مبحث هفتم

حافظه و منطق برنامه‌پذیر

Memory and Programmable Logic

Presented by Ali Maleki Spring Semester, 2021

تعریف حافظه و منطق برنامه‌پذیر

✓ حافظه (Memory)

مجموعه‌ای از سلول ها که قادر است اطلاعات دودویی را در خود ذخیره نماید.

✓ منطق برنامه‌پذیر (Programmable Logic)

قطعه ای با امکان آرایش دادن سخت افزار بر اساس بیت های ذخیره شده

فهرست مطالب

- مقدمه‌ای بر حافظه‌ها
- حافظه‌ی RAM
- تشخیص و تصحیح خطا
- حافظه‌ی ROM
- منطق برنامه‌پذیر
- PROM، PLA و PAL
- CPLD و FPGA

حافظه

✓ انواع حافظه

حافظه با دسترسی تصادفی (Random Access Memory = RAM)

حافظه‌ی فقط خواندنی (Read Only Memory = ROM)

چند اصطلاح در رابطه با حافظه

- **نوشتن در حافظه (Memory Write):**

  ذخیره کردن اطلاعات در حافظه

- **خواندن از حافظه (Memory Read):**

  انتقال اطلاعات ذخیره شده در حافظه به بیرون

- **کلمه (Word):**

  گروهی از بیت ها

- **بایت (Byte):**

  یک گروه ۸ بیتی

- **نیبل (Nible):**

  یک گروه ۴ بیتی

چند اصطلاح در رابطه با حافظه

ضرایب مربوط به تعیین ظرفیت حافظه:

کیلو ($k=2^{10}$) ، مِگا ($M=2^{20}$)، گیگا ($G=2^{30}$) ، تِرا ($G=2^{40}$)

حافظه‌ی فرار (Volatile Memory):

حافظه‌ای که با قطع شدن تغذیه، اطلاعات آن از دست خواهد رفت.

حافظه‌ی غیرفرار (Nonvolatile Memory):

حافظه‌ای که با قطع شدن تغذیه، اطلاعات آن حفظ می‌گردد.

فهرست مطالب

- ☑ مقدمه‌ای بر حافظه‌ها
- ☐ حافظه‌ی RAM
- ☐ تشخیص و تصحیح خطا
- ☐ حافظه‌ی ROM
- ☐ منطق برنامه‌پذیر
- ☐ PROM، PLA و PAL
- ☐ CPLD و FPGA

حافظه با دسترسی تصادفی (RAM)

✓علت نامگذاری RAM:

در هر زمان، به هر مکان حافظه می‌توان دسترسی داشت.

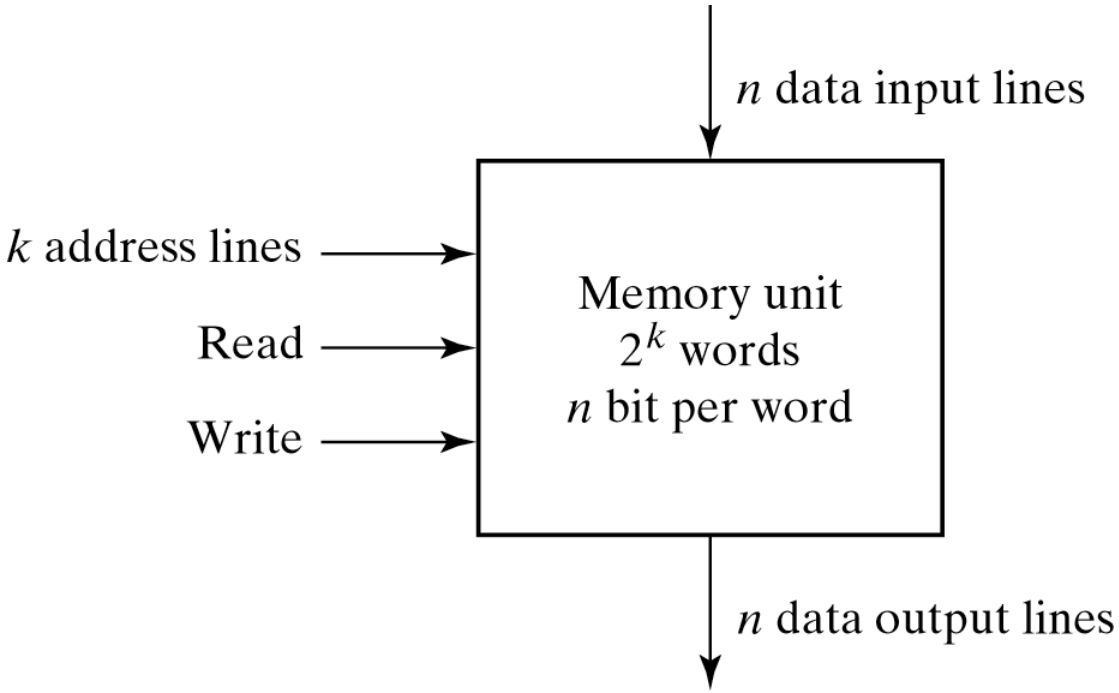
زمان لازم برای انتقال اطلاعات به/از هر مکان حافظه یکسان است.

RAM در قیاس با حافظه با دسترسی ترتیبی (sequential access memory) مطرح می گردد.

به عنوان نمونه‌ای از حافظه با دسترسی ترتیبی می‌توان به حافظه از نوع نوار مغناطیسی اشاره نمود.

RAM یک حافظه‌ی فرار است.

