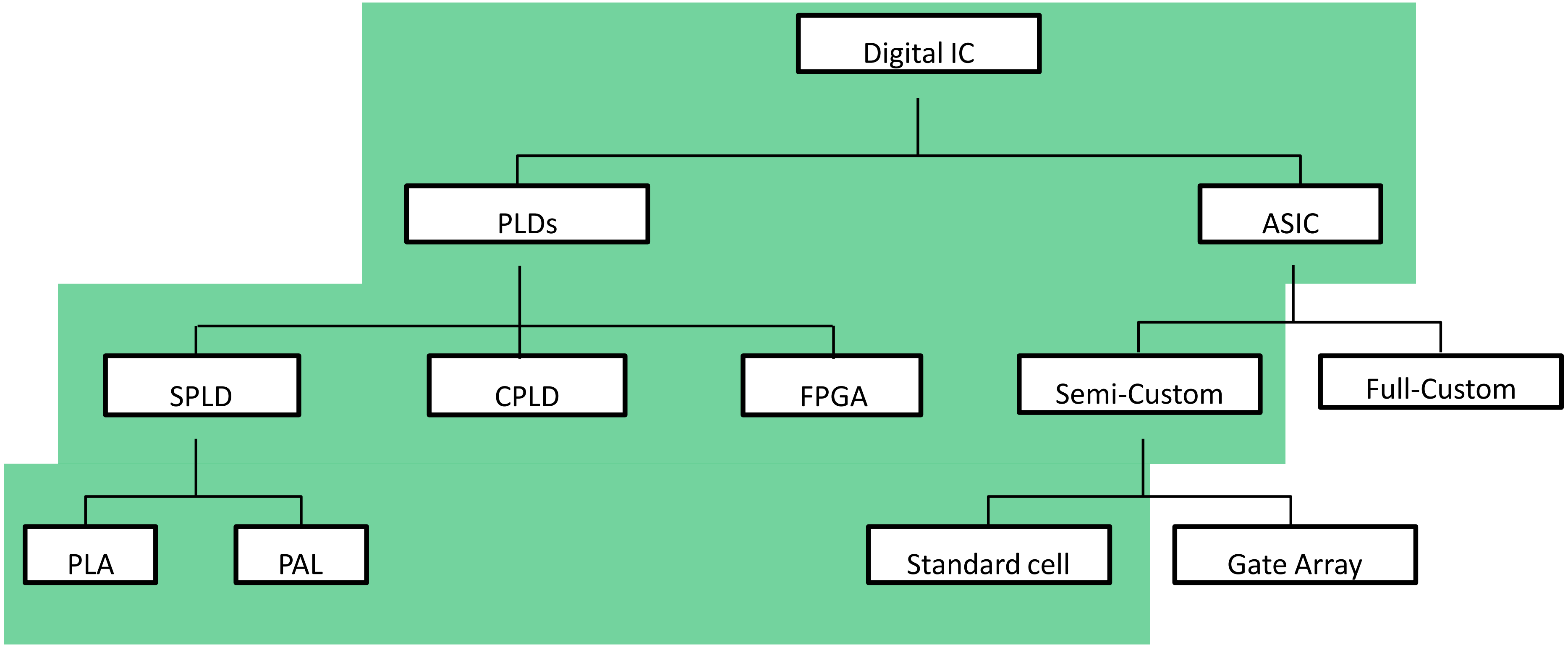
Chapter 2: Programmable Logic Devices

Outline

- Digital Systems Implementation Platforms
- Logic circuit design review
- Programmable Logic Devices (PLD)
    - SPLD (Simple PLD)
    - CPLD (Complex PLD)
    - FPGA (Field Programmable Gate Array)
- ASICs (Application-Specific Integrated Circuit)

Digital Systems Implementation Platforms

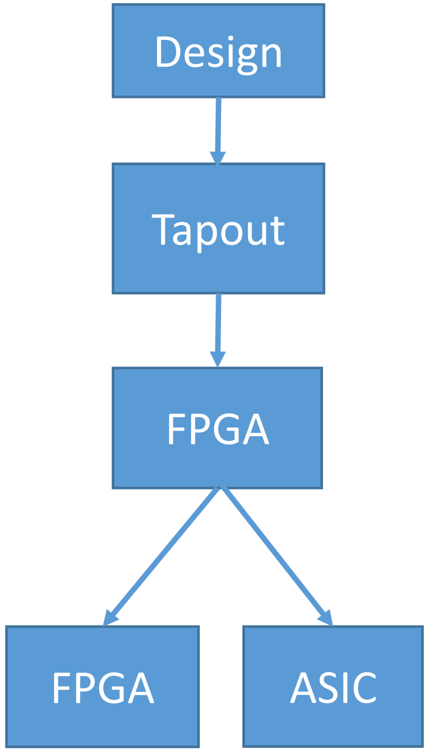


This course

FPGA vs. ASIC

❑ **Field Programmable Gate Array (FPGA) Advantages:**

- Fast programming and testing time by the end user (instant turn-around)
- Excellent for prototyping
  - Easy to migrate from prototype to the final design
- Can be re-used for other designs
- Cheaper (in small volumes) → lower start-up costs
- Re-programmable
- Lower financial risk
- Ease of design changes/modifications
- Cheaper design tools



Design

Tapout

FPGA

FPGA

ASIC

FPGA vs. ASIC

❑ **FPGA Drawbacks:**

- ➢Slower than ASIC (2-3 times slower)
- ➢Power hungry (up to 10 times more dynamic power)
- ➢Use more transistors per logic function
- ➢More area (20 to 35 times more area than a standard cell ASIC)

FPGA vs. ASIC

❑ Application Specific Integrated Circuit (ASIC) Advantages:

- ➢ Faster
- ➢ Lower power
- ➢ Cheaper (if manufactured in large volumes)
- ➢ Use less transistors per logic function

