

# Nội dung môn học

1. Nhắc lại kiến thức điện tử số (1 LT + 0 BT)
2. Giới thiệu công nghệ IC lập trình được (1 LT + 0 BT)
3. Ngôn ngữ lập trình Verilog HDL (2 LT + 2 BT)
4. Thiết kế mạch số thông dụng (2 LT + 2 BT)
5. Công cụ là quy trình thiết kế với IC. lập trình được của Xilinx (2 LT + 2 BT)
6. Thực hành thiết kế trên FPGA (3 LT + 3 BT)
7. Kiểm tra 1 buổi + dự phòng 1 buổi

# **Giới thiệu công nghệ IC lập trình**

- Giới thiệu chung
- Phân loại IC khả trình
- Kiến trúc IC khả trình
- Quy trình thiết kế IC khả trình

# Giới thiệu chung

- PLD: **P**rogramable **L**ogic **D**evice - IC số khả trình
  - Chứa các cấu trúc mạch có quy luật
  - Gồm dãy (ma trận) các ô nhớ đồng dạng, cho phép truy xuất tới từng ô nhớ
  - Có thể lập trình để thực hiện các hàm logic khác nhau

- Phân loại

Phân loại theo mức độ tích hợp		
SSI - Small Scale Intergration	Mạch tích hợp cỡ nhỏ	$N < 10$
MSI - Medium Scale Intergration	Mạch tích hợp cỡ trung bình	$10 < N < 100$
LSI - Large Scale Intergration.	Mạch tích hợp cỡ lớn	$100 < N < 1000$
VLSI - Very Large Scale Intergration	Mạch tích hợp cỡ rất lớn	$> 10^3$

# Giới thiệu chung

- IC số truyền thống
  - Sử dụng cổng logic cơ bản
  - Phương pháp thiết kế logic truyền thống
  - Các IC số tiêu chuẩn: IC74xx họ TTL hoặc 40xx họ CMOS

## Đặc điểm

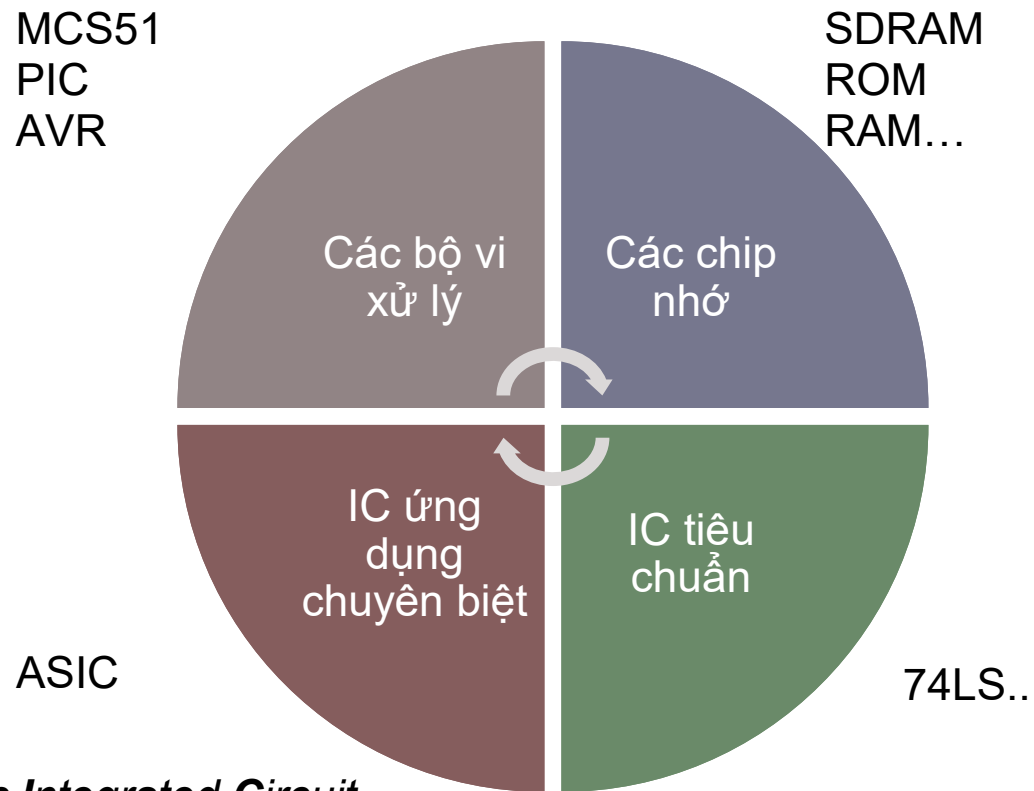
- Dễ dàng ghép nối IC chuẩn
- Thiết kế nhỏ và vừa
- Tốc độ làm việc thấp

- IC số khả trình
  - Gồm dãy hoặc ma trận các ô nhớ đồng dạng
  - Lập trình để thực hiện các hàm khác nhau
  - Chứa các cấu trúc mạch có quy luật

## Đặc điểm

- Tăng khả năng tích hợp
- Công suất thấp
- Dễ thay đổi

# Giới thiệu chung

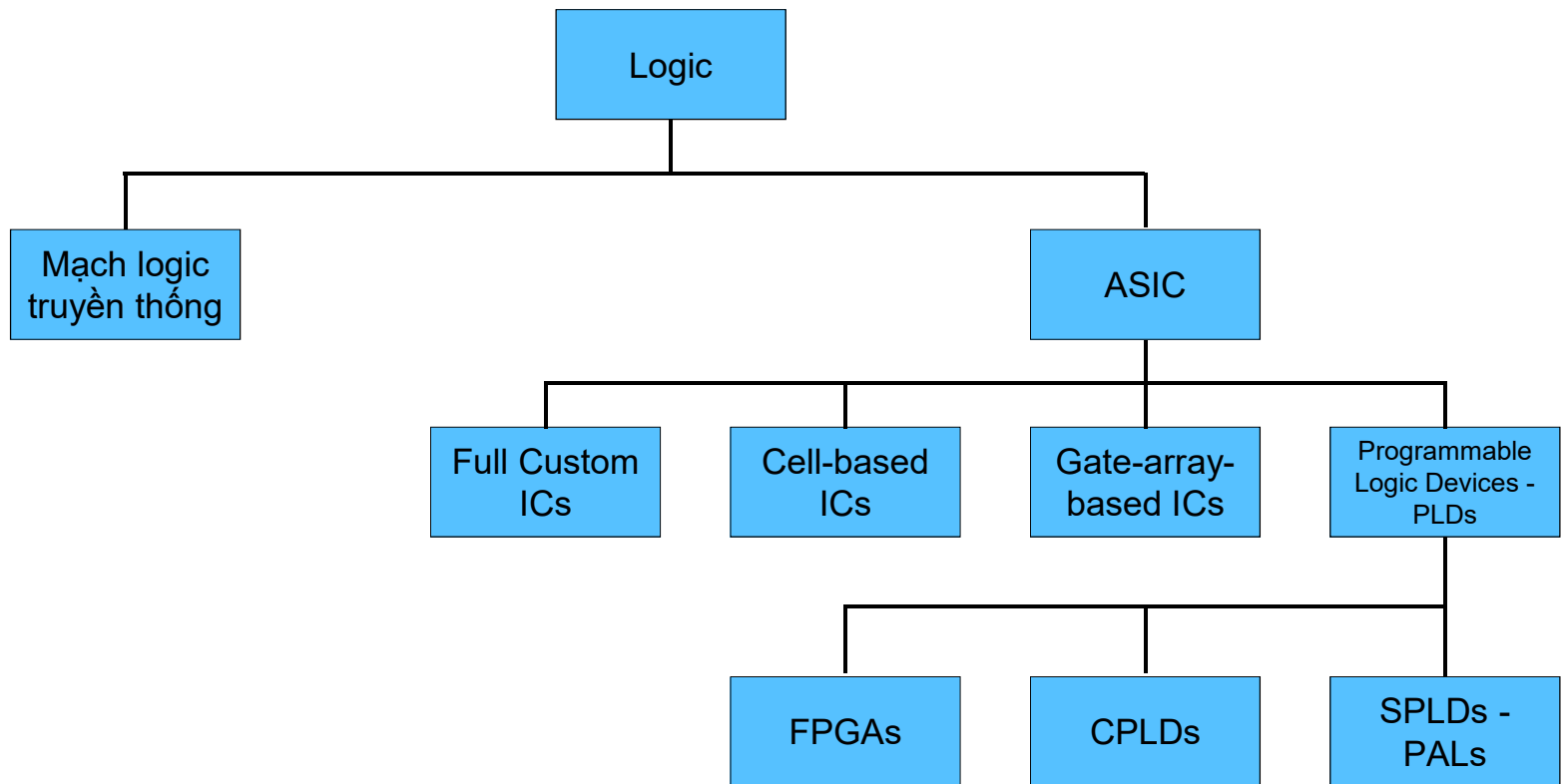


**ASIC** - *Application Specific Integrated Circuit*

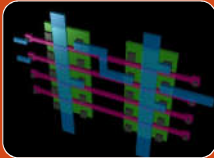
# Thuật ngữ ASIC

- Ra đời do nhu cầu hoạt động thiết kế vi mạch tăng cao
- ASIC = Application Specific Intergrated Circuit - IC chuyên dụng
- Lưu ý: Nếu một IC xuất hiện trong sách tra cứu thì đó không phải là ASIC
- Đặc điểm chung
  - Dùng trong các ứng dụng chuyên biệt
  - Chế tạo chuyên biệt
  - Hiệu quả hơn so với giải pháp dùng phần mềm trên bộ vi xử lý
  - Hầu hết các chip SoC (System-on-chip) là ASIC

# Phân loại IC số khả trình



# Full Custom ICs - ASIC đặc chế hoàn toàn



Logic cell và các mask layers được thiết kế riêng biệt



Mức bố trí Transistor và liên kết



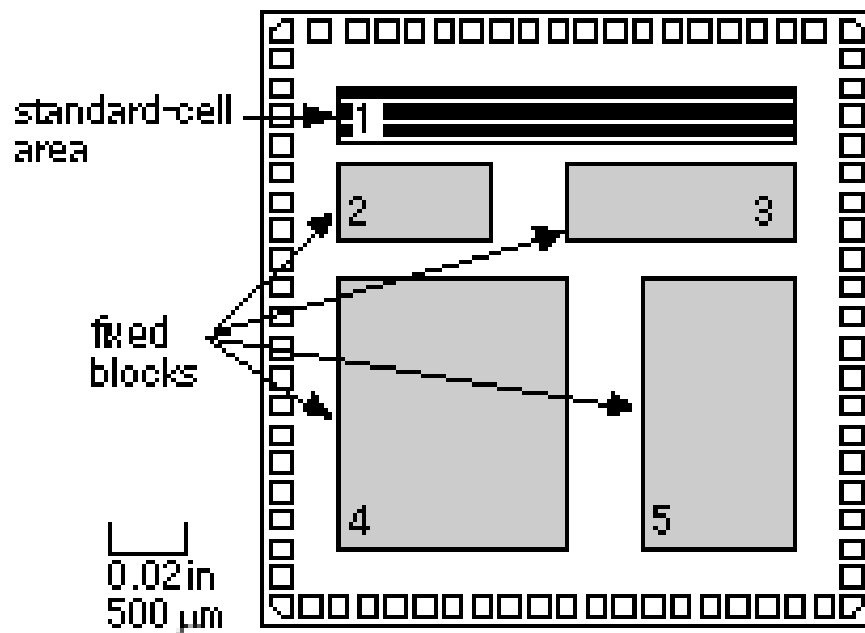
Cần thay đổi thiết kế khi công nghệ thay đổi

- Hiệu quả tối ưu
- Giá thành cao
- 8 tuần chế tạo

- Logic cell là một đơn vị linh kiện được sử dụng để lưu trữ thông tin 1 bit. VD 1 flip-flop
- Mask layers là các mạch hoặc nền tạo mạch điện trong IC, được sử dụng để tạo các kết nối giữa các logic cell



# Standard-cell-based-ICs - ASIC dựa trên các tế bào chuẩn

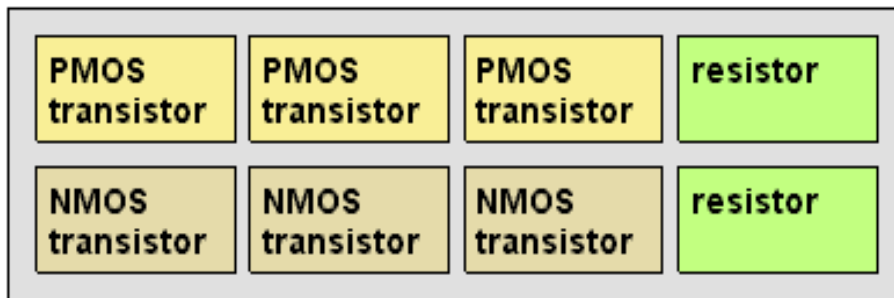


- Cấu trúc gồm các khối
  - Khối tế bào chuẩn (khối 1)
  - Khối cố định (khối 2,3,4 và 5)
- Các khối chứa các hàng của các tế bào chuẩn
- Thiết kế gần giống với các viên gạch xây tường
- Ưu: tiết kiệm thời gian, giảm rủi ro
- Nhược: Chi phí mua thư viện cell, cần thời gian chế tạo mask layer

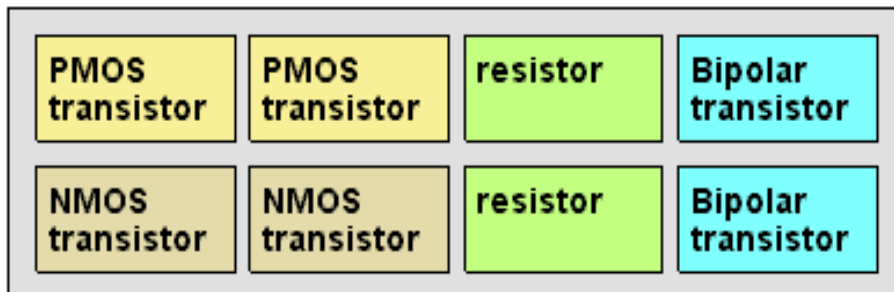
# Gate-array-based-ICs - ASIC dựa trên mảng công logic

From Computer Desktop Encyclopedia  
© 2004 The Computer Language Co., Inc.

CMOS Basic Cell



CMOS and Bipolar Basic Cell (BiCMOS)



- Cấu trúc gồm các mảng các cổng logic giống hệt nhau (mỗi cell có thể là các transistor và điện trở)
- Dùng phần mềm của hãng để kết nối các khối - dựa vào các thư viện cell và các macro
- Trong thư viện có thể có sẵn một số mask tùy biến

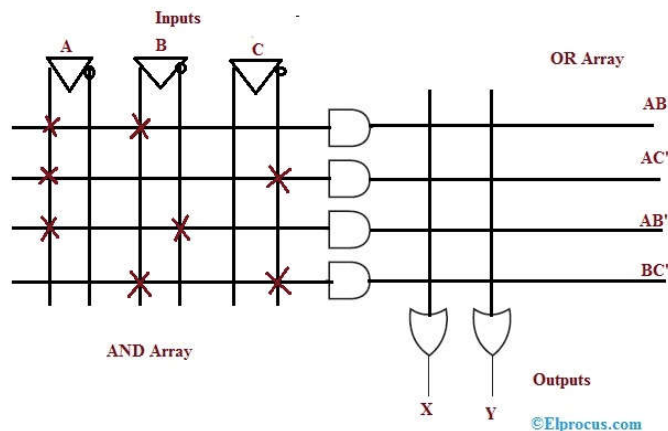
# Programmable Logic Device - PLD - các vi mạch lập trình

- là mạch kỹ thuật số có thể cấu hình lại bằng phần mềm
- Tại thời điểm sản xuất, mạch PLD mới chỉ gồm các thành phần logic, chưa được liên kết với nhau nên chưa thể hoạt động
- Trước khi sử dụng, PLD sẽ được lập trình bằng phần mềm chuyên biệt
- Phân loại
  - Simple PLD - SPLD
  - Complex PLD - CPLD
  - Field Programmable Gate Array - FPGA

# Simple PLD - SPLD

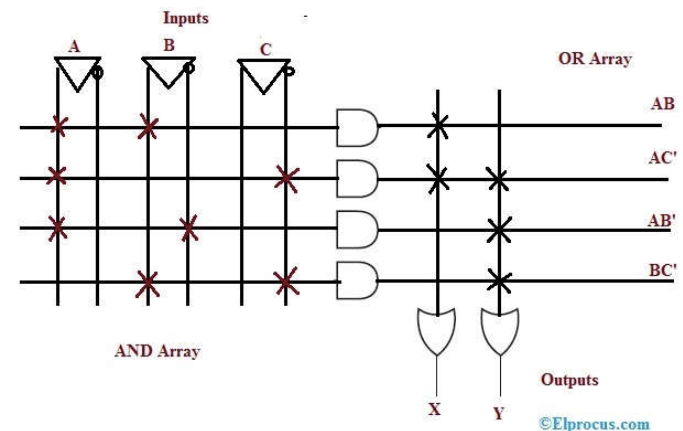
- Cấu tạo gồm từ vài chục - vài trăm cổng logic được tổ hợp dưới dạng ma trận các cổng AND và OR
- Hoạt động tốc độ cao, mạch cỡ nhỏ có thể lập trình 1 hoặc nhiều lần
- Phân chia thành 2 loại

PAL - Programmable Array Logic



AND khả trình, OR cố định

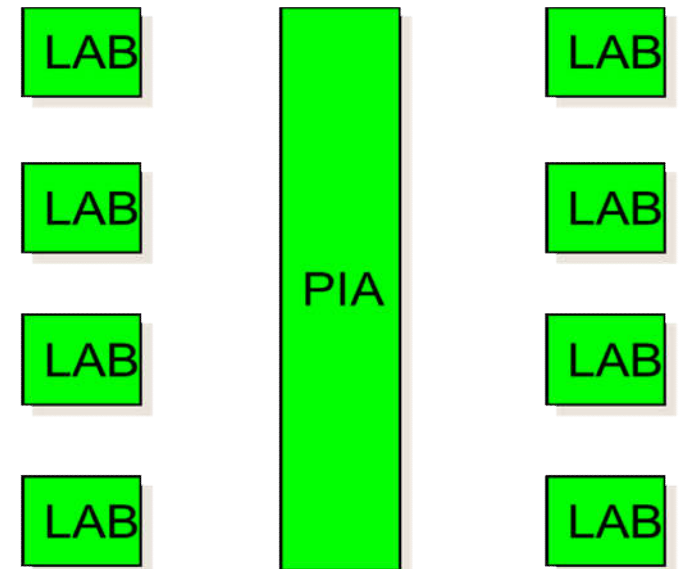
PLA - Programmable Logic Array



AND và OR đều khả trình

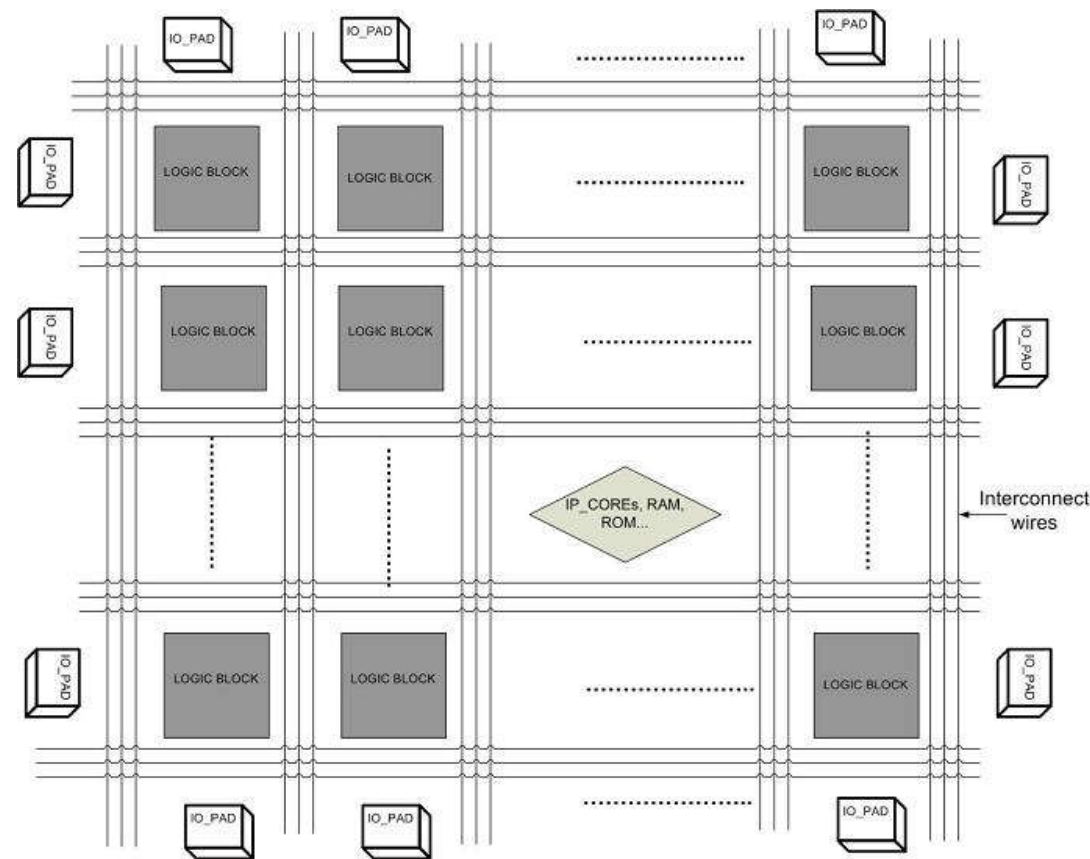
# Complex PLD - CPLD

- Gồm nhiều mảng Simple PLD
- Cấu tạo gồm khối SPLD được gọi là Logic Array Block LAB và
- khối Programmable Interconnect Array - PIA



