MODULE-V

PROGRAMMABLE LOGIC

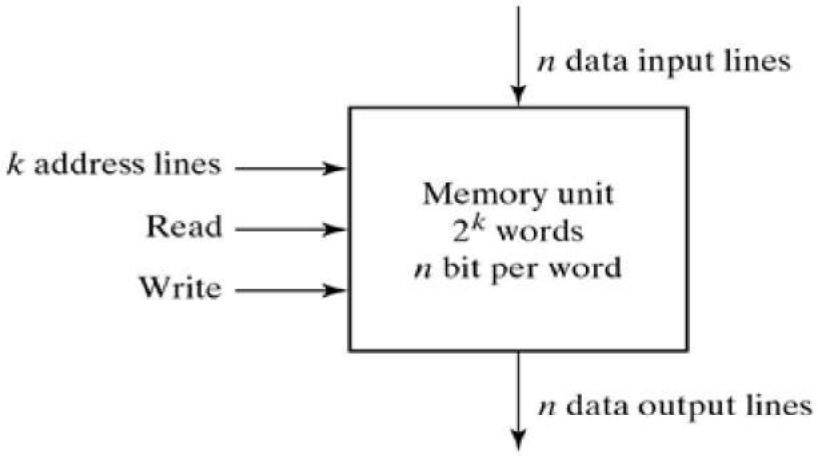
Programmable logic refers to the ability to modify the behavior or functionality of a digital logic circuit by programming. This allows for the

The components under discussion in this context are Random Access Memory (RAM), Read-Only Memory (ROM), Programmable Logic Array (PLA), Programmable Array Logic (PAL), and Sequential Programmable Devices.

5.1 RAM (Random-Access Memory)

The memory unit is responsible for storing binary data in word-sized groupings of bits. In the realm of digital information storage and transmission, it is widely accepted that one byte is equivalent to eight bits.

The size of a word is typically 2 bytes.The interaction between a memory and its surrounding environment is facilitated by various components, including data input and output lines, address selection lines, and control lines that dictate the direction of data transfer.



The following is a depiction of a block diagram representing a memory unit.

Content of a memory:

The subject matter contained within a recollection.In computer memory, a unique numerical identifier known as an address is assigned to each word. These addresses are sequentially assigned, starting from 0 and continuing up to 2k-1, where k is the total number of address lines.

The word count of a memory including one of the letters K is 210, M is 220, or G is 230. The value of 64K is equivalent to 216, while the value of 2M is equivalent to 221.The value of 4G is equal to 232.

For instance, a memory with a capacity of 64K * 10 will possess an address size of 16 bits, as 64K may be represented by 216. Additionally, each word within this memory will be composed of 10 bits. The quantity of address bits required in a memory system is contingent upon the overall capacity of words that may be accommodated inside the memory, and is not influenced by the number of bits included within each individual word. The determination of the number of bits in the address is based on the mathematical relationship 2km, where m represents the total number of words and k is the number of address bits required to fulfill the aforementioned relationship.

In certain instances, commercial memory may present the two control inputs for reading and writing in a configuration that differs considerably inside a table.The topic of discussion pertains to the concepts of write and read operations.

The process of encoding a novel term for the purpose of retention in memory:

Memory address		Memory content
Binary	Decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

Utilize the binary representation of the target word's address to connect it to the address lines. The data bits that are required to be saved in memory should be applied to the data input lines. Initiate the writing input. The process of relocating a stored word from memory: Utilize the binary representation of the target word's address to connect it to the appropriate address lines. Initiate the process of reading the input.

The various classifications of memory There are two potential operating modes for integrated circuit random access memory (RAM) components.The concepts of "static" and "dynamic" are being discussed.Static Random Access Memory (SRAM) is comprised of internal latches that are responsible for storing binary data.Please provide the necessary information. The information stored retains its validity for the duration of power availability.Submitted an application for the unit.Dynamic random-access memory (DRAM) is a type of memory that utilizes electric charges on capacitors, which are facilitated by metal-oxide-semiconductor (MOS) transistors, to store binary information.The degradation of charge on capacitors over time necessitates the periodic recharging of capacitors by the refreshing of dynamic memory at regular intervals of a few milliseconds.DRAM provides advantages such as less power consumption and increased integration of components on a single chip.SRAM, or Static Random Access Memory, has superior speed characteristics compared to other memory types. It possesses shorter read and write cycles, making it highly suitable for applications requiring fast data access. Notably, SRAM finds extensive usage in cache memory systems.There are several drawbacks associated with this technology, including its high power consumption, limited density, and pricey nature.

Volatile memory units are characterized by their tendency to lose stored information upon power off. CMOS integrated circuit Random Access Memories (RAMs), including both static and dynamic types, fall under this classification as the binary cells necessitate external power to sustain the stored data. On the other hand, nonvolatile memory, such as magnetic disk, is capable of preserving its stored data even in the absence of power supply. This particular sort of memory possesses the capacity to preserve information due to the representation of data on magnetic components through the direction of magnetization. This magnetization persists even after power is deactivated. Read-only memory (ROM) is an additional type of nonvolatile memory. Nonvolatile memory facilitates the storage of programs in digital computers, allowing for their retrieval following the computer's power cycle. Read-only memory (ROM) is utilized to store programs and data that are immutable, whilst magnetic disks are employed to retain larger applications. The aforementioned programs are dynamically loaded into the computer's random access memory (RAM) as required. Prior to the cessation of power supply, the binary data stored in the computer's random access memory (RAM) is moved to the disk in order to ensure its preservation.

