EDA Technology

Course Codez	A040523s	Course Type	Core	Course Nature	Required
Credit	3.0		Total Credit Hours	48	
Course offered by	School of Electronics and Information		Teaching Staff	Department of Electronic Engineering	

School of Electrical and information Engineering

Hangzhou Dianzi University

Instructors

- Jiye Huang(黄继业）

E-mail: hjynet@163.com

Mobile:

- Office：

Course Website

- http://www.icourses.cn/mooc/

Course Website

- http://www.icourses.cn/mooc/

QQ discussion group

Prerequisites

- Digital Logic Circuit （数字电路）
- **C language programming**

Course Objectives

- The course objective (1): establish the design thinking, train the basic design ability, develop the creativity and innovation ability, know the design techniques of the modern digital systems and the verification techniques,and address the economic globalization and the challenges of the knowledge economy.
- The course objective (2): use Verilog HDL/VHDL to design the digital system based on understanding the general concept of the EDA technology and by utilizing the basic approaches and procedures of the EDA design.

Course Objectives

- The course objective (3): use the EDA tool such as Quartus II and the FPGA hardware development system to design the relatively complicated digital circuits and systems.
- The course objective (4): have a good master of the basic experimental skills, and teach the students to make the experimental plan, undertake the experiments and analyze and explain the data.

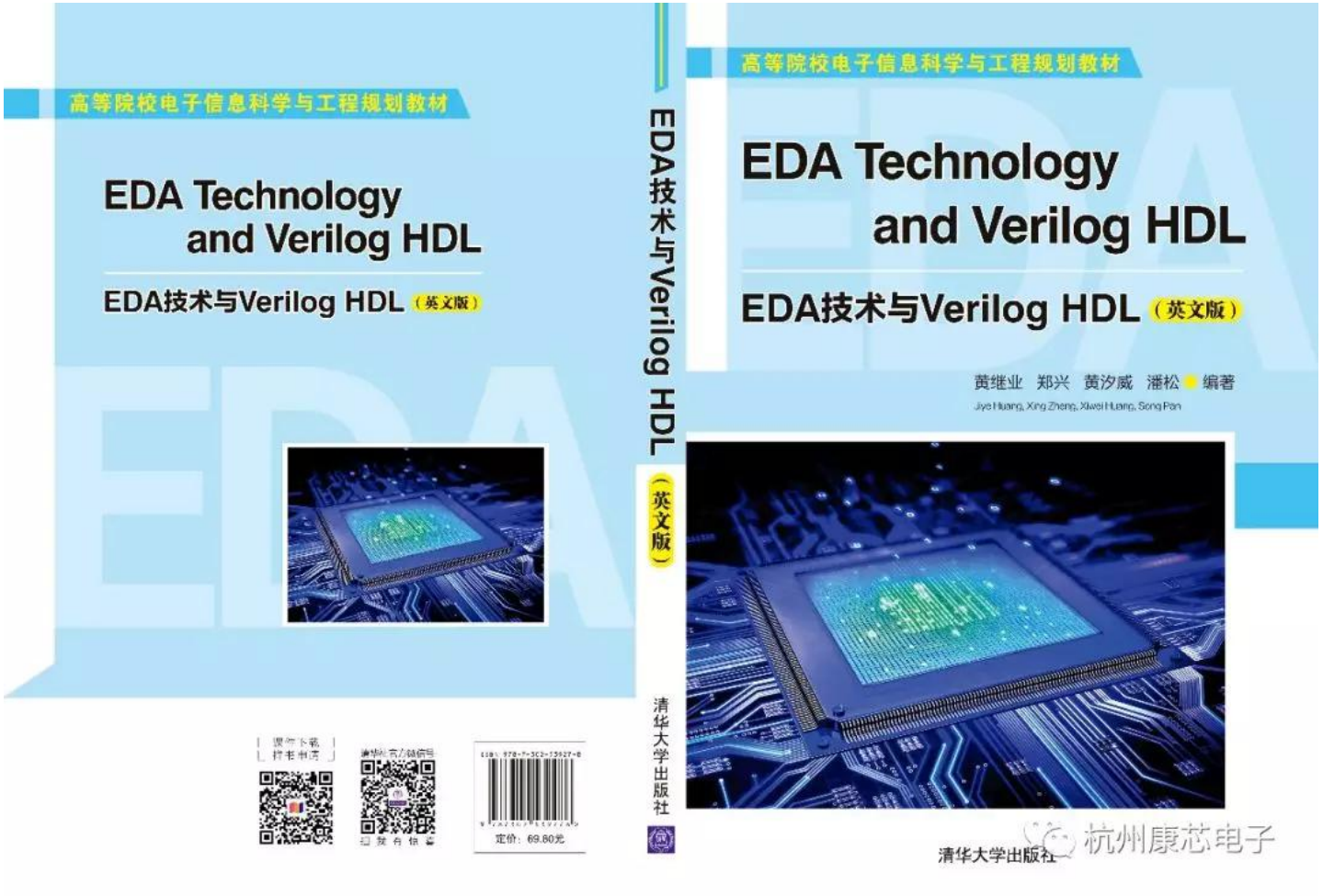
Graduation requirement

- 1-4：understand the system concept and its reflection in the circuit area; Can use the specialized knowledge to fix the complicated problem of the electronics information engineering.
- 3-1 can use the specialized knowledge and designing index to confirm the design plan of the circuit and system.
- 4-2 can choose the research route and design the promising experimental solution.

Course Overview

- **Total Credit Hours**：48
- **Lecture Hours**：30
- **Experiment (Computer) Hours**：**18**
- **Credit**：3
- Homeworks 9%
- Class Attendance 6%
- Experiment 15%
- Final examination 70%

Course Textbook



Course Textbook and Reference Book

[1] Jiye Huang，Song Pan《EDA technology—Verilog HDL》(Sixth Edition) Science Press，2018.

[2] Michael D.Ciletti. 《Advanced Digital Design with the Verilog HDL (2nd Edition)》 Prentice Hall. 2010.

[3] Donald E.Thomas，Philip R.Moorby《Verilog Hardware Description Language 》. Kluwer AcademicPublishers. 2002.

Experiment

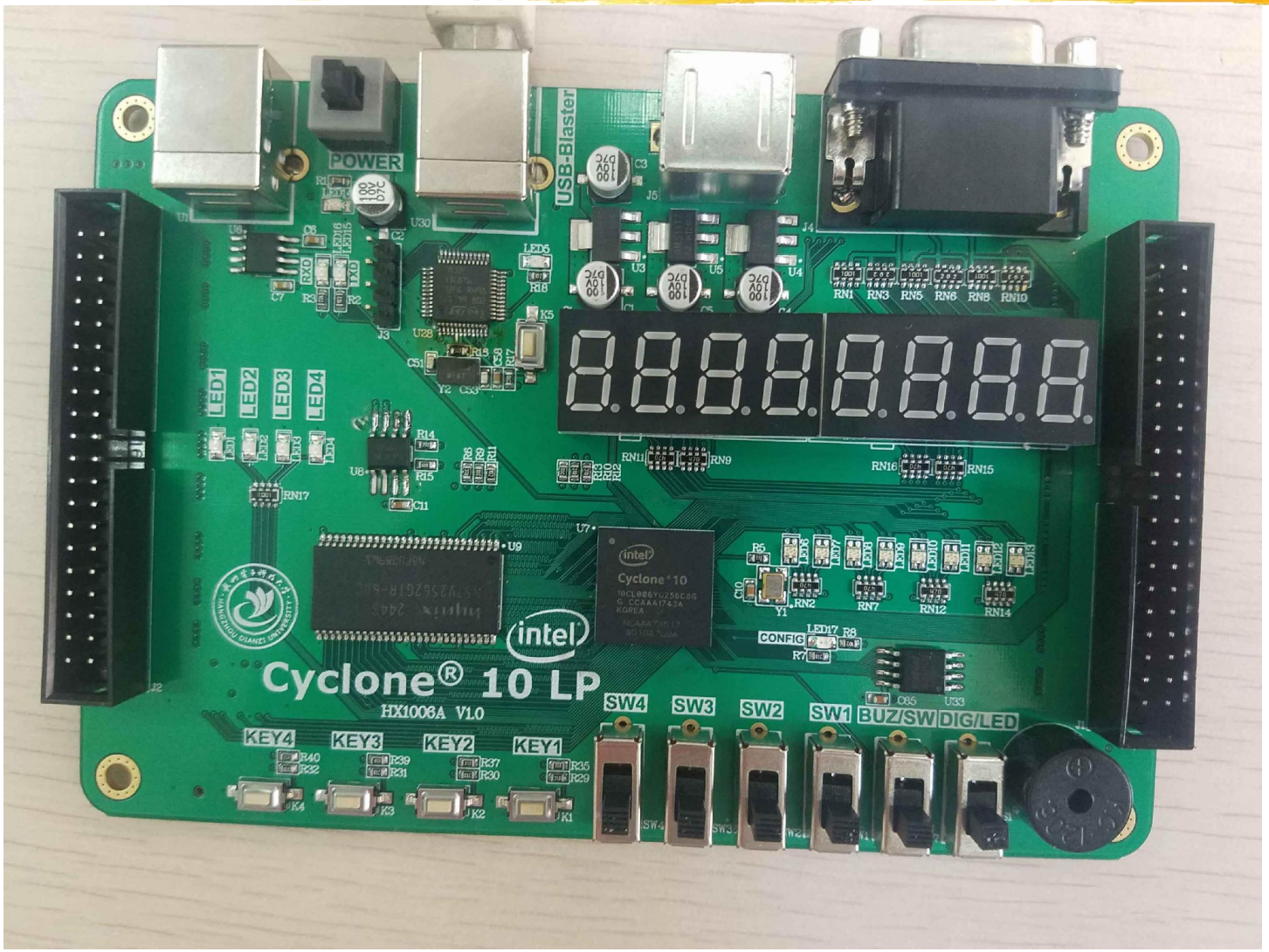
- For the miniproject, you will do

	Experiment 1：half adder and full adder	3			4		
	Experiment 2：The design of modulus controllable counter	3			6		
	Experiment 3：The signal generator of sinusoidal waveform	3			8		
	Experiment 4：The design of VGA display control circuit	3			10		
	Experiment 5：The design of sequence detector	3			12		
	Experiment 6：The design of music playing circuit	3			14		

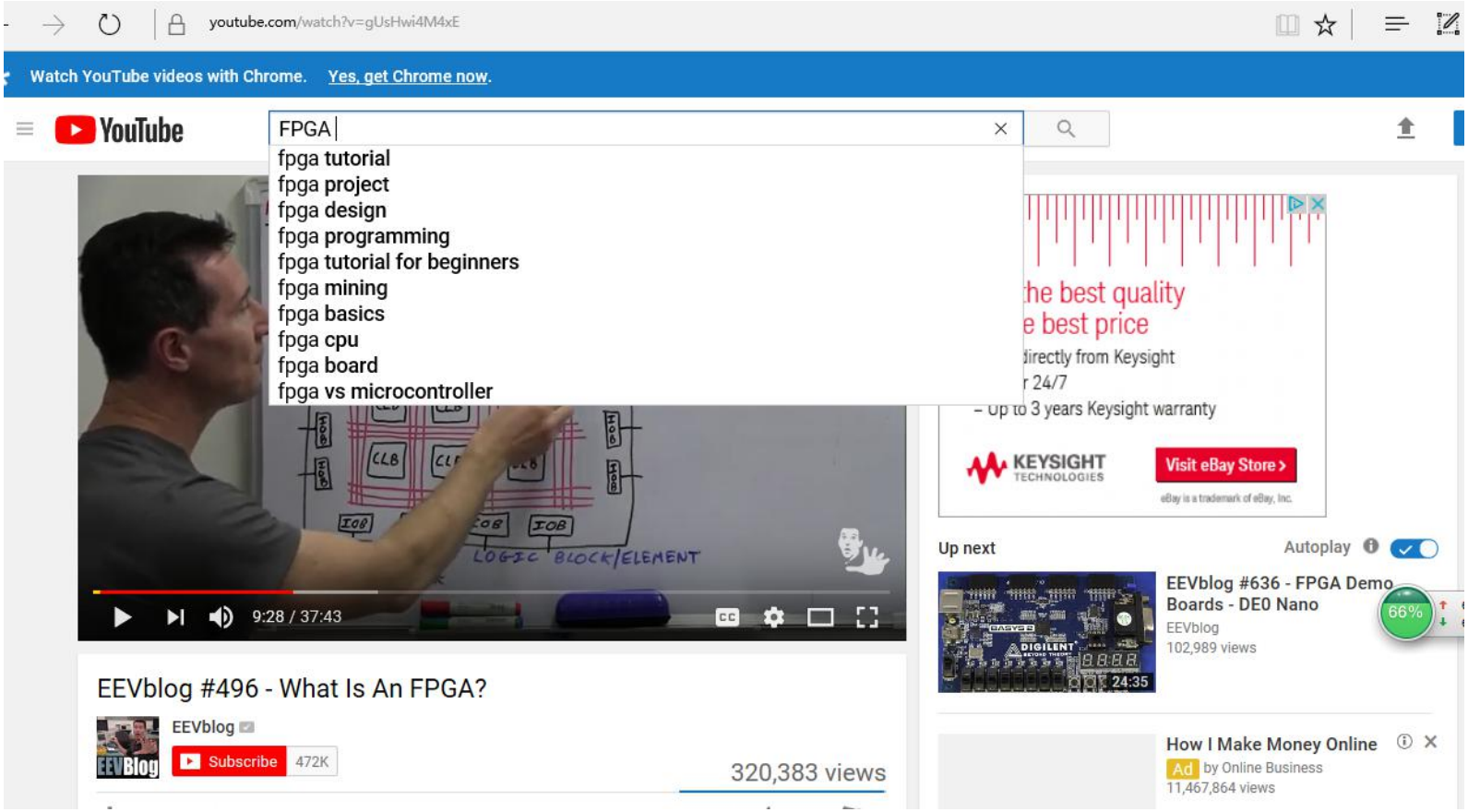
Lab Location

- Use your own computer for the first Experiment
- Install the software-Quartus on your own computer
- 2 教中楼 237 after week 10
- Where and when for the second and third Experiment ?

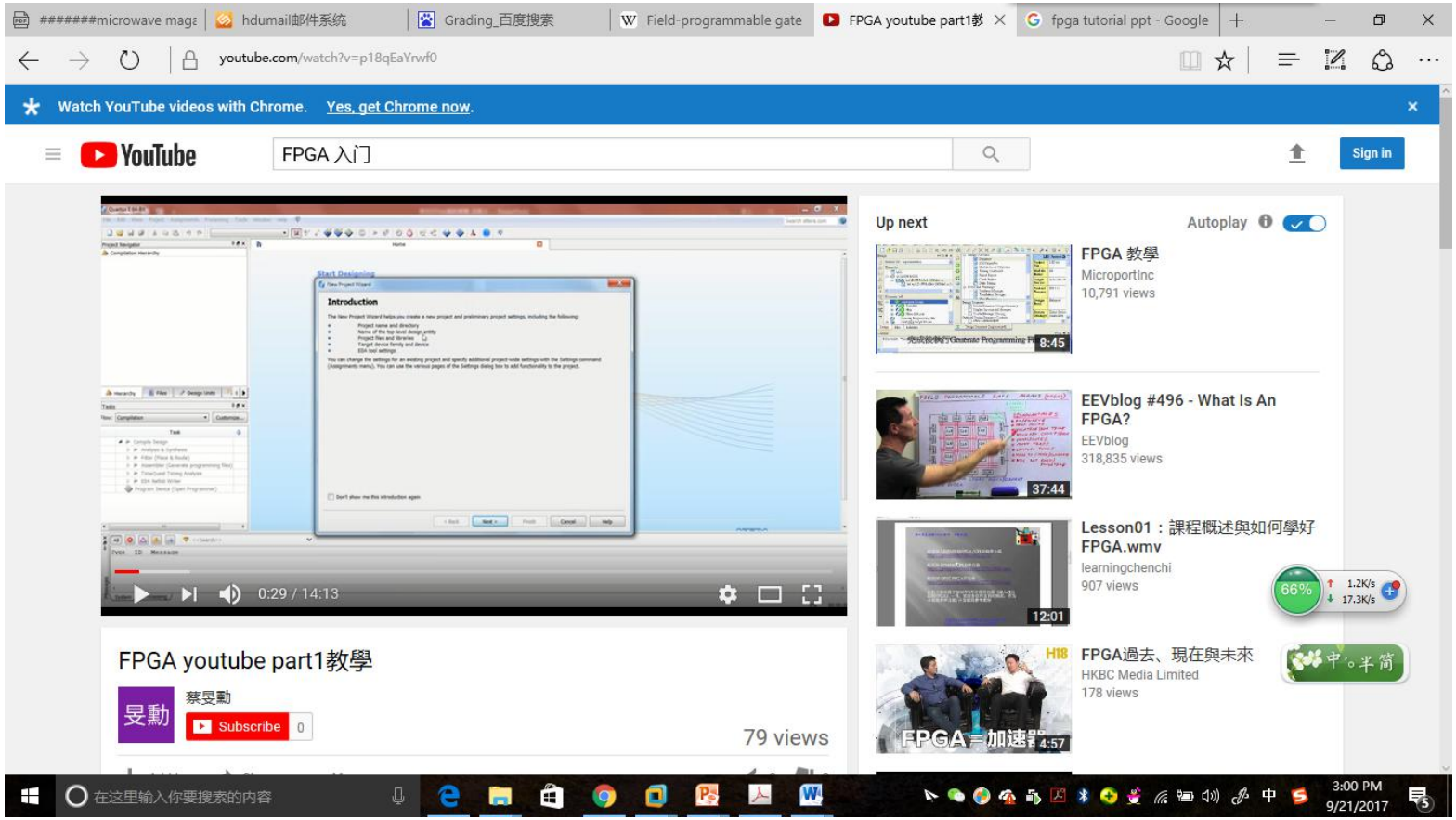
Experimental board



Course Reference Vedio



Course Reference Vedio



EDA Technology

Chapter 1

EDA Introduction

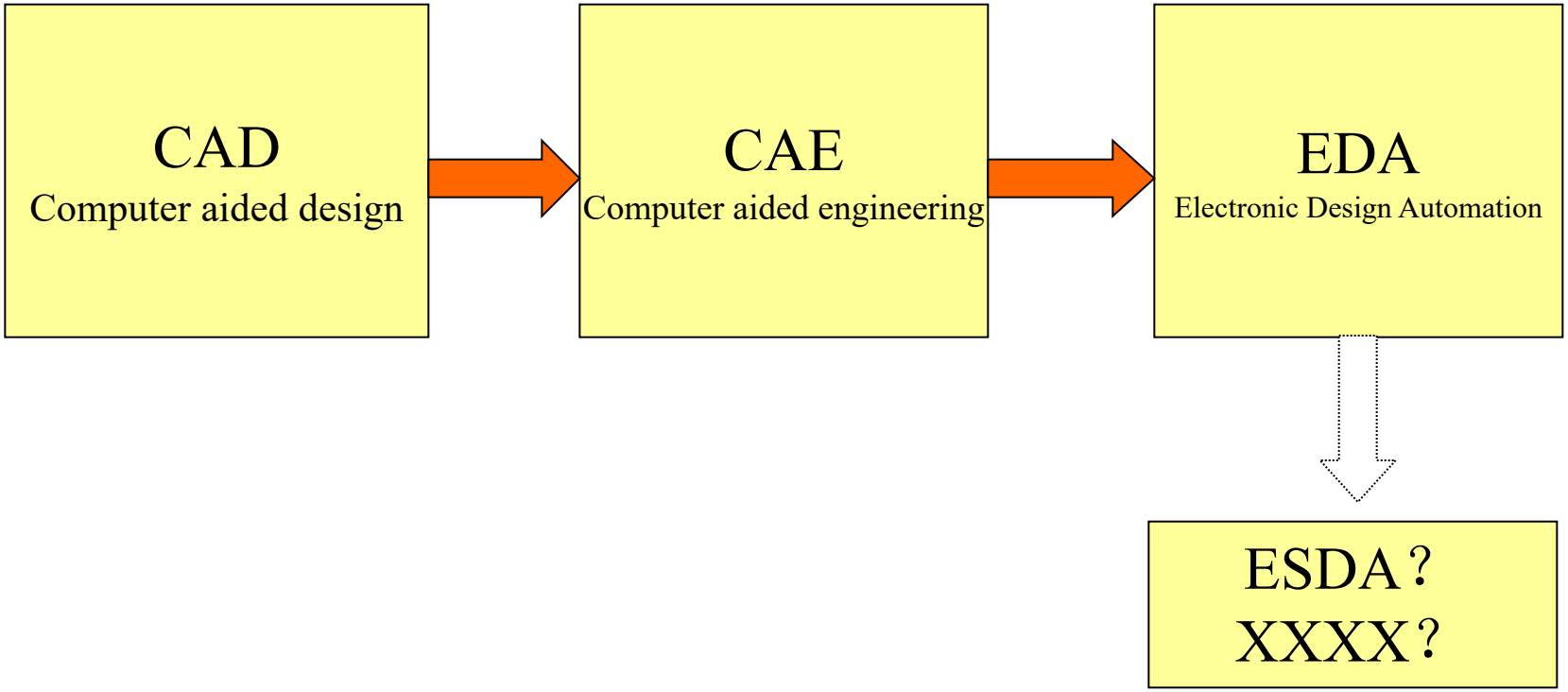
EDA?

- EDA： Electronic Design Automation

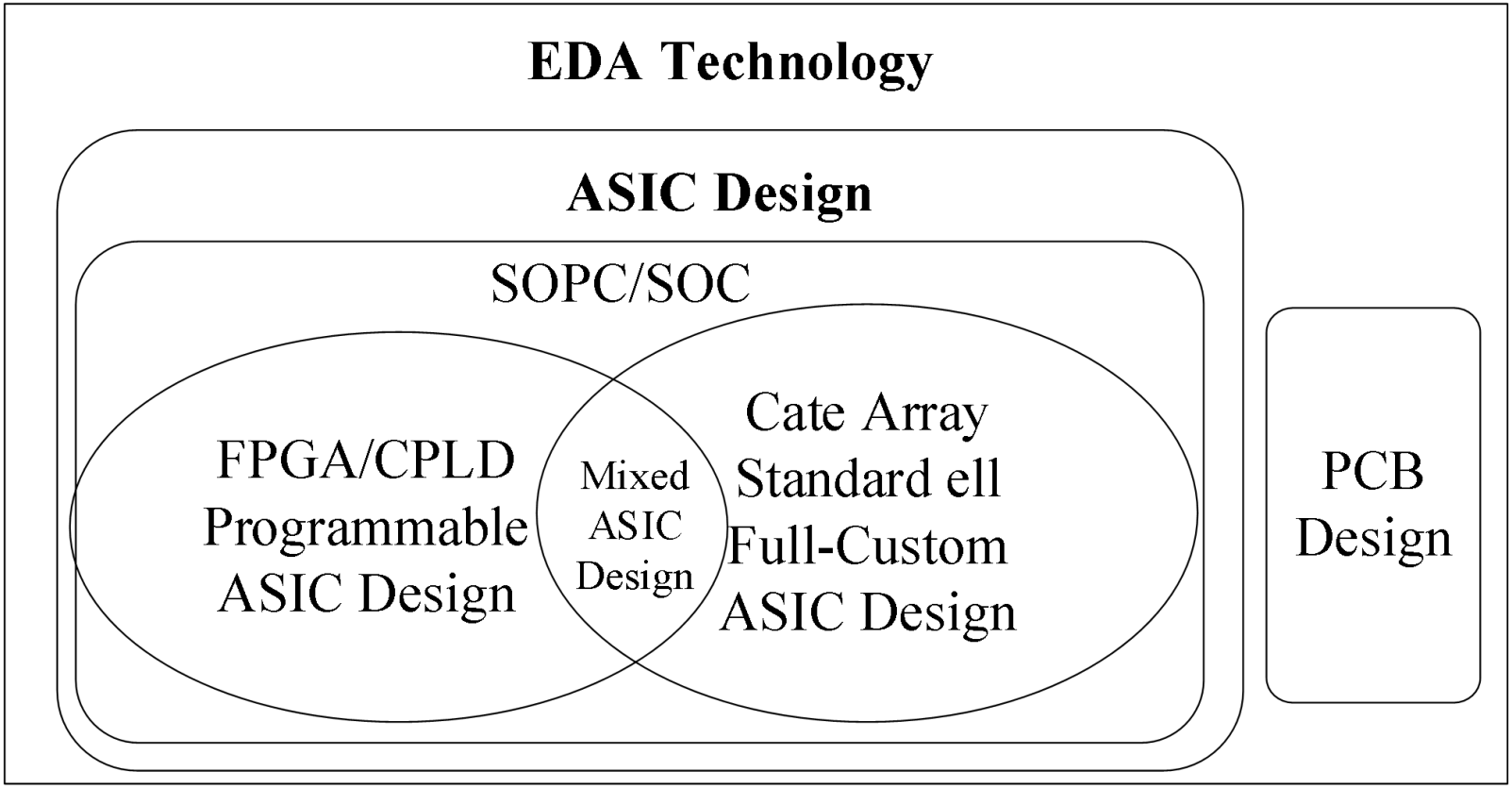
EDA？

- IC
- FPGA/CPLD
- PCB

EDA development histroy



EDA Technology



ASIC:

Application Specific Integrated Circuit

（专用集成电路）

FPGA - Field Programmable Gate Array

（现场可编程门阵列）

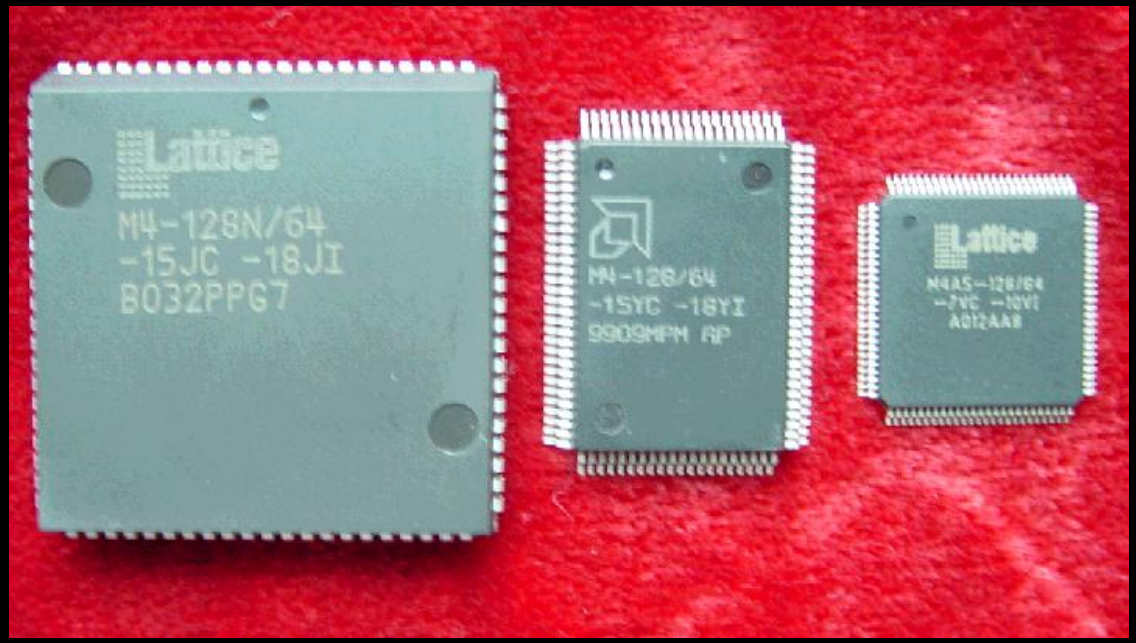
CPLD - Complex Programmable Logic Device

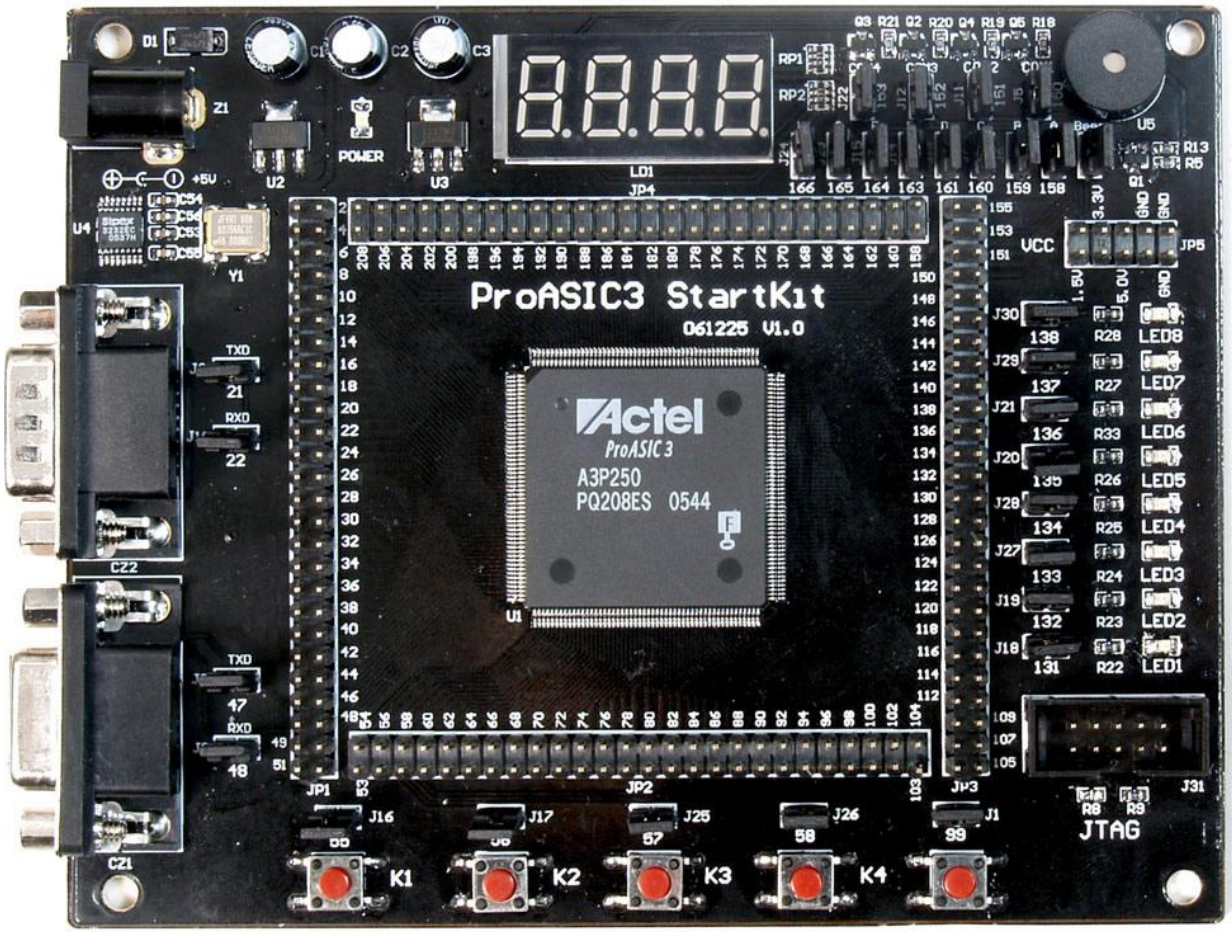
（复杂可编程逻辑器件）

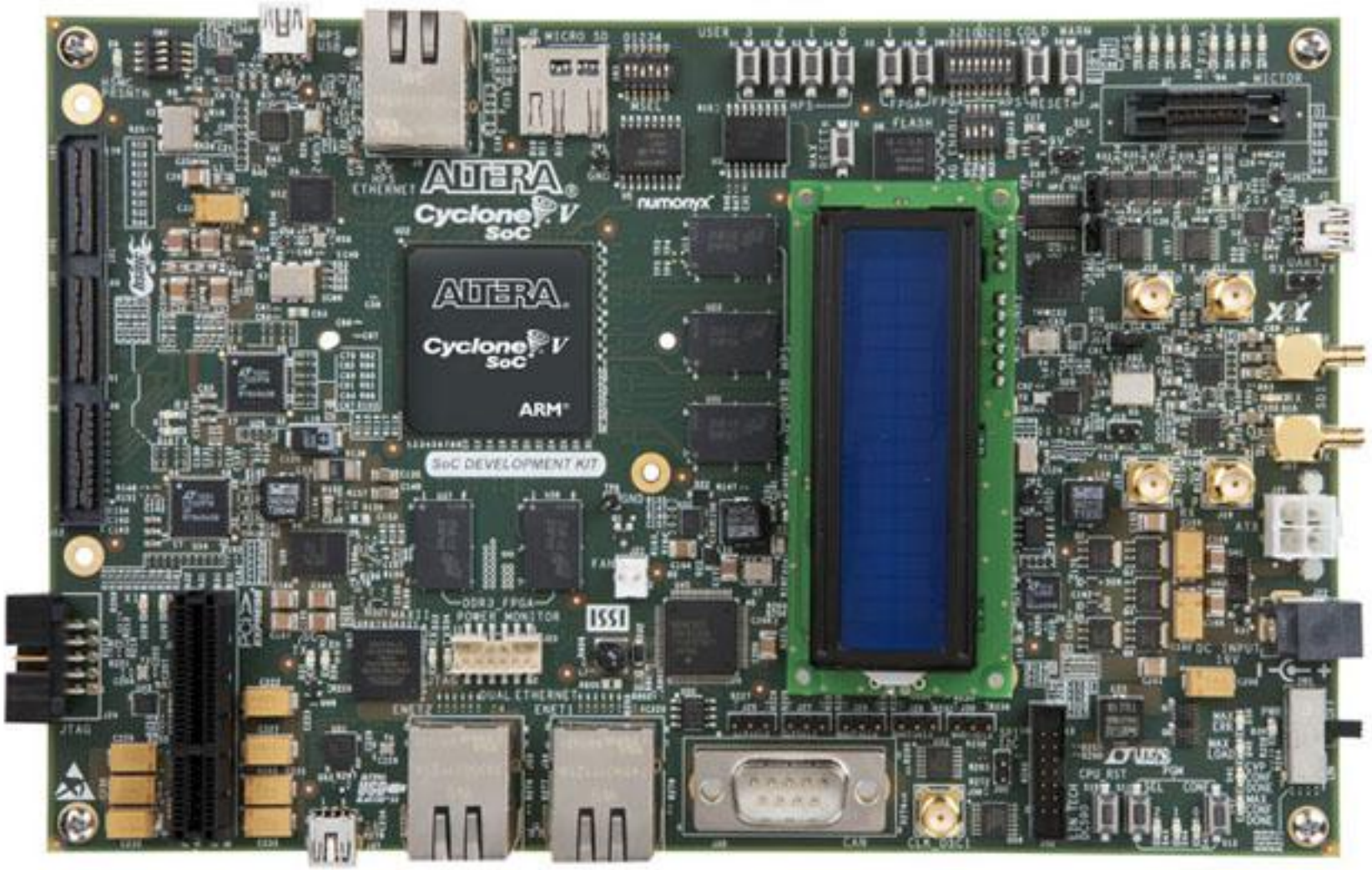
Welcome to the world of programmable logic!

