Vietnam National University HCMC

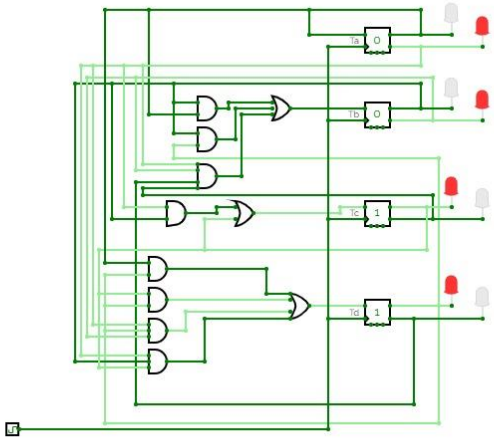
International University

School of Electrical Engineering

EE053IU

Digital Logic Design

Lecture 10: Programmable Logic



INSTRUCTOR: Dr. Vuong Quoc Bao

1. Simple Programmable Logic Devices (SPLDs)

- Two major types of simple programmable logic devices (SPLDs) are the PAL and the GAL.
- PAL stands for programmable array logic, and GAL stands for generic array logic.
- Generally, a PAL is one-time programmable (OTP), and a GAL is a type of PAL that is reprogrammable.
- The term GAL is a designation originally used by Lattice Semiconductor and later licensed to other manufacturers.
- The basic structure of both PALs and GALs is a programmable AND array and a fixed OR array, which is a basic sum-of-products architecture.

SPLD: The PAL

- A PAL (programmable array logic) consists of a programmable array of AND gates that connects to a fixed array of OR gates.
- Generally, PALs are implemented with fuse process technology and are, therefore, one-time programmable (OTP).
- The PAL structure allows any sum-of-products (SOP) logic expression with a defined number of variables to be implemented.

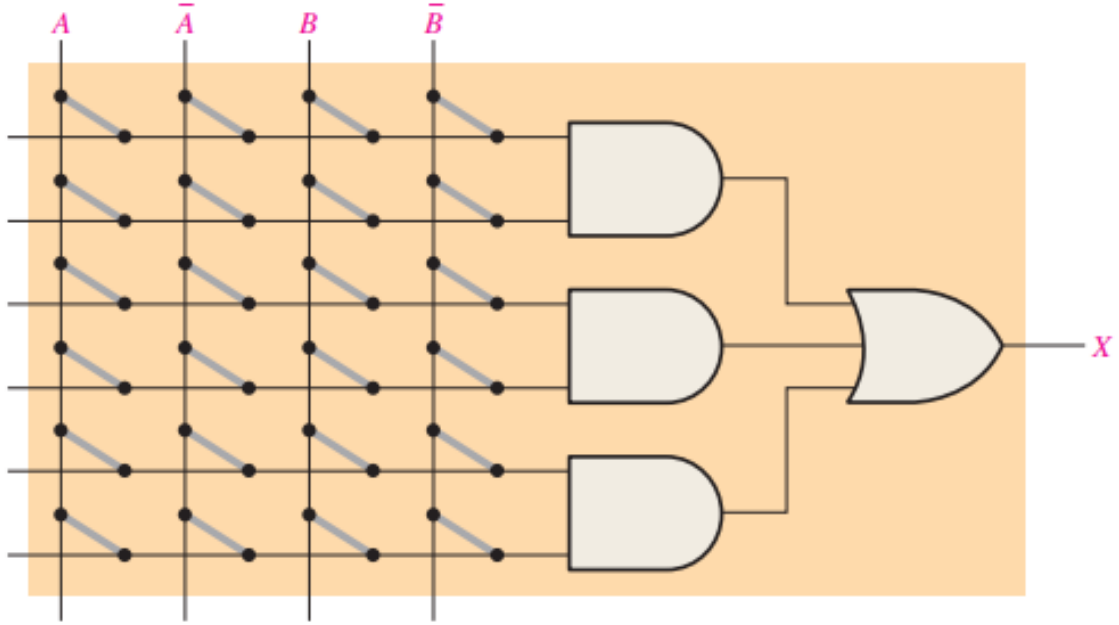


FIGURE 10–1 Basic AND/OR structure of a PAL.

Implementing a Sum-of-Products Expression

- An example of a simple PAL is programmed as shown in Figure 10–2 so that the product term $A\overline{B}$ is produced by the top AND gate, AB is produced by the middle AND gate, and $\overline{A}\overline{B}$ is produced by the bottom AND gate

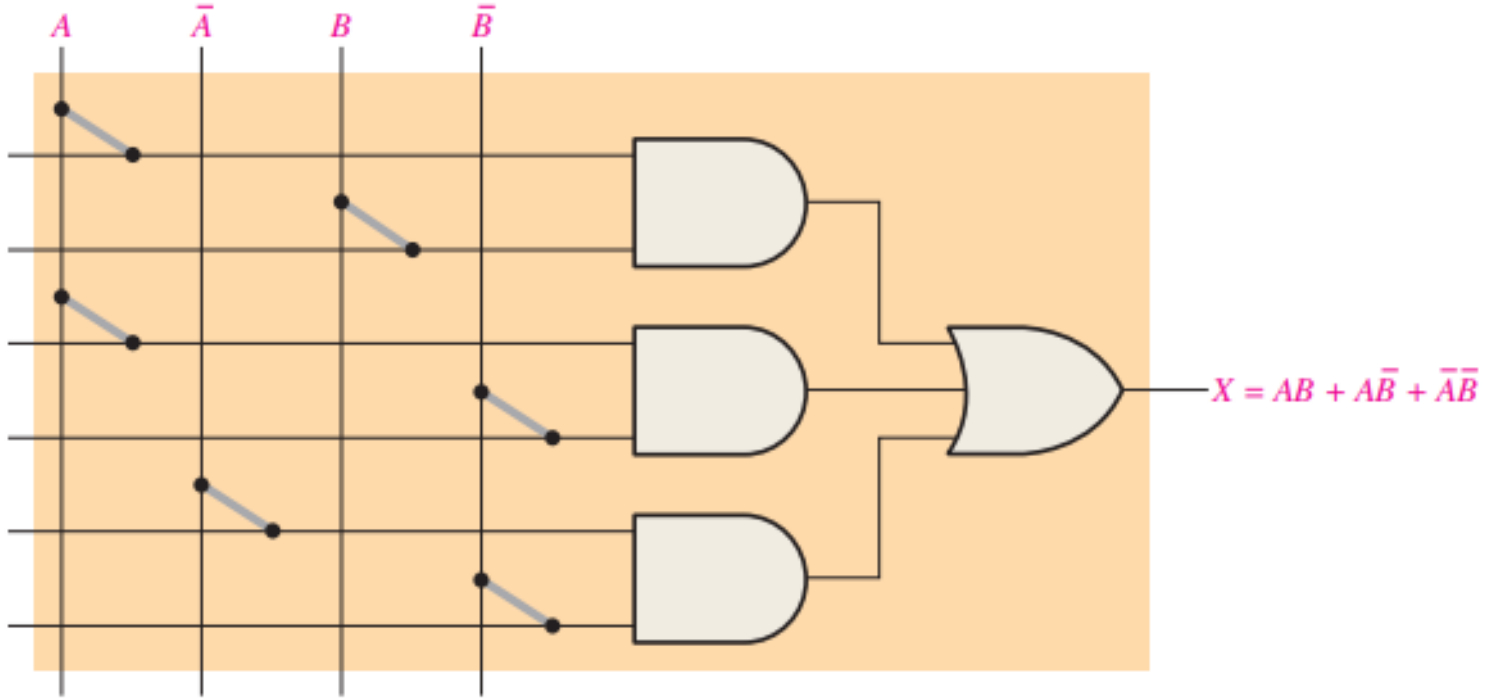


FIGURE 10–2 PAL implementation of a sum-of-products expression.

SPLD: The PAL

- The GAL is essentially a PAL that can be reprogrammed. It has the same type of AND/OR organization that the PAL does. The basic difference is that a GAL uses a reprogrammable process technology, such as EEPROM (E^2CMOS), instead of fuses.

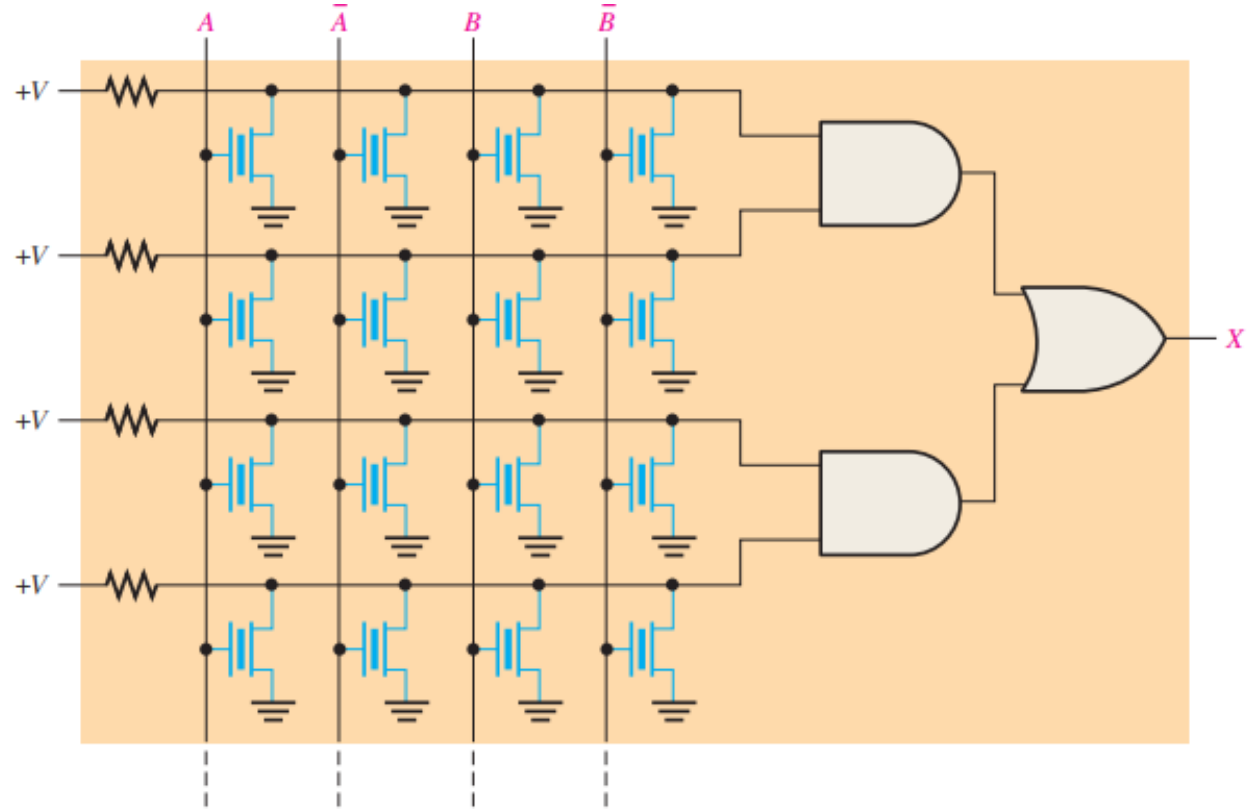


FIGURE 10–3 Simplified GAL array.