# Field Programmable Gate Array - FPGA

- Gồm các khối khả trình, đúc sẵn các khối logic và các kết nối
- Các khối được kết nối với nhau bằng cách lập trình
- Đặc điểm
  - Không cần chế tạo
  - Phù hợp với thiết kế có độ phức tạp trung bình (<1M gate)



Kiến trúc tổng quan FPGA

# FPGA vs Vi điều khiển

- Vi điều khiển

- Chỉ xử lý được 1 luồng lệnh
- Xung nhịp trong khoảng 10-100MHz
- Xử lý luồng lệnh đơn
- Hoạt động với các dữ liệu 8, 16 hoặc 32 bit

- FPGA

- Có khả năng xử lý đồng thời nhiều luồng dữ liệu song song
- Các hàm có thể hoạt động với các dữ liệu có độ lớn lên tới hàng trăm bit

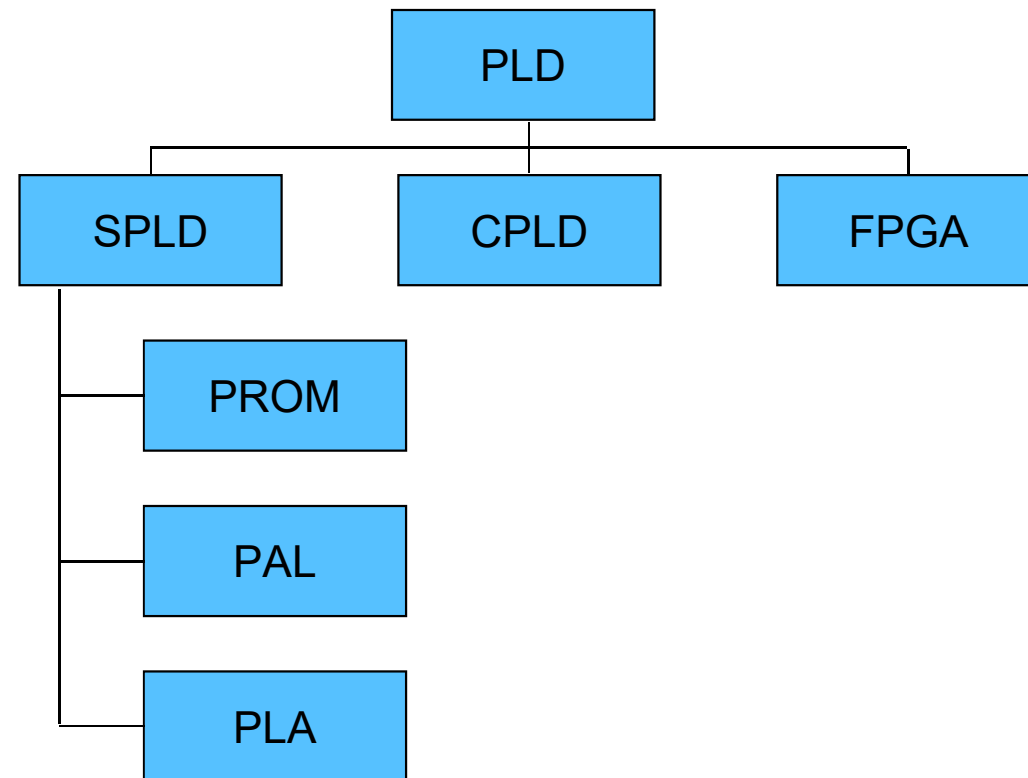
# FPGA vs CPLD

	FPGA	CPLD
<b>Khối logic</b>	Logic cell nằm ngoài, chia chung nguồn tài nguyên	Logic cell nằm giữa các nguồn tài nguyên
<b>Kết nối</b>	Phức tạp	Đơn giản hơn
<b>Công nghệ lập trình</b>	SRAM	EPROM, EEPROM
<b>Cấu trúc logic</b>	LUT	Mảng AND
<b>Ứng dụng</b>	Vừa và lớn	nhỏ



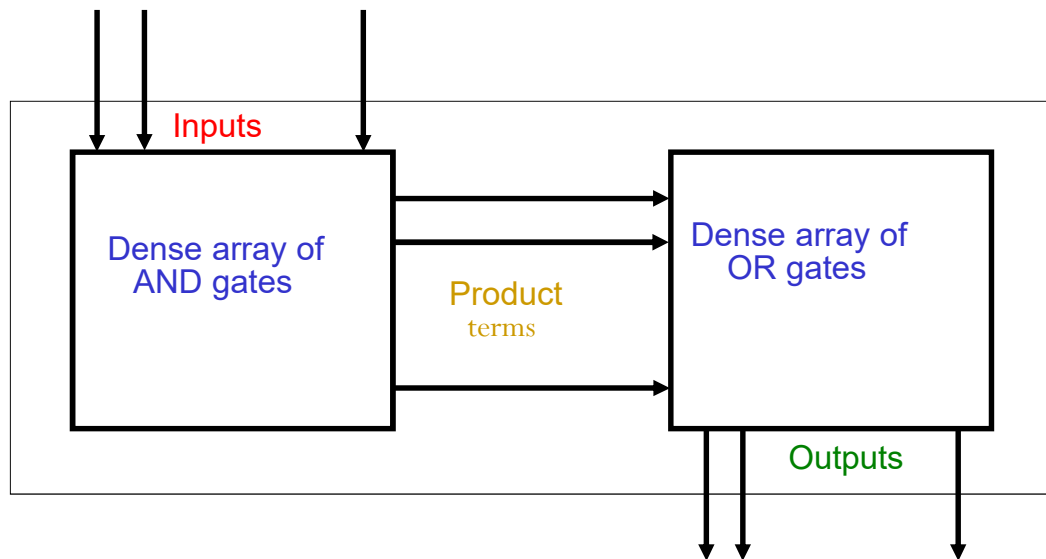
# Giới thiệu công nghệ IC lập trình

- Giới thiệu chung
- Phân loại IC khả trình
- **Kiến trúc IC khả trình**
- Quy trình thiết kế IC khả trình



# Kiến trúc IC khả trình

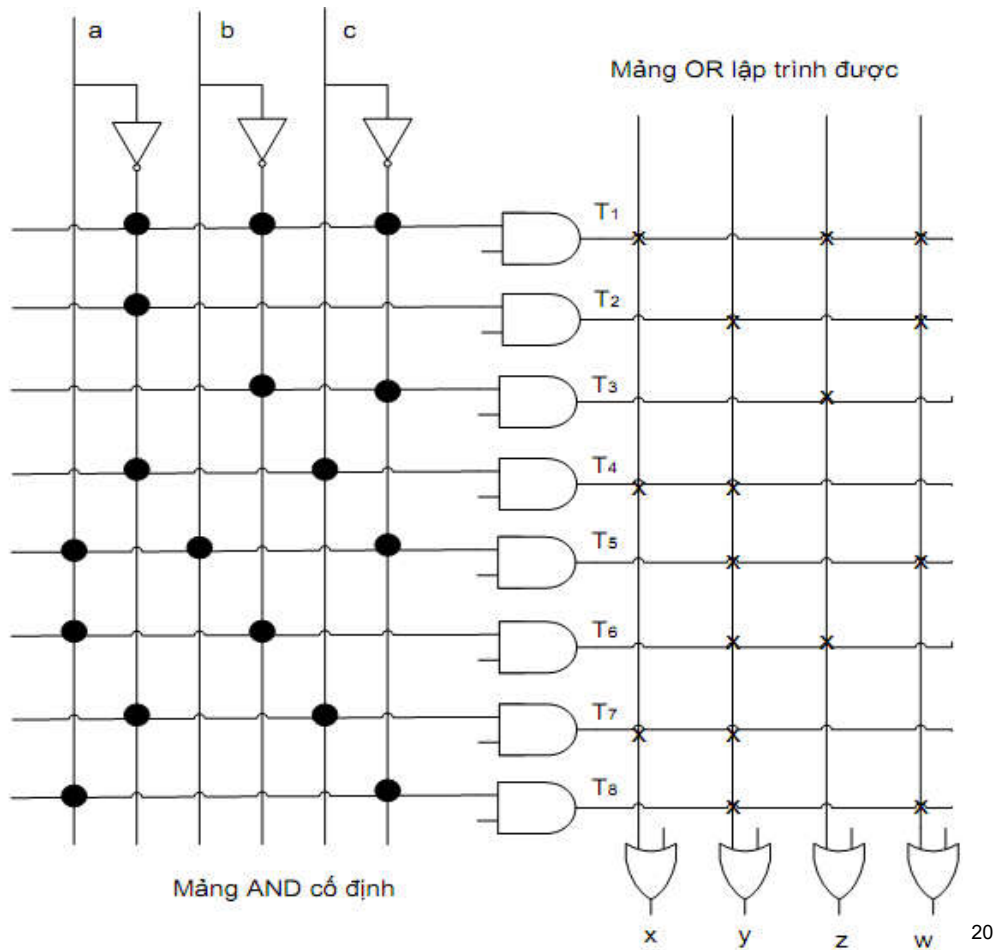
- Thực hiện hàm bằng cách ghép 2 mảng AND và OR
- Một trong 2 mảng khả trình, được gọi là IC khả trình



# Kiến trúc SPLD-PROM

- PROM - Programmable Read Only Memory - 1956
- 16 đến 32 input. Do đó chỉ thực hiện các hàm đơn giản
- Cấu trúc: AND cố định, mảng OR khả trình
- Thực hiện thông qua kết nối antifuse
- PROM có khả năng tái lập trình gồm
  - UEPROM - Ultraviolet-Erasable PROM
  - EEPROM - Electric-Erasable PROM

# Kiến trúc SPLD-PROM

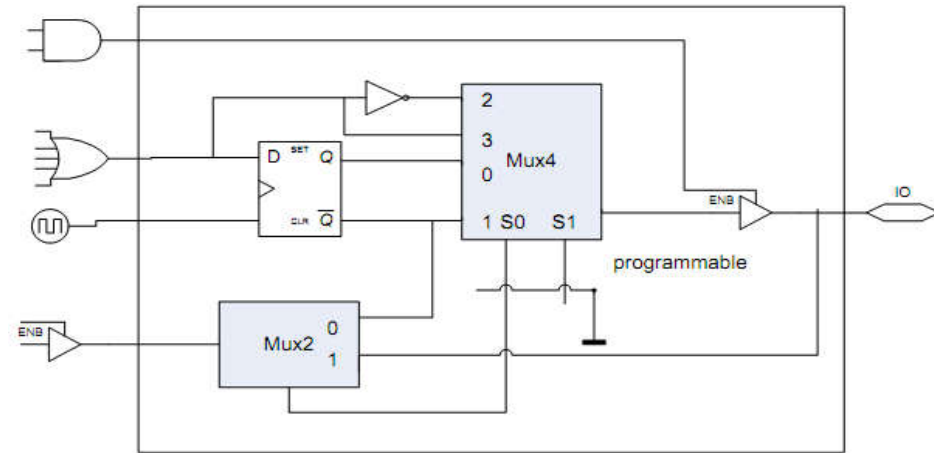
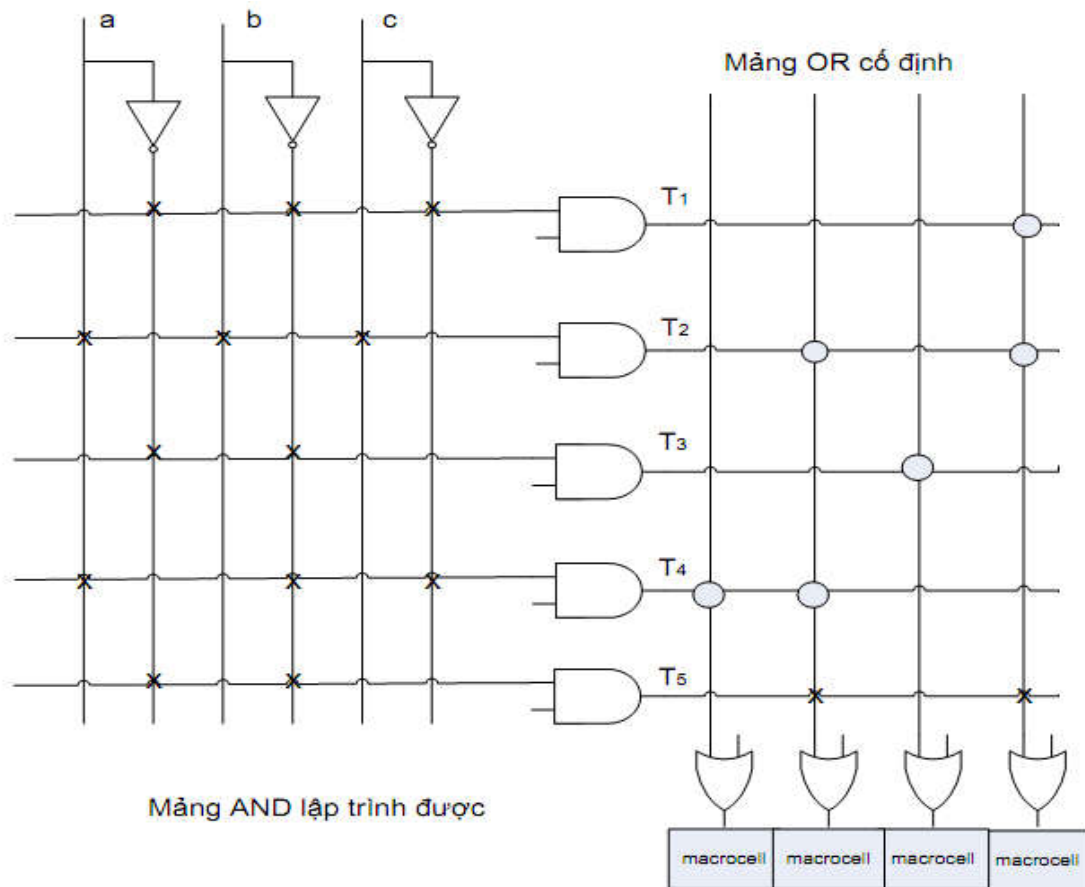


- $T_1 = \bar{a} . \bar{b} \bar{c}$
- $T_3 = \bar{b} \bar{c}$
- $y = T_1 + T_3$
- (x: antifuse chưa kết nối mảng OR lập trình được))
- ( : Kết nối cứng mảng AND cố định)

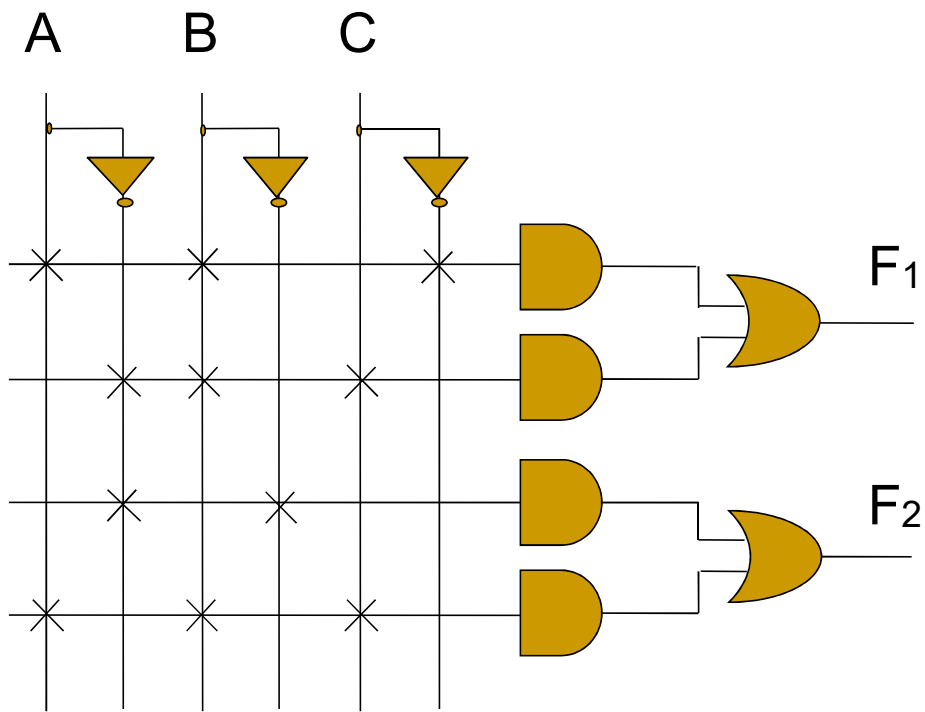
# Kiến trúc SPLD-PAL

- PROM - Programmable Array Logic - 1970
- Cấu trúc: AND khả trình, mảng OR cố định. Do vậy linh hoạt hơn
- Mỗi đầu ra của mảng OR lập trình được gắn với macro-cell
- Macro-cell = 1 flip-flop + 1 MUX2 + 1 MUX4, được dùng để thực hiện các mạch logic dãy

