



EIE 2050 Digital Logic and Systems

Chapter 10: Programmable Logic

Simon Pun, Ph.D.



Announcements

- Final Exam is scheduled on **5/13 (Sat) 10.00-12.00**
- Scope : Chapter 1 – Chapter 11
- No reference papers / calculators
- If you don't find your name, please contact me ASAP

Zhiren (志仁) 205

陈炜铭	彭一伟	毛曦	马涵	漆艺玲	陈健	阮骏文	聂若霄	李子樵			
薛浩楠	王志成	柳怡如	李子桓	孙逸飞	龙及远	Raymond TUMIWA	Xing zhe CHAI	金韬	严子衡	邓思彦	唐浩灏
沈驰皓	张旦阳	Juan WIBOWO	Annabel CHRISTABEL	Nicholas OH	Alfonsus RENDY	Bryan JONATHAN	Kelvin IGNATIUS	William ARCHIETA	孙家骏	童憧	王博阳
Bryan Halim	Nandita VENKATESH KURINCHI	Dokyung LEE	蔡柳渔	高若凡	何欣桐	黄子鸥	金成	刘昱含	梅傲涵	牛博琪	王广申
饶胜	夏渝	Daniel Tanuwijaya	徐瀚翔	罗啸行	梁馨予	Nikita TANJAYA	袁晨昕	杨超豪	Jason GUNAWAN	Adelita SELINDRA	Cahyadi BODHIPUTRA



Zhiren (志仁) 206

张集昭	赵梓浩	朱治平	胡瑜桢	安子航	白昱轩	包景致	蔡欣言	岑朗			
陈彦睿	陈雨芃	陈雨阳	成瑀	程超凡	程乐天	代一飞	戴子博	董译阳	段皓介	方瀚波	冯彦博
郭铮	韩铠瀛	何旦祺	何俊仪	何天与	何绪	侯天赐	侯一凡	胡琳悦	胡煜轩	黄子畅	姬嘉
蒋一歌	康子墨	亢健桐	李弘政	李瑾欣	李康硕	李心怡	李羽韬	李政达	林姝宜	林杨	刘金鹭
陈建臻	葛达	姜皓文	刘铭昊	陈可欣	耿靖远	蒋天依	刘奕琛	张祺楠	曾广骏	曾子曦	陈凡欣雨

Zhiren (志仁) 417

刘彧卓	吕欣荻	吕咏荷	吕游	马腾飞	毛杰	毛斯玄	欧海榕	潘婕			
史东泽	史婧瑄	史雪祺	宋刘天骄	隋卓远	孙怀韬	孙宇轩	谈常睿	唐世简	唐致宁	陶盛	田镭
涂喻钊	庹先益	万柯睿	汪子茼	王程玮	王昊轩	王浩哲	王家驹	王楷泽	王敏骏	王儒昕	王鑫宇
王子唯	韦嘉祺	邬骐骏	吴贻谋	吴羿樊	吴优	吴悦颜	伍尧杰	武彦睿	夏汪龙	夏鑫	夏禹扬
邵棋	田庆尧	王一凡	谢远哲	施杰傲	田一夫	王智一	谢子钰	周炫	彭志东	戚睿瑄	乔雅琦

Zhiren (志仁) 421

熊朗	熊宜亭	徐培哲	徐启睿	徐源	许烈骁	许如蓝	薛任祺	鄢子彧	闫正		
杨陈浩然	杨成果	杨嘉言	杨凯文	杨逸洲	叶行	于靖霖	俞卓言	喻嘉炜	袁彬源	苑一诺	詹天泽
张童	张晓宇	张旭凡	章洪涛	赵翔宇	赵浙绮	赵竹风	郑乐駢	郑立昊	郑雨涛	郑泽坤	钟格非
周震寰	周铮	朱新源	朱逸帆	邱梓伦	华培淳	杨致远	杨博超	张皓淇	周锐涛	阳易显	阳雨哲



Last Week

□ Finite State Machines

- ◆ Moore machines
- ◆ Mealy machines

□ Counters

- ◆ Asynchronous
- ◆ Synchronous

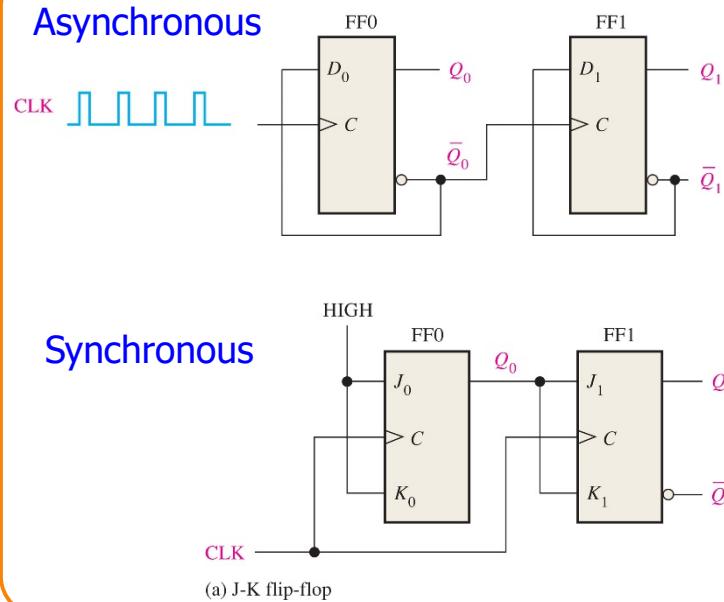
□ Up/Down Counters

□ Design of Synchronous Counters

□ Cascaded Counters

□ Counter Decoding

□ Troubleshooting



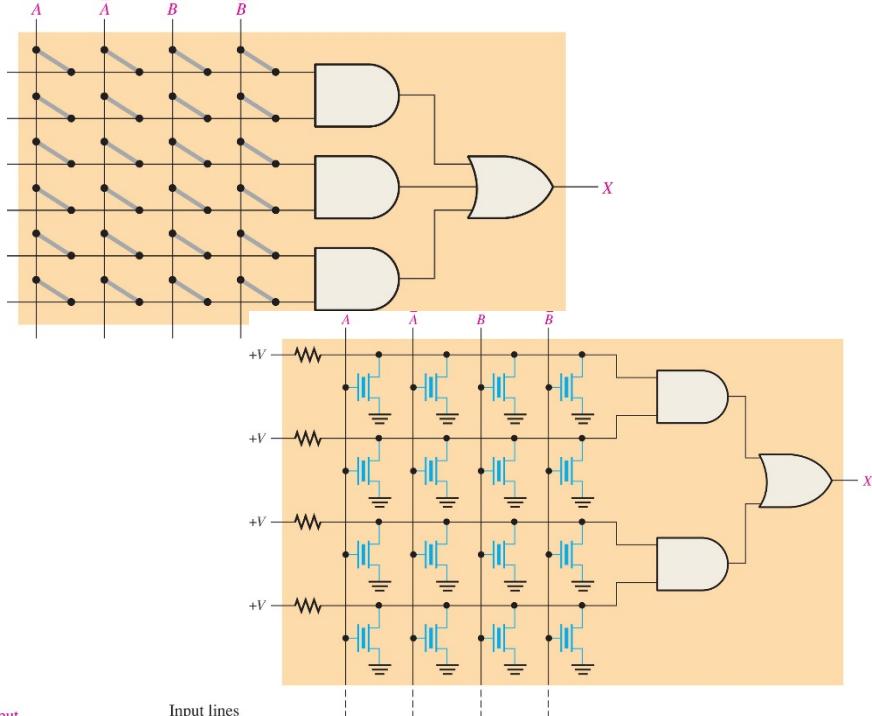
- Step 1: State Diagram
- Step 2: Next-State Table
- Step 3: Flip-Flop Transition Table
- Step 4: Karnaugh Maps
- Step 5: Logic Expressions for Flip-Flop Inputs
- Step 6: Counter Implementation