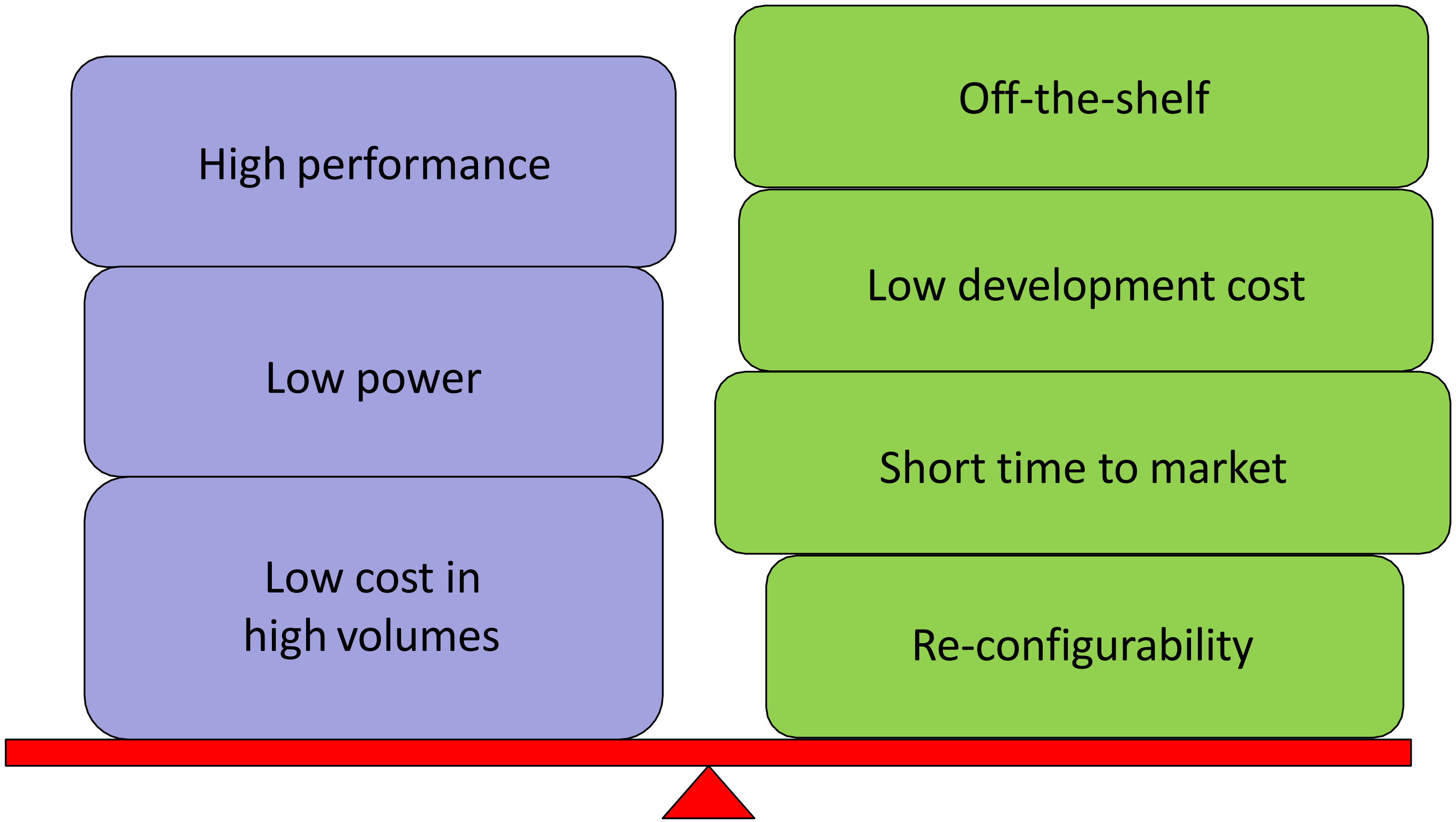
❑ ASIC Drawbacks:

- ➢ Implements a particular design (not programmable)
- ➢ Takes several months to fabricate (long turn-around)
- ➢ More expensive design tools
- ➢ Very expensive engineering/mask cost for the first successful design

Implementation Approaches (ASIC vs. FPGA)

ASICs

FPGAs



ASICs:

- High performance
- Low power
- Low cost in high volumes

FPGAs:

- Off-the-shelf
- Low development cost
- Short time to market
- Re-configurability

Outline

- Digital Systems Implementation Platforms
- Logic circuit design review
- Programmable Logic Devices
  - SPLD
  - CPLD
  - FPGA
- ASICs

Logic circuit design review

- Sum of Product (SoP)
- To get the desired canonical SOP expression we will add the **min-terms** (product terms) for which the output is 1 (F=1).

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

A	B	F	Minterm
0	0	0	A’B’
0	1	1	A’B
1	0	1	AB’
1	1	1	AB

- F = A’B + AB’ + AB

- Product of Sums (POS)
- To get the desired canonical POS expression we will multiply the **max-terms** (sum terms) for which the output is 0 (F=0).

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

A	B	F	Maxterm
0	0	0	A+B
0	1	1	A+B’
1	0	1	A’+B
1	1	0	A’+B’

- F = (A+B) . (A’+B’)

Outline

- Digital Systems Implementation Platforms
- Logic circuit design review
- Programmable Logic Devices
    - SPLD (Simple PLD)
    - CPLD (Complex PLD)
    - FPGA (Field Programmable Gate Array)
- ASICs (Application Specific Integrated Circuits)