# Kiến trúc SPLD-PAL

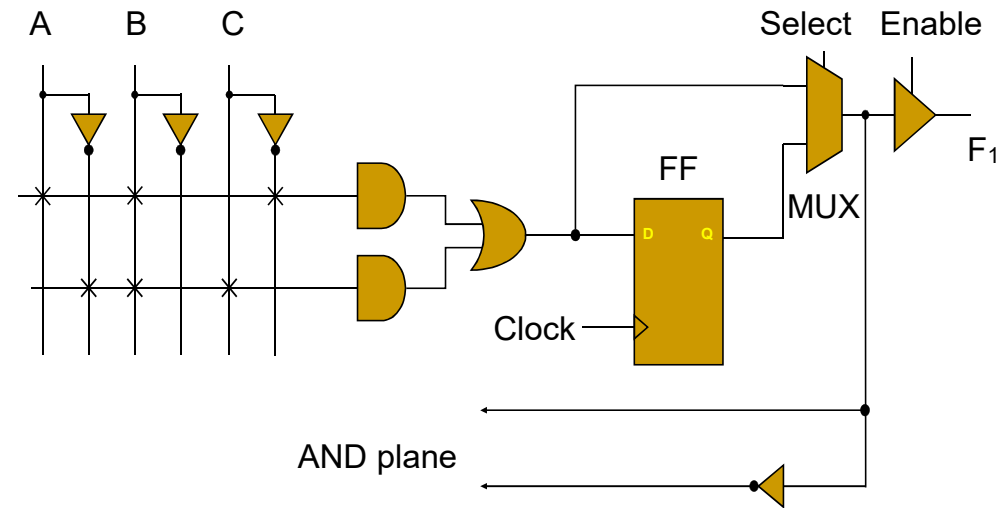


# Kiến trúc SPLD-PAL



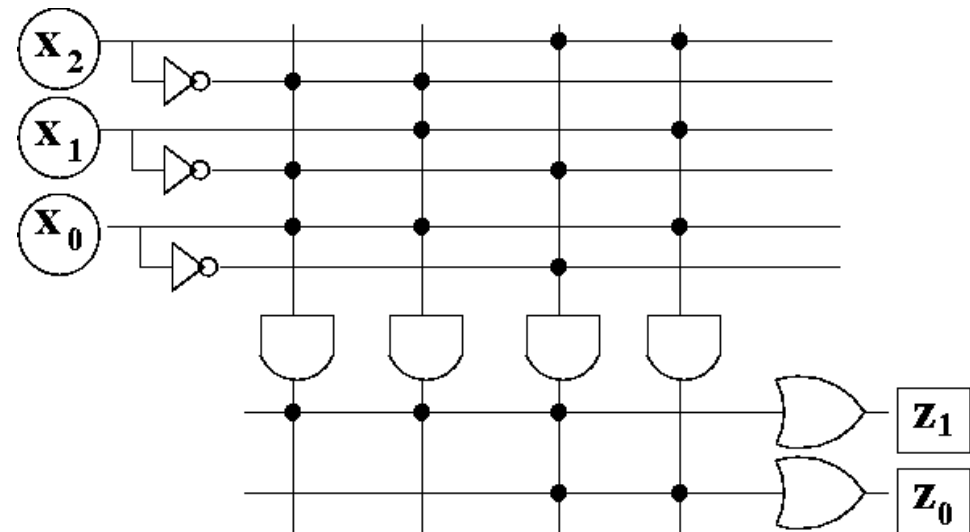
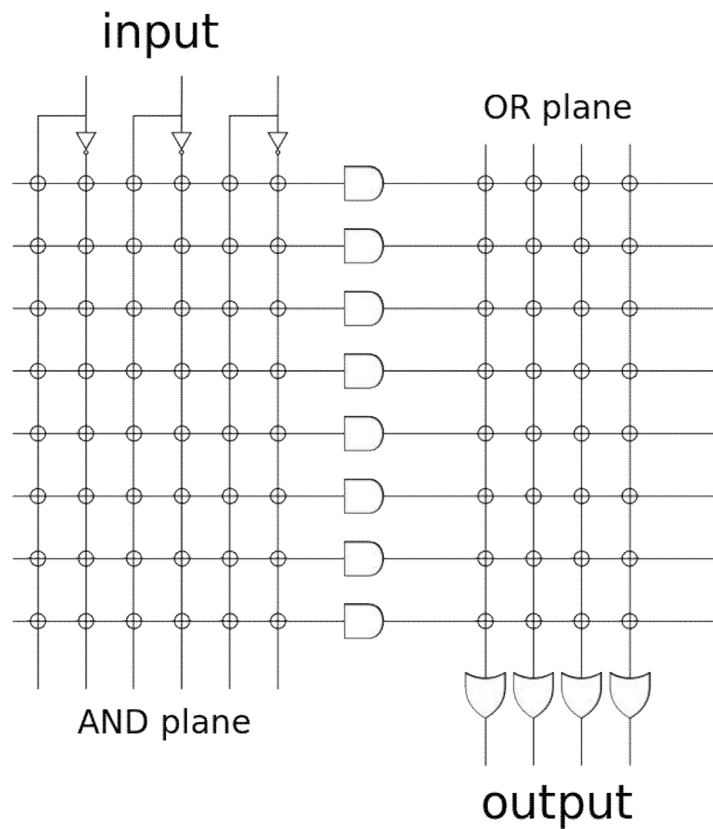
$$F_1 = A.B.\bar{C} + \bar{A}.B.C$$

$$F_2 = \bar{A}.\bar{B} + A.B.C$$



# Kiến trúc SPLD-PLA

- Cả 2 ma trận AND và OR đều khả trình





# Thực hiện PLA logic

- Cần thực hiện các hàm

$$F_0 = A + \overline{B}C$$

$$F_1 = A + \overline{B}.C$$

$$F_1 = A.B + A.\overline{C}$$

$$F_2 = A.B + \overline{B}.\overline{C}$$

## Personality Matrix

Product term	Inputs			Outputs			
	A	B	C	F <sub>0</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>
A B	1	1	-	0	①	①	0
$\overline{B} C$	-	0	1	0	0	0	1
A $\overline{C}$	1	-	0	0	1	0	0
$\overline{B} \overline{C}$	-	0	0	①	0	①	0
A	1	-	-	①	0	0	①

Reuse of term

### Input Side:

1 = asserted in term

0 = negated in term

- = does not participate

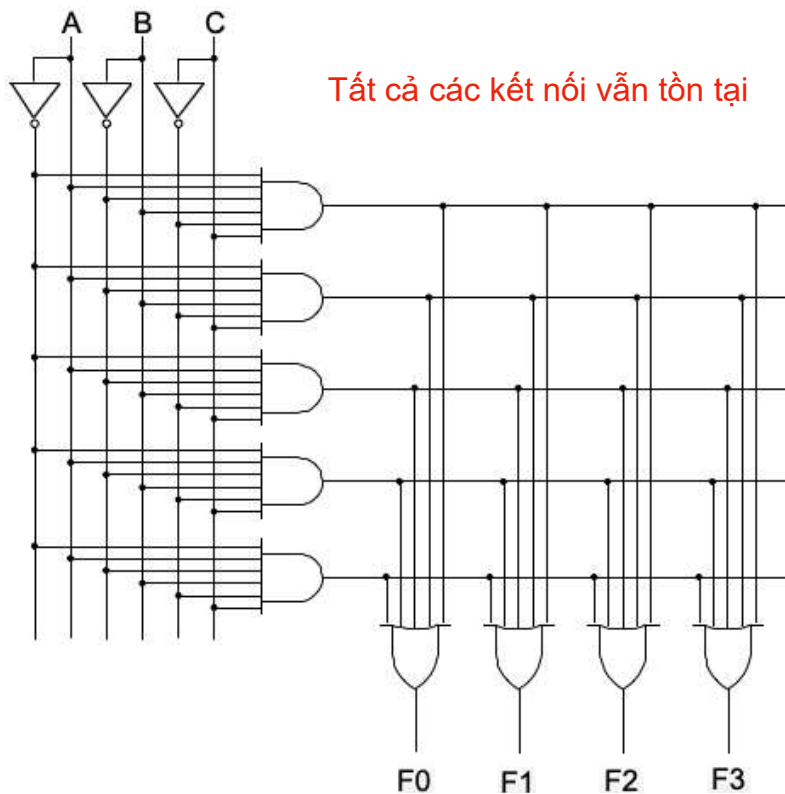
### Output Side:

1 = term connected to output

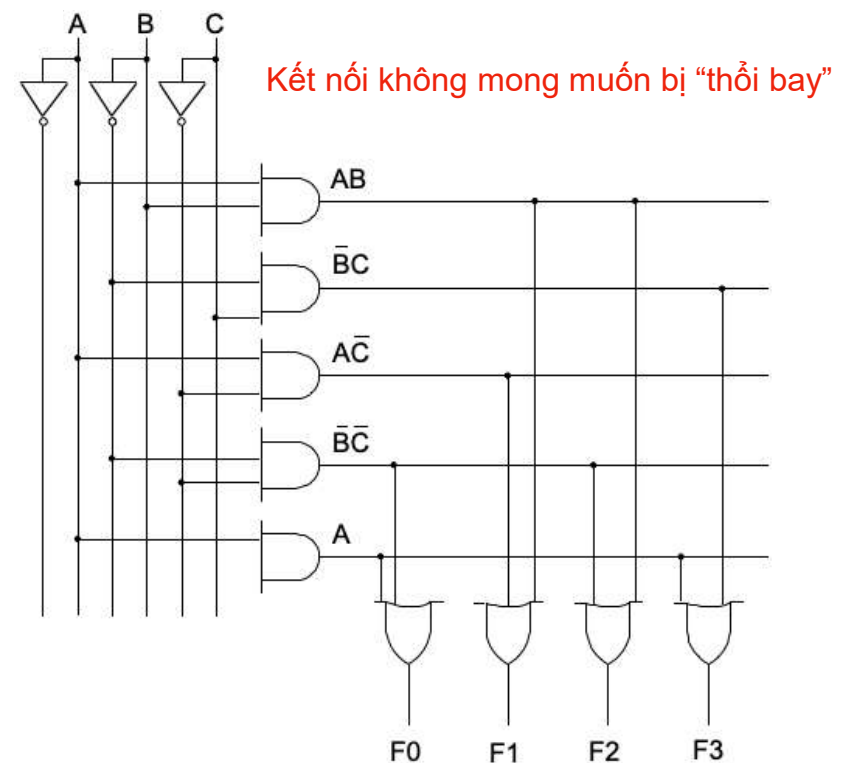
0 = no connection to output

# Thực hiện PLA logic

- Trước khi lập trình



- Sau khi lập trình



# Thực hiện PLA logic

Multiple functions of A, B, C

$$F1 = A B C$$

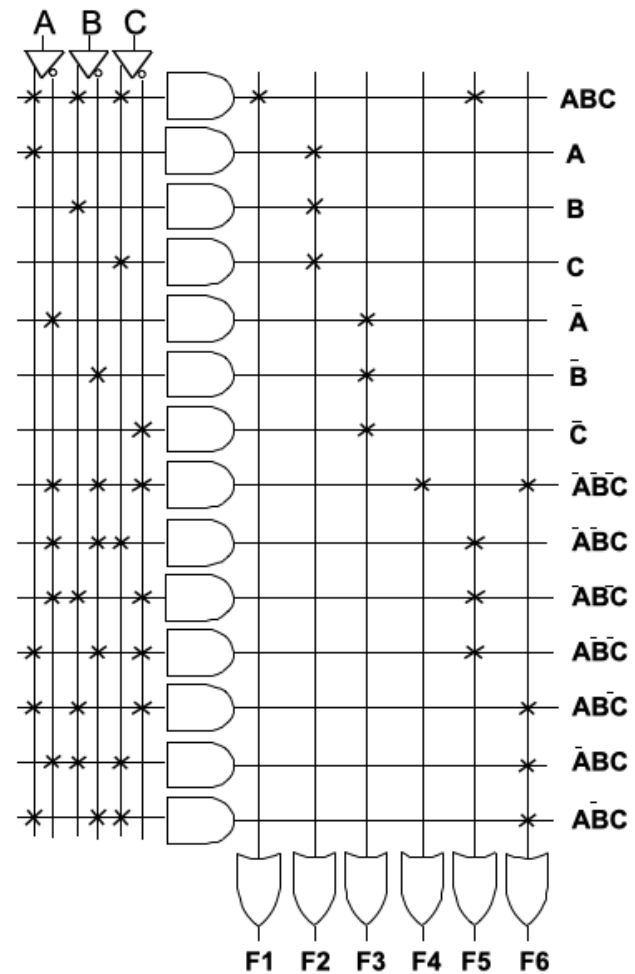
$$F2 = A + B + C$$

$$F3 = \overline{A B C}$$

$$F4 = \overline{A + B + C}$$

$$F5 = A \oplus B \oplus C$$

$$F6 = \overline{A \oplus B \oplus C}$$



# So sánh PALs và PLAs

PLAs	PALs
Cả mảng AND và OR đều lập trình được	Mảng AND lập trình được, mảng OR cố định
Linh hoạt hơn	Kém linh hoạt
Thực hiện được một số lượng lớn hàm logic, nhưng nhiều chân	Số lượng hàm output phụ thuộc số lượng cổng OR
Giá thành cao	Giá thành rẻ hơn so với PLAs

# Bộ chuyển mã BCD sang mã Gray

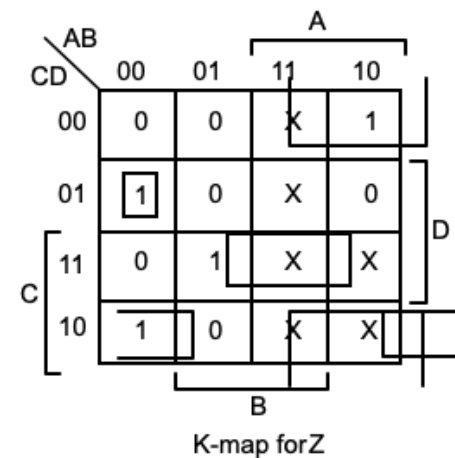
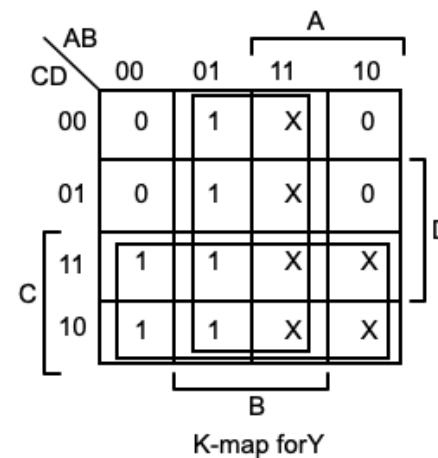
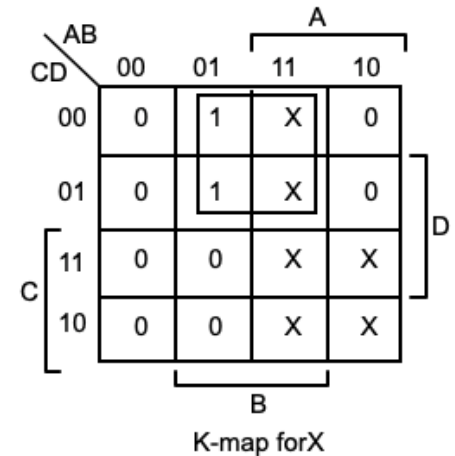
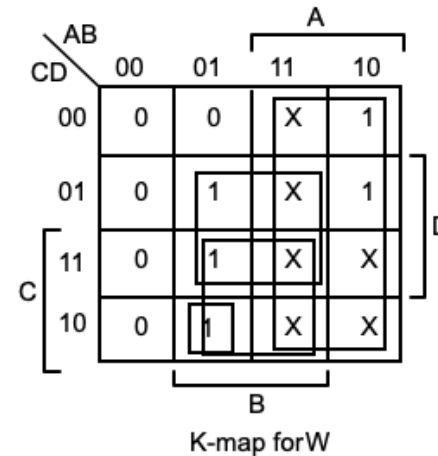
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	1	1	1	0
0	1	1	0	1	0	1	0
0	1	1	1	1	0	1	1
1	0	0	0	1	0	0	1
1	0	0	1	1	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