Simplified Notation for PAL/GAL Diagrams

- Actual PAL and GAL devices have many AND and OR gates in addition to other elements and are capable of handling many variables and their complements. Most PAL and GAL diagrams that you may see on a data sheet use simplified notation.

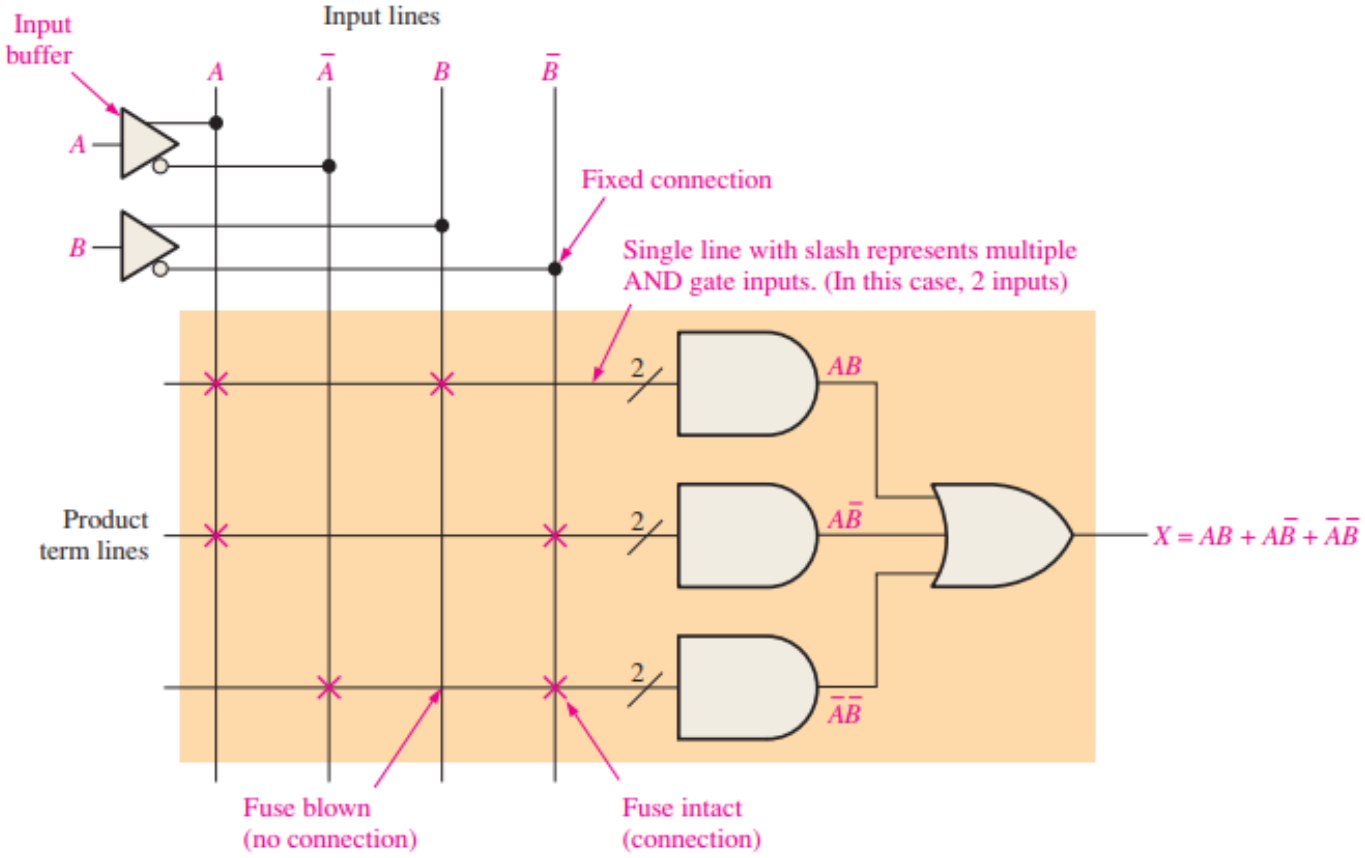


FIGURE 10–4 A portion of a programmed PAL/GAL.

EXAMPLE 10–1

Show how a PAL is programmed for the following 3-variable logic function:

$$X = A\overline{B}C + \overline{A}B\overline{C} + \overline{A}\,\overline{B} + AC$$

Solution

The programmed array is shown in Figure 10–5. The intact fusible links are indicated by small red Xs. The absence of an X means that the fuse is open.