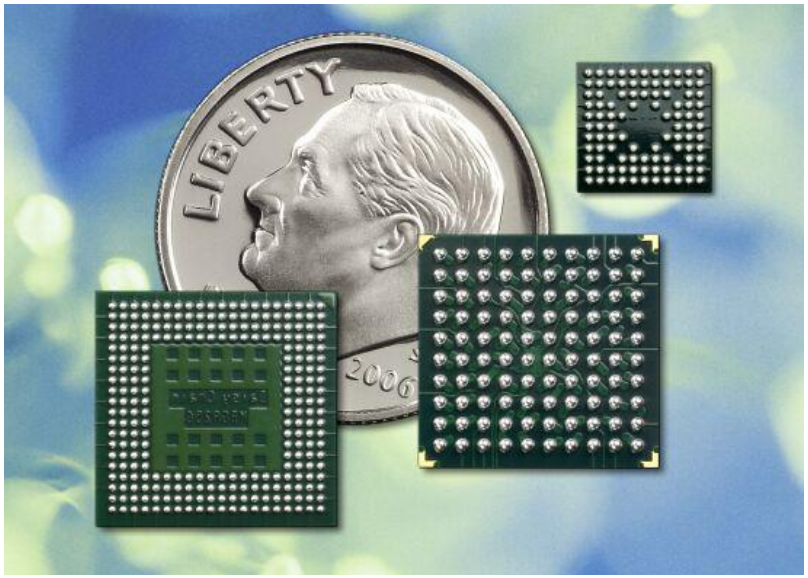
PLD


XILINX®
XC95108™
PQ160AEM9913
F1079760A

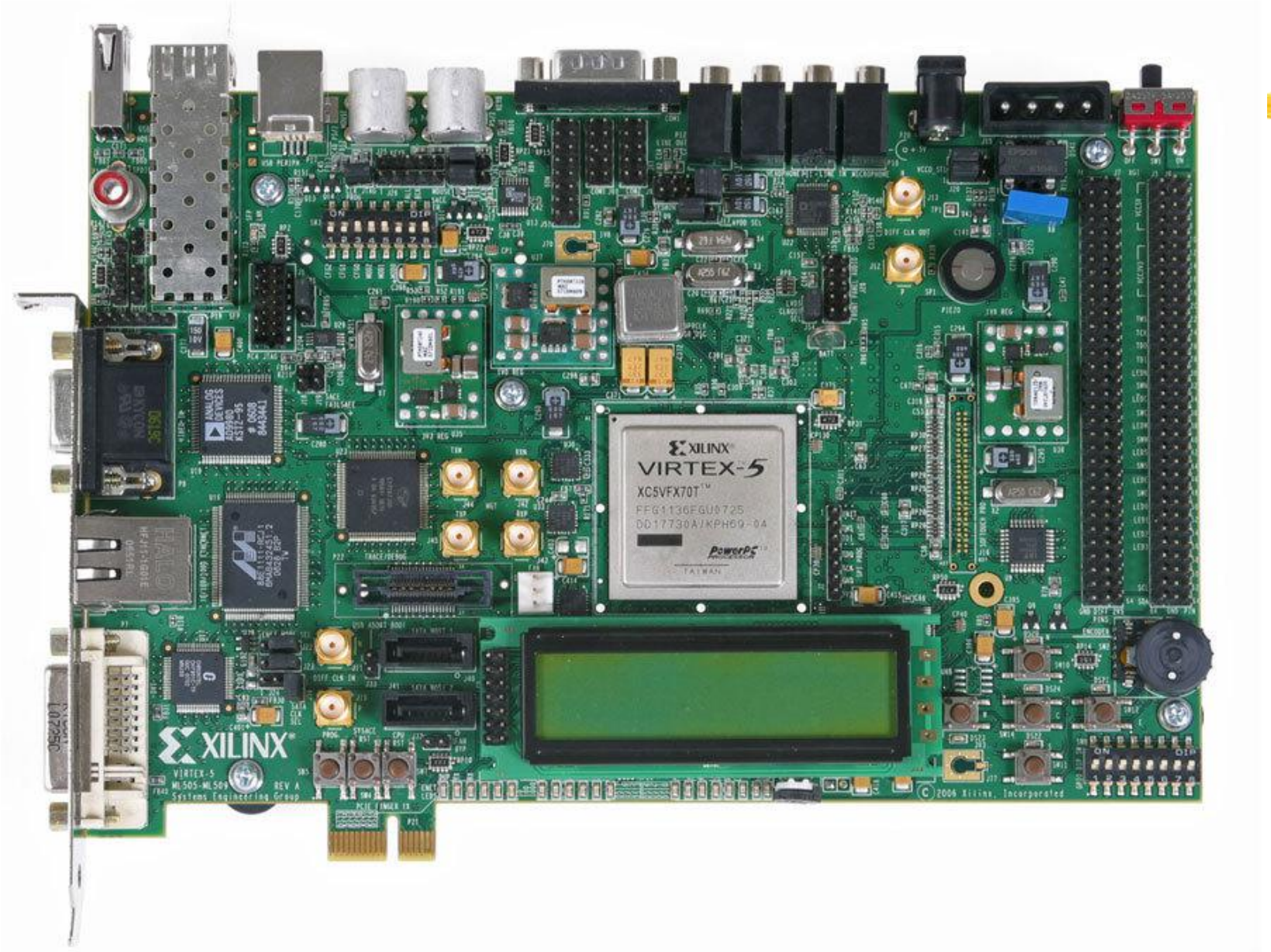

Lattice
M4-128N/64
-15JC -18JI
B032PPG7
M4-128/64
-15YC -18YI
9909MPM AP
Lattice

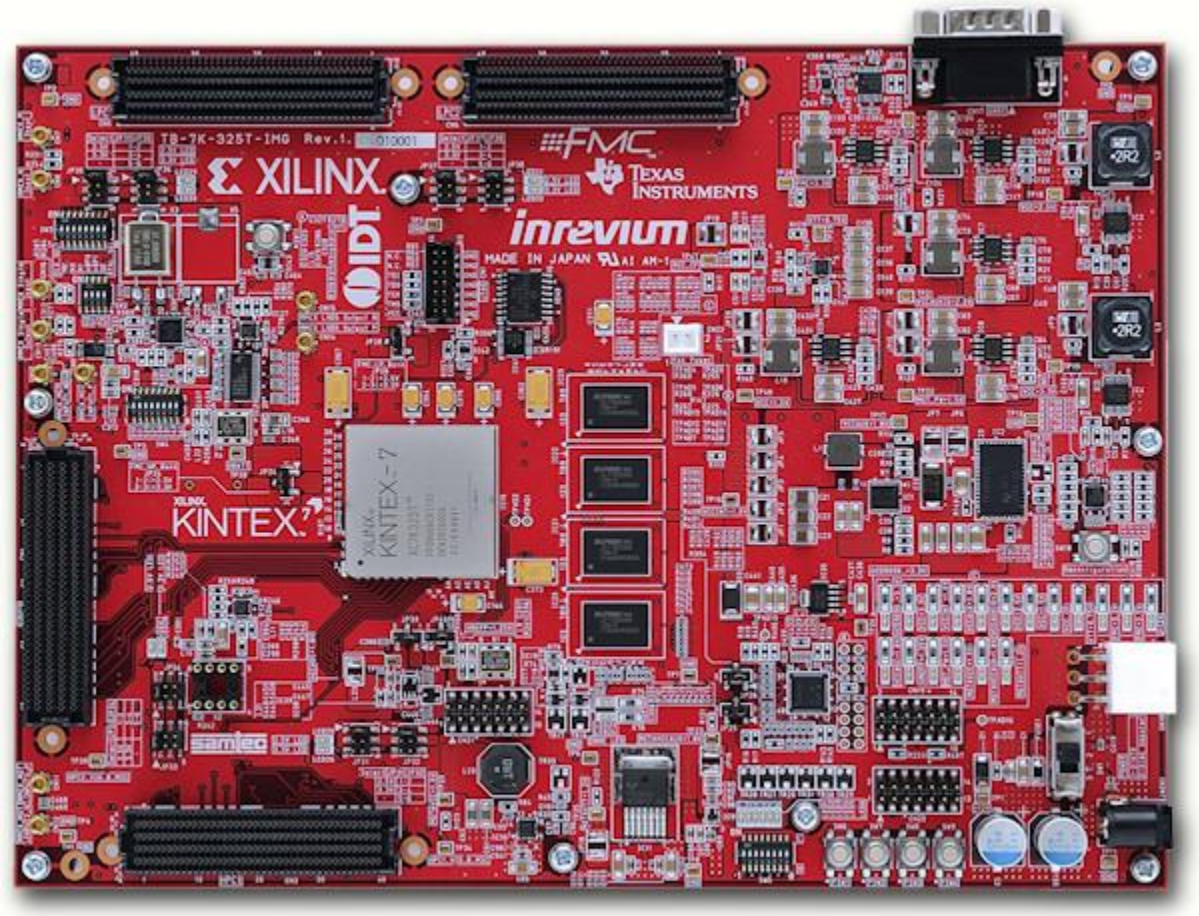

ProASIC3 StartKit
061225 V1.0
Actel
ProASIC3
A3P250
PQ208ES 0544
U1
JP1
JP2
JP3
JP4
JP5
J31
JTAG
K1
K2
K3
K4
LD1
POWER
VCC
LED8
LED7
LED6
LED5
LED4
LED3
LED2
LED1
TXD
RXD


ALTERA
Cyclone V
SoC
ALTERA
Cyclone V
SoC
ARM
SoC DEVELOPMENT KIT
numonyx
ISSI
DDR3_FPGA
POWER MONITOR
DUAL ETHERNET
JTAG
CAN
FLASH
MICRO SD

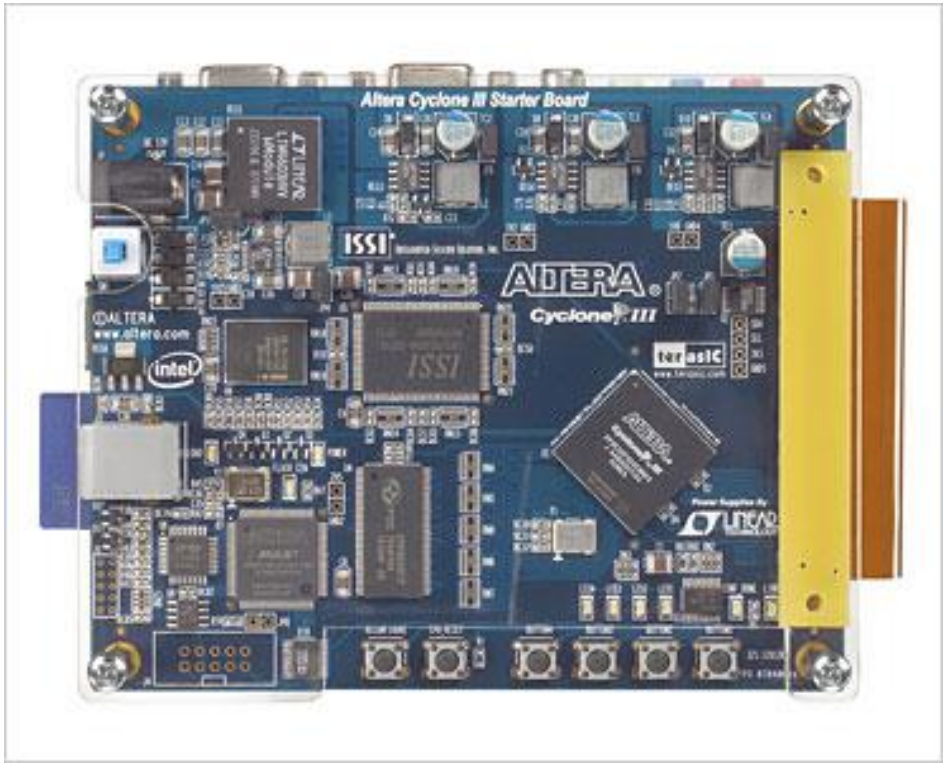

LIBERTY
2006

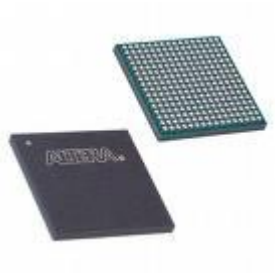

ALTERA
MAX 10
FPGAs
axelsys
www.axelsys.com
Enpirion


XILINX
VIRTEX-5
XC5VFX70T
FFG1136EGU0725
XILINX
VIRTEX-5
ML505-ML509
REV A
Systems Engineering Group
© 2006 Xilinx, Incorporated


TB-7K-325T-IMG Rev.1
FMC
XILINX
TEXAS INSTRUMENTS
IDT
inrevium
MADE IN JAPAN
XILINX KINTEX-7
KINTEX-7


ALTERA
Stratix V


Altera Cyclone III Starter Board
ALTERA
Cyclone III
ISSI
intel




Stratix V
ALTERA
GSI
LINEAR

Language

- HDL：

  Hardware Description Language

- VerilogHDL
- VHDL
- SystemC
- SystemVerilog
- Chisel

Verilog HDL

- It was created by Prabhu Goel, Phil Moorby and Chi-Lai Huang and Douglas Warmke between late 1983 and early 1984.
- Chi-Lai Huang had earlier worked on a hardware description LALSD, a language developed by Professor S.Y.H. Su, for his PhD work.
- The wording for this process was "Automated Integrated Design Systems" (later renamed to Gateway Design Automation in 1985) as a hardware modeling language.
- Gateway Design Automation was purchased by Cadence Design Systems in 1990.

Verilog HDL

⌘ **Verilog-95**

- With the increasing success of VHDL at the time, Cadence decided to make the language available for open standardization..

⌘ **Verilog 2001**

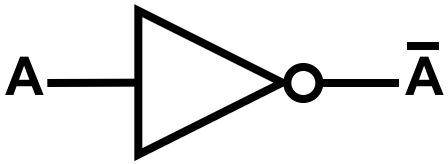
-In 2001, IEEE released the second standard version of Verilog HDL (Verilog 2.0), i.e. IEEE Std 1364-2001, also known as Verilog-2001.

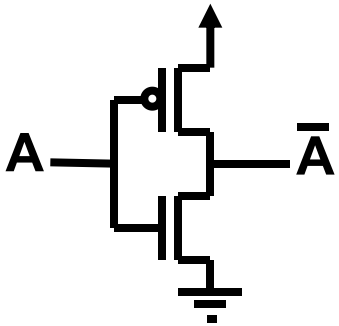
Verilog 2005

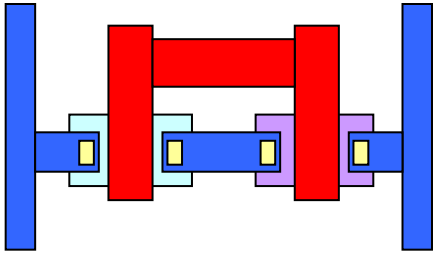
⌘ **SystemVerilog**

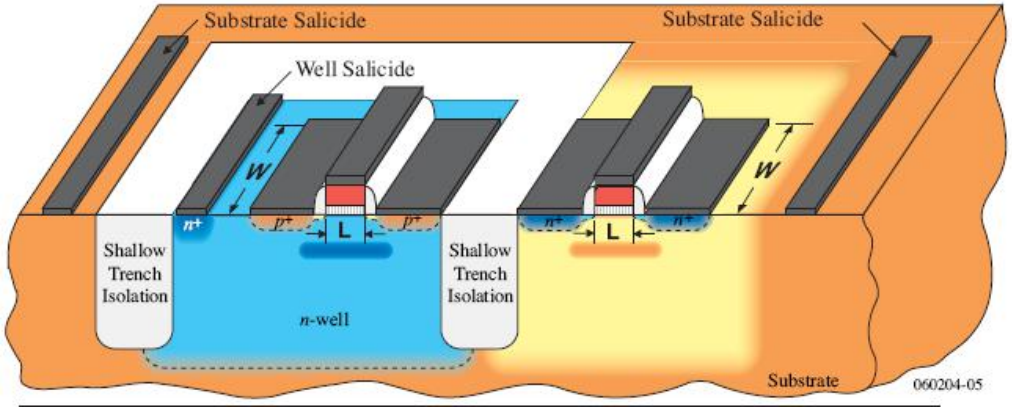
Circuit description level

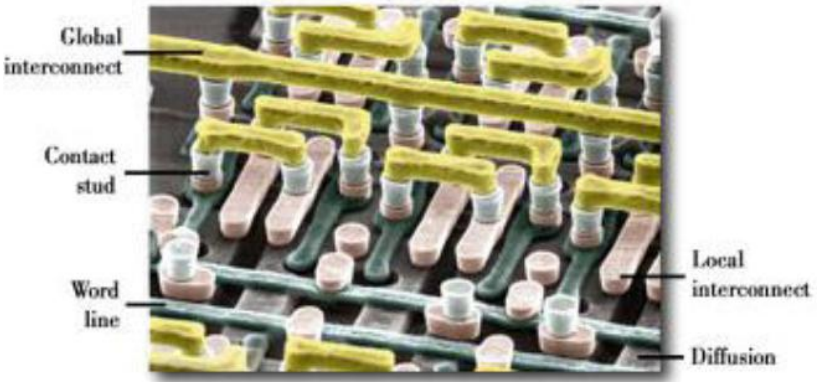
- Electronic system level (ESL)/System level
- Behavioral level
- Register transfer level （RTL）
- Gate Level
- Transistor level/ MOS Level
- Physical level

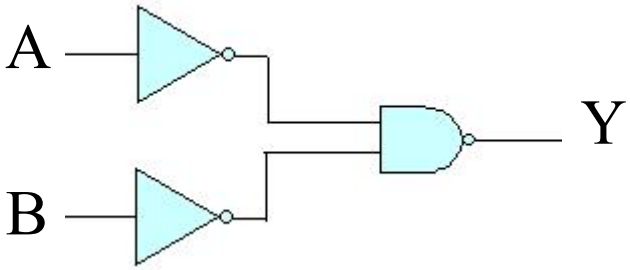

A
$\bar{A}$

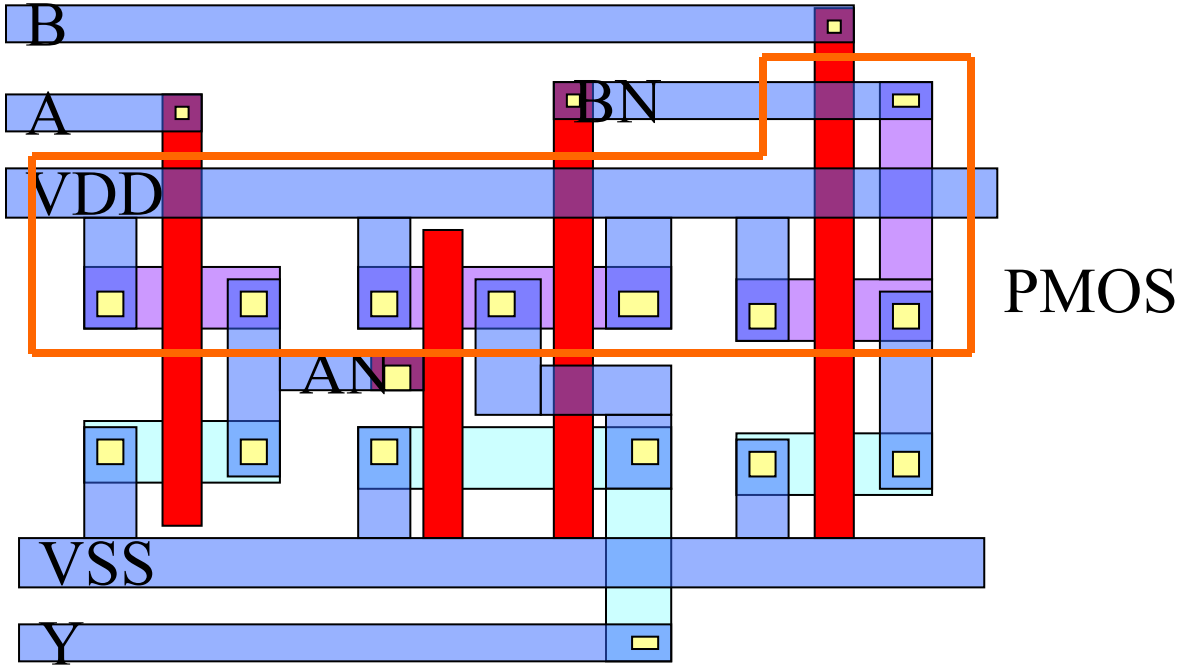

A
$\bar{A}$




Substrate Salicide
Substrate Salicide
Well Salicide
W
W
n+
p+
L
p+
n+
L
n+
Shallow Trench Isolation
Shallow Trench Isolation
n-well
Substrate
060204-05


Global interconnect
Contact stud
Word line
Local interconnect
Diffusion


A
B
Y


B
A
BN
VDD
PMOS
AN
VSS
Y

HDL Synthesis

HDL Synthesis：
Transforming HDL description into gate level circuit description

HDL Synthesis

Every step in the design process can be called a comprehensive link.

(1) Conversion from natural language to Verilog algorithm is the natural language synthesis;

(2) Conversion from the algorithm expression to the register transfer level (Register Transport Level, RTL) expression, which is the synthesis from the behavior domain to the structural domain, is the behavior synthesis;

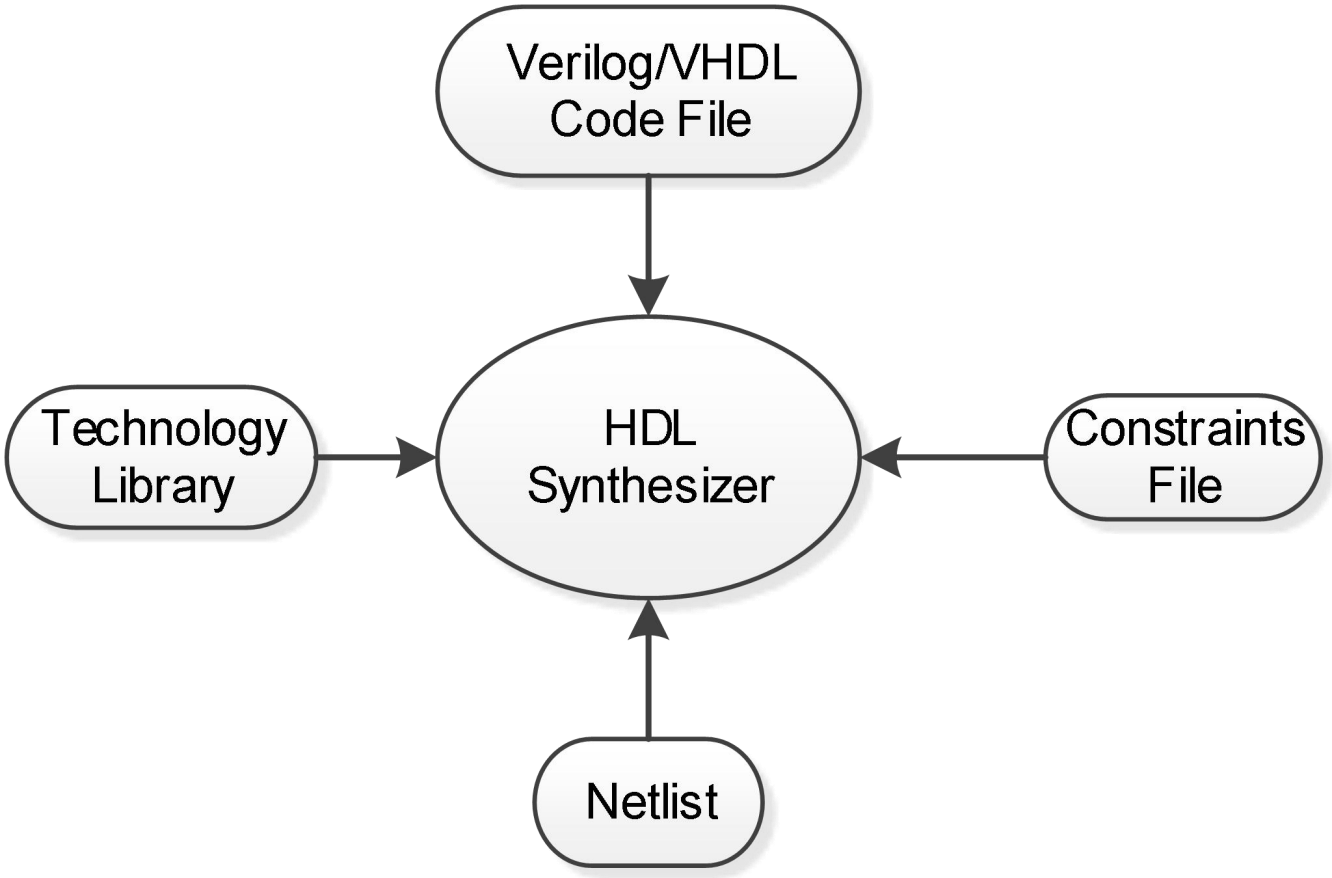
(3) Conversion from RTL level expression to logic gate (including trigger) expression, is the logic synthesis;

(4) Conversion from logic gates to layout level (such as ASIC design) or conversion to FPGA's configuration netlist file, is the layout synthesis or structural synthesis. With the layout information, chips can be manufactured. With the corresponding configuration files, the corresponding FPGA can be turned into a circuit device with special functions.

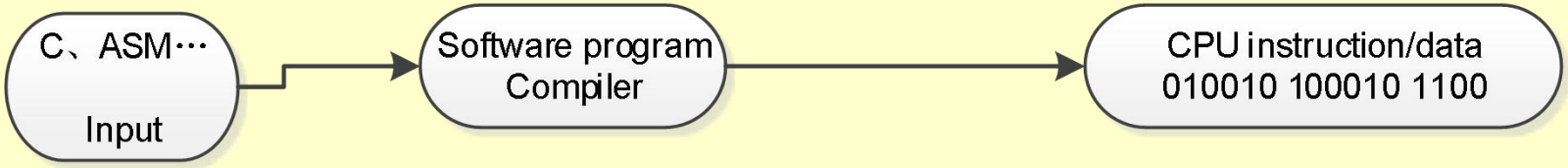
HDL Synthesis

Conversion from natural language to Verilog algorithm	→	Natural language synthesis
Conversion from the algorithm expression to the register transfer level (Register Transport Level, RTL) expression	→	Behavior synthesis
Conversion from RTL level expression to logic gate (including trigger) expression	→	Logic synthesis
Conversion from logic gates to layout level (such as ASIC design) or conversion to FPGA's configuration netlist file	→	Layout synthesis or structural synthesis

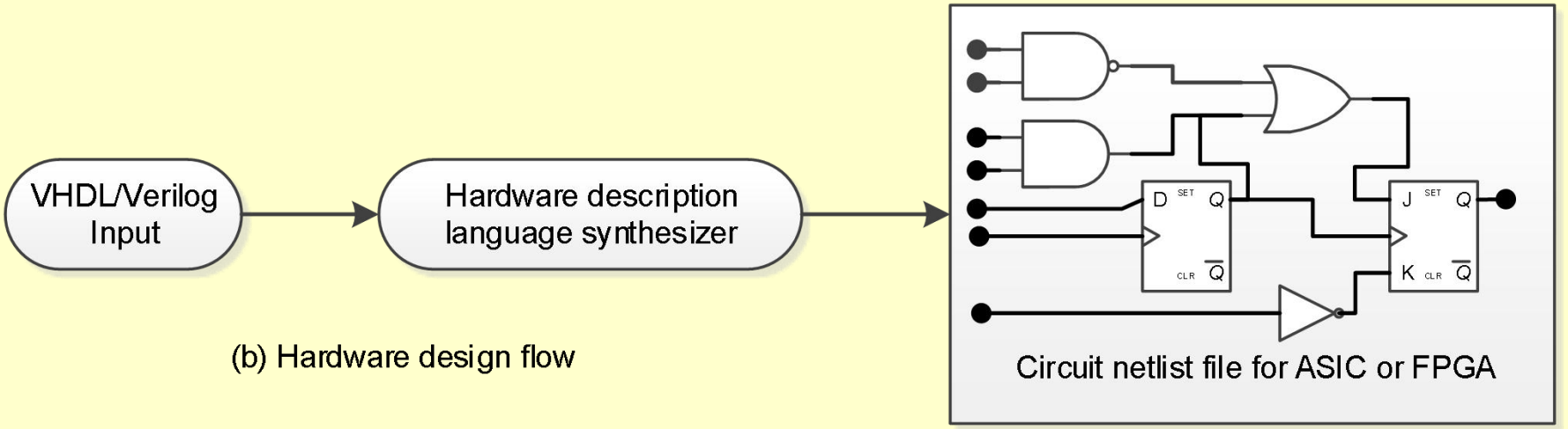
Running process for HDL synthesizer



Compiler vs. Synthesizer

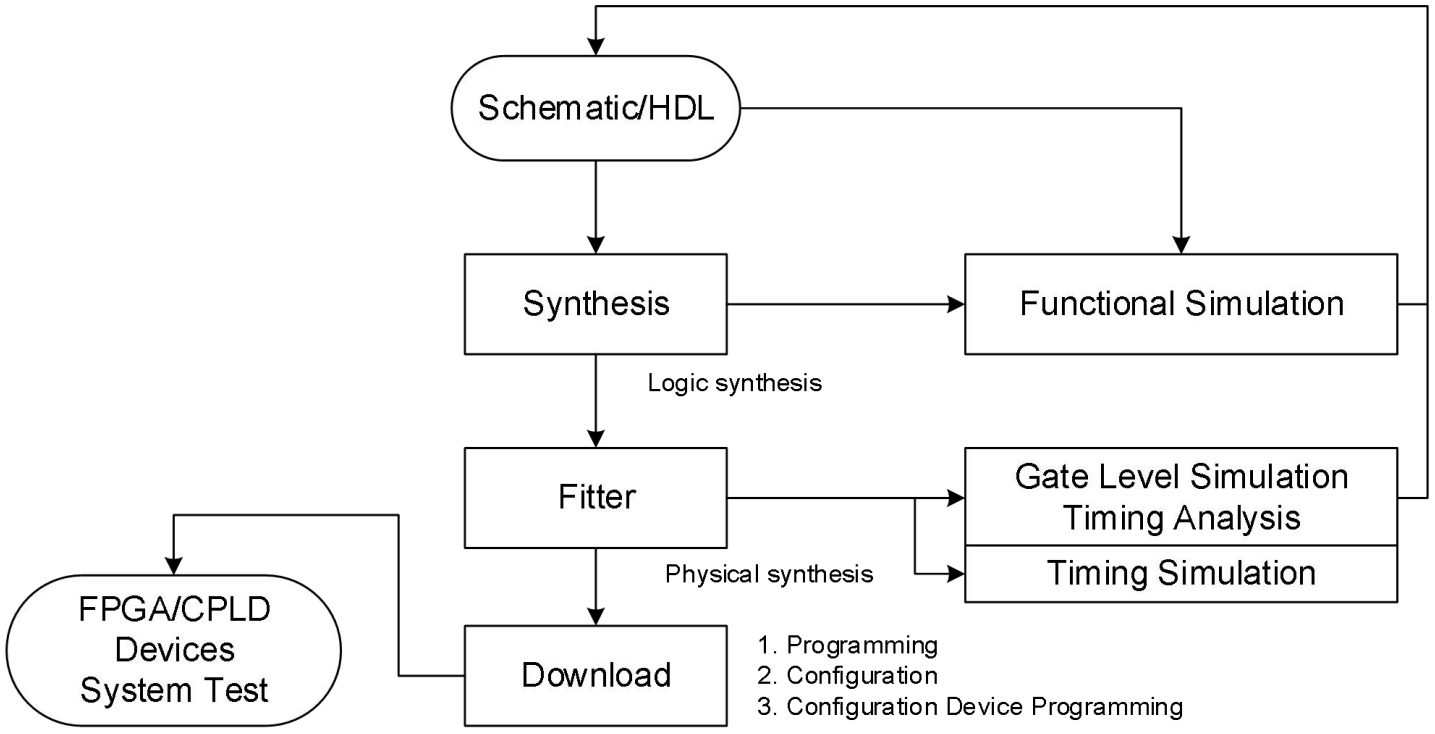


(a) Software design flow



(b) Hardware design flow

FPGA / CPLD Design Flow

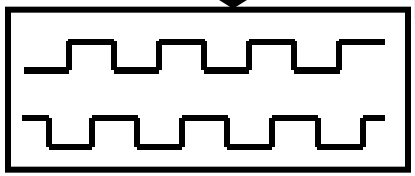


PLD Design Flow

Design Specification

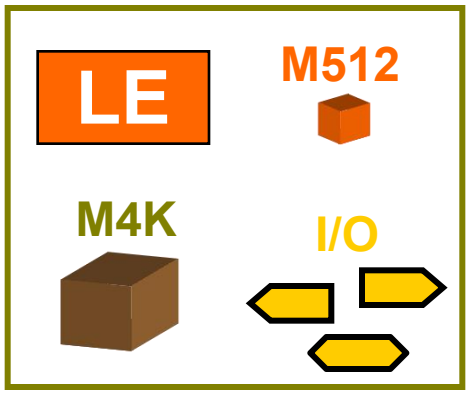
Design Entry/RTL Coding

- Behavioral or Structural Description of Design



RTL Simulation

- Functional Simulation
- Verify Logic Model & Data Flow (No Timing Delays)