5.2 Read-Only Memory (ROM)

A read-only memory (ROM) is a type of memory device that is designed to store permanent binary information. The designer is responsible for specifying the binary information, which is subsequently incorporated in the unit to create the desired connectivity pattern. Once the pattern has been established, it remains within the unit even after the power has been toggled on and off.

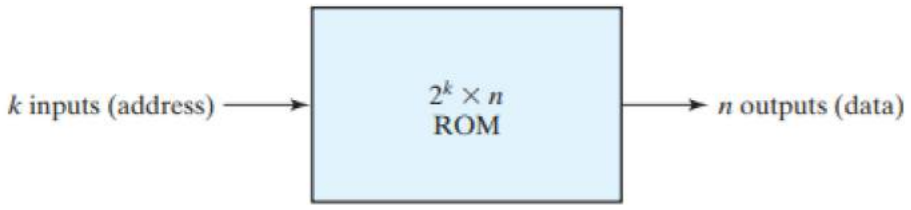


Fig. ROM block diagram

The inputs of the system are responsible for providing the memory address, while the outputs are responsible for providing the data bits corresponding to the selected memory word. The quantity of

words contained within a Read-Only Memory (ROM) is established based on the observation that k address input lines are required to uniquely identify 2k words. It should be noted that Read-Only Memory (ROM) lacks data inputs due to the absence of a write operation.

Take into consideration, as an illustrative instance, a read-only memory (ROM) with a capacity of 32 words and a word length of 8 bits. The unit comprises 32 words, with each word consisting of 8 bits. The address is formed by a sequence of five input lines representing binary values ranging from 0 to 31. The internal logic construction of this ROM is illustrated in the diagram below. The process of decoding the five inputs is accomplished using a decoder with a 5 * 32 configuration, resulting in the generation of 32 unique outputs. The decoder's output corresponds to a certain memory address. The thirty-two outputs of the decoder are interconnected with each of the eight OR gates.

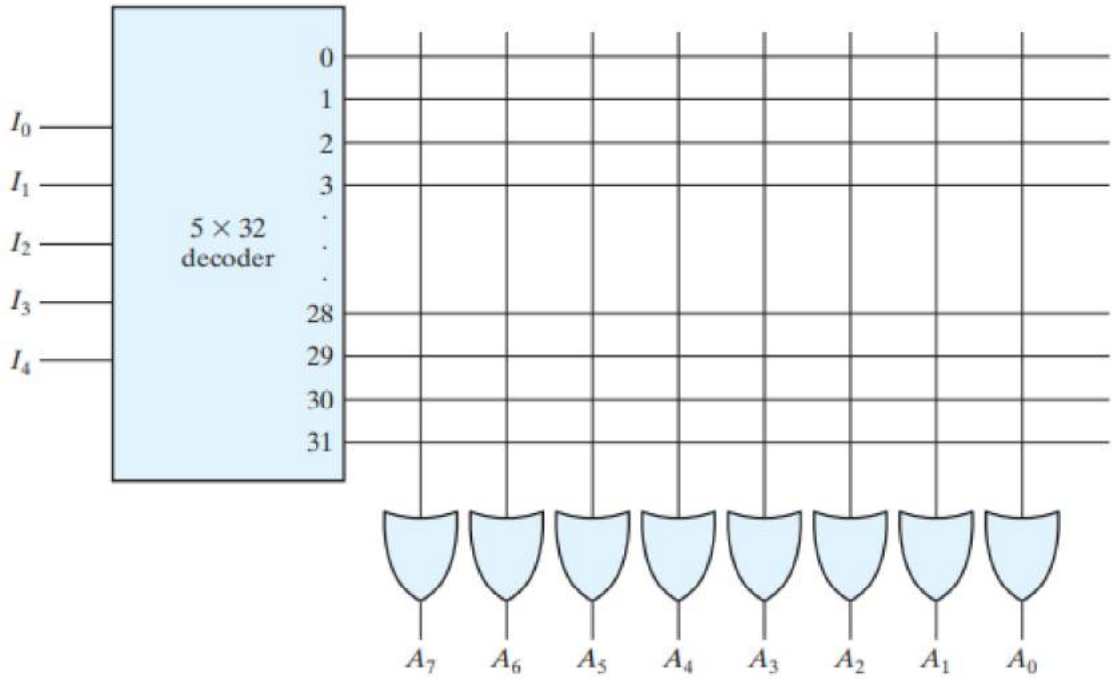


Fig. Internal logic of a 32 : 8 ROM

The internal logic of a 32:8 ROM refers to the underlying structure and organization of the Read-Only Memory (ROM) device, where the ratio of 32:8 indicates the number of address lines to the number of data output lines.

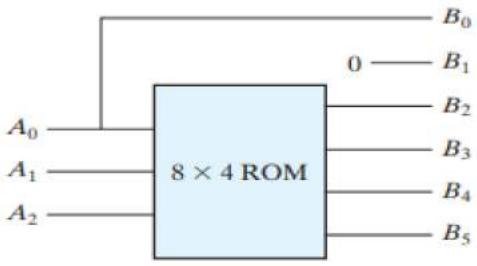
Inputs			Outputs						
A_2	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	Decimal
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

It is necessary to regard each OR gate as possessing 32 inputs. Each output signal generated by the decoder is linked to an individual input of every OR gate. The internal connections within the ROM may be determined by multiplying the number of input connections per OR gate, which is 32, by the total number of OR gates, which is 8. Thus, the ROM includes a total of 256 internal connections. Typically, a read-only memory (ROM) with dimensions 2k * n will incorporate an internal decoder of size k * 2k and n OR gates. Every OR gate in the system possesses a total of 2k inputs, all of which are directly linked to the outputs of the decoder.

