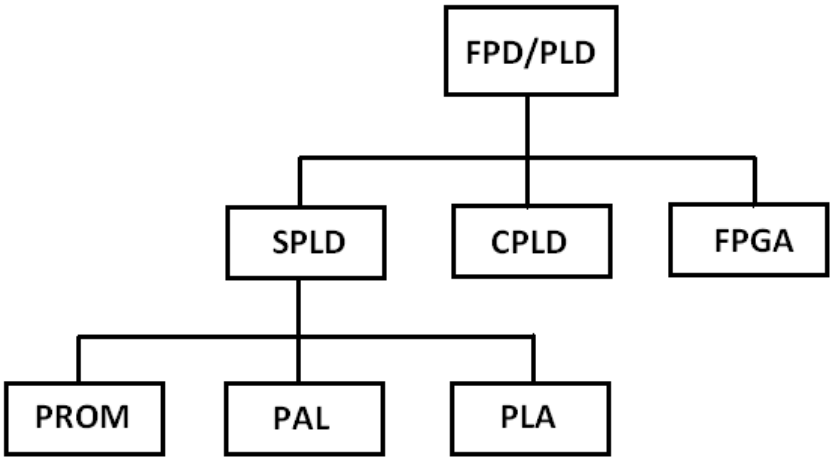
Programmable Logic Devices (PLDs)

A Programmable Logic Device (PLD) is a general purpose integrated circuit with internal logic gates that are connected through electronic fuses. Programming of the device involves burning of fuses along the paths that must be disconnected and a particular configuration is obtained for implementing a logic circuit. At the initial state of a PLD, all the fuses are intact.

Classification of Field Programmable Devices FPDs/PLDs



Classification of PLDs

The word programming refers to a hardware procedure that specifies the internal configuration of the device. There are two logic arrays in PLDs - AND array and OR array.

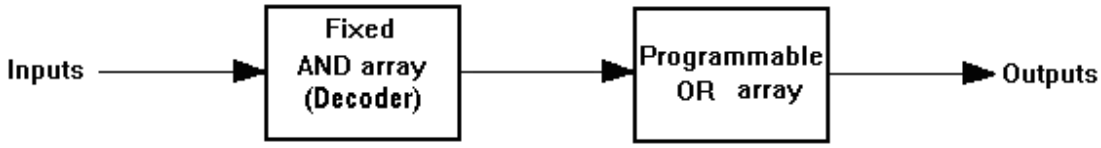
Simple Programmable Logic Device (SPLD)

Depending on the programmability of the arrays, SPLDs are classified into three types. They are:

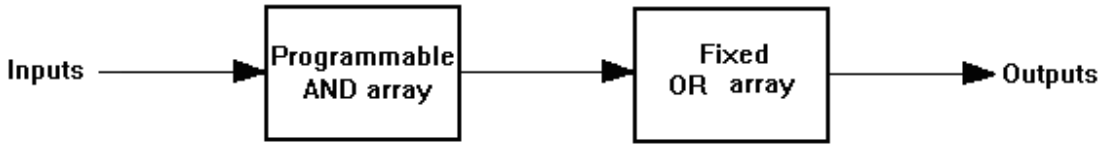
(i) PROM (Programmable Read Only Memory)

(ii) PAL (Programmable Array Logic or Programmable AND Logic)

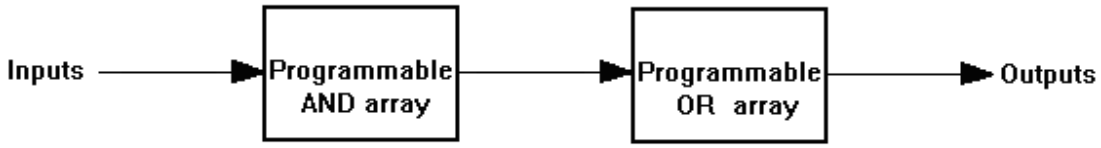
(iii) PLA (Programmable Logic Array)



(i) Programmable Read Only Memory



(ii) Programmable Array Logic



(iii) Programmable Logic Array

(i) **PROM**

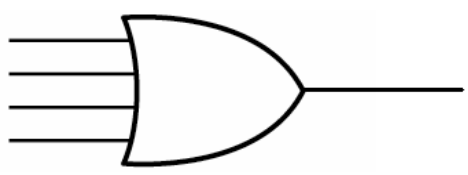
The PROM has a fixed AND array (constructed as a decoder) and programmable OR array. The PROM implements Boolean functions in canonical sum-of-product (SOP) form.

