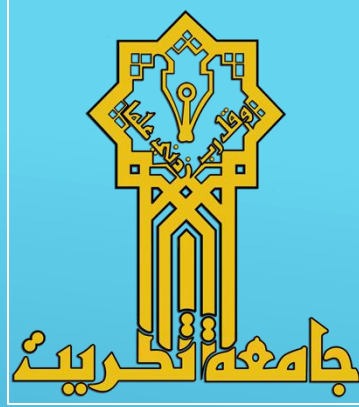




DIGITAL DEVICES AND LOGIC FAMILIES

ELECTRICAL DEPARTMENT

FOURTH STAGE

INSTRUCTOR: DR. ANAS MQDAD

LECTURE TWO

PROGRAMMABLE LOGIC

OUTLINES

- ❖ LUT CPLD Architecture
- ❖ PLA (Programmable Logic Array)
- ❖ Macrocell Modes
- ❖ Field-Programmable Gate Arrays (FPGAs)
- ❖ Logic Modules
- ❖ Operating Modes of a Logic Module

LUT CPLD Architecture

- ❑ This architecture differs from the classic CPLD previously discussed. As shown by the block diagram in Figure 1, this device contains logic array blocks (LABs) each with multiple **logic elements** (LEs). An LE is the basic logic design unit and is analogous to the macrocell.
- ❑ The programmable interconnects are arranged in a row and column arrangement running between the LABs, and input/output elements (IOEs) are oriented around the perimeter.
- ❑ A main difference between this type of CPLD and the classic AND/OR array CPLD previously discussed is the way in which a logic function is developed. Look-up tables (LUTs) are used instead of AND/OR arrays.

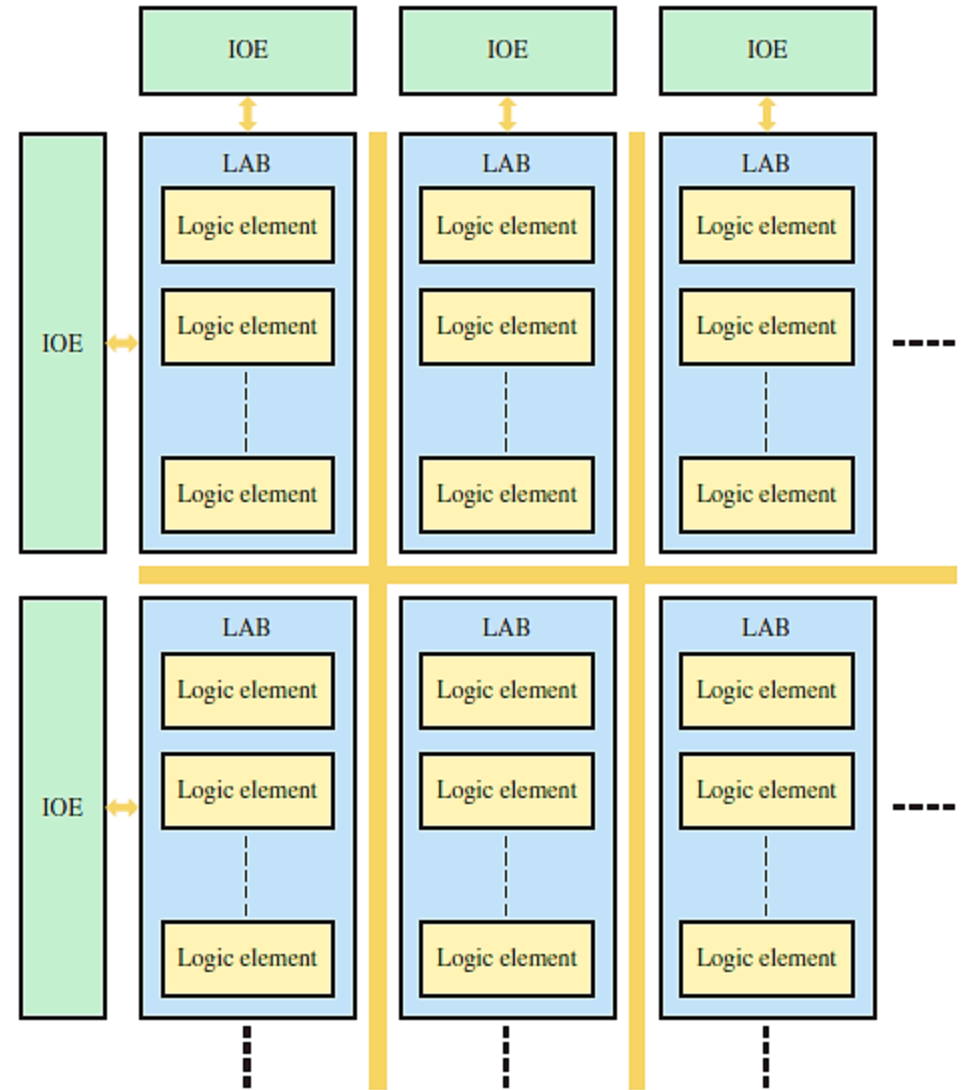


Figure 1 Simplified block diagram of an LUT CPLD.

- ❑ An **LUT** is basically a type of memory that can be programmed to produce SOP functions (discussed in more detail in next sections). These two approaches are contrasted in Figure 2.

