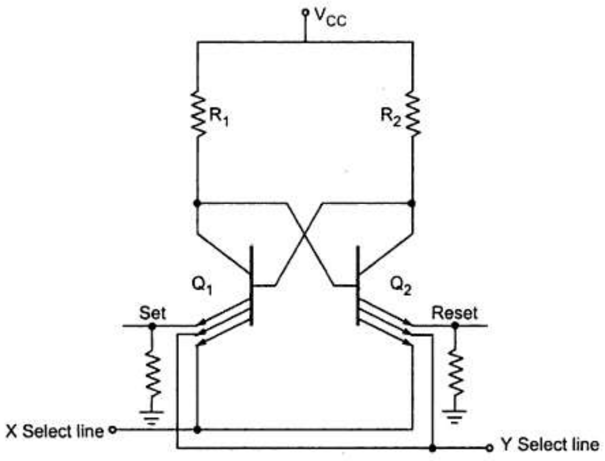


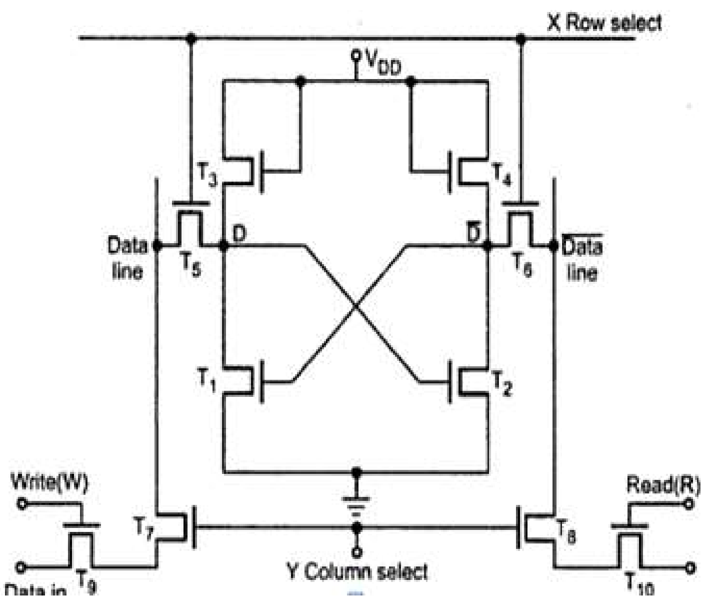
Figure 4.5: Bipolar RAM CELL

Operation:

The X and Y select input lines select a cell from matrix. The Q1 and Q2 are cross coupled inverters hence one is always OFF while the other is ON. A"1" is stored in the cell if Q1is conducting and Q2 is OFF. A "0" is stored in the cell if Q2 is conducting and Q1 is OFF. The state of the cell is changed to a "0" by pulsing a HIGH on the Q1 (set) emitter. This turns OFF Q1. When Q1 is turned OFF,Q2 is turned ON. As long as Q2 is ON, its collector is LOW and Q1 is held OFF. A 1 can be rewritten by pulsing the Q2 (reset) emitter high.

4.3.1.2MOSFET RAM cell

Figure shows a simplified schematic of MOS static RAM cell. Enhancement mode MOSFET transistors are used to make this RAM cell. It is very similar to TTL cell.



Here, T_1 and T_2 form the basic cross coupled inverters and T_3 and T_4 act as load resistors for T_1 and T_2. X and Y lines are used for addressing the cell. When X and Y both are high, cell is selected. When X = 1, T_5 and T_6 get ON and the cell is connected to the data and $\overline{\text{data}}$ line. When Y = 1, T_7 and T_8 are made ON. Due to this, either read or write operation is possible.

Write Operation : Write operation can be enabled by making W signal high. With write operation enabled, if data-in signal is logic 1, node D is also at logic 1. This turns ON T_2 and T_1 is cutoff. If new data on data-in pin is logic 0, T_2 will be cutoff and T_1 will be turned ON.

Read Operation : Read operation can be enabled by making R signal high. With read operation enabled, T_{10} becomes ON. This connects the data output ($\overline{\text{Data}}$) line to the data out and thus the complement of the bit stored in the cell is available at the output.

Fig. 11.8 shows the organization MOS static RAM IC Intel 2167. It is 16 K × 1 RAM.

To enable Read and Write operation $\overline{CS}$ must be low. If $\overline{CS}$ and $\overline{WE}$ both are low, it is a write operation. In write operation, data on the D_{in} input is written into the addressed cell while the output will remain in the high impedance state.

For Read operation $\overline{CS}$ signal must be low and $\overline{WE}$ signal must be high. In read operation, data from the addressed cell appears at the output while the input buffer is disabled.

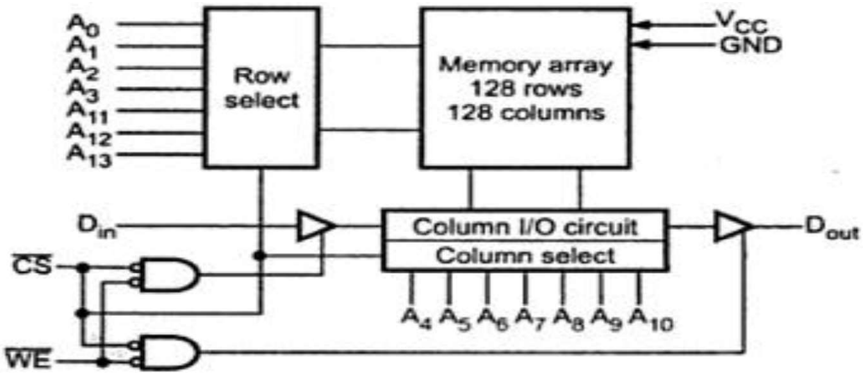


Figure4.7:Memory organization for IC 2167

When cs' is high both input and output buffers are disabled and the chip is effectively electrically disconnected. This makes the IC to operate in power down mode. In power down mode it takes only 40Ma current as compared to 125mA in the active mode.

4.3.2 DYNAMIC RAM:

•DRAM stores the data as a charge on the capacitor. Figure shows the dynamic RAM cell. A DRAM contains thousands of such memory cells. When column (sense) and row (control) lines go high, the MOSFET conducts and charges the capacitor. When the Column and RW lines go low ,the MOSFET opens and retain its charge. In the way, it stores 1 bit. Since only a single MOSFET and capacitor are needed , the DRAM contains more memory cells compared to static RAM per unit area.

•The disadvantage of DRAM is that it needs refreshing f charge on the capacitor after every few milliseconds. This complicates the system design , since it requires the extra hardware to control refreshing of DRAMs.

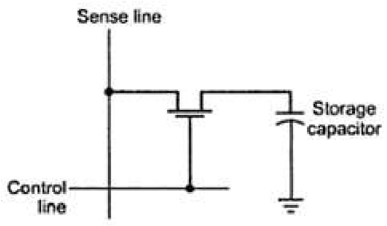


Fig:4.8 Dynamic RAM

Static RAM	Dynamic RAM
1. Static RAM contains less memory cells per unit area.	1. Dynamic RAM contains more memory cells as compared to static RAM per unit area.
2. It less access time hence faster memories	2. Its access time is greater than static RAMs.
3. Static RAM consists of number of Flip-flops. Each flip-flop stores one bit.	3. Dynamic RAM stores the data as a charge on the capacitor. It consists of MOSFET and the capacitor for each cell.
4. Refreshing circuitry is not required.	4. Refreshing circuitry is required to maintain the charge on the capacitors after every few milliseconds. Extra hardware is required to control refreshing. This makes system design complicated.
5. Cost is more.	5. Cost is less.