A programmable interconnection between two lines can be considered as a switch that can be modified to either establish a connection (closed state) or break the connection (open state) between the two lines, hence achieving logical equivalence. The term "crosspoint" is often used to refer to the programmed junction between two lines. A range of physical devices are employed for the implementation of crosspoint switches. One of the most basic technological approaches is the utilization of a fuse, which typically establishes a connection between two places. However, this connection is disrupted or "blown" when a strong voltage pulse is applied to the fuse.

The user's text is already academic in nature. Develop a combinational circuit by employing a Read-Only Memory (ROM). The circuit is designed to receive a three-bit numerical input and produce a binary output that is equivalent to the square of the input number.

In order to encompass all potential binary numbers, a total of three inputs and six outputs are required. It is observed that the output B0 consistently matches the input A0, rendering the generation of B0 using a read-only memory (ROM) unnecessary, as it is equivalent to an input variable. Additionally, it should be noted that output B1 consistently yields a value of 0, thereby establishing it as a predetermined constant. The generation of outputs using the ROM is limited to four, as the remaining two may be easily obtained. The minimum required size of the Read-Only Memory (ROM) should consist of a minimum of three input lines and four output lines. The size of the ROM must be 8 * 4, as it is determined by three inputs that indicate eight words.



(a) Block diagram

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
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	1
1	1	1	1	1	0	0

(b) ROM truth table

Fig.ROM implementation

Types of ROMs:

Read-only memory (ROM) is a type of non-volatile memory that retains recorded information even in the absence of power supply. The purpose of this technology is to facilitate the long-term retention of data. Furthermore, it exhibits the property of random access. Users and programmers are unable to write information into a Read-Only Memory (ROM). The determination of the contents of ROMs is made by the manufacturers.

The further varieties of Read-Only Memories (ROMs) are enumerated as follows:

PROM, short for Programmable Read-Only Memory, refers to a type of memory that may be programmed by the user. The determination of its contents is made by the user. Permanent programs, data, and other information can be stored into a Programmable Read-Only Memory (PROM) by the user. The data is inputted into the system through the utilization of PROM programs. EPROM, short for erasable programmable read-only memory, refers to a type of memory chip that can be programmed and erased several times. The erasure of data in EPROMs can be achieved with the use of ultraviolet (UV) light for a duration of approximately 20 minutes. The process of erasing the EPROM IC involves the removal of the component from the computer system and subjecting it to ultraviolet (UV) radiation. The entirety of the data is wiped rather than specific chunks as chosen by the user. EPROMs are known for their cost-effectiveness and high reliability. The Electrically Erasable Programmable Read-Only Memory (EEPROM) is a type of chip that allows for convenient byte-by-byte erasure and reprogramming while still on the circuit board. The erasure process can be completed within a few milliseconds. The reprogramming capability of EEPROMs is typically constrained by a finite limit, commonly set at approximately 10,000 cycles.

A combinational programmable logic device (PLD) refers to an integrated circuit that incorporates programmable gates, which are further divided into an AND array and an OR array. This division allows for the realization of an AND-OR sum of product implementation.The PROM is designed as a fixed decoder and programmable OR array.

The PAL, which stands for Programmable AND Array and Fixed OR Array, is a type of integrated circuit that combines programmable AND gates with fixed OR gates.The programmability of both the AND and OR arrays is possible in PLA.There are four distinct methods by which the necessary pathways within a Read-Only Memory (ROM) can be programmed.

The fabrication process of mask programming The topic of discussion pertains to read-only memory (ROM) or programmable read-only memory (PROM), specifically focusing on the state of the fuse within these memory devices, whether it is blown or intact.The Erasable Programmable Read-Only Memory (EPROM) or Electrically Programmable Read-Only Memory (EPROM) can be effectively erased by subjecting it to a specific duration of exposure to ultraviolet radiation. This process results in the removal of the stored data pattern within the Read-Only Memory (ROM).

Electrically-erasable programmable read-only memory (EEPROM) is a type of memory that can be erased using an electrical signal, as opposed to the traditional method of using UV light.

COMBINATIONAL PLDs

The Programmable Read-Only Memory (PROM) is a type of combinational programmable logic device (PLD). A combinational programmable logic device (PLD) is a type of integrated circuit that consists of programmable gates that are organized into separate AND and OR arrays. This configuration allows for the construction of an AND-OR sum of product logic function. There exist three primary categories of combinational programmable logic devices (PLDs), which exhibit variations in the arrangement of the programmable connections inside the AND-OR array. Figure 3 depicts the arrangement of three Programmable Logic Devices (PLDs).

The Programmable Read-Only Memory (PROM) consists of a fixed AND array that is designed as a decoder, together with a programmable OR array. The programmable OR gates are utilized to execute the Boolean functions in the form of a sum of minterms.