نمودار بلوکی حافظه‌ی RAM



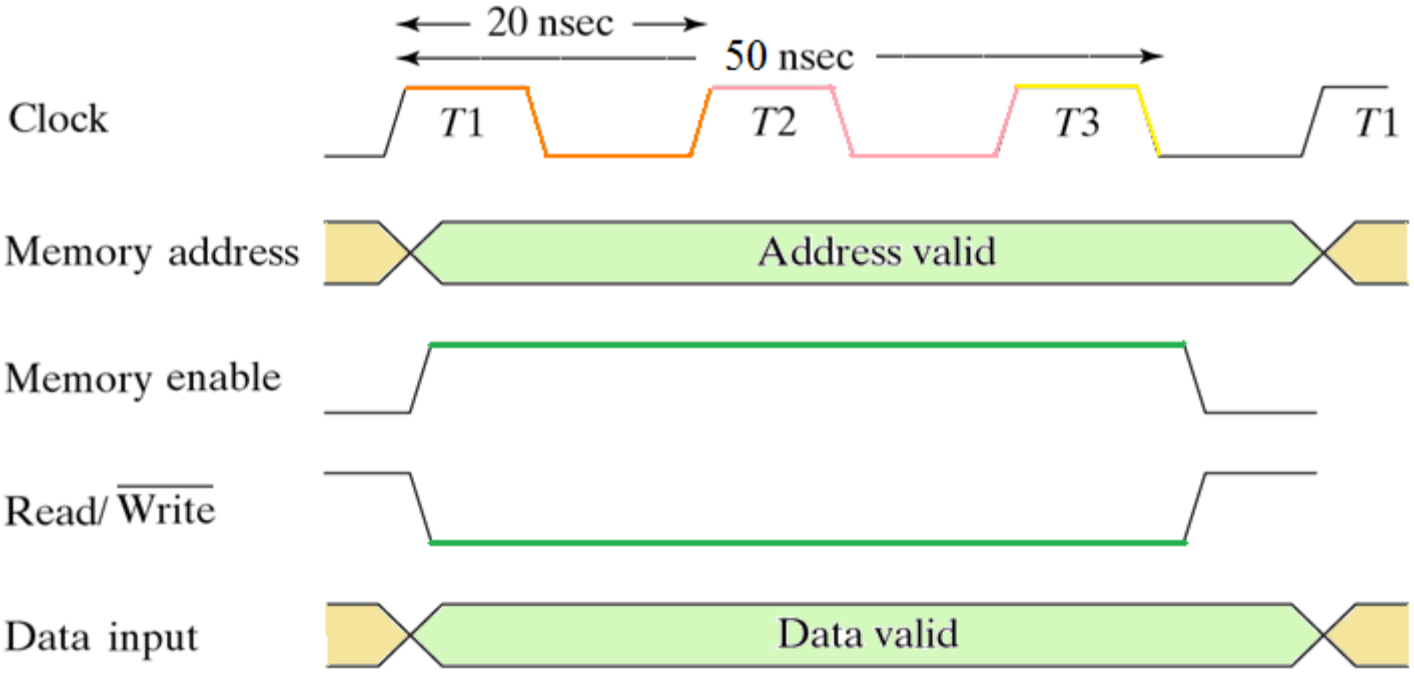
✓محتوای یک حافظه ی 16 * 1024

Memory address		
Binary	decimal	Memory contest
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

نمودار زمانی سیکل نوشتن حافظه

زمان سیکل (**cycle time**) : زمان لازم برای تکمیل نوشتن در یک حافظه

در این مثال، زمان سیکل را **۵۰ نانوثانیه** فرض کنید.

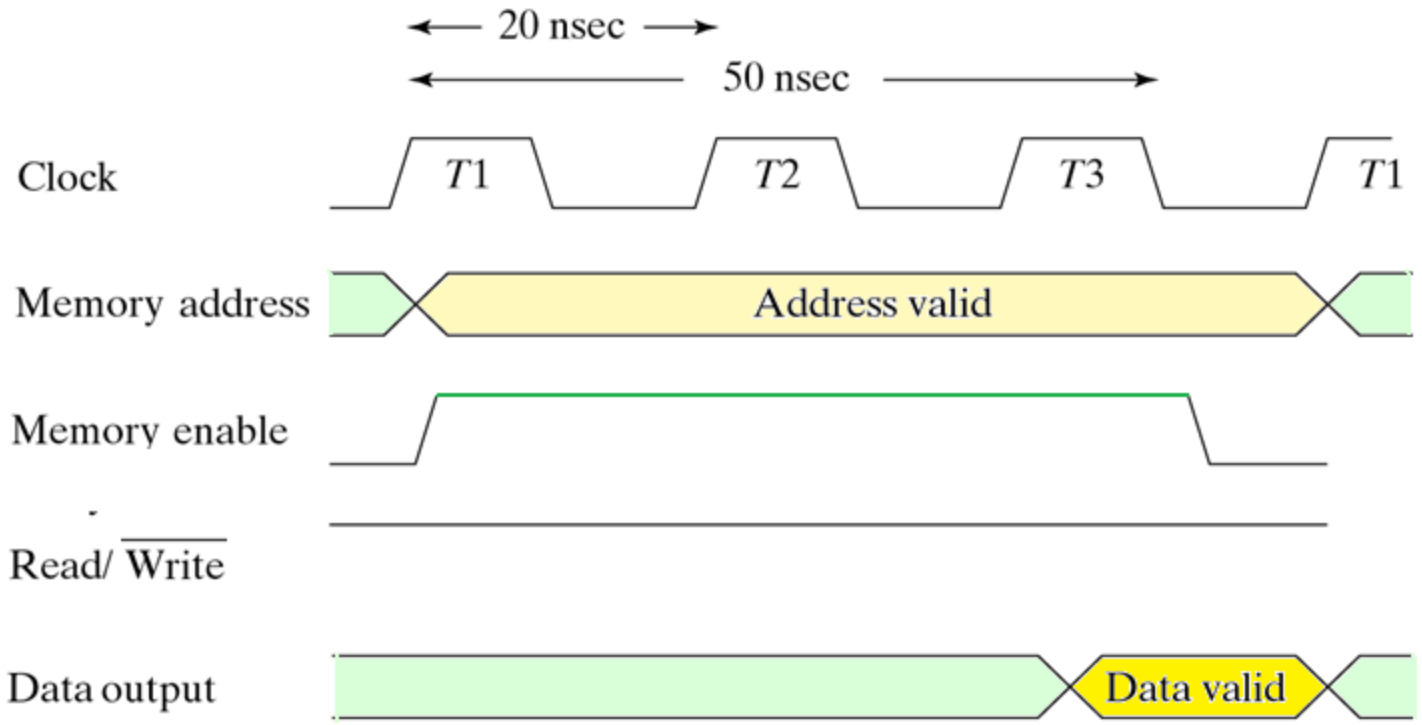


(a) Write cycle

نمودار زمانی سیکل خواندن حافظه

زمان دستیابی (access time) : زمان لازم برای انتخاب یک کلمه و خواندن آن از حافظه.

در این مثال، زمان دستیابی را ۵۰ نانوثانیه فرض کنید.