Simple Programmable Logic Devices (SPLDs)

□ PAL (Programmable Array Logic)

- ◆ Implemented with fuse process technology
- ◆ One-time programmable (OTP)
- ◆ Output: sum-of-products (SOP)

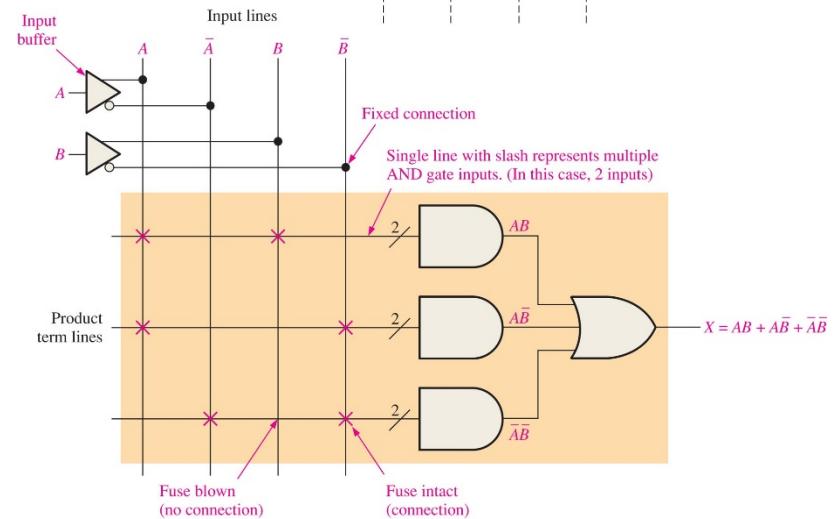


□ GAL (Generic Array Logic)

- ◆ In essence, is reprogrammed PAL
- ◆ Uses a reprogrammable process technology,
i.e. EEPROM, instead of fuses.

□ Simplified Notation

- ◆ The triangle symbol : a buffer
- ◆ The AND gate symbol has a single input line with a slash and a digit representing the actual number of inputs.