The programmable array logic (PAL) is comprised of a programmable AND array and a fixed OR array. The AND gates are configured to generate the product terms that are subsequently combined by logical summation in each OR gate. The programmable logic array (PLA) is often regarded as the most versatile programmable logic device (PLD) due to its ability to program both the AND and OR arrays. The product terms within the AND array have the potential to be utilized by any OR gate in order to achieve the desired implementation of sum of products.

The figure. The fundamental arrangement of three Programmable Logic Devices (PLDs)

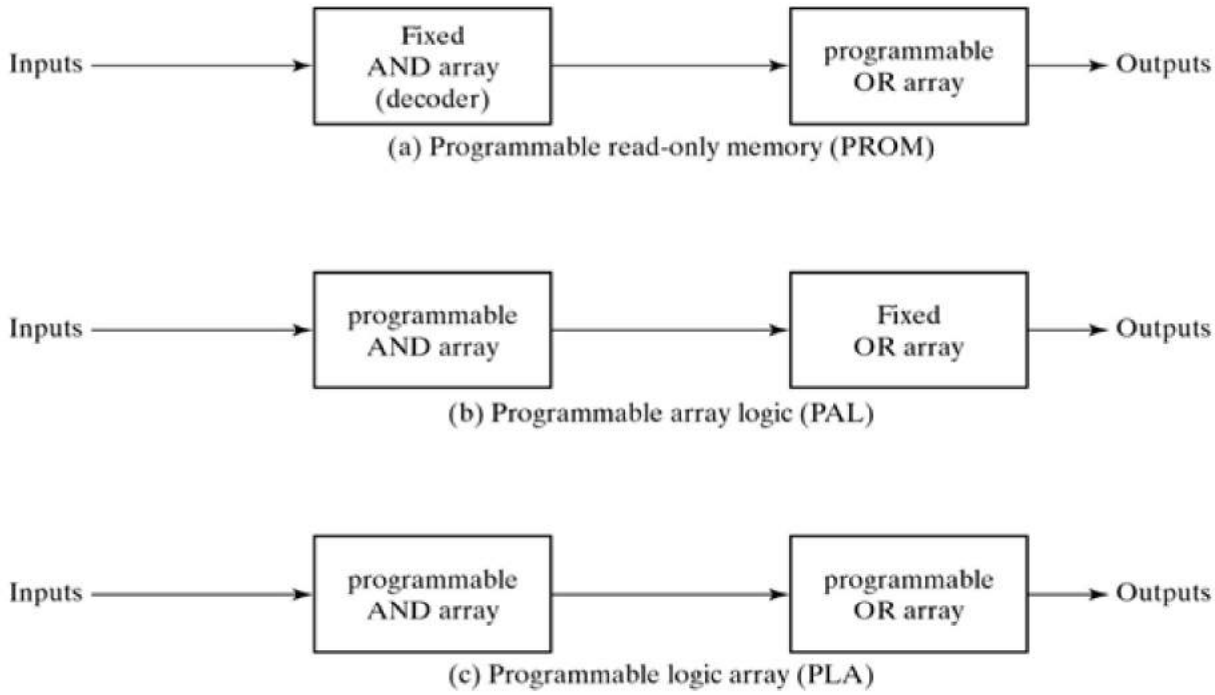


Fig. Basic configuration of three PLDs

A Programmable Logic Array (PLA) is a digital logic device that may be programmed to perform various logical functions.The Programmable Logic Array (PLA) shares conceptual similarities with Programmable Read-Only Memory (PROM), with the key distinction being that PLA lacks complete variable decoding and does not generate all possible minterms. The decoder is substituted with an array of AND gates, which can be programmed to generate any product term of the input variables. The product terms are subsequently interconnected with OR gates in order to generate the sum of products for the specified Boolean functions.The figure. The decoder in Programmable Read-Only Memory (PROM) is substituted by an array of AND gates, which can be programmed to produce any product term of the input variables.The product terms are subsequently interconnected with OR gates in order to generate the sum of products for the specified Boolean functions.The output undergoes inversion when the XOR input is linked to a logical high value of 1, as the XOR operation with 1 results in the complement of the input value (i.e., $x \oplus 1 = x'$). The output remains unchanged and is connected to 0, as the XOR operation with 0 results in the original value.

The equation F1 = AB'+AC+A'BC' represents a logical expression in Boolean algebra.

The formula F2 can be expressed as the complement of the logical disjunction of AC and BC.

The diagram employs the utilization of array logic graphic symbols to represent intricate circuits. The provided diagram illustrates the process in which each input is subjected to a buffer and an inverter. This is represented by a composite graphic symbol that encompasses both the true and complement outputs. The connection between each input and its complement is established by linking them to the inputs of each AND gate, as denoted by the intersections formed by the vertical and horizontal lines. The outputs of the AND gates are subsequently linked to the inputs of each OR gate.

The outputs of the OR gates are directed towards an XOR gate, wherein the second input can be designed to receive a signal that is equivalent to either a logic 1 or a logic 0. When the XOR input is connected to 1, the output becomes inverted. There is no alteration in the result when the XOR input is linked to 0.The product terms produced by each AND gate are enumerated beside the output of the gate in the diagram. The determination of the product term is contingent upon the identification of inputs that possess connected crosspoints, denoted by the marking of a "X". The result produced by an OR gate represents the logical addition of the chosen product terms. The resulting output can either be complemented or remain unchanged, depending on the connection of one of the inputs to the XOR gate.