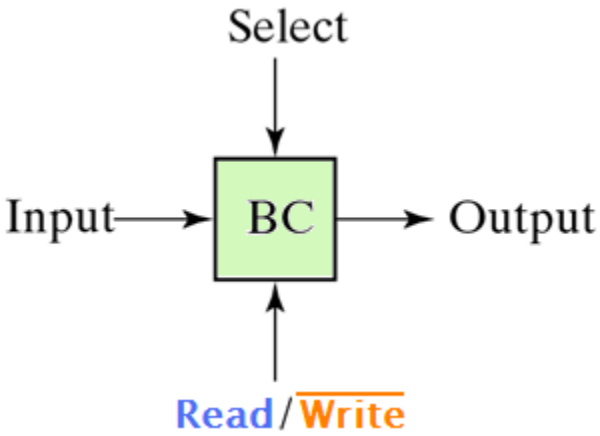
(b) Read cycle

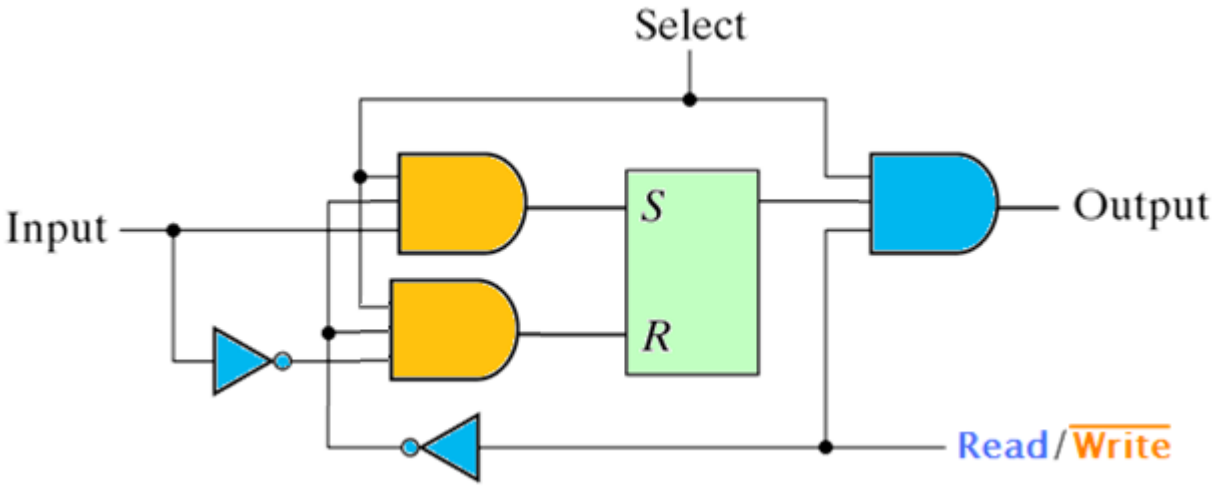
✓انواع RAM:

SRAM یا RAM استاتیک (Static RAM)

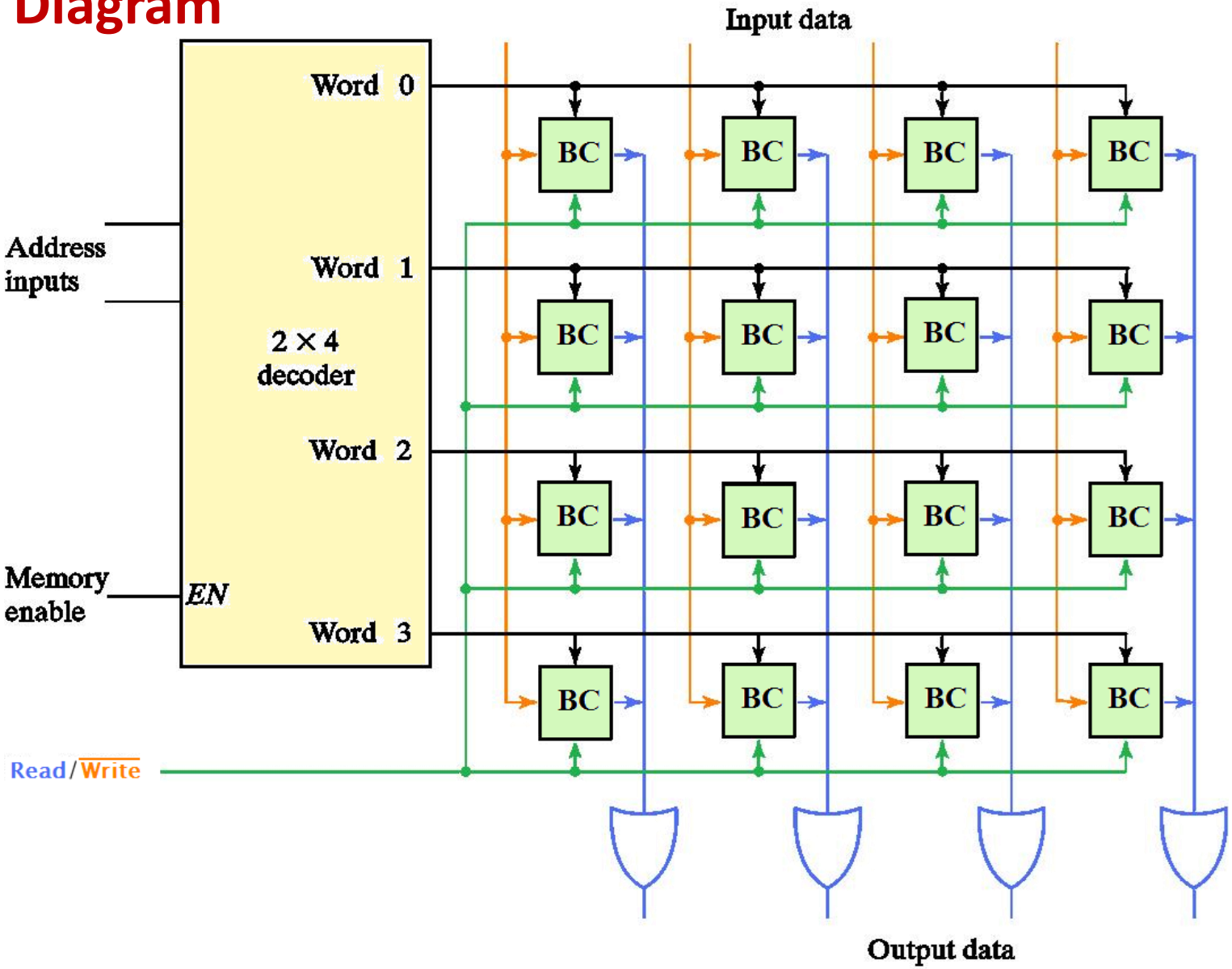
DRAM یا RAM دینامیک (Dynamic RAM)

سلول ذخیره‌سازی باینری (Binary Cell)