PAL/GAL General Block Diagram

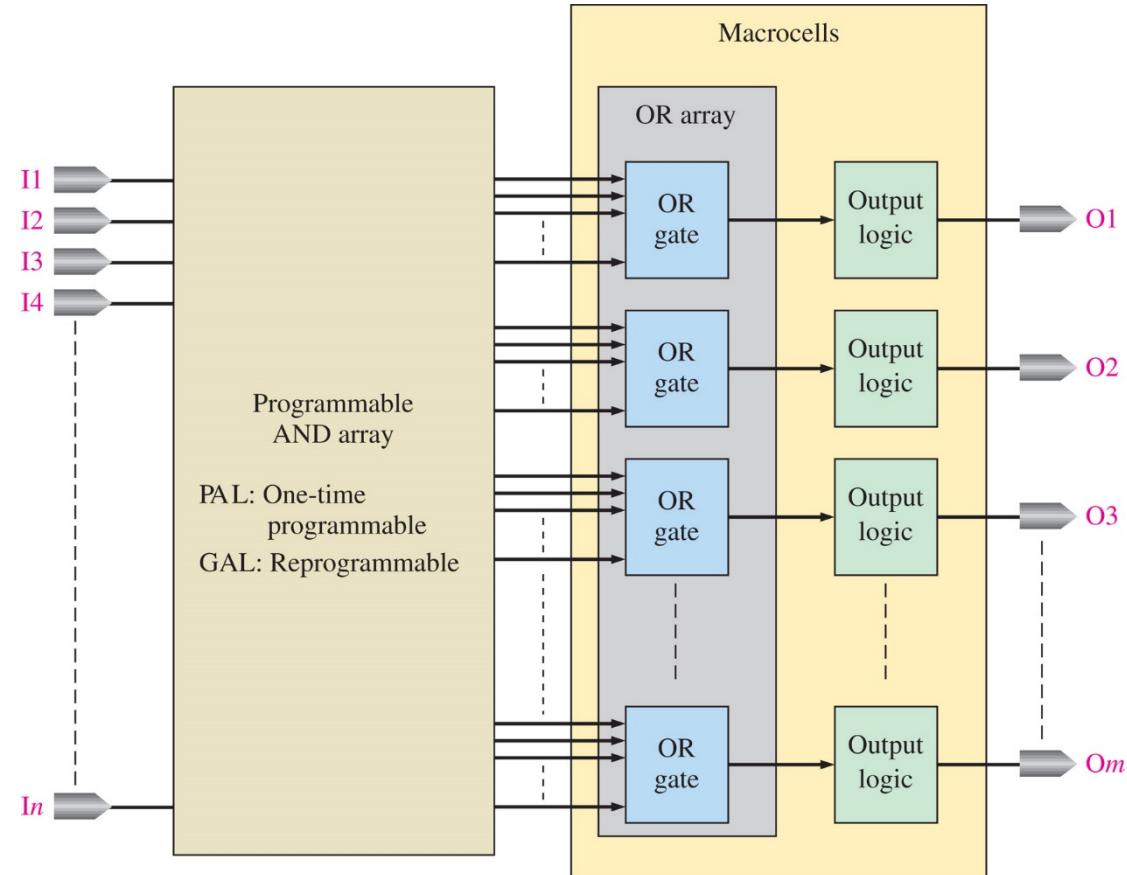
□ Input → PAL/GAL → Macrocells →

Output

□ Input/Output: 20~28 pins

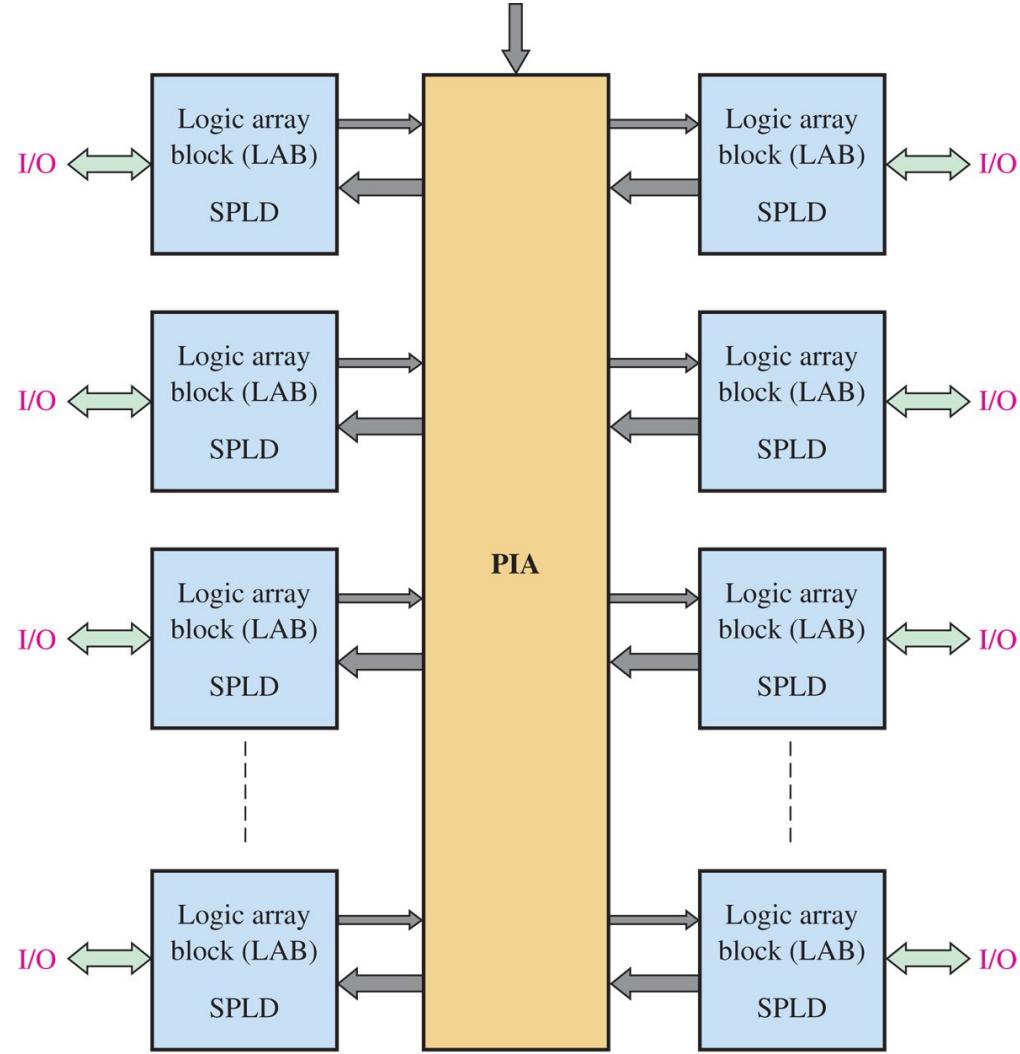
□ Macrocells

- ◆ Configured for combinational logic, registered logic, or a combination of both

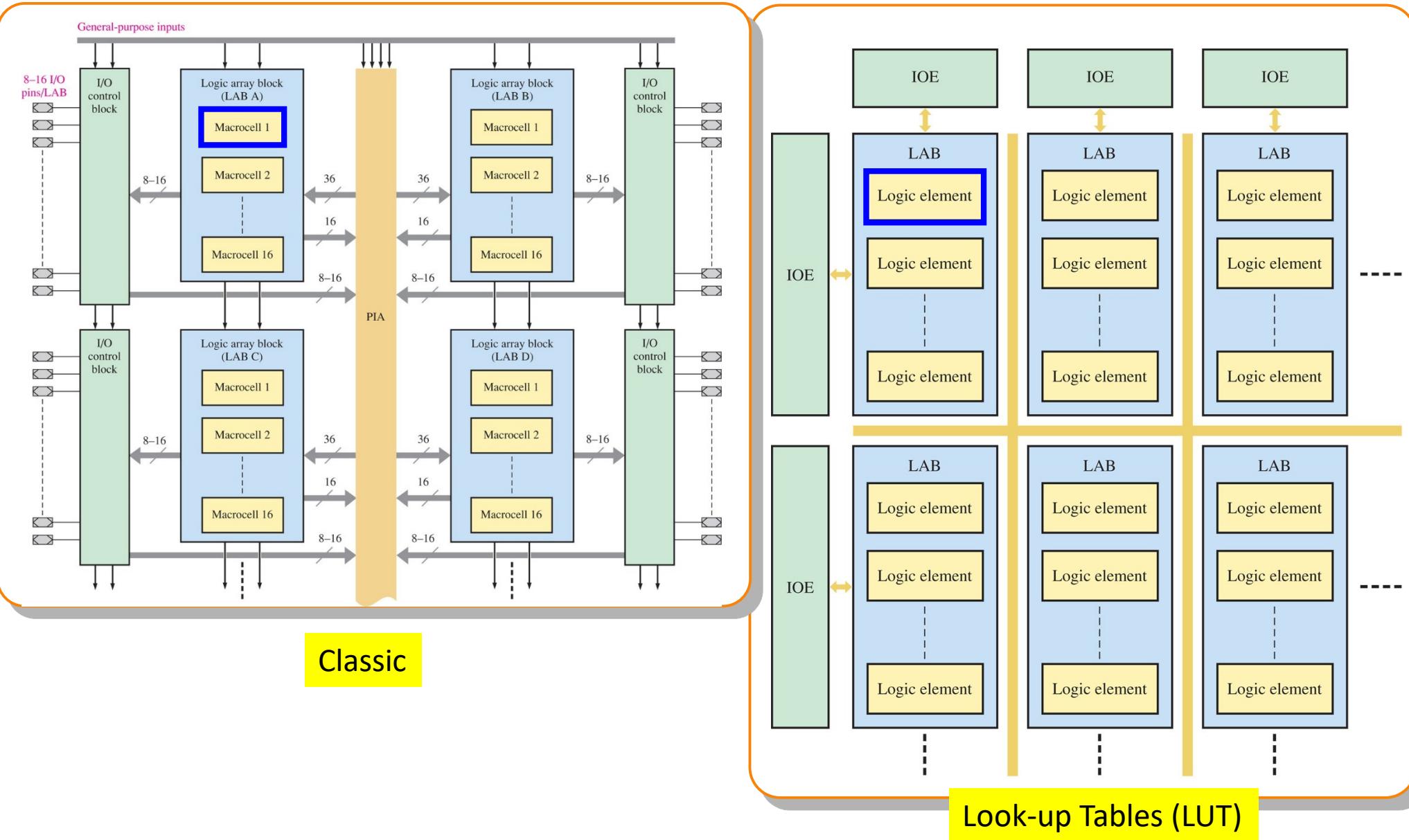


Complex Programmable Logic Devices (CPLDs)