The programming table that delineates the Programmable Logic Array (PLA) illustrated in Figure X. The item is included in Table 2. The programming table of the PLA is divided into three portions. The initial section presents the product terms in a numerical order. The subsequent section delineates the necessary connections between input components and AND gates. The third part delineates the interconnections between AND and OR gates.

Programmable Logic Array(PLA):

PLA Programming Table

		Inputs			Outputs (T)	(C)
	Product Term	A	B	C	F_1	F_2
AB′	1	1	0	–	1	–
AC	2	1	–	1	1	1
BC	3	–	1	1	–	1
A′BC′	4	0	1	0	1	–

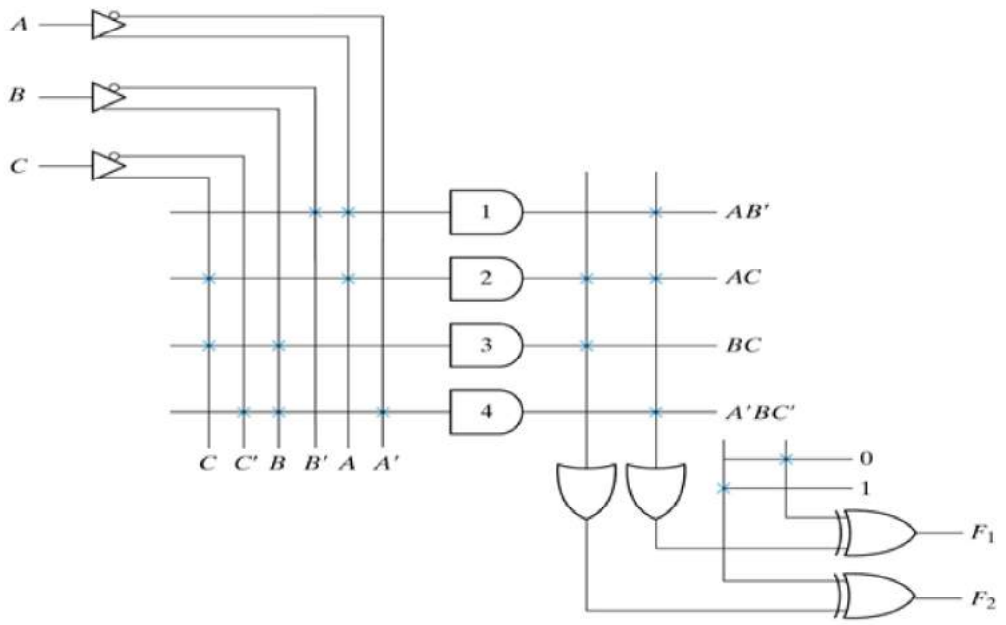


Fig. The Programmable Logic Array (PLA) is a digital logic device that consists of three inputs, four product terms, and two outputs

The output of a T input implies that the other input of the associated XOR gate should be linked to 0, whereas a C input indicates a connection to 1. The size of a Programmable Logic Array (PLA) is determined by the number of inputs, the number of product terms, and the number of outputs. In the process of creating a digital system utilizing a Programmable Logic Array (PLA), it is not necessary to explicitly depict the internal interconnections among the constituent components. A programming table for the Programmable Logic Array (PLA) is essential in order to program the PLA to provide the desired logic functionality.The programming table is a tool used in computer science to organize and present data in a structured format. It is commonly used to store Firstly, the product terms are enumerated in a numerical fashion. Secondly, it is necessary to define the specific pathways that connect the inputs to the AND gates. Thirdly, this section delineates the pathways connecting the AND and OR gates. In programming the XOR gate, it is possible to assign either a T (true) or C (complement) value to each output variable.

Ex. The objective is to design a Programmable Logic Array (PLA) to implement two Boolean functions, namely F1(A, B, C) and F2(A, B, C). The truth table for F1 is given by (0, 1, 2, 4), while the truth table for F2 is given by (0, 5, 6, 7).

The proposed solution is as follows:

Both the functions' true value and complement are simplified using the sum-of-products form. The combination that yields the lowest number of product terms is.

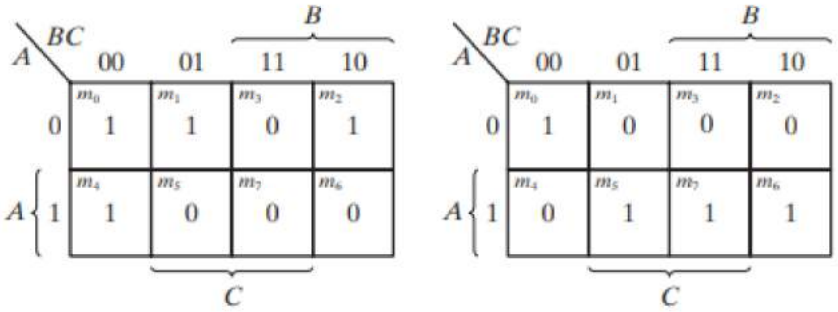
The formula F1 can be expressed as the complement of the logical disjunction of the variables AB, AC, and BC.

The equation F2 = AB + AC + A'B'C' is a logical expression in which the output F2 is determined by the inputs A, B, and C.

The aforementioned combination yields four unique product designations, namely AB, AC, BC, and ABC. The programming table for the combo is depicted in the provided figure. It is important to acknowledge that the output labeled as F1 in the table represents the actual output, despite being marked with a C. The reason for this is that the F1 signal is produced by the utilization of an AND-OR circuit, and it may be accessed at the output of the OR gate. The XOR gate performs the logical operation of complementing the input values to provide the real F1 output.