$$W = A + B.D + B.C$$

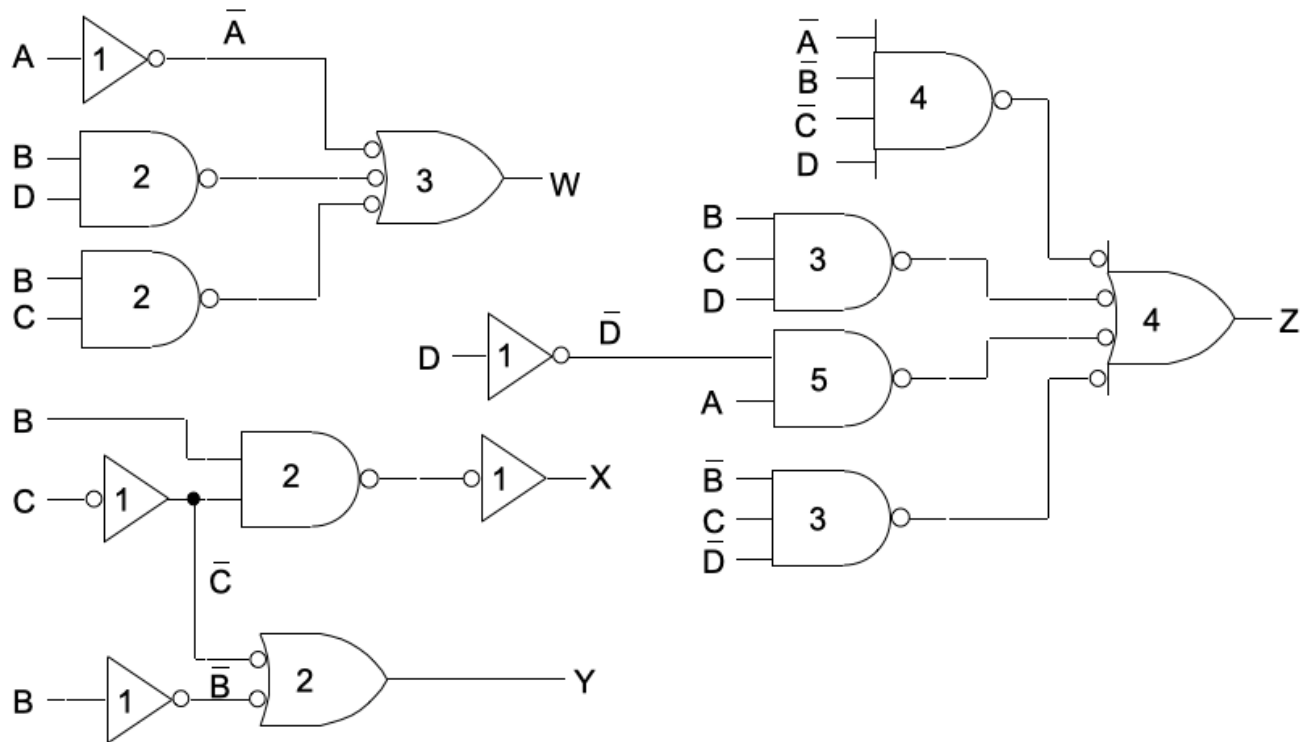
$$X = B.\bar{C}$$

$$Y = B + C$$

$$Z = \bar{A}.\bar{B}.\bar{C}.D + \bar{B}.C.\bar{D} + A.\bar{D} + B.C.D$$



# Bộ chuyển mã BCD sang mã Gray



1: 7404 hex inverters

2,5: 7400 quad 2-input NAND

3: 7410 tri 3-input NAND

4: 7420 dual 4-input NAND

**cần 4 packages SSI**

$$W = A + B.D + B.C$$

$$X = B.\bar{C}$$

$$Y = B + C$$

$$Z = \bar{A}.\bar{B}.\bar{C}.D + \bar{B}.C.\bar{D} + A.\bar{D} + B.C.D$$

## Bộ chuyển mã BCD sang mã Gray

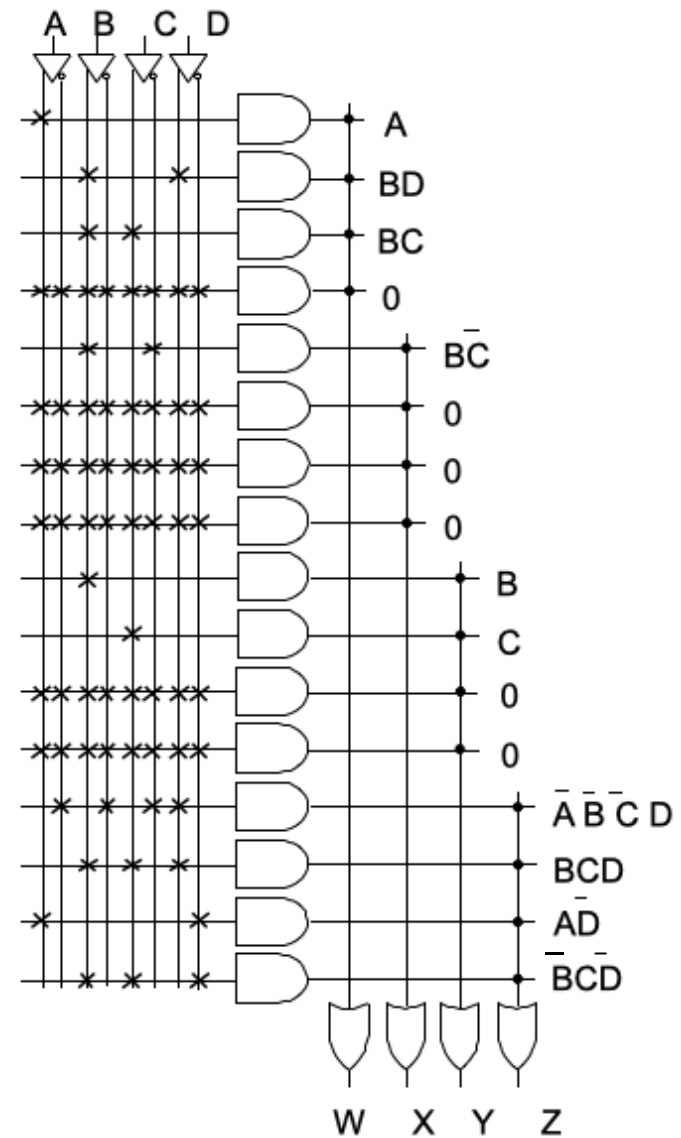
$$W = A + B.D + B.C$$

$$X = B.\bar{C}$$

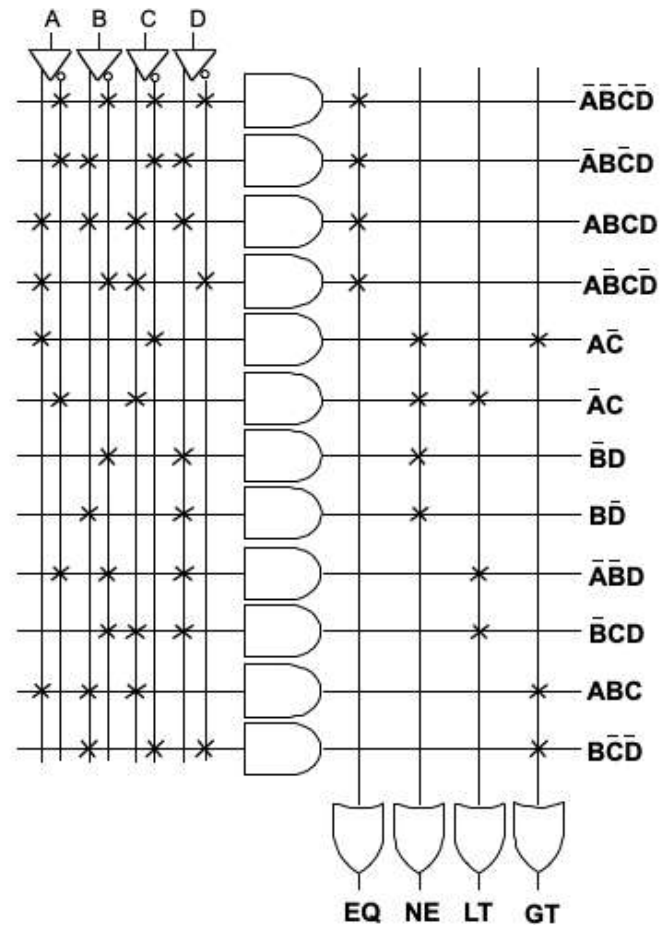
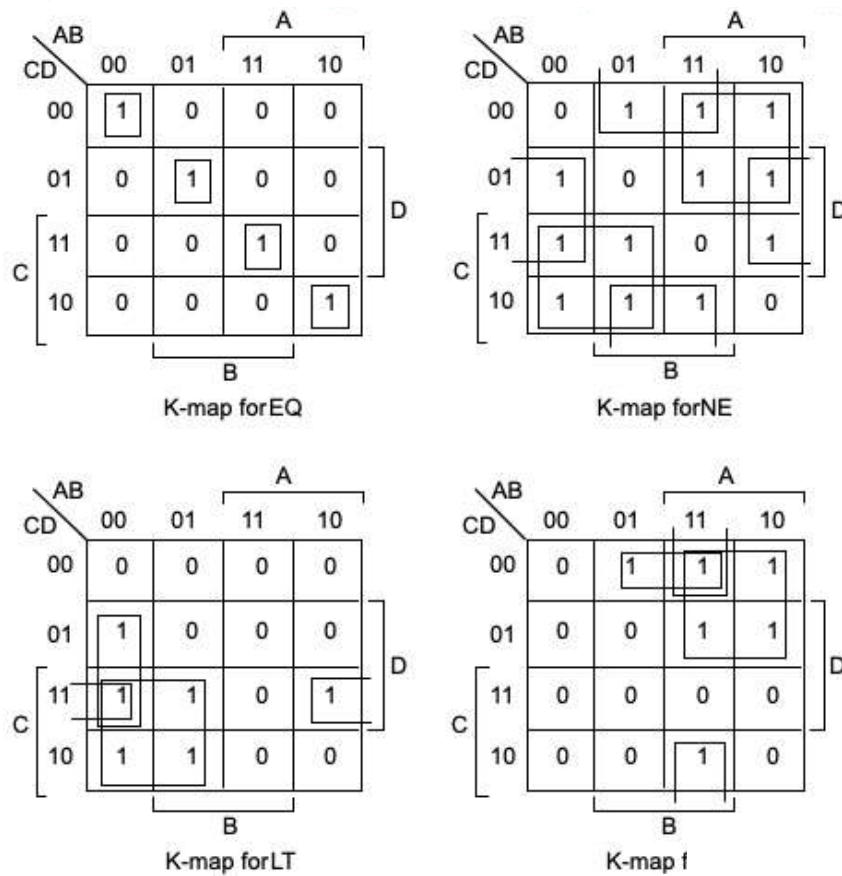
$$Y = B + C$$

$$Z = \bar{A}.\bar{B}.\bar{C}.D + \bar{B}.C.\bar{D} + A.\bar{D} + B.C.D$$

cần 1 package PLA/PAL



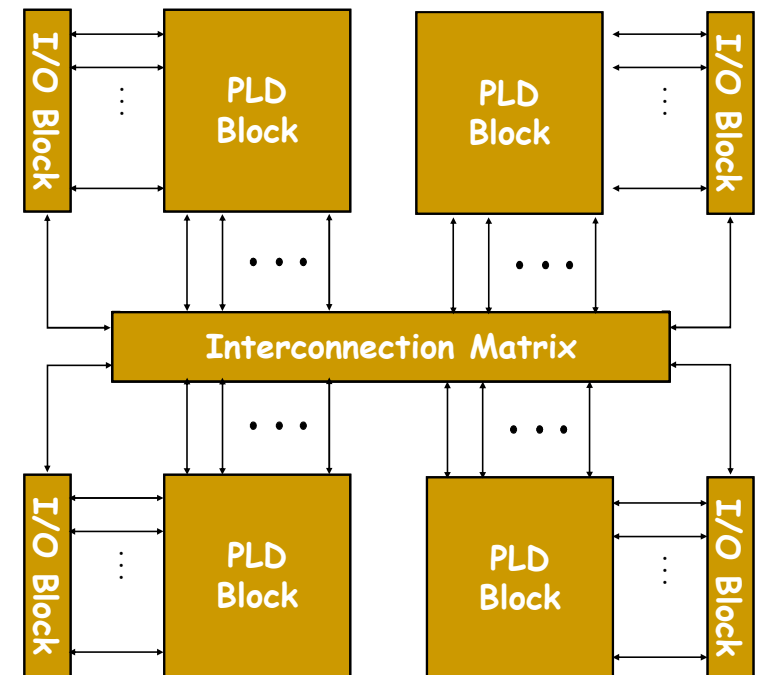
# Bộ so sánh số 2 bit





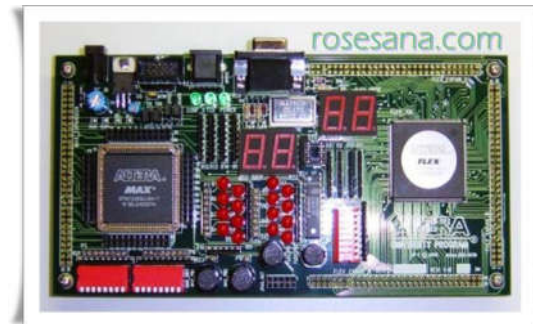
# Kiến trúc CPLD

- CPLD thường gồm logic tổ hợp PAL và các FF
- Cấu trúc thành các PLD Block - Logic Block là các SPLD cải tiến, chứa từ 8 đến 16 macrocell
- Ma trận kết nối khả trình, kết nối các khối PLD và I/O
- Dùng công nghệ lập trình EEPROM
- Kích thước mảng OR cố định
- Đầu ra tổ hợp hoặc dây (registered)
- Ứng dụng: Bộ đếm đơn, State machines, decoders,...



# Board mạch CPLD - Altera Max7000

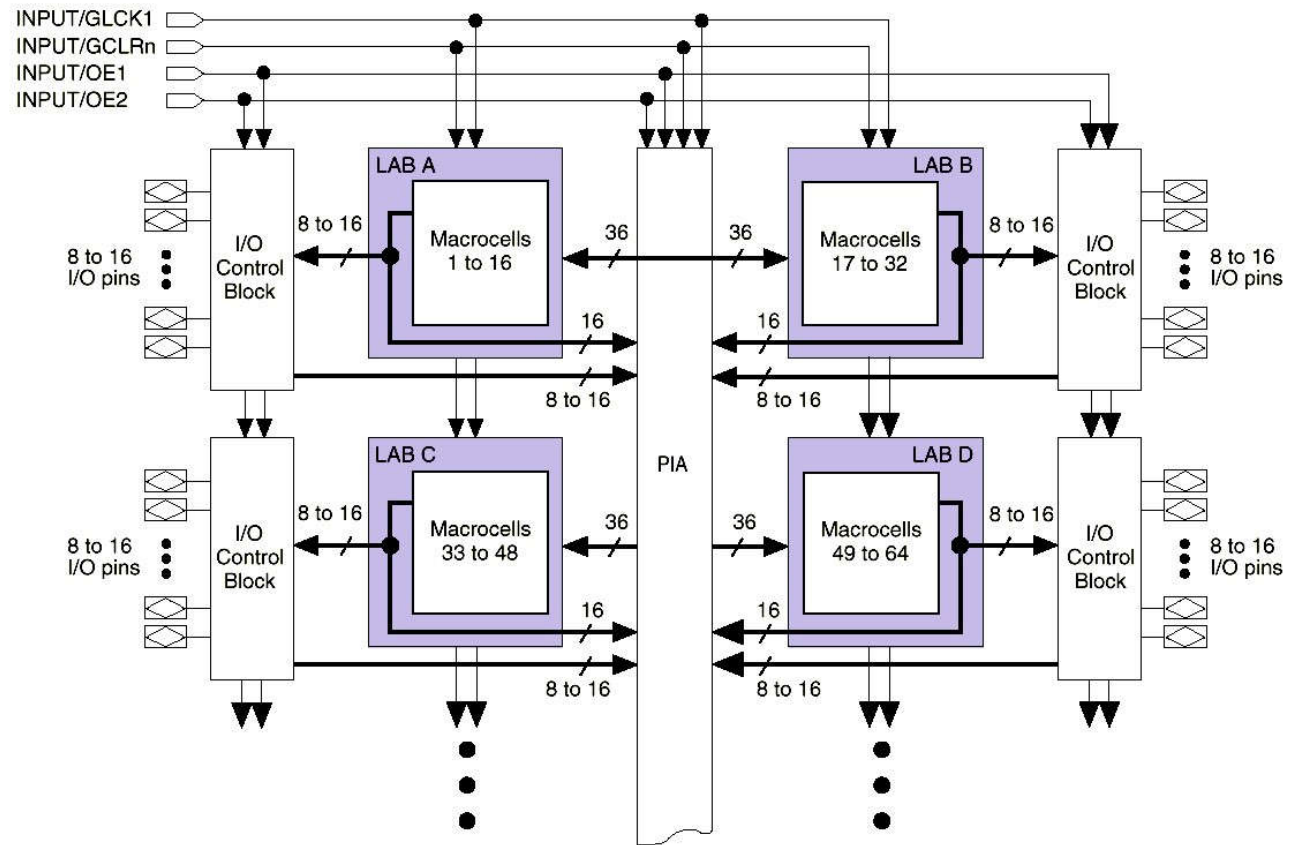
- Mật độ từ 2 LAB đến 16 LAB - Logic Array Block
- Mỗi LAB tương ứng với một SPLD dùng công nghệ EEPROM
- Mỗi LAB có 16 macrocell
- PIA - cấu trúc bú lập trình, kết nối các LAB, I/O, macrocell
- Kiểu lập trình ISP - In-System Programmable



Chức năng	EPM7032	EPM7064	EPM7096	EPM7128E	EPM7160E	EPM7192E	EPM7256E
Useable gates	600	1250	1800	2500	3200	3750	5000
Macro-cells	32	64	96	128	160	192	256
LABs	2	4	6	8	10	12	16
Mã I/O pin	36	68	76	100	104	124	164

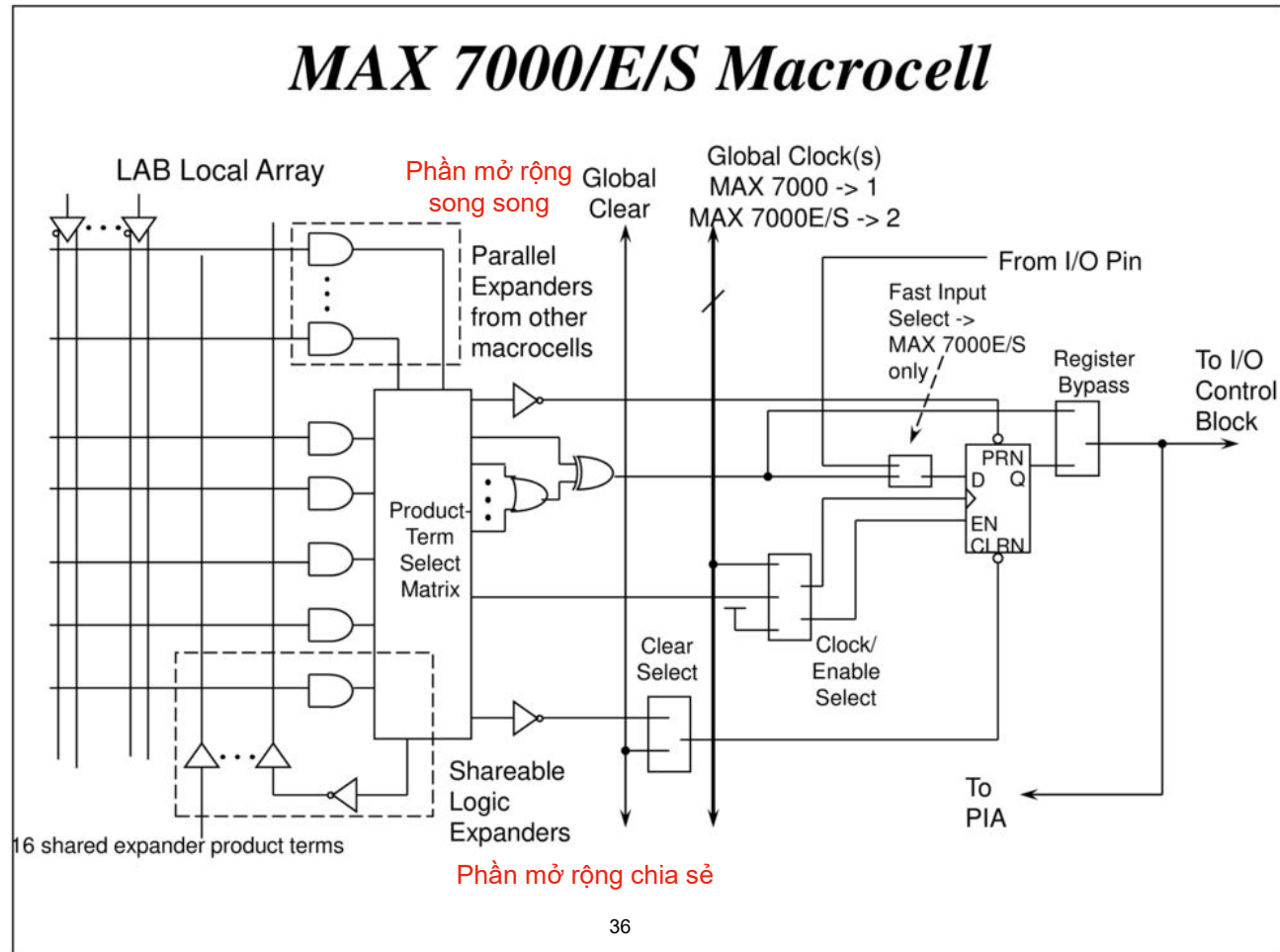
# Board mạch CPLD - Altera Max7000

- EPM7000 Series Block Diagram



# Board mạch CPLD - Altera Max7000

- Macrocell

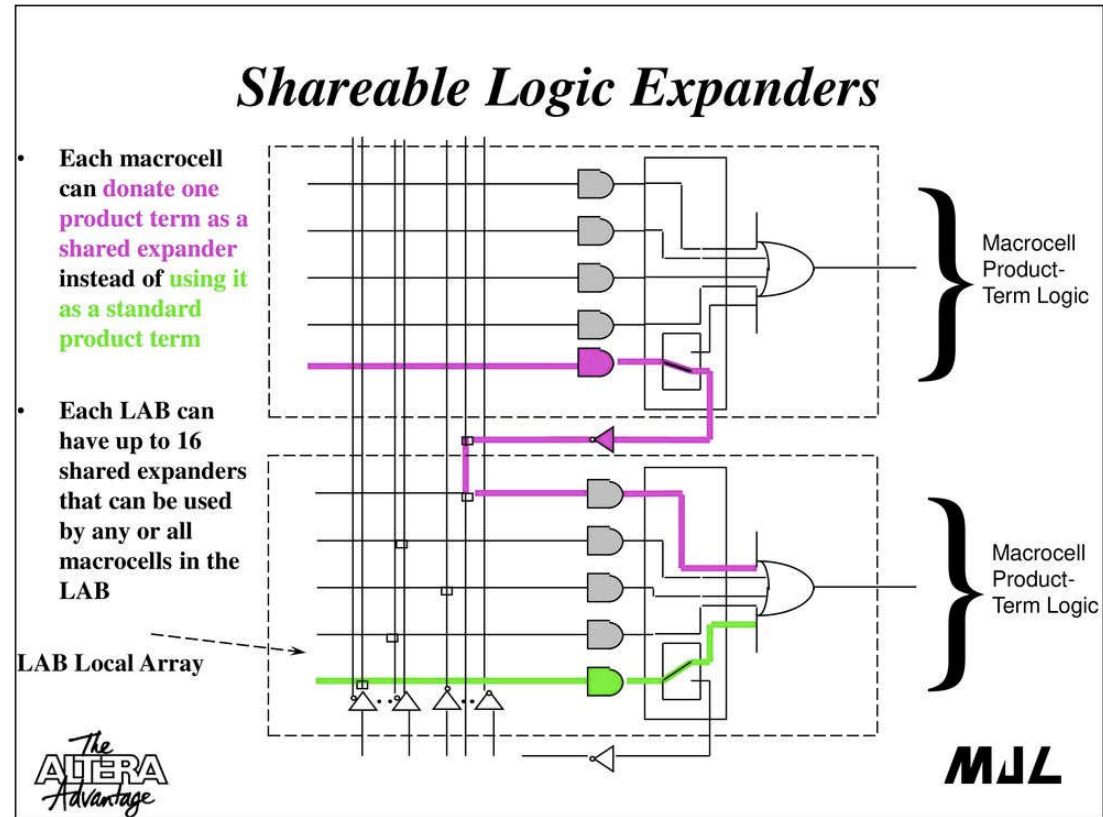
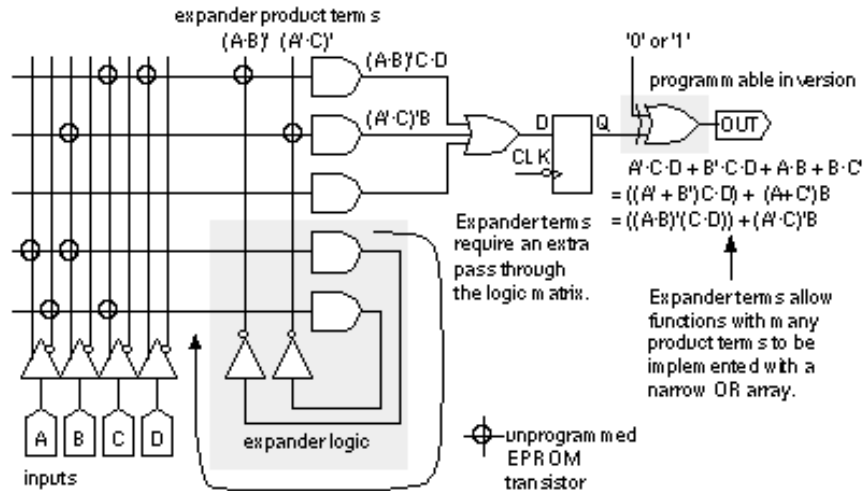