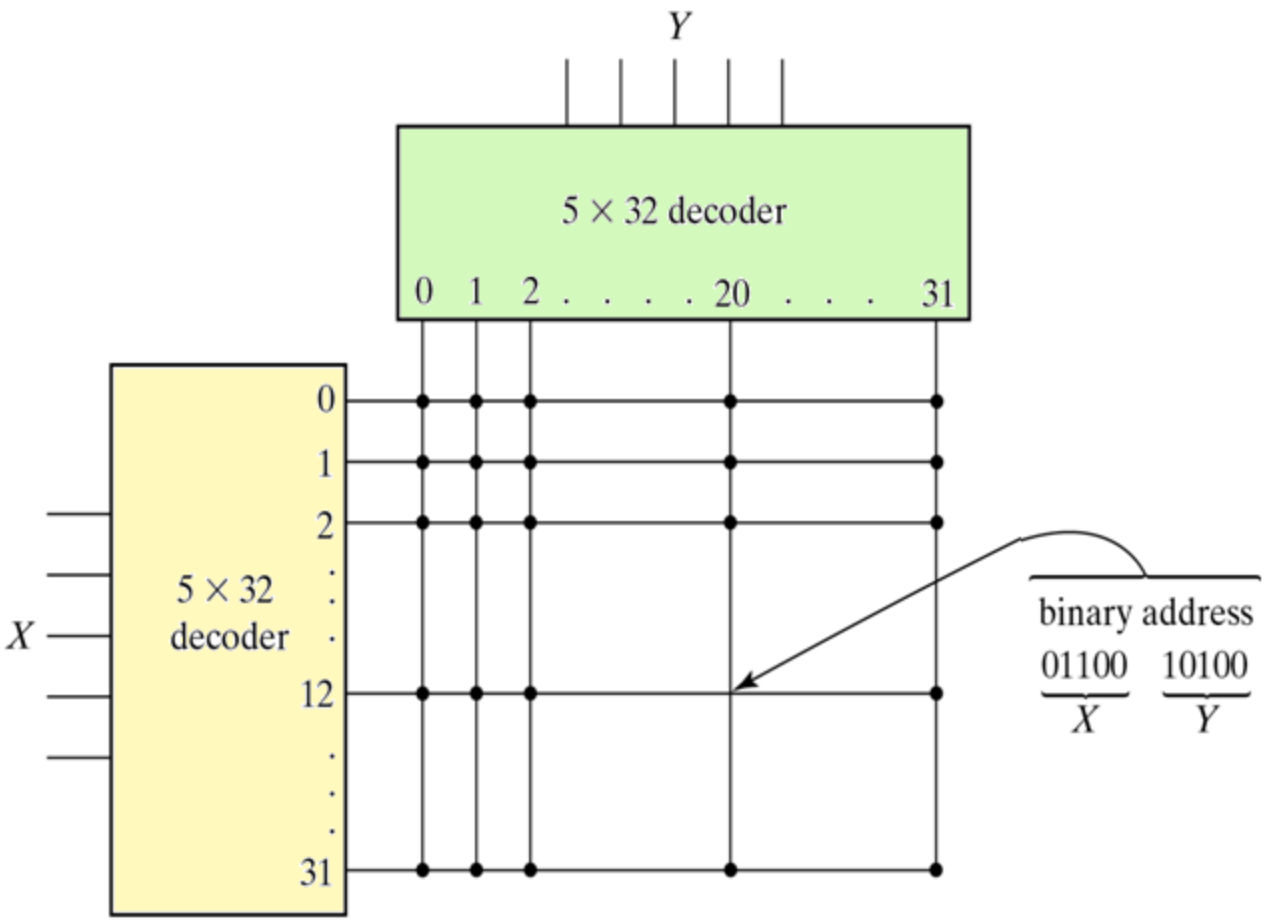




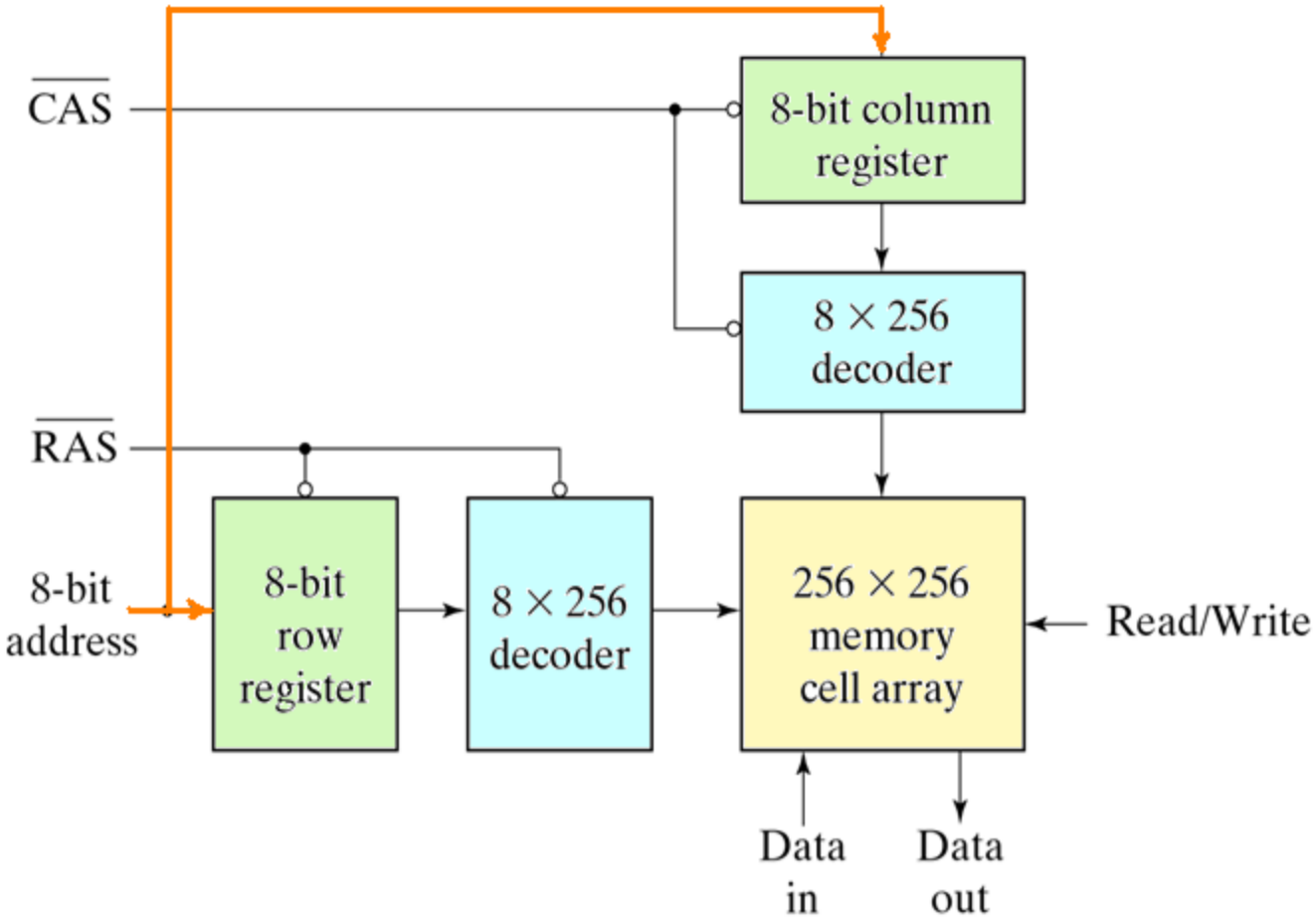
4*4 RAM Diagram



✓ دیکُد کردن متقارن (Coincident Decoding)



✓مالتی‌پلکس کردن آدرس (Address Multiplexing)



فهرست مطالب

- ☑ مقدمه ای بر حافظه ها
- ☑ حافظه ی RAM
- ☐ تشخیص و تصحیح خطا
- ☐ حافظه ی ROM
- ☐ منطق برنامه پذیر
- ☐ PROM، PLA و PAL
- ☐ CPLD و FPGA

تشخیص و تصحیح خطا (Error Detection and Correction)

- ✓ **تشخیص خطا (Error Detection)**
  - بیت توازن (Parity Bit)
- ✓ **تصحیح خطا (Error Correction)**
  - کد همینگ (Hamming Code)
- ✓ **تصحیح خطای یگانه، تشخیص خطای دوگانه**

(Single-Error Correction, Double-Error Detection)

روش ترکیبی کد هَمینگ و بیت توازن

کد هَمینگ (Hamming Code)

کد هشت بیتی 11000100 را در نظر بگیرید. در کد هَمینگ، موقعیت‌های توان‌های ۲ با بیت توازن و بقیه موقعیت‌ها با بیت‌های داده پر می‌شوند.