PLA programming table

	Product term	Inputs A	B	C	Outputs (C) F_1	(T) F_2
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



5.4 Programmable Array Logic:

The Programmable Array Logic (PAL) is a type of integrated circuit that consists of a fixed OR array and a programmable AND array. Due to the exclusive programmability of AND gates inside the Programmable Array Logic (PAL), it can be asserted that the PAL exhibits a relatively simplified programming process compared to the Programmable Logic Array (PLA). However, it is important to note that the PAL's programming capabilities are comparatively less versatile than those of the PLA. The diagram illustrates the logical arrangement of a standard Programmable Array Logic (PAL) device, featuring four input lines and four output lines. Every input is equipped with a buffer-inverter gate, while each output is produced by a predetermined OR gate. The unit is divided into four sections, with each section consisting of an AND–OR array that is three units wide. This terminology signifies that each section contains three programmable AND gates and one fixed OR gate. The diagram illustrates that each AND gate possesses 10 programmable input connections, represented by the intersection of 10 vertical lines with each horizontal line. The horizontal line serves as a representation of the multi-input arrangement found in the AND gate. One of the resulting signals is linked to a buffer-inverter gate and subsequently looped back to serve as input for two AND gates. A standard PAL integrated circuit typically possesses eight input ports, eight output ports, and eight segments, each including an array of AND-OR gates with a width of eight. The output terminals are occasionally stimulated by three-state buffers or inverters. When utilizing a Programmable Array Logic (PAL) device, it is necessary to simplify the Boolean functions so that they can be accommodated within each respective area. In contrast to the scenario observed in a Programmable Logic Array (PLA), it is not possible for a product term to be distributed over many OR gates. Hence, it is possible to simplify each function individually, without considering common product terms. The quantity of product terms inside each section remains constant, therefore in cases when the number

of terms within a function exceeds the capacity of a single section, it may be required to employ two sections in order to perform a single Boolean function.

To illustrate the utilization of a Programmable Array Logic (PAL) in the development of a combinational circuit, let us examine the subsequent Boolean functions, which are expressed in the sum of minterms format:

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

When reducing the four functions to their most concise form, the resulting Boolean functions are as follows:

$$w = ABC' + A'B'CD'$$
$$x = A + BCD$$
$$y = A'B + CD + B'D'$$
$$z = ABC' + A'B'CD' + AC'D' + A'B'C'D$$
$$= w + AC'D' + A'B'C'D$$

It should be noted that the function for variable z consists of four product terms. The algebraic expression representing the addition of two of these terms is equivalent to the variable w. The utilization of the variable w allows for a reduction in the quantity of words associated with z, decreasing the count from four to three. The PAL programming table exhibits similarities to the table employed for the PLA, with the distinction that solely the inputs of the AND gates necessitate programming.

Product Term	AND Inputs A	B	C	D	w	Outputs
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	—	