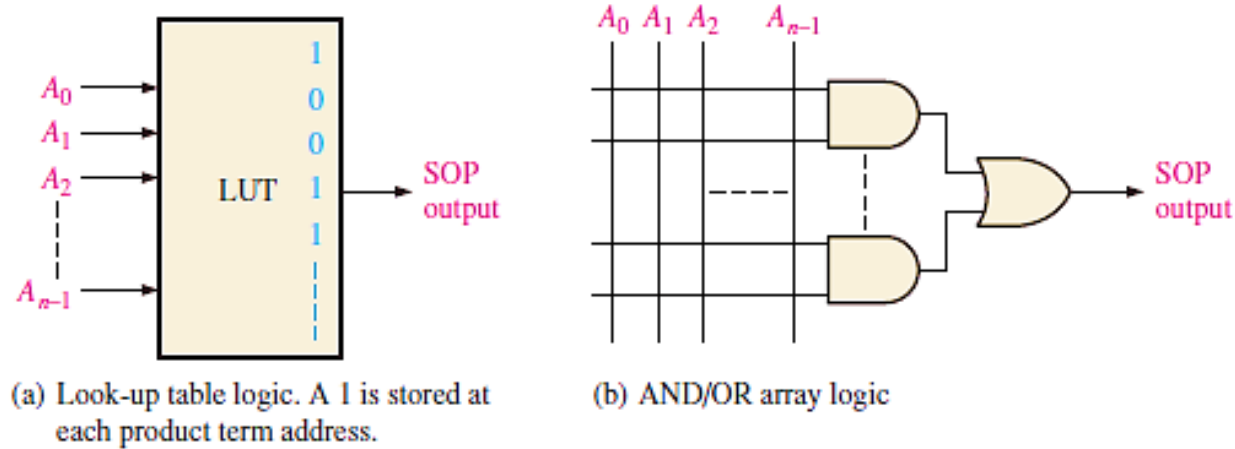


FIGURE 2 Two types of logic function generation in CPLDs.

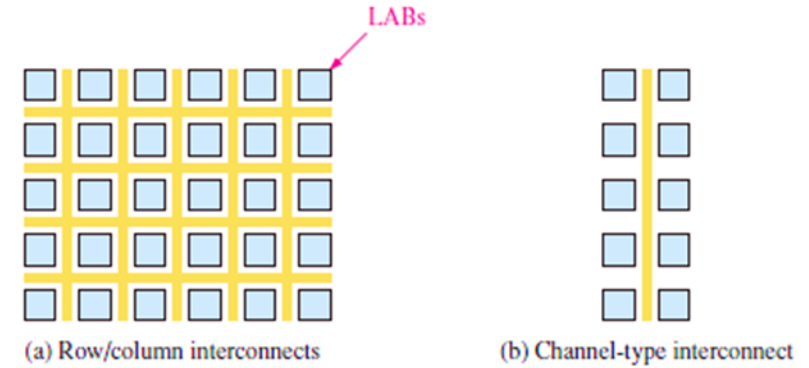


FIGURE 3 LUT CPLDs have row/column interconnects. Classic CPLDs have channel-type interconnects.

- ❑ As mentioned, the LUT CPLD has a row/column arrangement of interconnects instead of the channel-type interconnects found in most classic CPLDs. These two approaches are contrasted in Figure 3
- ❑ Most CPLDs use a nonvolatile process technology for the programmable links. The LUT CPLD, however, uses a SRAM-based process technology that is volatile—all programmed logic is lost when power is turned off. The memory embedded on the chip stores the program data using nonvolatile memory technology and reconfigures the CPLD on power up.

PLA (Programmable Logic Array)

- As you have learned, the architecture of a CPLD is the way in which the internal elements are organized and arranged.
- The architecture of some PLDs is based on a PLA (programmable logic array) structure rather than on a PAL (programmable array logic) structure, which we have discussed.
- Figure 4 compares a simple PAL structure with a simple PLA structure. The PAL has a programmable AND array followed by a fixed OR array and produces an SOP expression, as shown by the example in Figure 4(a).
- The **PLA** has a programmable AND array followed by a programmable OR array, as shown by the example in Figure 4(b).

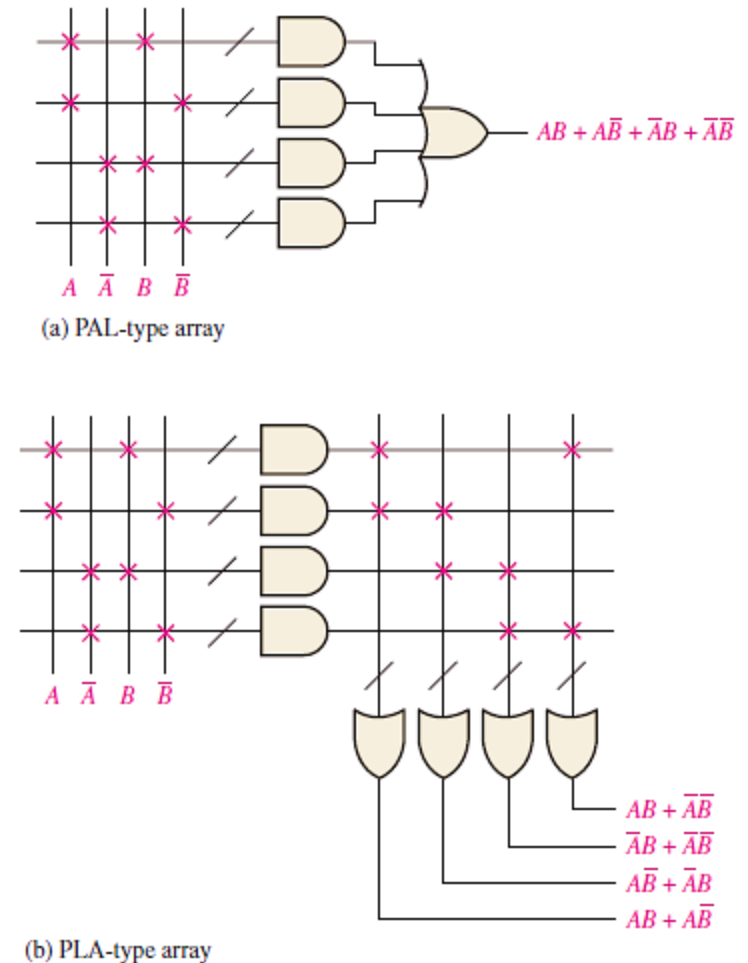


FIGURE 4 Comparison of a basic PLA to a basic PAL type PLD architecture.