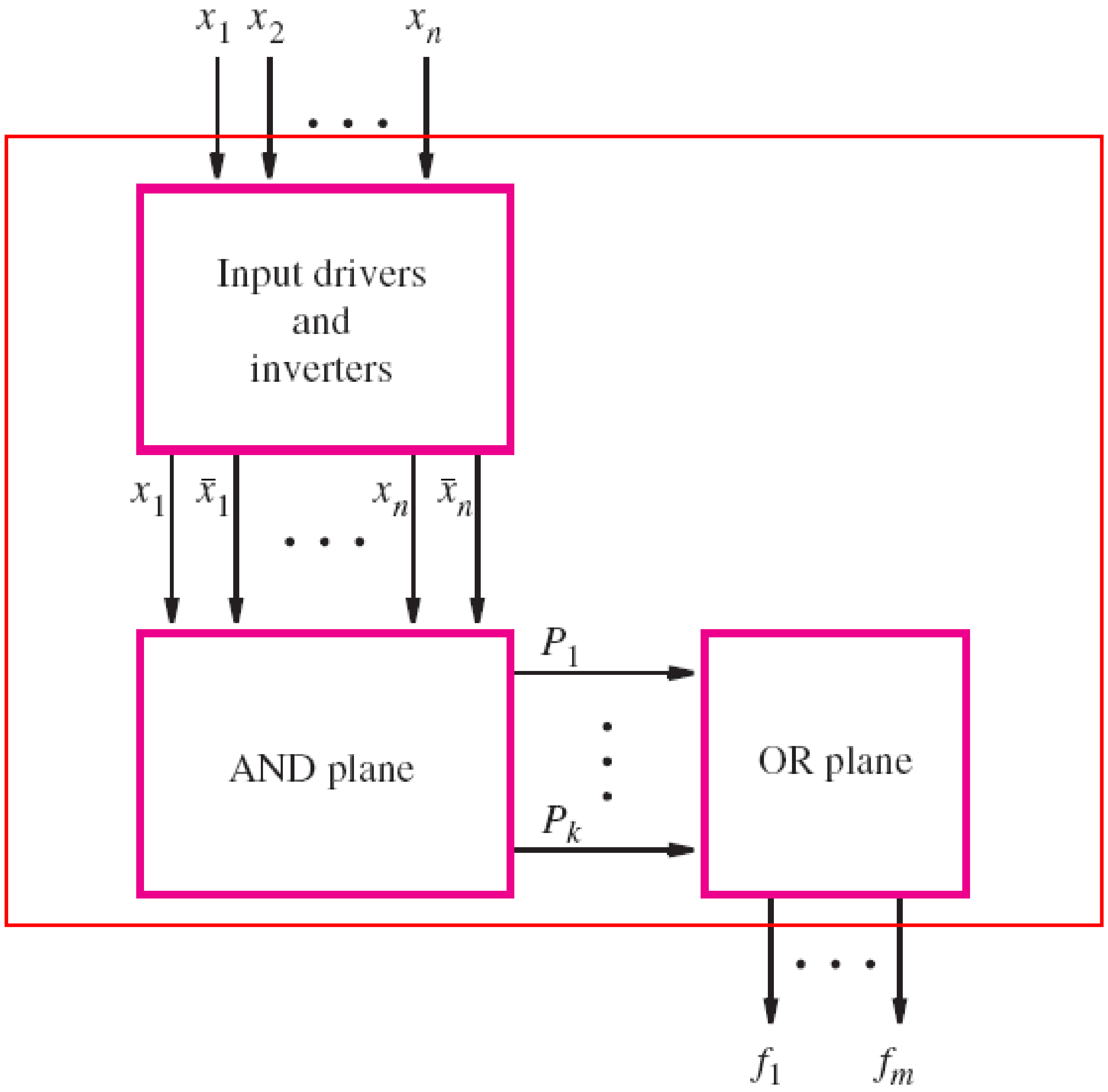
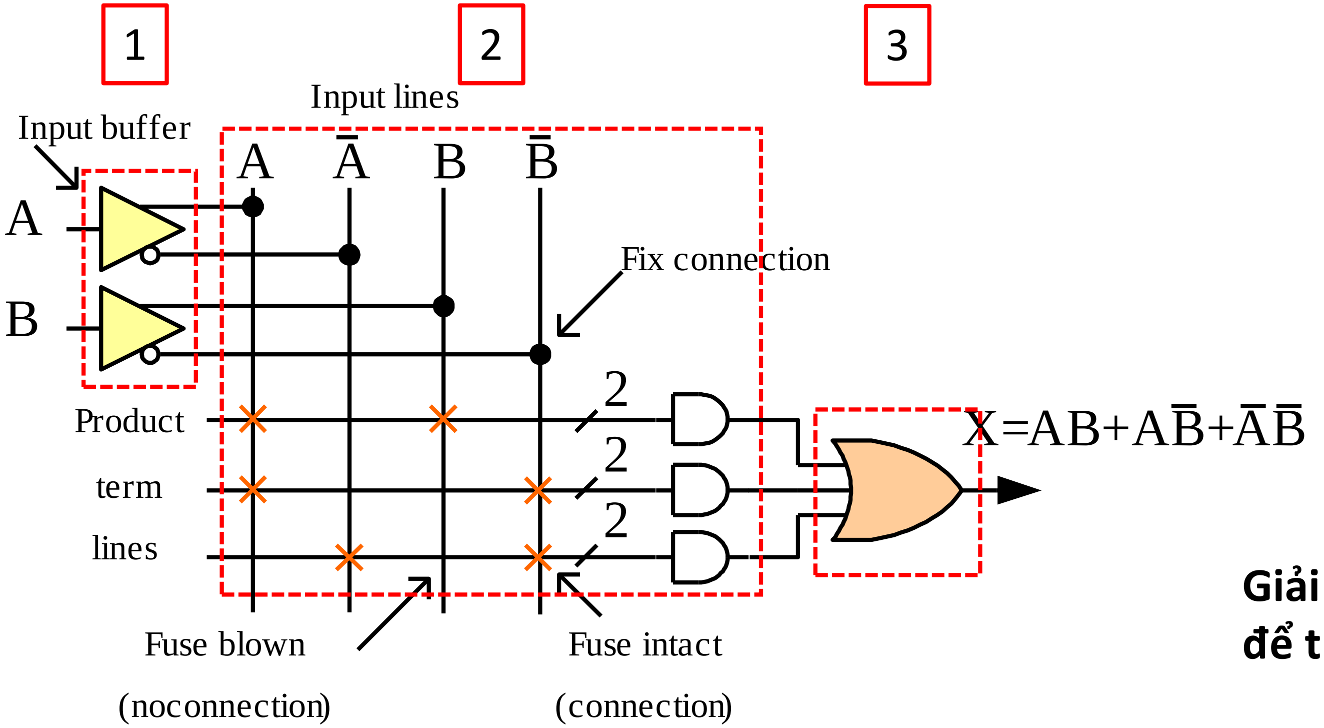

x_1 x_2 x_n
Input drivers and inverters
x_1 $\bar{x}_1$ x_n $\bar{x}_n$
AND plane
P_1
P_k
OR plane
f_1 f_m

Figure B.25 General structure of a PLA.

Simple Programmable Logic Device (SPLD)

- Programmable Array Logic (PAL): blowing an array of fuses
- Generic Array Logic (GAL): EEPROM arrays to store these connections rather than fuses=erasurable

- PAL, GAL can be used to configure the **Sum of Product** logic circuit.



A line = a bus

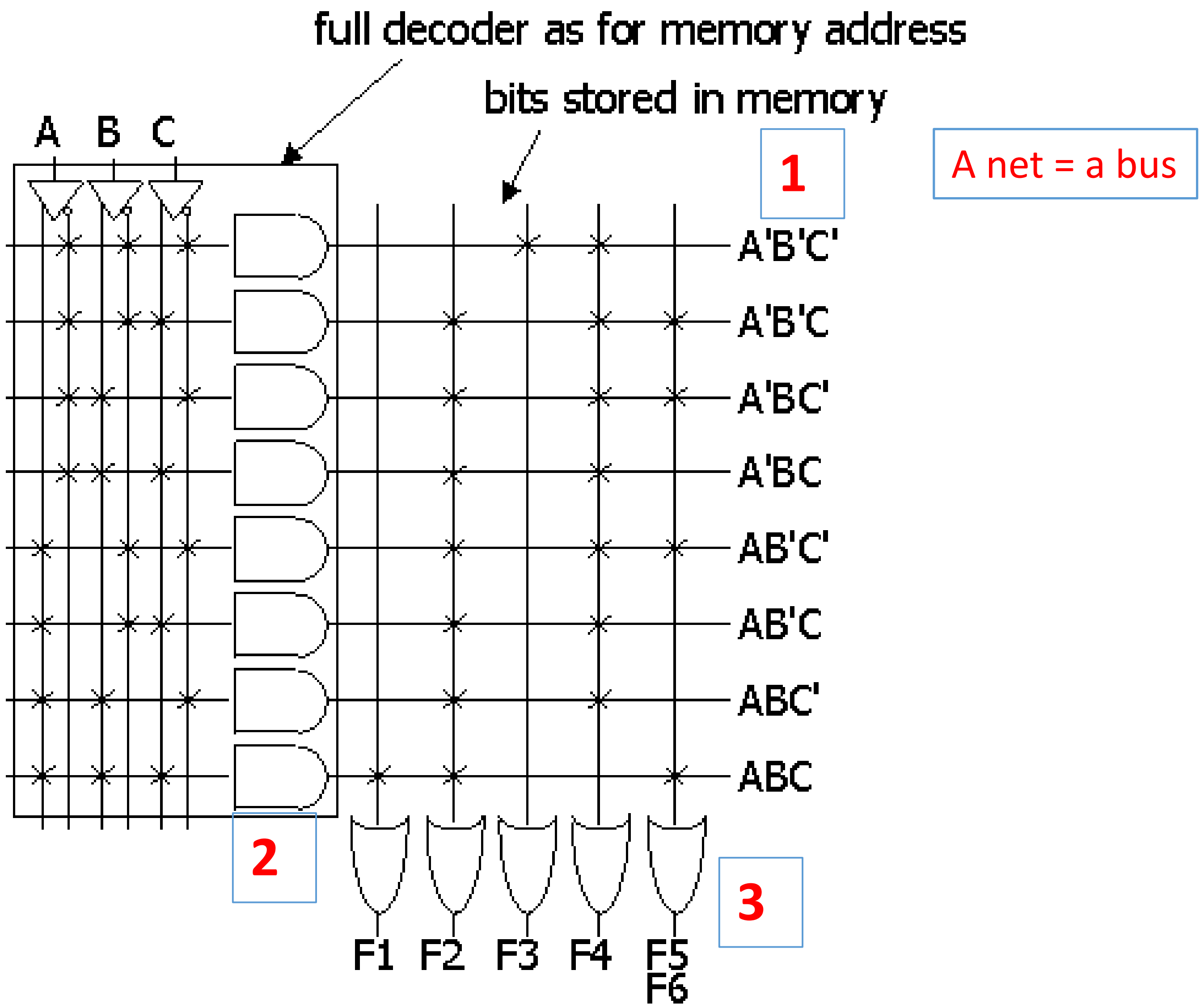
1: Input drivers and inverters
2: Switch matrix and AND plane
3: OR plan