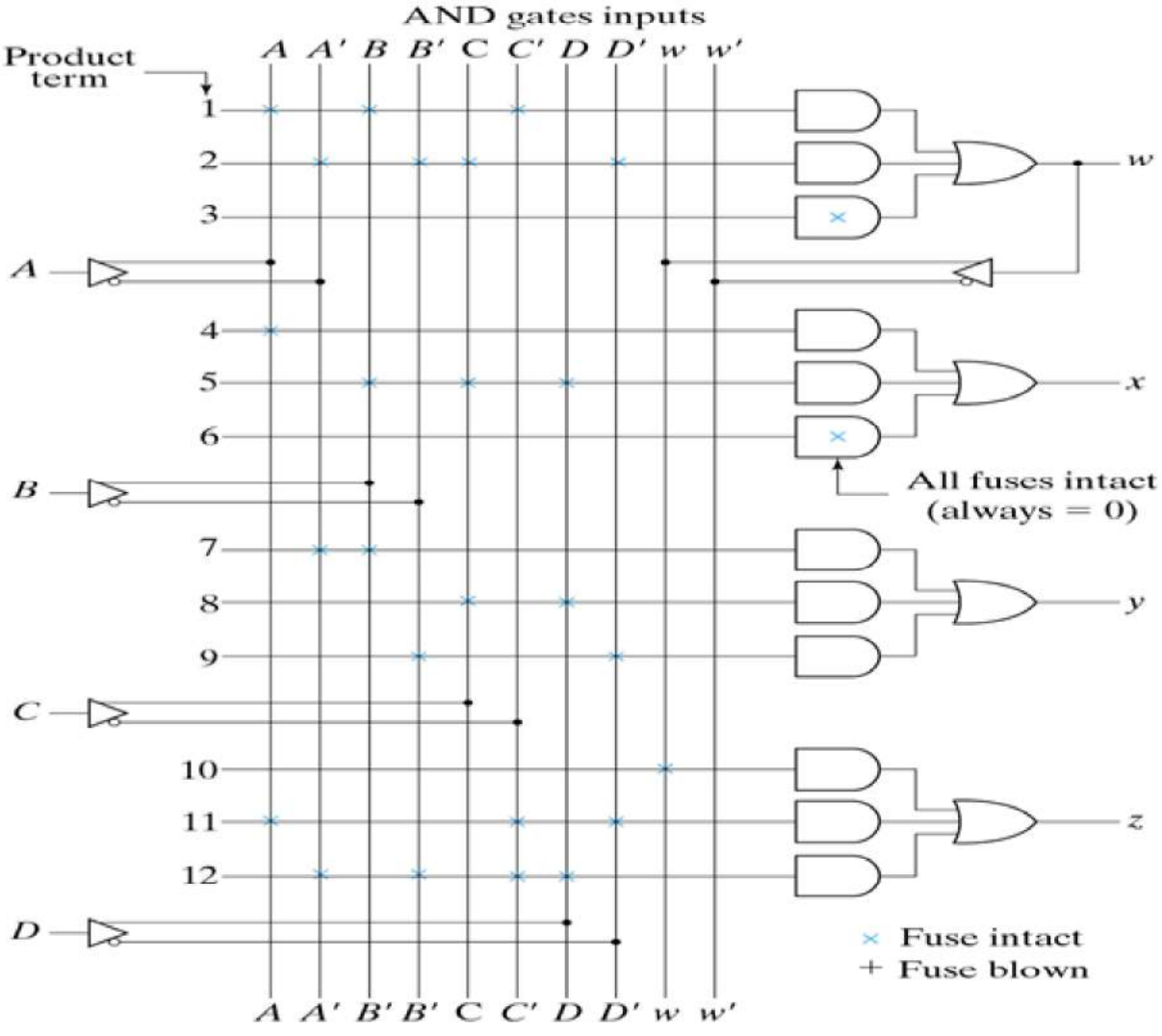


Fig. The fusion map for Phase Alternating Line (PAL) as defined in the provided table.

The table has been partitioned into four distinct sections, each containing three product terms, in order to adhere to the principles outlined in the PAL of Figure. The initial two portions necessitate a mere two product terms for the implementation of the Boolean function. The final element, pertaining to output z, necessitates the inclusion of four product names. By utilizing the output obtained from the function w, it is possible to simplify the function to a total of three terms. The graphic above displays the fuse map for the PAL, as indicated in the programming table. The symbol denoting an intact fuse is assigned to each intersection in the diagram based on the presence of a 1 or 0 in the relevant database entry. Each dash on the diagram represents a blown fuse in both the true and complement inputs. In the absence of utilizing the AND gate, it is customary to maintain the integrity of all input fuses associated with said gate. Given that the input variables receive both their true value and complement, it follows that AA equals 0, resulting in a constant output of 0 from the AND gate.

5.5 Sequential Programmable Devices:

Digital systems are typically constructed using flip-flops and gates. In the construction of combinational programmable logic devices (PLDs), the inclusion of external flip-flops becomes important due to the exclusive utilization of gates within the PLD. Sequential programmable devices encompass a combination of gates and flip-flops. By employing this method, the device has the capability to be programmed for the execution of a diverse range of functions pertaining to sequential circuits. There exist multiple commercially accessible sequential programmable devices, with each device encompassing vendor-specific variations within its respective kind. The intricacy of the internal logic of these gadgets surpasses the scope of this presentation. Hence, we shall delineate three primary categories without delving into their intricate composition:

In this study, we aim to investigate the effects of a particular drug on the growth rate A Sequential Programmable Logic Device (SPLD), sometimes known as a Simple Programmable Logic Device, is a type of electronic device that may be programmed to perform specific logic functions in a sequential manner.

The user's text is already academic and does not require any rewriting. A complex programmable logic device (CPLD) is a type of electronic component that is capable of being programmed to perform various logic functions.

The user's text is incomplete and does not provide any information to be rewritten in an academic A field-programmable gate array (FPGA) is a type of integrated circuit that can be programmed and reprogrammed to perform various digital functions.

The sequential programmable logic device (PLD) is occasionally denoted as the basic PLD in order to distinguish it from the intricate PLD. The integrated circuit chip incorporates flip-flops alongside the AND-OR array as part of the SPLD.

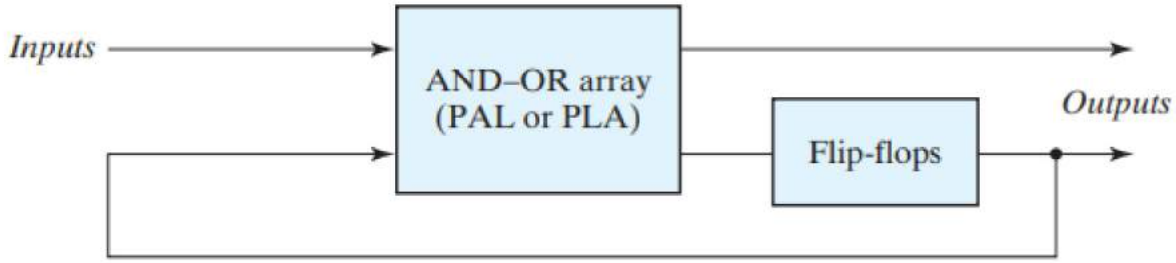


Fig. Sequential programmable logic device

A sequential programmable logic device (SPLD) refers to a type of digital electronic device that may be programmed to perform certain functions based on a sequence of inputs and outputs.

A Programmable Array Logic (PAL) or Programmable Logic Array (PLA) can be enhanced by using a series of flip-flops that are interconnected to create a register. The circuit's outputs may be obtained either from the OR gates or from the flip-flops' outputs. There exists the possibility of incorporating the outputs of flip-flops into the product terms that are constructed using the AND array, through the utilization of supplementary programmable connections. The flip-flops have the potential to be classified as either the D type or the JK type.