Macrocell Modes

- CPLD macrocells were introduced previously. A macrocell can be configured for combinational logic or registered logic outputs and inputs by programming. The term *registered* refers to the use of flip-flops. In this section, you will learn about the typical macrocell, including the combinational and the registered modes of operation. Although macrocell architecture varies among different CPLDs, a typical macrocell architecture is used for illustration.
- Logic diagrams often use the symbol shown in Figure 5 to represent a multiplexer. In this case, the multiplexer has two data inputs and a select input that provides for programmable selection; the select input is usually not shown on a logic diagram.
- Figure 6 shows a complete macrocell including the flip-flop (register). The XOR gate provides for complementing the SOP function from the OR gate to produce a function in POS form.

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

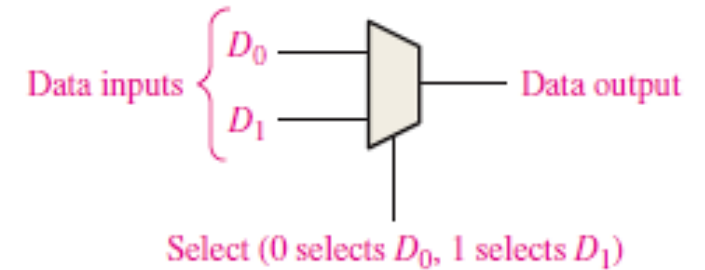


FIGURE 5 Commonly used symbol for a multiplexer. It can have any number of inputs.

- A 1 on the top input of the XOR gate complements the OR output, and a 0 lets the OR output pass uncomplemented (in SOP form).
- MUX 1 provides for selection of either the XOR output or an input from the I/O. MUX 2 can be programmed to select either the global clock or a clock signal based on a product term.
- MUX 3 can be programmed to select either a HIGH (V_{CC}) or a product-term enable for the flip-flop.

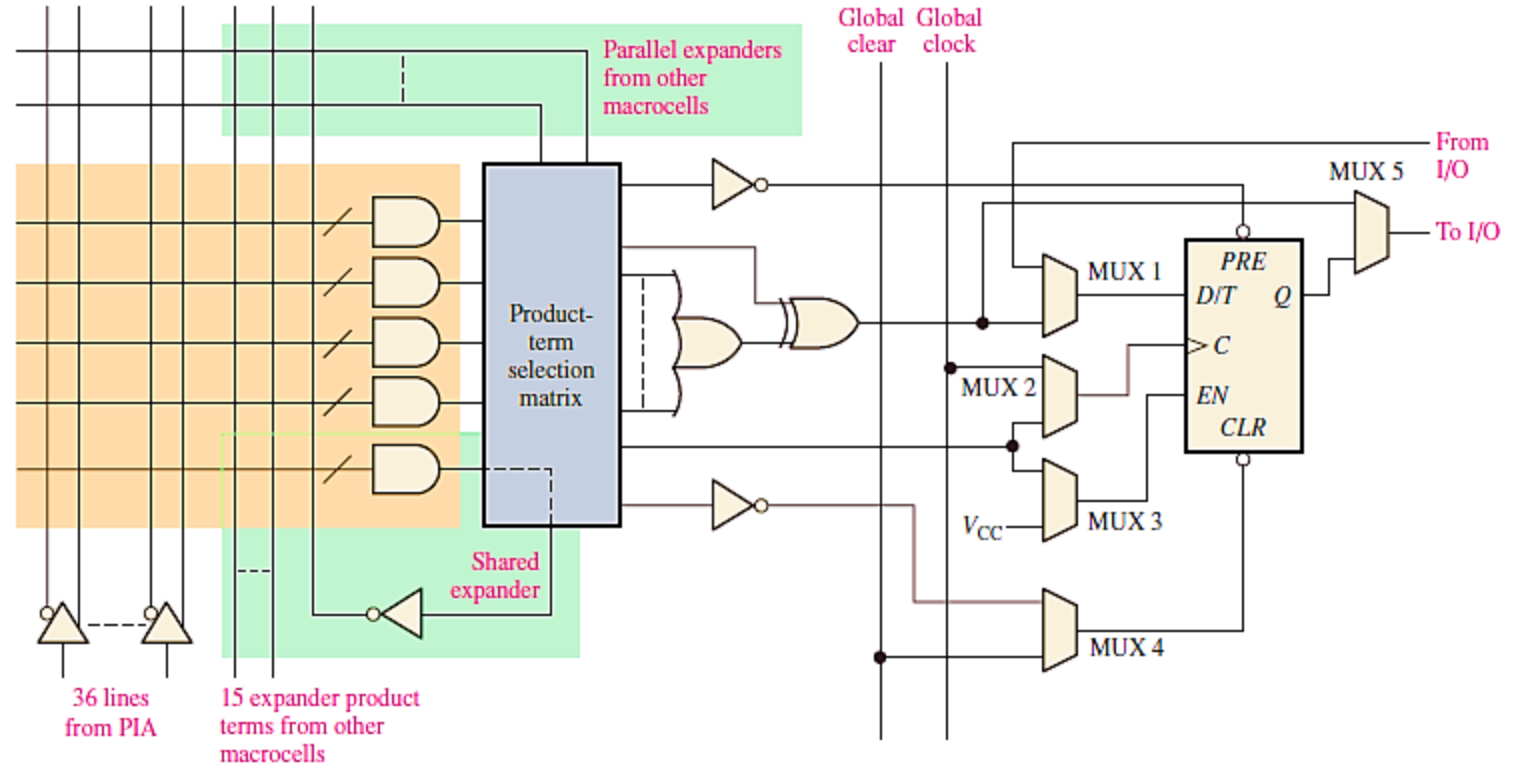


FIGURE 6 A CPLD macrocell.

- MUX 4 can select the global clear or a product-term clear. MUX 5 is used to bypass the flip-flop and connect the combinational logic output to the I/O or to connect the registered output to the I/O.
- The flip-flop can be programmed as a D, T (toggle), or J-K flip-flop.

The Combinational Mode

- When a macrocell is programmed to produce an SOP combinational logic function, the logic elements in the data path are as shown in red in Figure 7. As you can see, only one mux is used and the register (flip-flop) is bypassed.