4.4MEMORY DECDING:

Fig shows 16*8 bit memory chip: There are 16 words of 8 bits each.

A memory with 16 words needs 4 address lines: The 4 address inputs through a 4*16 decoder to select one of the 16 words.

The decoder is enabled with a memory enable input. When the memory enable is 0,all outputs of the decoder are 0 and none of the memory words are selected. With a memory select at 1, one of the 16 words are selected, dictated by the value in the 4 address lines .

Once a word has been selected, read/write inputs determine the operation. During write operation , the data available in the input lines are transferred into the 8 memory cells of the selected word . The memory cells that are not selected are disabled and their previous binary values remain unchanged.

During write operation , the data from 8 memory cells of the selected word is available in the data output lines.

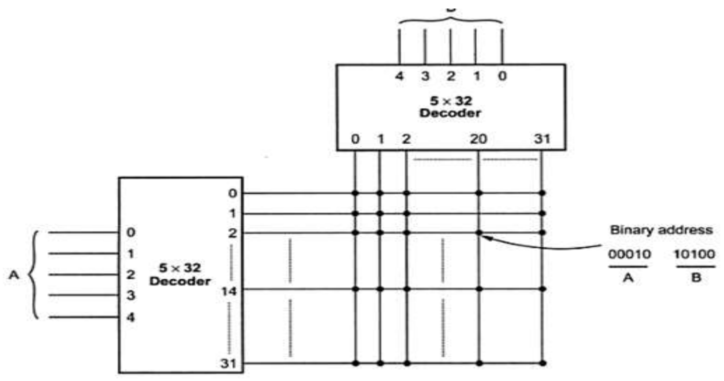


Figure: 4.9 Two dimensional decoding schemes

COINCIDENT DECODING

A decoder with k inputs and 2^k outputs requires 2^k AND gates with k inputs per gate We can reduce the total number of gates and the number of inputs per gate by employing 2 decoders in a 2 dimensional selection scheme.

The fig shows the 2 dimensional decoding scheme: here two k/2 input decoders are used instead of one k- input decoder . One decoder performs the row selection and the other column section in a 2 dimensional matrix configuration.

The 2 dimensional selection pattern shown in fig : k=10,i:e,10 decoder inputs.

These decoder inputs are divided by 2 get individual decoder inputs, i:e, 5 inputs for each decoder. Therefore ,here instead of using a single 10*1024decder, we use two 5*32 decoders.

With a single decoder we would need 1024 AND gates with 10 inputs in each. In the 2-decder case ,we need 64 AND gates with 5 inputs in each.

The 5 most significant bits of the address go to input A and the 5 least significant bits go to input B. Each word within the memory array is selected by the coincidence of one A and one B line. Thus, each word in memory is selected by the coincidence between 1 of 32 rows and 1 of 32 columns of a total of 1024 words.

4.5Programmable Logic Devices

Logic devices constitute one of the three important classes of devices used to build digital electronics systems, memory devices and microprocessors being the other two. Memory devices such as ROM and RAM are used to store information such as the software instructions of a program or the contents of a database, and microprocessors execute software instructions to perform a variety of functions, from running a word-processing program to carrying out far more complex tasks. Logic devices implement almost every other function that the system must perform, including device-to-device interfacing, data timing, control and display operations and so on. So far, we have discussed those logic devices that perform fixed logic functions decided upon at the manufacturing stage. Logic gates, multiplexers, demultiplexers, arithmetic circuits, etc., are some examples. Sequential logic devices such as flip-flops, counters, registers, etc., to be discussed in the following chapters, also belong to this category of logic devices. In the present chapter, we will discuss a new category of logic devices called programmable logic devices (PLDs). The function to be performed by a programmable logic device is undefined at the time of its manufacture. These devices are programmed by the user to perform a range of functions depending upon the logic capacity and other features offered by the device. We will begin with a comparison of fixed and programmable logic, and then follow this up with a detailed description of different types of PLDs in terms of operational fundamentals, salient features, architecture and typical applications. A brief introduction to the devices offered by some of the major manufacturers of PLDs and PLD programming languages is given towards the end of the chapter.

4.5.1 Programmable ROMs

PROM (Programmable Read Only Memory) and EPROM (Erasable Programmable Read Only Memory) can be considered to be predecessors to PLDs. The architecture of a programmable ROM allows the user to hardware-implement an arbitrary combinational function of a given number of inputs. When used as a memory device, n inputs of the ROM (called address lines in this case) and m outputs (called data lines) can be used to store 2nm-bit words. When used as a PLD, it can be used to implement different combinational functions, with each function being a chosen function of n variables. Any conceivable n-variable Boolean function can be made to appear at any of the m output lines. A generalized ROM device with n inputs and m outputs has 2n hard-wired AND gates at the input and m programmable OR gates at the output. Each AND gate has n inputs, and each OR gate has 2n inputs. Thus, each OR gate can be used to generate any conceivable Boolean function of n variables, and this generalized ROM can be used to produce m arbitrary n-variable Boolean functions. The AND array produces all possible minterms of a given number of input variables, and the programmable OR array allows only the desired minterms to appear at their inputs. Figure 9.3 shows the internal architecture of a PROM having four inputs lines, a hard-wired array of 16 AND gates and a programmable array of four OR gates. A cross (×) indicates an intact (or unprogrammed) fusible link or interconnection, and a dot (•) indicates a hard-wired interconnection. PROMs, EPROMs and EEPROMs (Electrically Erasable Programmable Read Only Memory) can be programmed using standard PROM programmers. One of the major disadvantages of PROMs is their inefficient use of logic capacity. It is not economical to use PROMs for all those applications where only a few minterms are needed. Other disadvantages include relatively higher power consumption and an inability to provide safe covers for asynchronous logic transitions. They are usually much slower than the dedicated logic circuits. Also, they cannot be used to implement sequential logic owing to the absence of flip-flops.

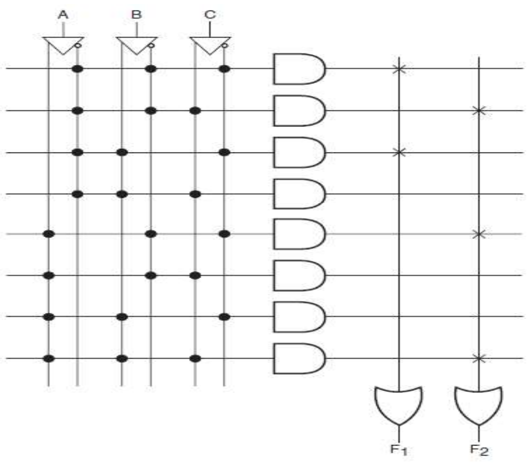


Figure 4.10. Internal architecture of PROM

Problem

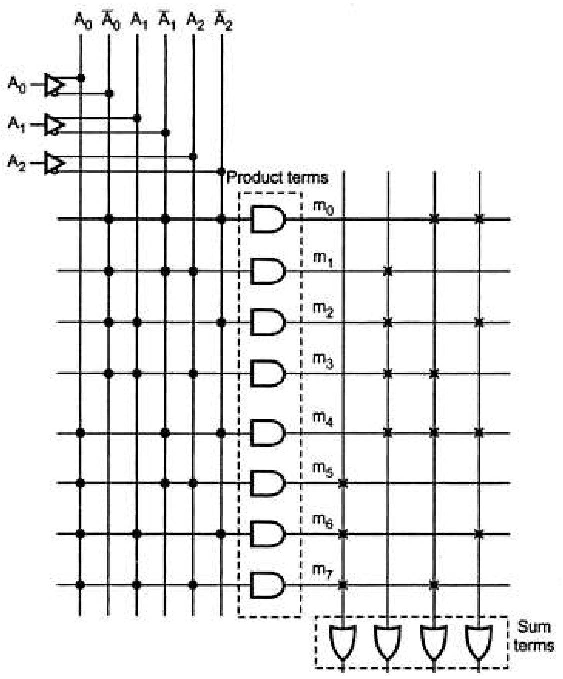
Example: Design a combinational using a PROM. The circuit accepts 3 bit binary number and generates its equivalent excess 3 code