Giải thích nguyên lý hoạt động của SPLD như hình để thực thi hàm X=AB + AB’ + A’B’?

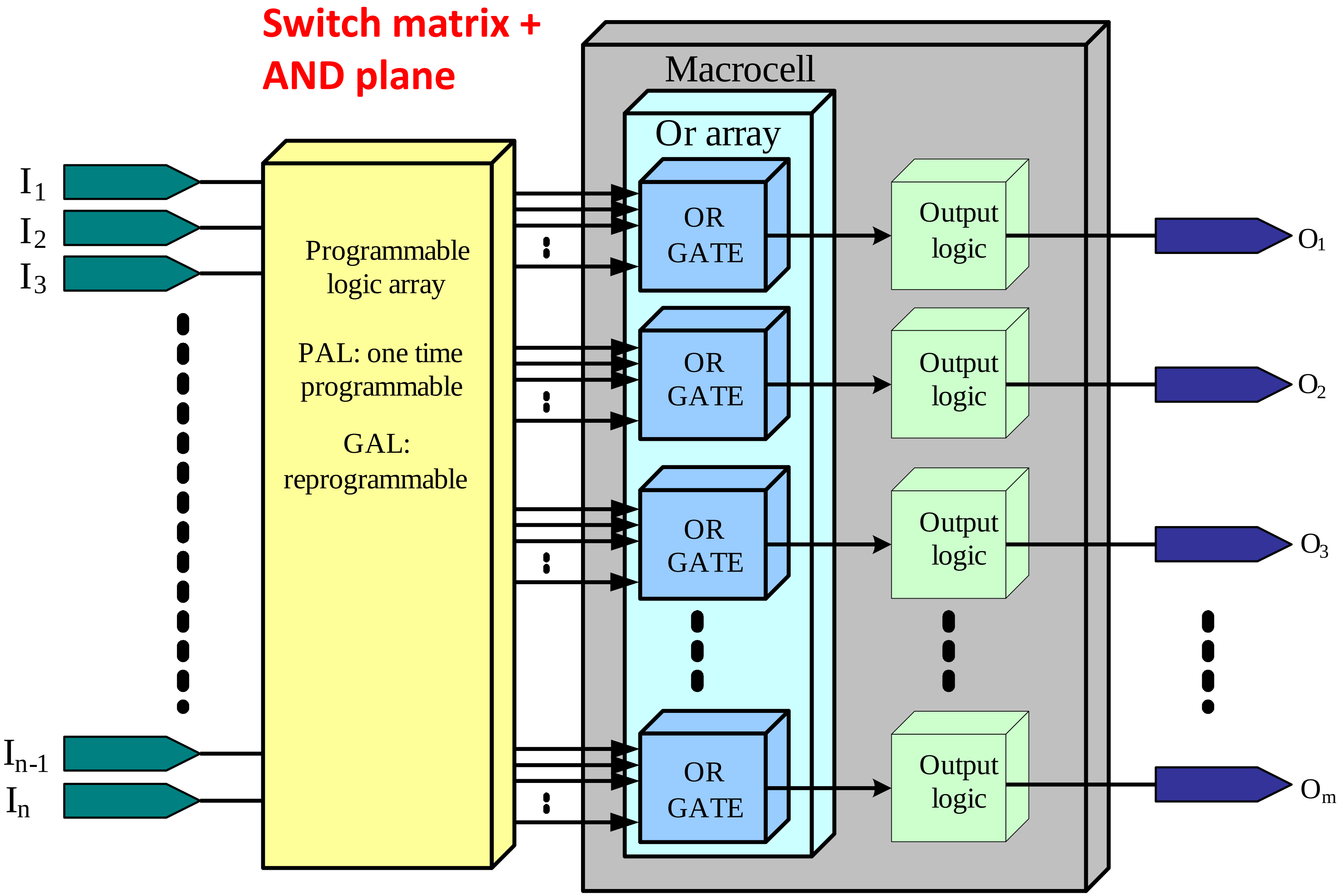
Example of PAL configuration

- Multiple functions of A, B, C
  - F1 = A B C
  - F2 = A + B + C
  - F3 = A' B' C'
  - F4 = A' + B' + C'
  - F5 = A xor B xor C
  - F6 = A xnor B xnor C

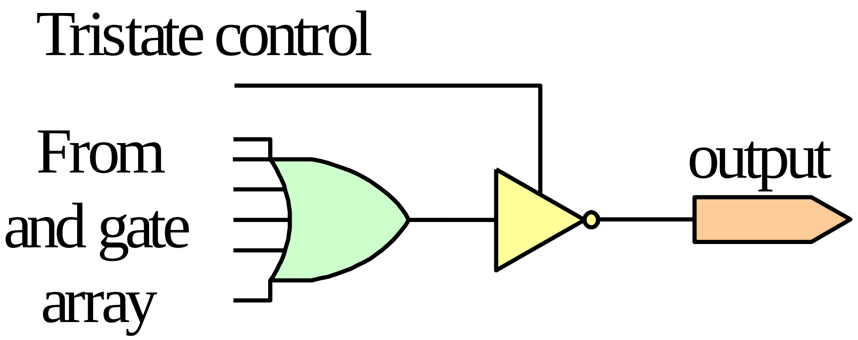
A	B	C	F1	F2	F3	F4	F5	F6
0	0	0	0	0	1	1	0	0
0	0	1	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1
0	1	1	0	1	0	1	0	0
1	0	0	0	1	0	1	1	1
1	0	1	0	1	0	1	0	0
1	1	0	0	1	0	1	0	0
1	1	1	1	1	0	0	1	1