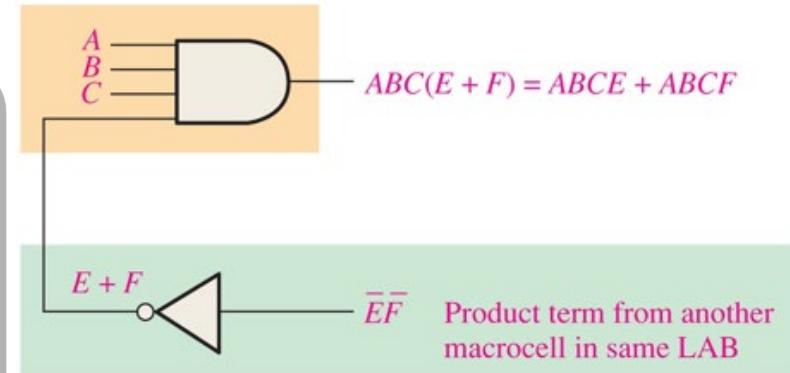
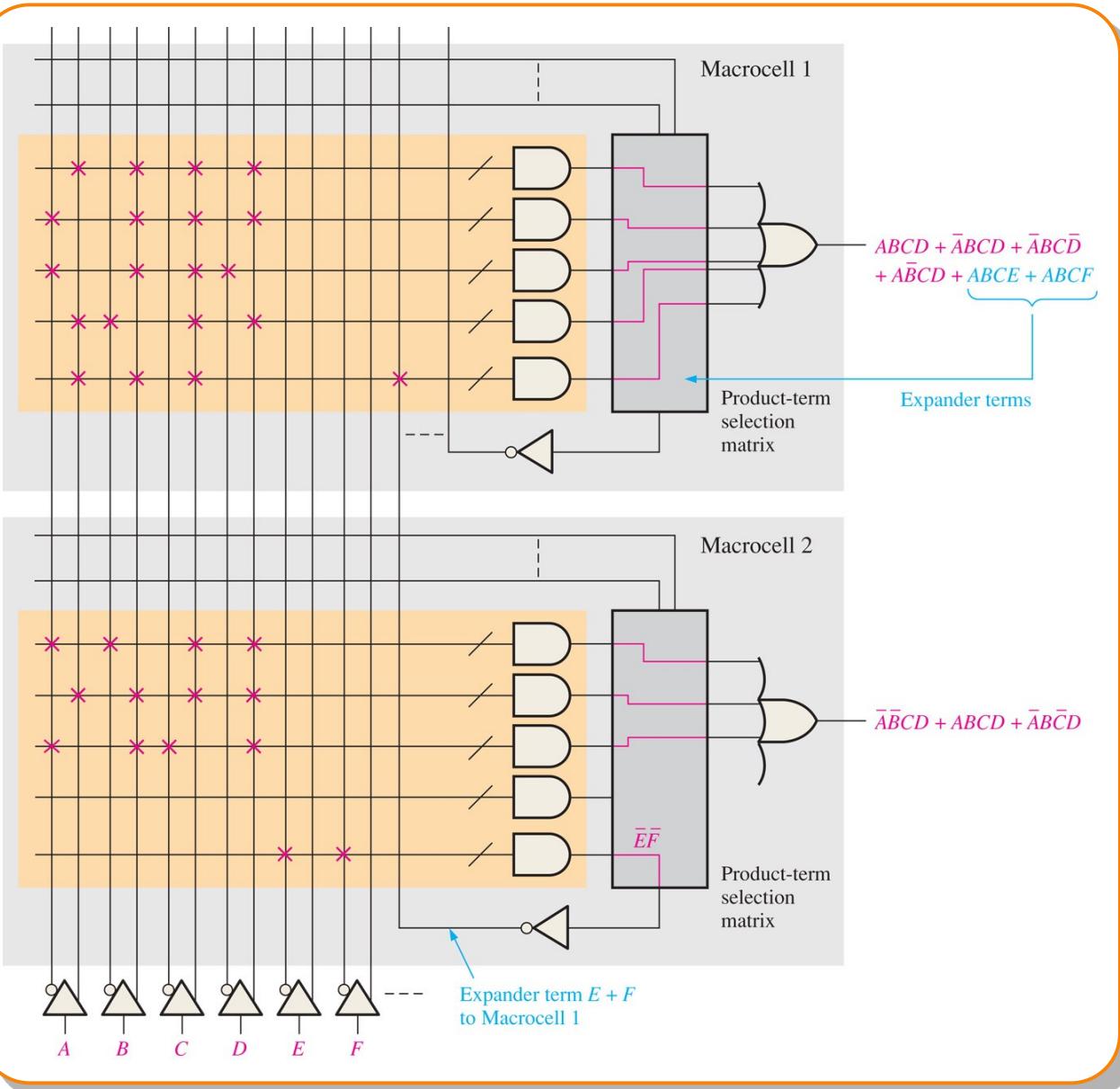
- Composed of multiple SPLD arrays with programmable interconnections called PIA (Programmable Interconnect Array).
- Each SPLD array in a CPLD is referred to as a LAB (Logic Array Block).
- The CPLD density ranging from tens of macrocells to over 1,500 macrocells in packages with up to several hundred pins.
- Major manufacturers: Altera, Xilinx, Lattice, and Atmel



Classic versus LUT CPLD Architectures



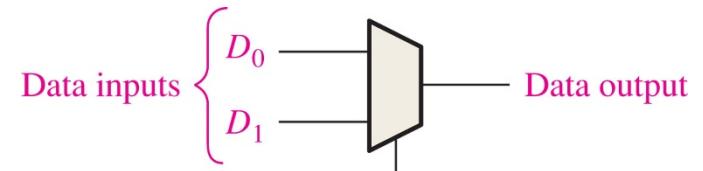
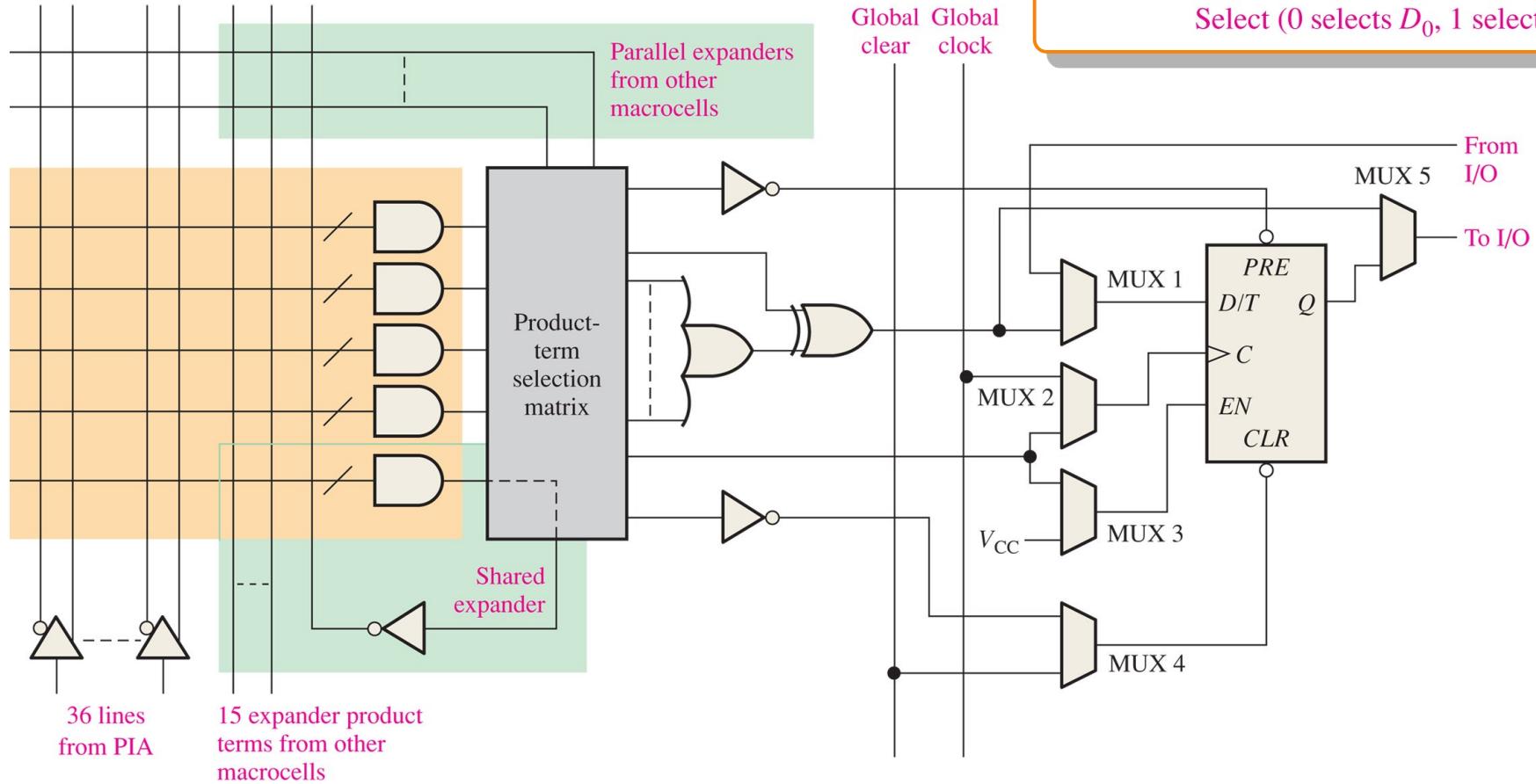
Shared Expander



- Macrocell 2 is underutilized
- It generates a shared expander term ($E + F$) that connects to the fifth AND gate in Macrocell 1 to produce an SOP expression with six product terms.