Solution:

Step 1: Draw the truth table for the given combinational circuits

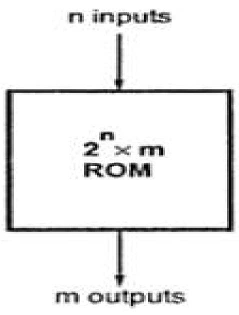
Inputs			Outputs			
A_2	A_1	A_0	B_3	B_2	B_1	B_0
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	1	0

Step 2: Draw the implementation diagram



4.5.2Programmable Logic Array

A programmable logic array (PLA) device has a programmable AND array at the input and a programmable OR array at the output, which makes it one of the most versatile PLDs. Its architecture differs from that of a PROM in the following respects. It has a programmable AND array rather than a hard-wired AND array. The number of AND gates in an m-input PROM is always equal to 2m. In the case of a PLA, the number of AND gates in the programmable AND array for m input variables.



A programmable read only memory (PROM) is a device that includes both the decoder and the OR gates within a single IC. It consists of n input lines m output lines. Each bit combination of the input variable is called an address. Each bit combination that comes out of the output lines is called a word. The number of bits per word is equal to the number of output lines m. The address specified in binary number denotes one of the minterms of n variables. The number of distinct addresses possible with n input variables is 2^n. An output word can be selected by a unique address and since there are 2^n distinct addresses in PROM, The word available on the output lines at any given time depends on the address value applied to the input lines.

Let us consider 64x4 PROM. The PROM consists of 64 words of 4 bits each. This means that there are four output lines and particular word from 64 words presently available on the output lines is determined from the six input lines. There are only six inputs in a 64x4 PROM because 2^6=64 and with six variables, we can specify 64 addresses or minterms. For each address input, there is a unique selected word. Thus, if the input address is 000000, word number 0 is selected and applied to the output lines. If the input address is 111111, word number 63 is selected and applied to the output lines.

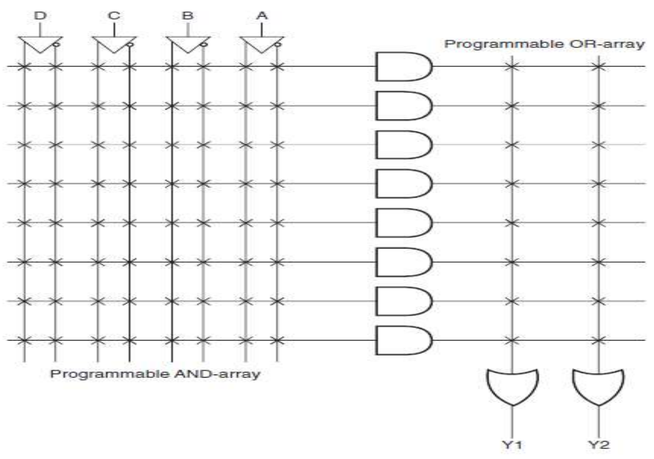


Figure 4.11.Internal architecture of PLA

Example: A combinational circuit is defined by the functions

F1=∑m (3,5,7)   F2=∑m(4,5,7)

Implement the circuit with a PLA having 3 inputs, 3 product terms and 2 outputs.

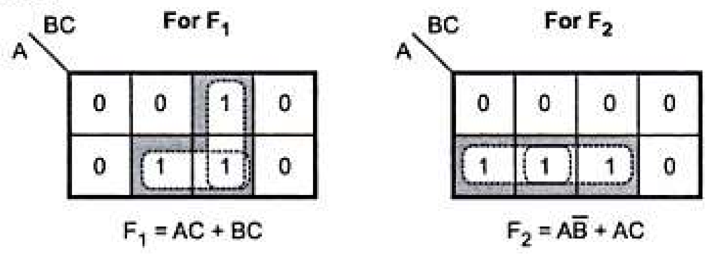
Solution:

Step 1: Determine the truth table for the given Boolean functions

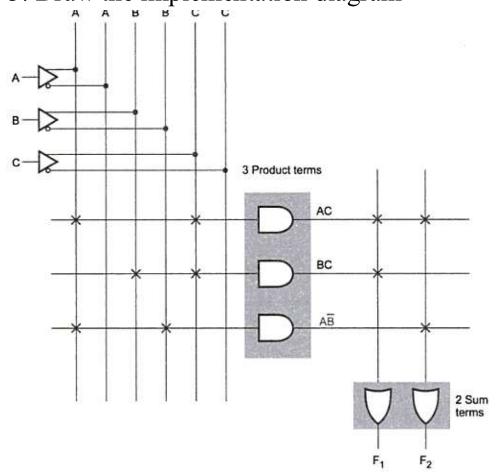
A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Step 2:

K-map simplification



Step 3: Draw the implementation diagram



4.5.3 Programmable array logic (PAL)

PAL architecture has a programmable AND array at the input and a fixed OR array at the output. The programmable AND array of a PAL device is similar to that of a PLA device. That is, the number of programmable AND gates is usually smaller than the number required to generate all possible minterms of the given number of input variables. The OR array is fixed and the AND outputs are equally divided between available OR gates. For instance, a practical PAL device may have eight input variables, 64 programmable AND gates and four fixed OR gates, with each OR gate having 16 inputs. That is, each OR gate is fed from 16 of the 64 AND outputs. Figure 9.5shows the internal architecture of a PAL device that has four input lines, an array of eight AND gates at the input and two OR gates at the output, to introduce readers to the arrangement of various building blocks inside a PAL device and allow them a comparison between different programmable logic devices.

4.5.3.1PAL Logic Structure

In order to illustrate the logical organization of PALs, we go back to our 3-input,4-output example of Figure 4.1., Figure 4.15shows PAL implementation of this circuit.

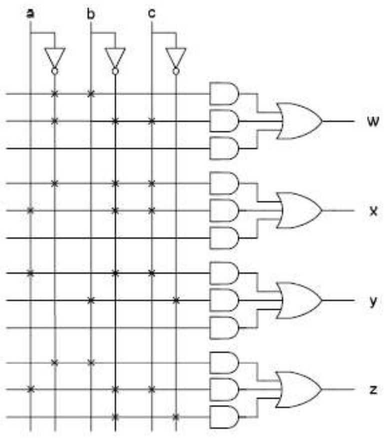


Figure 4.12 PAL implementation

The PAL structure of Figure 4.15 has a programmable AND-plane and a fixed OR-plane. Product terms are formed in the AND-plane and three such terms are used as OR gate inputs in the OR-plane. This structure allows a maximum of three product terms per output. Implementing expressions of Figure 4.13is done by programming fuses of the AND-plane of the PAL. The z output uses all three available product terms and all other outputs use only two.

Example: Implement the following Boolean functions using PAL.