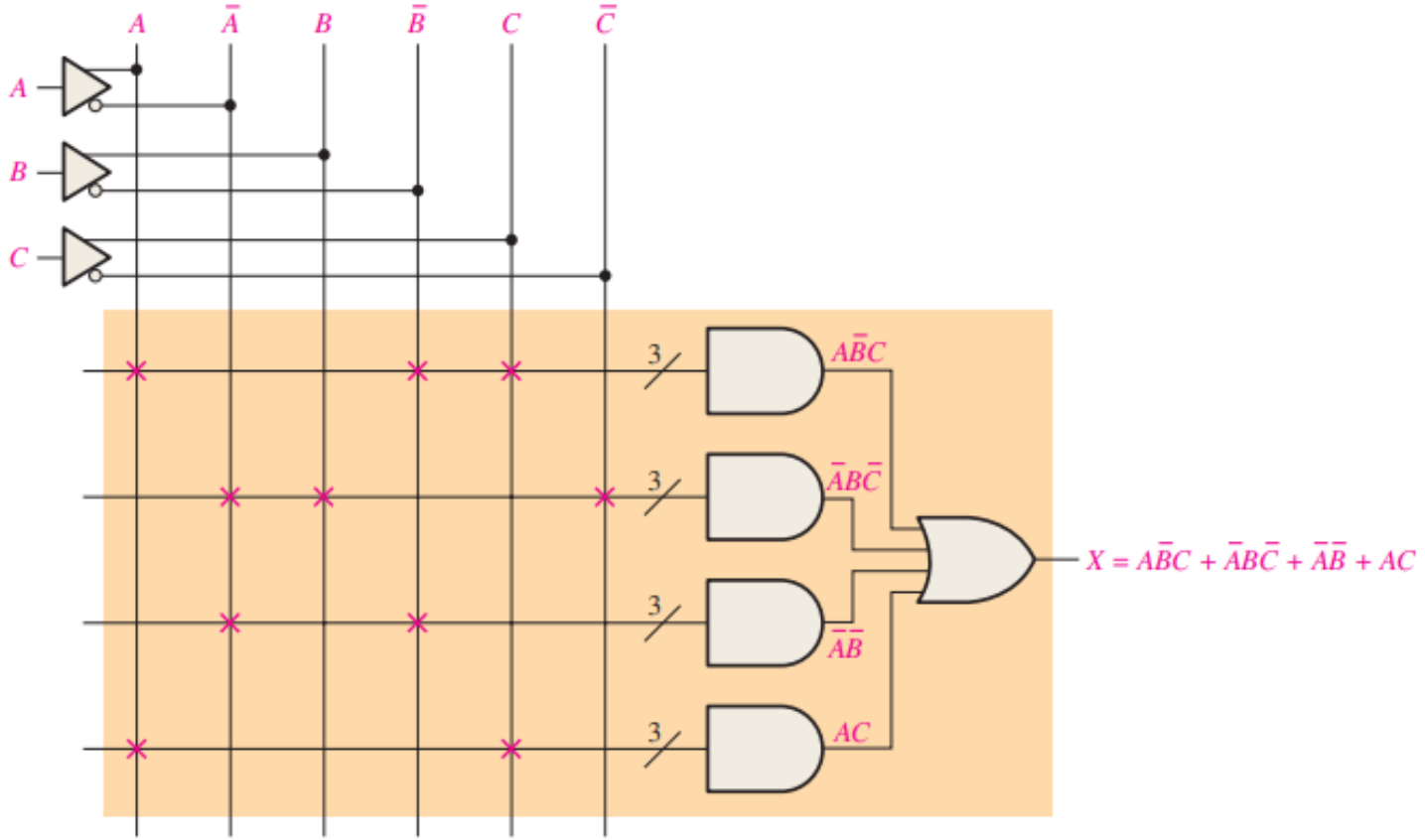


FIGURE 10–5

PAL/GAL General Block Diagram

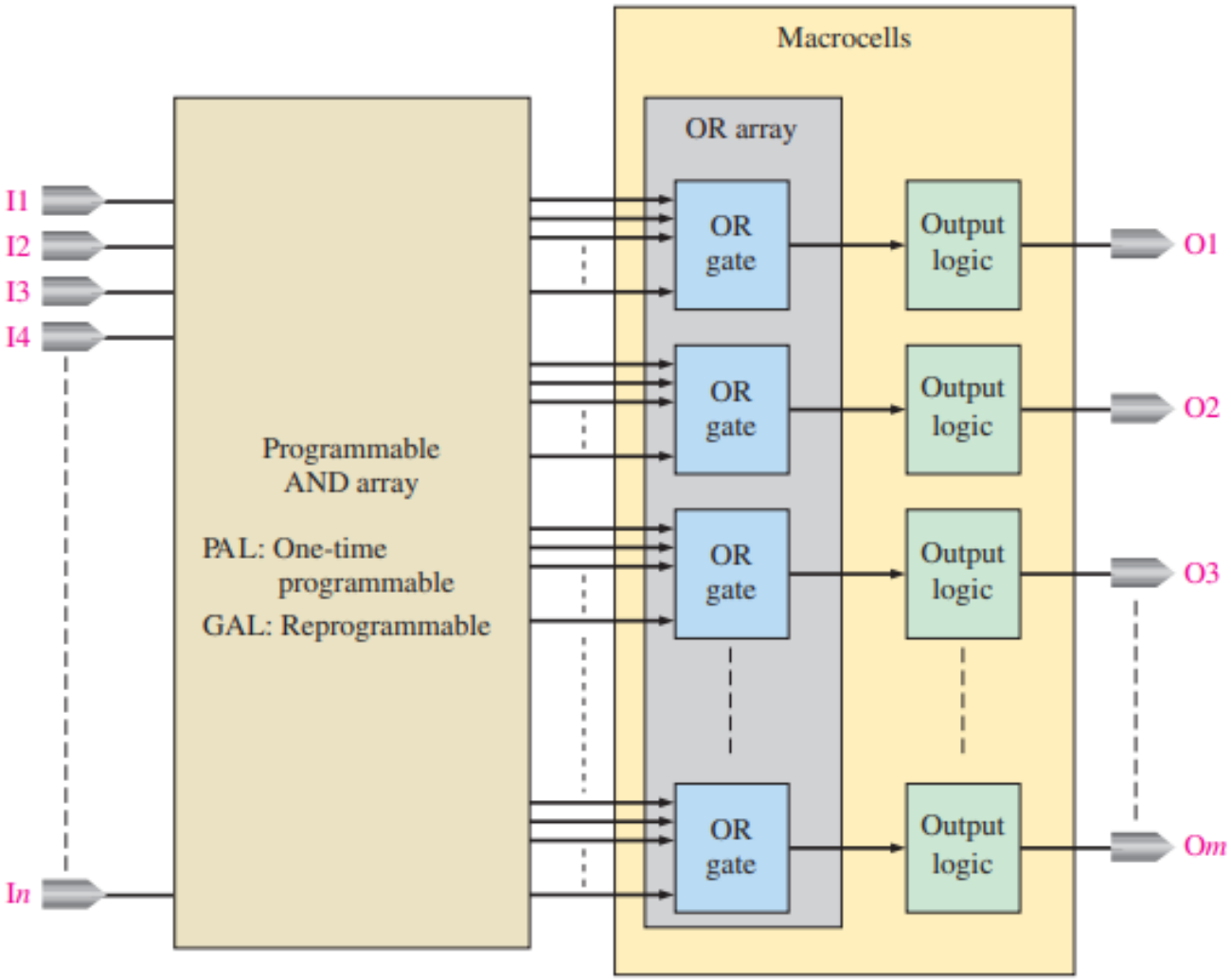
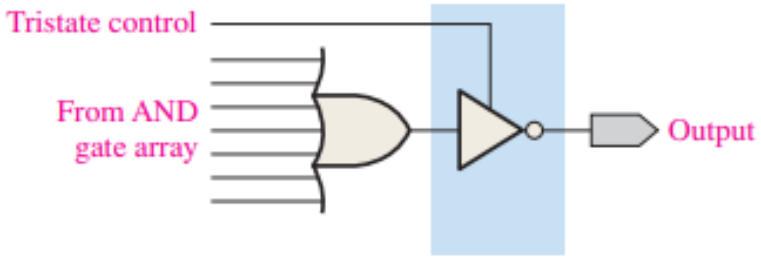


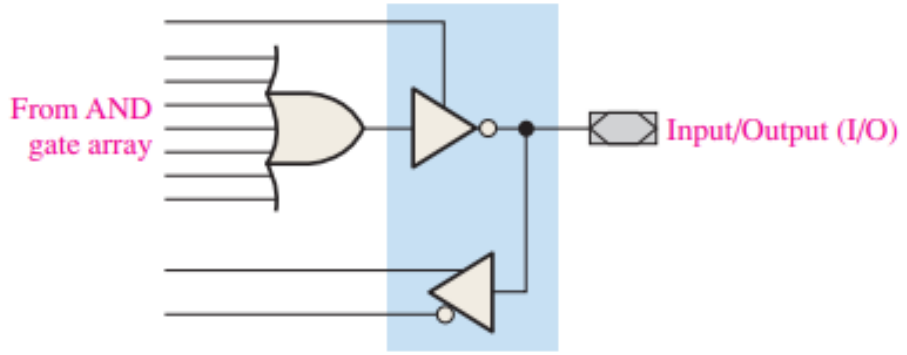
FIGURE 10–6 General block diagram of a PAL or GAL.

Macrocells

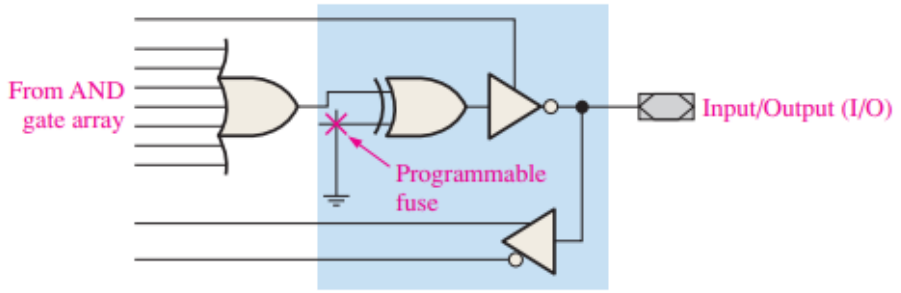
- A macrocell generally consists of one OR gate and some associated output logic. The macrocells vary in complexity, depending on the particular type of PAL or GAL. A macrocell can be configured for combinational logic, registered logic, or a combination of both.
- Registered logic means that there is a flip-flop in the macrocell to provide for sequential logic functions.



(a) Combinational output (active-LOW). An active-HIGH output would be shown without the bubble on the tristate gate symbol.



(b) Combinational input/output (active-LOW)



(c) Programmable polarity output

FIGURE 10–7 Basic types of PAL/GAL macrocells for combinational logic.

2. Complex Programmable Logic Devices (CPLDs)

The CPLD

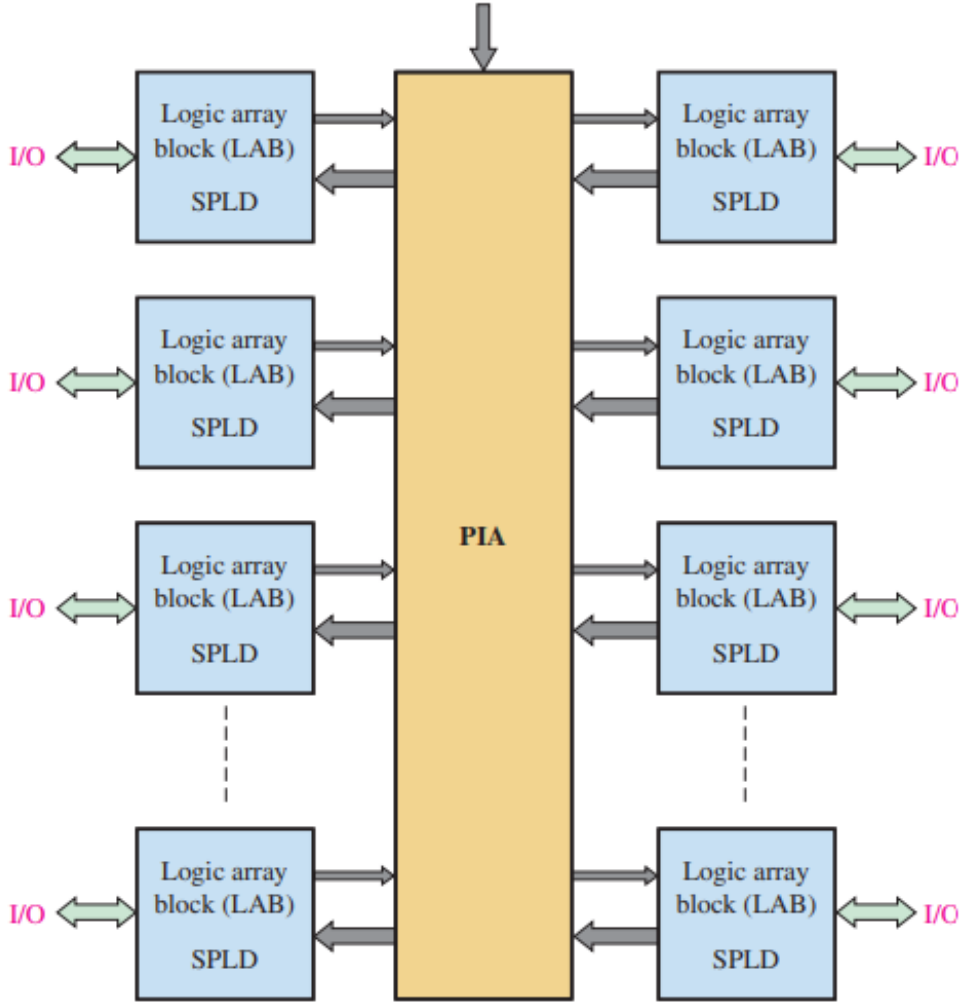


FIGURE 10–8 Basic block diagram of a generic CPLD.