Macrocell Modes

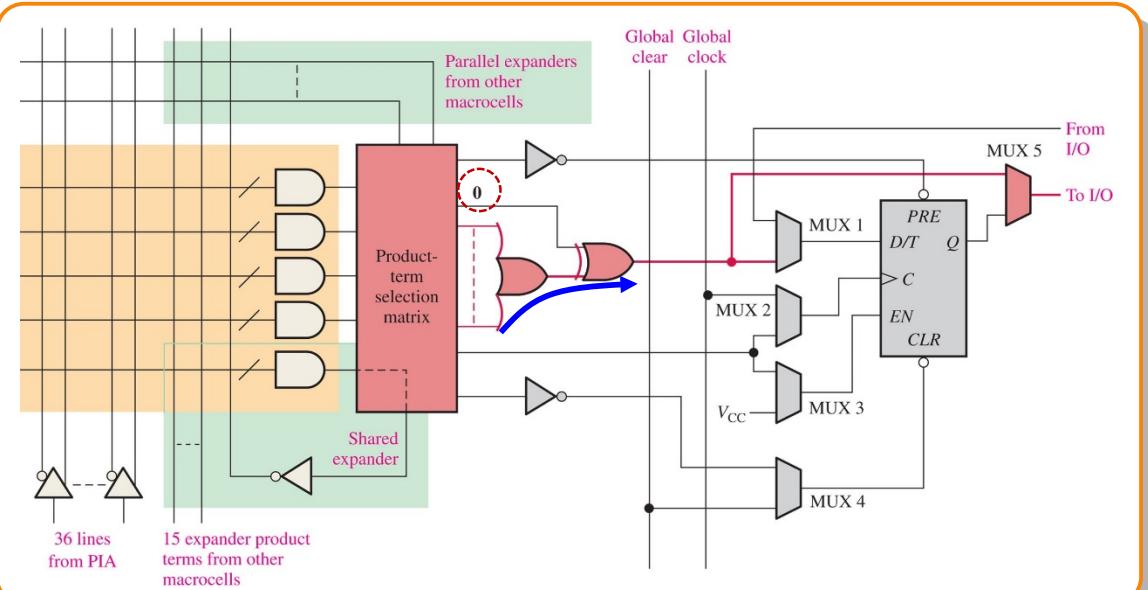


Select (0 selects D_0 , 1 selects D_1)

- Multiplexer has 2 data inputs and a select input for programmable selection
- The Combinational versus Registered Mode