# Board mạch CPLD - Altera Max7000

- Macrocell
  - Chứa 1 mảng cổng AND khả trình gồm: 5 cổng AND + 1 cổng OR, và ma trận lựa chọn tích
  - Ma trận lựa chọn tích: kết nối output AND với input OR, mạch logic tổ hợp phục vụ lập trình cho đầu vào/ra
  - Phần mở rộng chia sẻ: hồi tiếp các thành phần tích trở lại ma trận lập trình để chia sẻ với các macrocell khác
  - Phần mở rộng song song: mượn các thành phần tích dùng từ các macrocell khác

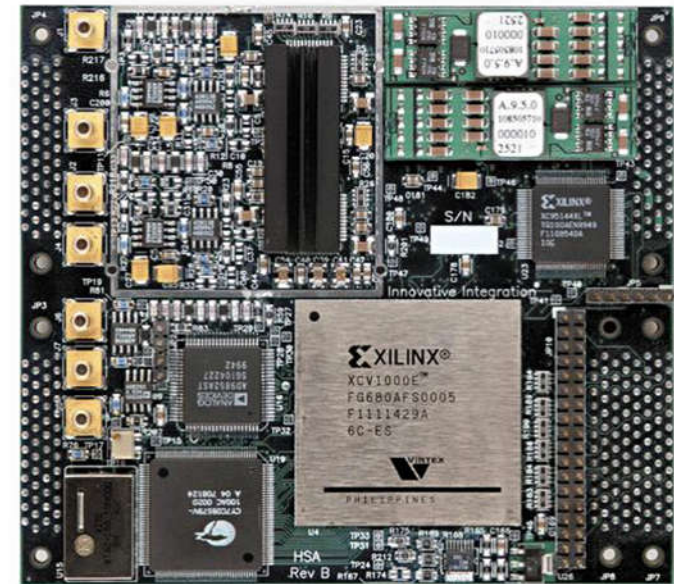
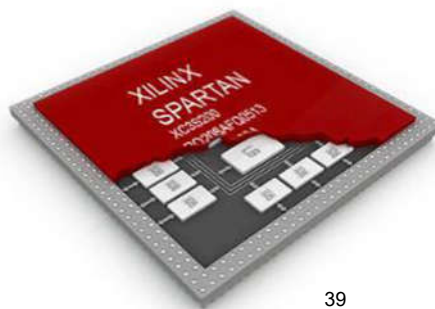
# Board mạch CPLD - Altera Max7000

- Phần mở rộng chia sẻ



# Field-Programmable Gate Array - FPGA

- FPGA: mảng cổng logic có thể lập trình được theo trường/miền
- Mật độ tích hợp cao hơn nhiều so với CPLD
- Cấu trúc mảng phần tử logic - CLB có thể lập trình được
- Kết nối theo kiểu hàng và cột
- Phần tử logic tạo hàm nhỏ hơn nhiều so với CPLD
- Công nghệ lập trình SDRAM



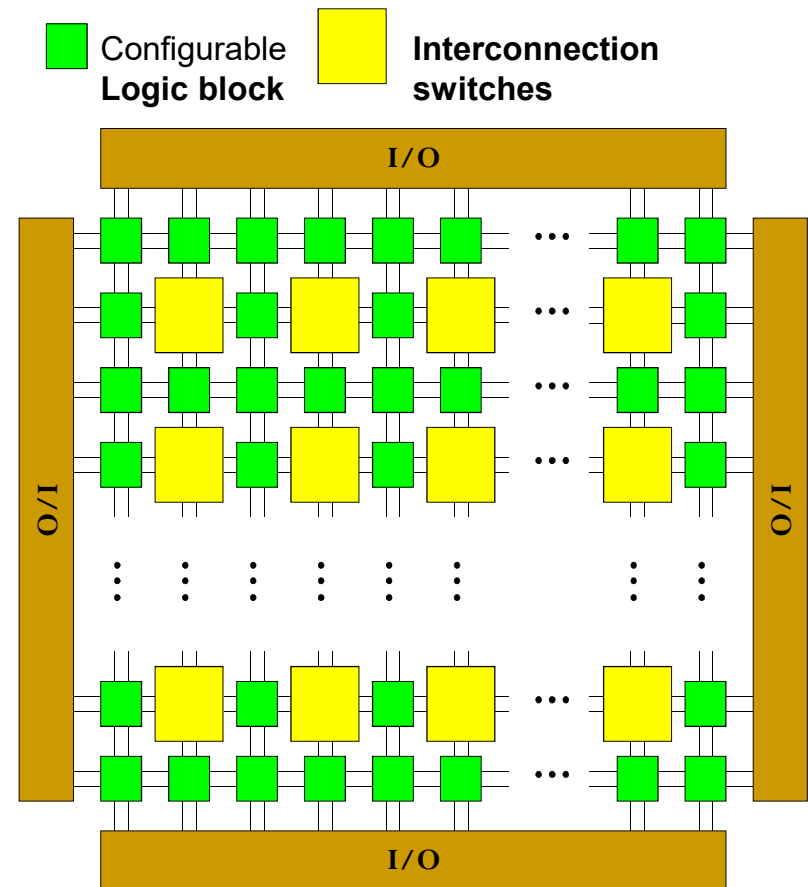
# Field-Programmable Gate Array - FPGA

- FPGA có số lượng cổng logic lớn hơn rất nhiều so với CPLD
  - 2K đến 10M cổng
  - Đòi hỏi nhiều công nghệ khác nhau
  - FPGA chia làm 2 loại: RAM-based hoặc Flash-based
- RAM based FPGA
  - Lập trình khi power-on
  - Cần bộ nhớ mở rộng để lập trình
  - Có thể cấu hình động
- Flash based FPGA
  - Lưu chương trình trong bộ nhớ bền vững
  - Lập trình lại khó khăn
  - Cấu hình giữ nguyên khi mất điện



# Field-Programmable Gate Array - FPGA

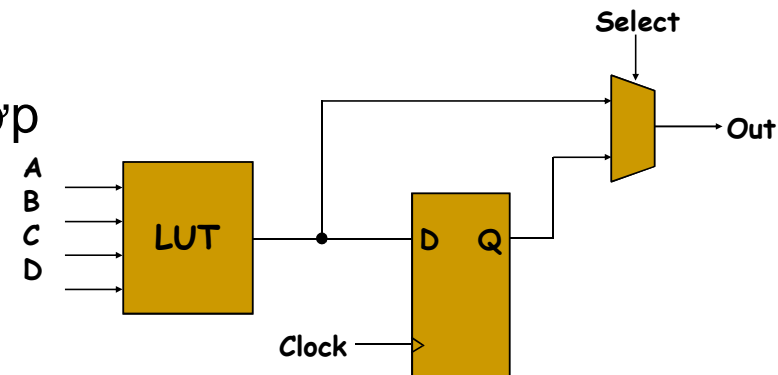
- Cấu trúc chung
  - Configurable Logic Block - CLB: Khả trình, thực hiện chức năng logic dãy và tổ hợp
  - Interconnection: Khả trình, kết nối đầu vào/ra với các logic Block CLB
  - Khối I/O khả trình
  - Khối khác
    - ▶ Phân phối Clock
    - ▶ Bộ nhớ nhúng
    - ▶ DSP, RAM, ROM,...



# Field-Programmable Gate Array - FPGA

- Configurable Logic Block - CLB
  - Chứa phần tử logic cơ bản Logic Element - LE
  - Kết nối bên trong CLB khả trình, kết nối giữa LE với CLB
  - Mỗi LE

- ▶ Khối bảng tham chiếu LUT - mạch tổ hợp
- ▶ Thanh ghi thực hiện mạch logic
- ▶ Một số mạch logic khác: số học, mạch mở rộng cho các chức năng có hơn 4 input

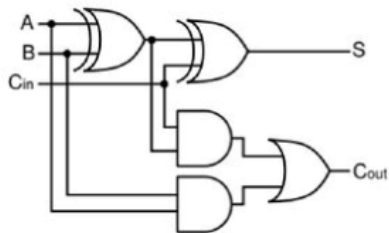


# Field-Programmable Gate Array - FPGA

- Look-Up Table (LUT)
  - Bảng tham chiếu n-input có thể thực hiện bất kì hàm logic hay tổ hợp các đầu vào
  - Lập trình bằng bảng chân lý

*Truth Table*

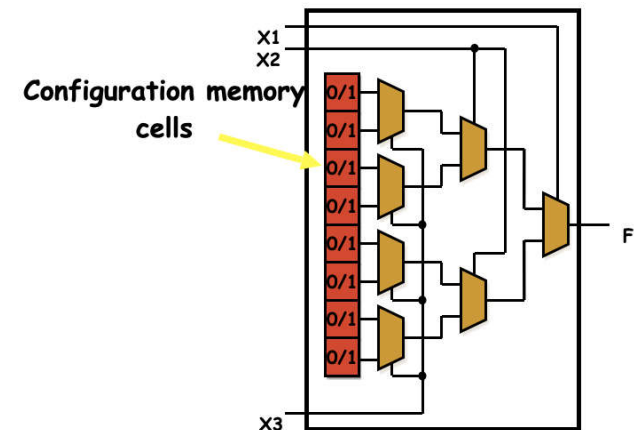
Inputs			Outputs	
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



*3-input, 2-output LUT*

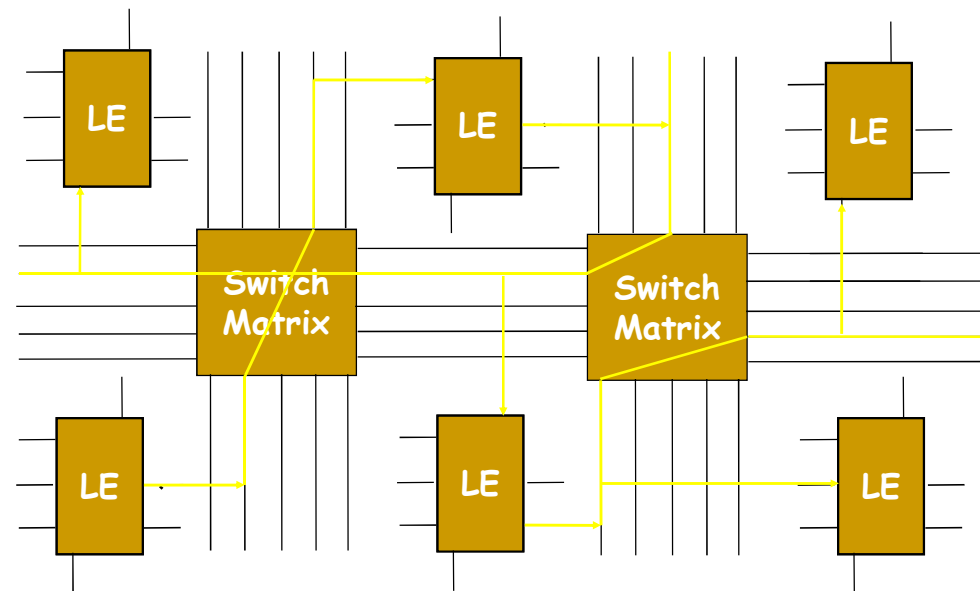
A	0	0
B	1	0
Cin	0	1
	1	0
	0	1
	0	1
	1	1
	S	Cout

43



# Field-Programmable Gate Array - FPGA

- Kết nối khả trình
  - có kiến trúc định tuyến, gồm các đoạn dây nối và các chuyển mạch lập trình được
  - Chuyển mạch lập trình được có thể có nhiều cấu tạo: anti-fuse, EPROM và EEPROM transistor



# Field-Programmable Gate Array - FPGA

- Ưu điểm
  - Khả năng tái lập trình khi đang sử dụng
  - Công đoạn thiết kế đơn giản so với ASIC, giảm được chi phí, thời gian
  - Khả năng cấu hình động
- Ứng dụng
  - Nền tảng lý tưởng để tạo mẫu
  - Thiết kế và hoàn thiện nhanh chóng, giảm thời gian đưa ra thị trường
  - Giải pháp hiệu quả cho các sản phẩm không yêu cầu số lượng lớn
  - Thực hiện các hệ thống phần cứng yêu cầu khả năng tái lập trình
  - Thực hiện các hệ thống tái cấu hình động

# Field-Programmable Gate Array - FPGA

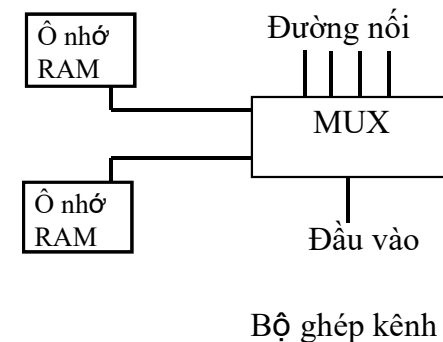
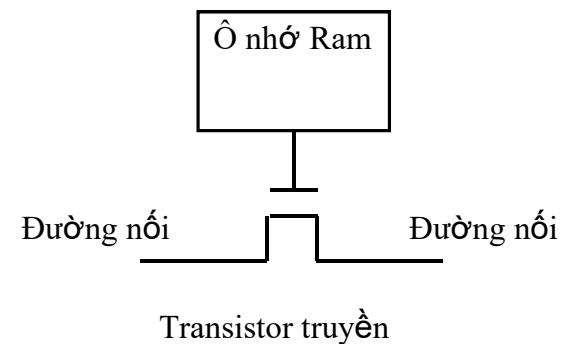
- Công nghệ lập trình chip
  - Các phần tử lập trình: cho phép các kết nối có thể lập trình được giữa các đoạn dây nối
  - Chức năng: thực hiện các kết nối lập trình được giữa các khối logic của FPGA
  - Được cấu hình ở trạng thái ON hoặc OFF
  - Chiếm diện tích của chip càng nhỏ càng tốt
  - Trở kháng khi ON thấp và khi OFF rất cao
  - Điện dung kí sinh thấp khi nối các đoạn dây
  - Có thể tích hợp một số lượng lớn các phần tử lập trình trên 1 chip

# Field-Programmable Gate Array - FPGA

- Phân loại công nghệ lập trình chip
  - Dùng SRAM
  - Dùng EPROM (UV Light Erasable PROM) và EEPROM
  - Dùng cầu chì nghịch (anti-fuse)

# Field-Programmable Gate Array - FPGA

- Lập trình dùng SRAM
  - Các kết nối lập trình được điều khiển bằng các cell nhớ SRAM
  - Mỗi chuyển mạch là một transistor truyền/MUX được điều khiển bằng trạng thái của một bit SRAM
  - Cần Power-on để nạp chương trình
  - Chip có diện tích lớn vì cần ít nhất 5 transistor cho mỗi RAM cell cũng như các transistor cần thêm cho mỗi cổng truyền hay MUX
  - Cho phép FPGA được tái cấu hình ngay trên mạch rất nhanh và có thể được chế tạo bằng công nghệ CMOS chuẩn

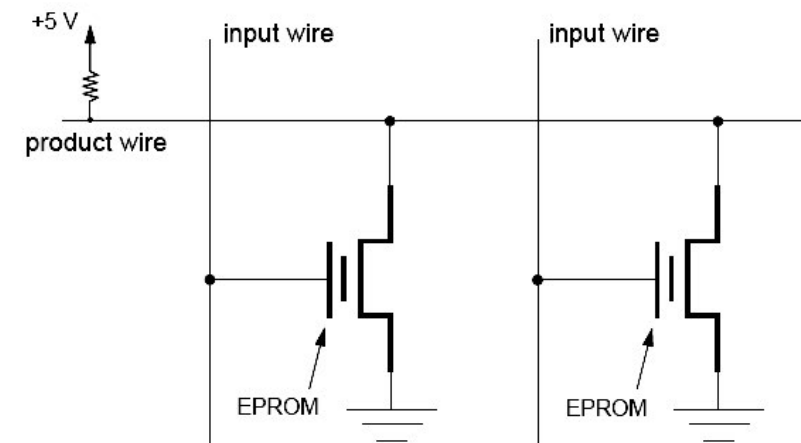




# Field-Programmable Gate Array - FPGA

- Lập trình dùng EPROM

- Thường dùng cho CPLD và SPLD
- Tái lập trình không cần bộ nhớ ngoài
- Không tái cấu hình trực tiếp trên mạch
- EPROM hay EEPROM transistor được đặt giữa 2 dây thực hiện các chức năng nối AND

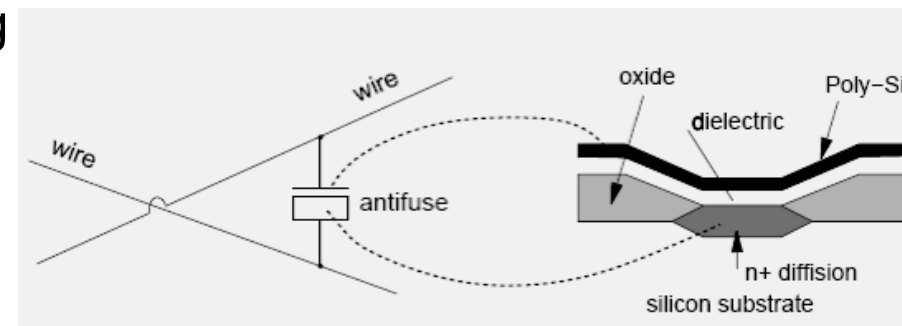


- EEPROM transistor 2 cổng

- ON: Không có điện tích giữa 2 cổng
- OFF: cho dòng lớn chạy giữa nguồn và kênh, điện tích được giữ lại cổng treo, Transistor OFF

# Field-Programmable Gate Array - FPGA

- Lập trình dùng cầu chì nghịch (anti-fuse)
  - Cấu trúc gồm 3 lớp: Silic mang nhiều điện tích dương (n+), điện môi và Poly-Silic
  - Lập trình bằng cách đặt điện áp cao ( $\sim 18V$ ) giữa 2 đầu antifuse, dòng điều khiển làm điện môi nóng chảy tạo lớp liên kết giữa Poly-Silic và n+. Hai lớp này được nối với các dây kim loại ( $300-500\Omega$ )
  - Chỉ thực hiện 1 lần, không có khả năng tái lập trình
  - Diện tích chip nhỏ. Tuy nhiên cần có không gian lớn cho các transistor điện thế cao để giữ cho dòng và áp cao lúc lập trình