$W(A,B,C,D)=\sum m(0,2,6,7,8,9,12,13)$

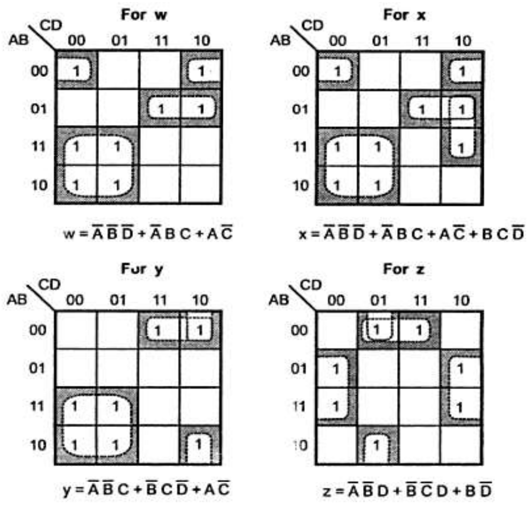
$X(A,B,C,D)=\sum m(0,2,6,7,8,9,12,13,14)$

$Y(A,B,C,D)=\sum m(2,3,8,9,10,12,13)$

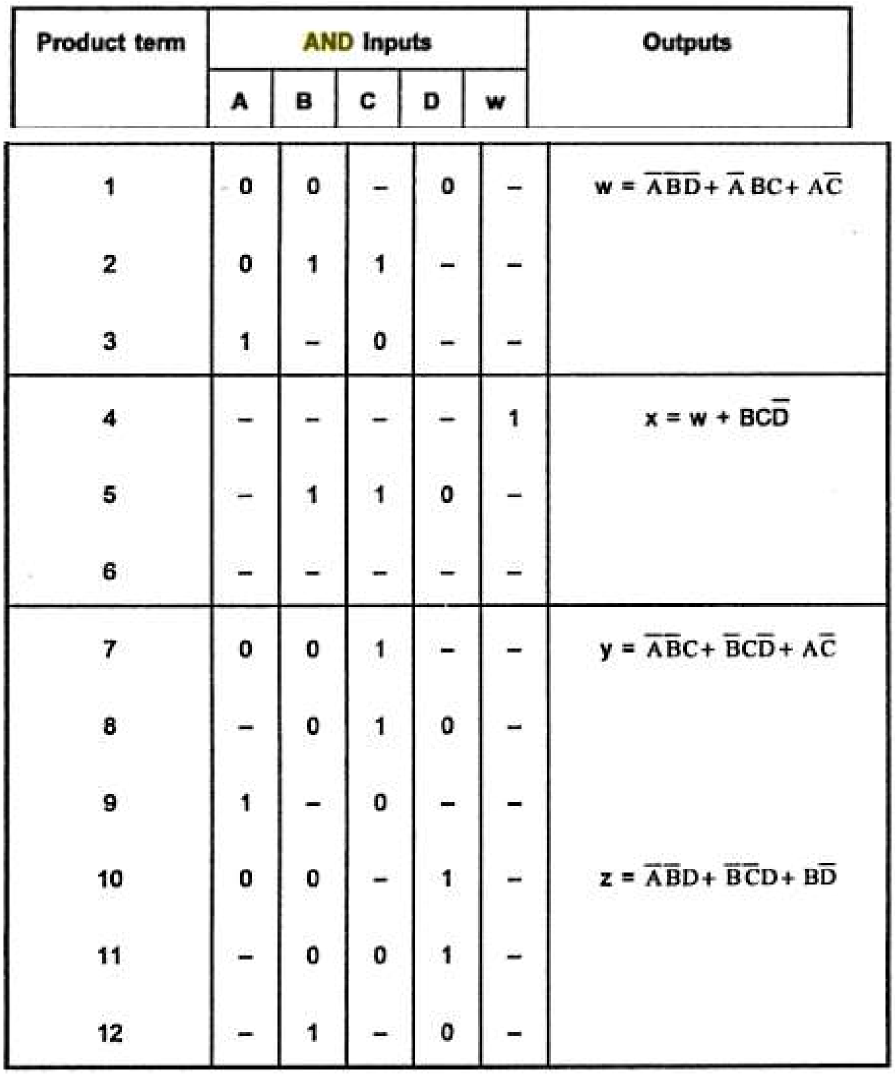
$Z(A,B,C,D)=\sum m(1,3,4,6,9,12,14)$

Solution: Let us simplify the four functions for minimum number of terms.

K-map simplification

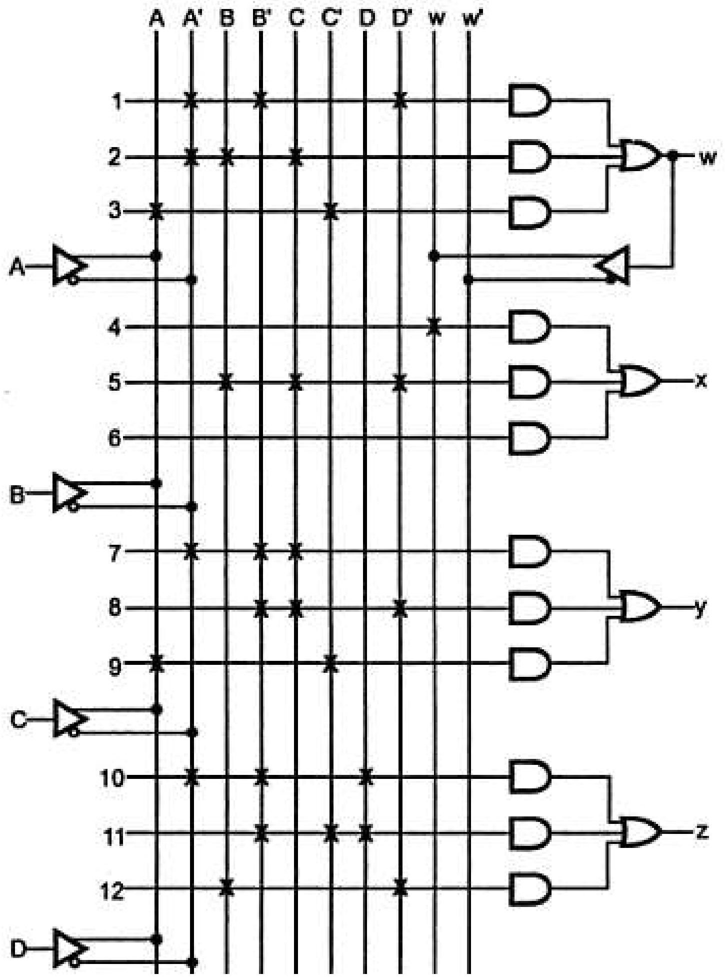

For w
For x
For y
For z
$w=\bar{A}\bar{B}\bar{D}+\bar{A}BC+A\bar{C}$
$x=\bar{A}\bar{B}\bar{D}+\bar{A}BC+A\bar{C}+BC\bar{D}$
$y=\bar{A}\bar{B}C+\bar{B}C\bar{D}+A\bar{C}$
$z=\bar{A}\bar{B}D+\bar{B}\bar{C}D+B\bar{D}$

Note that function x has four product terms. Three of them are equal to w. Therefore, we can write x=w+BCD'.
In the last section we have seen the PLA program table. The program table for PAL is similar to PLA table. Table shows PAL program table with product terms, AND inputs and outputs.



Product term	AND Inputs					Outputs
	A	B	C	D	w	
1	0	0	–	0	–	$w = \overline{A}\,\overline{B}\,\overline{D} + \overline{A}BC + A\overline{C}$
2	0	1	1	–	–	
3	1	–	0	–	–	
4	–	–	–	–	1	$x = w + BC\overline{D}$
5	–	1	1	0	–	
6	–	–	–	–	–	
7	0	0	1	–	–	$y = \overline{A}\,\overline{B}C + \overline{B}C\overline{D} + A\overline{C}$
8	–	0	1	0	–	
9	1	–	0	–	–	
10	0	0	–	1	–	$z = \overline{A}\,\overline{B}D + \overline{B}\,\overline{C}D + B\overline{D}$
11	–	0	0	1	–	
12	–	1	–	0	–	

PAL Implementation diagram



4.5.3.2 Product Term Expansion

The limitation on the number of product terms per output in a PAL device can be overcome by providing feedbacks from PAL outputs back into the AND-plane. These feedbacks are used in the AND-plane just like regular inputs of the PAL. Such a feedback allows ORing a subset of product terms of a function to be fed back into the array to further be ORed with the remaining product terms of the function.