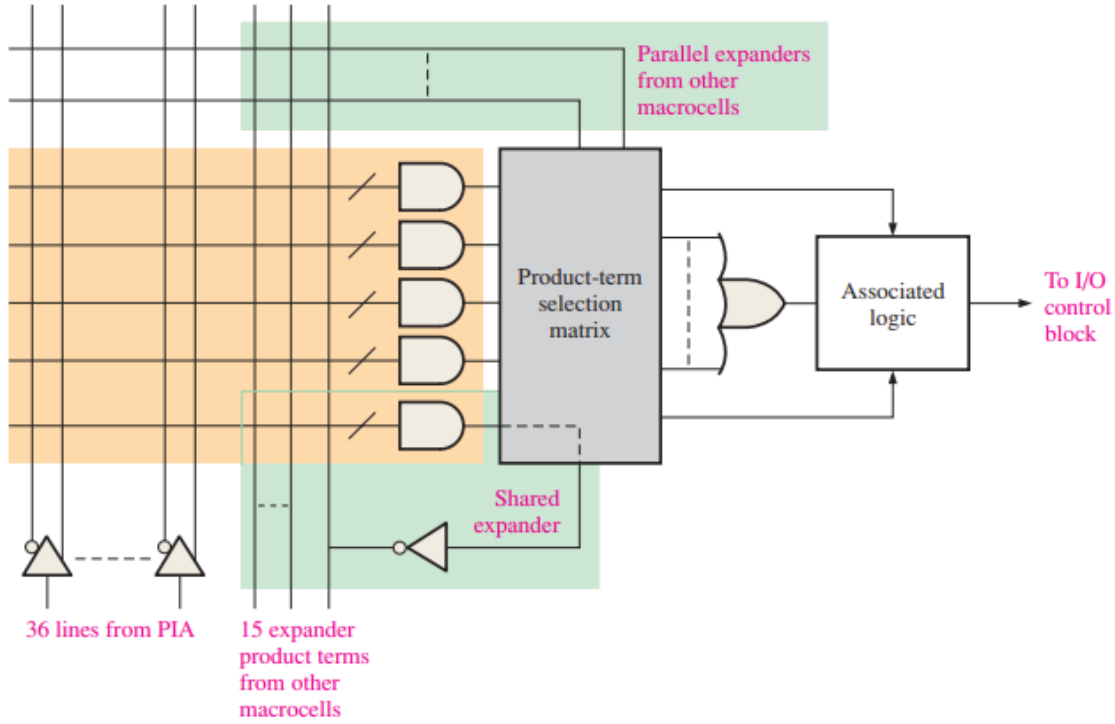
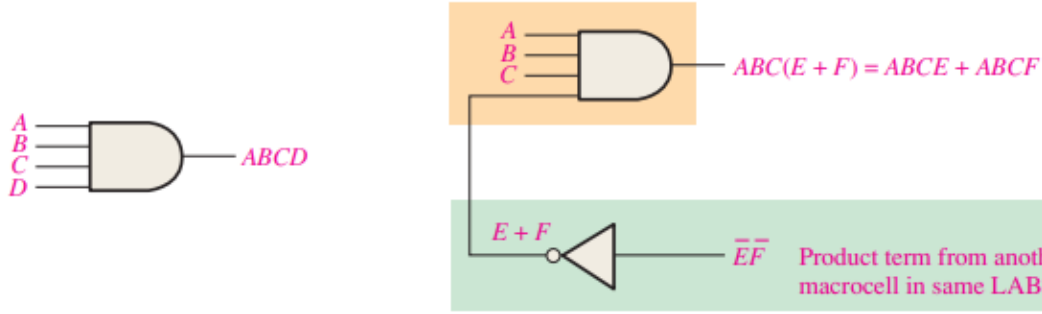


FIGURE 10–10 Simplified diagram of a macrocell in a typical CPLD.



(a) A 4-input AND array gate can produce one 4-variable product term.

(b) AND gate is expanded to produce two product terms.

FIGURE 10–11 Example of how a shared expander can be used in a macrocell to increase the number of product terms.

3. Macrocell Modes

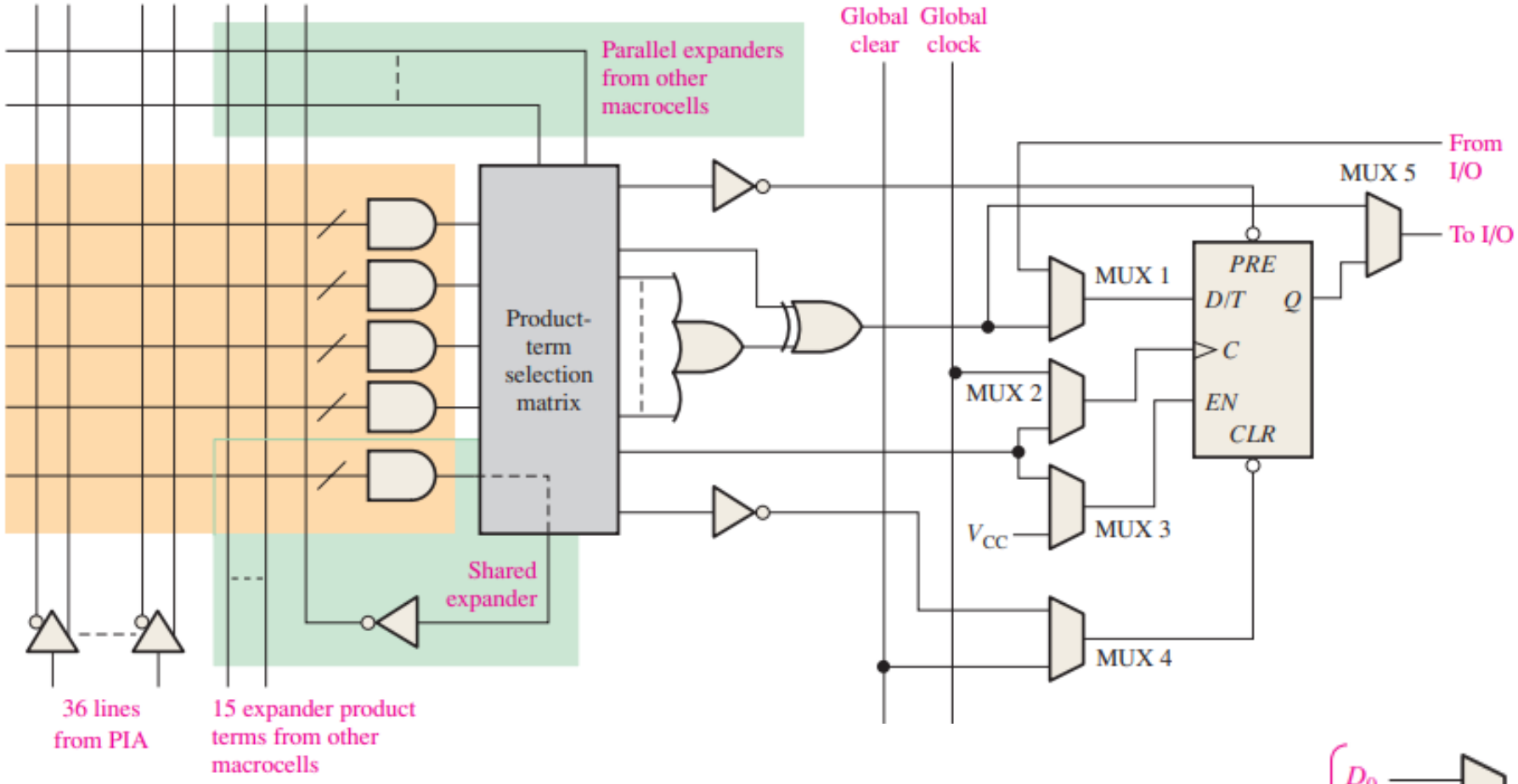


FIGURE 10–20 A CPLD macrocell.

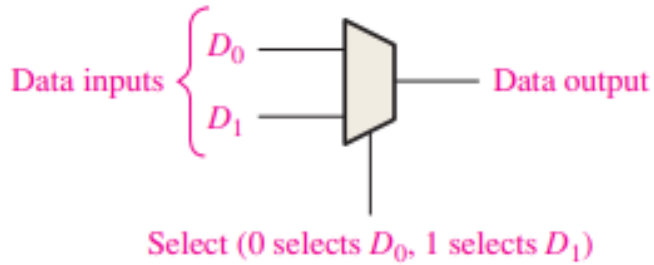


FIGURE 10–19 Commonly used symbol for a multiplexer. It can have any number of inputs.

The Combinational Mode

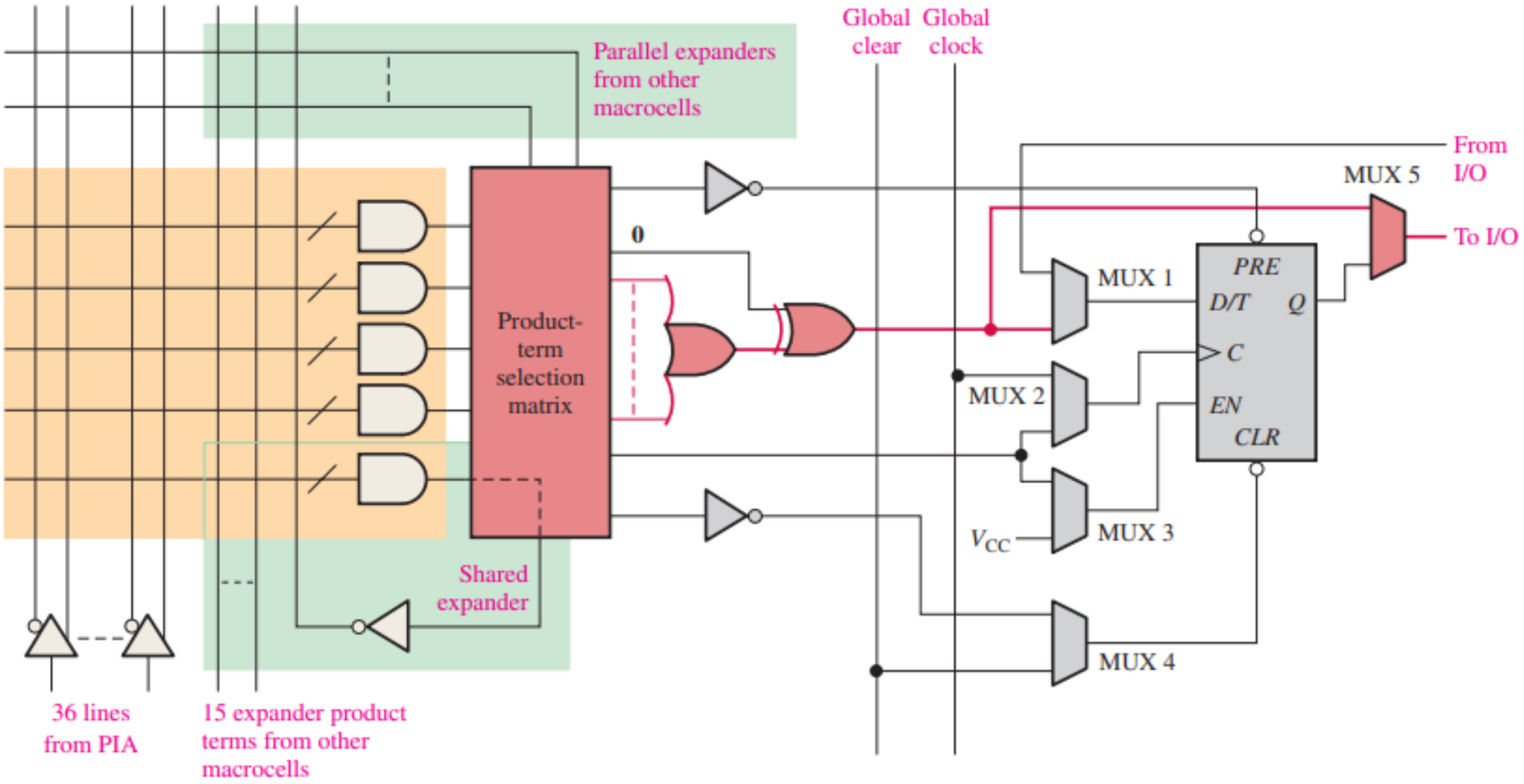


FIGURE 10–21 A macrocell configured for generation of an SOP logic function. Red indicates data path.