(ii) **PAL**

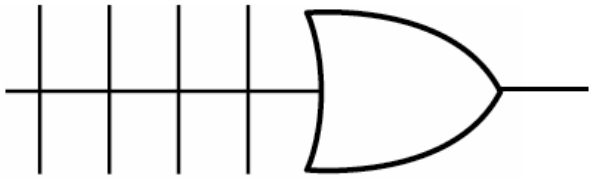
The PAL device has a programmable AND array and fixed connections for the OR array. Being the AND array programmable, PAL is also known as Programmable AND Logic. Unlike PROM, a PAL can realize no-canonical SOP form of expression.

(iii) **PLA**

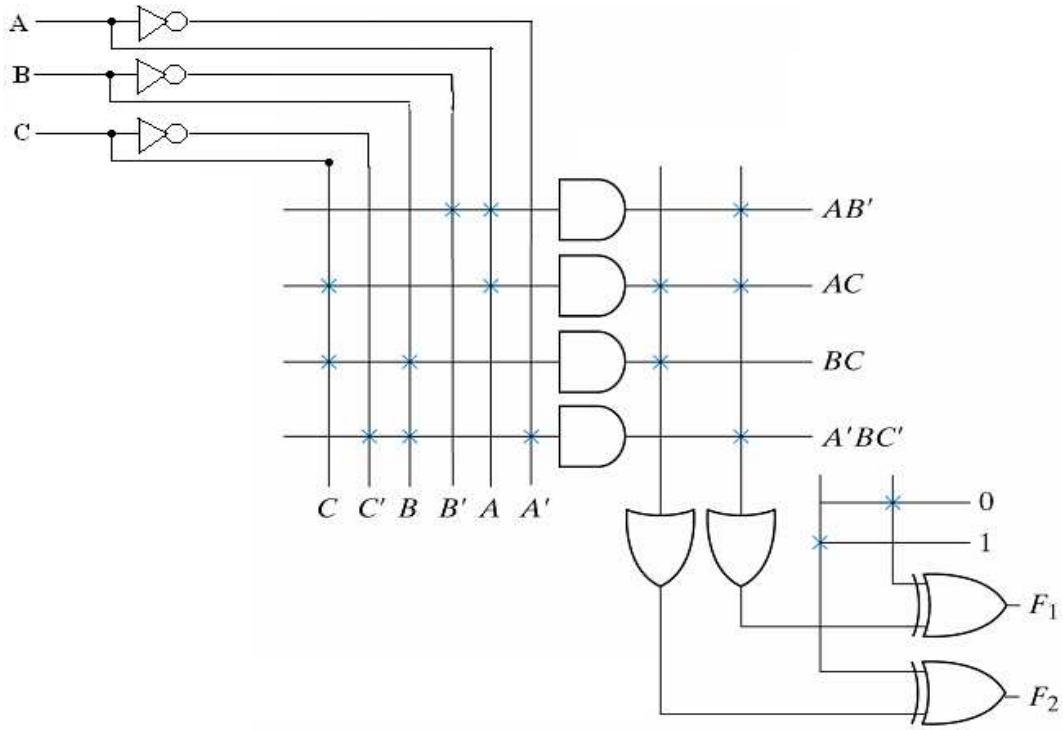
In PLA both AND and OR arrays are programmable. So it is the most flexible type of PLD.



Conventional symbol of OR gate



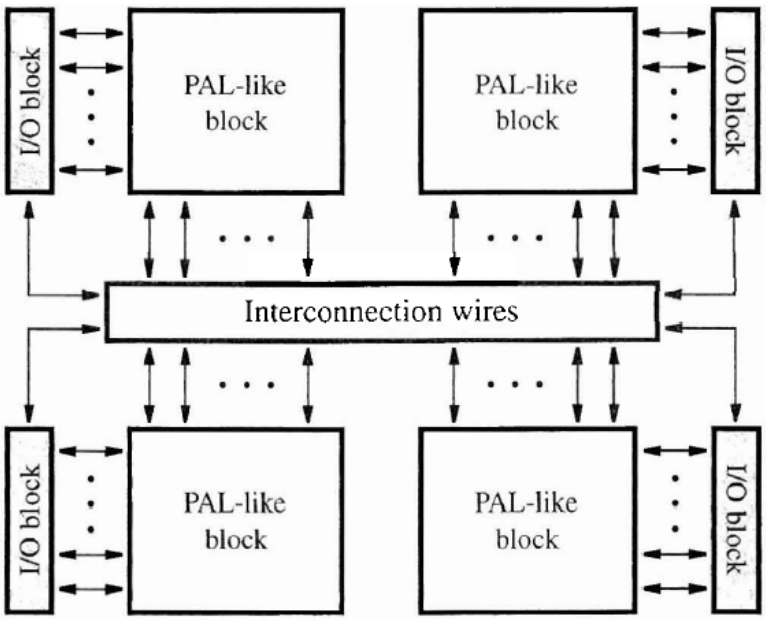
Array Logic Symbol of OR gate



A Programmed PLA with 3 inputs, 4 Product terms and 2 outputs

Complex Programmable Logic Device (CPLD)