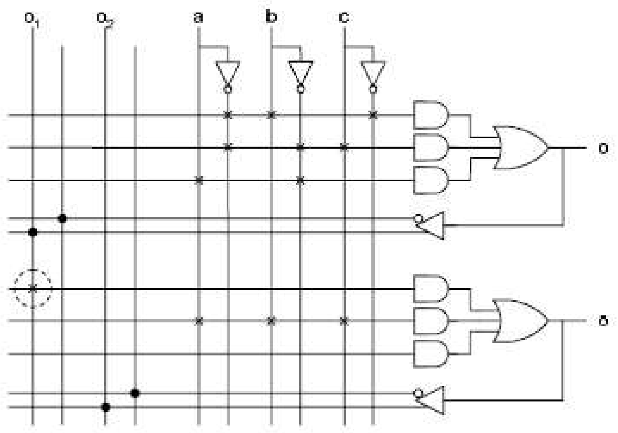


Figure 4.13A PAL with Product Term Expandability

4.5.3.3Three-State Outputs

A further improvement to the original PAL structure of Figure 4.15 is done by adding three-state controls to its outputs as shown in the partial structure of Figure 4.17.

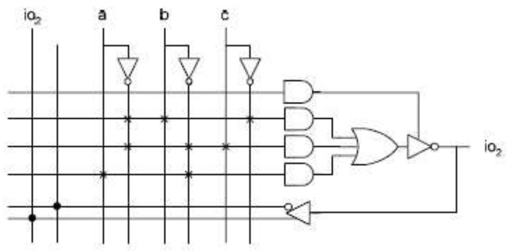


Figure 4.14PAL Structure with Three Output Control

In addition to the feedback from the output, this structure has two more advantages. First, the pin used as output or partial sum-of-products terms can also be used as input by turning off the three-state gate that drives it. Note that the lines used for feeding back outputs into the AND-plane in Figure 4.16,become connections from the io2 input into the AND-plane. The second advantage of this structure is that when io2 is used as output it becomes a three-state output that is controlled by a programmable product term. Instead of using a three-state inverting buffer, an XOR gate with three-state output and a fusible input.

4.5.3.4 Registered Outputs

A major advantage of PALs over PLAs and ROMs is the capability of incorporating registers into the logic structure. Where registers can only be added to the latter two structures on their inputs and outputs, registers added to PAL arrays become more integrated in the input and output of the PAL logic.

As an example structure, consider the registered output of Figure 4.19.The input/output shown can be used as a registered output with three-state, as a two-state output, as a registered feedback into the logic array, or as an input into the AND-plane.

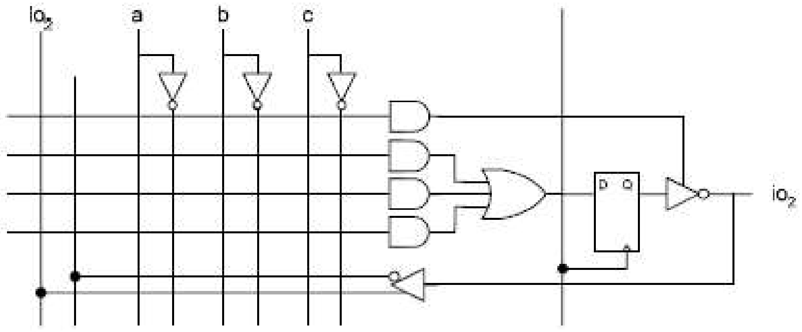


Figure 4.15 Output inversion control

A further enhancement to this structure provides logic for bypassing the output flip-flop when its corresponding I/O pin is being used as output. This way, PAL outputs can be programmed as registered or combinational pins. Other enhancements to the register option include the use of asynchronous control signals for the flip-flop, direct feedback from the flip-flop into the array, and providing a programmable logic function for the flip-flop output and the feedback line.

4.6Generic Array Logic

A generic array logic (GAL) device is similar to a PAL device and was invented by Lattice Semiconductor. It differs from a PAL device in that the programmable AND array of a GAL device can be erased and reprogrammed. Also, it has reprogrammable output logic. This feature makes it particularly attractive at the device prototyping stage, as any bugs in the logic can be corrected by reprogramming. A similar device called PEEL (Programmable Electrically Erasable Logic) was introduced by the International CMOS Technology (ICT) Corporation.

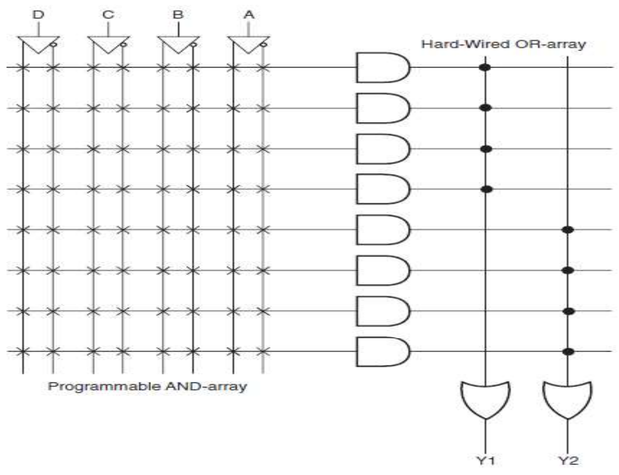
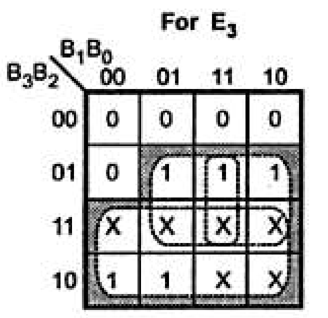


Figure 4.16 internal architecture of GAL

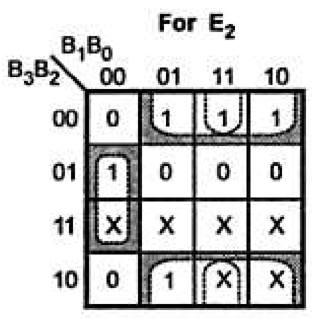
Example : Design a BCD to Excess -3 code converter and implement using suitable PLA.
Solution: Let us derive the truthtable of BCD to Excess-3 converter as shown in table

Decimal	BCD Code				Excess-3 Code			
	B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