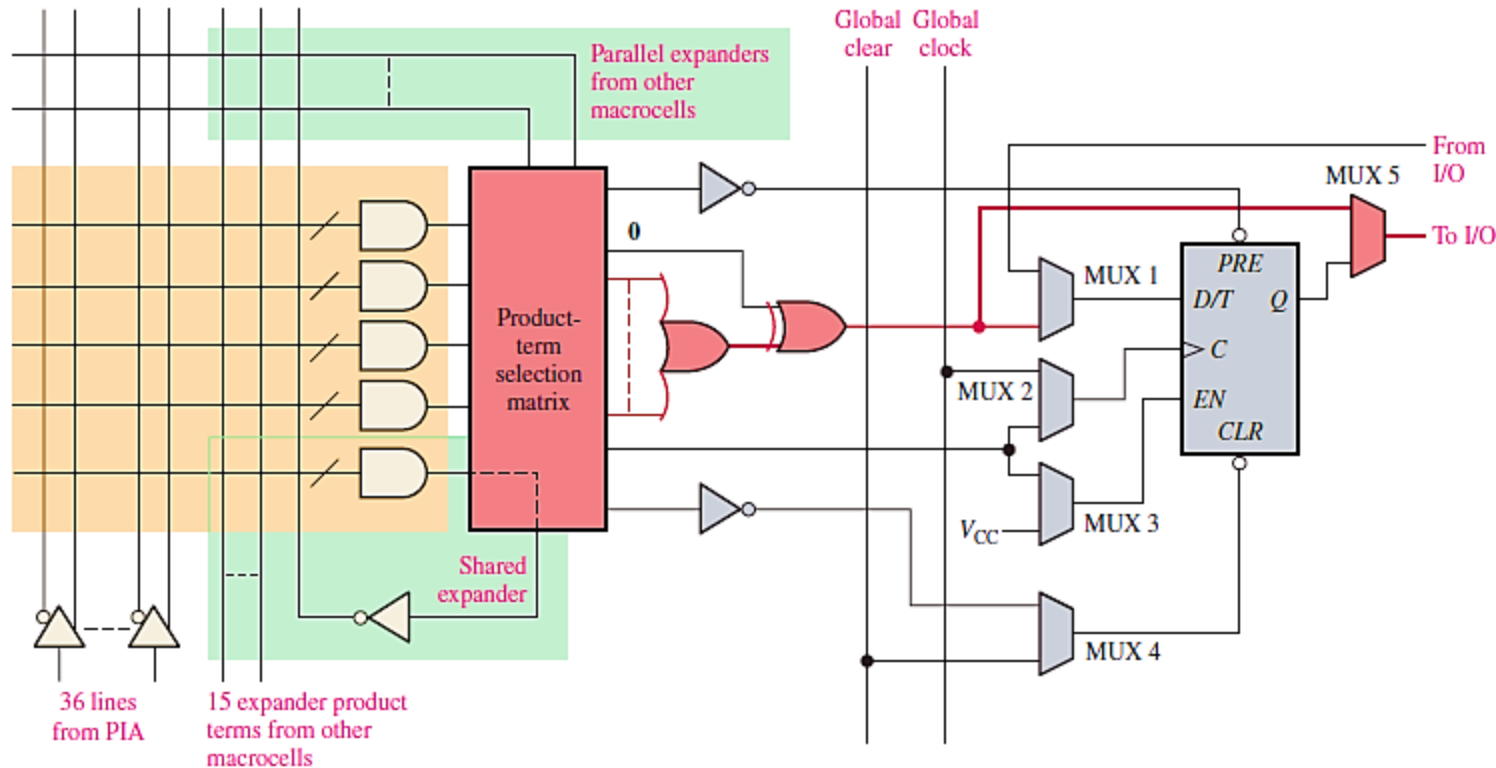


FIGURE 7 A macrocell configured for generation of an SOP logic function. Red indicates data path

The Registered Mode

When a macrocell is programmed for the registered mode with the SOP combinational logic output providing the data input to the register and clocked by the global clock, the elements in the data path are as shown in red in Figure 8. As you can see, four multiplexers (mux) are used and the register (flip-flop) is active.