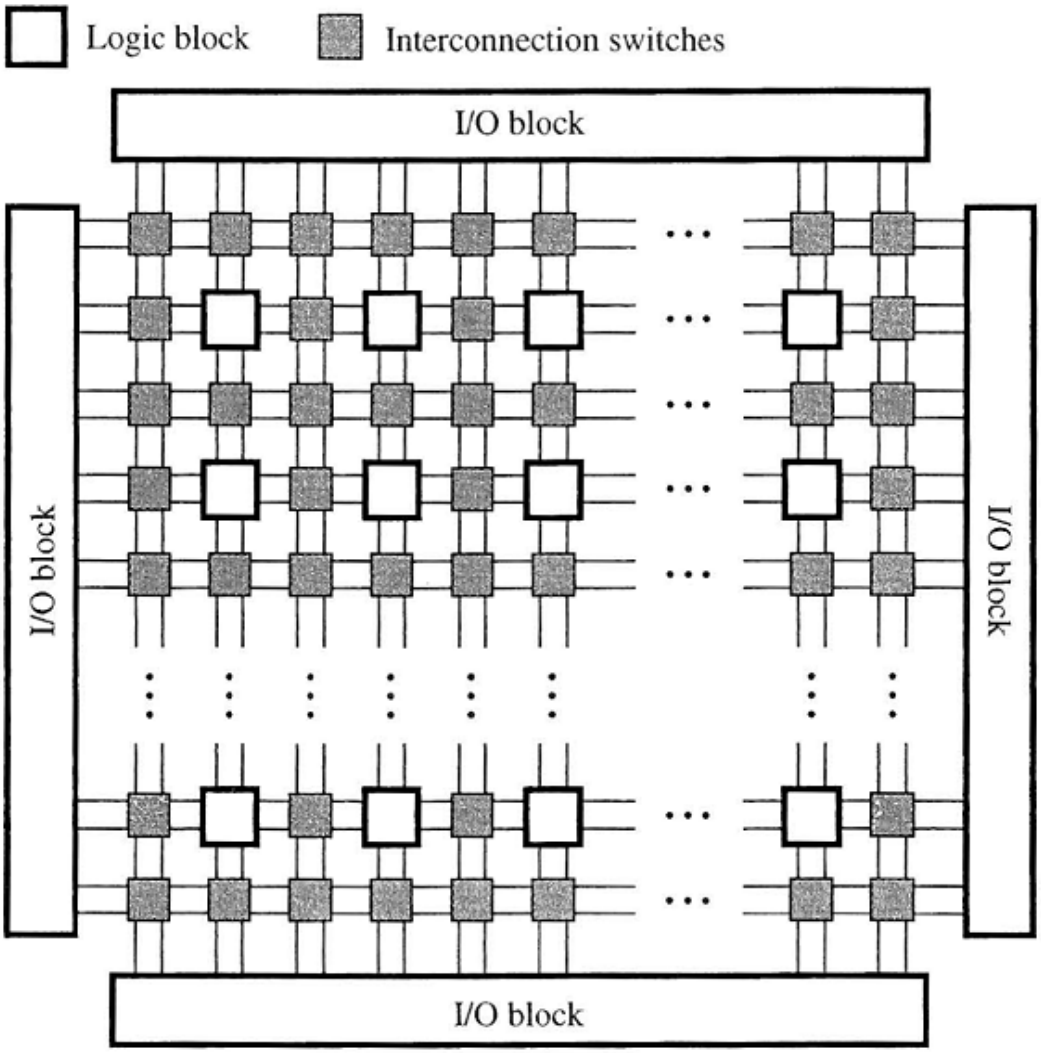
A CPLD has a few PLD blocks or macrocells on a single device or chip. Each of these blocks is similar to PAL or PLA. These blocks are interconnected with some programmable connections in between. Each PAL-like block is also connected to I/O block. Thus a CPLD has two levels of programmability: (i) each PLD block can be programmed and (ii) the interconnections between the PLDs can also be programmed.