The Registered Mode

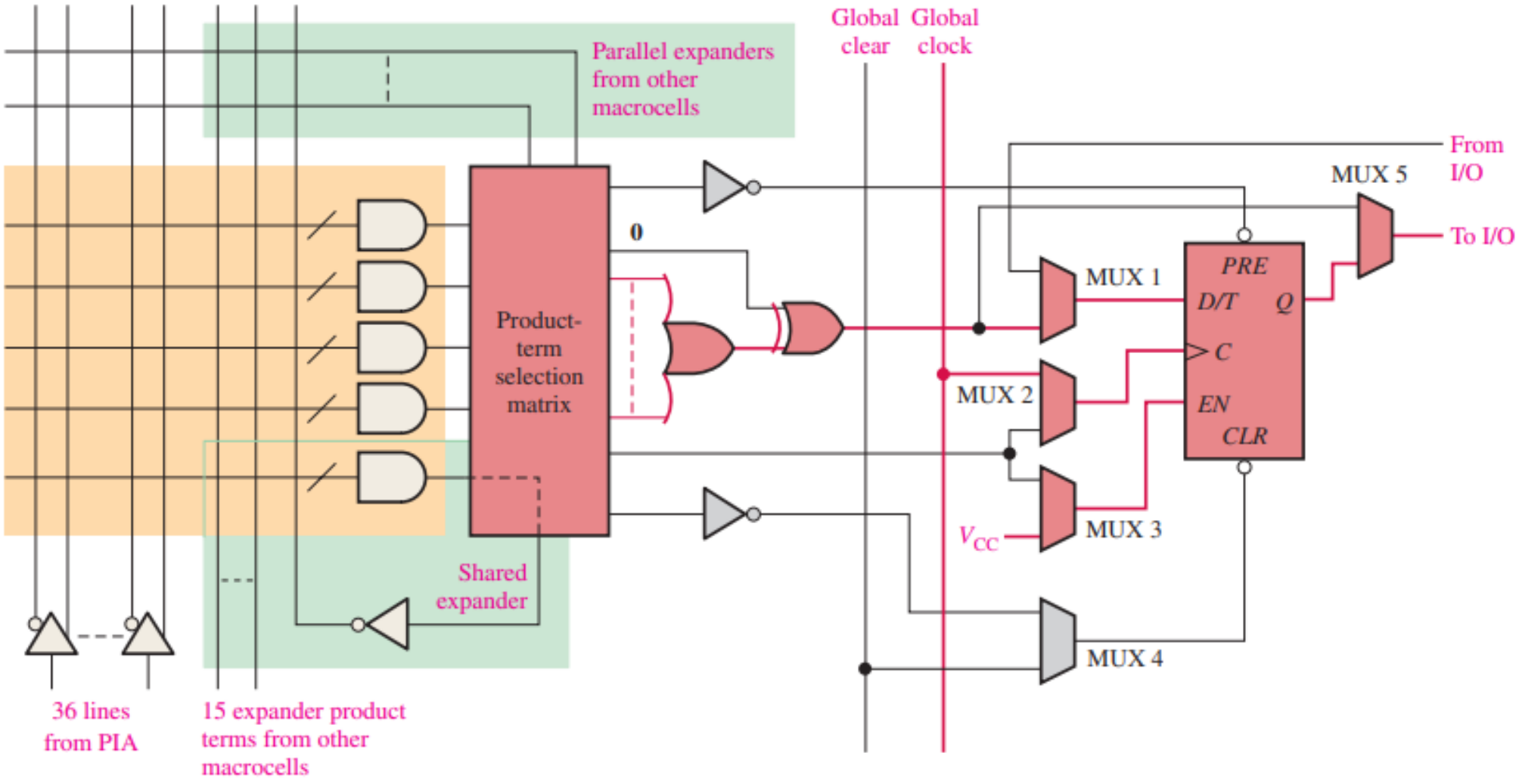


FIGURE 10–22 A macrocell configured for generation of a registered logic function. Red indicates data path.

4. Field-Programmable Gate Arrays (FPGAs)

- 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.

Configurable Logic Blocks

An FPGA logic block consists of several smaller logic modules that are the basic building units, somewhat analogous to macrocells in a CPLD.

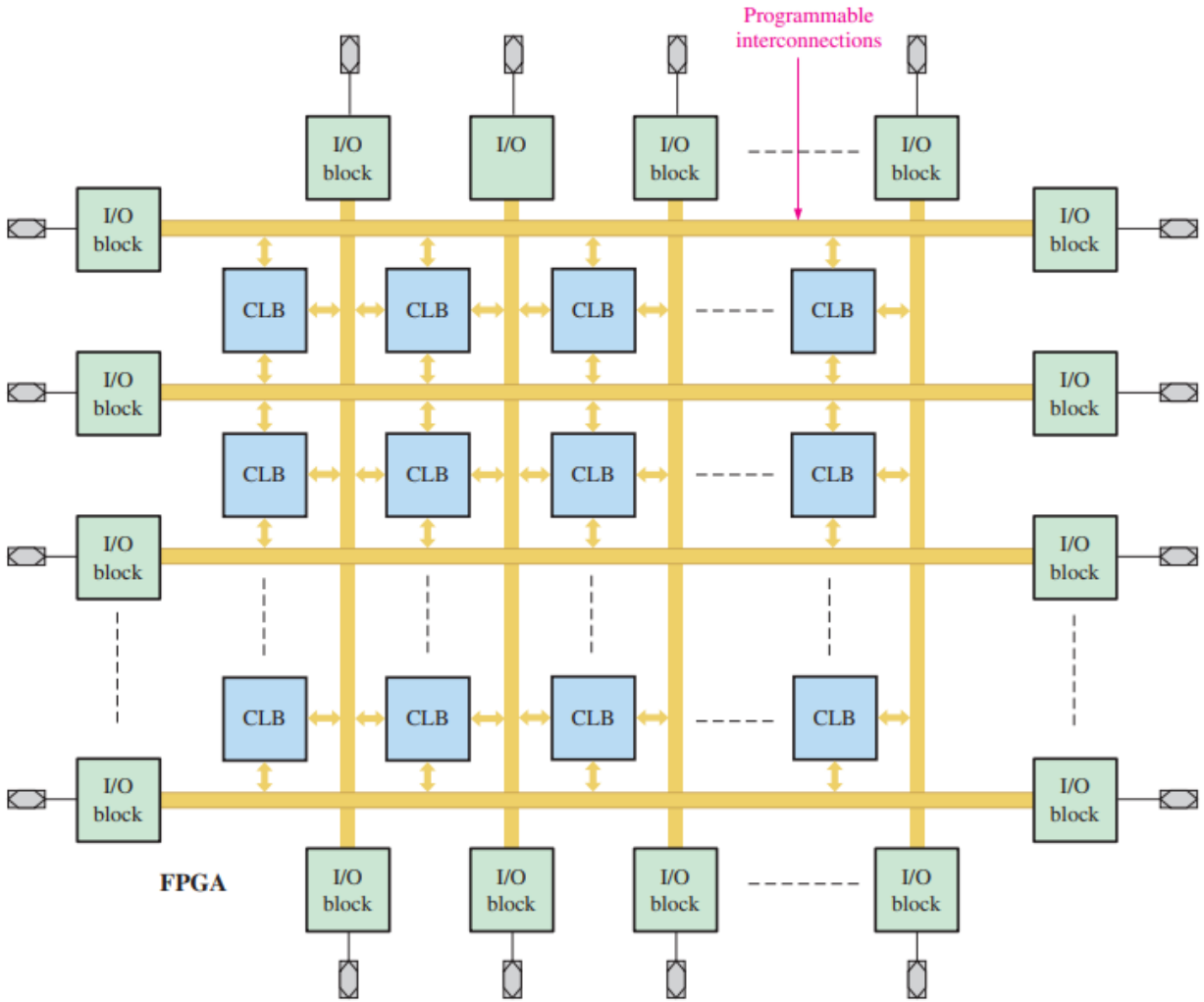


FIGURE 10–23 Basic structure of an FPGA. CLB is configurable logic block, also known as logic array block (LAB).

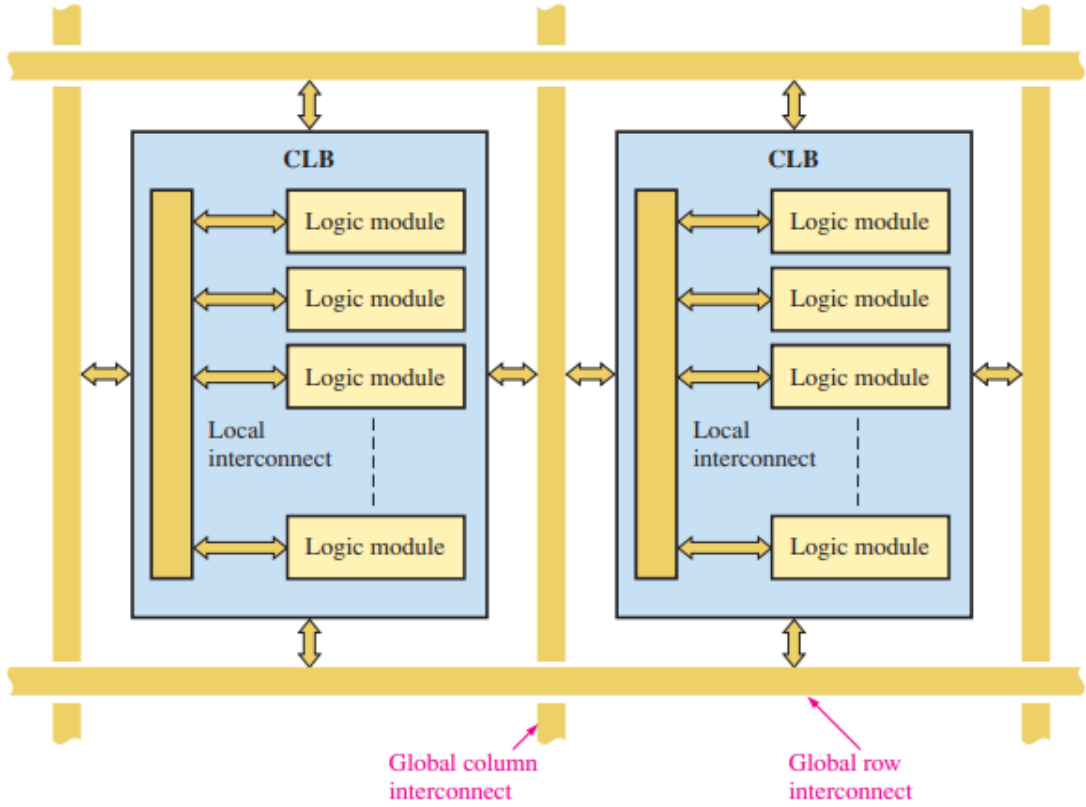


FIGURE 10–24 Basic configurable logic blocks (CLBs) within the global row/column programmable interconnects.