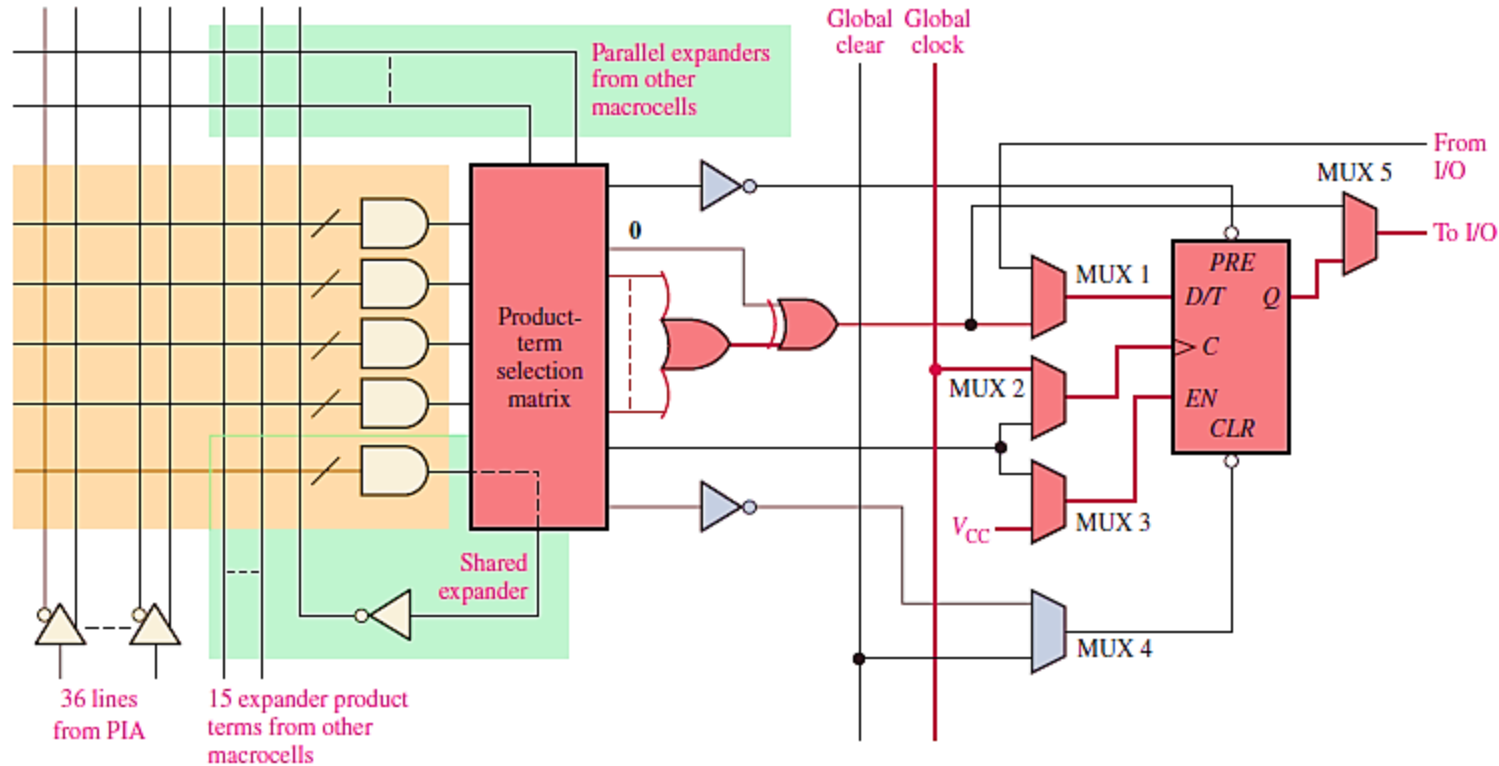


FIGURE 8 A macrocell configured for generation of a registered logic function. Red indicates data path.

Field-Programmable Gate Arrays (FPGAs)

- ❖ As you have learned, the classic CPLD architecture consists of PAL/GAL or PLA-type logic blocks with programmable interconnections.
 - ❖ Basically, the FPGA (field-programmable gate array) differs in architecture, does not use PAL/PLA type arrays, and has much greater densities than CPLDs. A typical FPGA has many times more equivalent gates than a typical CPLD.
 - ❖ The logic-producing elements in FPGAs are generally much smaller than in CPLDs, and there are many more of them. Also, the programmable interconnections are generally organized in a row and column arrangement in FPGAs.
-
- ❖ The three basic elements in an FPGA are the configurable logic block (CLB), the interconnections, and the input/output (I/O) blocks, as illustrated in Figure 9.
 - ❖ The configurable logic blocks (CLBs) in an FPGA are not as complex as the LABs or function block (FBs) in a CPLD, but generally there are many more of them. When the CLBs are relatively simple, the FPGA architecture is called fine grained.
 - ❖ When the CLBs are larger and more complex, the architecture is called coarse grained. The I/O blocks around the perimeter of the structure provide individually selectable input, output, or bidirectional access to the outside world.

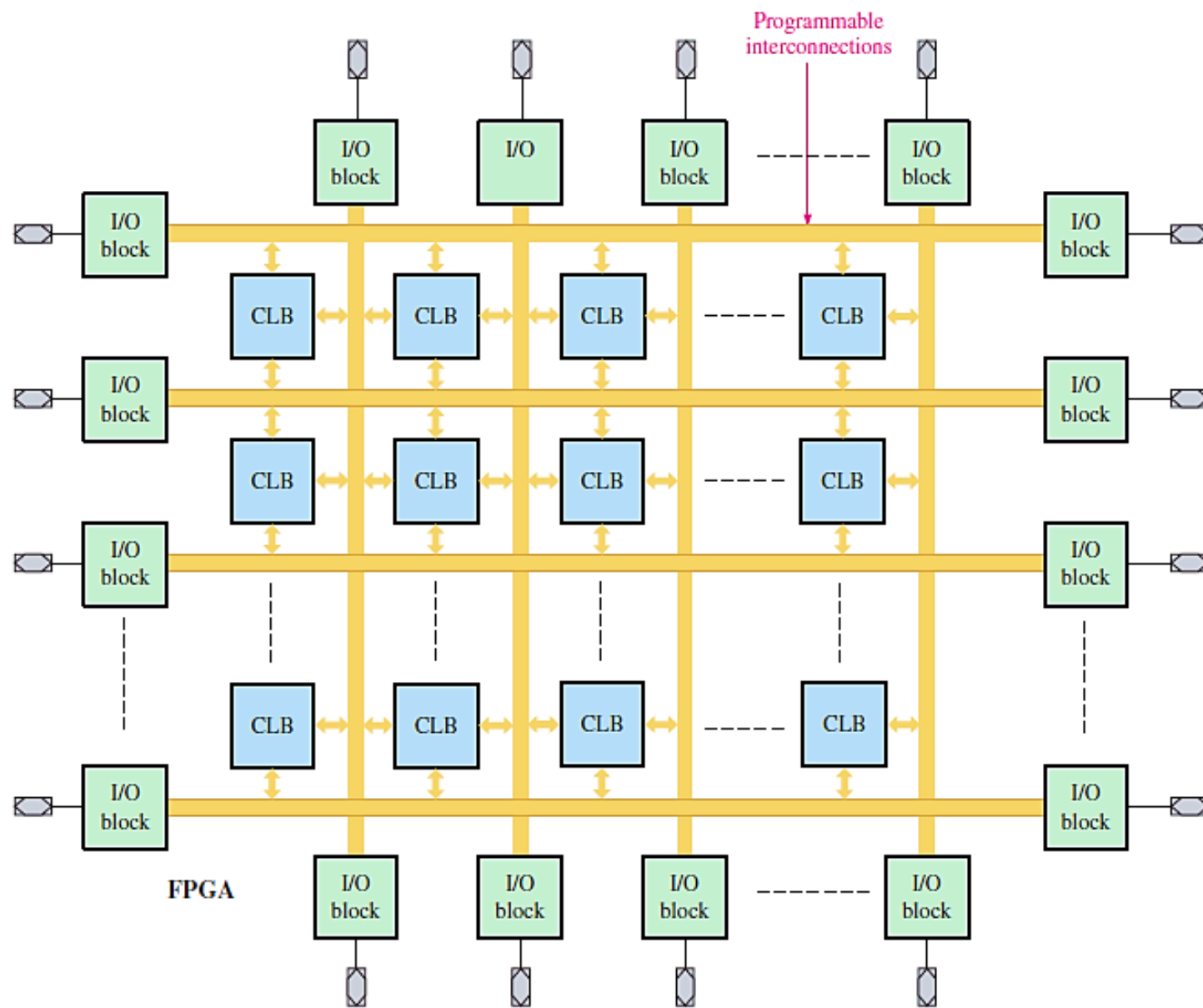


FIGURE 9 Basic structure of an FPGA. CLB is configurable logic block, also known as logic array block (LAB).