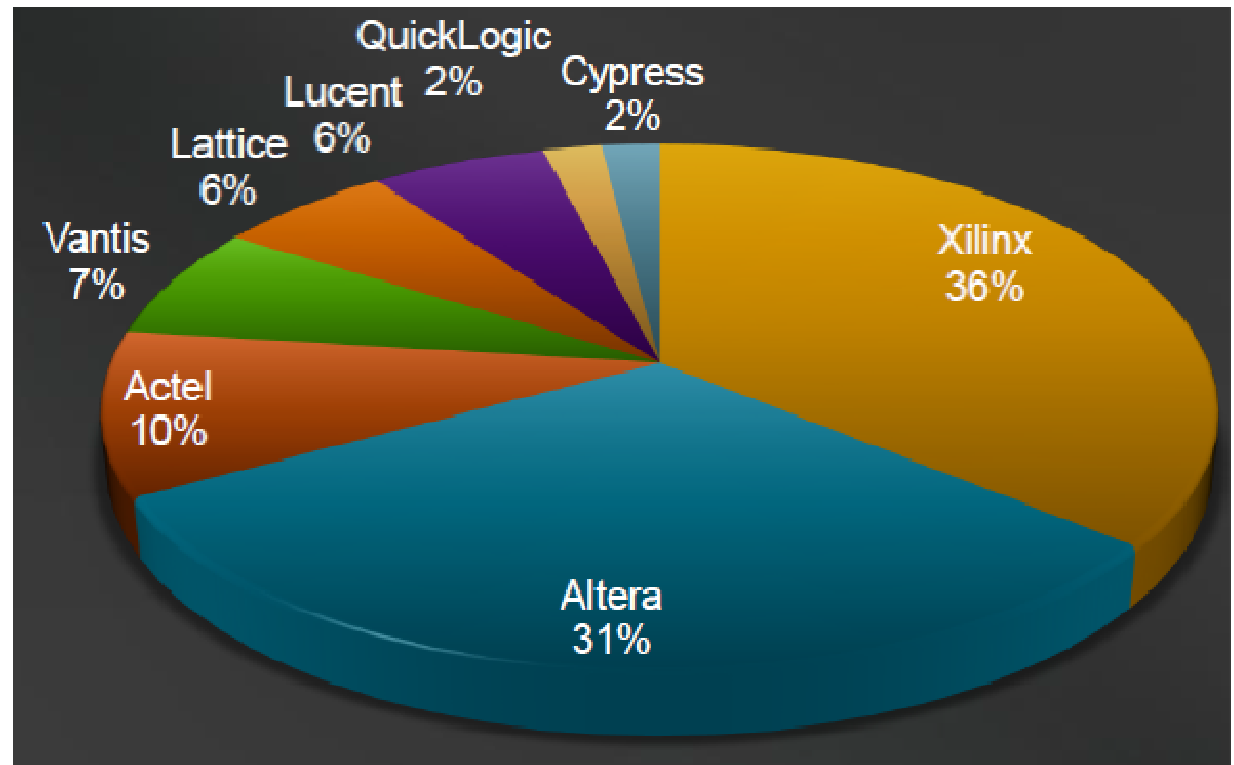
# Field-Programmable Gate Array - FPGA

- Tóm tắt

Name	Re-programmable	Volatile	Technology
Fuse	no	no	Bipolar
EPROM	yes out of circuit	no	UVC MOS
EEPROM	yes in circuit	no	EECMOS
SRAM	yes in circuit	yes	CMOS
Antifuse	no	no	CMOS+

# Cấu trúc FPGA của một số hãng

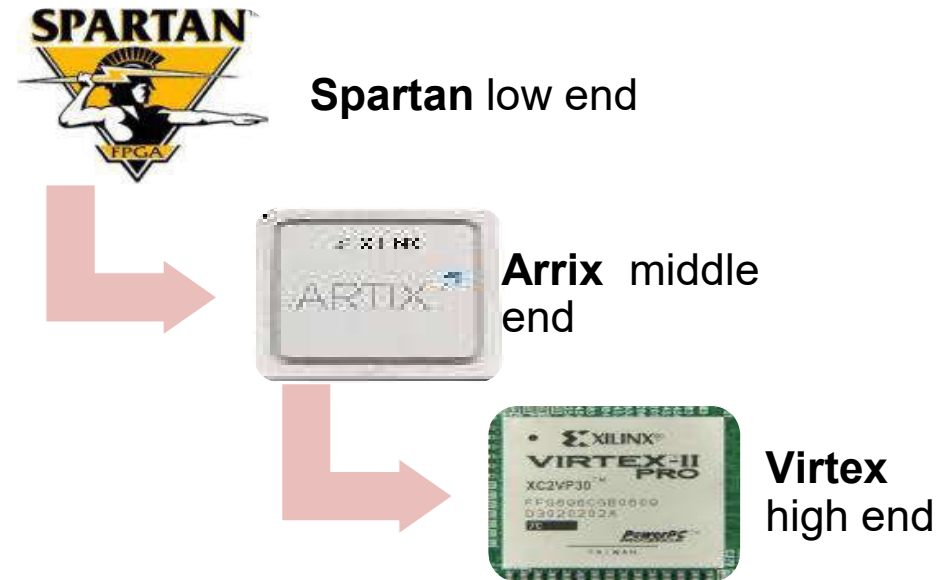
- Xilinx
- Altera
- Actel
- Lattice
- QuickLogic



# Cấu trúc FPGA của một số hãng

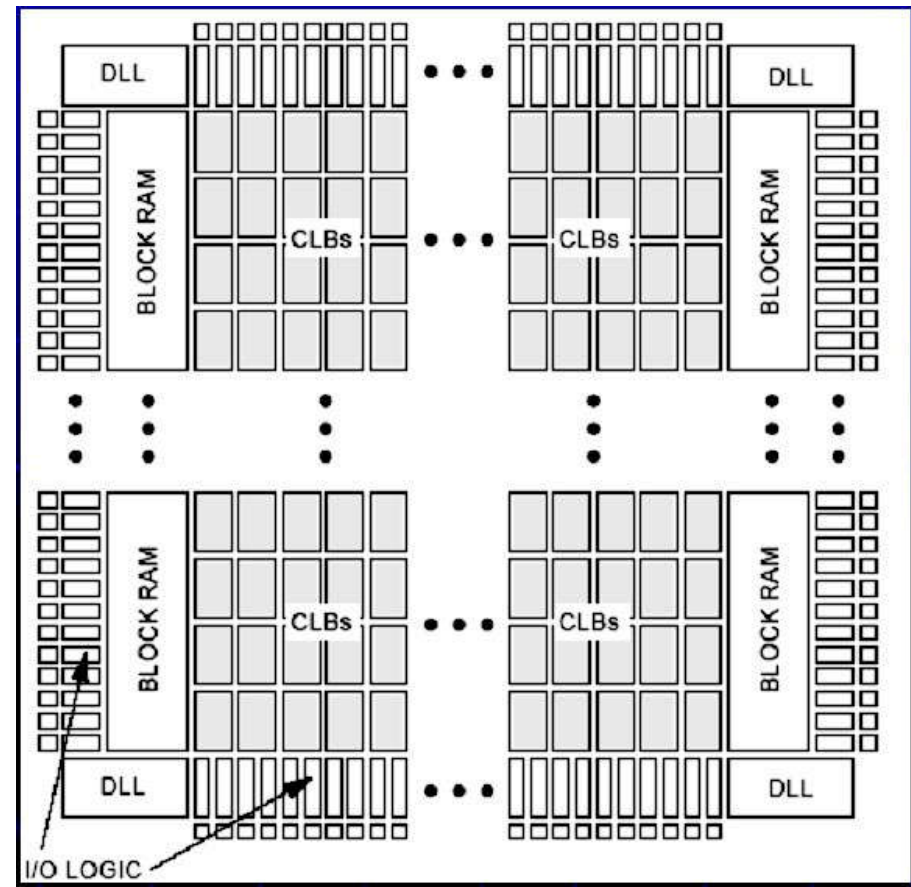
- Xilinx

- Virtex-II/Virtex-4: Feature-packed-high-performance SRAM-based FPGA
- Spartan 3: low-cost feature reduced version
- CoolRunner: CPLDs



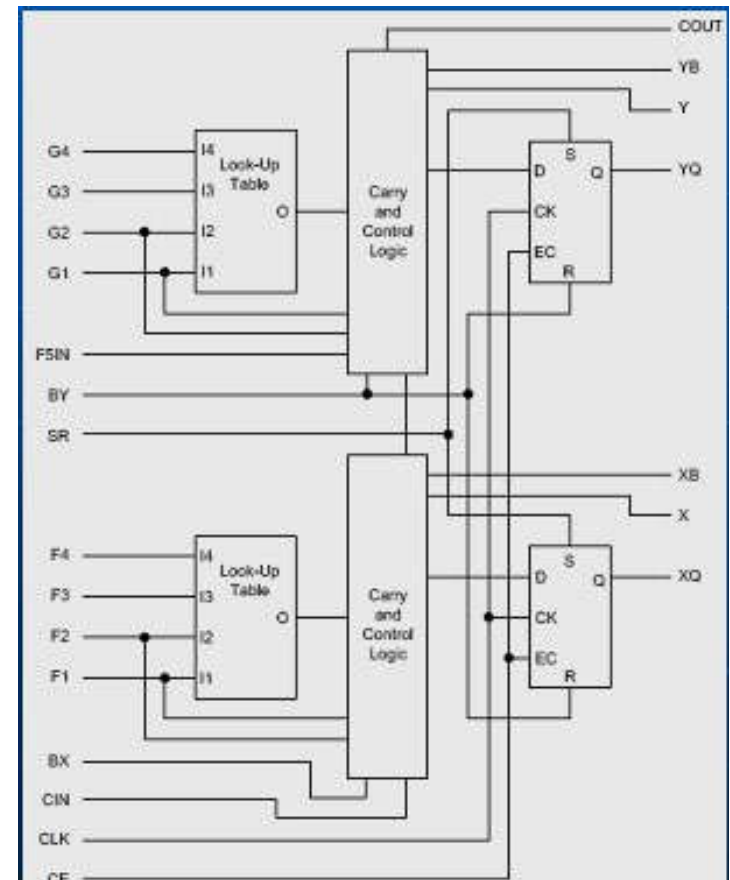
# Xilinx FPGA Spartan IIE

- Gồm mảng 2 chiều các khối lập trình được CLB
- Một CLB gồm 4 logic-cell
- Điều khiển lập trình bằng SRAM
- I/O logic block
- DLL(Delay Lock Loops-vòng khoá trễ): điều khiển xung Clock để giảm trễ, tạo trễ, đồng bộ giữa các tín hiệu Clock



# Xilinx FPGA Spartan IIE

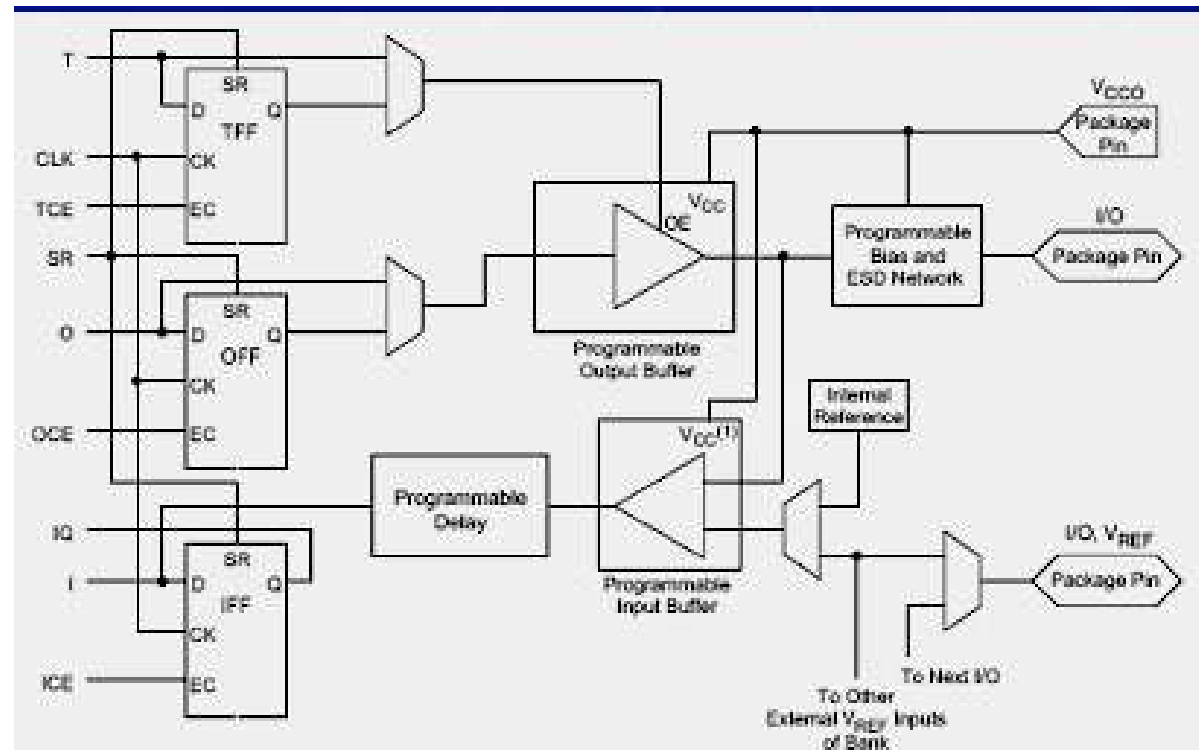
- Logic cell
  - Mỗi Logic-cell gồm 2 LUT giống nhau
  - Mỗi LUT gồm 4 chân đầu vào, tín hiệu điều khiển và các D-FlipFlop
  - 2 bộ điều khiển ngõ ra 2 trạng thái BUFT



# Xilinx FPGA Spartan IIE

- I/O Logic Block

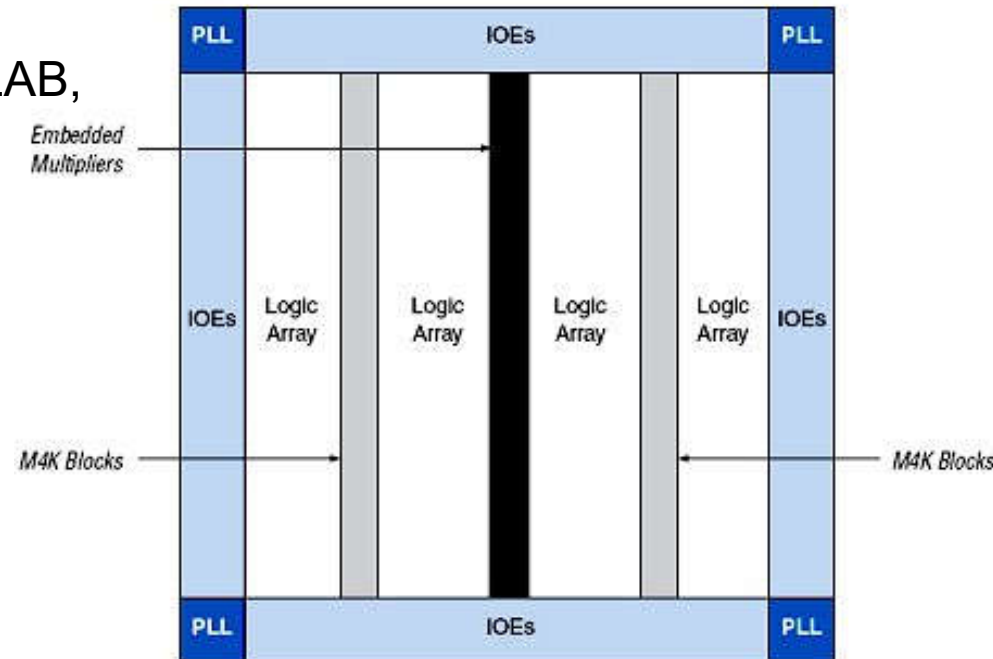
- Gồm 3 FF chung Clock và các tín hiệu Clock Enable (CE) điều khiển độc lập cho từng FF
- Tín hiệu vào qua 1 bộ đệm, tín hiệu ra qua bộ đệm 3 trạng thái theo các chuẩn bộ nhớ/giao tiếp Bus





# Altera FPGA Cyclone

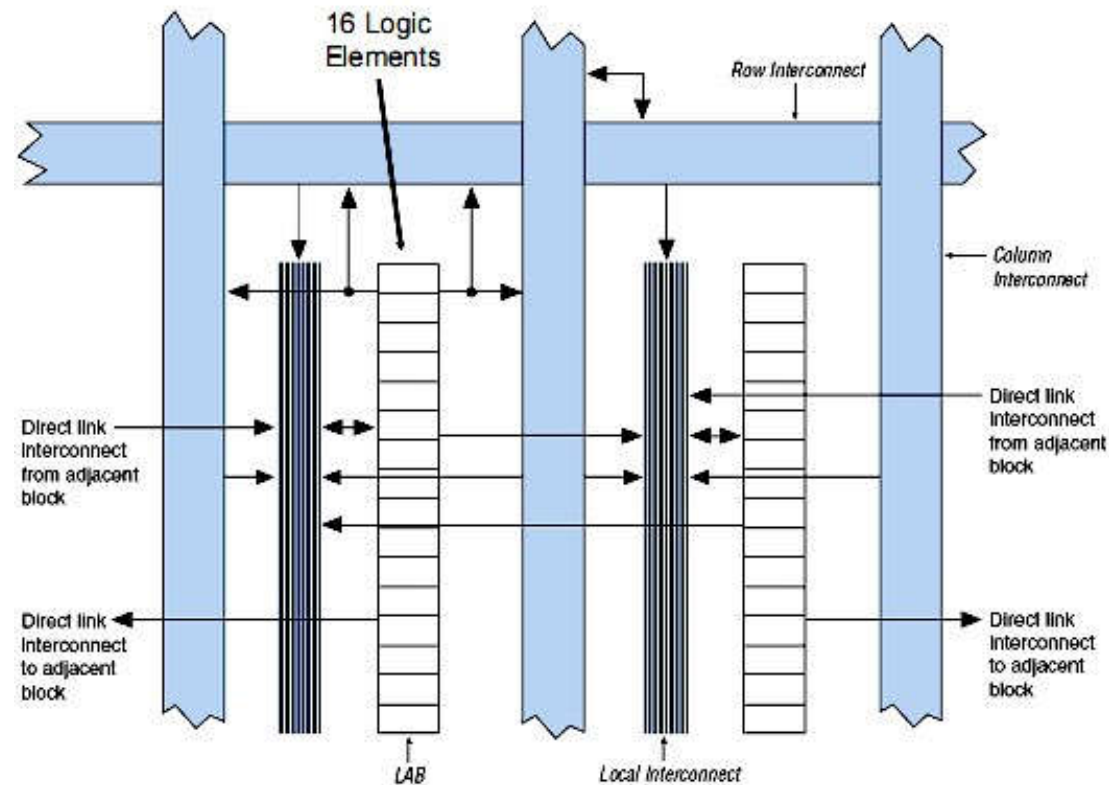
- Gồm một mảng lớn các dãy Block lập trình LAB, kết nối với nhau bởi PIA - Programmable Interconnect Array
- Gồm các khối
  - Các dãy logic chứa các bảng LUT
  - Bộ nhớ dạng khối - M4K
  - Bộ ghép kênh tích hợp sẵn
  - Khối vào ra IOE
  - vòng khoá pha PLL cung cấp xung nhịp, tạo sự dịch pha cho các yêu cầu hỗ trợ đầu ra tốc độ cao



# Altera FPGA Cyclone

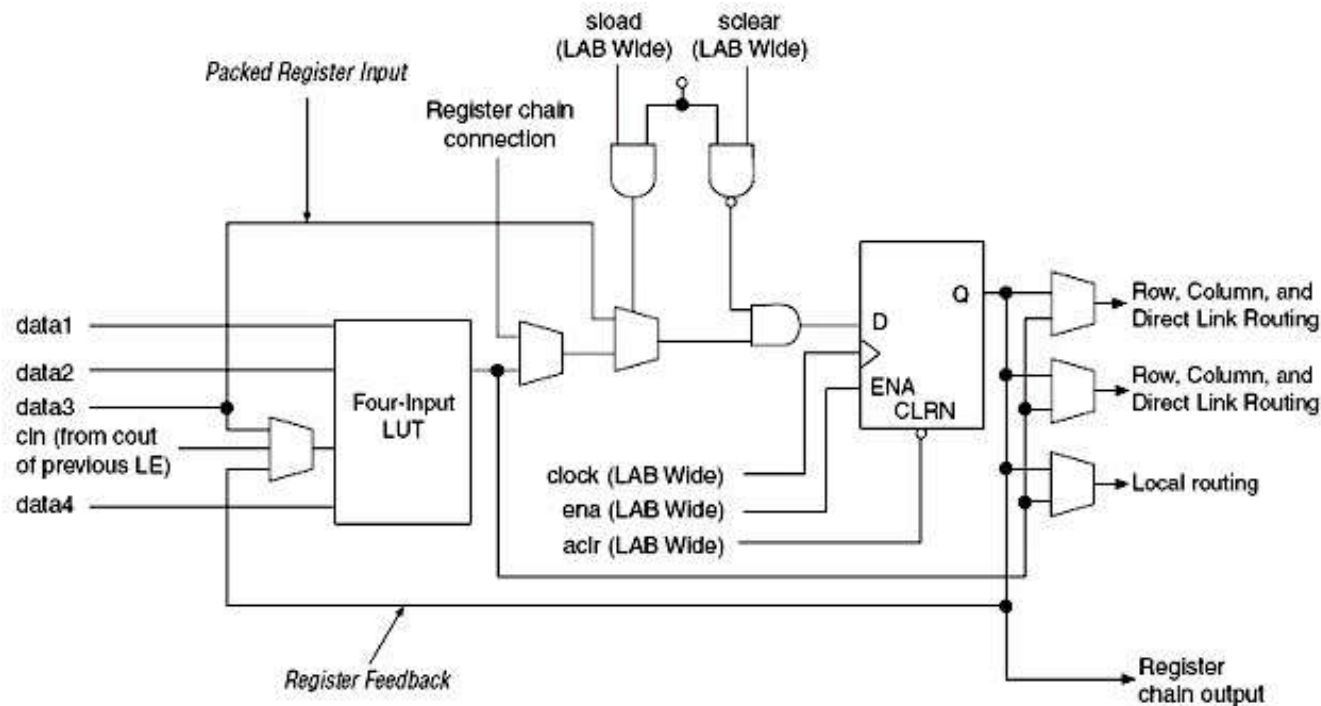
- LAB

- Kích thước 1 LAB tương đương dung lượng nhớ 4K
- Các đường kết nối hàng và cột giúp đẩy nhanh tốc độ kết nối giữa các LE trong LAB và giữa các LAB với nhau