PAL /GAL


Switch matrix +
AND plane
Macrocell
Or array
I_1
I_2
I_3
I_{n-1}
I_n
Programmable logic array
PAL: one time programmable
GAL: reprogrammable
OR GATE
OR GATE
OR GATE
OR GATE
Output logic
Output logic
Output logic
Output logic
O_1
O_2
O_3
O_m

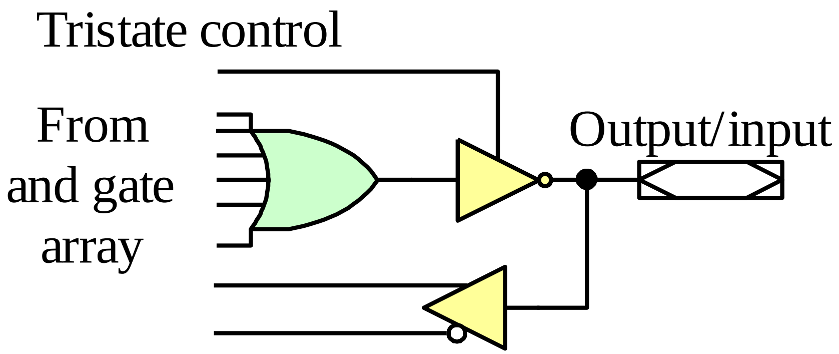
Macrocell



Tristate control

From and gate array

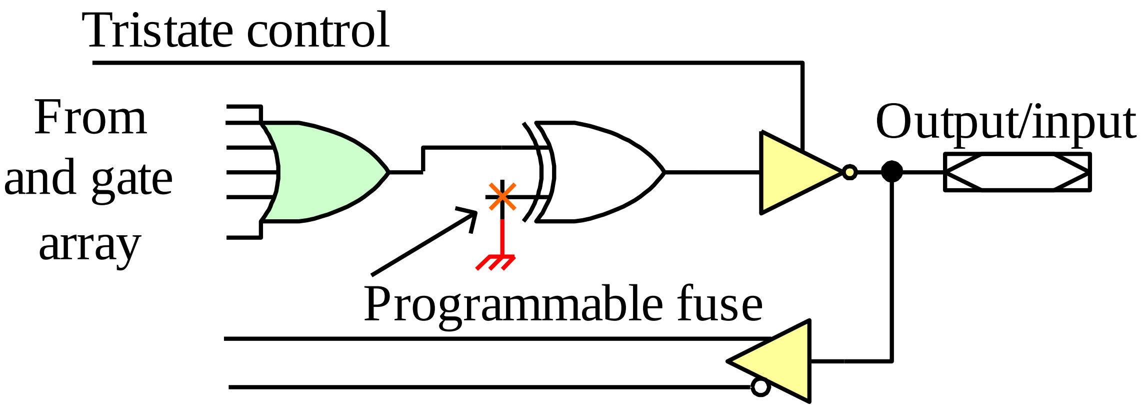
output



Tristate control

From and gate array

Output/input



Tristate control

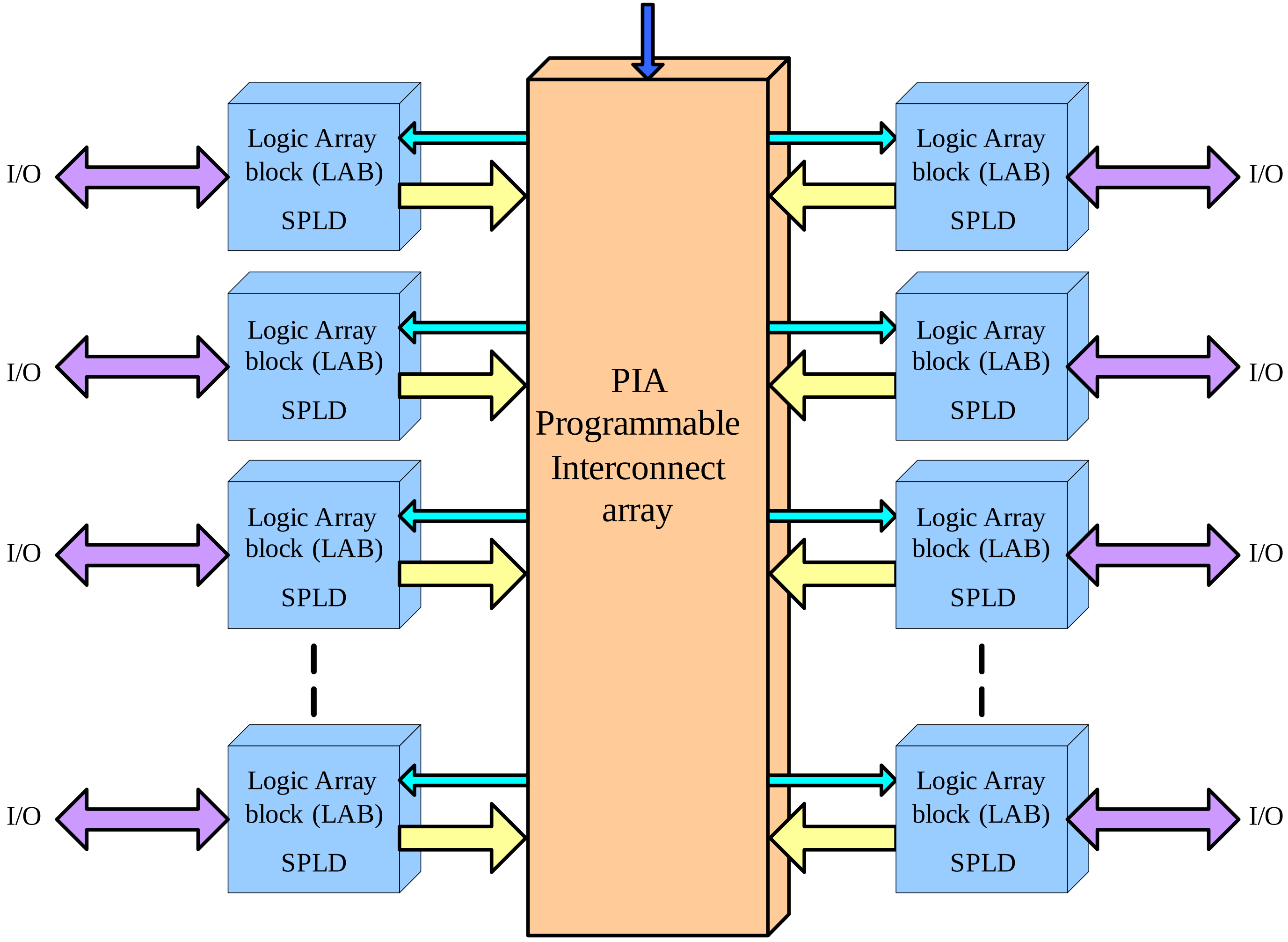
From and gate array

Output/input

Programmable fuse

Complex Programmable Logic Device (CPLD)