- ❖ The distributed matrix of programmable interconnections provide for interconnection of the CLBs and connection to inputs and outputs. Large FPGAs can have tens of thousands of CLBs in addition to memory and other resources.
- ❖ Most programmable logic manufacturers make a series of FPGAs that range in density, power consumption, supply voltage, speed, and to some degree vary in architecture. FPGAs are reprogrammable and use SRAM or antifuse process technology for the programmable links.
- ❖ Densities can range from hundreds of logic modules to hundreds of thousands of logic modules in packages with up to over 1,000 pins. DC supply voltages are typically 1.8 V to 5 V, depending on the specific device.

Configurable Logic Blocks

- ❖ Typically, an FPGA logic block consists of several smaller logic modules that are the basic building units, somewhat analogous to macrocells in a CPLD.
- ❖ Figure 10 shows the fundamental configurable logic blocks (CLBs) within the global row/column programmable interconnects that are used to connect logic blocks.
- ❖ Each CLB (also known as logic array block, LAB) is made up of multiple smaller logic modules and a local programmable interconnect that is used to connect logic modules within the CLB.

Logic Modules

- ❑ A logic module in an FPGA logic block can be configured for combinational logic, registered logic, or a combination of both. A flip-flop is part of the associated logic and is used for registered logic.
- ❑ A block diagram of a typical LUT-based logic module is shown in Figure 11. As you know, an LUT (look-up table) is a type of memory that is programmable and used to generate SOP combinational logic functions. The LUT essentially does the same job as the PAL or PLA does.

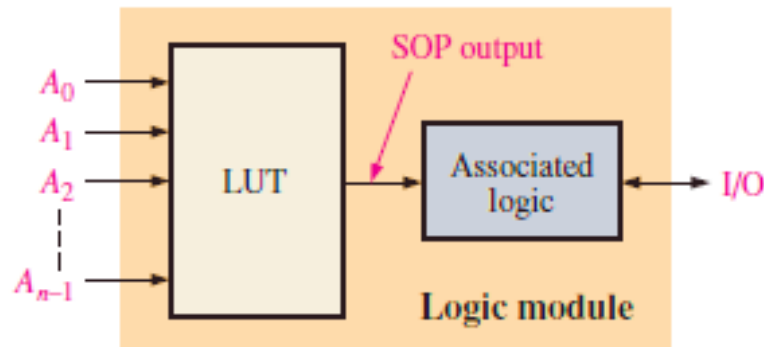


FIGURE 11 Basic block diagram of a logic module in an FPGA.

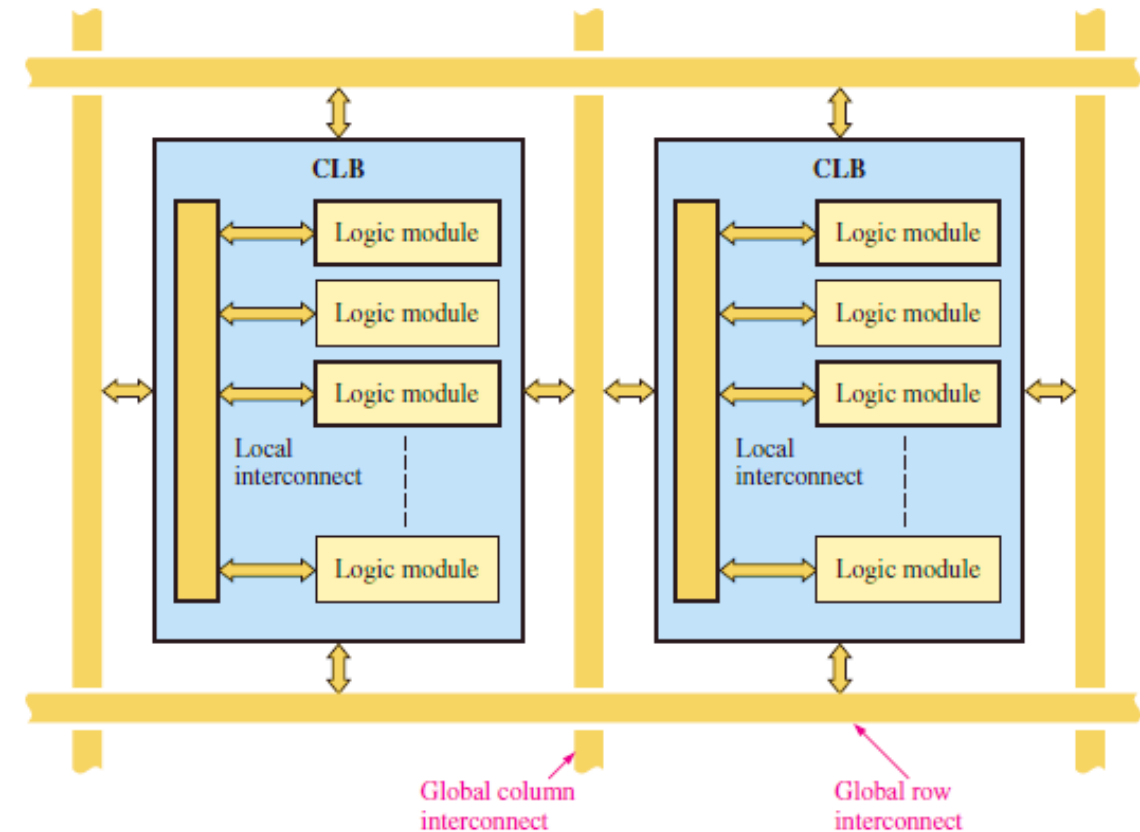


FIGURE 10 Basic configurable logic blocks (CLBs) within the global row/column programmable interconnects.