Bit position	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

P_1 = XOR of bits(3,5,7,9,11)=0

P_2 = XOR of bits(3,6,7,10,11)=0

P_4 = XOR of bits(5,6,7,12)=1

P_8 = XOR of bits(9,10,11,12)=1

In memory, 0 0 1 1 1 0 0 1 0 1 0 0

تشخیص و تصحیح خطای یگانه با استفاده از کد هَمینگ:

In memory, 0 0 1 1 1 0 0 1 0 1 0 0

Check Bits:

C_1=XOR of bits (1,3,5,7,9,11)

C_2=XOR of bits (2,3,6,7,10,11)

C_4=XOR of bits (4,5,6,7,12)

C_8=XOR of bits (8,9,10,11,12)

$C = C_8C_4C_2C_1$

C = 0 : No Error

$C \neq 0$: Check bit gives error position, toggle bit to correct

مثال۱ : با فرض استفاده از کد هَمینگ، بروز خطا در کد دریافتی از حافظه را بررسی نموده و در صورت نیاز آن را تصحیح کنید.

In memory, 0 0 1 1 1 0 0 1 0 1 0 0

Check Bits:

0 0 1 1 1 0 0 1 0 1 0 0 C_1=XOR of bits (1,3,5,7,9,11) = 0

0 0 1 1 1 0 0 1 0 1 0 0 C_2=XOR of bits (2,3,6,7,10,11) = 0

0 0 1 1 1 0 0 1 0 1 0 0 C_4=XOR of bits (4,5,6,7,12) = 0

0 0 1 1 1 0 0 1 0 1 0 0 C_8=XOR of bits (8,9,10,11,12) = 0

$C = C_8C_4C_2C_1 \Rightarrow C = 0000$

بنابراین خطایی رخ نداده است.

مثال۲ : با فرض استفاده از کد هَمینگ، بروز خطا در کد دریافتی از حافظه را بررسی نموده و در صورت نیاز آن را تصحیح کنید.

In memory, 1 0 1 1 1 0 0 1 0 1 0 0

Check Bits:

1 0 1 1 1 0 0 1 0 1 0 0 C_1=XOR of bits (1,3,5,7,9,11) = 1

1 0 1 1 1 0 0 1 0 1 0 0 C_2=XOR of bits (2,3,6,7,10,11) = 0

1 0 1 1 1 0 0 1 0 1 0 0 C_4=XOR of bits (4,5,6,7,12) = 0

1 0 1 1 1 0 0 1 0 1 0 0 C_8=XOR of bits (8,9,10,11,12) = 0

$C = C_8C_4C_2C_1$ ⇨ C = 0001

خطایی در بیت اول رخ داده است.

مثال۳ : با فرض استفاده از کد هَمینگ، بروز خطا در کد دریافتی از حافظه را بررسی نموده و در صورت نیاز آن را تصحیح کنید.

In memory, 0 0 1 1 0 0 0 1 0 1 0 0

Check Bits:

0 0 1 1 0 0 0 1 0 1 0 0 C_1=XOR of bits (1,3,5,7,9,11) = 1

0 0 1 1 0 0 0 1 0 1 0 0 C_2=XOR of bits (2,3,6,7,10,11) = 0

0 0 1 1 0 0 0 1 0 1 0 0 C_4=XOR of bits (4,5,6,7,12) = 1

0 0 1 1 0 0 0 1 0 1 0 0 C_8=XOR of bits (8,9,10,11,12) = 0

C = 0101

خطایی در بیت پنجم رخ داده است.

تصحیح خطای یگانه، تشخیص خطای دوگانه

کد هَمینگ می‌تواند **خطای یگانه** را تشخیص داده و تصحیح نماید ولی نمی‌تواند چند خطا را تشخیص دهد. با اضافه کردن یک بیت توازن به کد همینگ می‌توان بروز دو خطا را تشخیص داد.

به عنوان مثال برای کد هشت بیتی 11000100

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0	P_{13}

اگر C=0 و P=0 باشد خطا رخ نداده است.

اگر C≠0 و P=1 باشد یک خطا رخ داده است که قابل تصحیح است.

اگر C≠0 و P=0 باشد دو خطا رخ داده است که قابل تشخیص است ولی قابل تصحیح نیست.

اگر C=0 و P=1 باشد یک خطا در محل بیت P_{13} رخ داده است.

فهرست مطالب

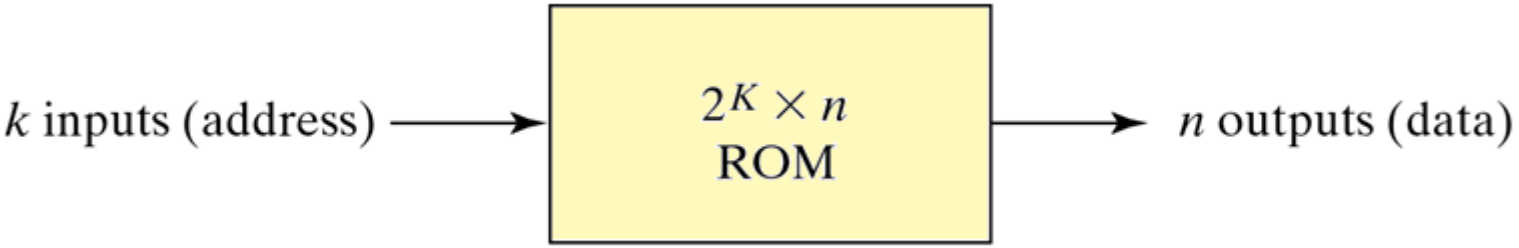
- ☑ مقدمه ای بر حافظه ها
- ☑ حافظه ی RAM
- ☑ تشخیص و تصحیح خطا
- ☐ حافظه ی ROM
- ☐ منطق برنامه پذیر
- ☐ PROM، PLA و PAL
- ☐ CPLD و FPGA

حافظه‌ی فقط خواندنی (ROM)

✓علت نامگذاری ROM:

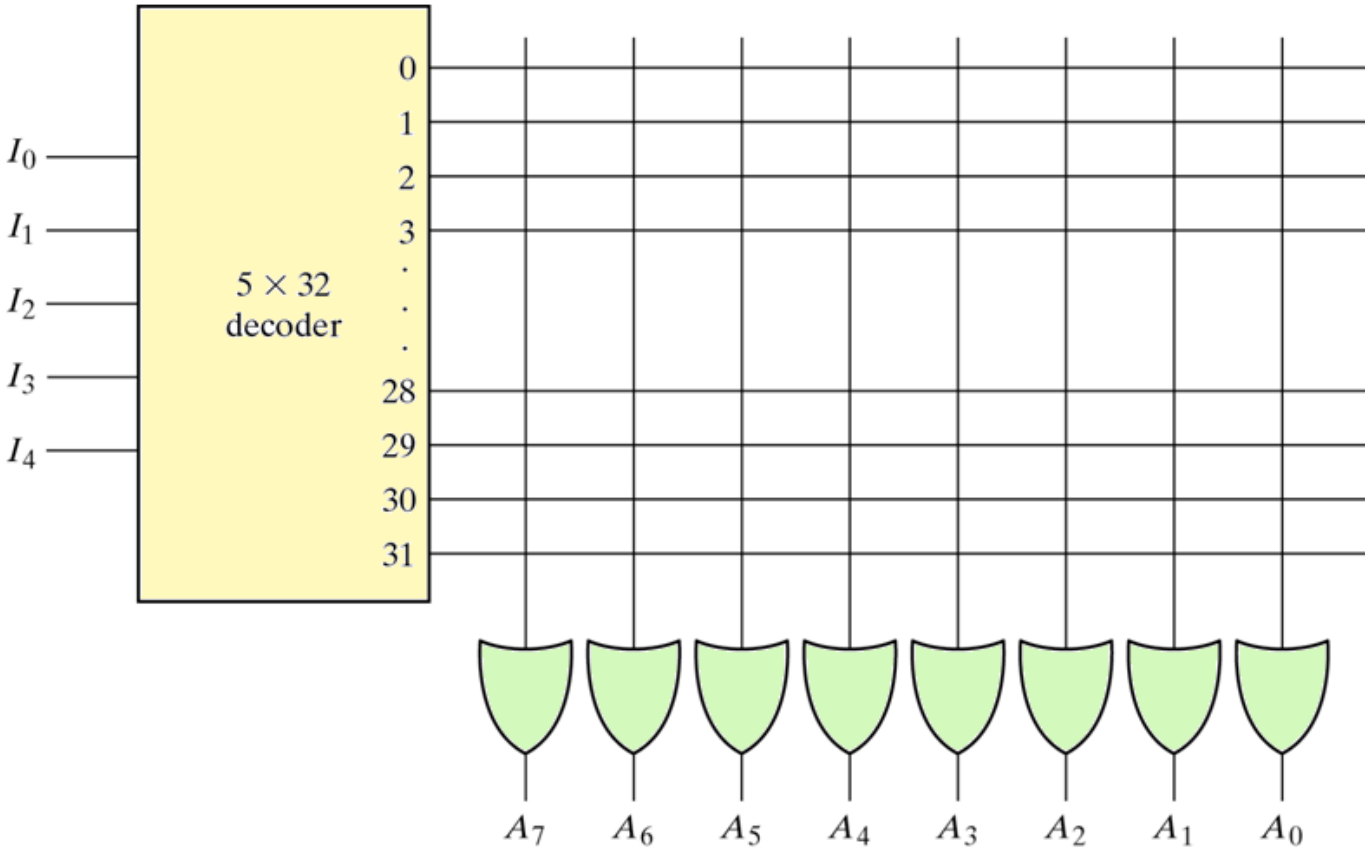
فقط از این حافظه خوانده می‌شود.

ROM یک حافظه‌ی غیرفرار است.



ROM

منطق داخلی یک ROM ۸×۳۲ :

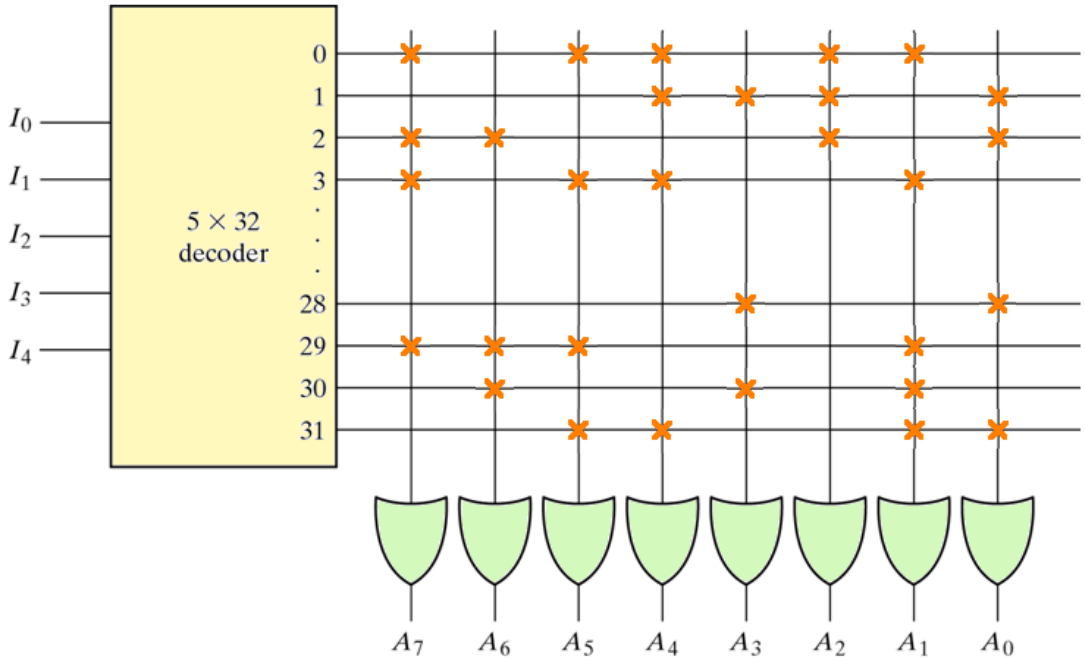


ROM

برنامه ریزی ROM

ROM Truth Table (Partial)

Inputs					Outputs							
I4	I3	I2	I1	I0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮					⋮					
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1



پیاده سازی مدار ترکیبی

ROM از یک دیکدر و تعدادی گیت OR تشکیل شده است.

با استفاده از دیکدر می‌توان 2^k مینترم برای k متغیر فراهم نمود.

بنابراین می‌توان با ROM ، تابع را به صورت جمع مینترم‌ها پیاده‌سازی نمود.

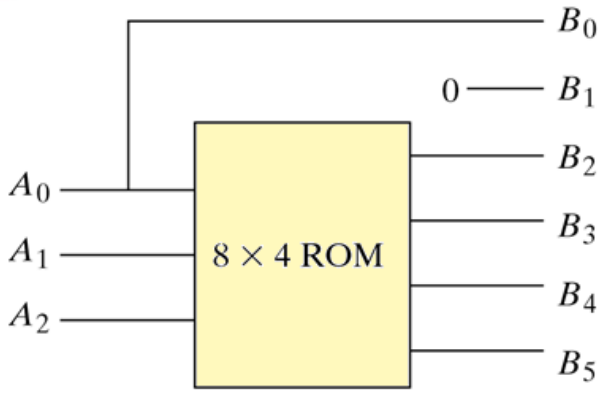
پیاده سازی مدار ترکیبی

مداری با جدول درستی توصیف شده است. چگونه می‌توان مدار را با ROM پیاده‌سازی نمود؟

Inputs			Outputs						Decimal
A_1	A_1	A_0	B_5	B_4	B_3	B_2	B_1	B_0	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49

A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table



(a) Block diagram

ROM

انواع ROM:

ROM	(Read Only Memory)
PROM	(Programmable ROM)
EPROM	(Erasable Programmable ROM)
EEPROM or E^2PROM	(Electrically Erasable Programmable ROM)

PROM نوعی قطعه‌ی منطقی
قابل برنامه‌ریزی (PLD) است.