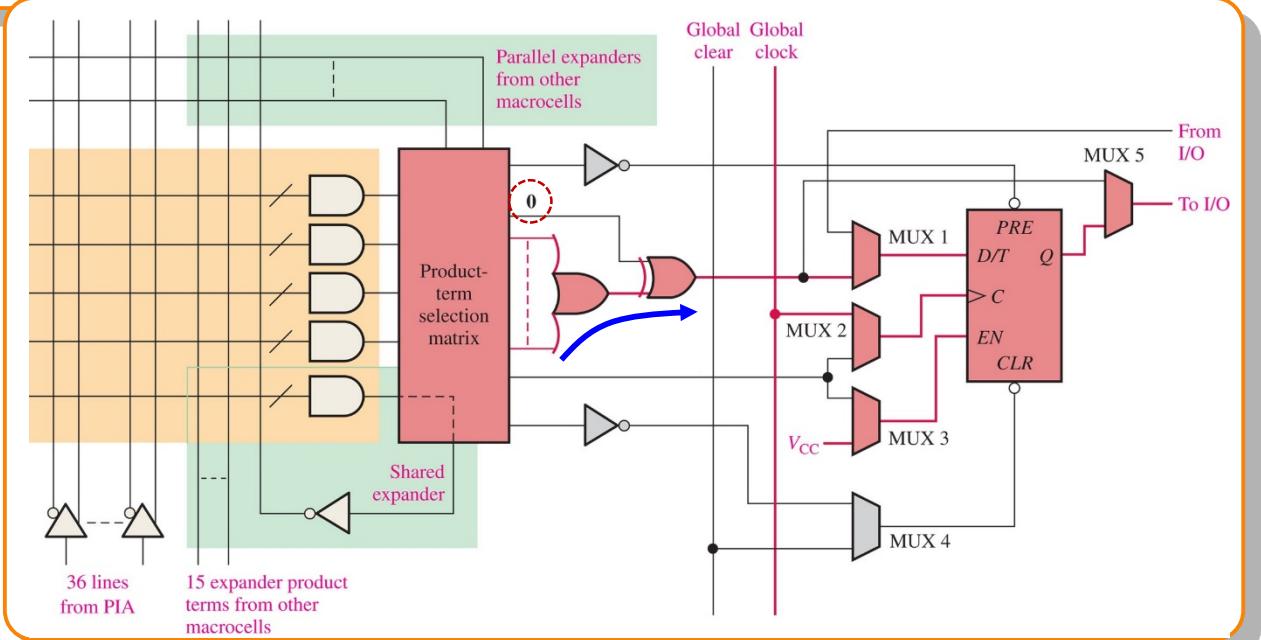


Combinational versus Registered Mode

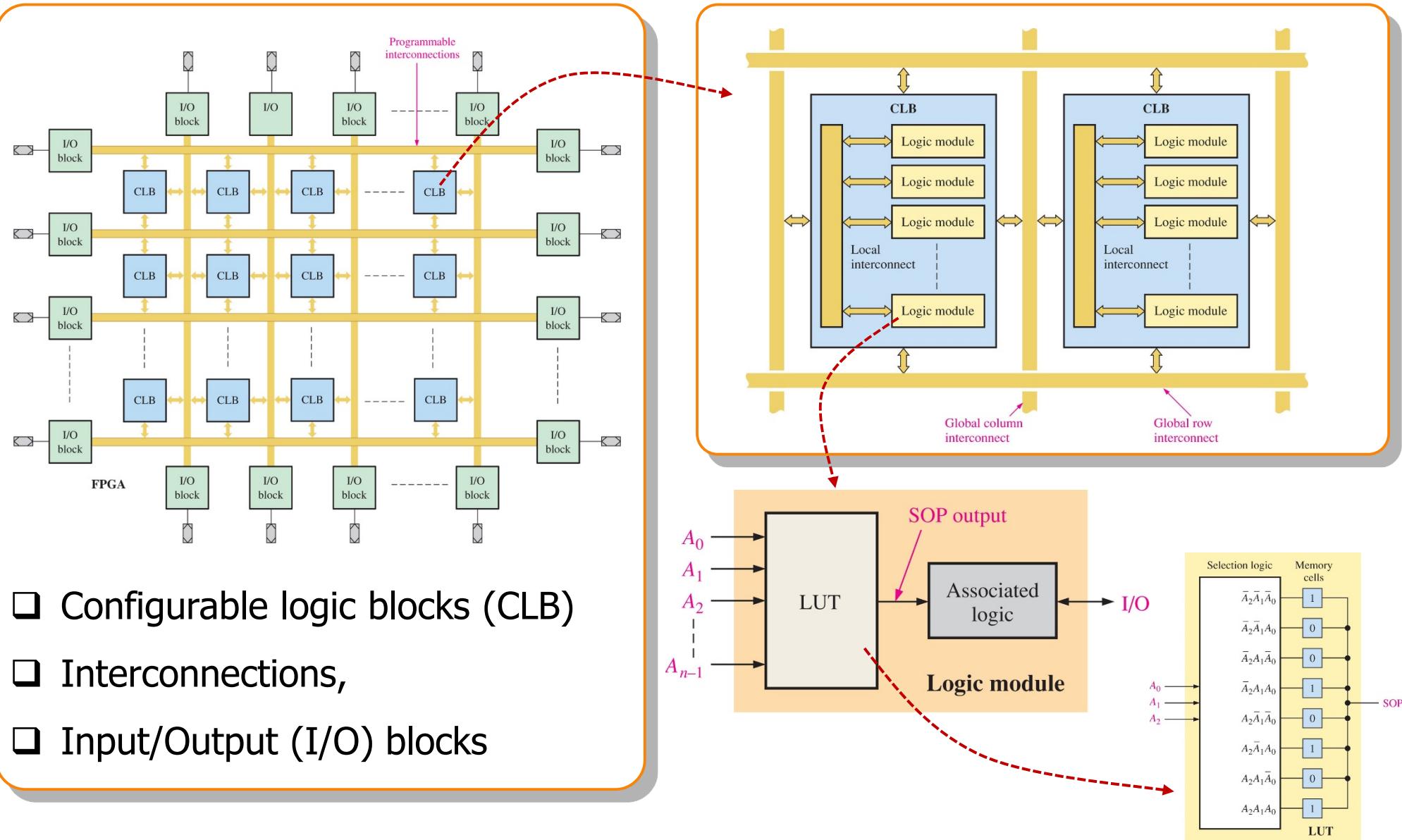


Combinational: Only MUX5 is used while the register is bypassed.

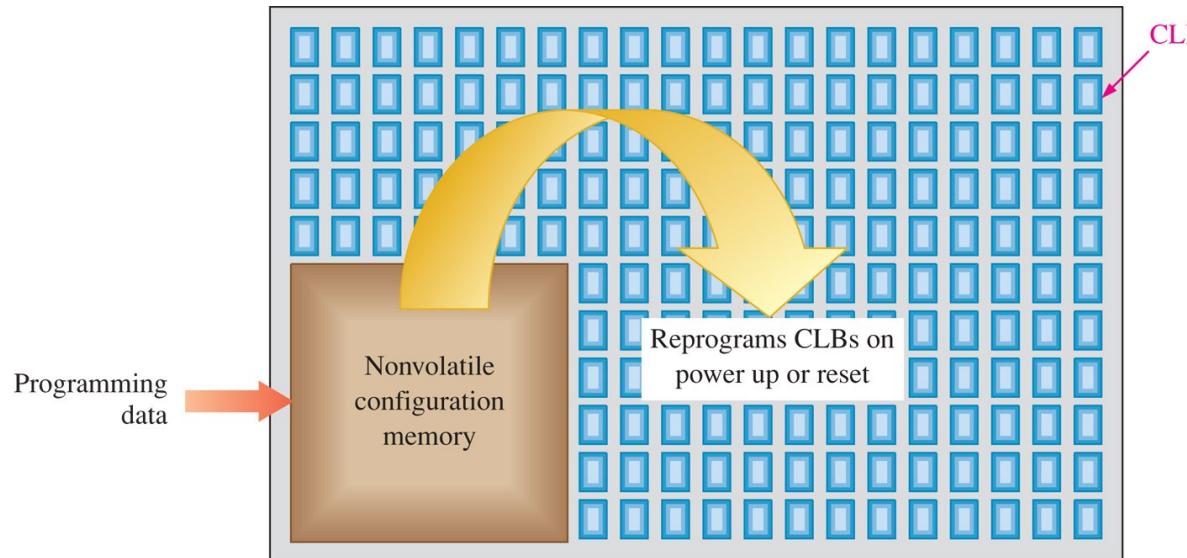
Registered: Four MUX's are used while the register is active.



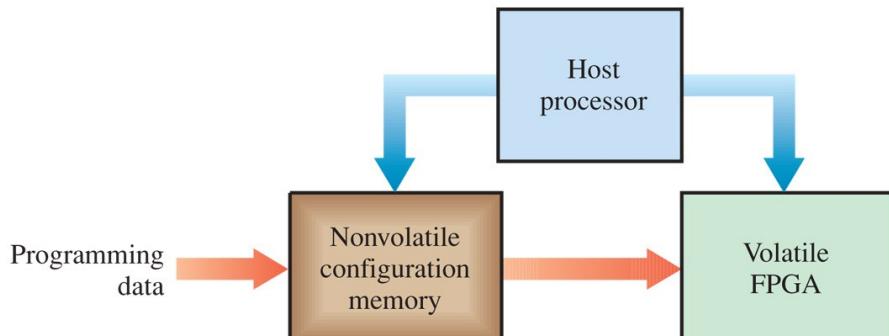
Field-Programmable Gate Arrays (FPGAs)



SRAM-Based FPGAs



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

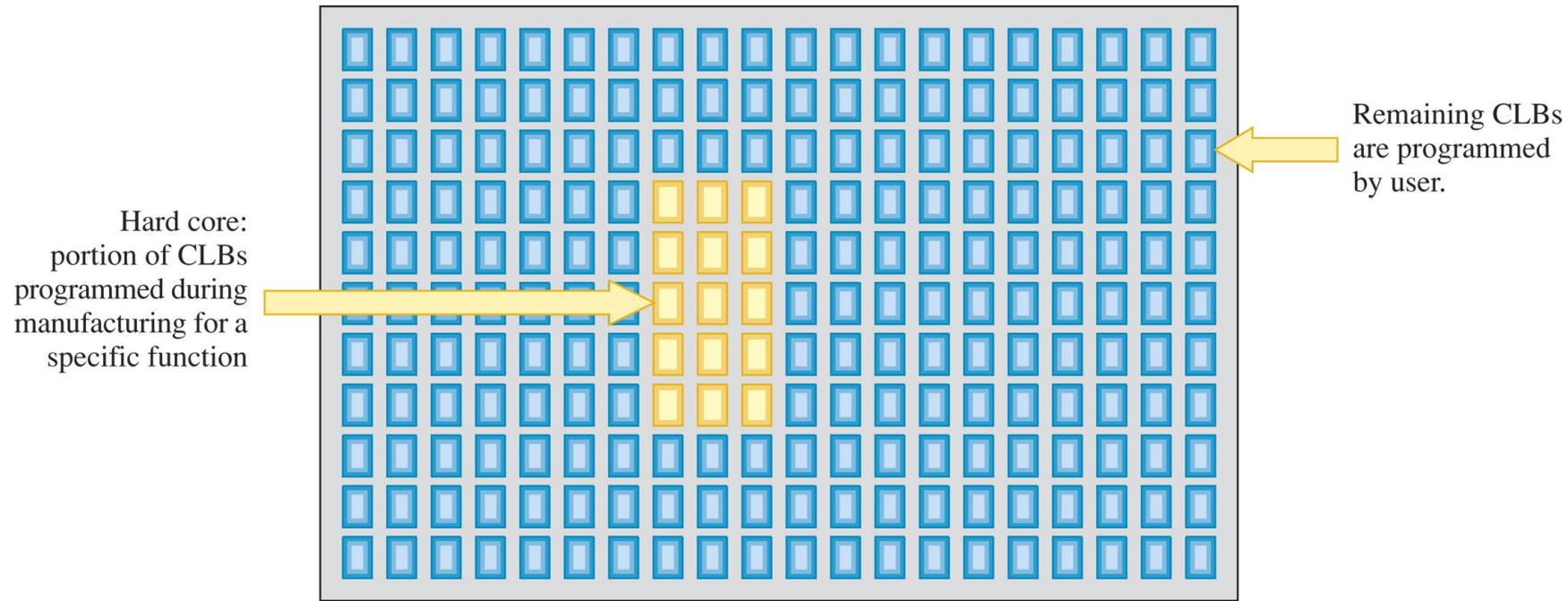


(b) Volatile FPGA with on-board memory and host processor

- On-the-chip nonvolatile configuration memory: store the program data and reconfigure the device each time power is turned back on **OR**
- External memory with data transfer controlled by a host processor.