Synthesis

- Translate Design into Device Specific Primitives
- Optimization to Meet Required Area & Performance Constraints

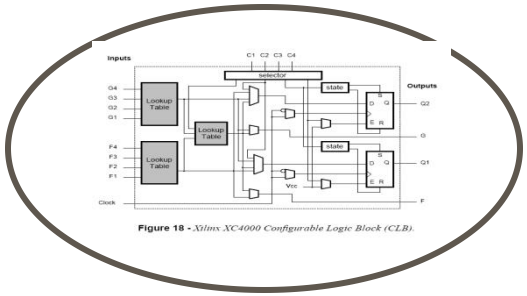
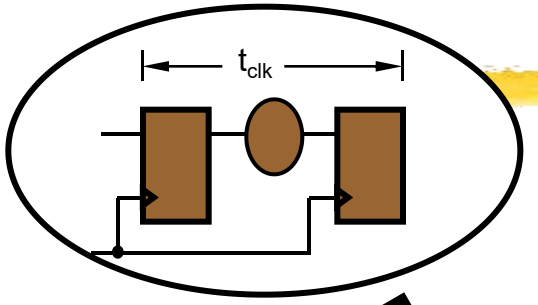


Figure 18 - *Xilinx XC4000 Configurable Logic Block (CLB).*

Place & Route

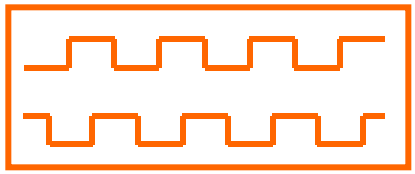
- Map Primitives to Specific Locations inside Target Technology with Reference to Area & Performance Constraints
- Specify Routing Resources to Be Used

PLD Design Flow



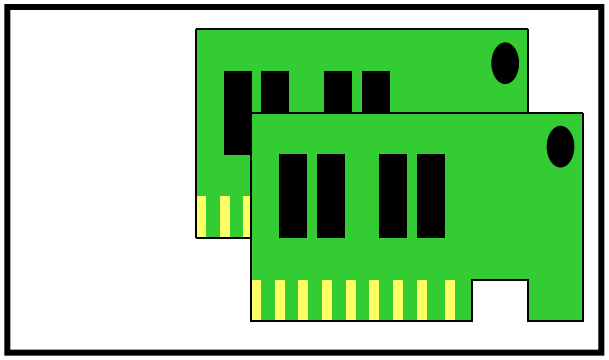
Timing Analysis

- Verify Performance Specifications Were Met
- Static Timing Analysis



Gate Level Simulation

- Timing Simulation
- Verify Design Will Work in Target Technology



PC Board Simulation & Test

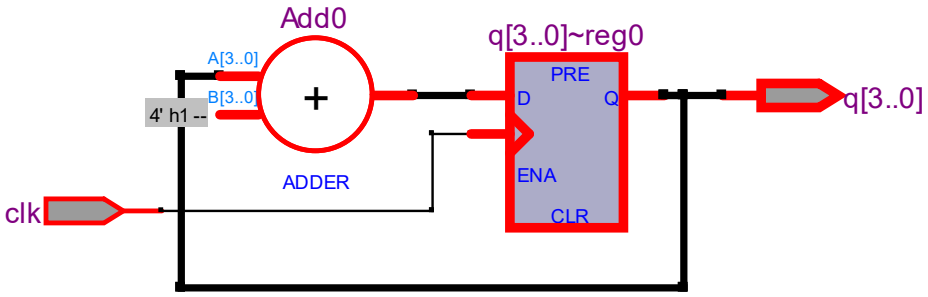
- Simulate Board Design
- Program & Test Device on Board

Counter

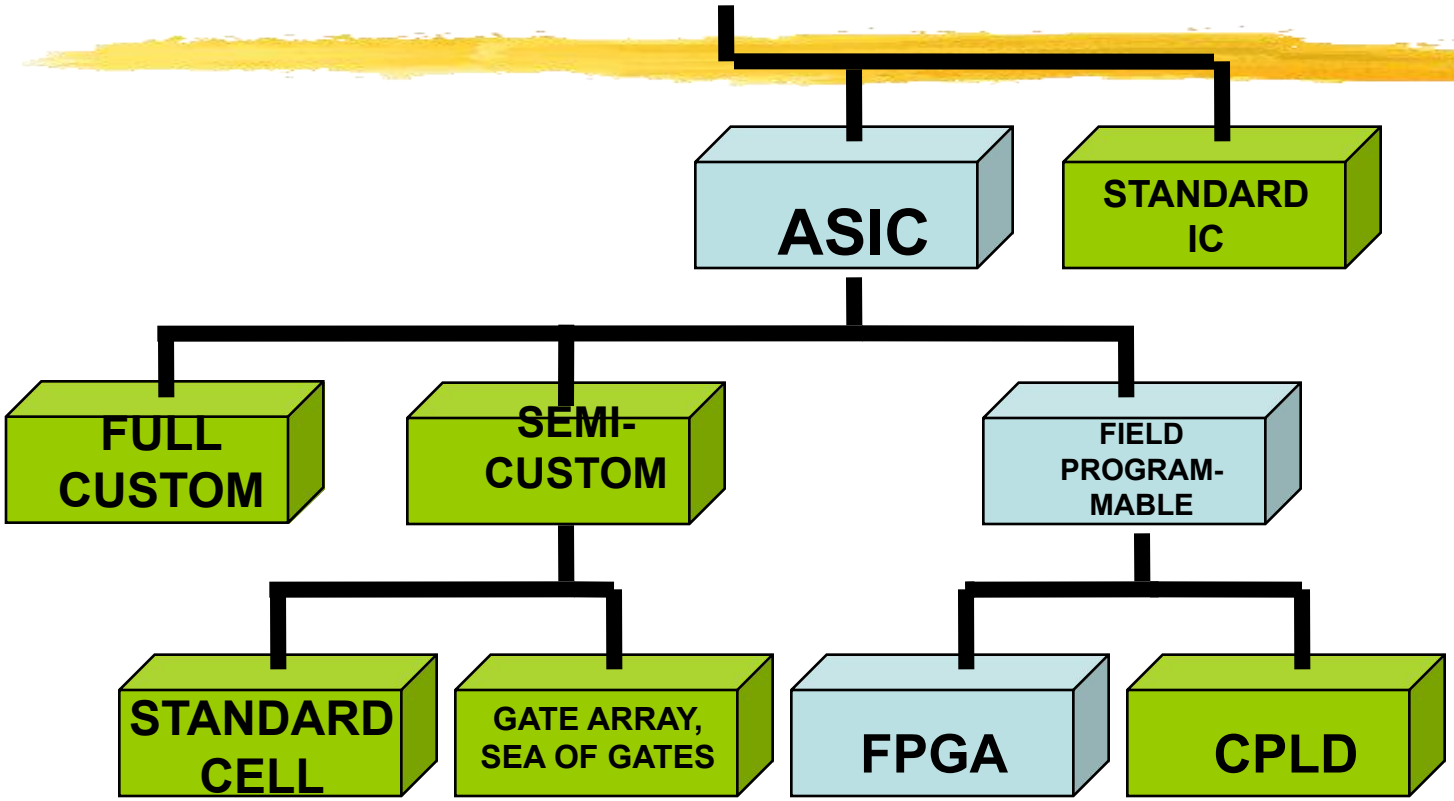
```
module cnt4(clk,q);
input clk;
output [3:0] q;
reg q;
always @(posedge clk)  begin
   q = q + 1;
end
endmodule
```

Counter

- module cnt4(input clk,output reg[3:0] q);
- always @(posedge clk) q = q + 1;
- endmodule



CMOS IC Design Alternatives



•Field Programmable Gate Array (FPGA) – a hardware device with programmable logic, routing, memory, and I/O

EDAr

Synopsys
Cadence
MENTOR GRAPHICS(Siemens)

FPGA/CPLD Tools

1、Intel： Quartus Prime

2、XILINX： Vivado、ISE

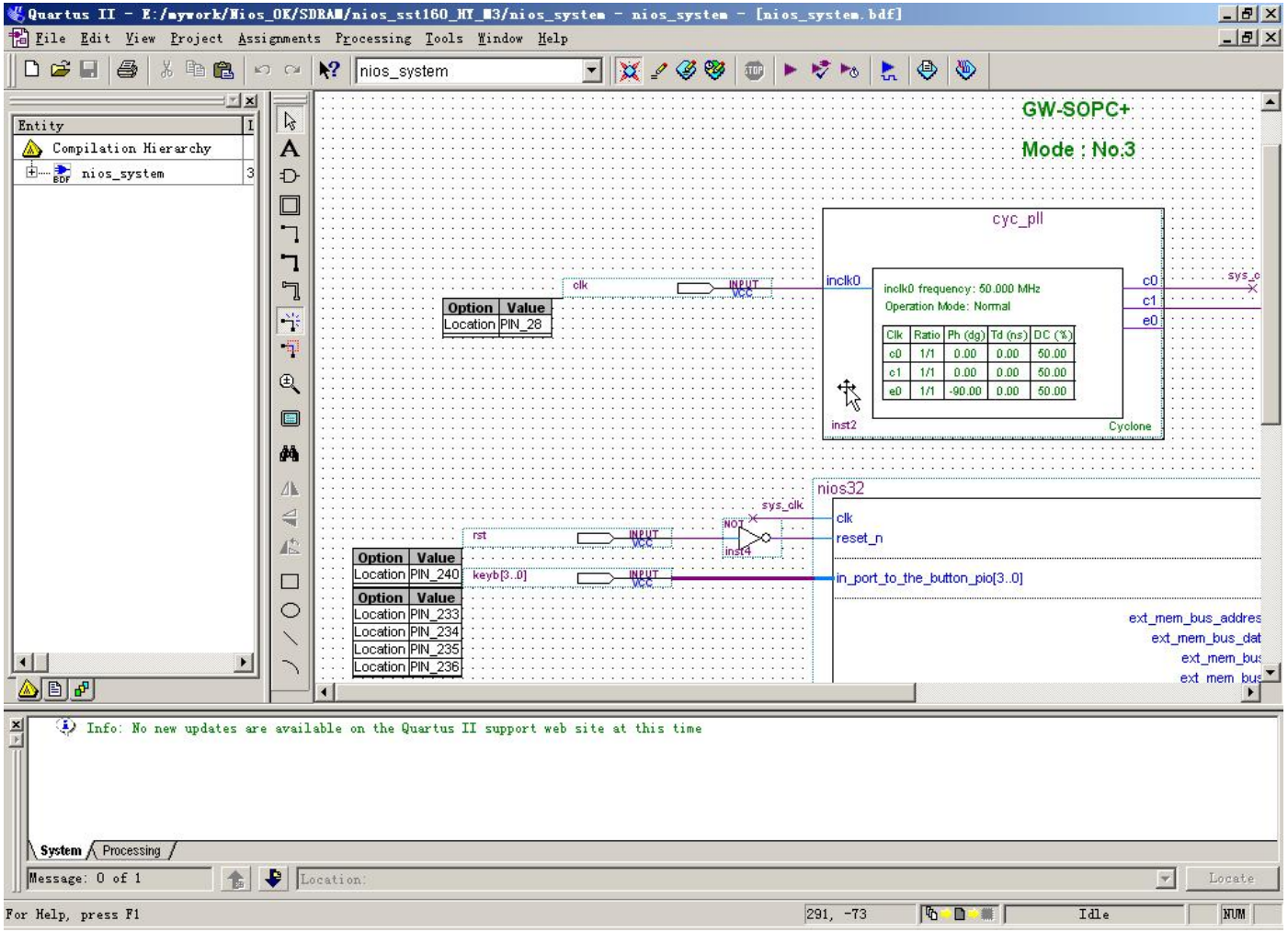
3、MicroChip： Libero

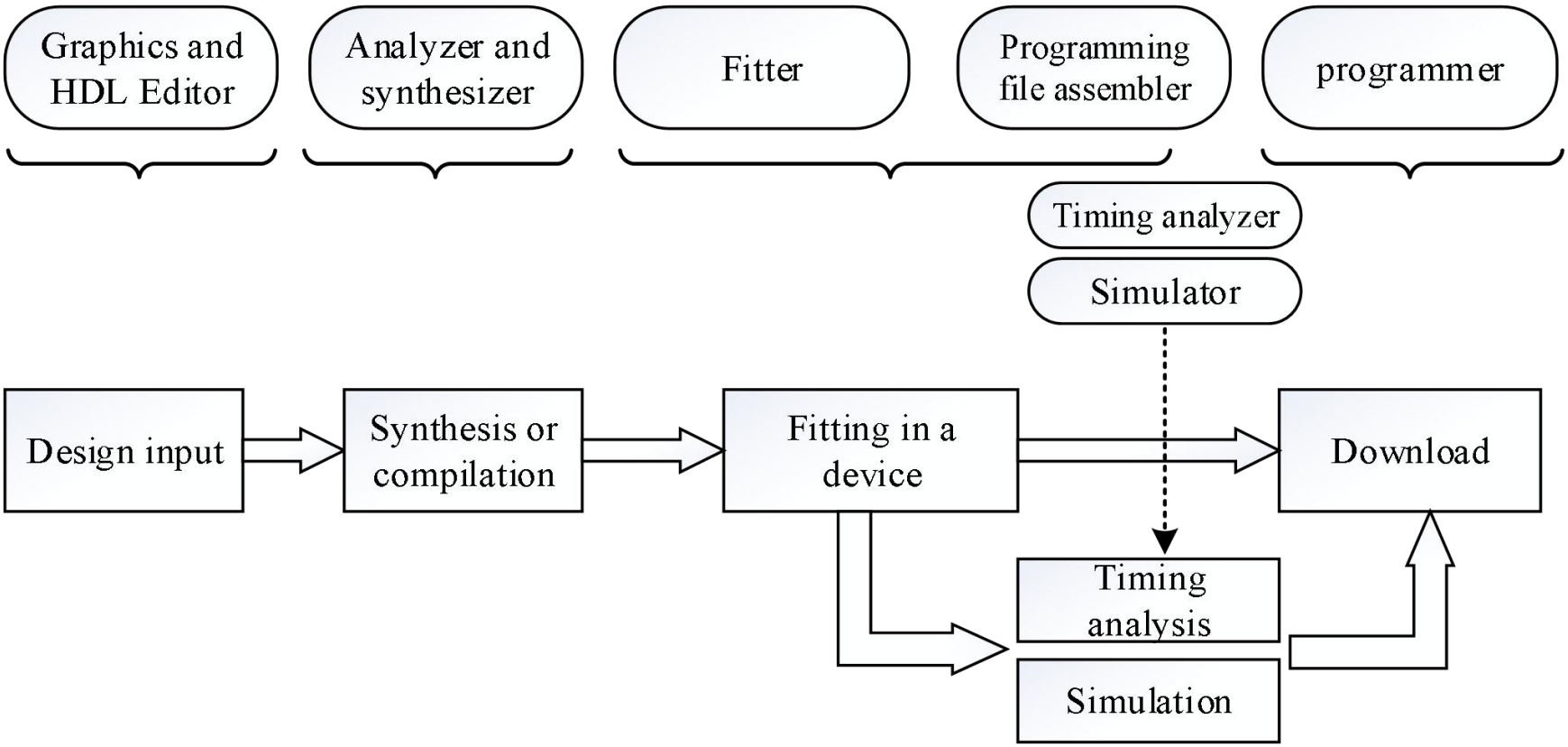
4、Lattice Semi： IispLever

Quartus II


QUARTUS® II


#1 in Performance & Productivity
QUARTUS® II
QUARTUS® II
Version 5.1
ALTERA®


Quartus II - E:/mywork/Nios_OK/SDRAM/nios_sst160_HT_M3/nios_system - nios_system - [nios_system.bdf]
File Edit View Project Assignments Processing Tools Window Help
nios_system
Entity
Compilation Hierarchy
nios_system
GW-SOPC+
Mode : No.3
cyc_pll
inclk0
inclk0 frequency: 50.000 MHz
Operation Mode: Normal
c0
c1
e0
Clk Ratio Ph (dg) Td (ns) DC (%)
c0 1/1 0.00 0.00 50.00
c1 1/1 0.00 0.00 50.00
e0 1/1 -90.00 0.00 50.00
inst2
Cyclone
clk
INPUT
Option Value
Location PIN_28
nios32
sys_clk
clk
reset_n
rst
NOT
inst4
keyb[3..0]
in_port_to_the_button_pio[3..0]
Option Value
Location PIN_240
Option Value
Location PIN_233
Location PIN_234
Location PIN_235
Location PIN_236
ext_mem_bus_addres
ext_mem_bus_dat
ext_mem_bus
ext_mem_bus
Info: No new updates are available on the Quartus II support web site at this time
System
Processing
Message: 0 of 1
Location:
Locate
For Help, press F1
291, -73
Idle
NUM

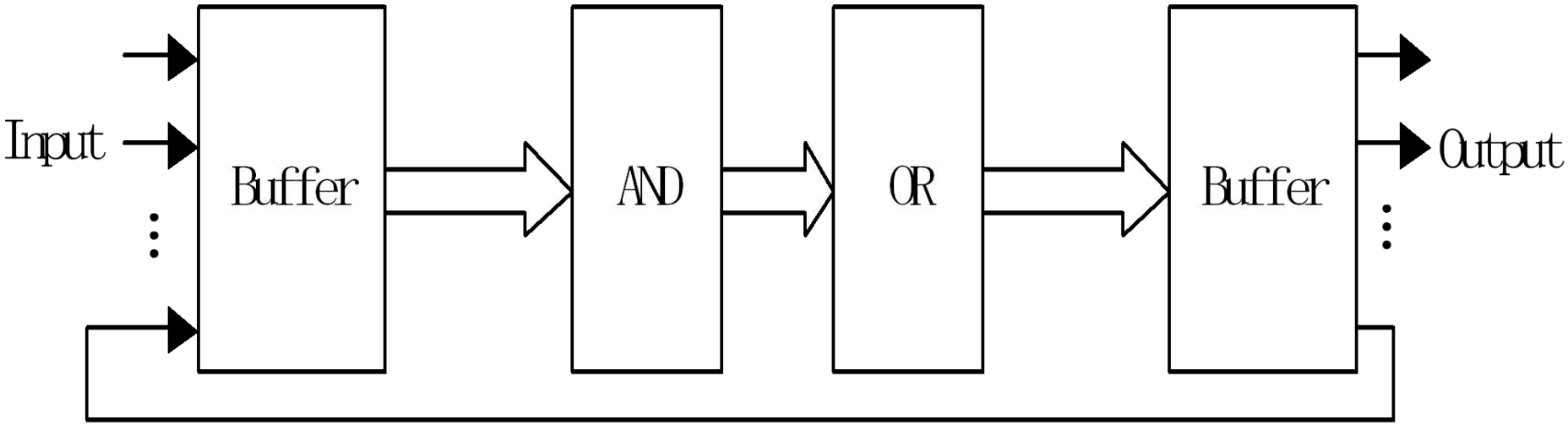

Graphics and HDL Editor
Analyzer and synthesizer
Fitter
Programming file assembler
programmer
Timing analyzer
Simulator
Design input
Synthesis or compilation
Fitting in a device
Download
Timing analysis
Simulation

FPGA - Field Programmable Gate Array

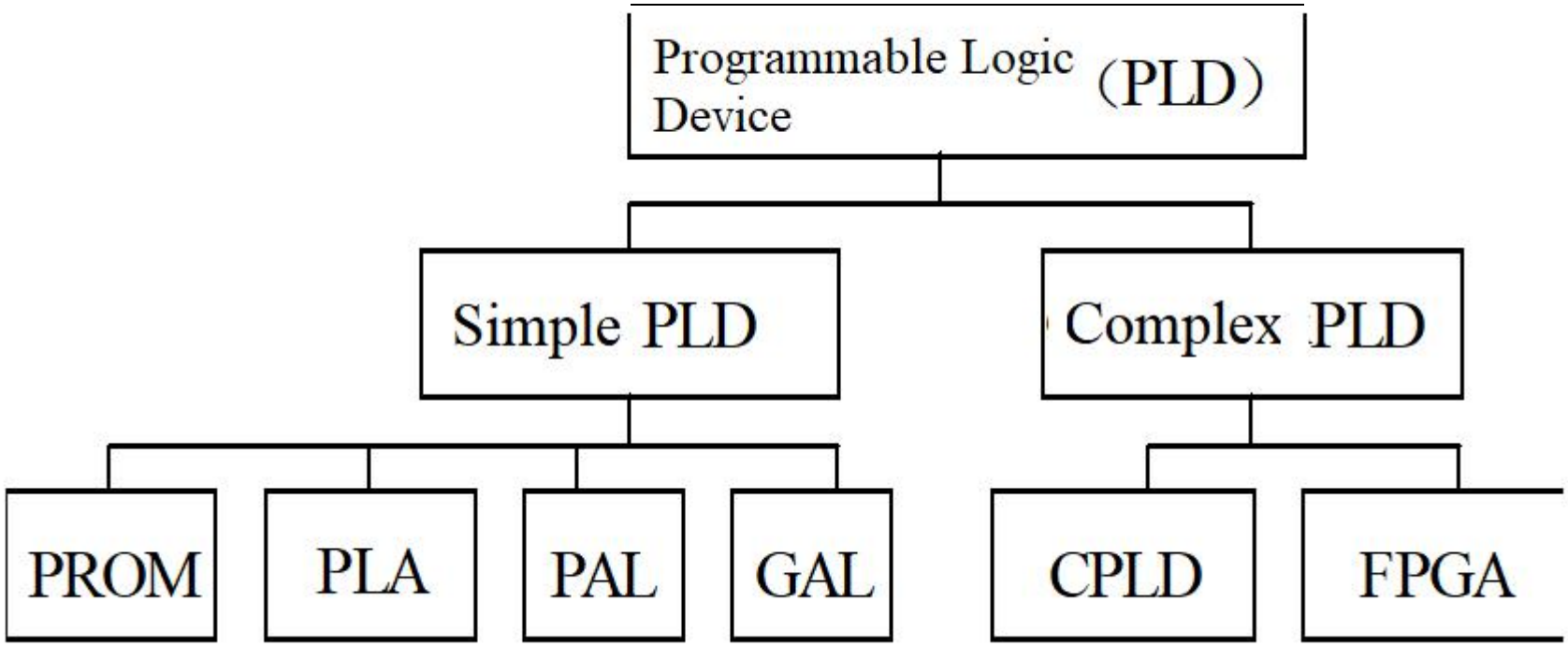
CPLD - Complex Programmable Logic Device

Introduction

Fig. PLD diagram

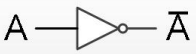
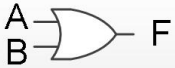
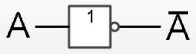
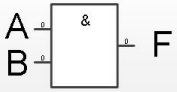
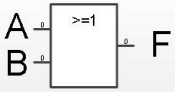
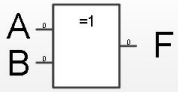


PLD


Programmable Logic Device (PLD)
Simple PLD
Complex PLD
PROM
PLA
PAL
GAL
CPLD
FPGA

PLD

Fig. symbol

	NOT GATE	AND GATE	OR GATE	XOR GATE
IEEE 1991 Standard logic symbols				
IEEE 1984 Standard logic symbols				
Logical expression	$\overline{A}$ =NOT A	F=A · B	F=A+B	F=A ⊕ B

电路符号表示

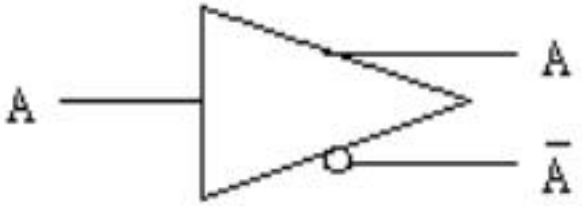


Fig. PLD buffer

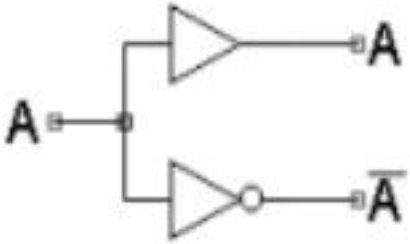


Fig. PLD input

Fig. PLD and array

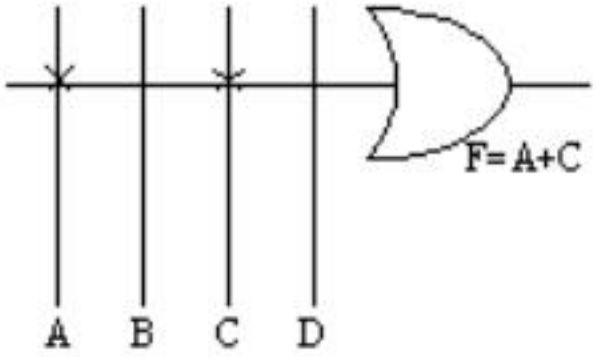
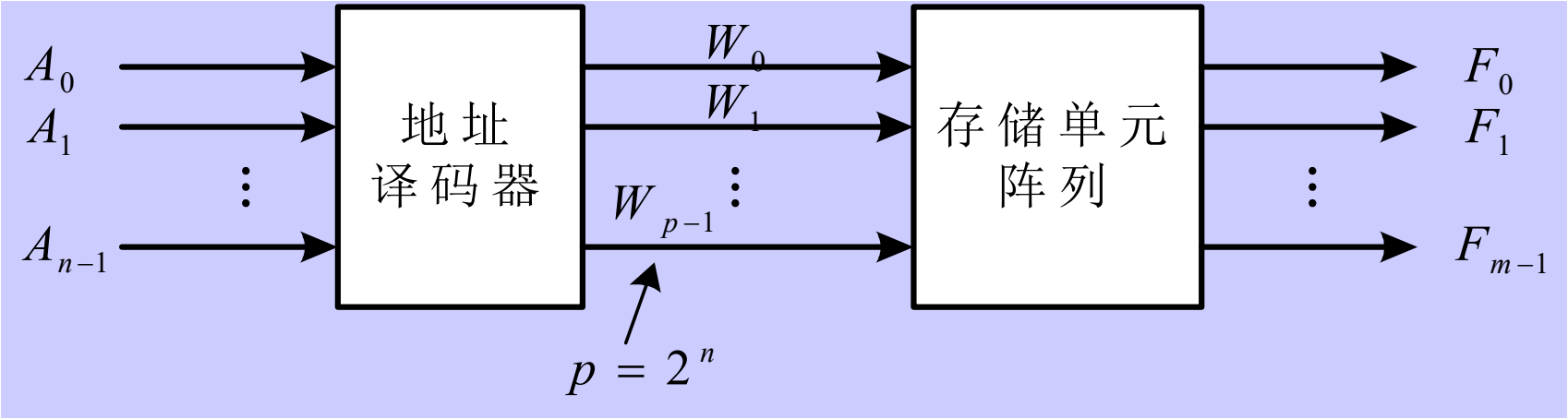


Fig. PLD OR Array

Fig. connection

PROM

Fig PROM:



function:

$$W_0 = \overline{A}_{n-1} \cdots \overline{A}_1 \overline{A}_0$$

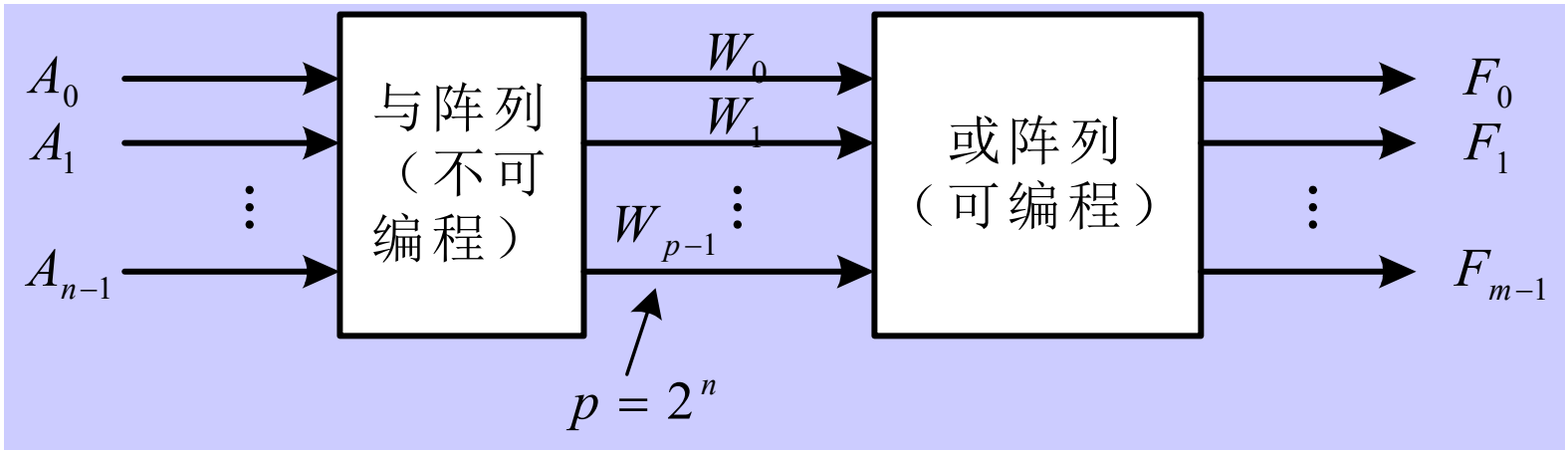
$$W_1 = \overline{A}_{n-1} \cdots \overline{A}_1 A_0$$

$$\cdots$$

$$W_{2^n-1} = A_{n-1} \cdots A_1 A_0$$

PROM

Fig. PROM



function：

$$F_0 = M_{p-1,0}W_{p-1} + \cdots + M_{1,0}W_1 + M_{0,0}W_0$$

$$F_1 = M_{p-1,1}W_{p-1} + \cdots + M_{1,1}W_1 + M_{0,1}W_0$$

$$\vdots$$

$$F_{m-1} = M_{p-1,m-1}W_{p-1} + \cdots + M_{1,m-1}W_1 + M_{0,m-1}W_0$$

PROM

Fig. PROM

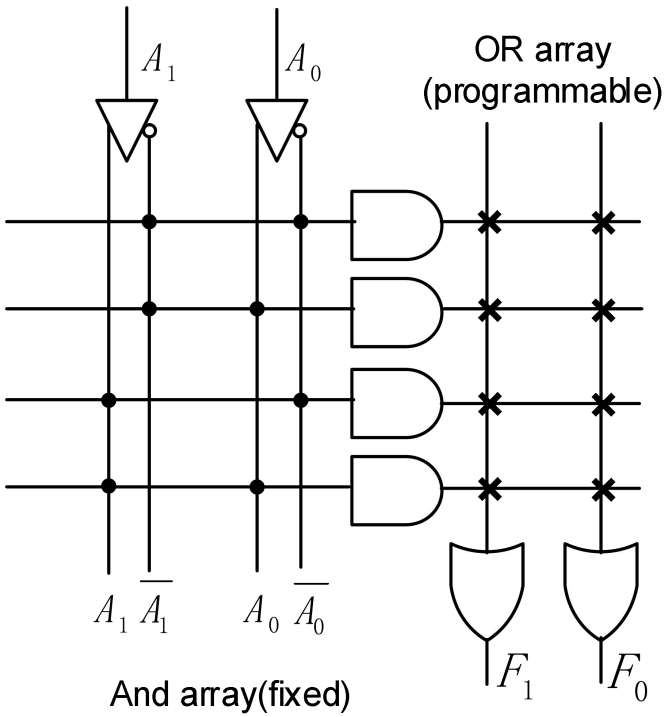
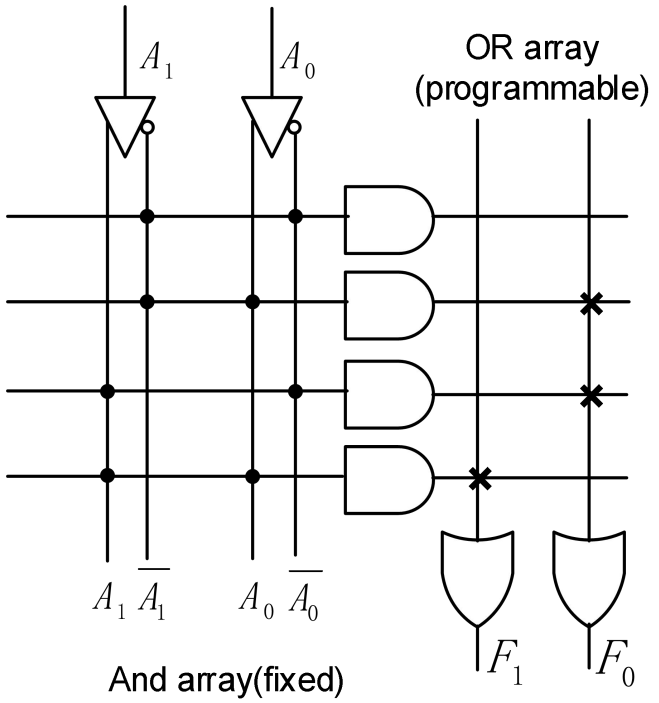