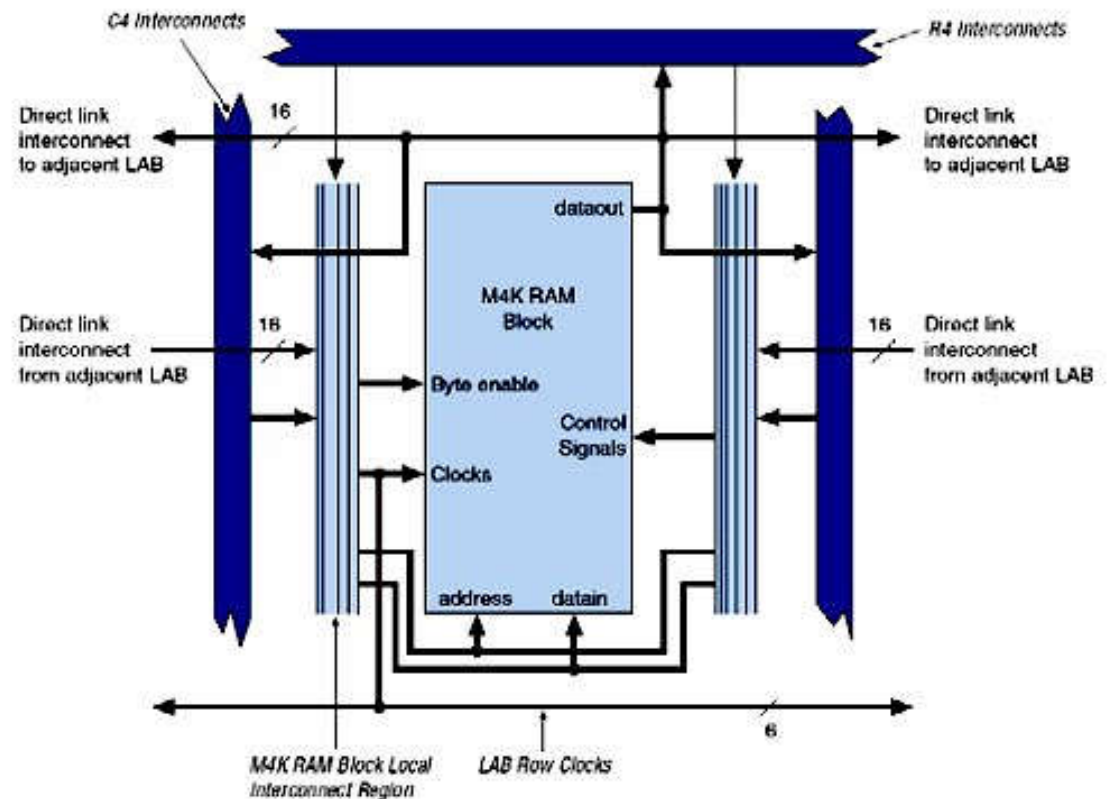
# Altera FPGA Cyclone

- LE - Logic Elements
  - 16 khối LE tạo nên một khối LAB



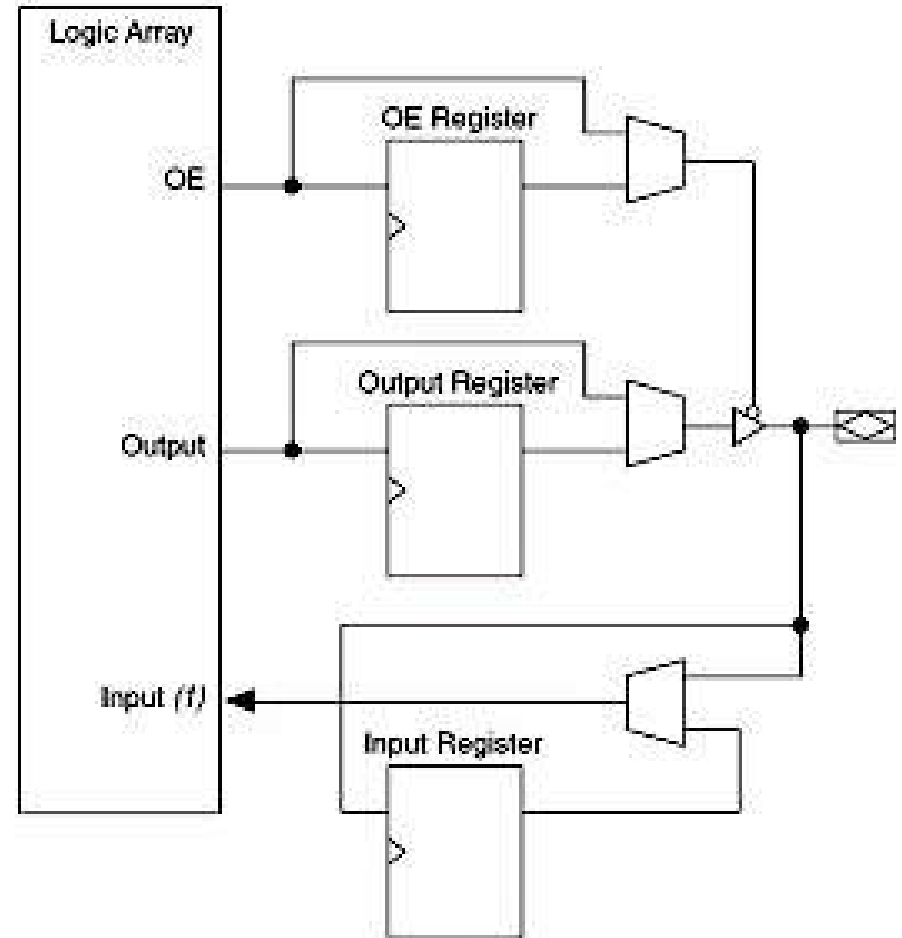
# Altera FPGA Cyclone

- Bộ nhớ M4K
  - Tích hợp sẵn thường có 1 hoặc 2 khối bộ nhớ dung lượng M4K
  - Thực hiện thanh ghi dịch và nhiều kiểu bộ nhớ khác nhau như RAM, ROM, các bộ đệm FIFO, thanh ghi dịch



# Altera FPGA Cyclone

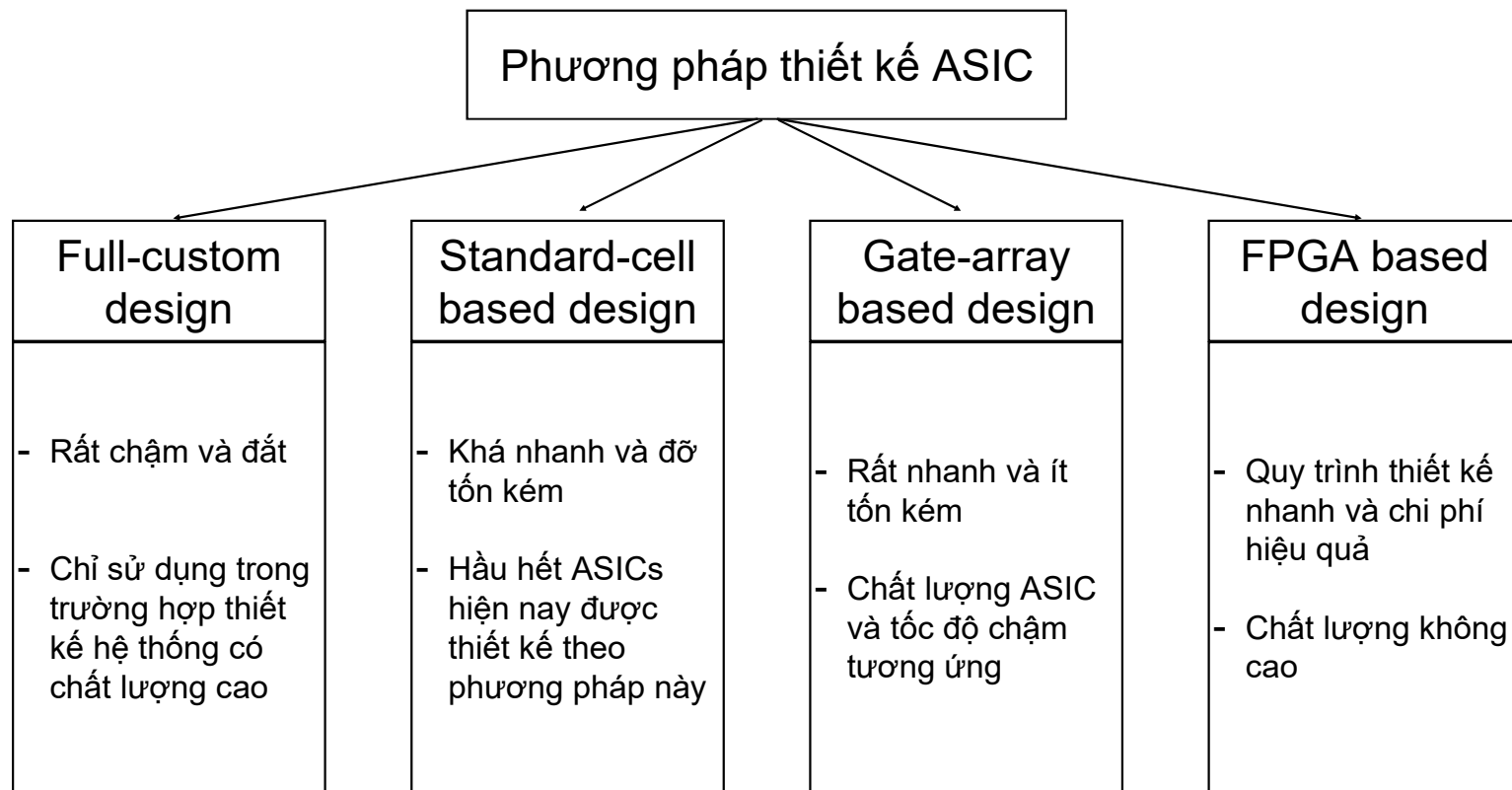
- Khối vào ra IOE
  - Mỗi phân tử vào ra IOE chứa 1 bộ đệm vào ra 2 hướng và 3 thanh ghi để truyền dẫn tín hiệu theo 2 hướng



# Giới thiệu công nghệ IC lập trình

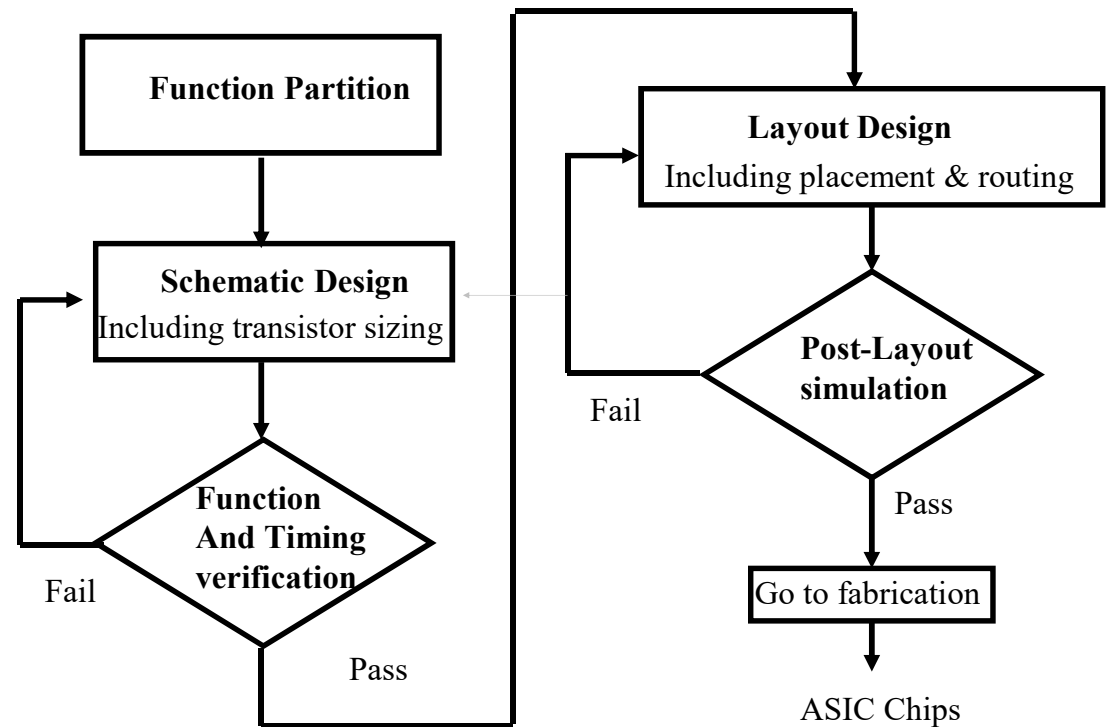
- Giới thiệu chung
- Phân loại IC khả trình
- Kiến trúc IC khả trình
- Quy trình thiết kế IC khả trình

# Các hướng tiếp cận thiết kế ASIC



# Full-custom design

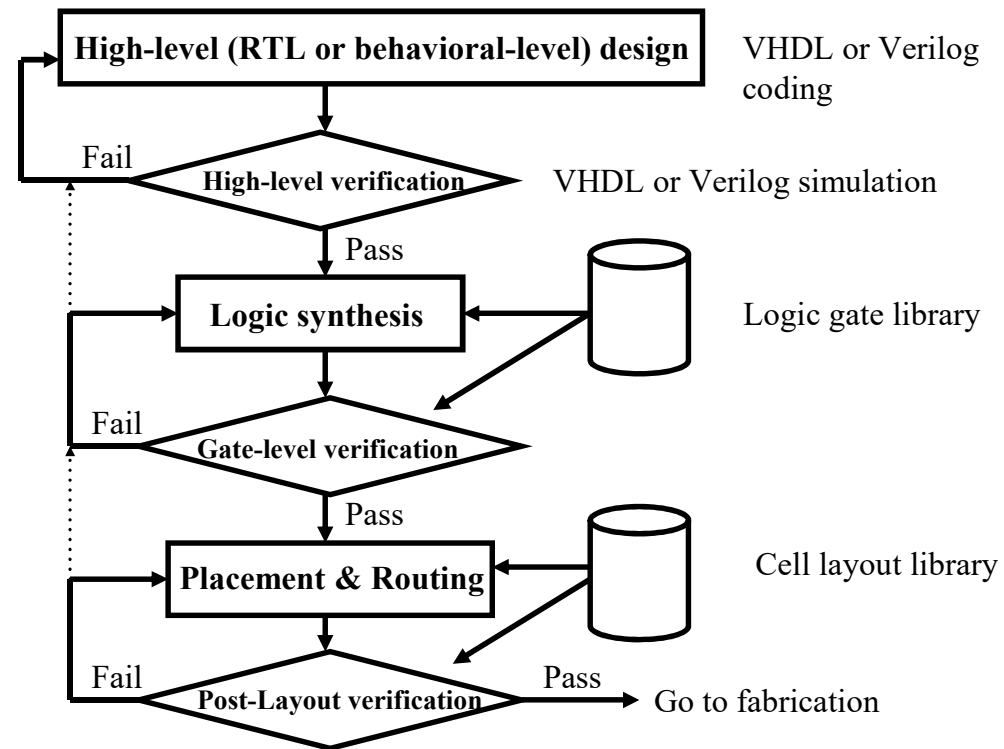
- Thiết kế chip từ những phần tử đơn giản nhất
- Thiết kế được từ các phần tử nhỏ đến cả IC
- Rất linh hoạt, có khả năng tối ưu
- Tốn nhiều công sức, chi phí cao





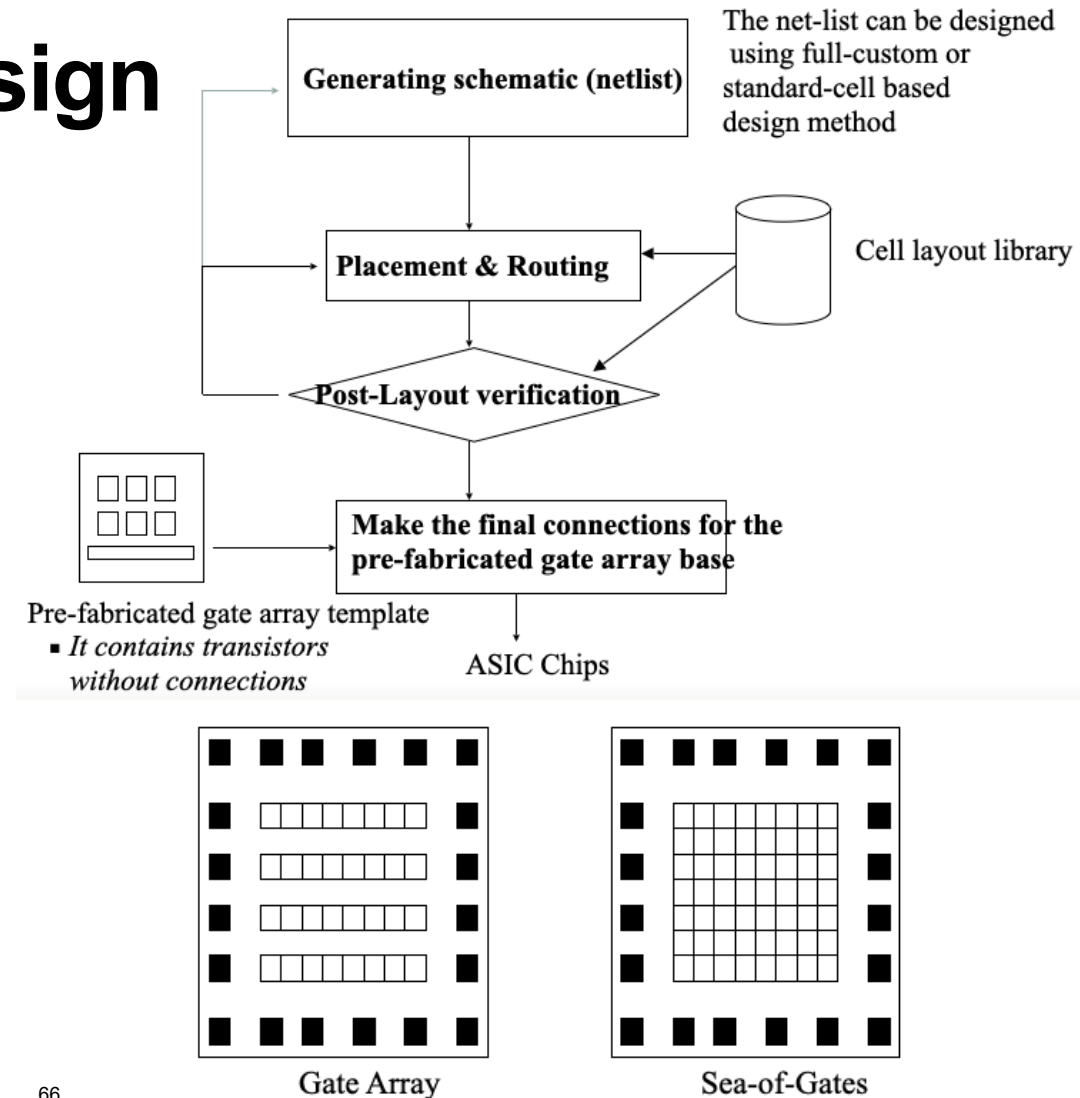
# Standard-cell based design

- Có khả năng tự động thực hiện
- Cần sự hỗ trợ từ thư viện
- Do sử dụng thư viện nên tiết kiệm được thời gian thiết kế, giảm nhiều công sức