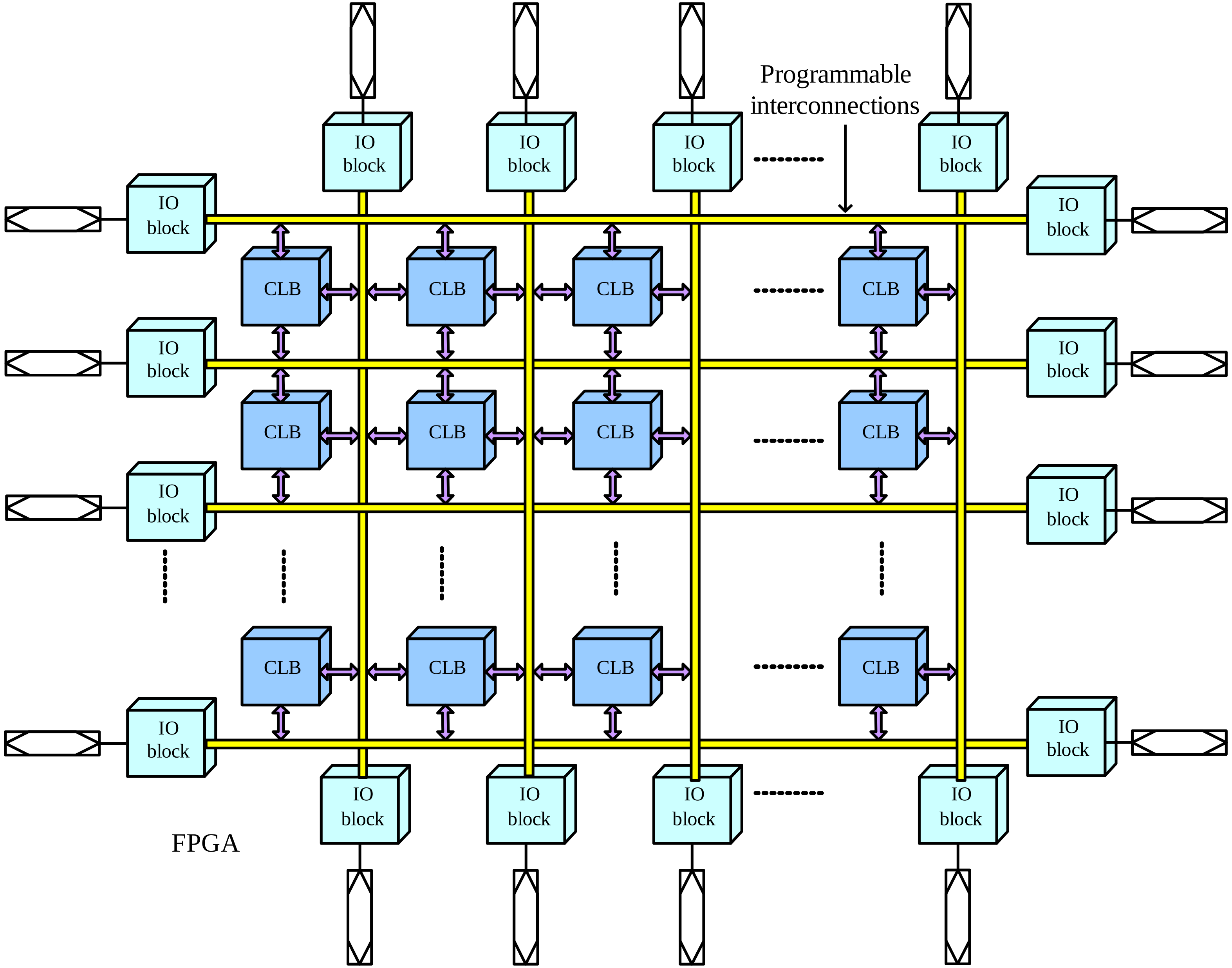
- CPLD is consist of SPLDs



I/O
Logic Array block (LAB) SPLD
I/O
Logic Array block (LAB) SPLD
I/O
Logic Array block (LAB) SPLD
I/O
Logic Array block (LAB) SPLD
PIA Programmable Interconnect array
Logic Array block (LAB) SPLD
I/O
Logic Array block (LAB) SPLD
I/O
Logic Array block (LAB) SPLD
I/O
Logic Array block (LAB) SPLD
I/O

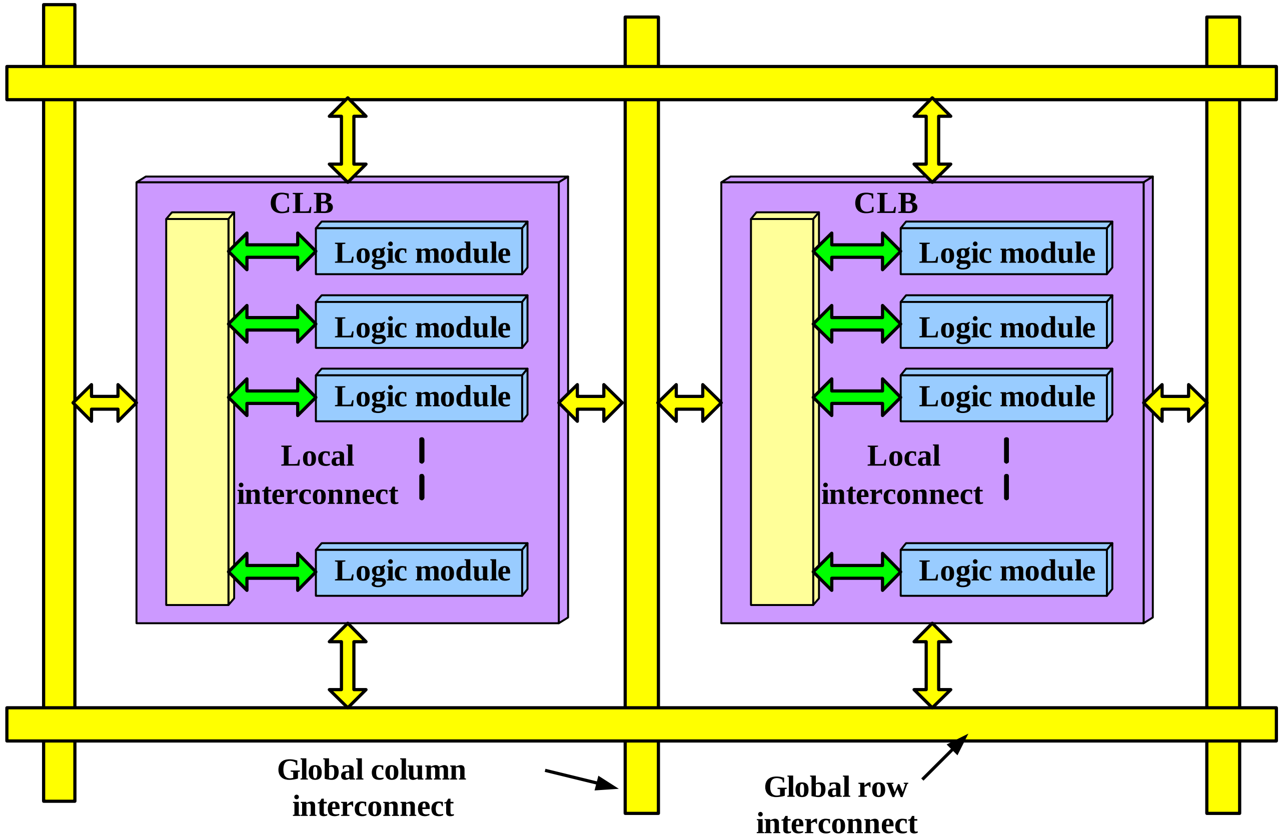
Switch matrix

Field Programmable Gate Array (FPGA)