فهرست مطالب

- ☑ مقدمه ای بر حافظه ها
- ☑ حافظه ی RAM
- ☑ تشخیص و تصحیح خطا
- ☑ حافظه ی ROM
- ☐ منطق برنامه پذیر
- ☐ PROM، PLA و PAL
- ☐ CPLD و FPGA

منطق برنامه‌پذیر (Programmable Logic)

Combinatorial PLD:

- PROM
- PAL
- PLA

Sequential PLD

- SPLD
- CPLD
- FPGA

برنامه‌ریزی (Programming):

روالی که اطلاعات بیت‌ها را وارد سخت‌افزار می‌کند.

منطق برنامه‌پذیر (Programmable Logic)

PROM: • Programmable Read Only Memory

PLA: • Programmable Logic Array

PAL: • Programmable Array Logic

SPLD: • Simple Programmable Logic Device

CPLD: • Complex Programmable Logic Device

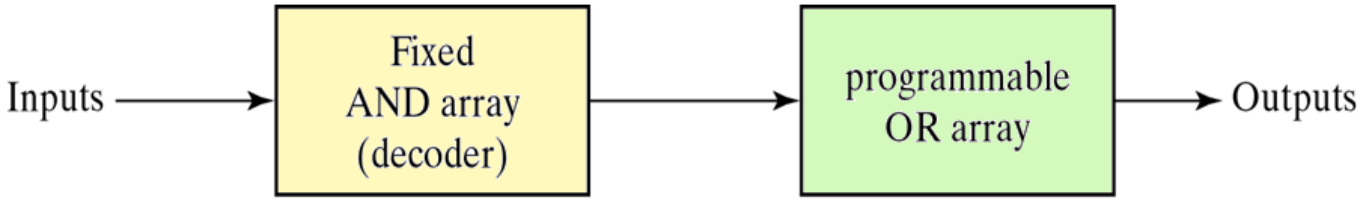
FPGA: • Field Programmable Gate Array

فهرست مطالب

- ☑ مقدمه ای بر حافظه ها
- ☑ حافظه ی RAM
- ☑ تشخیص و تصحیح خطا
- ☑ حافظه ی ROM
- ☑ منطق برنامه پذیر
- ☐ PROM، PLA و PAL
- ☐ CPLD و FPGA

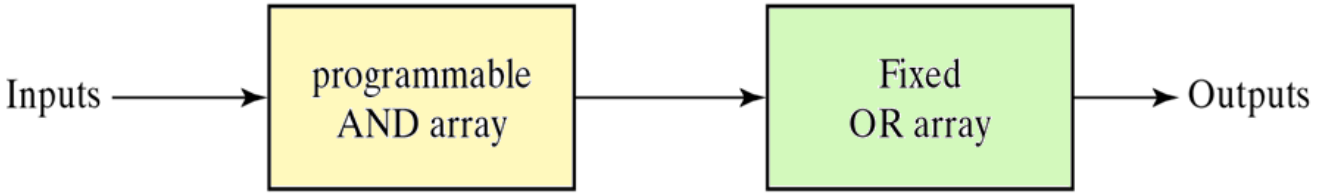
معماری‌های مختلف PLDهای ترکیبی

PROM



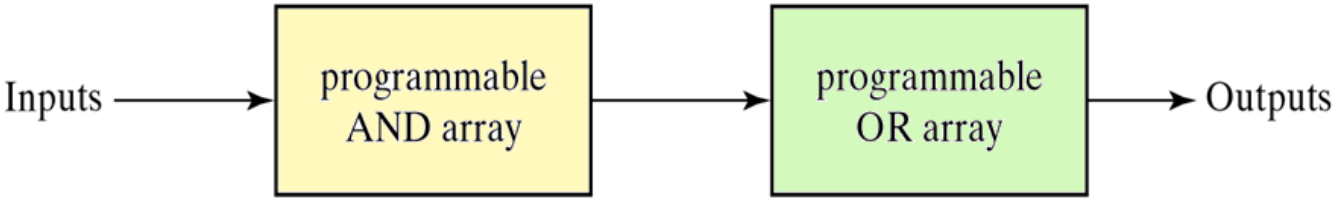
(a) Programmable read-only memory (PROM)

PAL



(b) Programmable array logic (PAL)

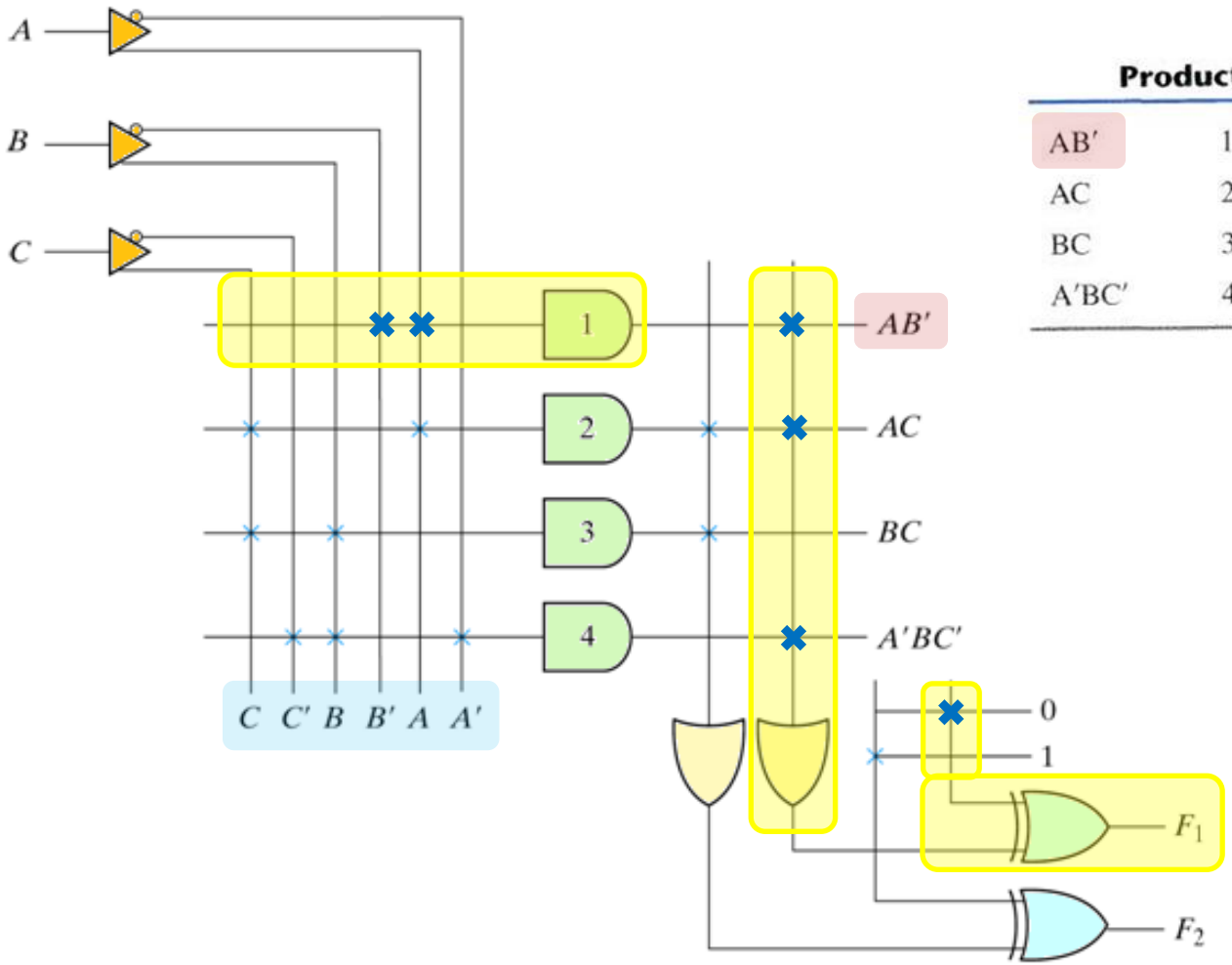
PLA



(c) Programmable logic array (PLA)