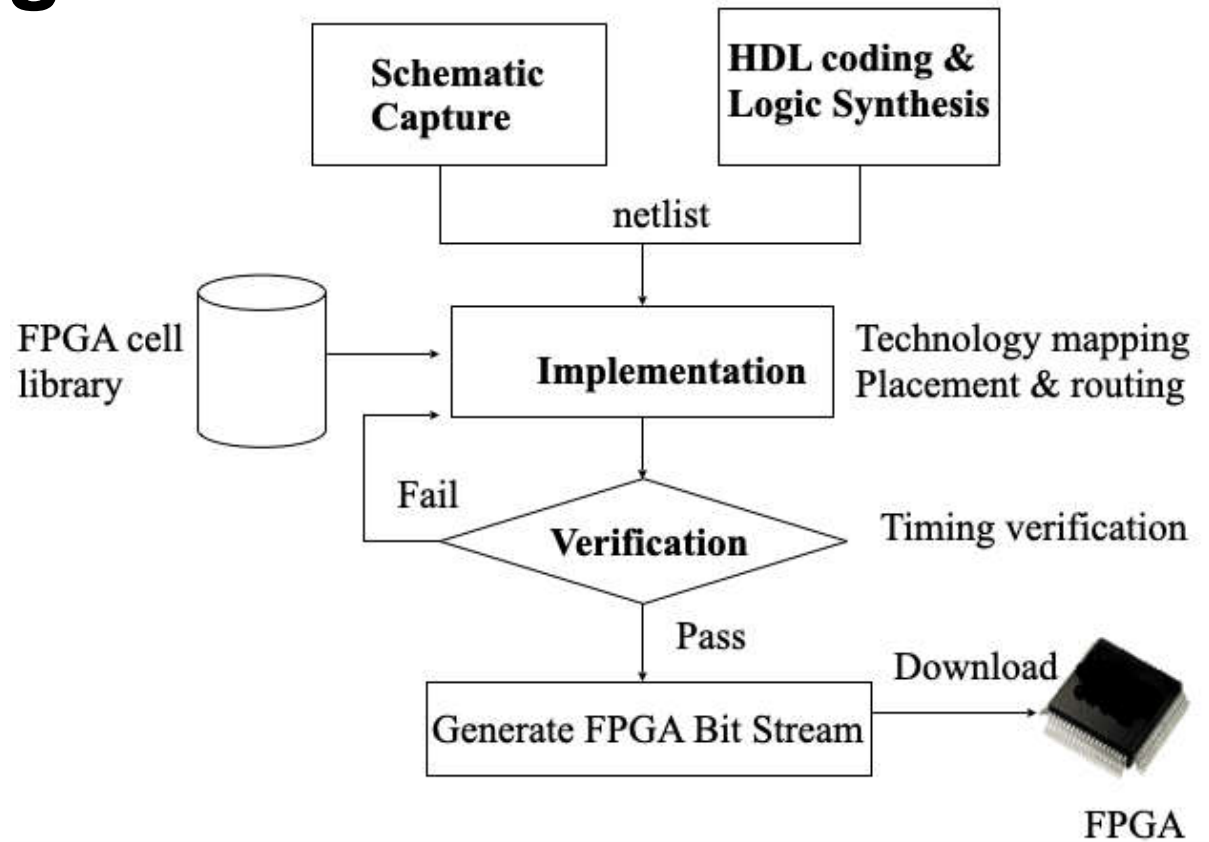
# Gate Array based design

- Hướng tiếp cận này nhanh hơn so với Standard-cell based design do một số thành phần của IC đã được thực hiện xong
- Một số thành phần của chip (transistor) đã có sẵn. Thành phần còn lại (wire) sẽ được thiết kế, chế tạo thêm để hoàn thiện mạch
- Ưu điểm là tiết kiệm
- Nhược điểm là hệ thống không được tối ưu như 2 phương pháp đầu



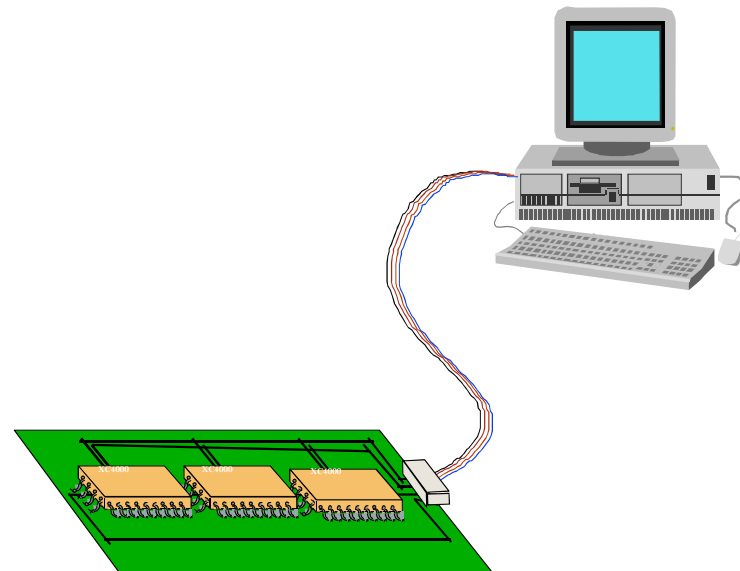
# FPGA based design

- Hướng tiếp cận nhanh nhất do phần lớn các thành phần của IC đã được chế tạo xong
- Việc thay đổi các kết nối được thực hiện bằng phần mềm



# Quy trình thiết kế

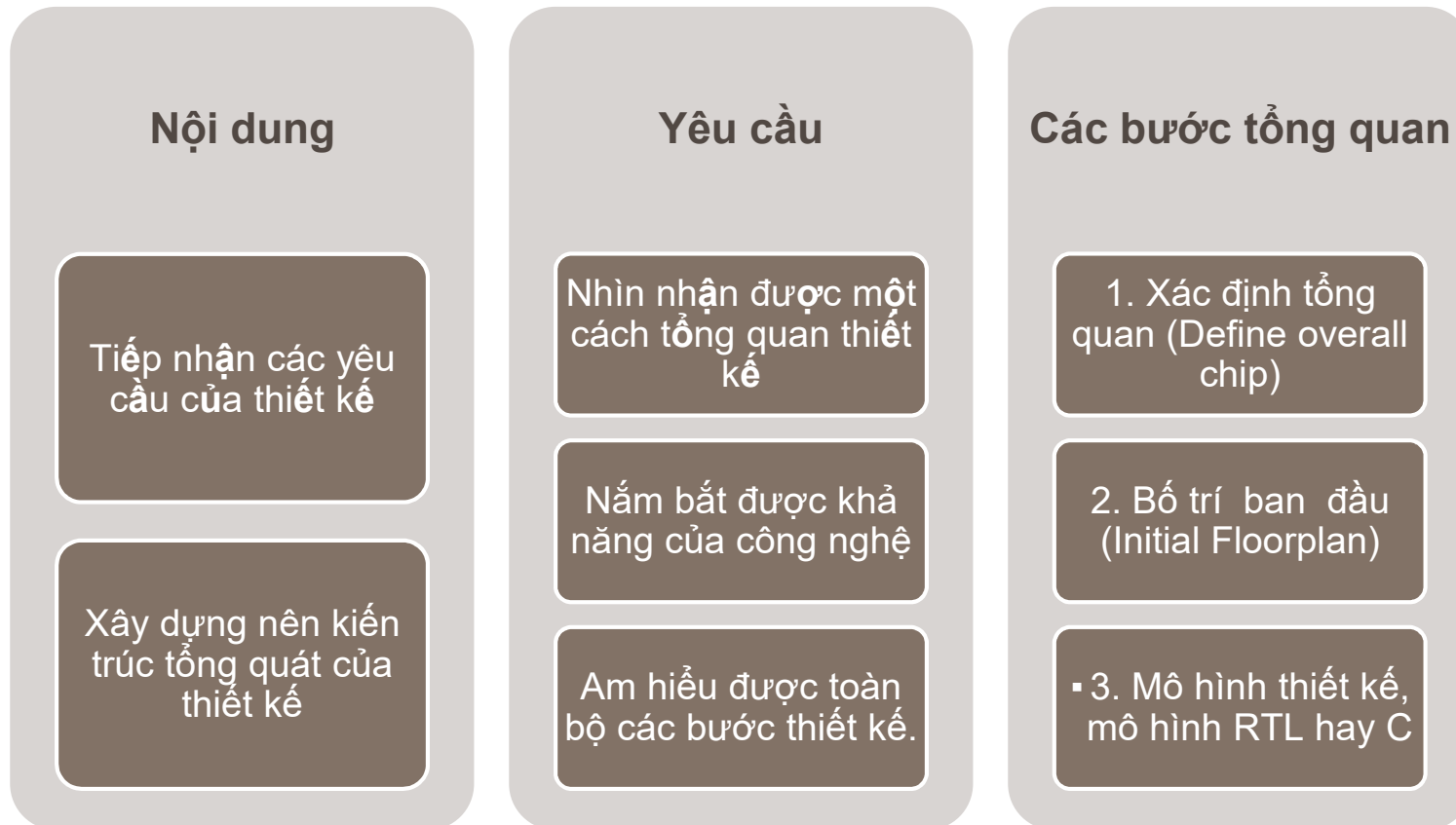
- Xây dựng mạch nguyên lý, lưu đồ VHDL hoặc Verilog
- Thực hiện mạch bao gồm xác định khối chức năng, đi dây nối, kết hợp phân tích lưu đồ thời gian, sắp xếp vị trí,...
- Download xuống mạch phần cứng



# Quy trình thiết kế

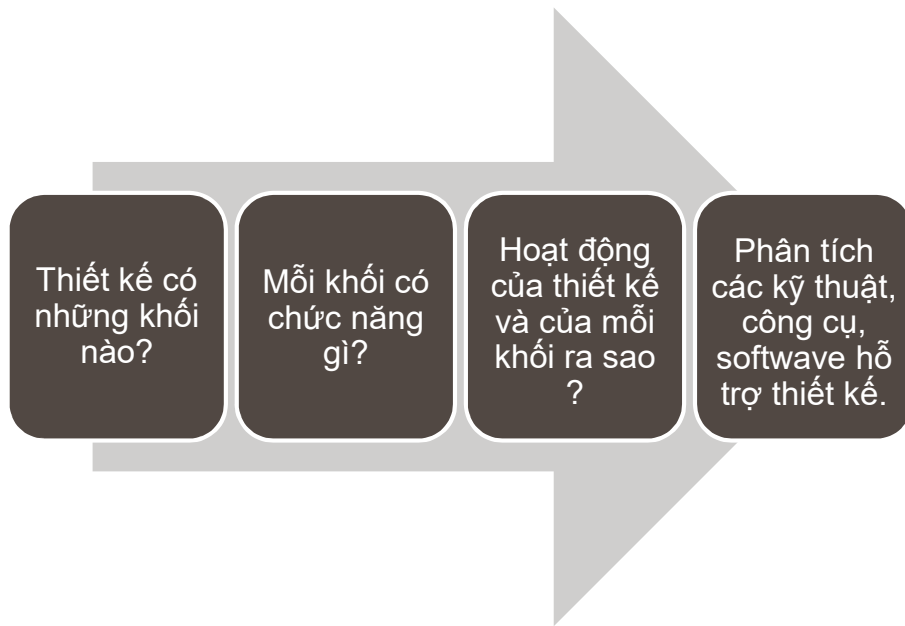
- Bước 1: Thiết kế kiến trúc (Architecture design)
- Bước 2: Thiết kế logic (Logic Design)
- Bước 3: Thiết kế mạch (Circuit Design)
- Bước 4: Thiết kế vật lý cho các cell đặc trưng (Mask design)
- Bước 5: Thiết kế vật lý (Physical design)

# Bước 1: Thiết kế kiến trúc



# Bước 1: Thiết kế kiến trúc

- Xây dựng tổng quan



## Bước 2: Thiết kế logic



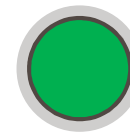
### Nhiệm vụ

- Mô phỏng tổng thể các chức năng logic
- Tối ưu một thiết kế.



### Yêu cầu

- Tạo ra một mô hình hành vi HDL
- Định nghĩa những bộ vector để kiểm tra
- Biên dịch mạng kết nối (netlist)
- Tối ưu hóa thiết kế



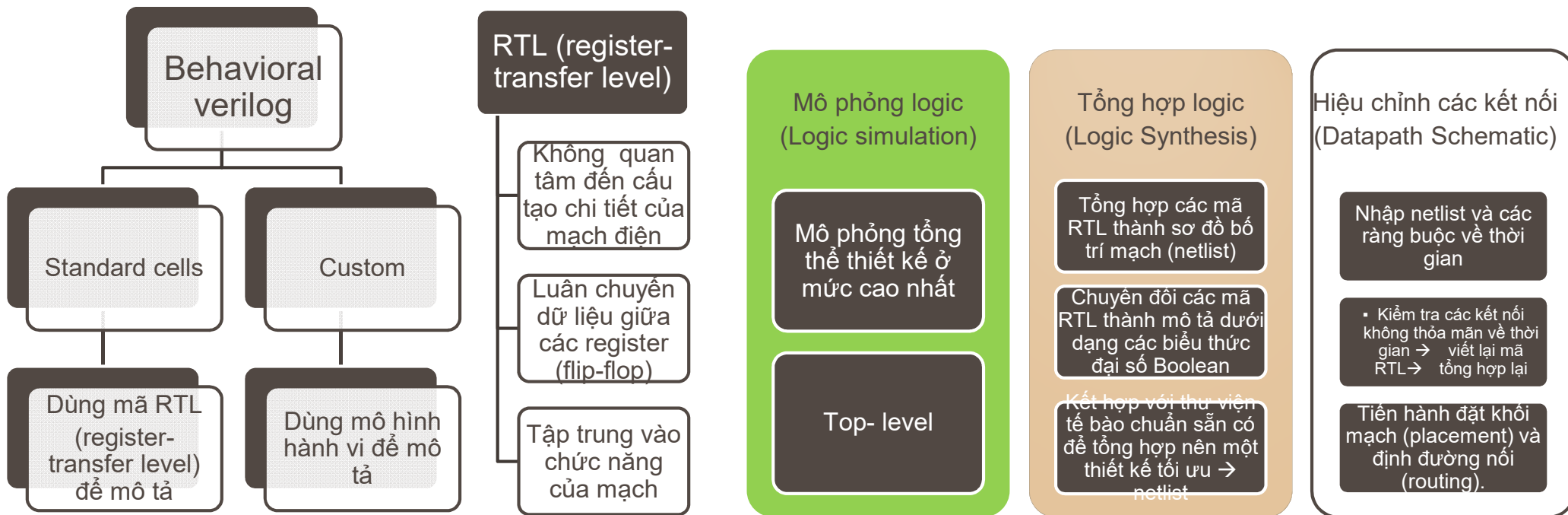
### Thực hiện

- Thiết kế mô hình hành vi (Behavioral verilog)
- Mô phỏng logic (Logic simulation).
- Tổng hợp logic (Logic Synthesis).
- Hiệu chỉnh các kết nối (Datapath Schematic)

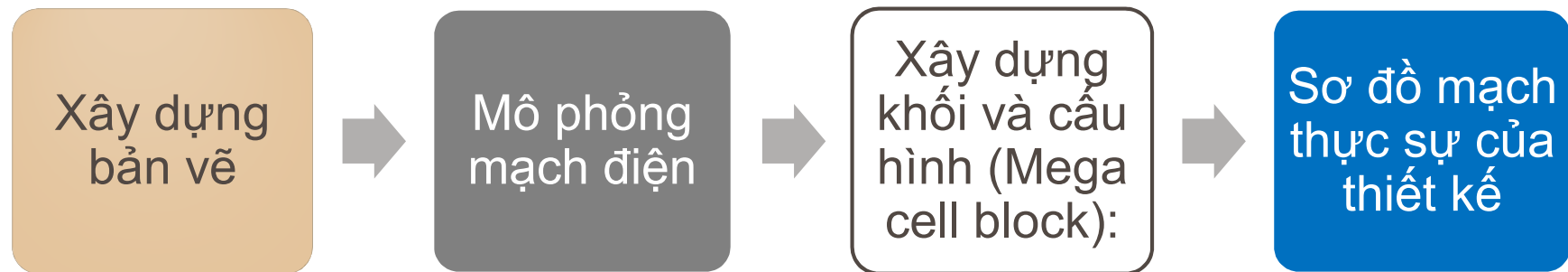


# Bước 2: Thiết kế logic

- Thiết kế mô hình hành vi

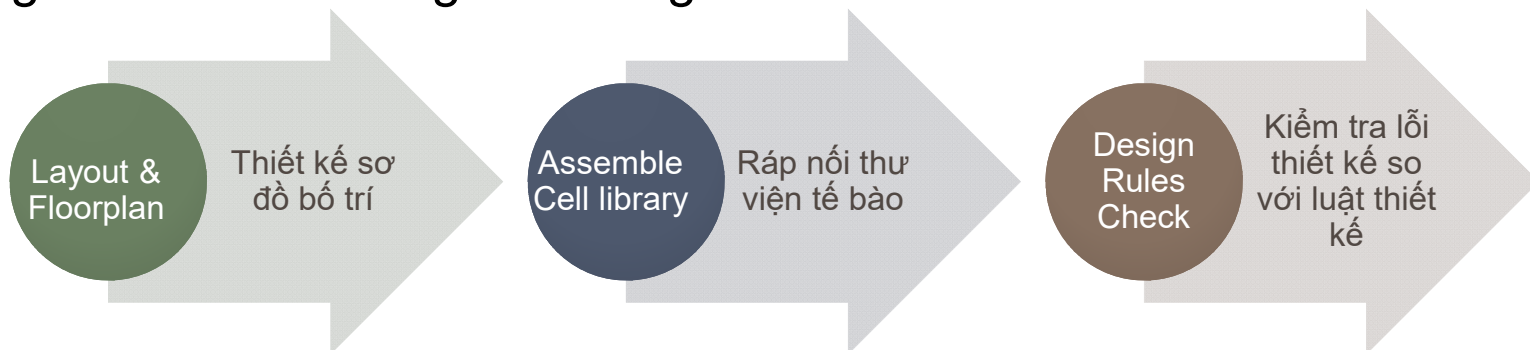


## Bước 3: Thiết kế mạch



# Bước 4: Thiết kế vật lý cho các cell đặc trưng

- Thiết kế ở cấp độ transistor
- Layer design
  - Chuyển từ sơ đồ lý luận sang sơ đồ mạch vật lý
  - Nhà sản xuất sẽ đổ các lớp mẫu đã được quy định trên cơ sở các lớp mặt nạ (mask layers) để được các lớp vật lý biểu thị các transistor, các cổng logic và các kết nối giữa chúng.



# Bước 5: Thiết kế vật lý (Physical design)

- Thực hiện ở cấp độ chip trong quy trình SoC
- Sắp xếp các mạch thiết kế trên một vùng diện tích nhất định dựa và thiết kế lý luận của chip (gate-level netlist)
- File GDS của chip được đưa xuống nhà sản xuất để làm ra thành phẩm



## Floorplan

Sắp xếp thiết kế  
Sắp xếp các vùng  
đặt các IO cells,  
các power pad,  
standard cells,  
cung cấp nguồn  
cho các khối.



## Place & Route

Đặt và nối dây  
các cell tầng trên  
cùng (chip level)



## Post Route View, Edit and plot

xem xét, kiểm tra,  
và chỉnh sửa →  
thực hiện biên  
dịch sang định  
dạng file thiết kế  
hệ thống  
GDS(Graphic  
Design System).



## Final Check

DRC (design rule  
check), LVS (layout  
vesus Schematic),  
ERC (Electrical Rules  
Check)