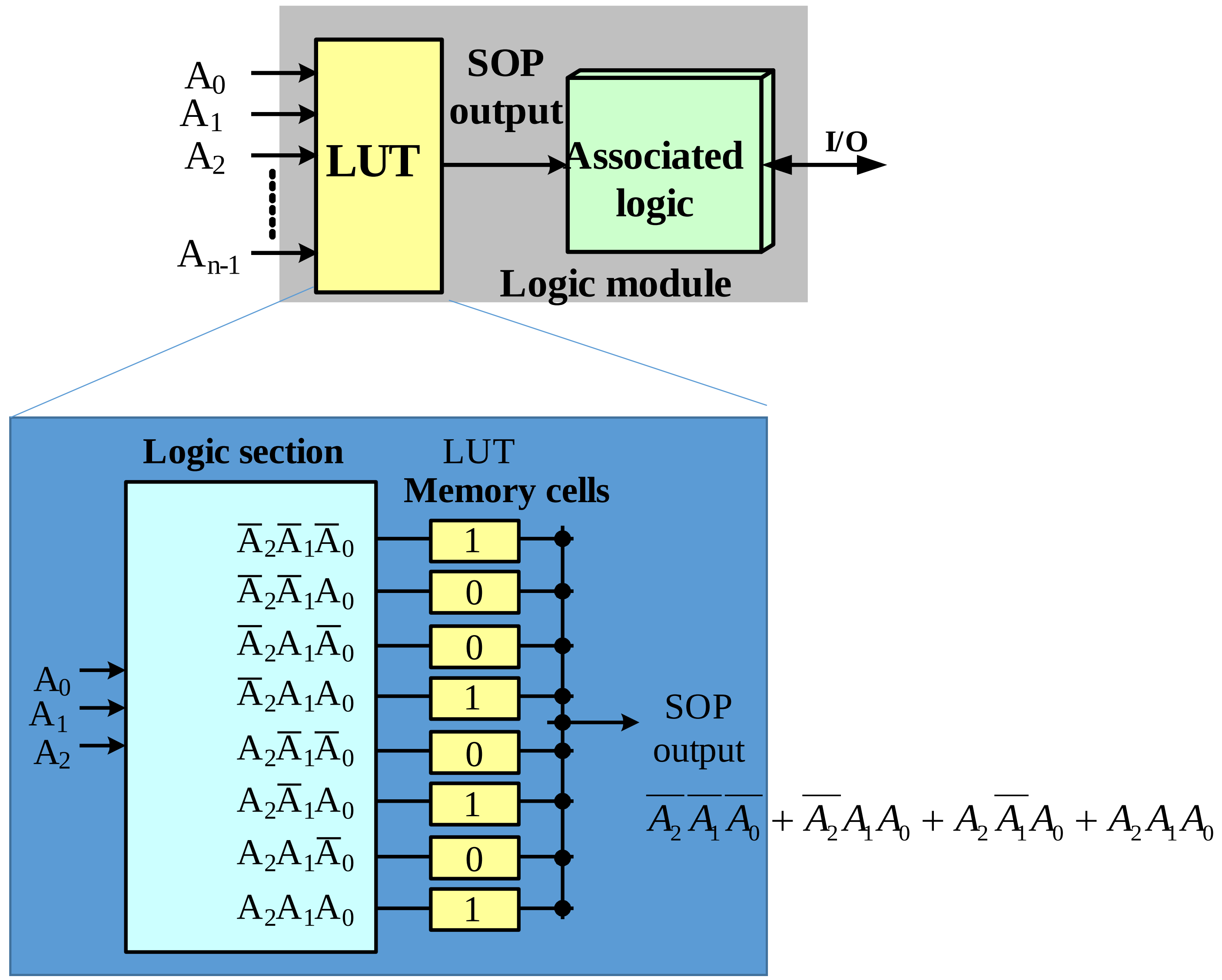


Configurable Logic Block (CLB)

CLB


CLB
Logic module
Logic module
Logic module
Local interconnect
Logic module
CLB
Logic module
Logic module
Logic module
Local interconnect
Logic module
Global column interconnect
Global row interconnect

Logic Module – Look-Up Table (LUT)

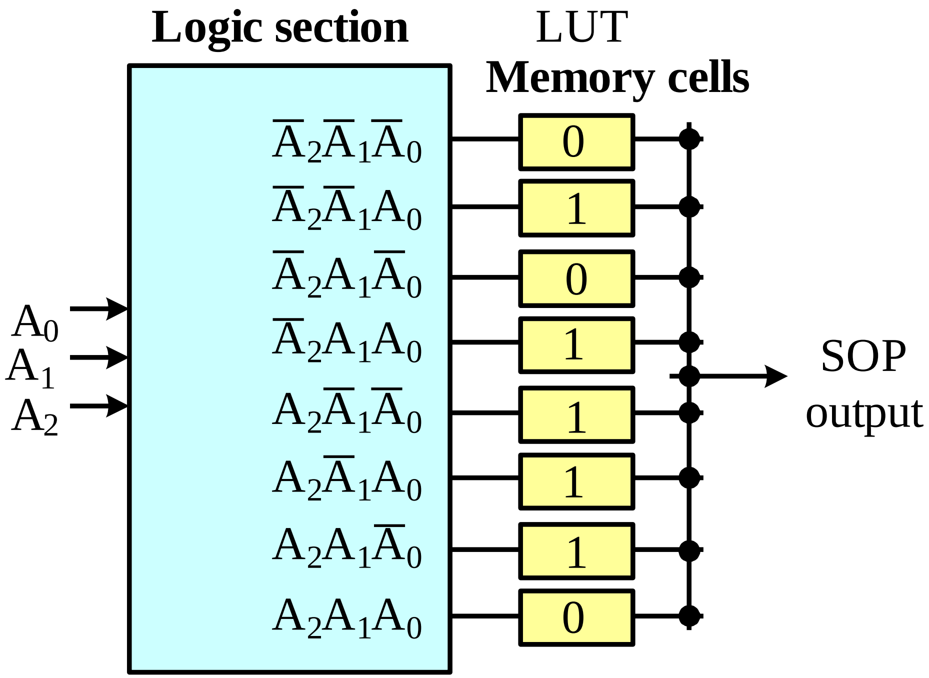


$$\overline{A}_2\overline{A}_1\overline{A}_0 + \overline{A}_2A_1A_0 + A_2\overline{A}_1A_0 + A_2A_1A_0$$