Logic Modules

A logic module in an FPGA logic block can be configured for combinational logic, registered logic, or a combination of both.

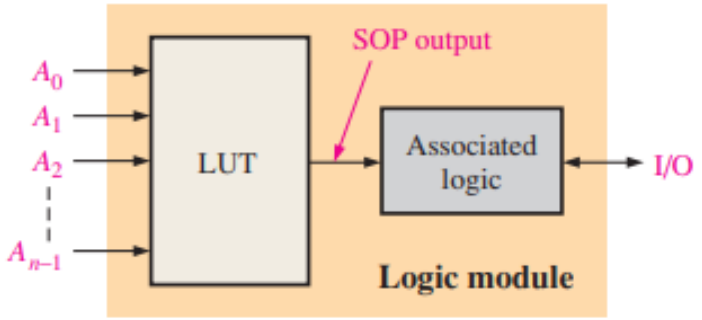


FIGURE 10–25 Basic block diagram of a logic module in an FPGA.

The resulting SOP output expression is:

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

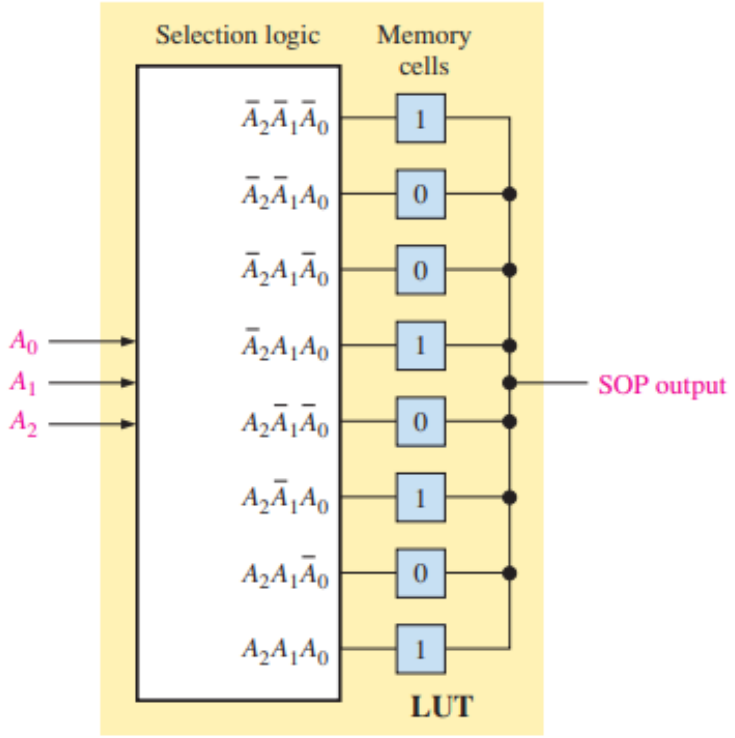


FIGURE 10–26 The basic concept of an LUT programmed for a particular SOP output.

EXAMPLE 10–2

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

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

Solution

A 1 is stored for each product term in the SOP expression, as shown in Figure 10–27.

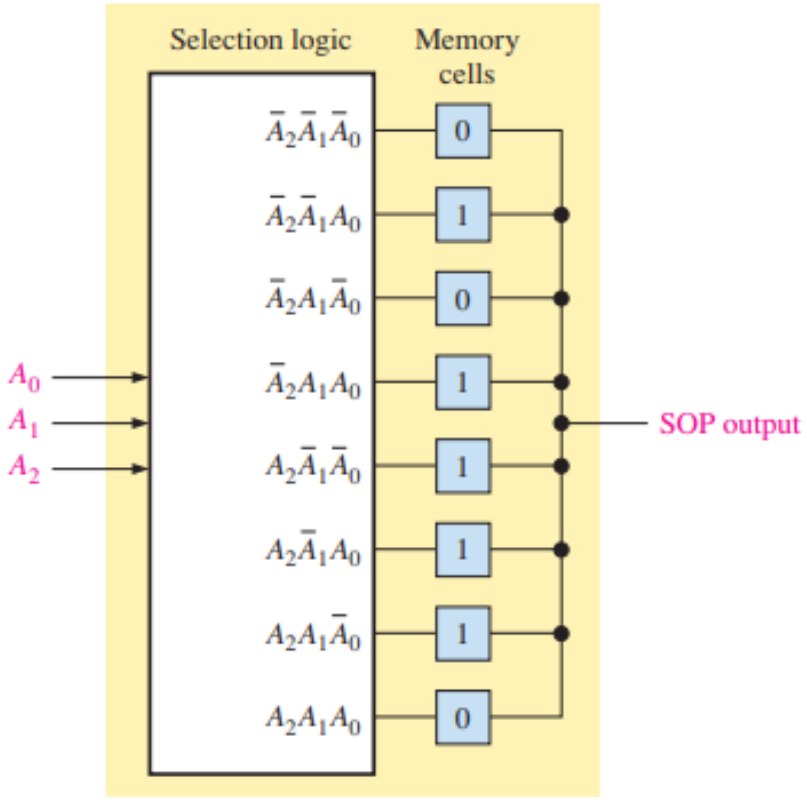


FIGURE 10–27

Operating Modes of a Logic Module

The normal mode is used primarily for generating combinational logic functions.

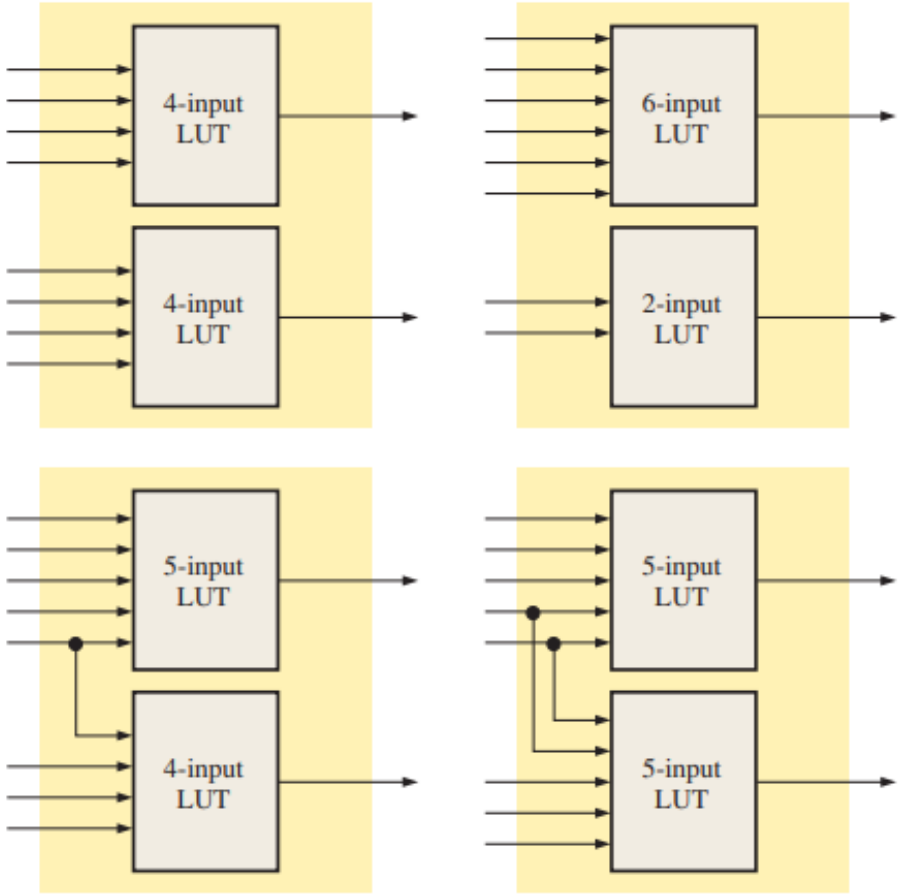


FIGURE 10–28 Examples of possible LUT configurations in a logic module (LM) in the normal mode.

The extended LUT mode allows expansion to a 7-variable function:

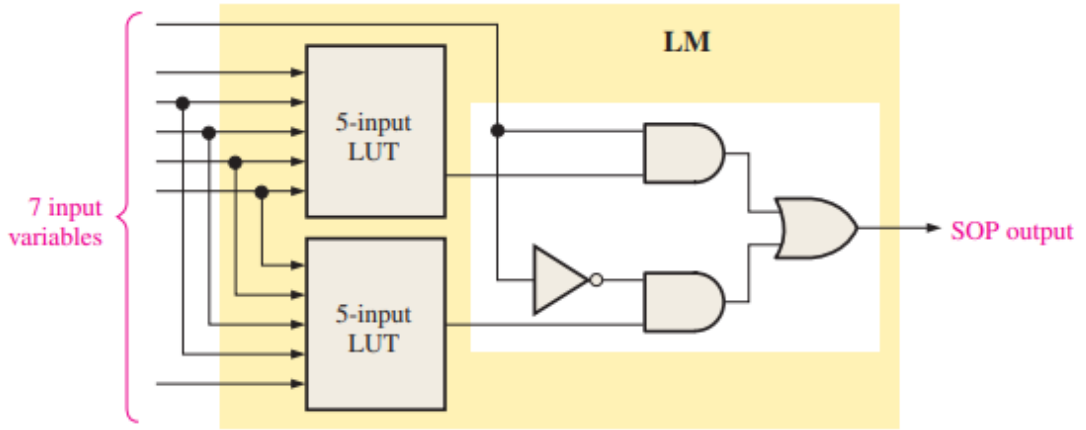


FIGURE 10–29 Expansion of a logic module (LM) to produce a 7-variable SOP function in the extended LUT mode.