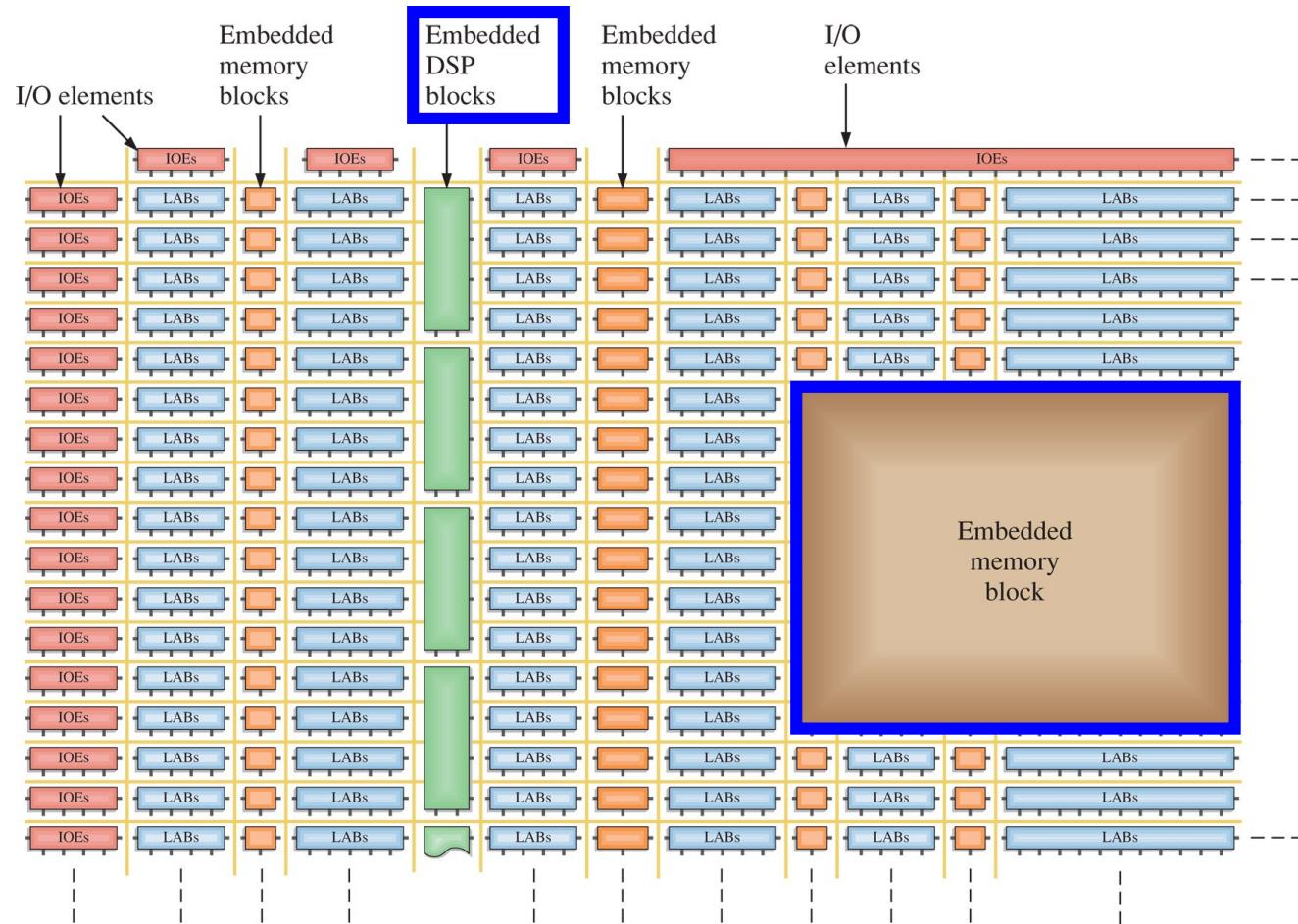
FPGA Cores



- While end users can program an FPGA for any logic design, a portion of logic in an FPGA (known as **hard cores**) is put in by the manufacturer to provide a specific function and that cannot be reprogrammed.
- Hard cores implement functions commonly used in a microprocessor, standard input/output interfaces, and digital signal processors.



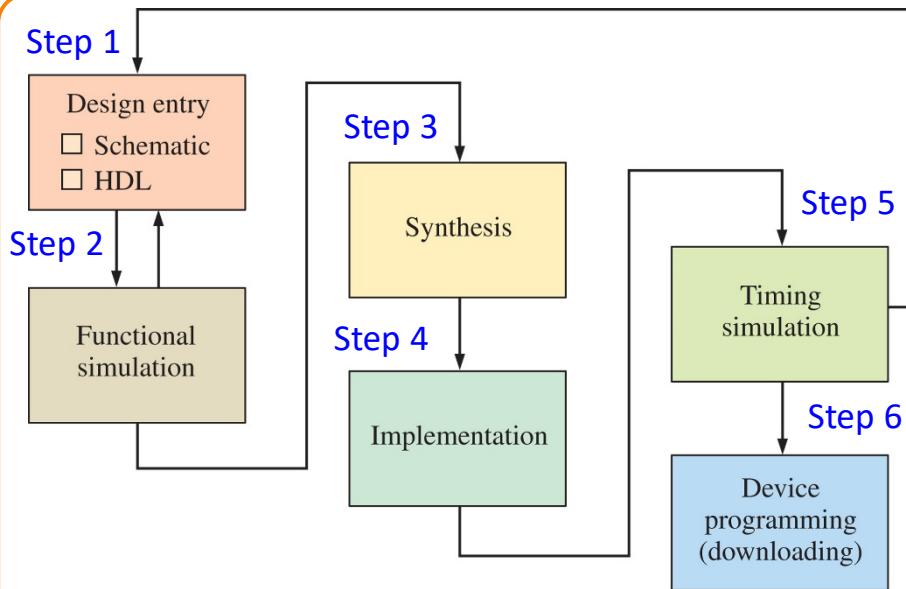
Embedded Functions



- Typical FPGAs contain embedded memory functions and digital signal processing (DSP) functions
- DSP functions, such as digital filters, are commonly used in many systems



Programmable Logic Software



General design flow diagram for programming a SPLD, CPLD, or FPGA.