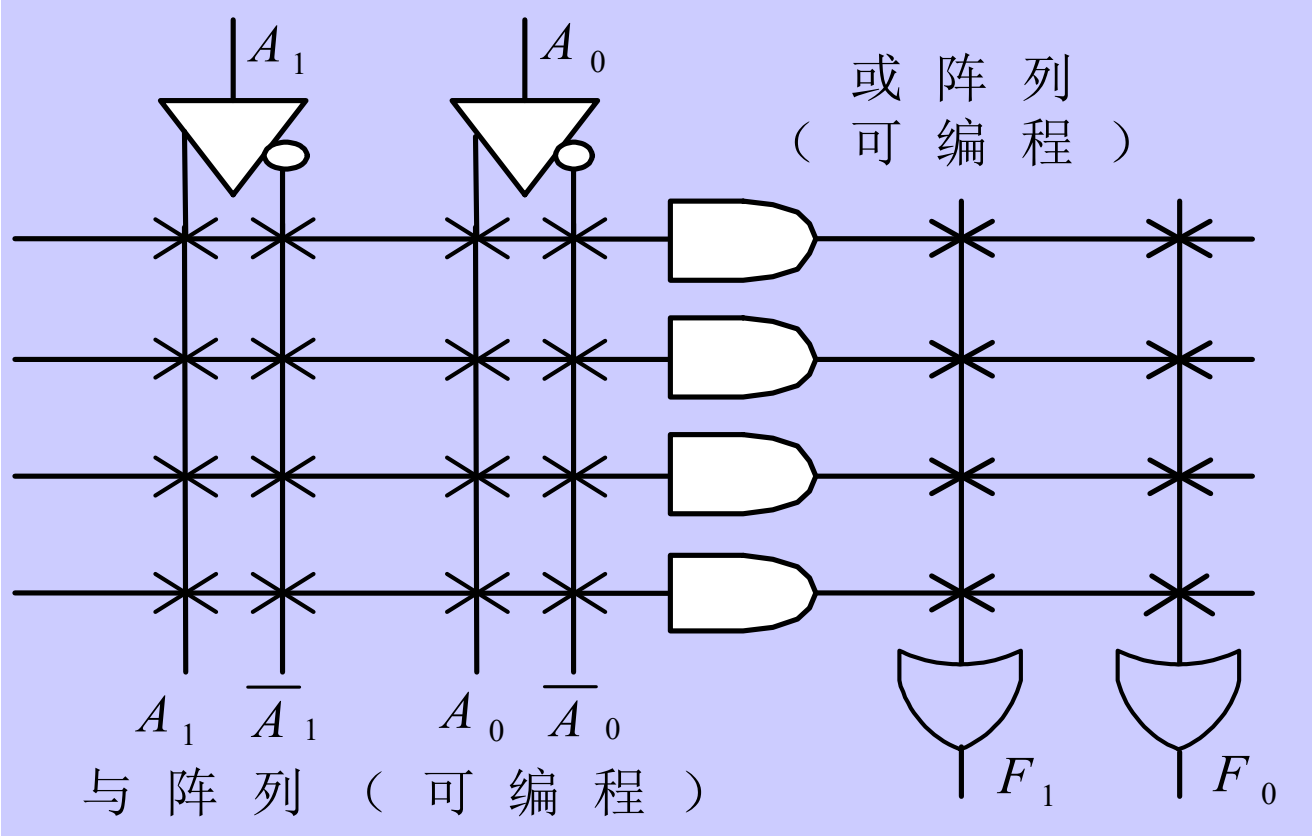
Fig. Half_adder by PROM



$$F0 = A0\overline{A1} + \overline{A0}A1$$

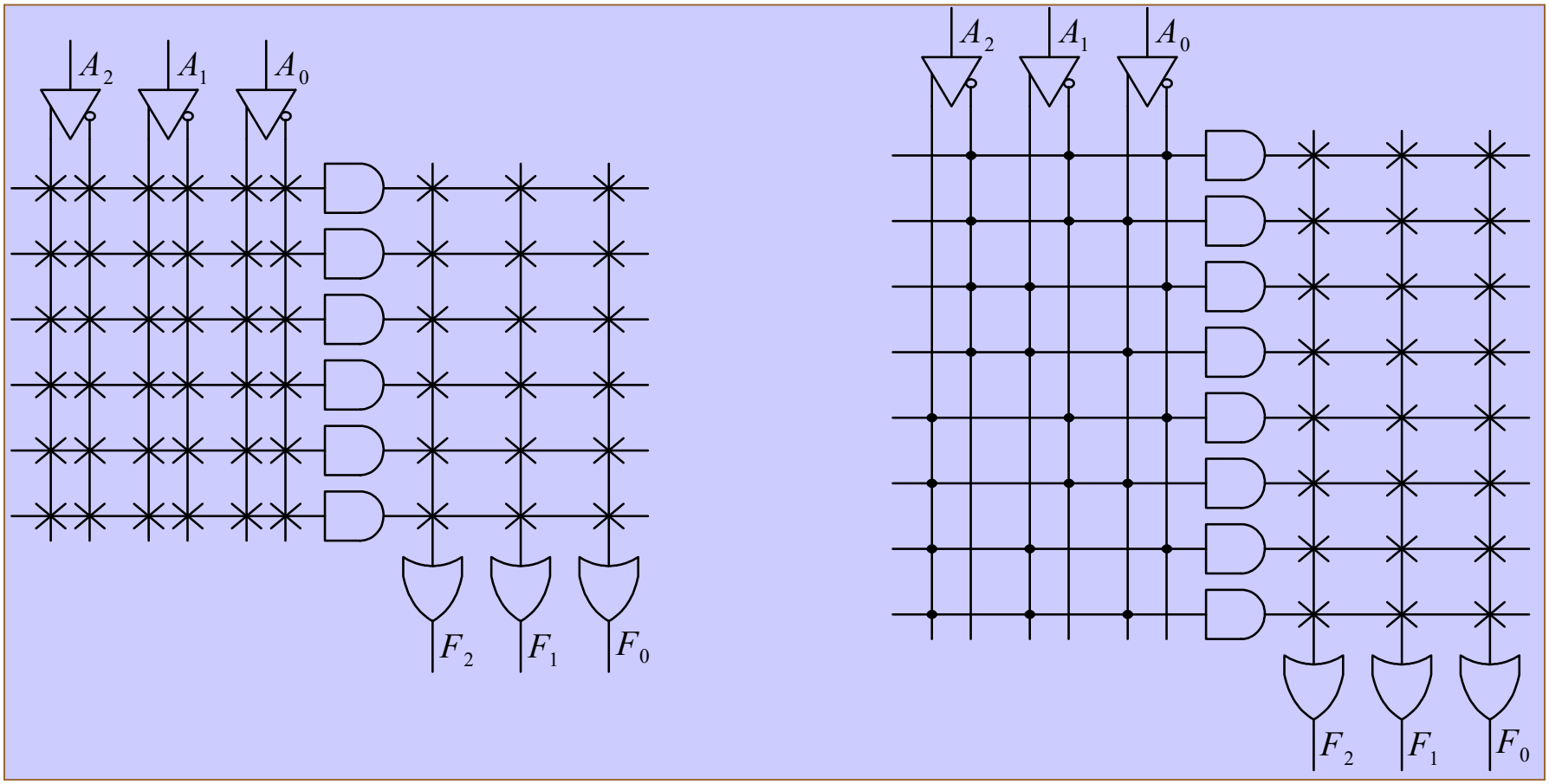
$$F1 = A1A0$$

PLA



PLA

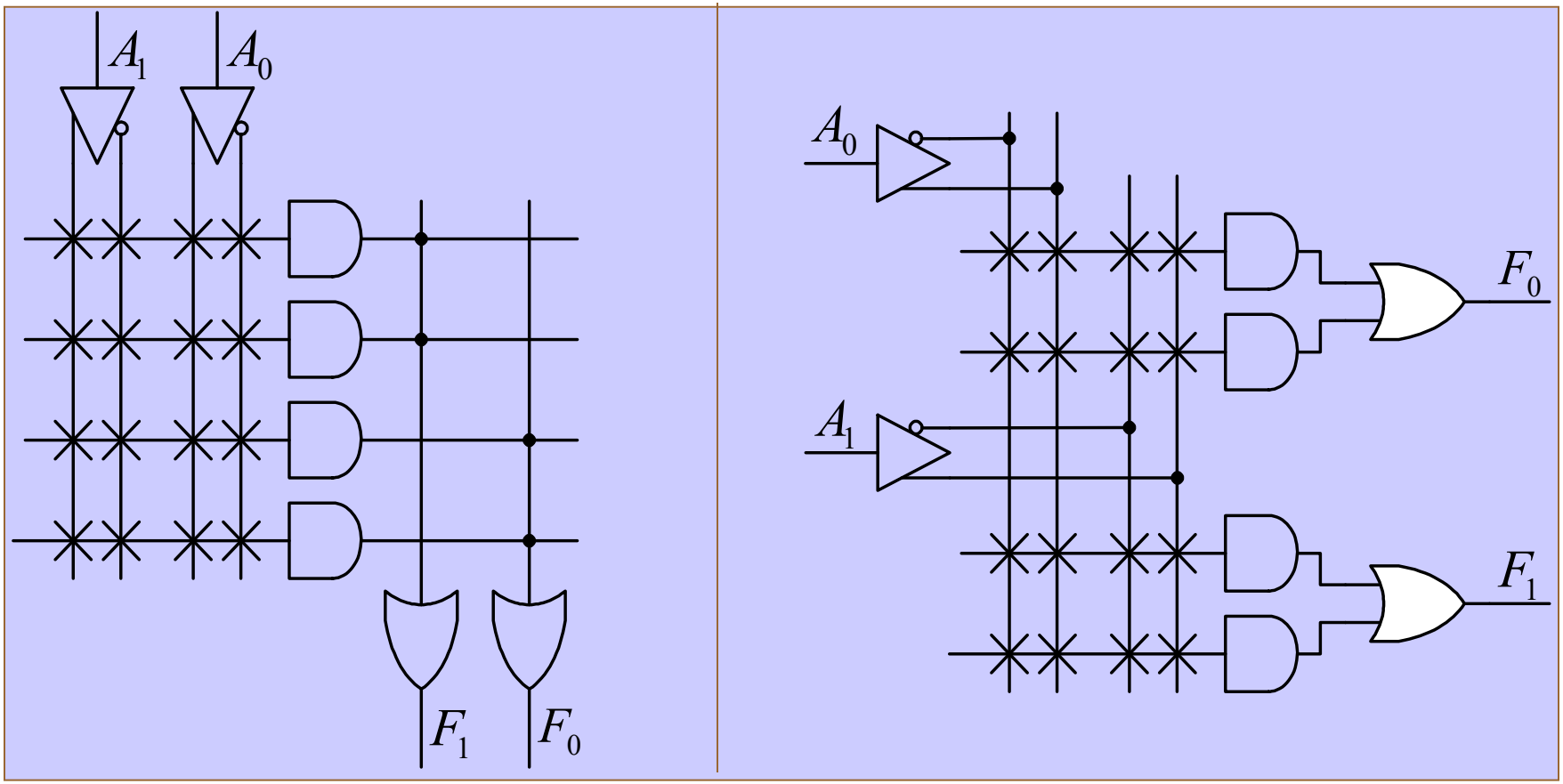
Fig. PLA vs. PROM



PAL

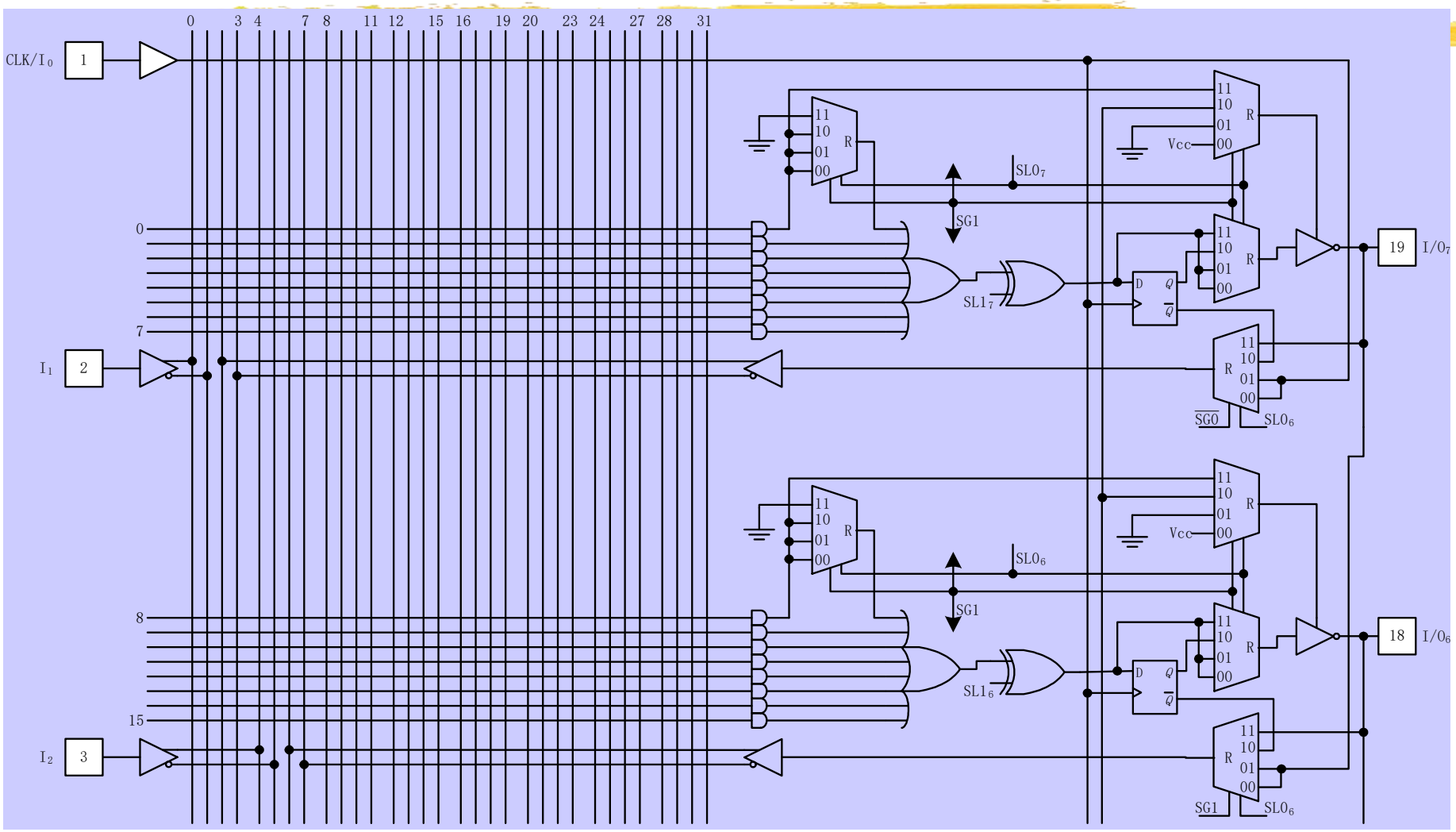
PAL:

Fig. PAL diagram:



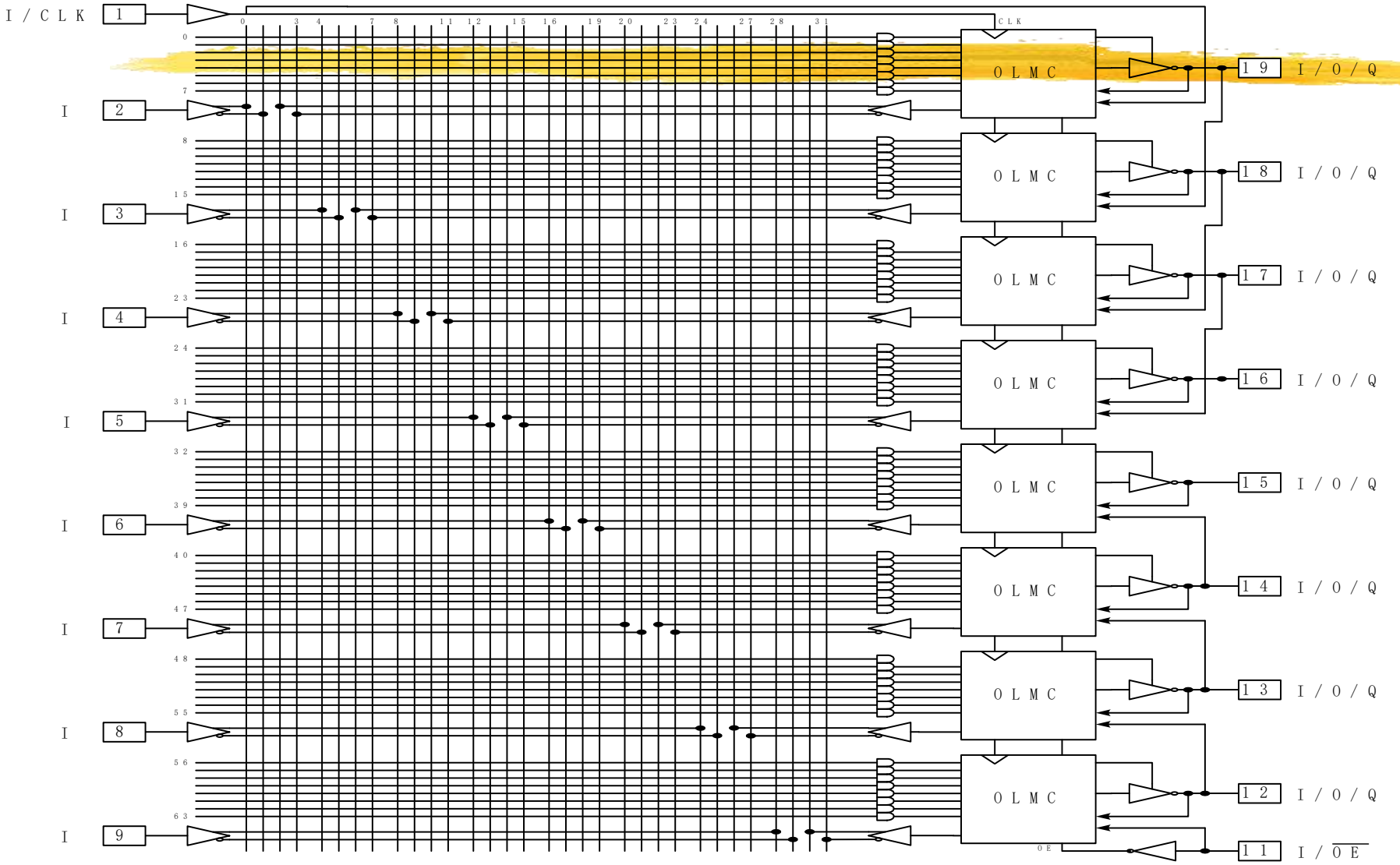
PAL

Fig. PAL16V8



GAL

GAL16V8


I / C L K
1
I
2
I
3
I
4
I
5
I
6
I
7
I
8
I
9
O L M C
O L M C
O L M C
O L M C
O L M C
O L M C
O L M C
O L M C
C L K
O E
1 9 I / O / Q
1 8 I / O / Q
1 7 I / O / Q
1 6 I / O / Q
1 5 I / O / Q
1 4 I / O / Q
1 3 I / O / Q
1 2 I / O / Q
1 1 I / $\overline{O E}$

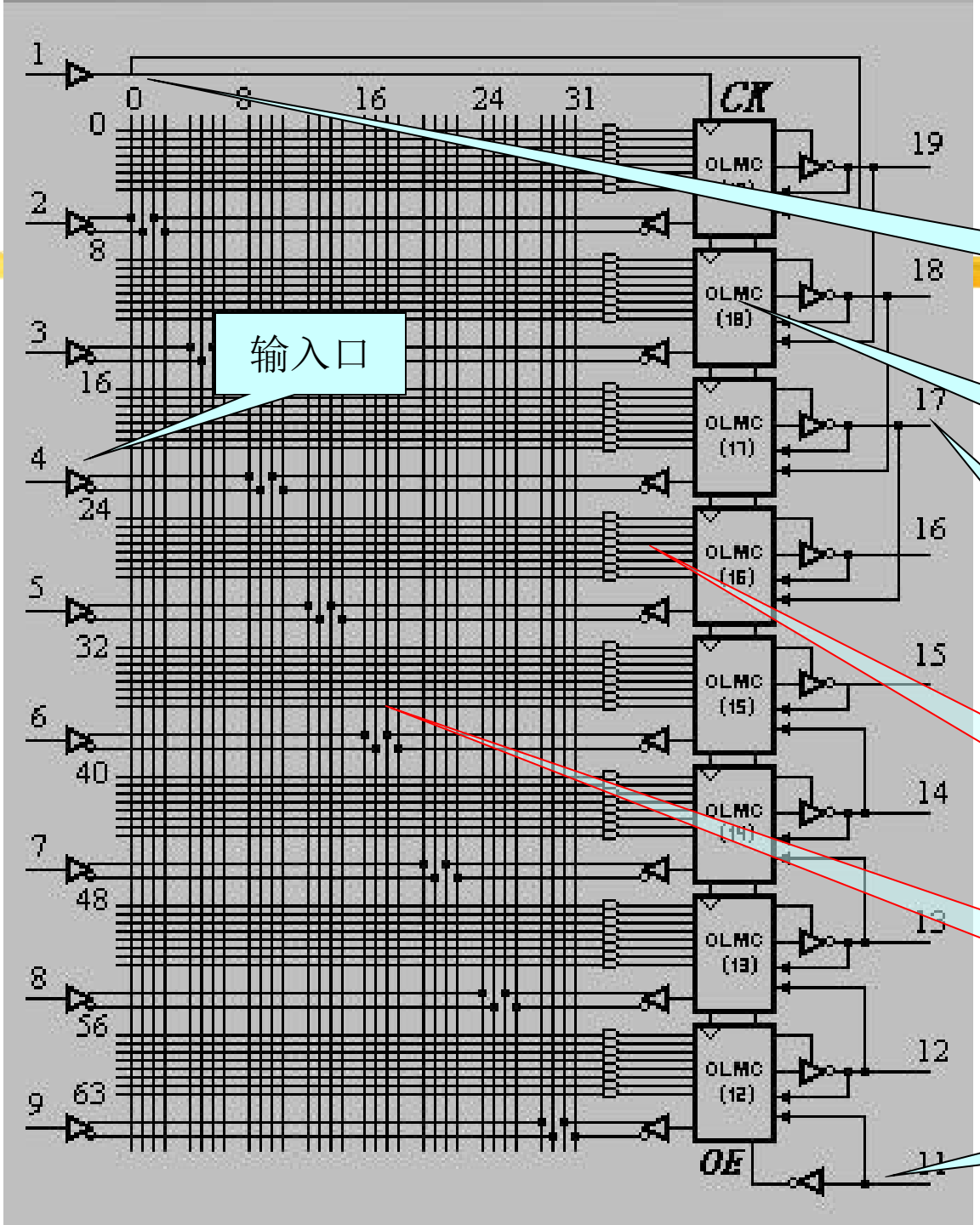
GAL

乘积项逻辑

GAL：
General Array Logic Device

EPLD
Erasable Programmable Logic Device

GAL16V8


1
0
8
16
24
31
CK
OLMC
19
2
8
OLMC
(18)
18
3
16
OLMC
(17)
17
4
24
OLMC
(16)
16
5
32
OLMC
(15)
15
6
40
OLMC
(14)
14
7
48
OLMC
(13)
13
8
56
OLMC
(12)
12
9
63
OE
11
输入口
时钟信号输入
逻辑宏单元
输入/输出口
固定或阵列
可编程与阵列
三态控制

GAL

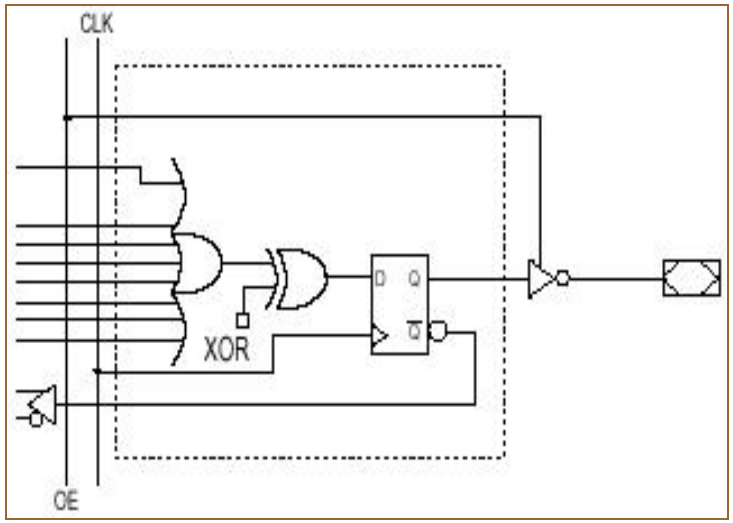
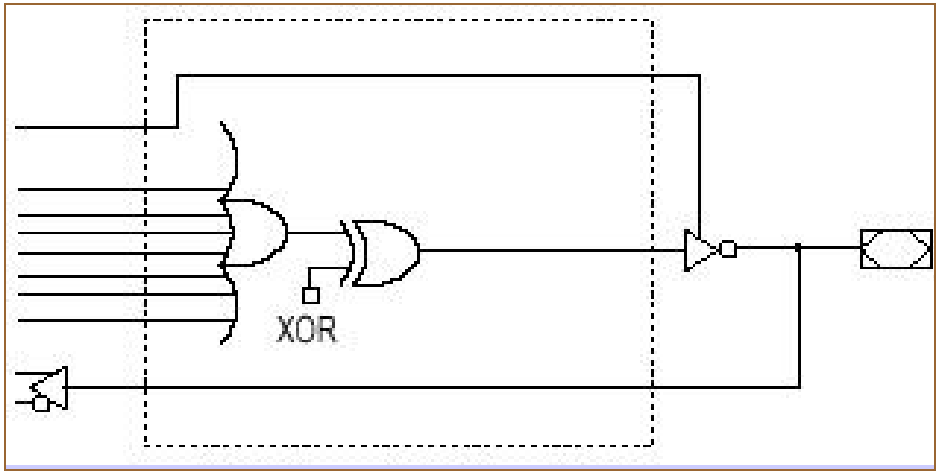


图 寄存器输出结构

图 存器模式组合双向输出结构



GAL

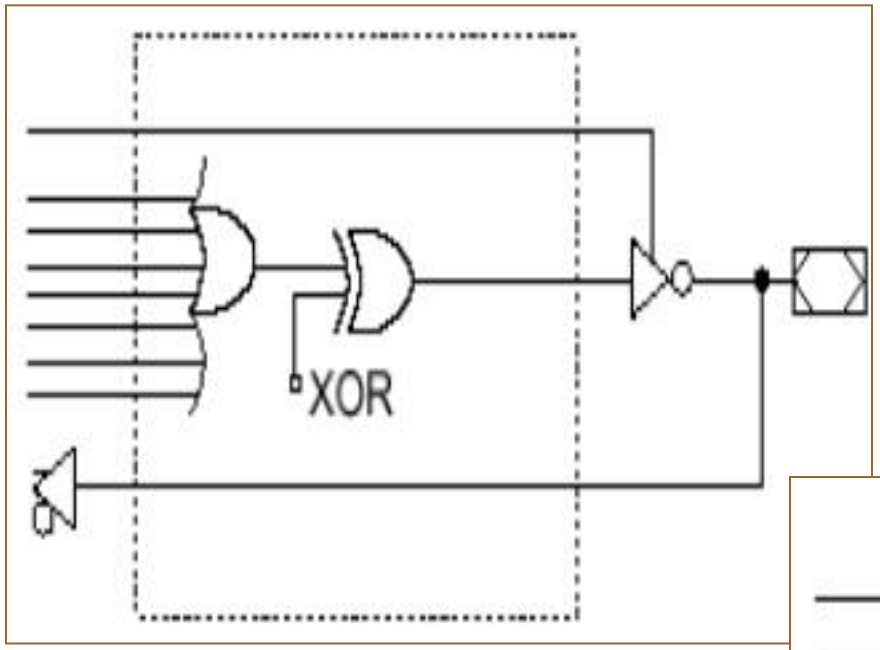
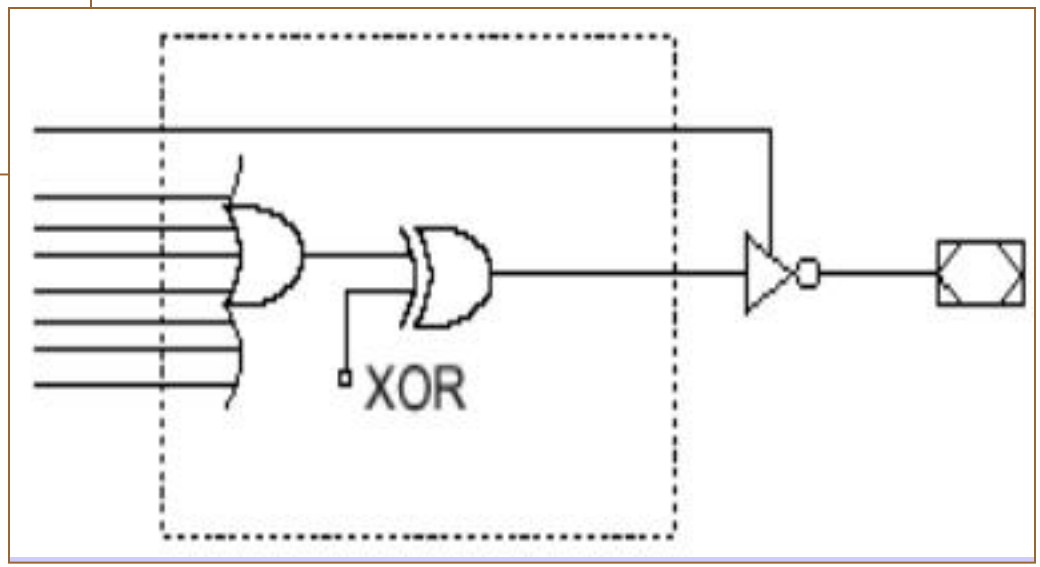


图 组合输出双向结构

图 复合型组合输出结构



GAL

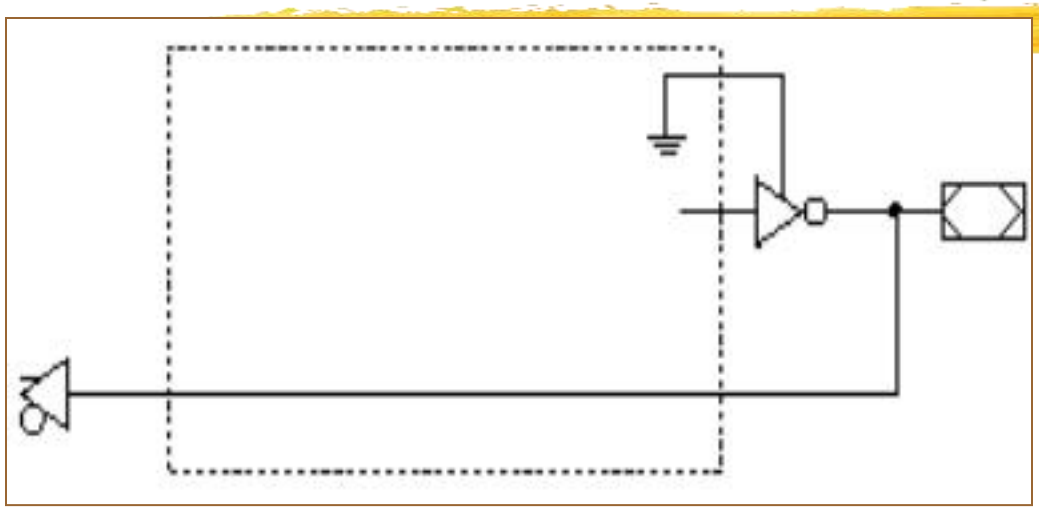
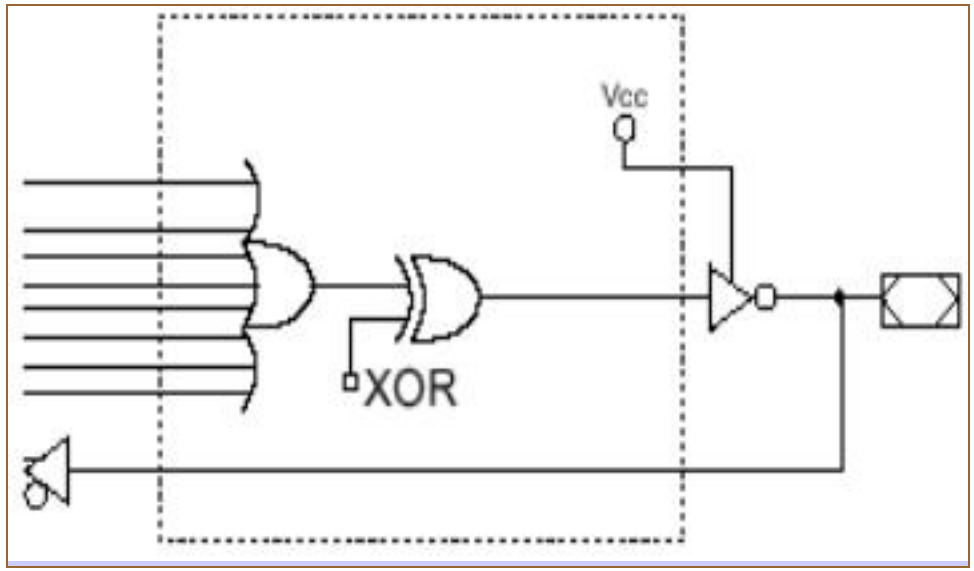