Structure of a CPLD

Field Programmable Gate Array (FPGA)

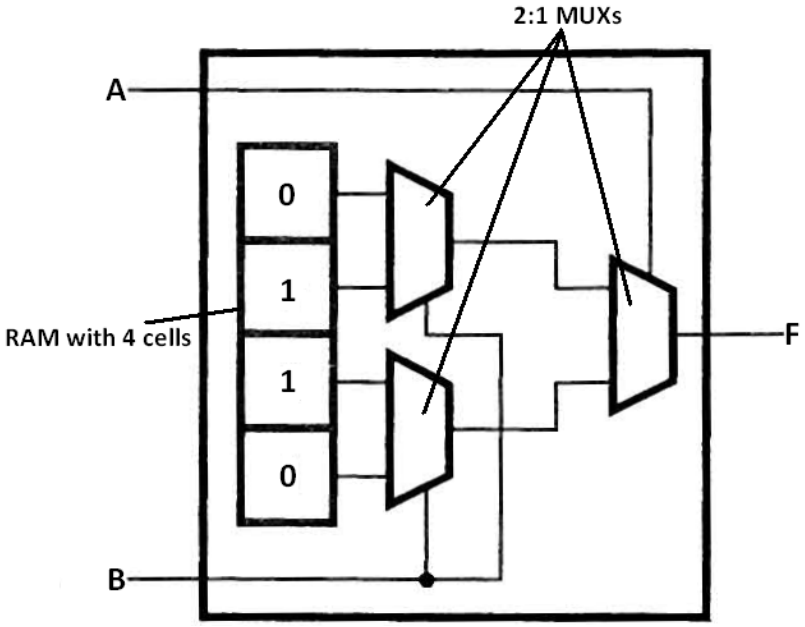
FPGA is an array of configurable logic blocks (CLBs) surrounded by programmable input and output blocks. These CLBs are connected together through programmable interconnections. A CLB consists of look-up tables (LUT), multiplexers, gates, and flip-flops. Each CLB is individually programmed to perform a logic function (such as AND, OR, XOR) and then the switches are programmed to connect the blocks so that the complete logic functions are implemented.



Structure of an FPGA

Look-up table is constructed with RAM and multiplexer. In the RAM bits are stored which are connected as inputs of the multiplexer. To realize any function, the required output bit pattern is stored in the RAM.

In the following figure, an LUT with 4 cells is shown. It has 3 no of 2:1 multiplexers (MUXs) combined together to work as a 4:1 MUX. A & B are the select inputs of the 4:1 MUX. The stored data in the RAM are 0,1,1,0 and this sequence is the output sequence of a 2 input XOR gate. So, the LUT is realizing the function of an XOR gate. This is how storing different data in the RAM, different logic functions can be realized.



An LUT with 4 cells

i/p		o/p
A	**B**	**F**
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of the function F which is being implemented in the LUT

Comparison of CPLD and FPGA

CPLD	FPGA
1. CPLD is constructed with PLDs.	1. FPGA is constructed with logic blocks.
2. CPLD contains only a few logic blocks that reaches up to a few thousands.	2. FPGA contains up to 1,00,000 of tiny logic blocks.
3. CPLDs are made of larger blocks. So, CPLDs are 'coarse-grain' devices.	3. FPGAs are made up of tiny logic blocks. So, they are considered as 'fine-grain' devices.
4. CPLDs are better for simpler circuits.	4. FPGAs are great for more complex applications.
5. CPLD is EEPROM-based. Hence it is non-volatile.	5. FPGA is RAM-based. So, it is volatile.
6. Delays are much more predictable in CPLDs.	6. Prediction of delay is difficult in FPGA.
7. CPLDs are much cheaper.	7. FPGAs are expensive.
8. CPLD is suited for control circuit because they have more combinational circuit.	8. Suited for timing circuit because they have more registers.
9. CPLD can operate at relatively low speed.	9. FPGA can operate at very high speed.
10. The CPLD could work immediately after power up.	10. FPGA could not work until the configuration is done.
11. CPLD has less flexibility and design capacity.	11. FPGA has more flexibility as well as design capacity.

Implementation procedure of a logic design with the FPGA or CPLD:

Following are the steps for implementing a logic circuit with an FPGA or a CPLD:

Step 1:
The description of the logic circuit is entered using a hardware description language (HDL) such as VHDL or Verilog. This can also be done by drawing the design using a schematic editor.

Step 2:
Using a logic synthesizer program, the HDL or schematic is transformed into a netlist. The netlist is a description of the various logic gates in the design and their interconnection behavior.

Step 3:
The netlist is fitted to the actual FPGA architecture using a process called place-and-route. Implementation tools are used to map the logic gates and interconnections into the FPGA. The configurable logic blocks in the FPGA can be further decomposed into look-up tables that perform logic operations. The configurable Logic Blocks (CLBs) or Look up tables (LUTs) are closely linked with various routing resources.

Step 4:
Once the implementation phase is complete, a program extracts the state of the switches in the routing matrices and a bit stream is generated where the ones and zeroes correspond to open or closed switches.

Step 5:
The bit stream is downloaded into a physical FPGA chip (usually embedded in some larger system). The electronic switches in the FPGA are opened or closed in response to the binary bits in the bit stream. Upon completion of the downloading, the FPGA will perform the operations specified by HDL code or schematic. Input signals can be applied to the I/O pins of the FPGA to check the operation of the design.