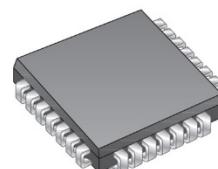
- ❑ 4 things for programming an FPGA
 - ◆ A computer
 - ◆ An FPGA
 - ◆ Development software
 - ◆ Connection btw the FPGA & the computer: programming fixture or a development board



(a) Computer



(b) Software (CD or Website download)



(c) Device



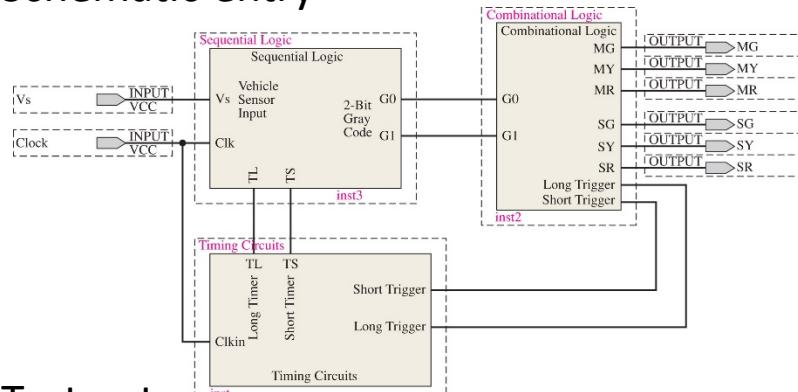
(d) Programming hardware (programming fixture or development board with cable for connection to computer port)



Design flow (I)

Step 1: Design Entry

Schematic entry

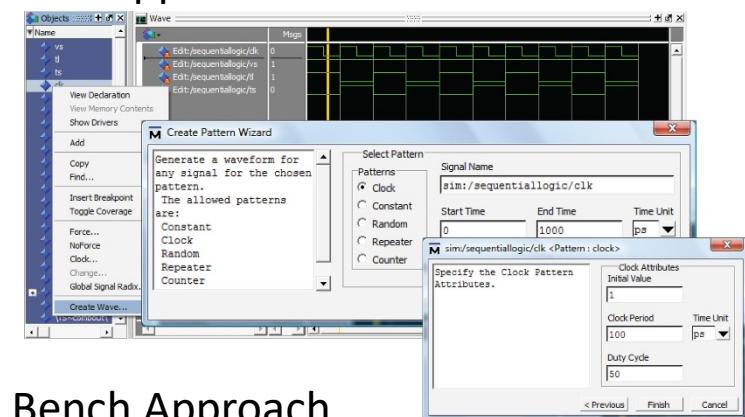


Text entry

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity SequentialLogic is
5 port(VS, TL, TS, Clk: in std_logic;
6      G0, G1: inout std_logic);
7 end entity SequentialLogic;
8
9 architecture SequenceBehavior of SequentialLogic is
10 component dff is
11 port(D,Clk: in std_logic;
12      Q: out std_logic);
13 end component dff;
14
15 signal D0, D1: std_logic;
16 begin
17   D1 <= (G0 and not TS) or
18         (G1 and TS);
19
20   D0 <= (not G1 and not TL and VS) or
21         (not G1 and G0) or
22         (G0 and TL and VS);
23
24 DFF0: dff port map(D => D0, Clk => Clk, Q => G0);
25 DFF1: dff port map(D => D1, Clk => Clk, Q => G1);
26 end architecture SequenceBehavior;
```

Step 2: Functional Simulation

Graphical Approach



Test Bench Approach

A programmatic approach to design simulation is to create an additional program file called a test bench.

Netlist(SequentialLogic)

```
Net<name>: instance<name>,<from>,<to>;
Instances: and0,and1,and2,and3,and4,or0,or1,inv0,inv1,inv2,DFF0,DFF1;
Input/outputs:11,I2,I3,I4,O1,O2
net1: DFF0, import2; DFF1, import2; I1;
net2: and0, import2; inv1, outport1; I2;
net3: inv0, outport1; and4, import2; I3;
net4: and2, import3, and4, import4; I4;
net5: and2, import2;
```

Step 3: Synthesis

The design is optimized by minimizing the # of gates, replacing logic elements with more efficient logic elements and eliminating any redundant logic.

The final output from the synthesis phase is a **netlist**.



Design flow (II)

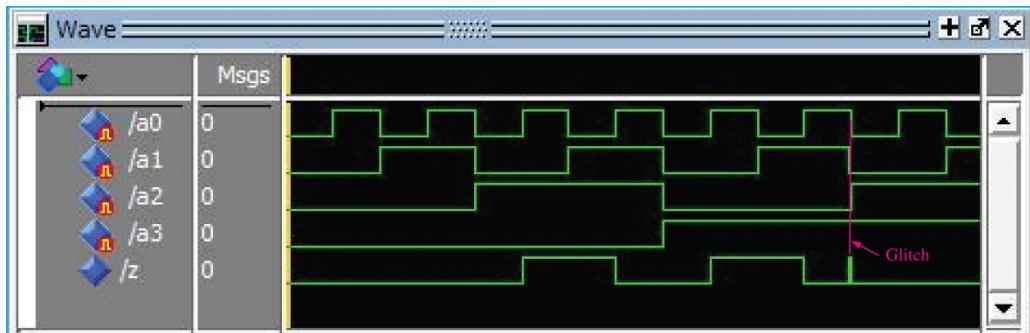
Step 4: Implementation (Software)

A compiler implements the netlist design by “mapping” the design to fit in the specific target device based on its architecture and pin configurations.

This process is called fitting or place and routing

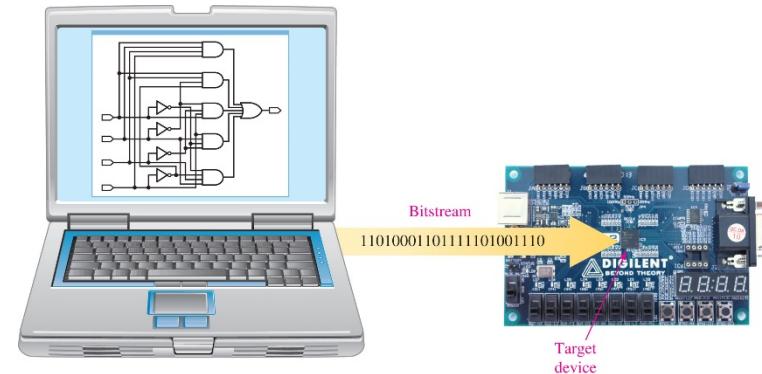
Step 5: Timing Simulation

To verify that the circuit works at the design frequency and that there are no timing problems that will affect the overall operation.



Step 6: Device Programming (Downloading)

A bitstream representing the final design is generated and sent to the target device; Upon completion, the design is actually in hardware and can be tested in-circuit.



Chapter Review

- Simple Programmable Logic Devices (SPLDs)
 - ◆ PAL, GAL, Macrocells
- Complex Programmable Logic Devices (CPLDs)
 - ◆ Class versus LUT architectures
 - ◆ Shared expander
- Macrocell Modes
 - ◆ Combinational versus Registered
- Field-Programmable Gate Arrays (FPGAs)
 - ◆ 3 components: Configurable logic blocks (CLB); Interconnections; Input/Output (I/O) blocks
 - ◆ SRAM-based FPGAs, FPGA cores
- Programmable Logic software
 - ◆ 6-step design flow: Design entry → Functional simulation → Synthesis --> Implementation (software) → Timing simulation → Downloading

Netlist



True/False Quiz

- A PAL consists of a programmable array of OR gates connected to a fixed array of AND gates.
- SPLD stands for simple programmable logic device.
- Typically, a macrocell consists of an AND gate and its associated output logic.
- CPLD stands for complex programmable logic device.
- An FGPA is a field programmable gate array.
- A typical FPGA has a greater gate density than a CPLD.
- Logic array blocks are found in CPLDs.
- The process of programming a PLD is known as design flow.
- The device being programmed is called a target device.
- Two types of programmable design entry are schematic and HDL.



True/False Quiz

- A PAL consists of a programmable array of OR gates connected to a fixed array of AND gates.
- SPLD stands for simple programmable logic device.
- Typically, a macrocell consists of an AND gate and its associated output logic.
- CPLD stands for complex programmable logic device.
- An FPGA is a field programmable gate array.
- A typical FPGA has a greater gate density than a CPLD.
- Logic array blocks are found in CPLDs.
- The process of programming a PLD is known as design flow.
- The device being programmed is called a target device.
- Two types of programmable design entry are schematic and HDL.