EXAMPLE 10–3

A logic module is configured in the extended LUT mode, as shown in Figure 10–30. For the specific LUT outputs shown, determine the final SOP output.

Solution

The SOP output expression is as follows:

$$\overline{A}_5A_4A_3A_2A_1A_0 + A_5\overline{A}_4A_3A_2A_1A_0 + A_5A_4A_3A_2A_1A_0 + A_6A_5A_4A_3\overline{A}_2\overline{A}_0 + A_6A_5\overline{A}_4A_3A_2\overline{A}_0 + A_6A_5A_4A_3A_2\overline{A}_0$$

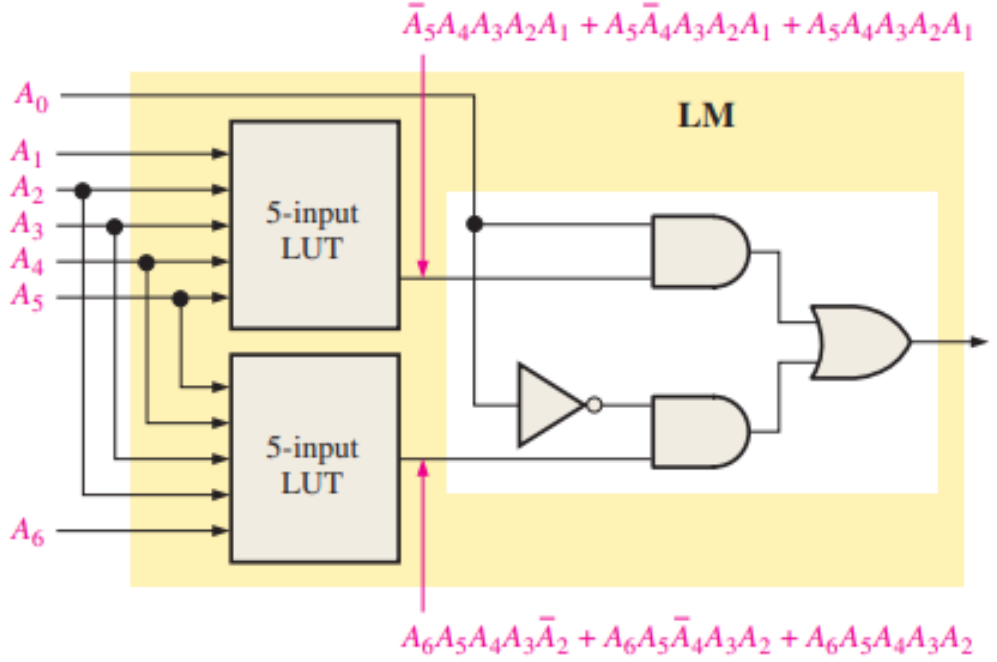
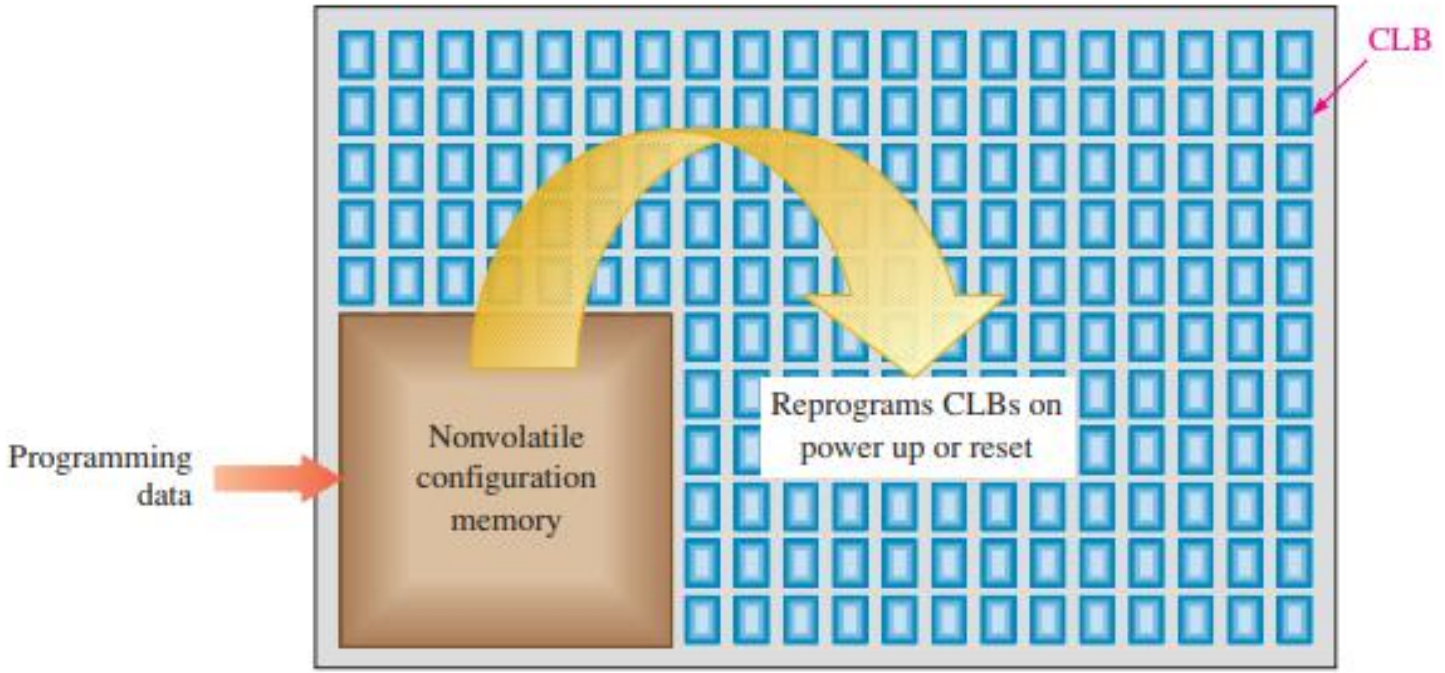
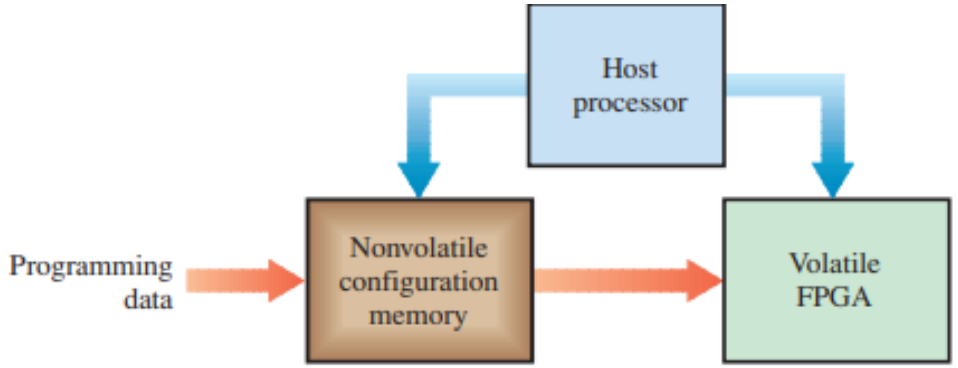


FIGURE 10–30

SRAM-Based FPGAs



(a) Volatile FPGA with on-the-chip nonvolatile configuration memory



(b) Volatile FPGA with on-board memory and host processor

FIGURE 10–31 Basic concepts of volatile FPGA configurations.

FPGA Cores

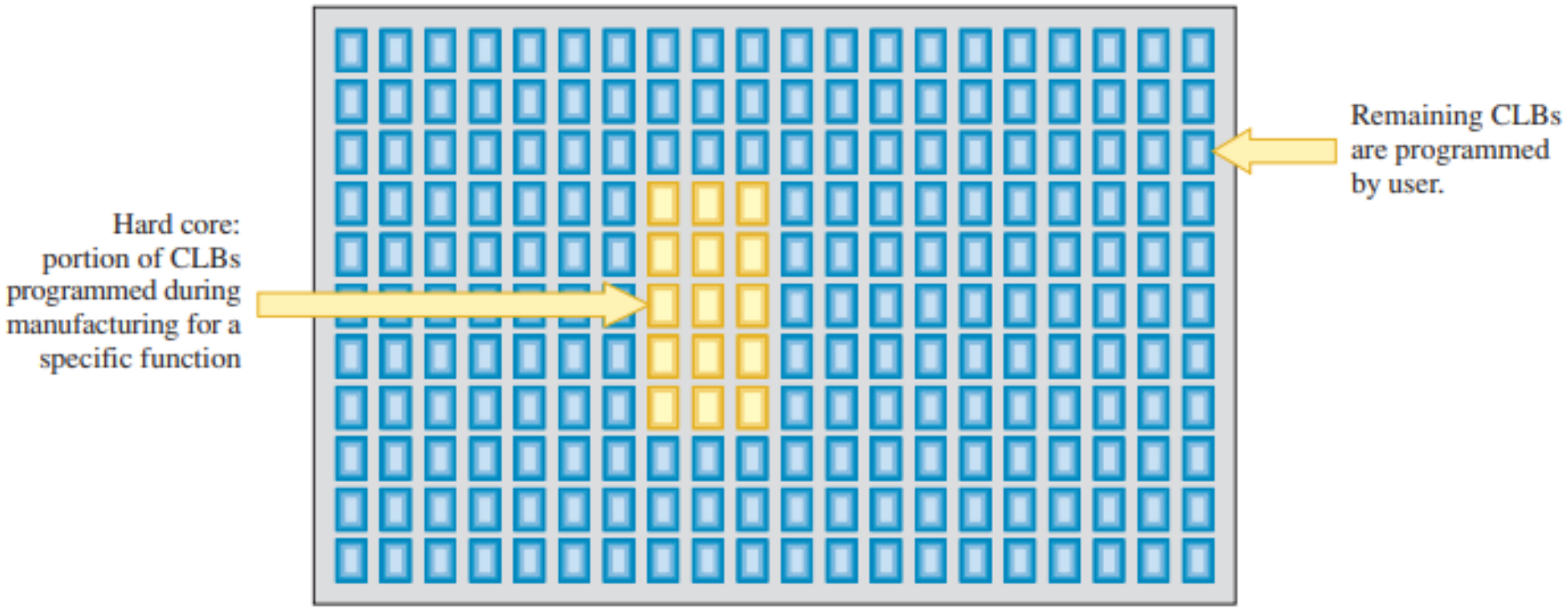


FIGURE 10–32 Basic idea of a hard-core function embedded in an FPGA.