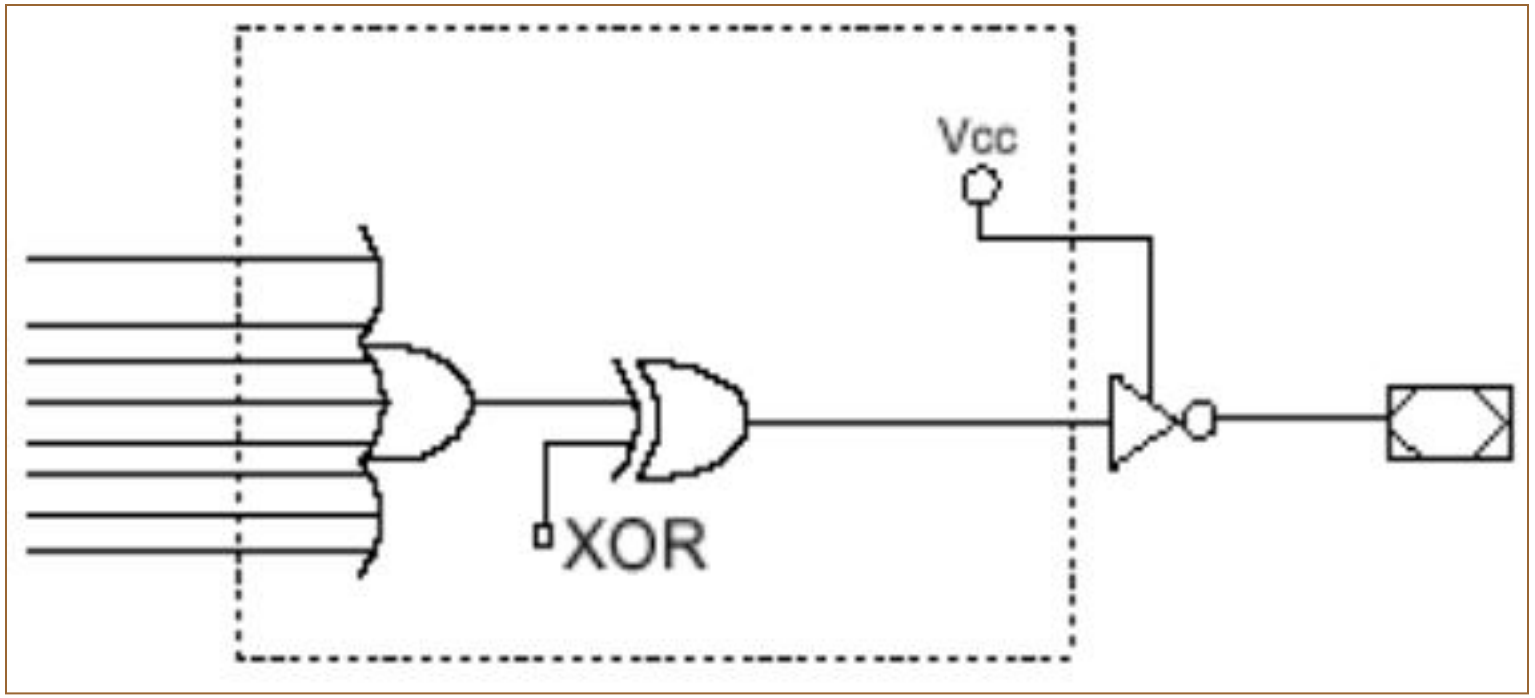
图 反馈输入结构

图 输出反馈结构

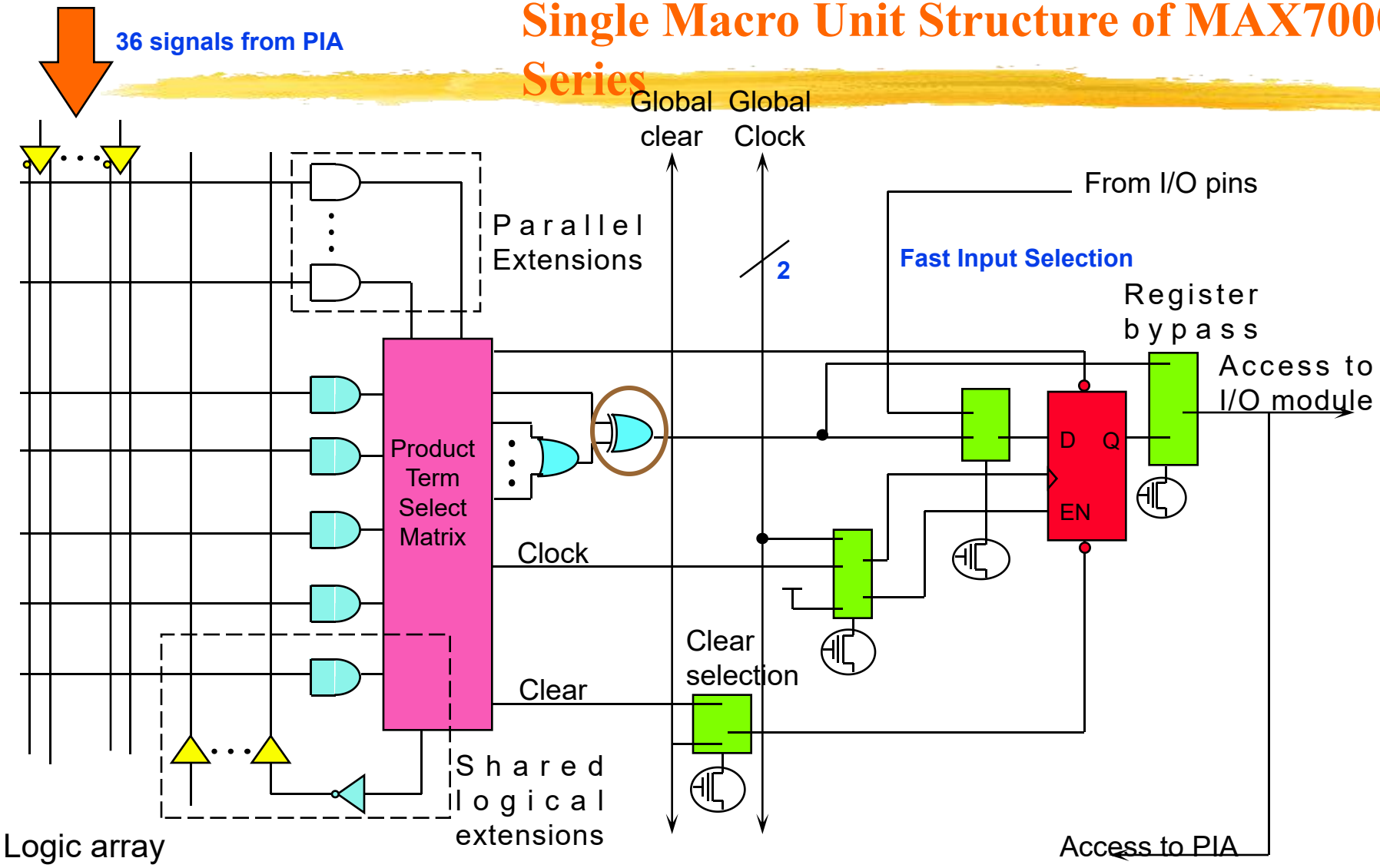


GAL

图　简单模式输出结构



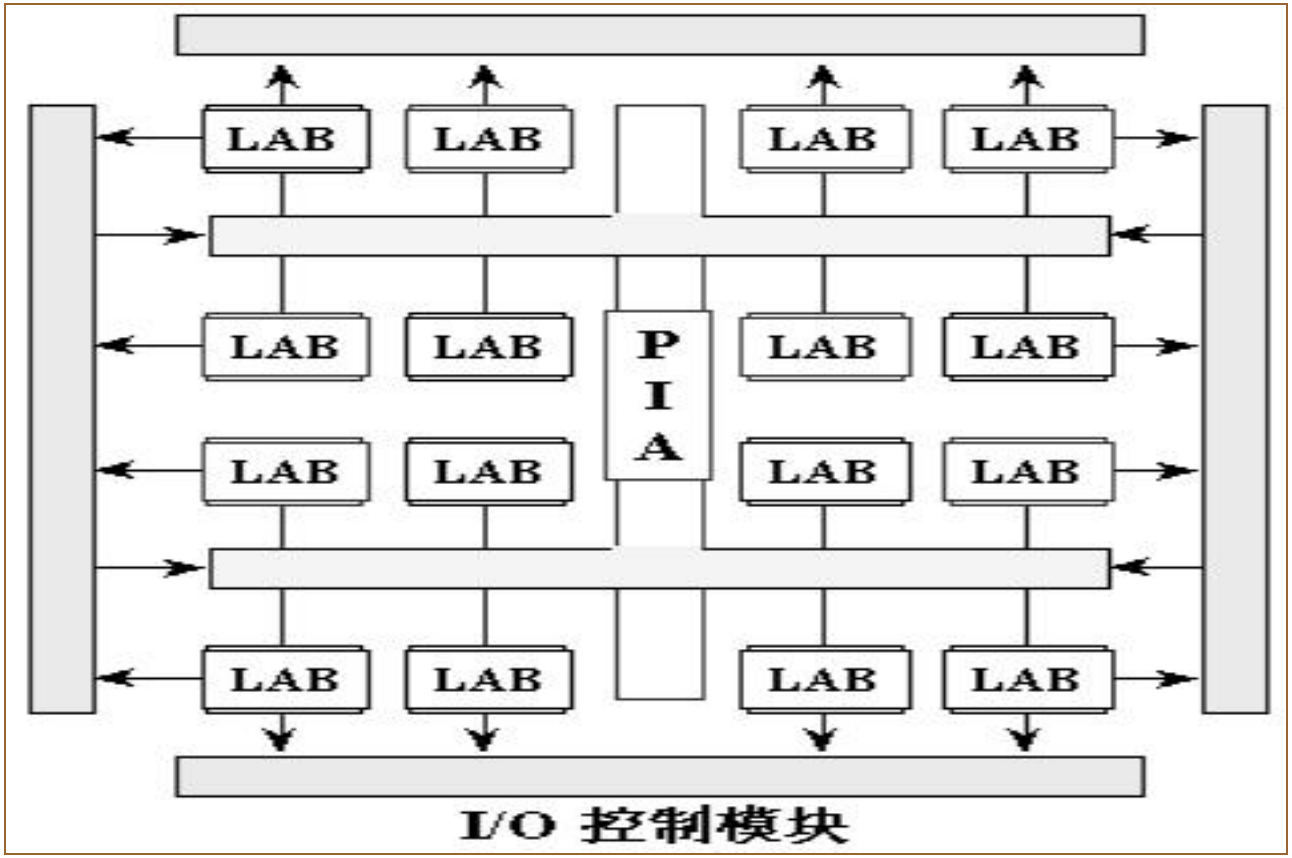
Single Macro Unit Structure of MAX7000 Series



CPLD

(1) 逻辑阵列块(Logic Array Block, LAB)

MAX7128S



CPLD

(2) 宏单元macro units

(3) 扩展乘积项 Extended product term

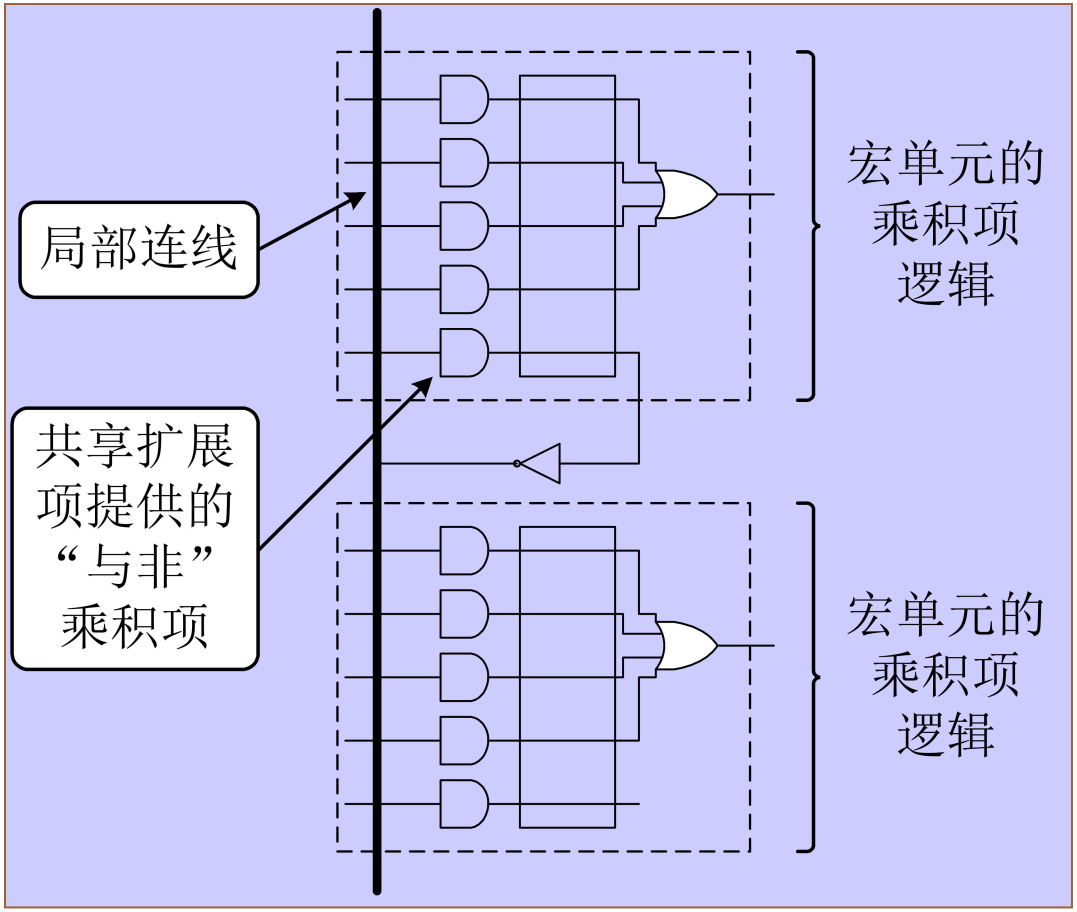
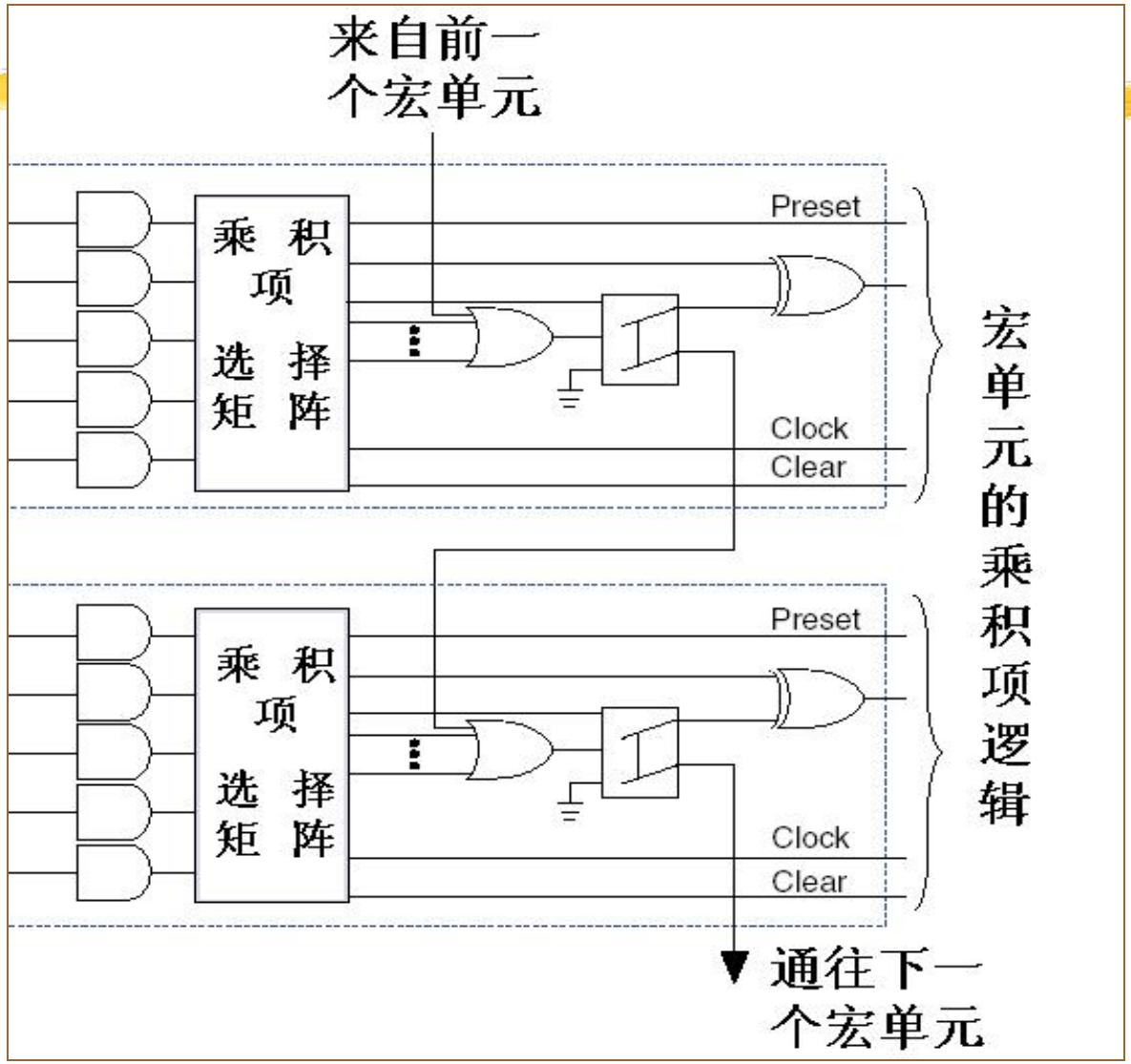


Figure The structure of the shared extended product term

Figure: Parallel Extension Feeding



CPLD

(4) 可编程连线阵列Programmable Interconnect Array

(5) 不同的LAB通过在可编程连线阵列(PIA)上布线，以相互连接构成所需的逻辑。 Different LAB wires are connected to the Programmable Interconnect Array (PIA) to form the required logic.

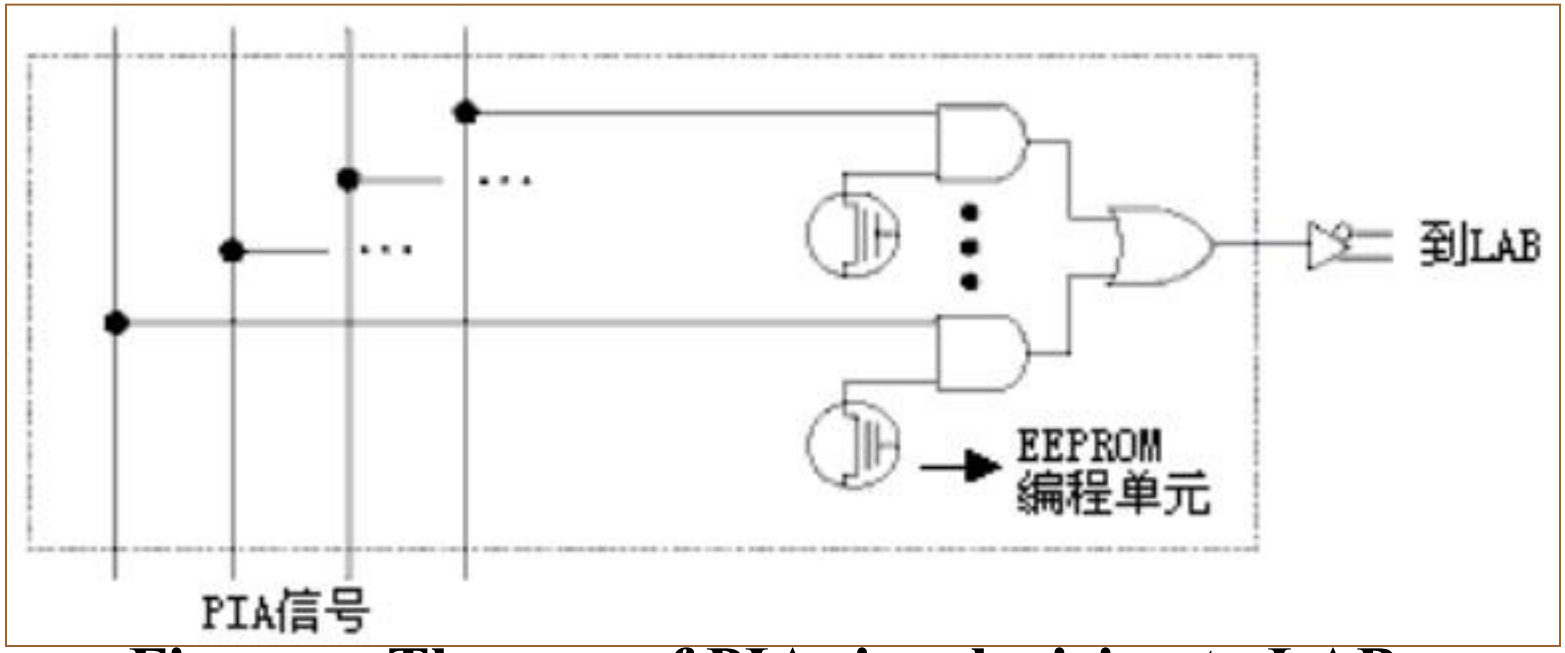


Figure The way of PIA signal wiring to LAB

(6)I/O控制块
I/O control block

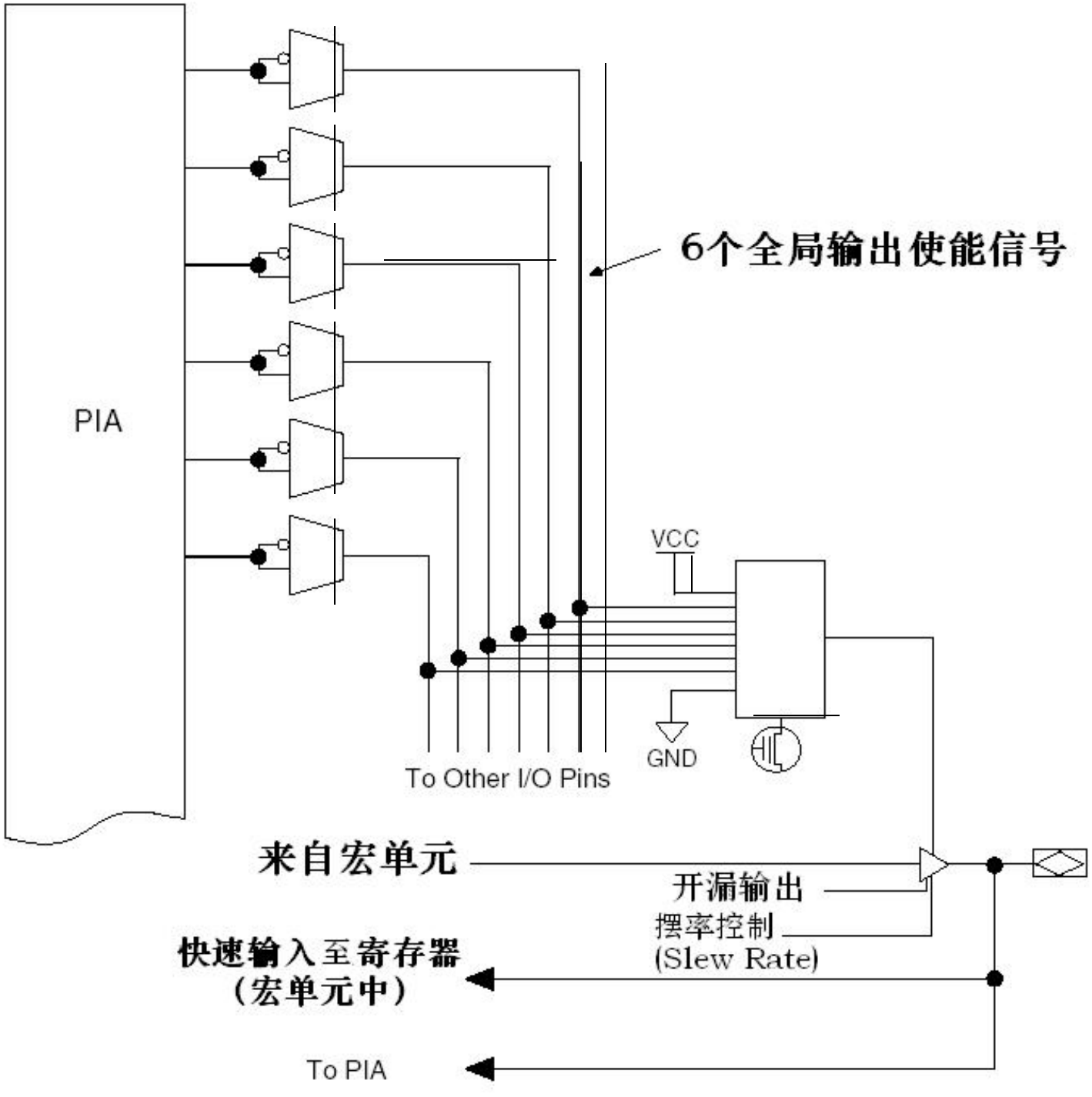
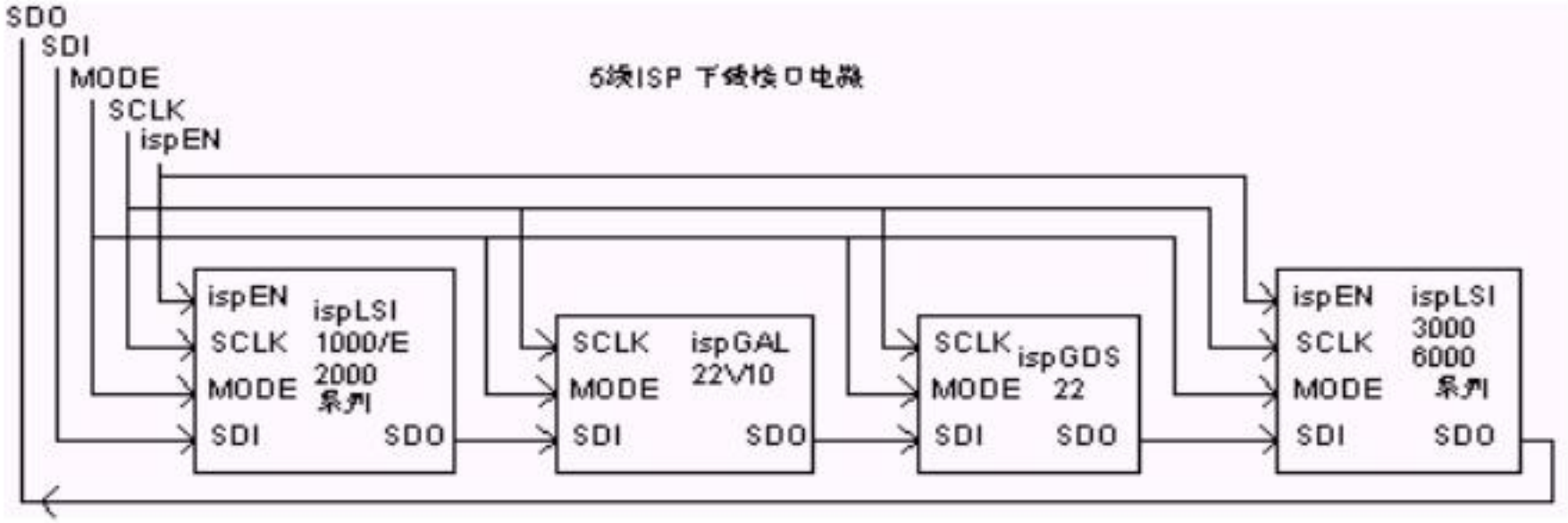


Figure- I/O Control Block of EPM7128S Device

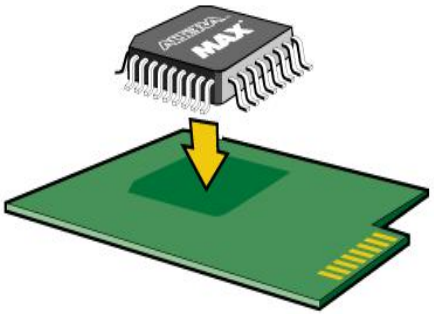
5、FPGA/CPLD Download

- CPLD
  - isp --IN-SYSTEM-PROGRAMMERBALE
- FPGA
  - SRAM
    1、直接配置（CONFIGUERING）
    2、ROM
    3、模拟ROM (analog ROM)
  - OTP