- Generally, the organization of an LUT consists of a number of memory cells equal to 2^n , where n is the number of input variables. For example, three inputs can select up to eight memory cells, so an LUT with three input variables can produce an SOP expression with up to eight product terms.
- A pattern of 1s and 0s can be programmed into the LUT memory cells, as illustrated in Figure 12 for a specified SOP function. Each 1 means the associated product term appears in the SOP output, and each 0 means that the associated product term does not appear in the SOP output. The resulting SOP output expression is

$$\bar{A}_2\bar{A}_1\bar{A}_0 + \bar{A}_2A_1A_0 + A_2\bar{A}_1A_0 + A_2A_1A_0$$

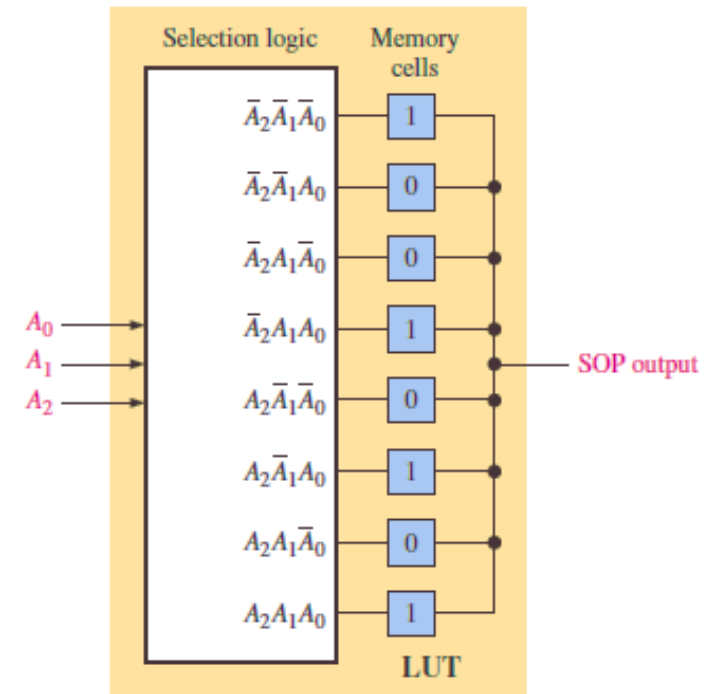


FIGURE 12 The basic concept of an LUT programmed for a particular SOP output.

EXAMPLE

Show a basic 3-variable LUT programmed to produce the following SOP function:

$$A_2A_1\bar{A}_0 + A_2\bar{A}_1\bar{A}_0 + \bar{A}_2A_1A_0 + A_2\bar{A}_1A_0 + \bar{A}_2\bar{A}_1A_0$$

Solution

A 1 is stored for each product term in the SOP expression, as shown in Figure 13.

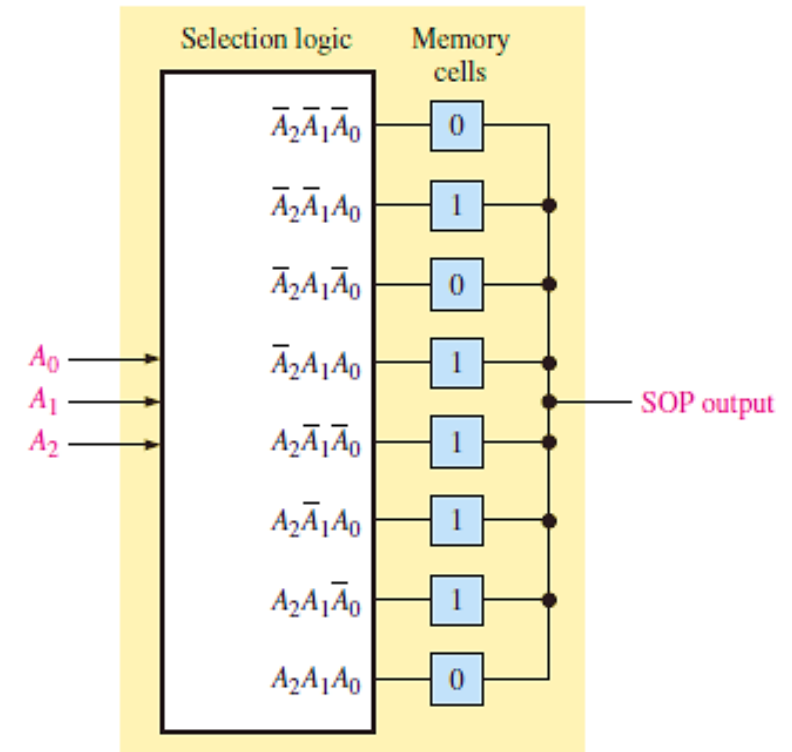


Figure 13

Operating Modes of a Logic Module

Typically, a logic module (LM) can be programmed for the following modes of operation:

- Normal mode
- Extended LUT mode
- Arithmetic mode
- Shared arithmetic mode

In addition to these four modes, a logic module can be utilized as a register chain to create counters and shift registers. In this section, we will discuss the normal mode and the extended LUT mode.

The *normal mode* is used primarily for generating combinational logic functions. A logic module can implement one or two combinational output functions with its two LUTs. Examples of four LUT configurations are illustrated in Figure 14. Generally, two SOP functions, each with four variables or less, can be implemented in an LM without sharing inputs. For example, you can have two 4-variable functions, one 4-variable function and one 3-variable function, or two 3-variable functions. By sharing inputs, you can have any combination of a total of eight inputs up to a maximum of six inputs for each LUT. In the normal mode, you are limited to 6-variable SOP functions.