LUT

LUT

$$A_2 A_1 \overline{A_0} + A_2 \overline{A_1}\,\overline{A_0} + \overline{A_2} A_1 A_0 + A_2 \overline{A_1} A_0 + \overline{A_2}\,\overline{A_1} A_0$$


Logic section
LUT
Memory cells
$\overline{A}_2\overline{A}_1\overline{A}_0$
$\overline{A}_2\overline{A}_1A_0$
$\overline{A}_2A_1\overline{A}_0$
$\overline{A}_2A_1A_0$
$A_2\overline{A}_1\overline{A}_0$
$A_2\overline{A}_1A_0$
$A_2A_1\overline{A}_0$
$A_2A_1A_0$
0
1
0
1
1
1
1
0
A_0
A_1
A_2
SOP
output

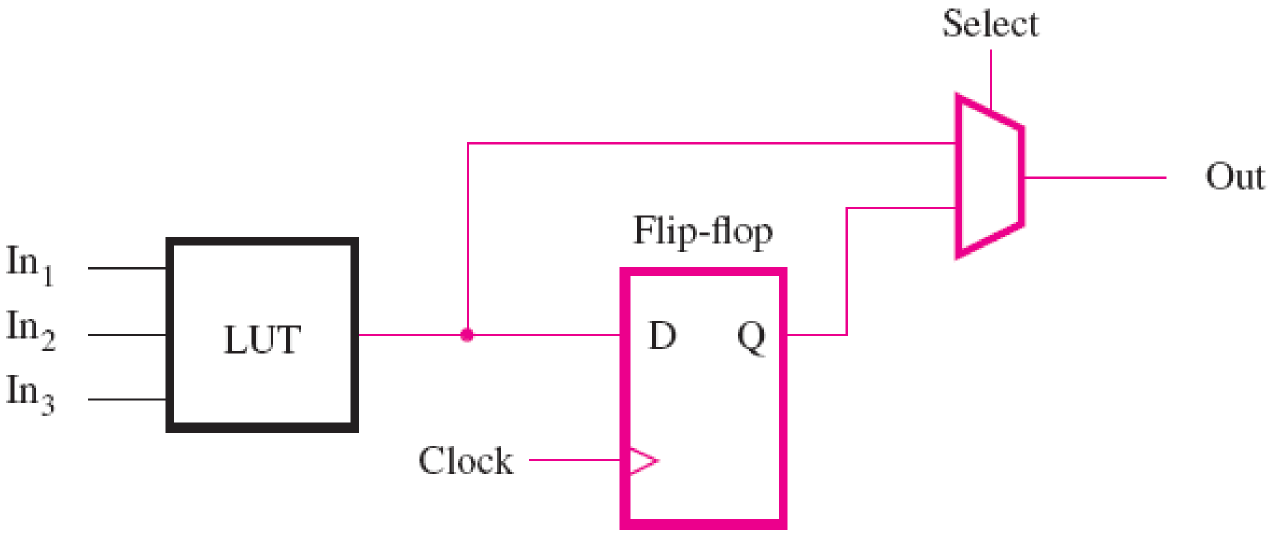


Figure B.38 Inclusion of a flip-flop in an FPGA logic element.

Outline

- Digital Systems Implementation Platforms
- Logic circuit design review
- Programmable Logic Devices
  - SPLD
  - CPLD
  - FPGA
- ASICs

ASICs (Application-Specific Integrated Circuits)

- **Full-Custom ASICs:** custom chip
  - Some (possible all) logic cells are customized and all mask layers are customized
  - Complete flexibility to decide the size of the chip, the number of transistors the chip contains, the placement of each transistor on the chip, and the way the transistors are connected together → chip layout process
  - **Ex: Microprocessors and memory chips**
- **Semi-custom ASICs:** Standard-cell based and Gate-array-based ASICs
  - Does not need complete flexibility for the layout of each individual transistor in a custom chip
  - **All logic cells are predesigned** (defined in cell library) and some (possibly all) of the mask layers are customized

https://www.xilinx.com/video/hardware/what-is-the-difference-between-an-fpga-and-an-asic.html

No.	FPGA	ASIC
1	Reconfigurable circuit. FPGAs can be reconfigured with a different design. They even have capability to reconfigure a part of chip while remaining areas of chip are still working! This feature is widely used in accelerated computing in data centres.	Permanent circuitry. Once the application specific circuit is taped-out into silicon, it cannot be changed. The circuit will work same for its complete operating life.
2	Design is specified generally using hardware description languages (HDL) such as VHDL or Verilog.	Same as for FPGA. Design is specified using HDL such as Verilog, VHDL etc.
3	Easier entry-barrier. One can get started with FPGA development for as low as USD $30.	Very high entry-barrier in terms of cost, learning curve, liaising with semiconductor foundry etc. Starting ASIC development from scratch can cost well into millions of dollars.
4	Not suited for very high-volume mass production.	Suited for very high-volume mass production.
5	Less energy efficient, requires more power for same function which ASIC can achieve at lower power.	Much more power efficient than FPGAs. Power consumption of ASICs can be very minutely controlled and optimized.
6	Limited in operating frequency compared to ASIC of similar process node. The routing and configurable logic eat up timing margin in FPGAs.	ASIC fabricated using the same process node can run at much higher frequency than FPGAs since its circuit is optimized for its specific function.

No.	FPGA	ASIC
7	Analog designs are not possible with FPGAs. Although FPGAs may contain specific analog hardware such as PLLs, ADC etc, they are not much flexible to create for example RF transceivers.	ASICs can have complete analog circuitry, for example WiFi transceiver, on the same die along with microprocessor cores. This is the advantage which FPGAs lack.
8	FPGAs are highly suited for applications such as Radars, Cell Phone Base Stations etc where the current design might need to be upgraded to use better algorithm or to a better design. In these applications, the high-cost of FPGAs is not the deciding factor. Instead, programmability is the deciding factor.	ASICs are definitely not suited for application areas where the design might need to be upgraded frequently or once-in-a-while.
9	Preferred for prototyping and validating a design or concept. Many ASICs are prototyped using FPGAs themselves! Major processor manufacturers themselves use FPGAs to validate their System-on-Chips (SoCs). It is easier to make sure design is working correctly as intended using FPGA prototyping.	It is not recommended to prototype a design using ASICs unless it has been absolutely validated. Once the silicon has been taped out, almost nothing can be done to fix a design bug (exceptions apply).
10	FPGA designers generally do not need to care for back-end design. Everything is handled by synthesis and routing tools which make sure the design works as described in the RTL code and meets timing. So, designers can focus into getting the RTL design done.	ASIC designers need to care for everything from RTL down to reset tree, clock tree, physical layout and routing, process node, manufacturing constraints (DFM), testing constraints (DFT) etc. Generally, each of the mentioned area is handled by different specialist person.