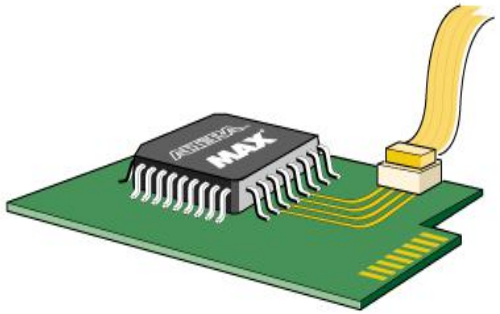
多芯片菊花链连接方式的 ISP 编程下载接口电路图

ISP功能提高设计和应用的灵活性



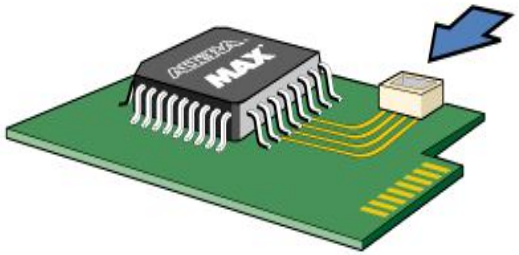
未编程前先焊接安装

- 减少对器件的触摸和损伤
- 不计较器件的封装形式



系统内编程--ISP

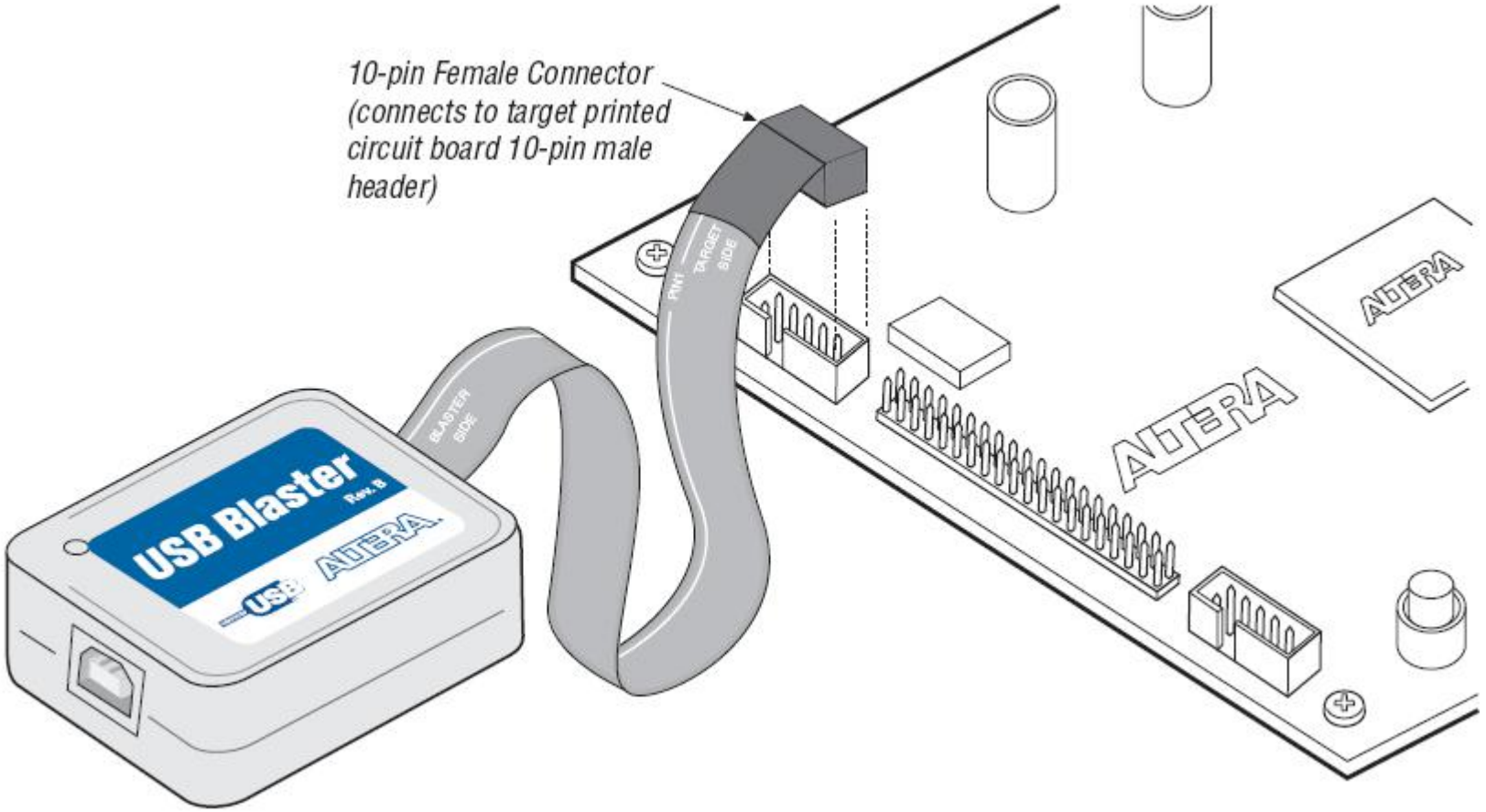
- 允许一般的存储
- 样机制造方便
- 支持生产和测试流程中的修改



在系统现场重编程修改

- 允许现场硬件升级
- 迅速方便地提升功能

ALTERA's USB BlasterII Download Interface

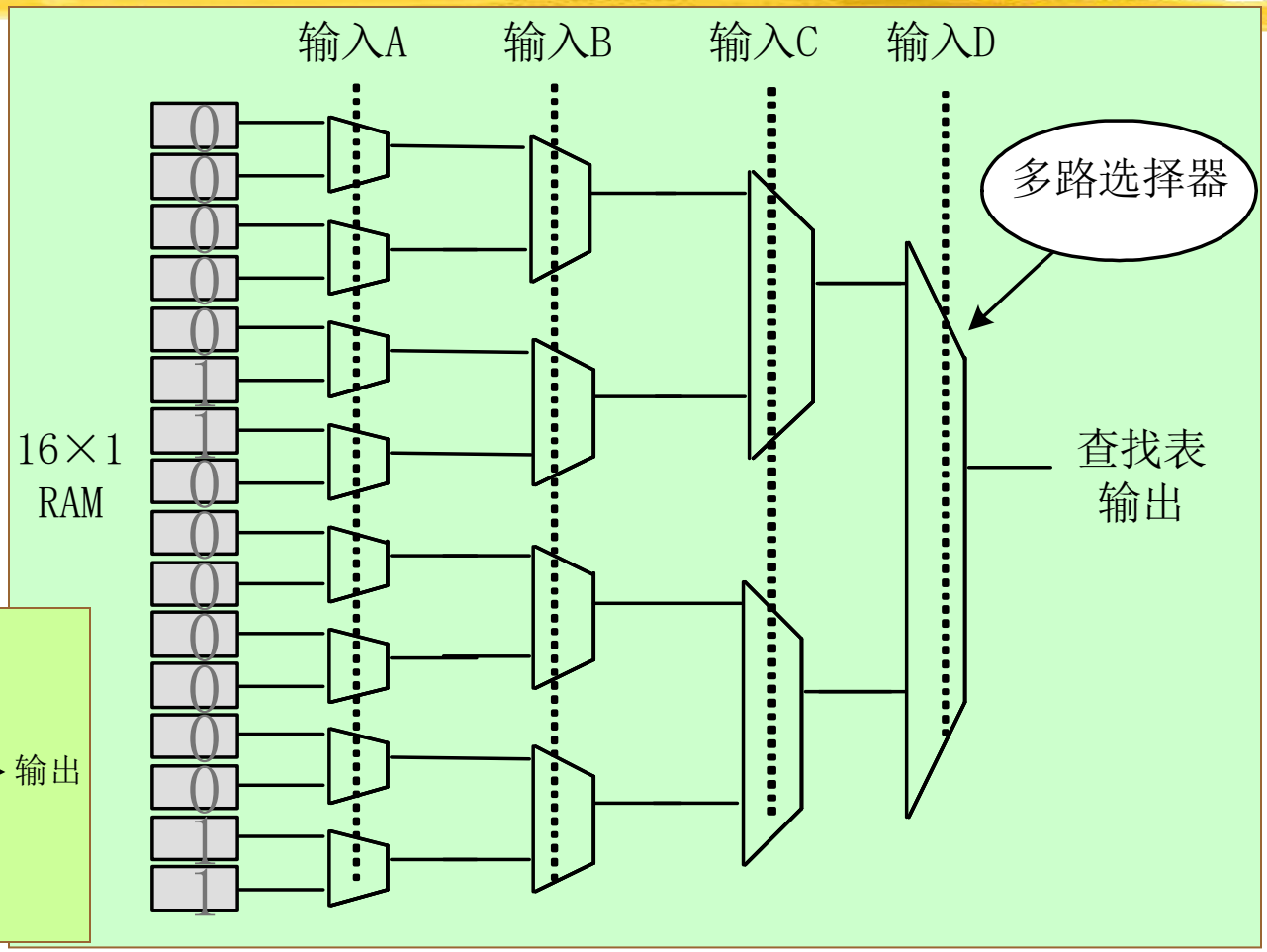


USB Blaster

- USB
  - Altera USB-Blaster
    - PC→USB→FPGA/CPLD JTAG
    - LPT（打印机口、并口 ）
  - Altera ByteBlasterII
  - Altera ByteBlasterMV
- Ethernet
  - Altera EthernetBlaster
  - 远程升级

FPGA

LUT



FPGA LUT：

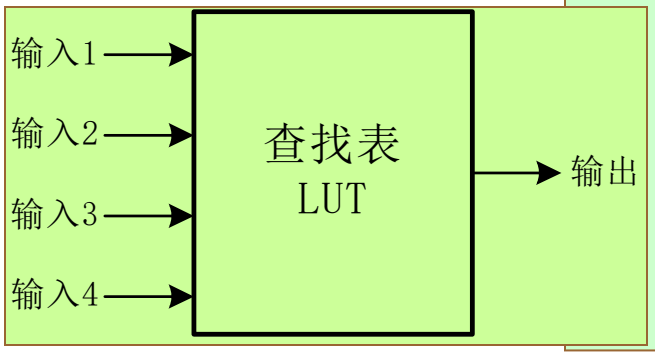
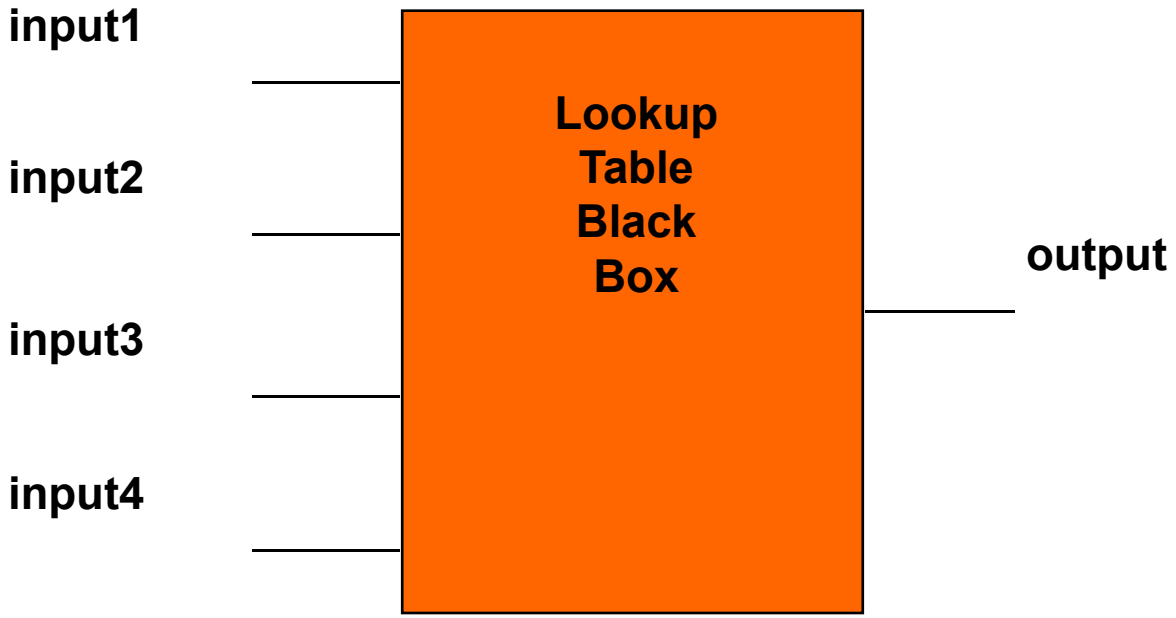


Figure Internal structure of FPGA LUT unit

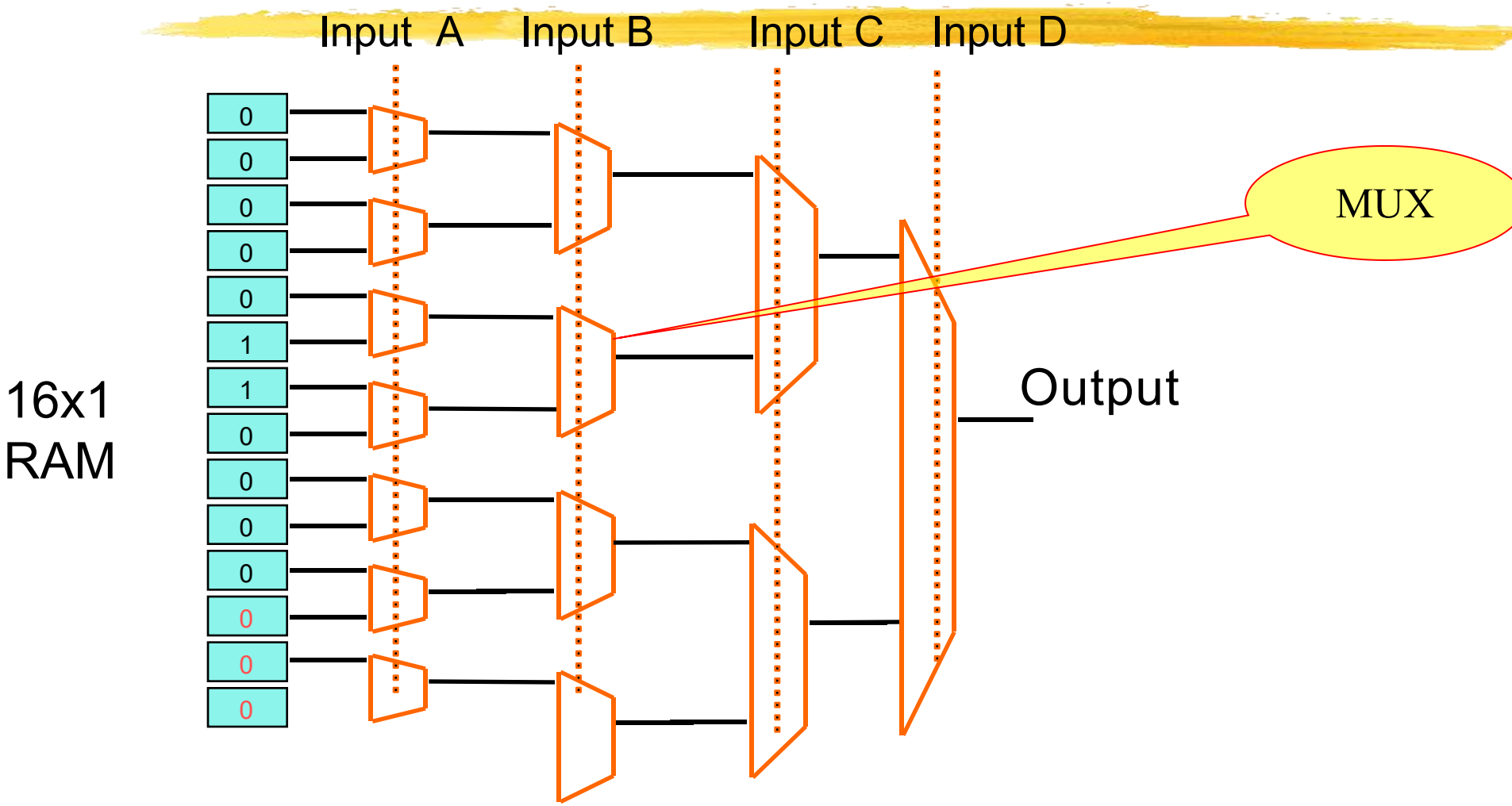
Structural module based on lookup table

LUT ?

- An N-input lookup table (LUT) can implement any logic function of N input variables, such as N-input AND, N-input XOR, etc.
- Functions and equations with more than N inputs must be implemented separately with several lookup tables (LUTs).



LUT


Input A
Input B
Input C
Input D
0
0
0
0
0
1
1
0
0
0
0
0
0
0
MUX
Output
16x1
RAM

主观题

设置

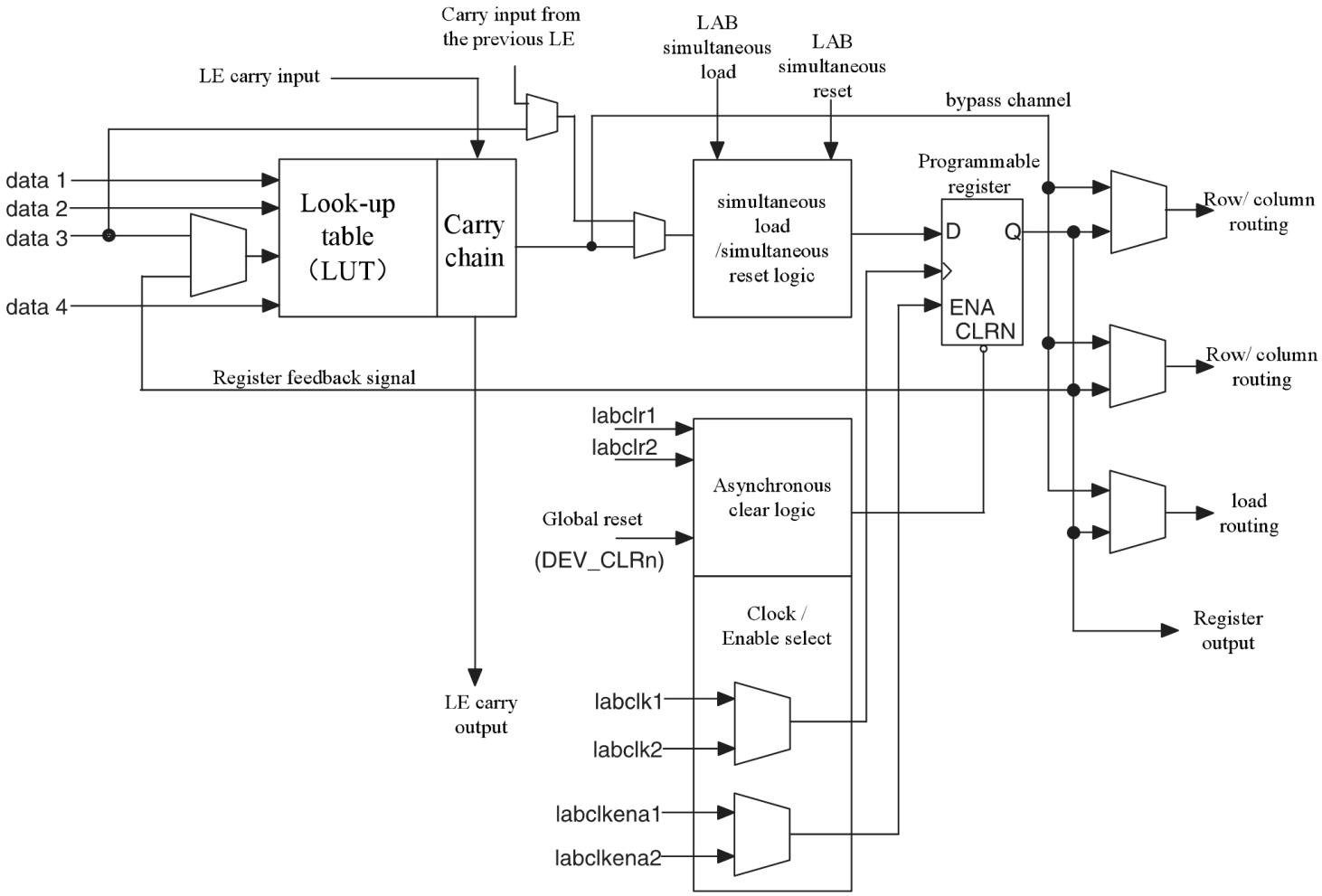
FPGA和CPLD有没有什么区别？

Is there any difference between FPGA and CPLD?