The initial programmable device designed to facilitate the creation of sequential circuits is known as the field-programmable logic sequencer (FPLS). A conventional Forward Propagation Logic System (FPLS) is structured based on a Programmable Logic Array (PLA) that operates several outputs to control flip-flops. The flip-flops possess a notable degree of flexibility as they may be configured to function as either the JK or the D kind. I'm sorry, but I cannot provide a response without any text from you. Please provide A Programmable Array Logic (PAL) that incorporates flip-flops is commonly known as a registered PAL. This nomenclature is used to indicate that the device not only comprises an AND-OR array but also incorporates flip-flops. The individual components within a structured programmable logic device (SPLD) are referred to as macrocells. These macrocells consist of a combinational logic function implemented using the sum-of-products method, along with an optional flip-flop element.

Utilizing a complex programmable logic device (CPLD), which comprises multiple individual programmable logic devices (PLDs) combined into a single circuit, presents a more cost-effective solution. A programmable interconnection structure facilitates the connectivity of programmable logic devices (PLDs) in a manner that is analogous to the interconnection of individual PLDs. The

image presented below illustrates the overall arrangement of a Complex Programmable Logic Device (CPLD).

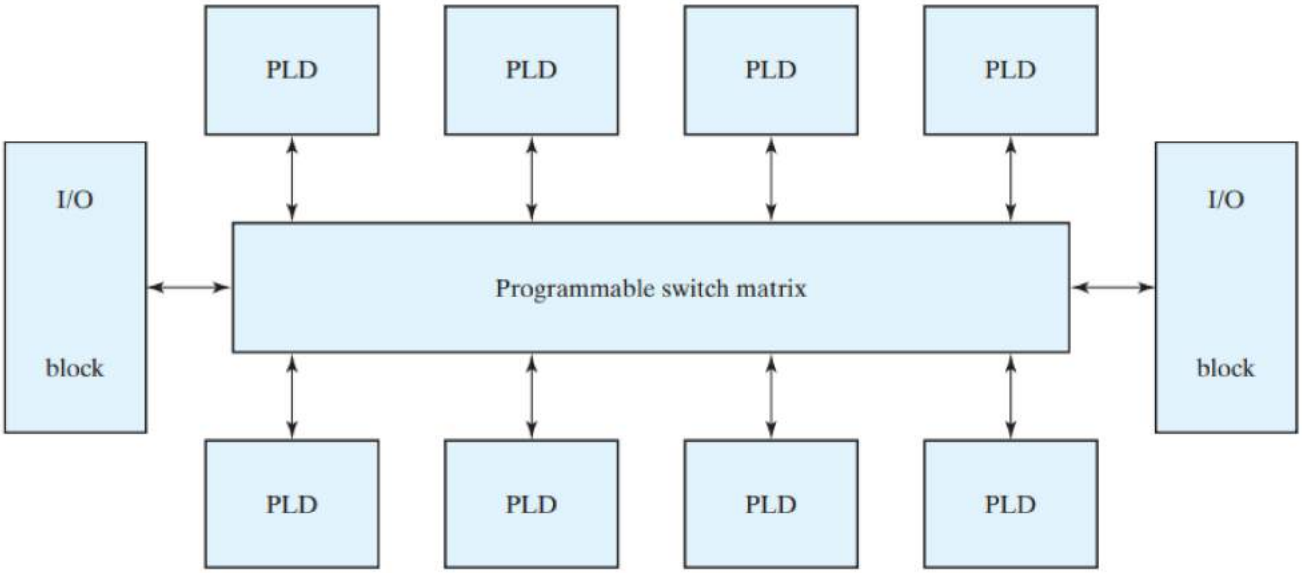


Fig. General CPLD configuration

The gadget is composed of several programmable logic devices (PLDs) that are interconnected by a switch matrix that may be programmed. The input-output (I/O) blocks facilitate the establishment of connections to the integrated circuit (IC) pins. Every input/output (I/O) pin is controlled by a three-state buffer and can be configured to function as either an input or an output. The switch matrix is responsible for receiving inputs from the input/output (I/O) block and subsequently routing them to the respective macrocells. In a similar manner, the outputs that are chosen from macrocells are transmitted to the outputs as required. normally, a Programmable Logic Device (PLD) comprises a range of 8 to 16 macrocells, which are normally interconnected in a completely integrated manner. In the event that a macrocell possesses unutilized product terms, it is possible to allocate them for utilization by neighboring macrocells. In certain instances, the macrocell flip-flop can be configured to function as a D, JK, or T flip-flop.

The fundamental element employed in very large-scale integration (VLSI) design is the gate array. It comprises a recurring arrangement of gates, constructed within a silicon substrate, and replicated numerous times until the entire chip surface is populated with gates. A field-programmable gate array (FPGA) refers to a type of very large-scale integration (VLSI) circuit that possesses the capability to be programmed by the user directly at their designated spot. A standard field-programmable gate array (FPGA) is comprised of a vast array of several logic blocks, encompassed by programmable input and output blocks, and interconnected through programmable interconnections. This set of gadgets exhibits a diverse range of internal arrangements. The operational efficiency of each device

variant is contingent upon the configuration of its logic blocks and the effectiveness of its programmed interconnections. A standard field-programmable gate array (FPGA) logic block is comprised of lookup tables, multiplexers, gates, and flip-flops.