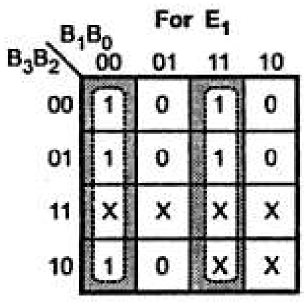
Kmap Simplification



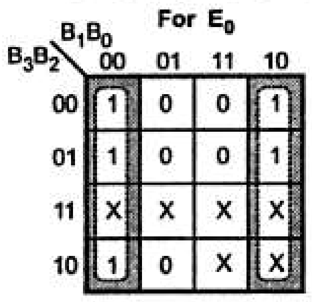
$E_3 = B_3 + B_2B_0 + B_2B_1$



$E_2 = B_2\overline{B}_1\overline{B}_0 + \overline{B}_2B_0 + \overline{B}_2B_1$



$E_1 = \overline{B}_1\overline{B}_0 + B_1B_0$



$E_0 = \overline{B}_1\overline{B}_0 + B_1\overline{B}_0$

PLA program table

	Product terms	Inputs				Outputs			
		B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
B_3	1	1	–	–	–	1	–	–	–
$B_2 B_0$	2	–	1	–	1	1	–	–	–
$B_2 B_1$	3	–	1	1	–	1	–	–	–
$B_2 \overline{B}_1 \overline{B}_0$	4	–	1	0	0	–	1	–	–
$\overline{B}_2 B_0$	5	–	0	–	1	–	1	–	–
$\overline{B}_2 B_1$	6	–	0	1	–	–	1	–	–
$\overline{B}_1 \overline{B}_0$	7	–	–	0	0	–	–	1	1
$B_1 B_0$	8	–	–	1	1	–	–	1	–
$B_1 \overline{B}_0$	9	–	–	1	0	–	–	–	1

4.7Field Programmable Gate Arrays

A more advanced programmable logic than the CPLD is the Field Programmable Gate Array (FPGA). An FPGA is more flexible than CPLD, allows more complex logic implementations, and can be used for implementation of digital circuits that use equivalent of several Million logic gates. An FPGA is like a CPLD except that its logic blocks that are linked by wiring channels are much smaller than those of a CPLD and there are far more such logic blocks than there are in a CPLD. FPGA logic blocks consist of smaller logic elements. A logic element has only one flip-flop that is individually configured and controlled. Logic complexity of a logic element is only about 10 to 20 equivalent gates. A further enhancement in the structure of FPGAs is the addition of memory blocks that can be configured as a general purpose RAM. Figure 4.30shows the general structure of an FPGA.

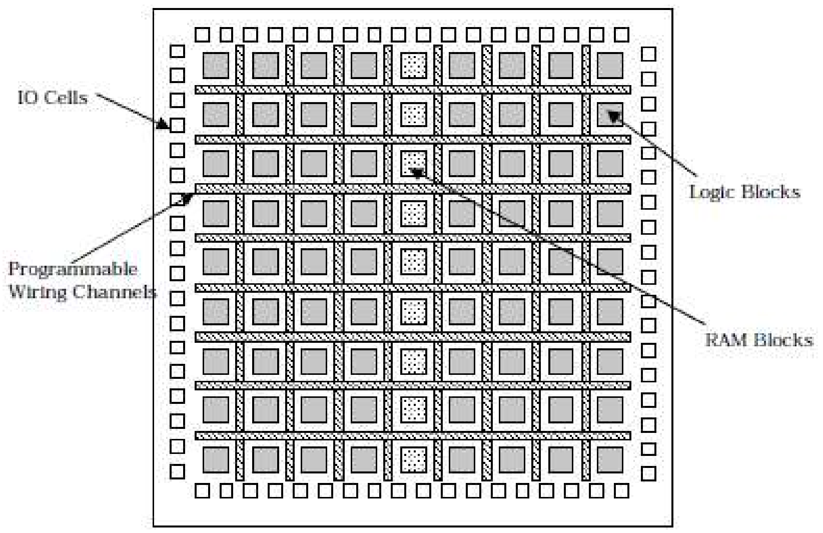


Figure 4.17 FPGA General Structure

As shown in Figure 4.17, an FPGA is an array of many logic blocks that are linked by horizontal and vertical wiring channels. FPGA RAM blocks can also be used for logic implementation or they can be configured to form memories of various word sizes and address space. Linking of logic blocks with the I/O cells and with the memories are done through wiring channels. Within logic blocks, smaller logic elements are linked by local wires. FPGAs from different manufacturers vary in routing mechanisms, logic blocks, memories and I/O pin capabilities. As a typical FPGA, we will discuss Altera's EPF10K70 that is a member of this manufacturer's FLEX 10K Embedded Programmable Logic Device Family.

4.7.1 Internal Architecture

An FPGA consists of an array of uncommitted configurable logic blocks, programmable interconnects and I/O blocks. The basic architecture of an FPGA was shown earlier in Fig. 9.7when presenting an overview of programmable logic devices. As outlined earlier, the basic difference between a CPLD and an FPGA lies in their internal architecture. CPLD architecture is dominated by a relatively smaller number of programmable sum-of-products logic arrays feeding a small number of clocked flip-flops, which makes the architecture less flexible but with more predictable timing characteristics. On the other hand, FPGA architecture is dominated by programmable interconnects, and the configurable logic blocks are relatively simpler. Logic blocks within an FPGA can be as small as the macro cells in a PLD, called fine-grained architecture, or larger and more complex, called coarse-grained architecture. However, they are never as large as the entire PLD like the logic blocks of a CPLD. This feature makes these devices far more flexible in terms of the range of designs that can be implemented with these devices. Contemporary FPGAs have an on-chip presence of higher-level embedded functions and embedded memories. Some of them even come with an on-chip microprocessor and related peripherals to constitute what is called a complete 'system on a programmable chip'. Virtex-II Pro and Virtex-4 FPGA devices from Xilinx are examples. These devices have one or more PowerPC processors embedded within the FPGA logic fabric. Figure 9.27shows a typical logic block of an FPGA. It consists of a four-input look-up table (LUT) whose output feeds a clocked flip-flop. The output can either be a registered output or an unregistered LUT output. Selection of the output takes place in the multiplexer. An LUT is nothing but a small one-bit wide memory array with its address lines representing the inputs to the logic block and a one-bit output acting as the LUT output. An LUT with n inputs can realize any logic function of n inputs by programming the truth table of the desired logic function directly into the memory.

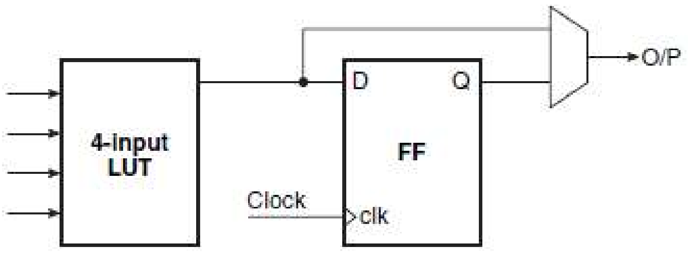


Figure 4.18.Logic block of a typical FPGA.

Logic blocks can have more than one LUT and flip-flops also to give them the capability of realizing more complex logic functions. Figure 4.18 shows the architecture of one such logic block. The architecture shown in Fig.4.18 is that of a logic block of the XC4000 series of FPGAs from Xilinx. This logic block has two four-input LUTs fed with logic block inputs and a third LUT that can be used in conjunction with the two LUTs to offer a wide range of functions. These include two separate logic functions of four inputs each, a single logic function of up to nine inputs and many more. The logic block contains two flip-flops. Figure4.18 shows another similar LUT-based architecture that uses multiple LUTs and flip-flops.

The architecture shown in Fig.4.18 is that of a logic block called a programmable function unit (PFU) by the manufacturer of AT&T FPGA devices. This logicblock can be configured either as four four-input LUTs or two five-input LUTs or one six-input LUT.