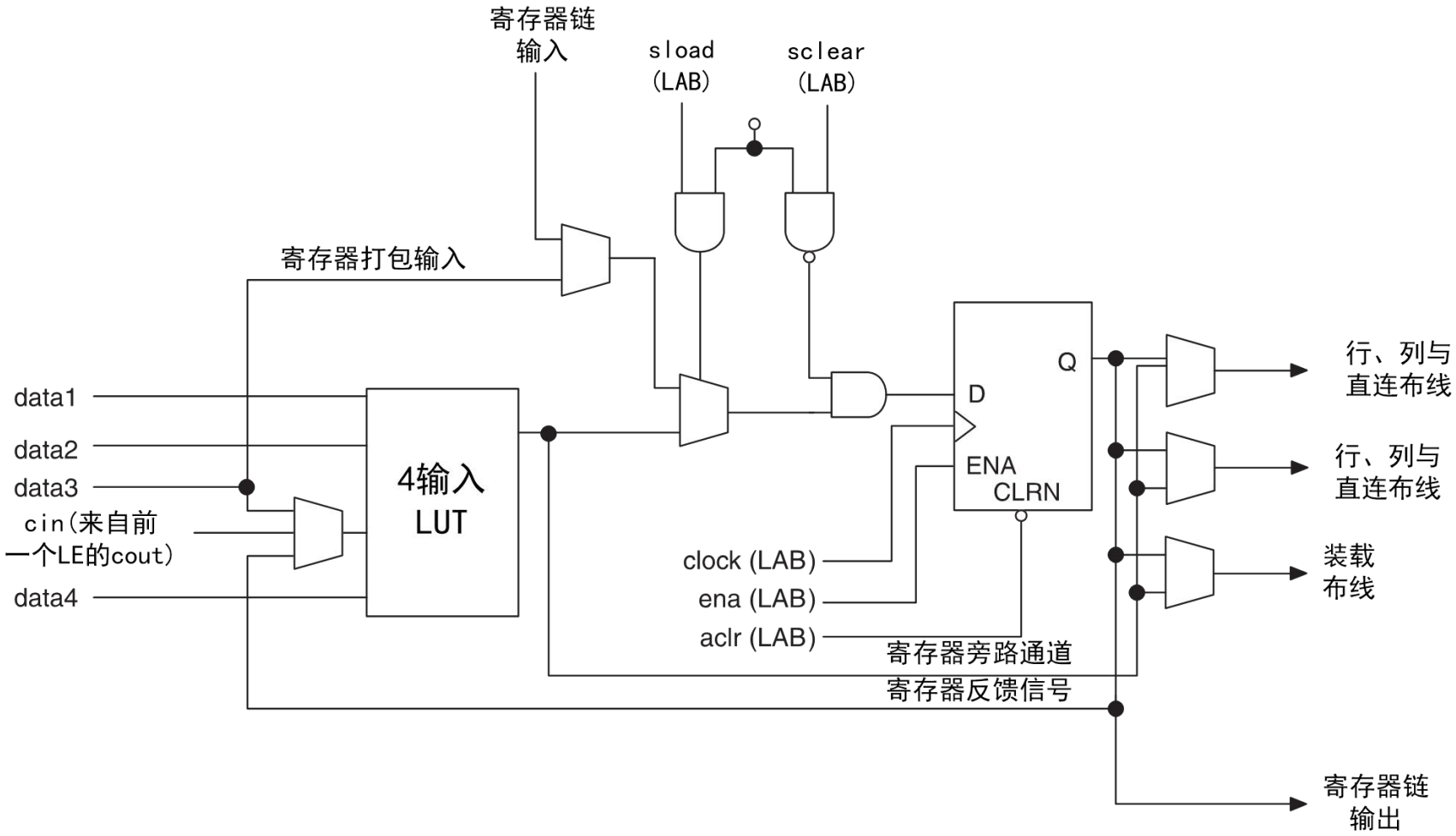
正常使用主观题需2.0以上版本雨课堂

作答

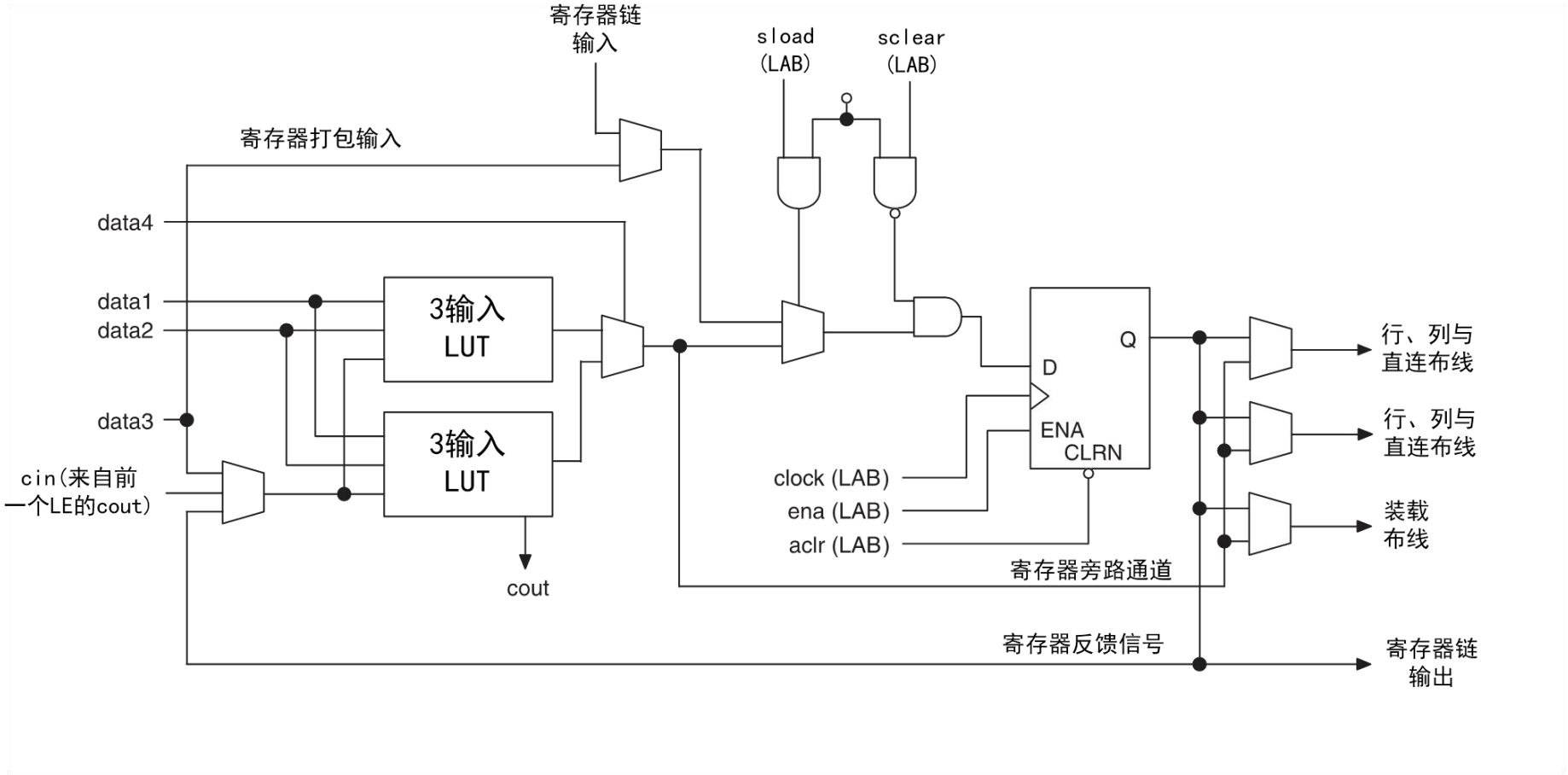
LE structure diagram of Cyclone 4E



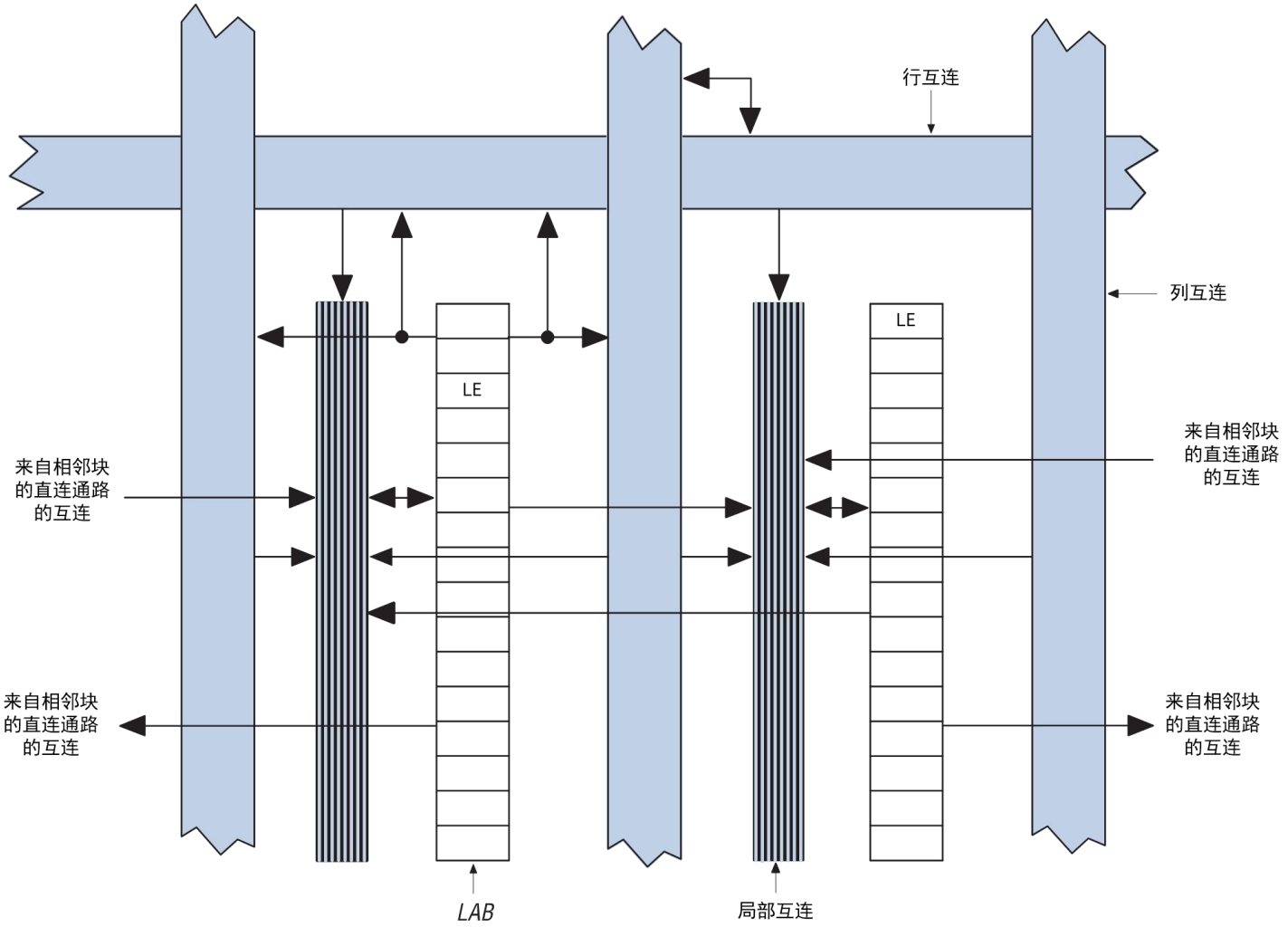
Cyclone 4E LE General Model



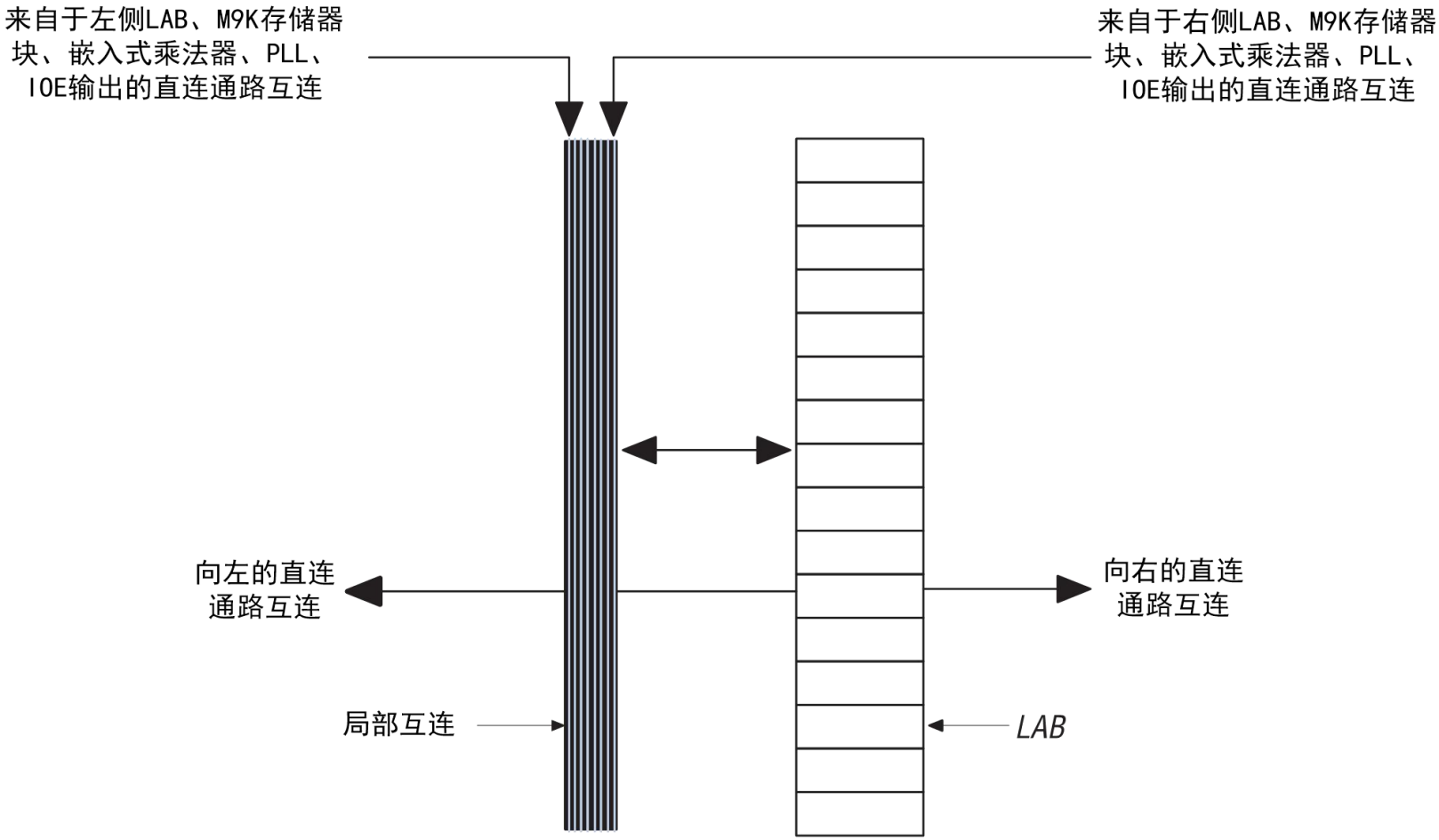
Cyclone 4E LE Dynamic Arithmetic Mode



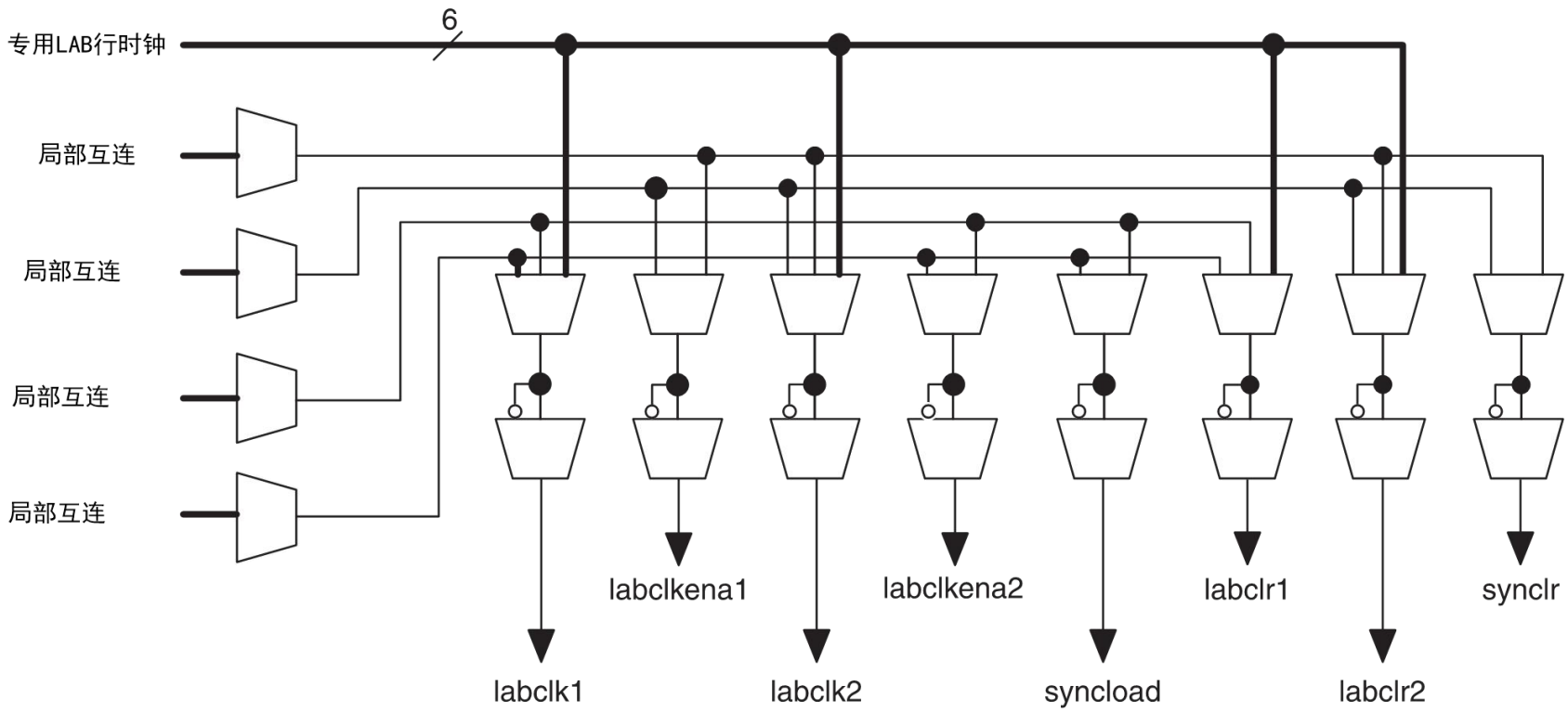
Cyclone 3 LAB structure



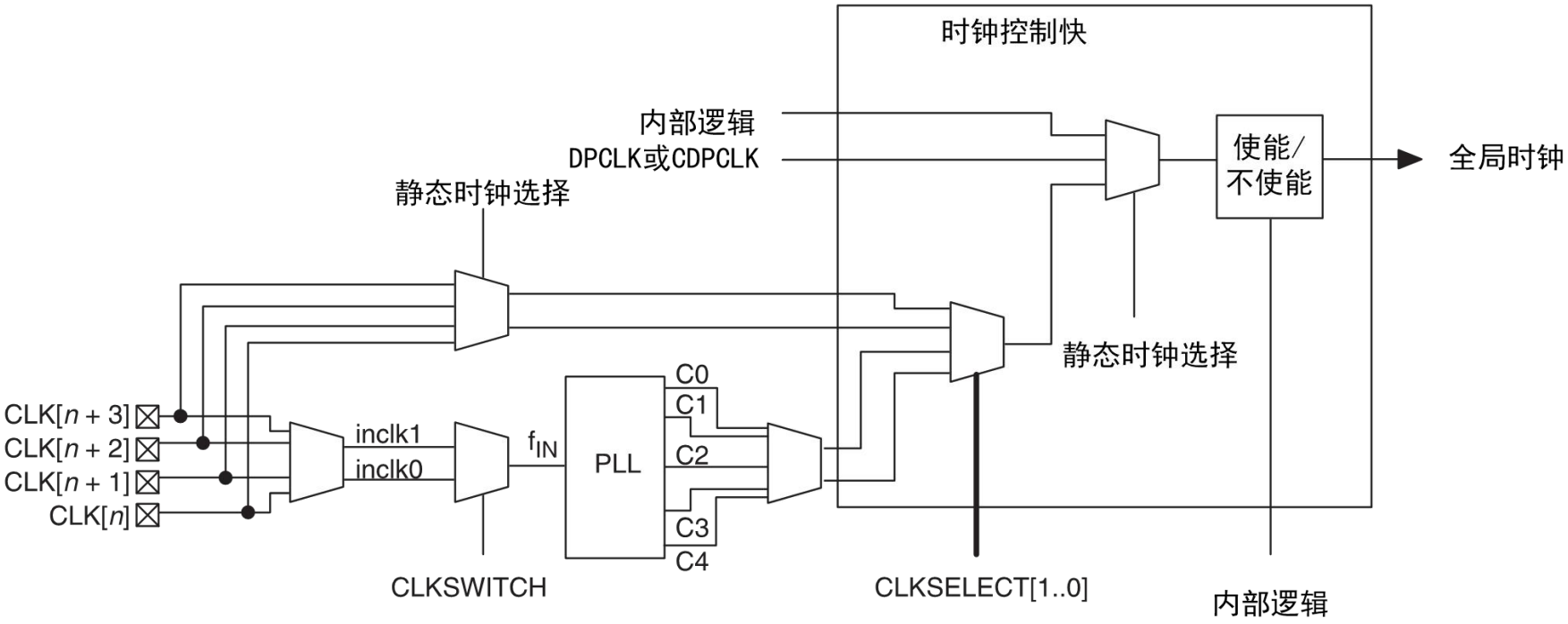
Interconnection between LAB arrays



Control Signal Generation of LAB



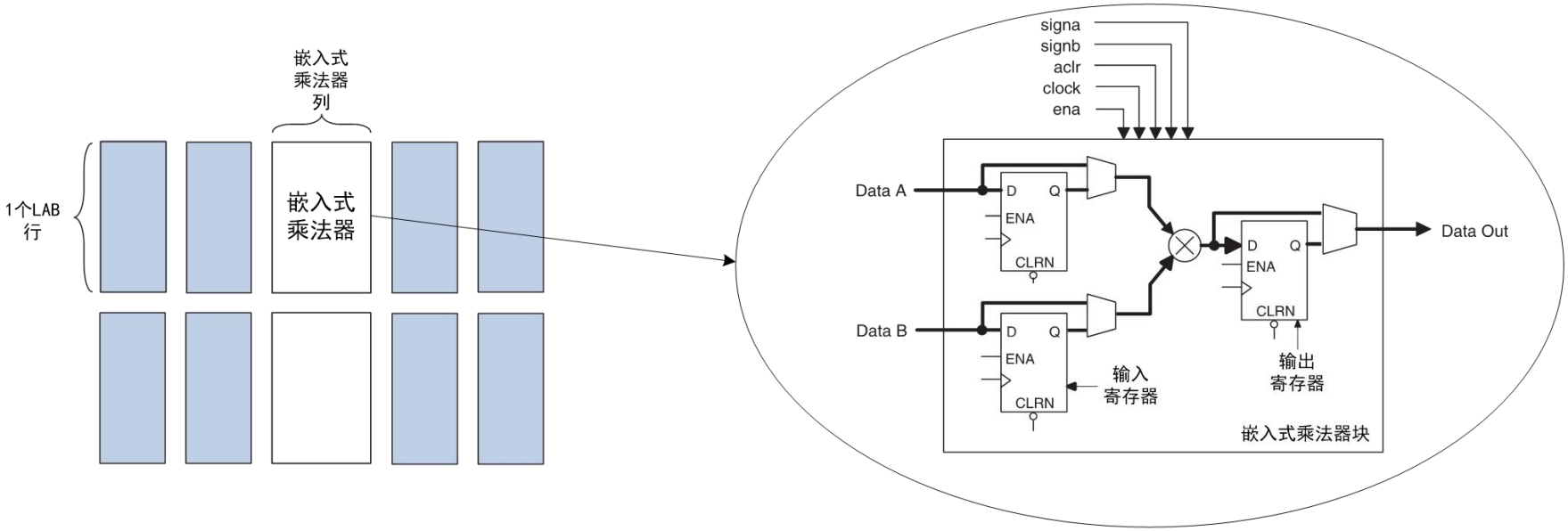
Clock Control in Clock Network



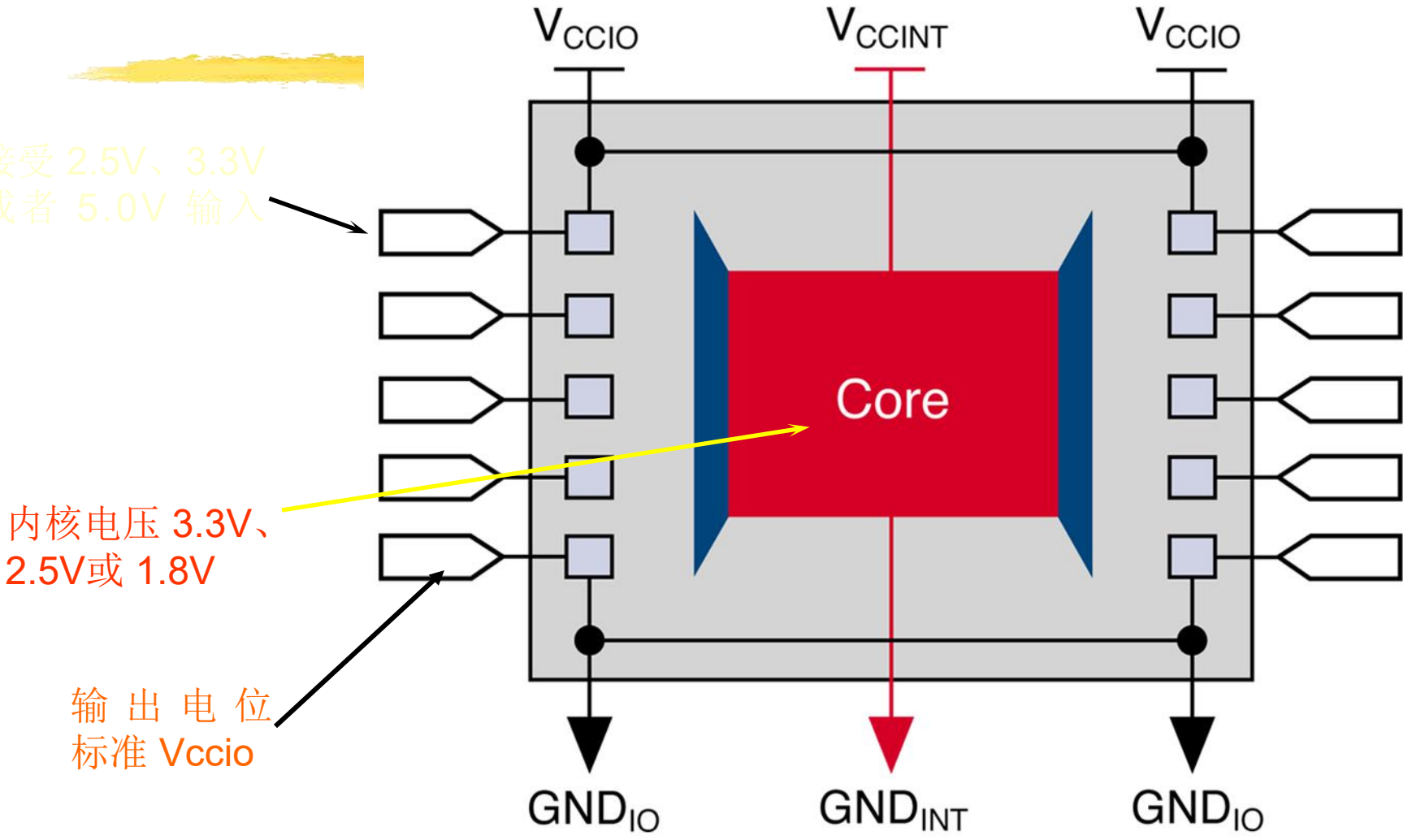
Embedded function &module

- Embedded memory
- Multiplier
- Float Point Function
- PLL
- LVDS
- External memory controller
- ……

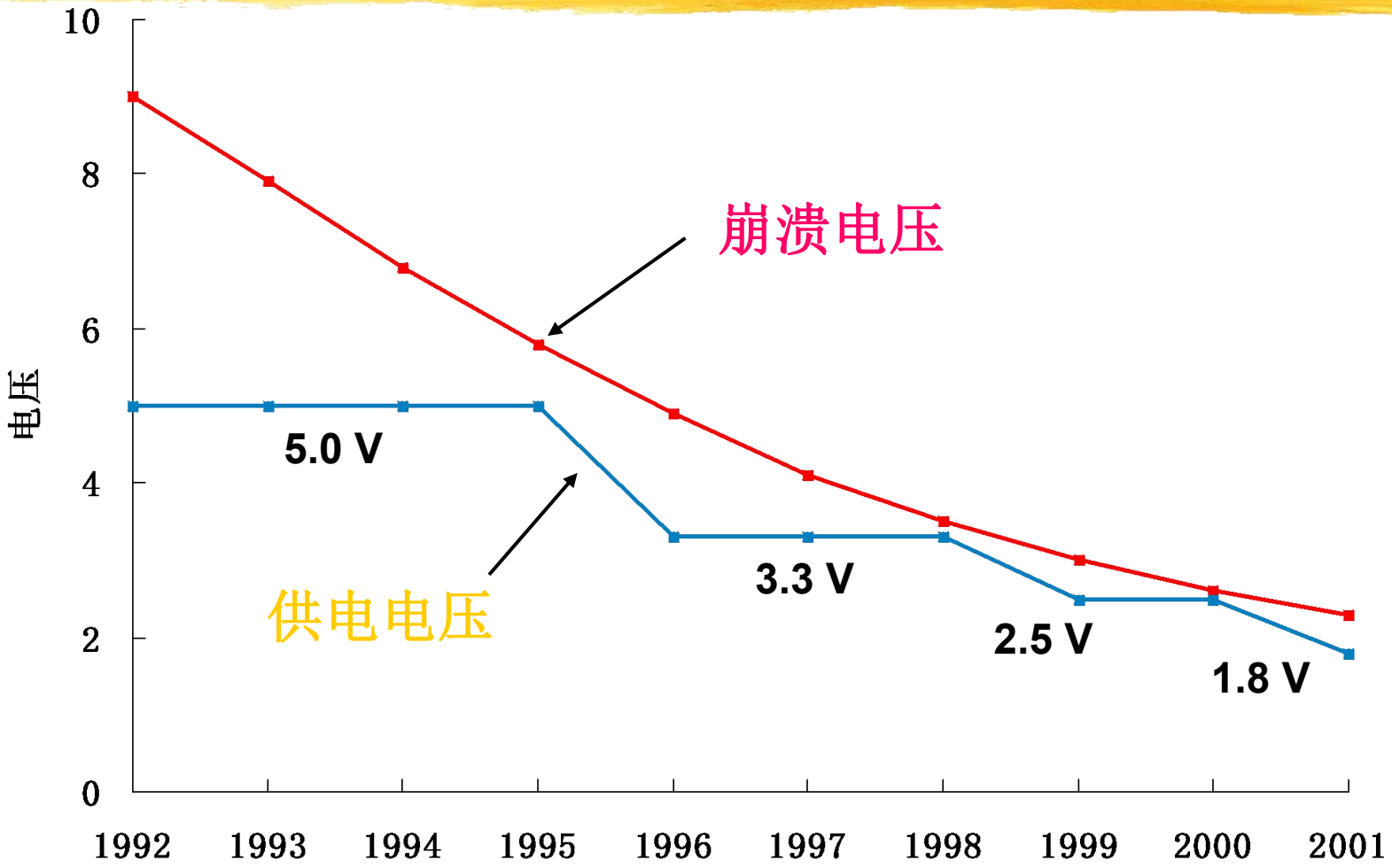
Multiplier


嵌入式
乘法器
列
1个LAB
行
嵌入式
乘法器
signa
signb
aclr
clock
ena
Data A
Data B
D
Q
ENA
CLRN
Data Out
输入
寄存器
输出
寄存器
嵌入式乘法器块

FPGA/CPLD多电压兼容系统



Power supply



PLD Provider

- Xilinx
- Intel PSG（Altera）
- MciroChip（收购MicroSemi）
- Lattice Semiductor
- China:
- AGM
- GoWin
- 国微

FPGA/CPLD Test

Internal Logic Test

JTAG Boundary Scan Test

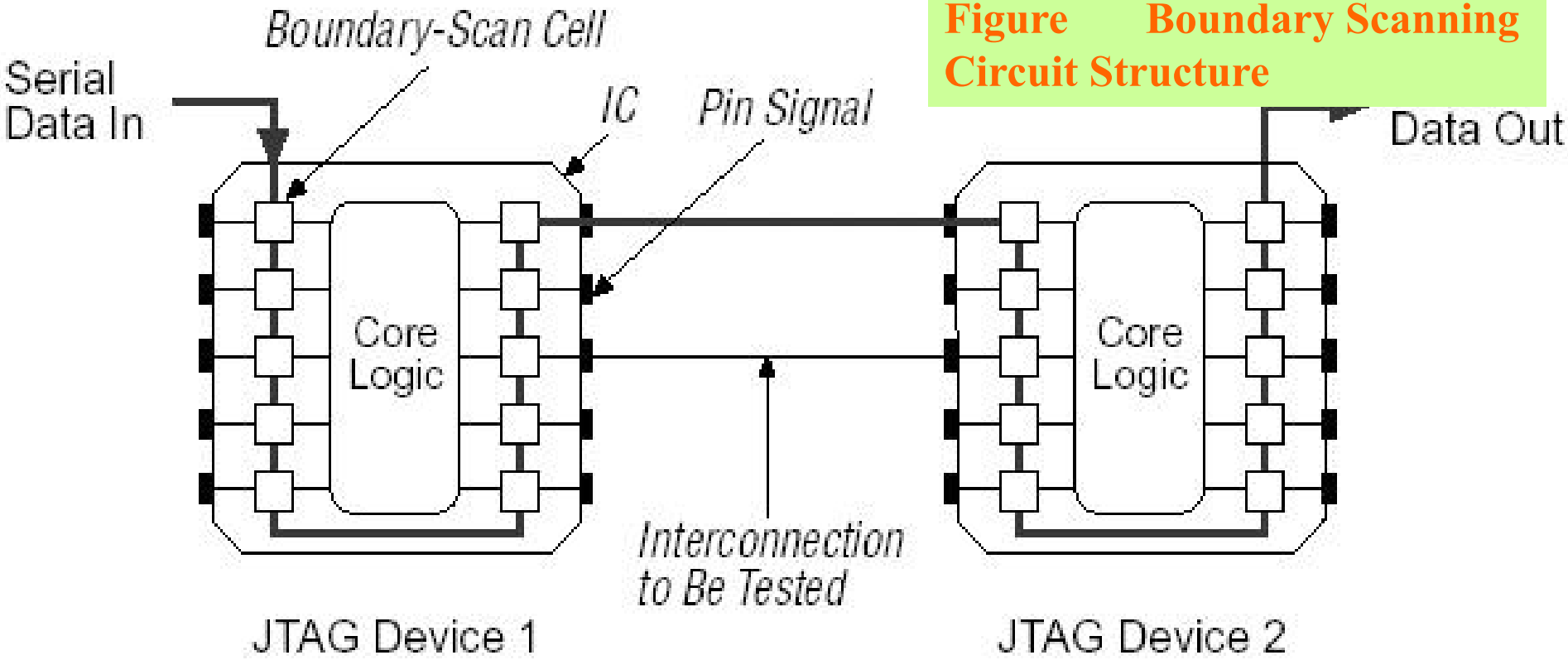
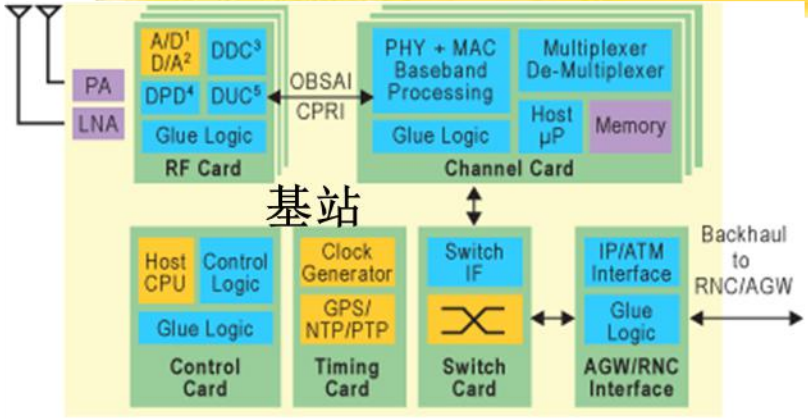


Figure Boundary Scanning Circuit Structure

FPGA application system



汽车电子 **RADAR** 自适应巡航控制系统

Intel PSG

- FPGA
  - Statix 10/Stratix V/Stratix IV
  - Cyclone 10/Cyclone V/Cyclone IV
  - Array 10/ Array V
  - MAX 10
- CPLD
  - MAX V/MAX II
  - MAX3000A

HARDCOPY™II

Xilinx

- FPGA
  - Artix-7、Kintex-7 、Virtex 7/Virtex 6
  - Zynq UltraScale、Zynq
  - Spartan 6
- CPLD
  - XC9500XL
  - CoolRunner II

MicroChip(MicroSemi)

- FPGA
  - Fusion
  - IGLOO
  - ProASIC3
- SoC FPGA
  - SmartFusion2 (CM3)

Lattice

- FPGA
  - iCE
  - LatticeXP
  - LatticeECP
- CPLD
  - MachXO
  - ispMACH

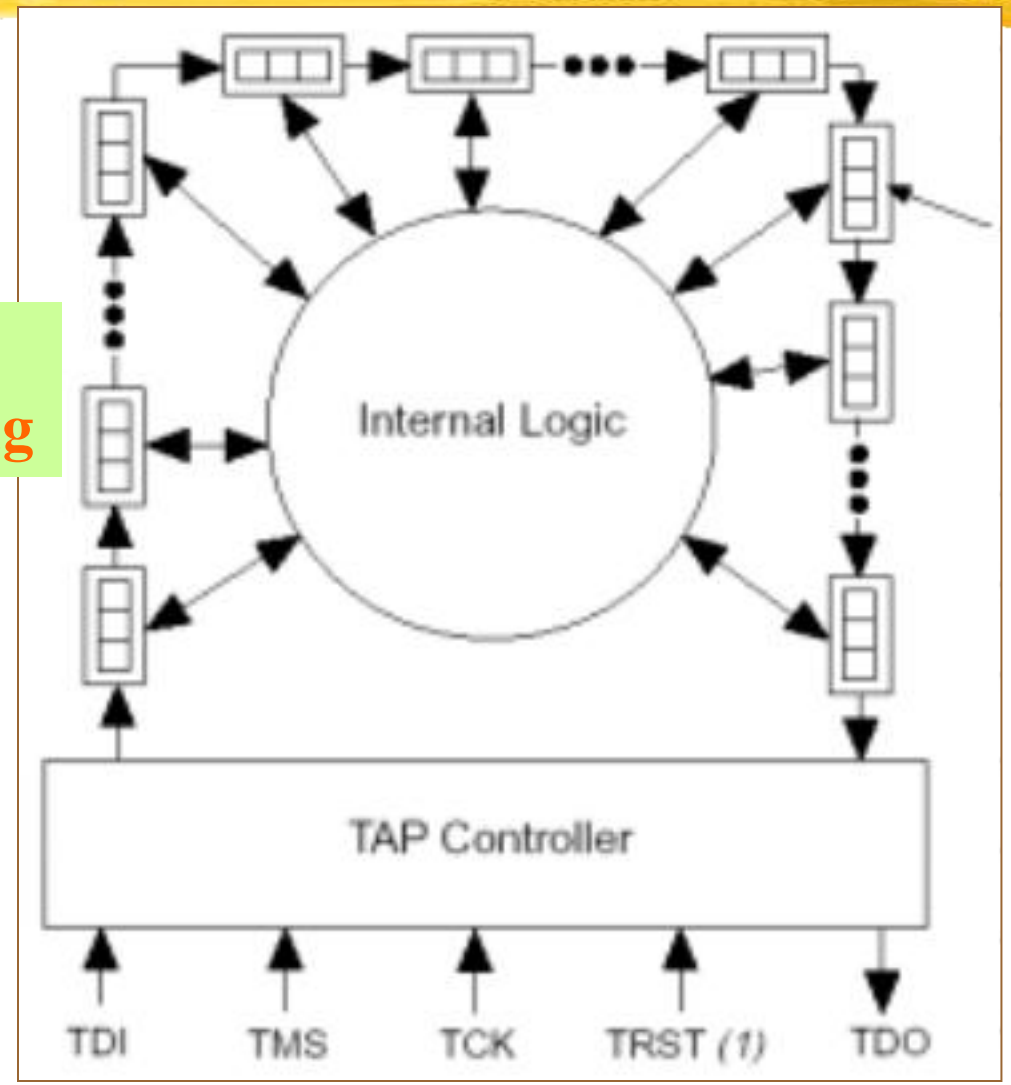
JTAG Test

IO function

引　脚	描　述	功　能
TDI	测试数据输入 (Test Data Input)	测试指令和编程数据的串行输入引脚。数据在 TCK 的上升沿移入。
TDO	测试数据输出 (Test Data Output)	测试指令和编程数据的串行输出引脚，数据在 TCK 的下降沿移出。如果数据没有被移出时，该引脚处于高阻态。
TMS	测试模式选择 (Test Mode Select)	控制信号输入引脚，负责 TAP 控制器的转换。TMS 必须在 TCK 的上升沿到来之前稳定。
TCK	测试时钟输入 (Test Clock Input)	时钟输入到 BST 电路，一些操作发生在上升沿，而另一些发生在下降沿。
TRST	测试复位输入 (Test Reset Input)	低电平有效，异步复位边界扫描电路(在 IEEE 规范中，该引脚可选)。

JTAG Test

Figure Boundary Scanning Data Shifting



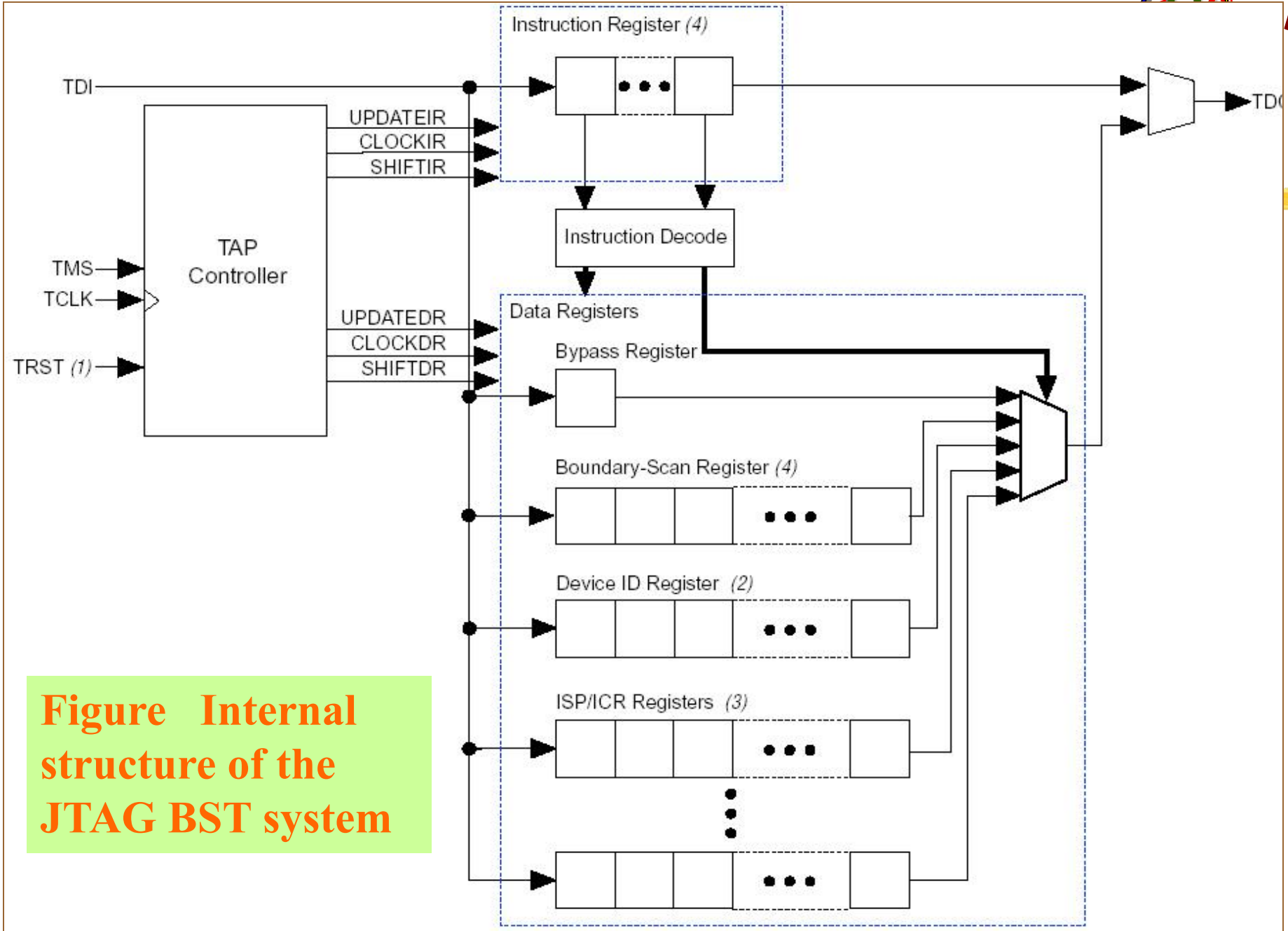


Figure Internal structure of the JTAG BST system

图 The relevant structure diagram of JTAG BST system and FLEX device

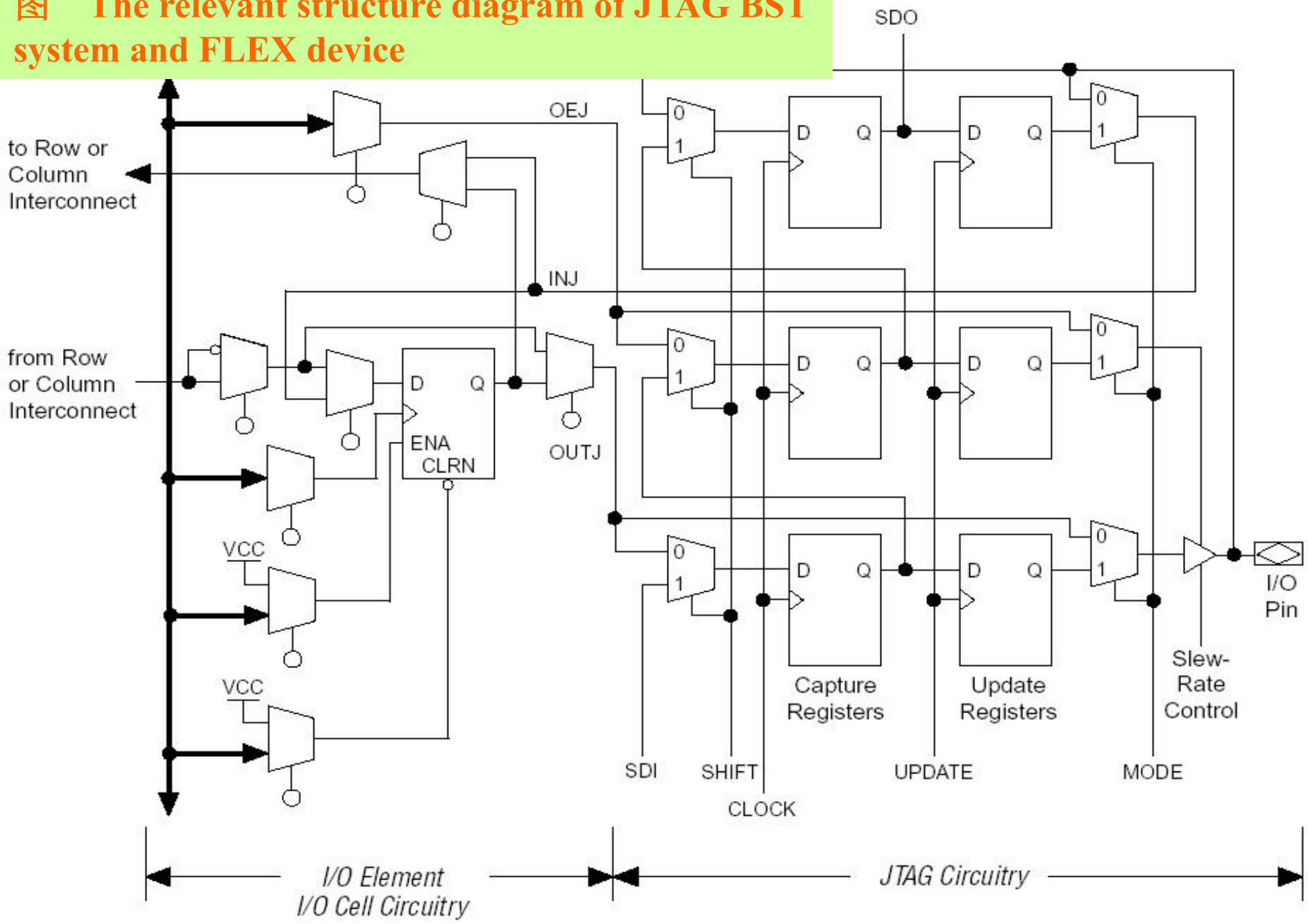
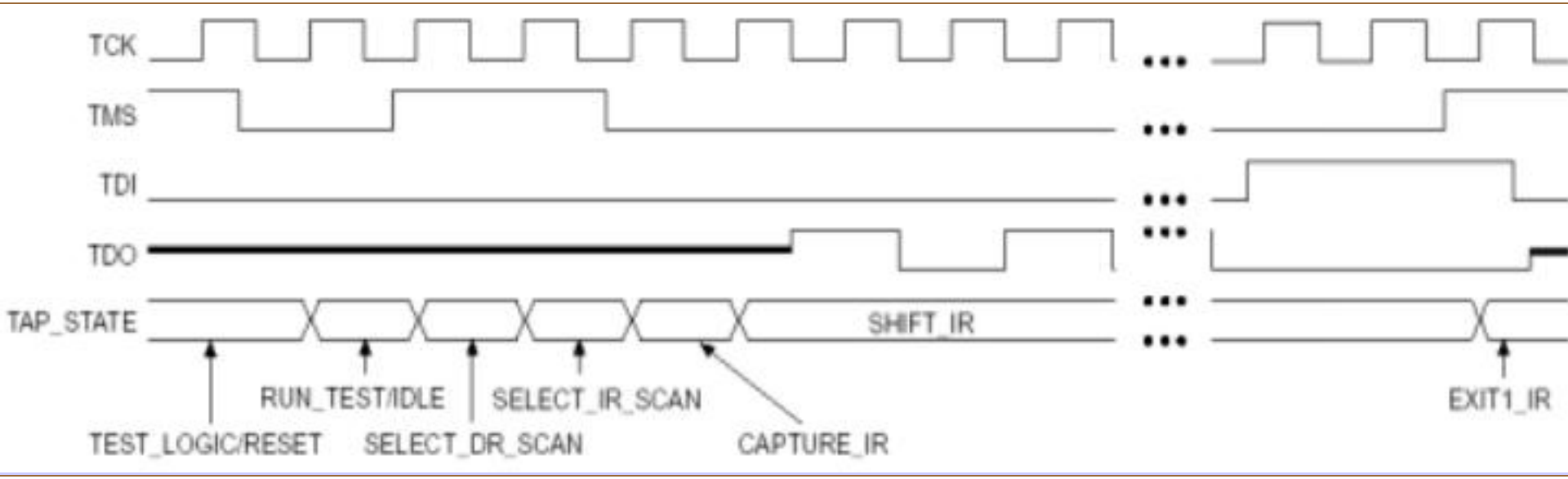