Those FPGAs containing either or both hard-core and soft-core embedded processors and other functions are known as platform FPGAs because they can be used to implement an entire system without the need for external support devices

Embedded Functions

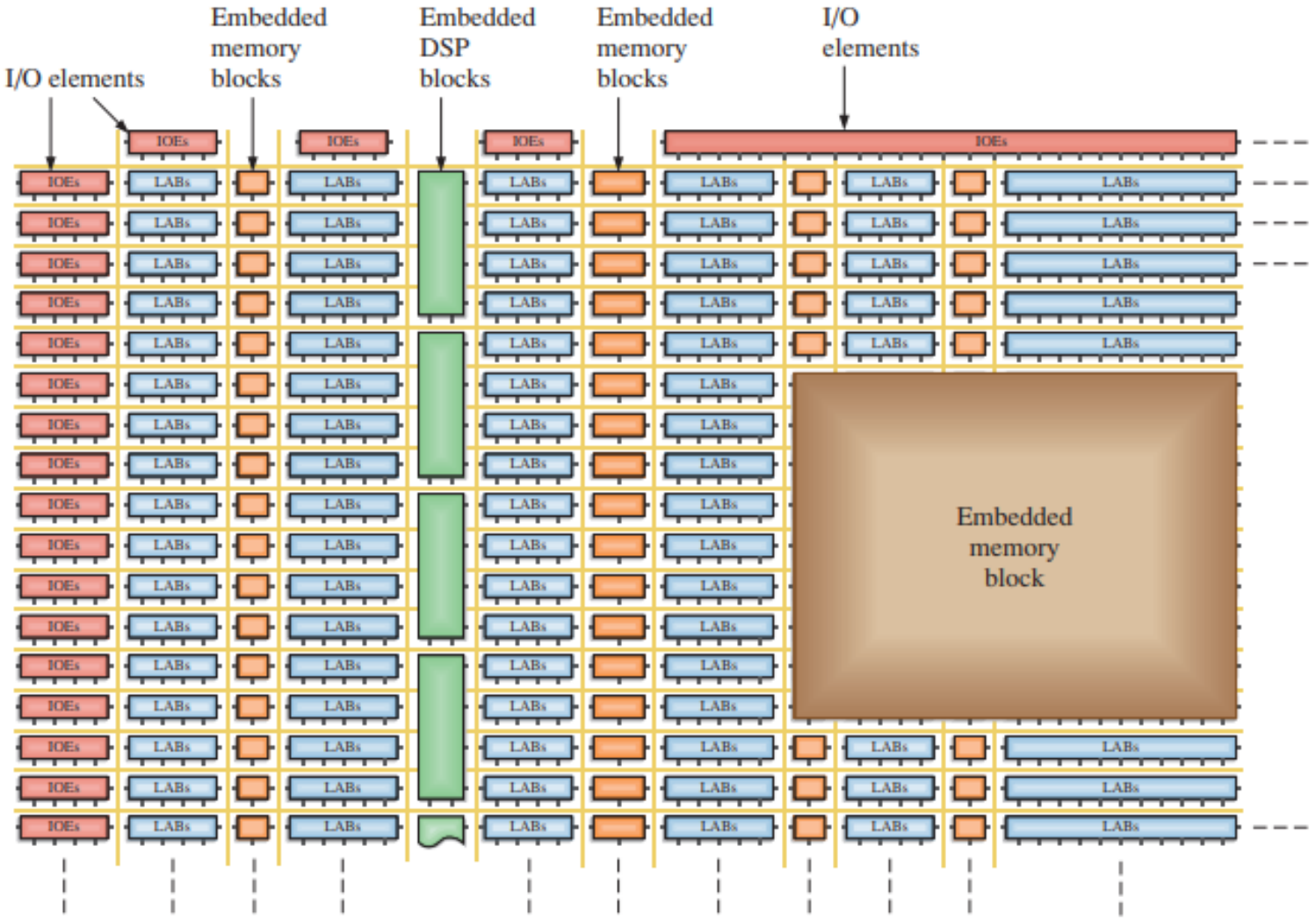


FIGURE 10–33 Example FPGA block diagram.

Specific FPGA Devices

TABLE 10–3

FPGA manufacturers.

Manufacturer	Series Name(s)	Design Software	Website
Altera	Stratix Aria Cyclone	Quartus II	Altera.com
Xilinx	Spartan Artix Kintex Virtex	ISE Design Suite	Xilinx.com
Lattice	iCE40 MachX02 Lattice ECP3 LatticeXP2 LatticeGC/M	Lattice Diamond iCEcube2	Latticesemi.com
Atmel	AT40	IDS	Atmel.com

TABLE 10–4

Selected FPGA parameters.

Feature	Range
Number of LEs	1,500–813,000
Number of CLBs	26–359,000
Embedded memory	26 kb–63 Mb
Number of I/Os	18–1200
DC operating voltage	1.8 V, 2.5 V, 3.3 V, 5 V

5. Programmable Logic Software

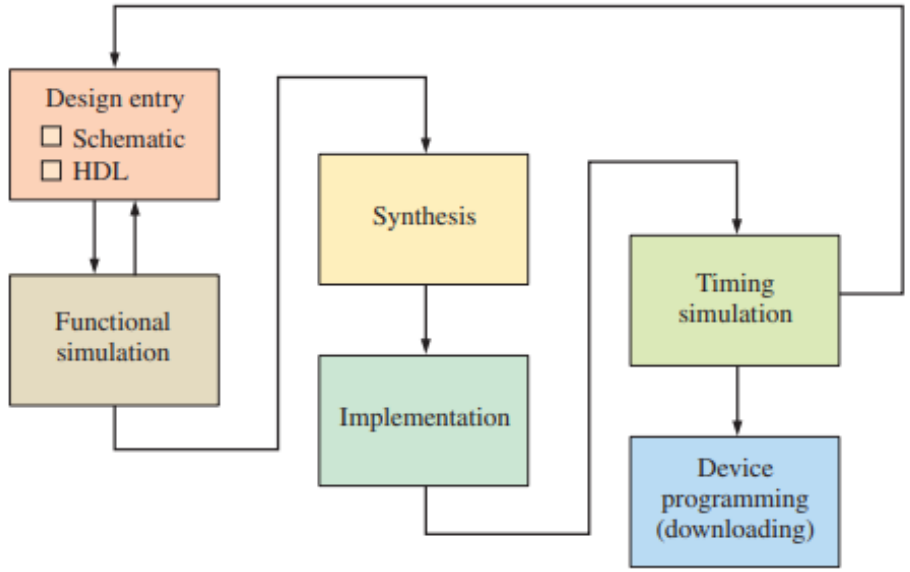


FIGURE 10–34 General design flow diagram for programming a SPLD, CPLD, or FPGA.

(a) Computer

(b) Software (CD or Website download)

(c) Device

(d) Programming hardware (programming fixture or development board with cable for connection to computer port)

FIGURE 10–35 Essential elements for programming an SPLD, CPLD, or FPGA.
(d) photo courtesy of Digilent, Inc.

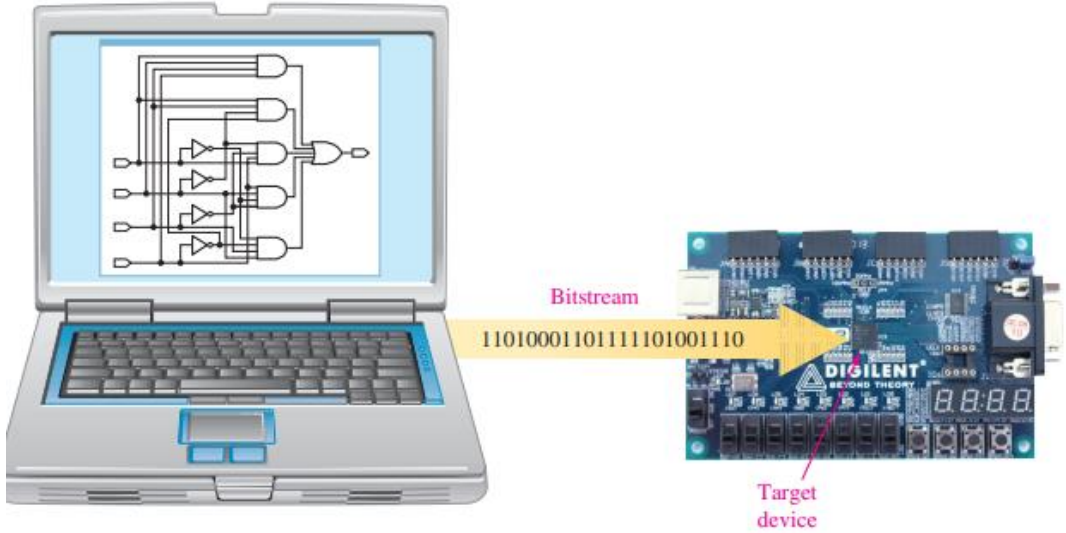
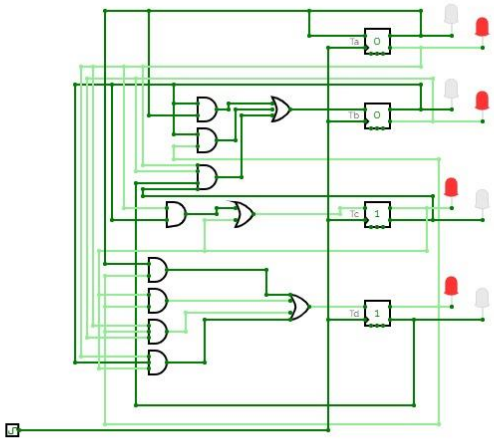


FIGURE 10–44 Downloading a design to the target device. (Photo courtesy of Digilent, Inc.)

THE END

Lecture 10: Programmable Logic



INSTRUCTOR: Dr. Vuong Quoc Bao