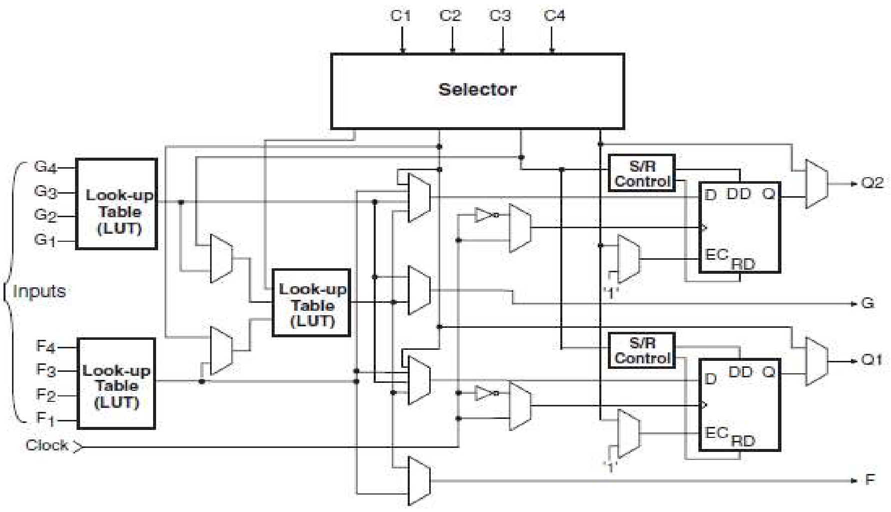


Figure 4.20.Logic blocks architecture of the XC4000 FPGA from Xilinx.

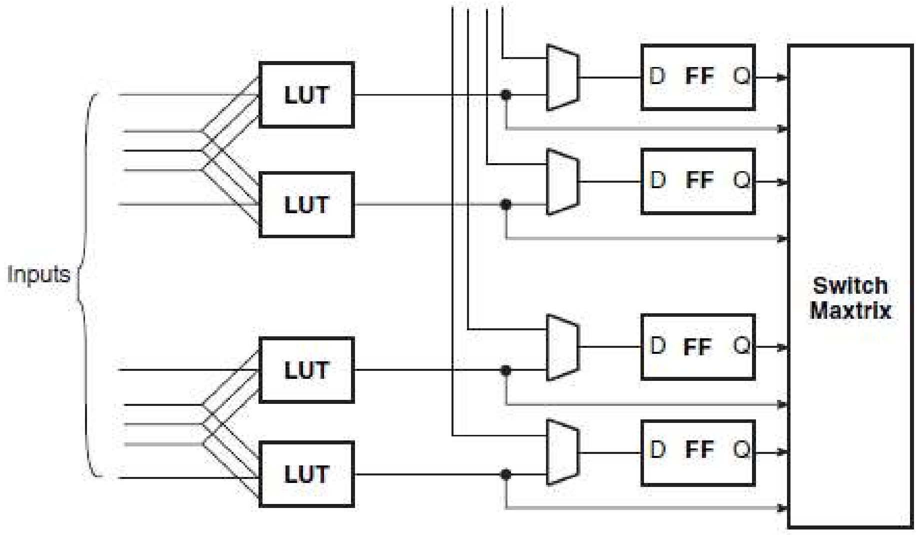


Figure: 4.21Block architecture of an AT&T FPGA.

4.7.2 Applications

In the early days of their arrival on the scene, FPGAs began as competitors to CPLDs for applications such as glue logic for PCBs. With increase in their logic capacity and capability, the availability of a large embedded memory, higher-level embedded functions such as adders and multipliers, the emergence of hybrid technologies combining the logic blocks and interconnects of traditional FPGAs with embedded microprocessors and the facility of full or partial in-system reconfiguration have immensely widened the scope of applications of FPGAs. FPGAs today offer a complete system solution on a single chip, although very complex systems might be implemented with more than one FPGA device. Some of the major application areas of FPGA devices include digital signal processing, data storage and processing, software-defined radio, ASIC prototyping, speech recognition, computer vision, cryptography, medical imaging, defense systems, bioinformatics, computer hardware emulation and reconfigurable computing. Reconfigurable computing, also called customized computing, involves the use of programmable parts to execute software rather than compiling the software to be run on a regular CPU. This has been made possible by in-system reconfiguration, which allows the internal design to be altered on-the-fly.

4.7.3Programmable Interconnect Array

Logic is routed between LABs via the programmable interconnect array (PIA). This global bus is a programmable path that connects any signal source to any destination on the device. All MAX 7000 dedicated inputs, I/O pins, and macro cell outputs feed the PIA, which makes the signals available throughout the entire device. Only the signals required by each LAB are actually routed from the PIA into the LAB. An EEPROM cell controls one input to a 2-input AND gate, which selects a PIA signal to drive into the LAB. The PIA has a fixed delay that eliminates skew between signals and makes timing performance easy to predict.

4.7.4I/O Control Blocks

The I/O control block allows each I/O pin to be individually configured for input, output, or bidirectional operation. All I/O pins have a tristate buffer that is individually controlled by one of the global output enable signals or directly connected to ground or VCC. Figure 4.29shows the I/O control block for the EPM7128S of the MAX 7000 family. The I/O control block shown here has six global output enable signals that are driven by the true or complement of two output enable signals, a subset of the I/O pins, or a subset of the I/O macro cells. When the tri-state buffer control is connected to ground, the output is tristated (high impedance) and the I/O pin can be used as a dedicated input. When the tri-state buffer control is connected to VCC, the output is enabled. The MAX 7000 architecture provides dual I/O feedback, in which macrocell and pin feedbacks are independent. When an I/O pin is configured as an input, the associated macrocell can be used for buried logic.

4.8 Complex Programmable Logic Device

Programmable logic devices such as PLAs, PALs, GALs and other PAL-like devices are often grouped into a single category called simple programmable logic devices (SPLDs) to distinguish them from the ones that are far more complex. A complex programmable logic device (CPLD), as the name suggests, is a much more complex device than any of the programmable logic devices discussed so far. A CPLD may contain circuitry equivalent to that of several PAL devices linked to each other by programmable interconnections. Figure 9.6shows the internal structure of a typical CPLD. Each of the four logic blocks is equivalent to a PLD such as a PAL device. The number of logic blocks in a CPLD could be more or less than four. Each of the logic blocks has programmable interconnections. A switch matrix is used for logic block to logic block interconnections. Also, the switch matrix in a CPLD may or may not be fully connected. That is, some of the possible connections between logic block outputs and inputs may not be supported by a given CPLD. While the complexity of a typical PAL device may be of the order of a few hundred logic gates, a CPLD may have a complexity equivalent to tens of thousands of logic gates. When compared with FPGAs, CPLDs offer predictable timing characteristics owing to their less flexible internal architecture and are thus ideal for critical control applications and other applications where a high performance level is required. Also, because of their relatively much lower power consumption and lower cost, CPLDs are an ideal solution for battery-operated portable applications such as mobile phones, digital assistants and so on. A CPLD can be programmed either by using a PAL programmer or by feeding it with a serial data stream from a PC after soldering it on the PC board. A circuit on the CPLD decodes the data stream and configures it to perform the intended logic function.

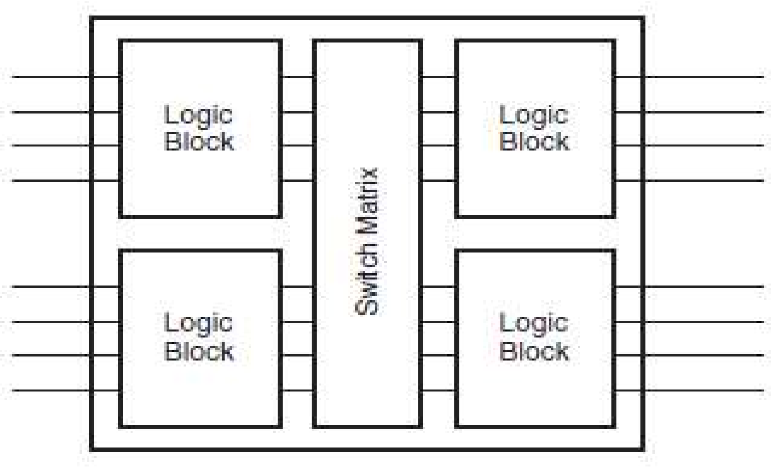


Figure 4.21. CPLDarchitecture

4.8.1 Internal Architecture

As outlined in the previous paragraph, a CPLD is nothing but the integration of multiple PLDs, a programmable interconnect matrix and an I/O control block on a single chip. Each of the identical PLDs is referred to as a logic block or function block. Figure 9.24shows the architecture of a typical CPLD. As is evident from the block schematic arrangement, the programmable interconnect matrix is capable of connecting the input or output of any of the logic blocks to any other logic block. Also, input and output pins connect directly to both the interconnect matrix as well as logic blocks.