Figure: The Selection Command Mode Sequence of JTAG BST



The command mode of TAP controller is as follows: :

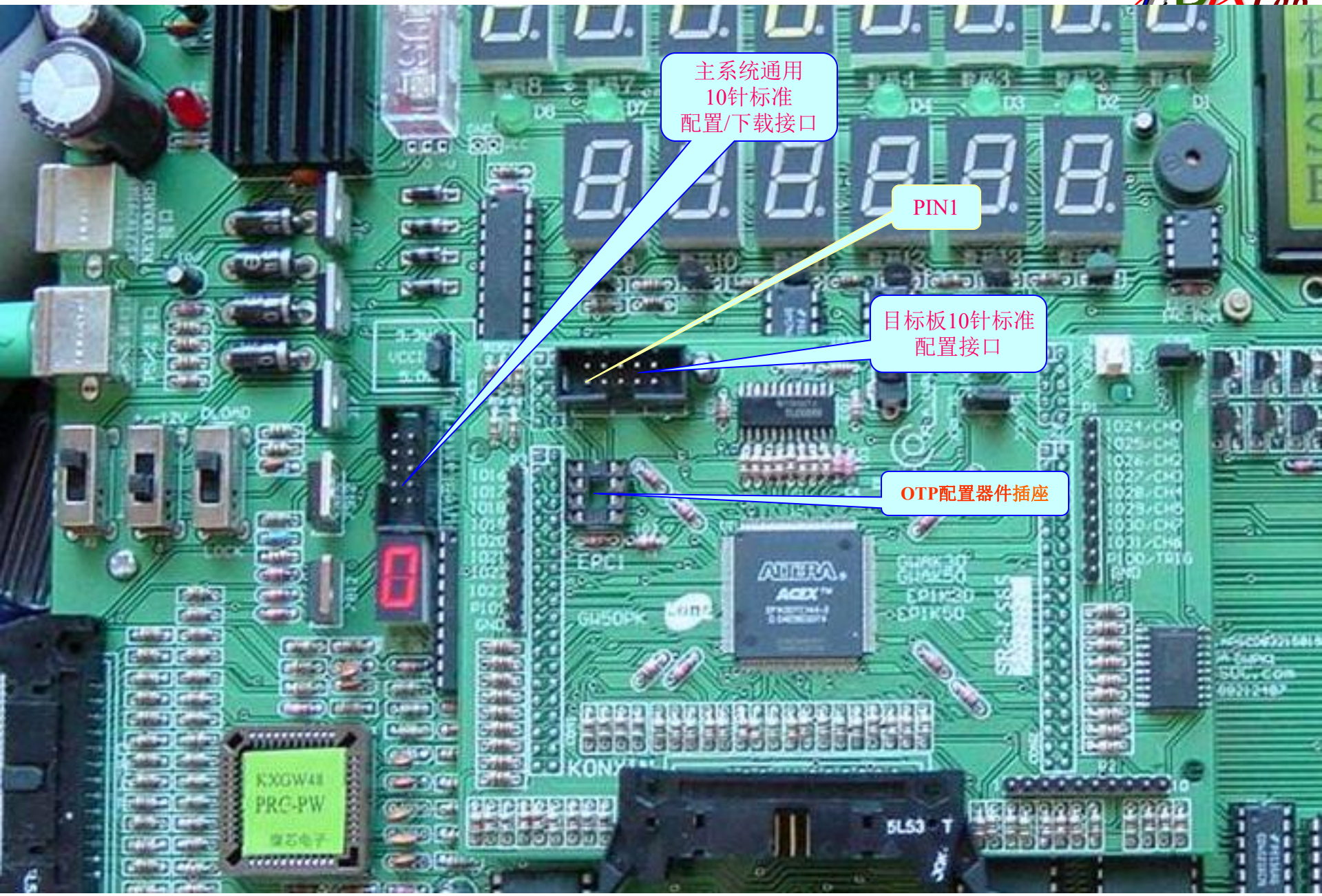
- **SAMPLE / PRELOAD instruction mode**
- **EXTEST instruction mode**
- **BYPASS instruction mode**
- **IDCODE instruction mode**
- **USERCODE instruction mode**

CPLD and FPGA Program and configure

10 Pin interfance

IO function

引脚	1	2	3	4	5	6	7	8	9	10
PS 模式	DCK	GND	CONF_DONE	VCC	nCONFIG	-	nSTATUS	-	DATA0	GND
JATG 模式	TCK	GND	TDO	VCC	TMS	-	-	-	TDI	GND


主系统通用
10针标准
配置/下载接口
PIN1
目标板10针标准
配置接口
OTP配置器件插座

CPLD/FPGA Program and configure

CPLD在系统编程

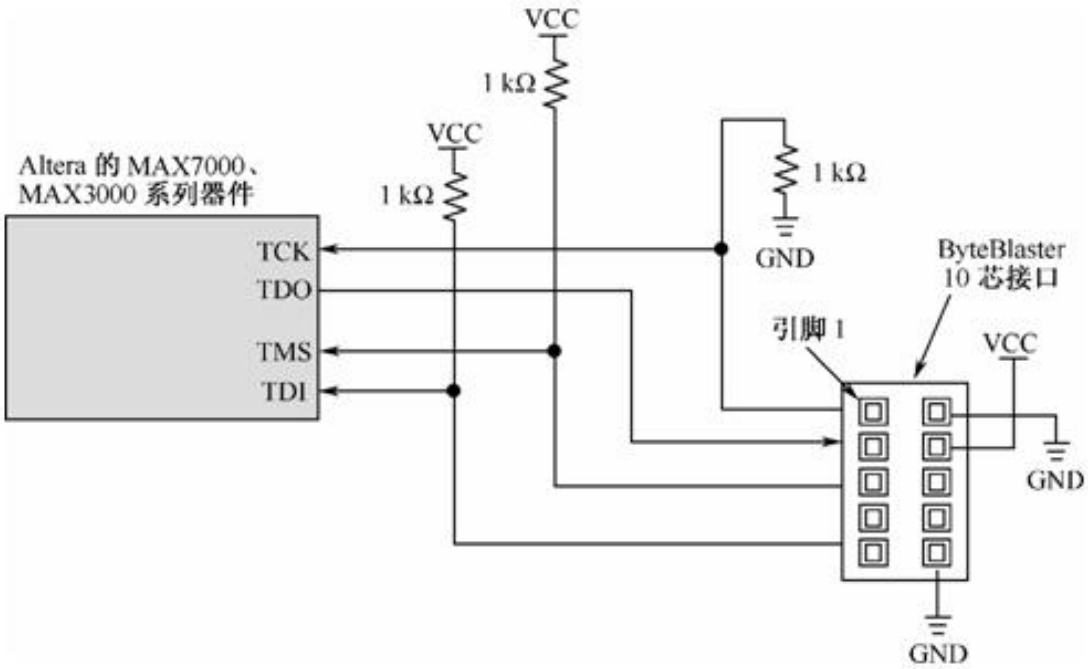


图 2-48　CPLD 编程下载连接图

CPLD/FPGA Program and configure

CPLD在系统编程

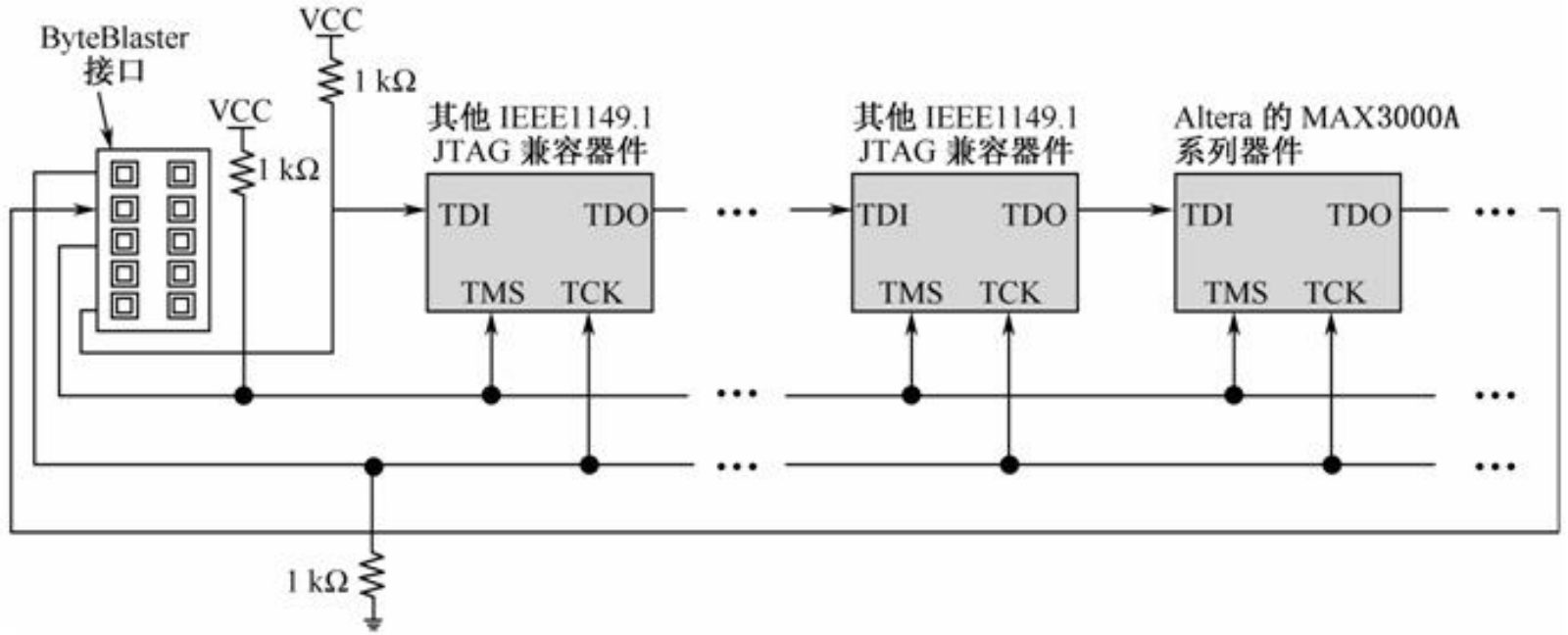


图 2-49　多 CPLD 芯片 ISP 编程连接方式

CPLD/FPGA Program and configure

FPGA配置方式

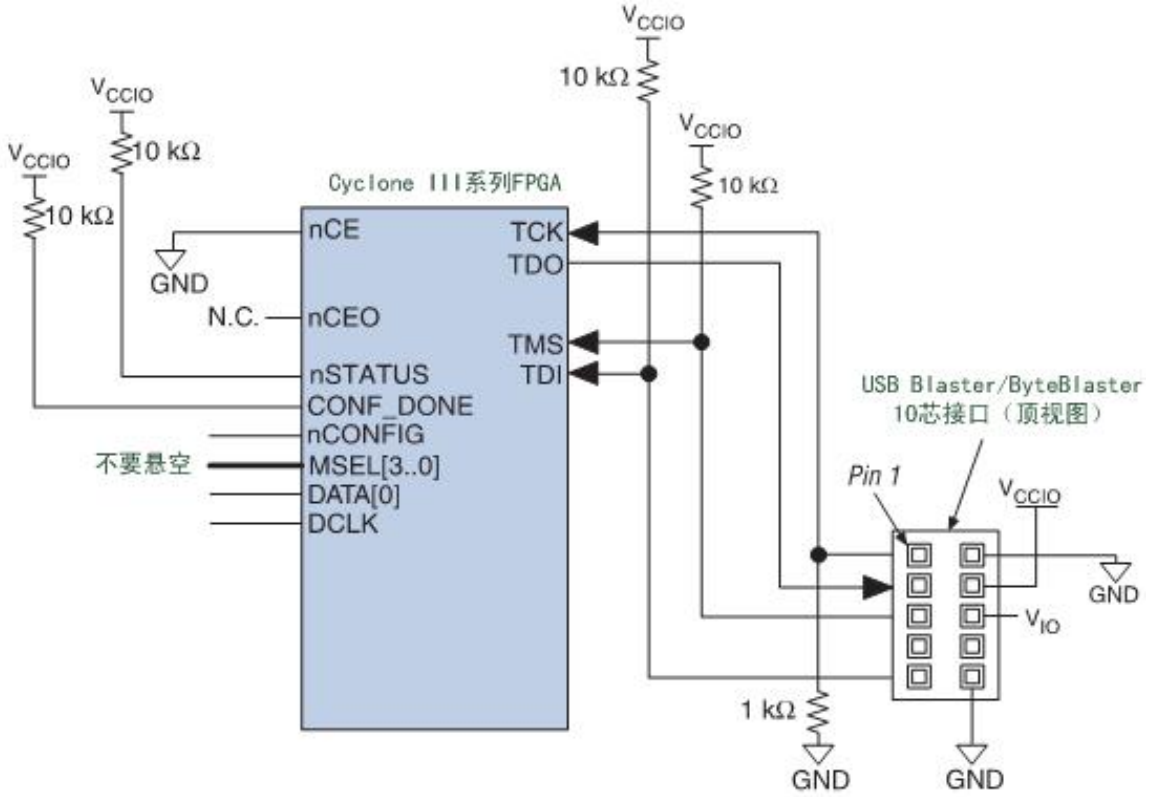


图 2-50　JTAG 在线配置 FPGA 的电路原理图

CPLD/FPGA Program and configure

FPGA专用配置器件

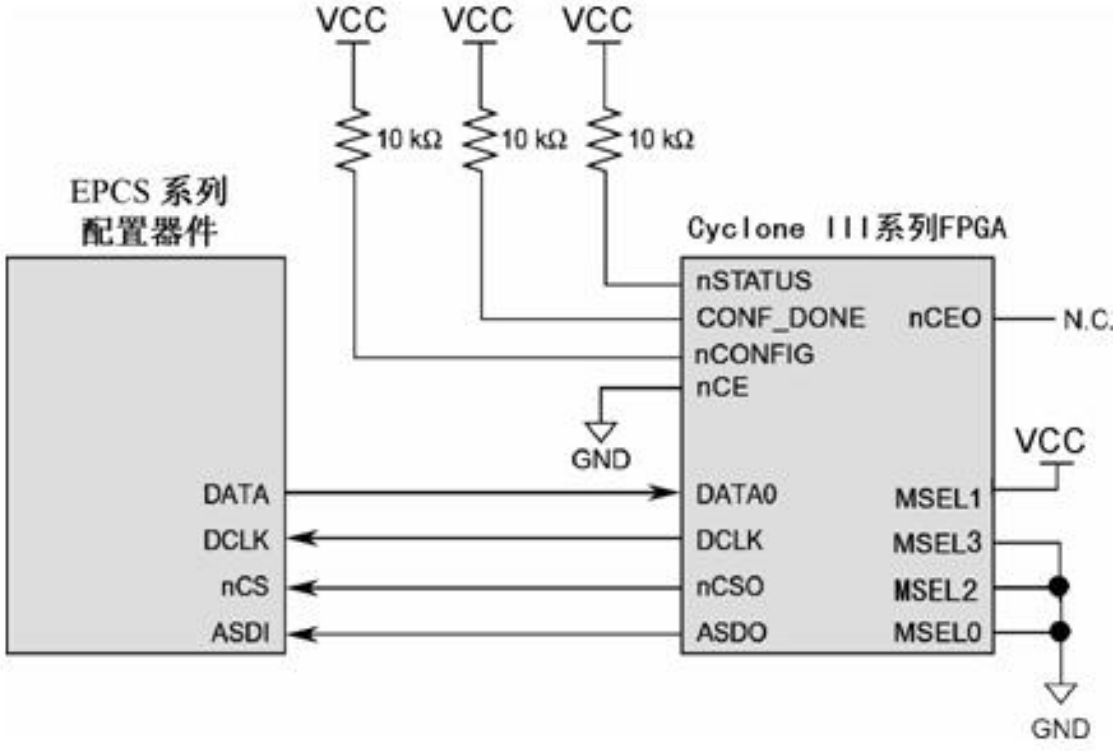


图 2-51　EPCS 器件配置 FPGA 的电路原理图

CPLD/FPGA Program and configure

使用单片机配置FPGA

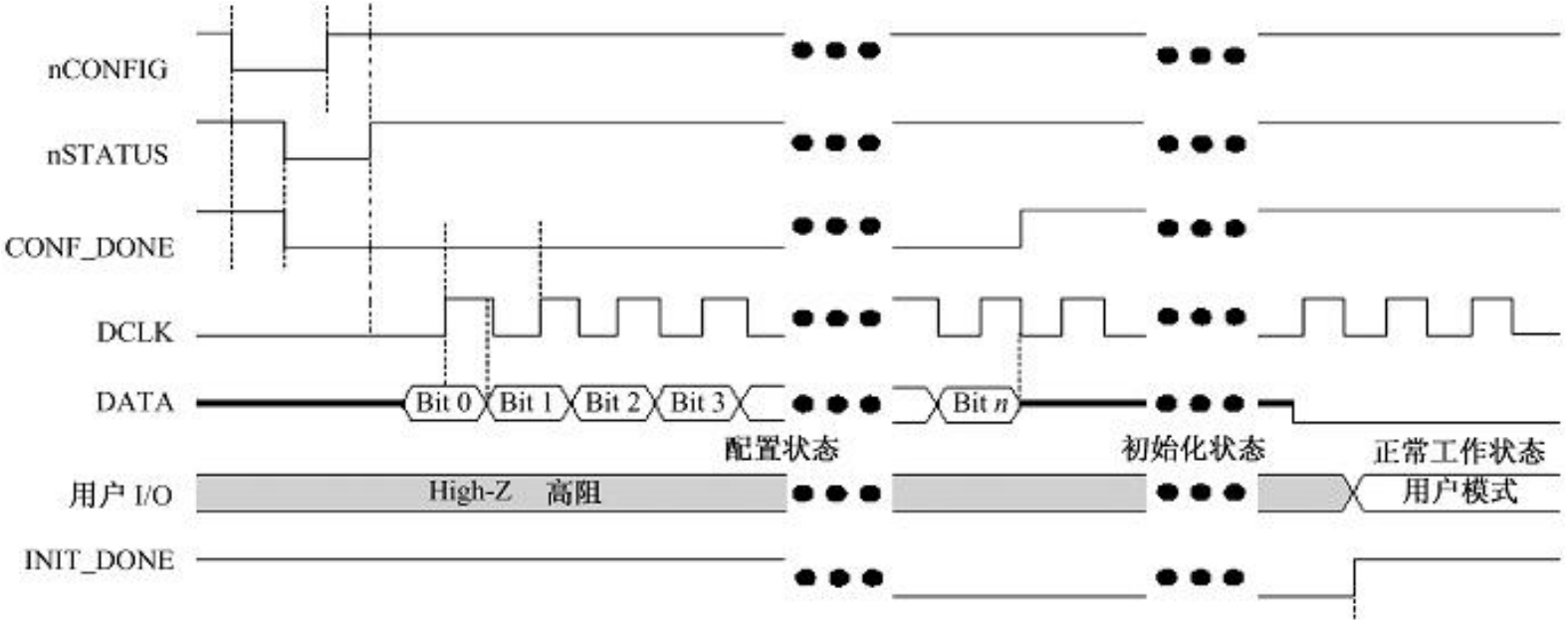


图 2-52 PS 模式的 FPGA 配置时序

CPLD/FPGA Program and configure

使用单片机配置FPGA

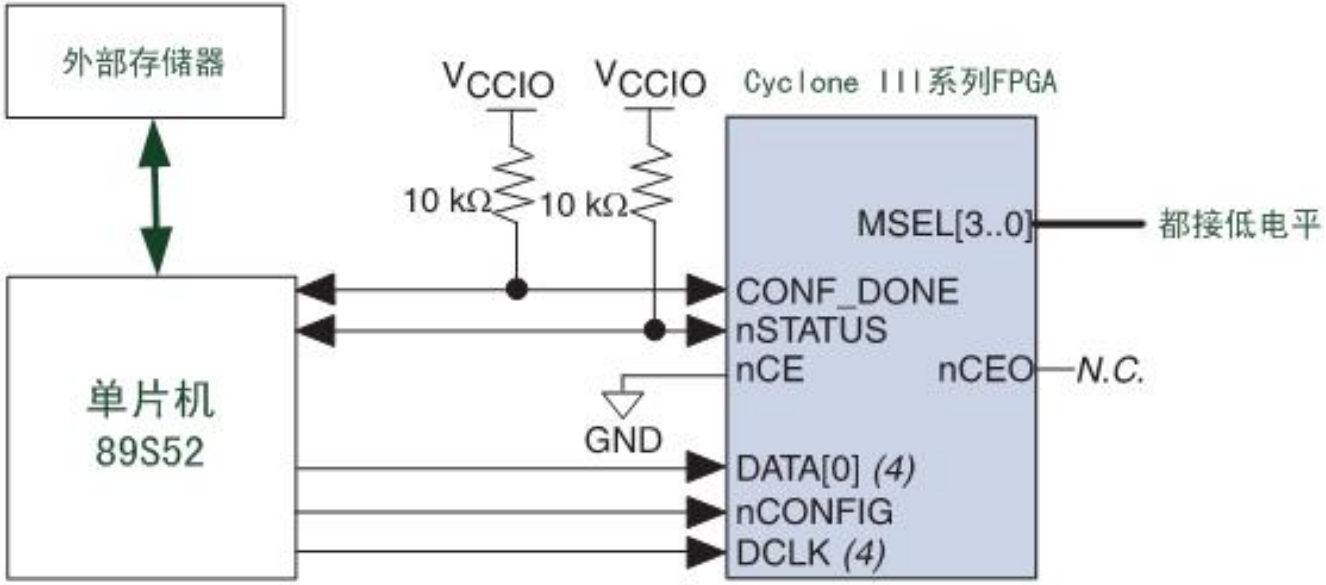


图 2-53　用 89C52 进行配置

CPLD/FPGA Program and configure

使用CPLD配置FPGA

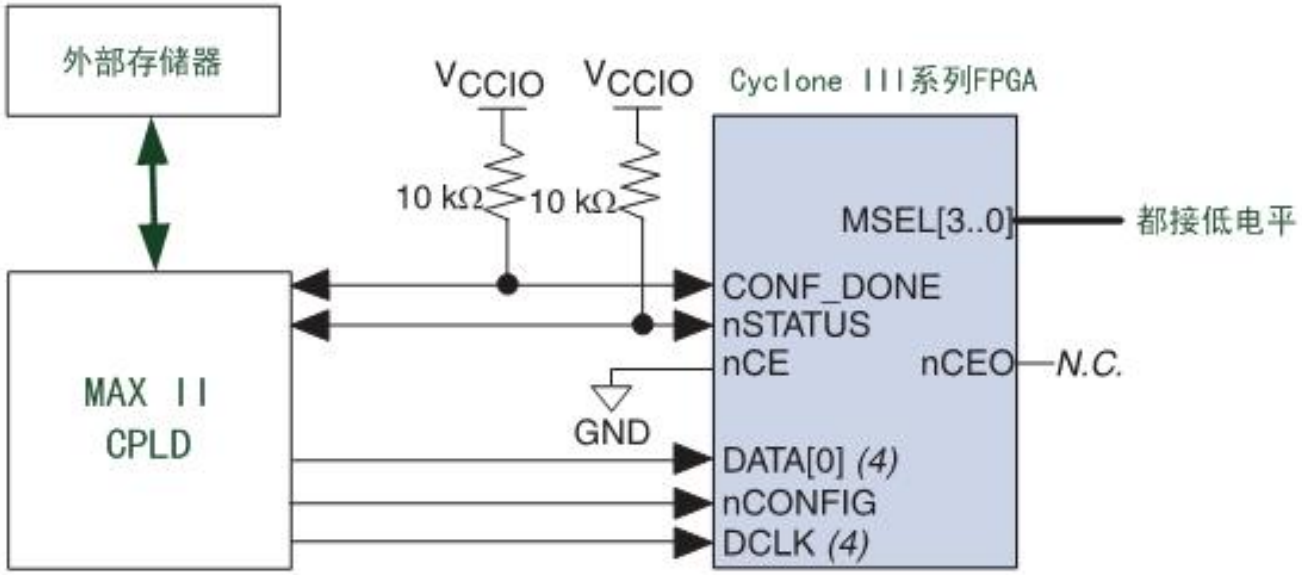
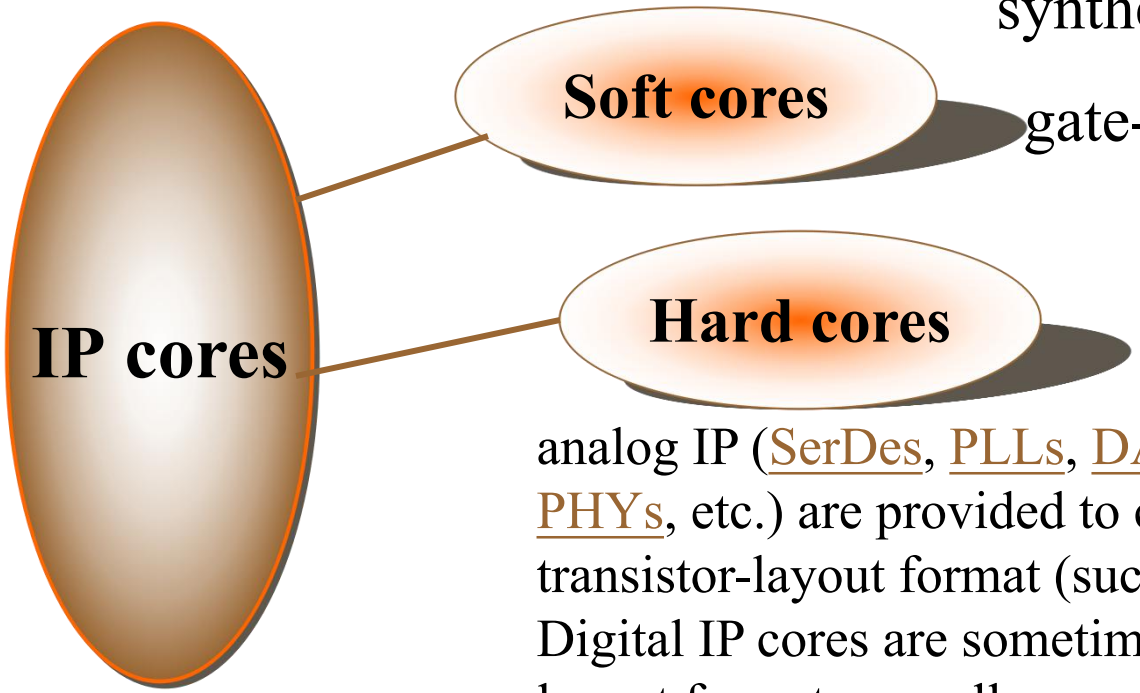


图 2-54　用 MAX II CPLD 进行配置

IP Core

IP(Intellectual Property)



synthesizable RTL

gate-level netlists

analog IP (SerDes, PLLs, DAC, ADC, PHYs, etc.) are provided to chip makers in transistor-layout format (such as GDSII). Digital IP cores are sometimes offered in layout format, as well.

IP Company

- ARM

FPGA/CPLD Development

采用更小线宽的工艺

JTAG在系统可编程

JTAG

加密功能。

短路保护功能。

DSP模块

多种内嵌Memory

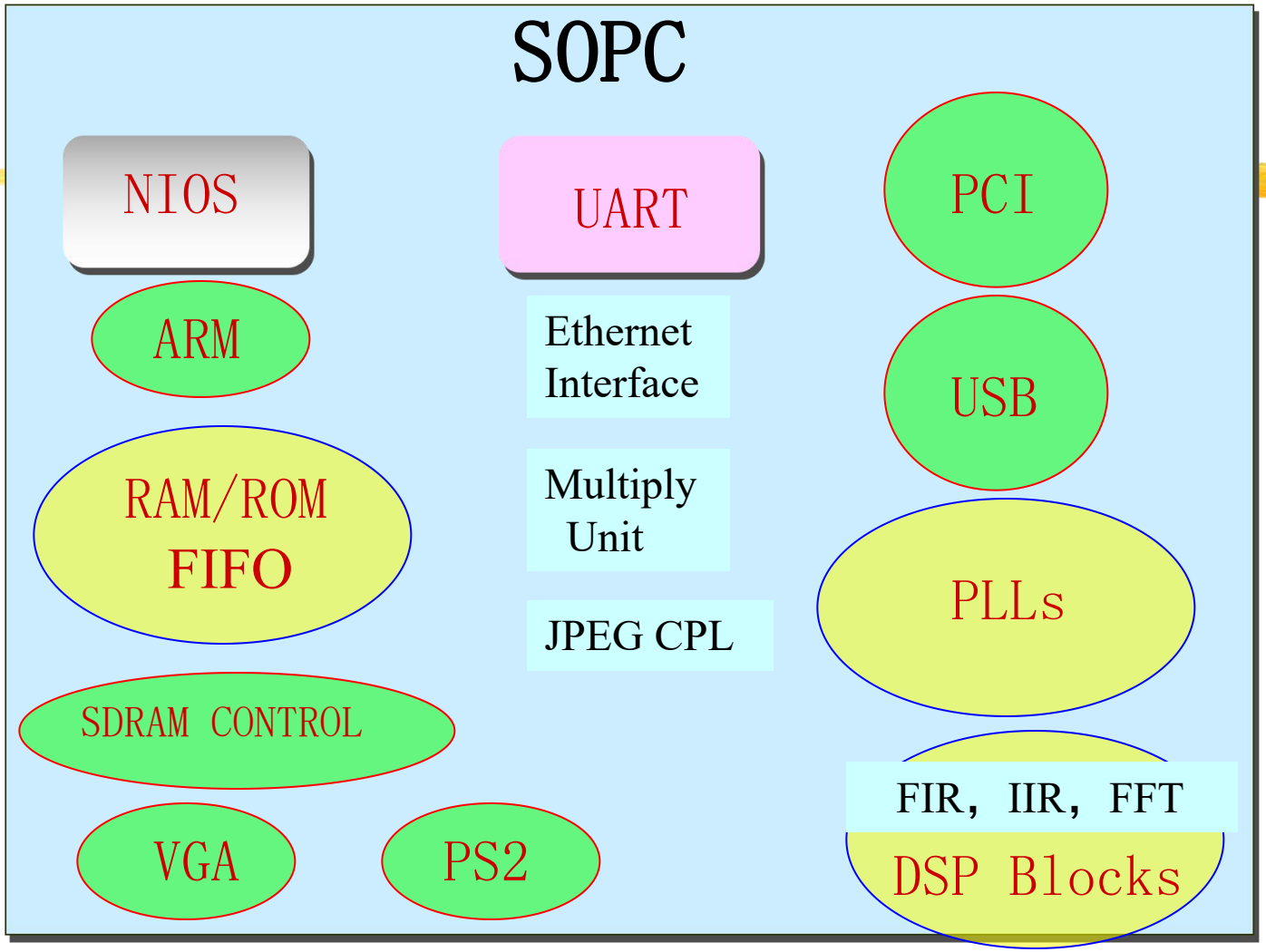
嵌入式逻辑分析

PLL模块

高速串行I/O

处理器核

……



SOC: SYSTEM ON A CHIP

SOPC: SYSTEM ON A PROGAMMABLE CHIP

Hard Copy

- Once an FPGA design is verified, validated and used successfully, there is an option to migrate it to structured ASIC
- This option is known as Hard Copy
- Using hard copy, FPGA design can be migrated to hard-wired design removing all configuration circuitry and programmability so that the target chip can be produced in high volume
- Hard copied chip uses 40% less power than FPGA and the internal delays are reduced

Presentation References

- The Three Ages of the FPGA https://www.youtube.com/watch?v=4ntXSyOhlBY
- S. Brown and J. Rose, "Architecture of FPGAs and CPLDs: A Tutorial", Department of Electrical and Computer Engineering, University of Toronto
- https://en.wikipedia.org/wiki/Verilog
- FPGAs; Lesson 1: Concept Guide and Step by Step Tutorial to Flash https://www.youtube.com/watch?v=pDE2qenDXKQ