- The *extended LUT mode* allows expansion to a 7-variable function, as illustrated in Figure 15. The multiplexer formed by the AND-OR circuit with a complemented input is part of the dedicated logic in a logic module.

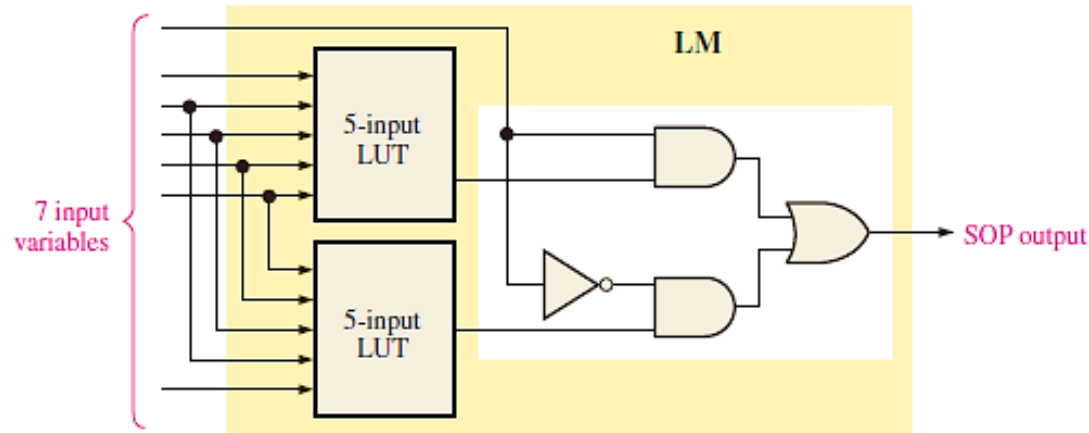


FIGURE 15 Expansion of a logic module (LM) to produce a 7-variable SOP function in the extended LUT mode.

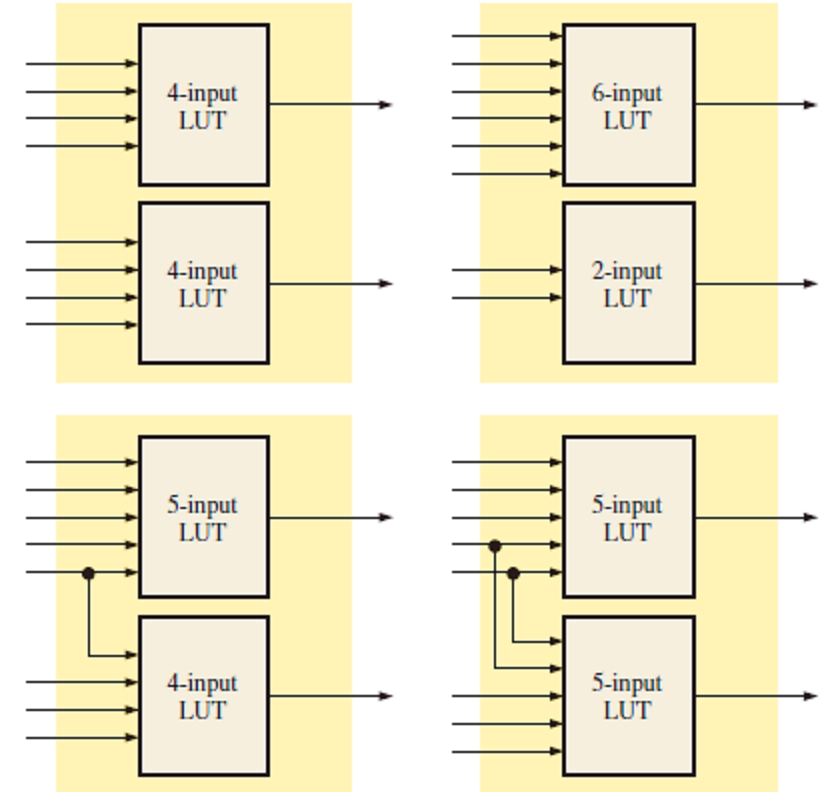


FIGURE 14 Examples of possible LUT configurations in a logic module (LM) in the normal mode.

EXAMPLE

A logic module is configured in the extended LUT mode, as shown in Figure 16. For the specific LUT outputs shown, determine the final SOP output.

Solution

The SOP output expression is as follows:

$$\bar{A}_5 A_4 A_3 A_2 A_1 A_0 + A_5 \bar{A}_4 A_3 A_2 A_1 A_0 + A_5 A_4 A_3 A_2 A_1 A_0 + A_6 A_5 A_4 A_3 \bar{A}_2 \bar{A}_0 + A_6 A_5 \bar{A}_4 A_3 A_2 \bar{A}_0 + A_6 A_5 A_4 A_3 A_2 \bar{A}_0$$

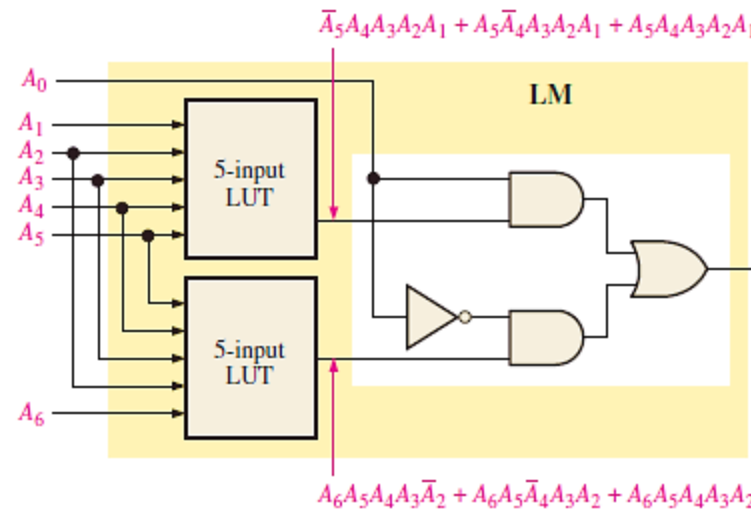


Figure 16

Thank You