Logic blocks may further comprise smaller logic units called macrocells, where each of the macrocells is a subset of a PLD-like logic block. Figure 4.21 shows the structure of a logic block along with its interconnections with the programmable interconnect matrix and I/O block. The horizontal grey coloured bars inside the logic block constitute an array of macrocells. Typically, each macrocell comprises a set of product terms generated by a subset of the programmable AND array and feeding a configurable output logic. The output logic typically comprises an OR gate, an EX-OR gate and a flip-flop. The flip-flop in the case of most contemporary CPLDs is configurable as a D-type, J-K, T, or R-S flip-flop or can even be transparent. Also, the OR gate can be fed with any or all of the product terms generated within the macrocell.

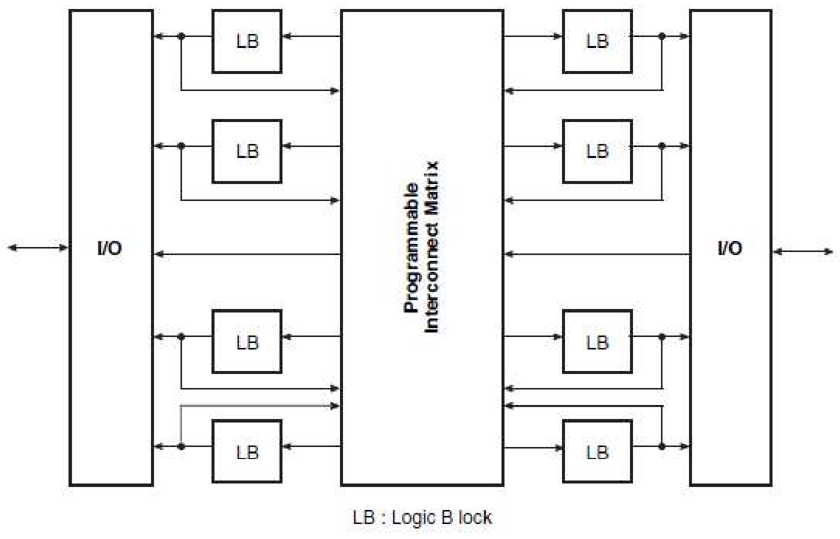


Figure 4.22CPLD architecture

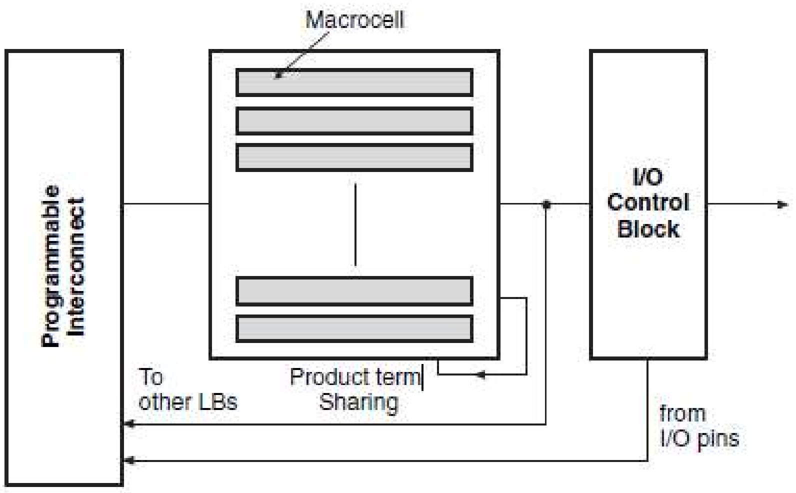


Figure 4.23 Logic block structure.

Most contemporary CPLDs also offer an architecture where the OR gate can also be fed with some additional product terms generated within other macrocells of the same logic block. For example, a logic block in the case of the MAX-7000 series of CPLDs from Altera offers this product-term flexibility, where the OR gate of each macrocell can have up to 15 additional product terms from other macrocells in the same logic block, apart from a maximum of five product terms from within the same macrocell.

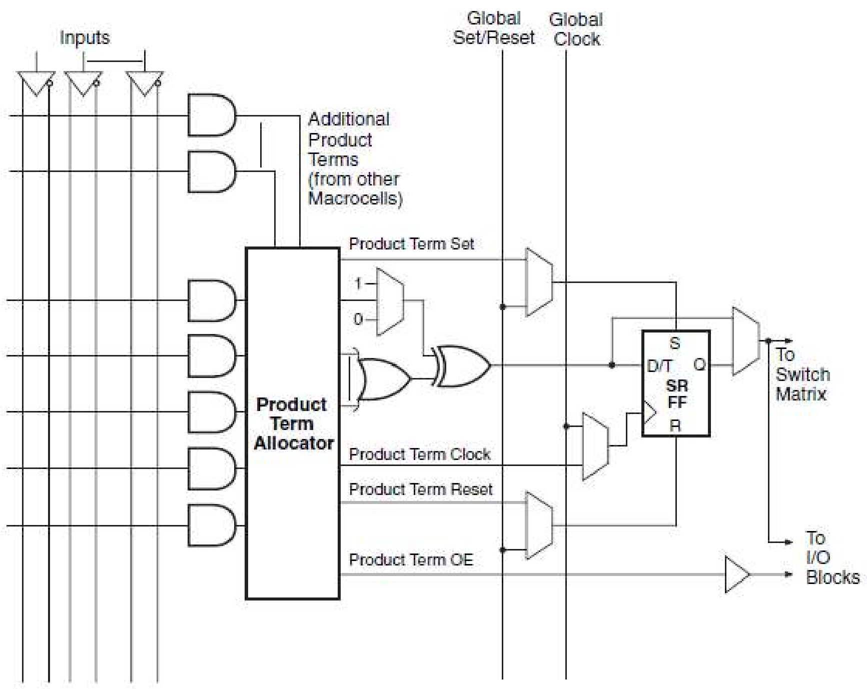


Figure 4.23Macrocell architecture

Figure 4.23 showsthe logic diagram of a macrocell typical of macrocells in the logic blocks of most contemporary CPLDs. The diagram is self-explanatory. There may be minor variations in devices from different manufacturers. For example, macrocells in the XC-7000 series CPLDs from Xilinx have two OR gates fed from a two-bit arithmetic logic unit (ALU) and its output feeds a configurable flip-flop.

4.9 Two marks Questions and Answers

1. **Explain ROM**
   A read only memory(ROM) is a device that includes both the decoder and the OR gates within a single IC package. It consists of n input lines and m output lines. Each bit combination of the input variables is called an address. Each bit combination that comes out of the output lines is called a word. The number of distinct addresses possible with n input variables is 2n.

2. **What are the types of ROM?**
   1.PROM 2.EPROM 3.EEPROM

3. **Explain PROM.**
   PROM (Programmable Read Only Memory)