PLA

PLA با سه ورودی، چهار جمله‌ی ضرب و دو خروجی:



PLA Programming Table

	Product Term	Inputs A	Inputs B	Inputs C	Outputs (T) F_1	Outputs (C) F_2
AB′	1	1	0	–	1	–
AC	2	1	–	1	1	1
BC	3	–	1	1	–	1
A′BC′	4	0	1	0	1	–

$F_1 = AB' + AC + A'BC'$

$F_2 = (AC + BC)'$

مثال طراحی با PLA

با استفاده از PLA، مدار توابع زیر را پیاده‌سازی نمایید.

$$F_1(A,B,C) = \sum(0,1,2,4)$$
$$F_2(A,B,C) = \sum(0,5,6,7)$$

مراحل طراحی

ترسیم نقشه و ساده‌سازی به فرم اصلی و مکمل

انتخاب فرم‌های مناسب

ترسیم جدول برنامه‌ریزی

A \ BC	00	01	11	10
0	1	1	0	1
1	1	0	0	0

C

A \ BC	00	01	11	10
0	1	0	0	0
1	0	1	1	1

C

$$F_1 = A'B' + A'C' + B'C'$$
$$F_1 = (AB + AC + BC)'$$

$$F_2 = AB + AC + A'B'C'$$
$$F_2 = (A'C + A'B + AB'C')'$$

$$F_1 = (AB + AC + BC)'$$
$$F_2 = AB + AC + A'B'C'$$

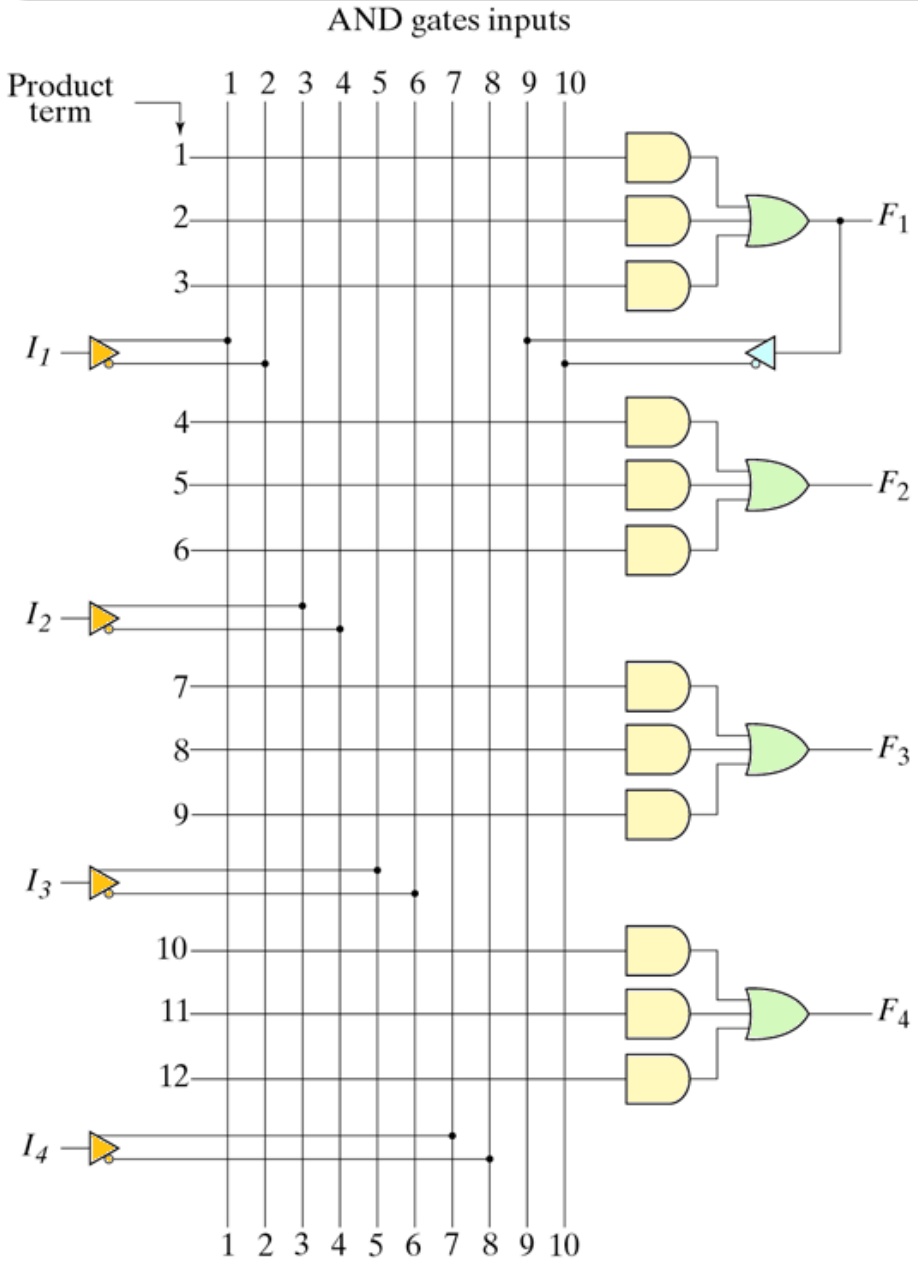
PLA programming table

	Product term	Inputs A	B	C	Outputs (C) F_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1

PAL

PAL با چهار ورودی، چهار خروجی

و ساختار AND-OR سه ردیفی



طراحی مدار با PAL

طرحی با توابع زیر توصیف شده است. می خواهیم طرح را با PAL چهار ورودی، چهار خروجی با ساختار AND-OR سه ردیفی پیاده‌سازی نماییم.

$$w(A,B,C,D)=\sum(2,12,13)$$

$$x(A,B,C,D)=\sum(7,8,9,10,11,12,12,14,15)$$

$$y(A,B,C,D)=\sum(0,2,3,4,5,6,7,8,10,11,15)$$

$$z(A,B,C,D)=\sum(1,2,8,12,13,)$$

ساده‌سازی با نقشه

ترسیم جدول برنامه‌ریزی

$$w=ABC'+A'B'CD'$$

$$x=A+BCD$$

$$y=A'B+CD+B'D'$$

$$z=ABC'+A'B'CD'+AC'D'+A'B'C'D$$

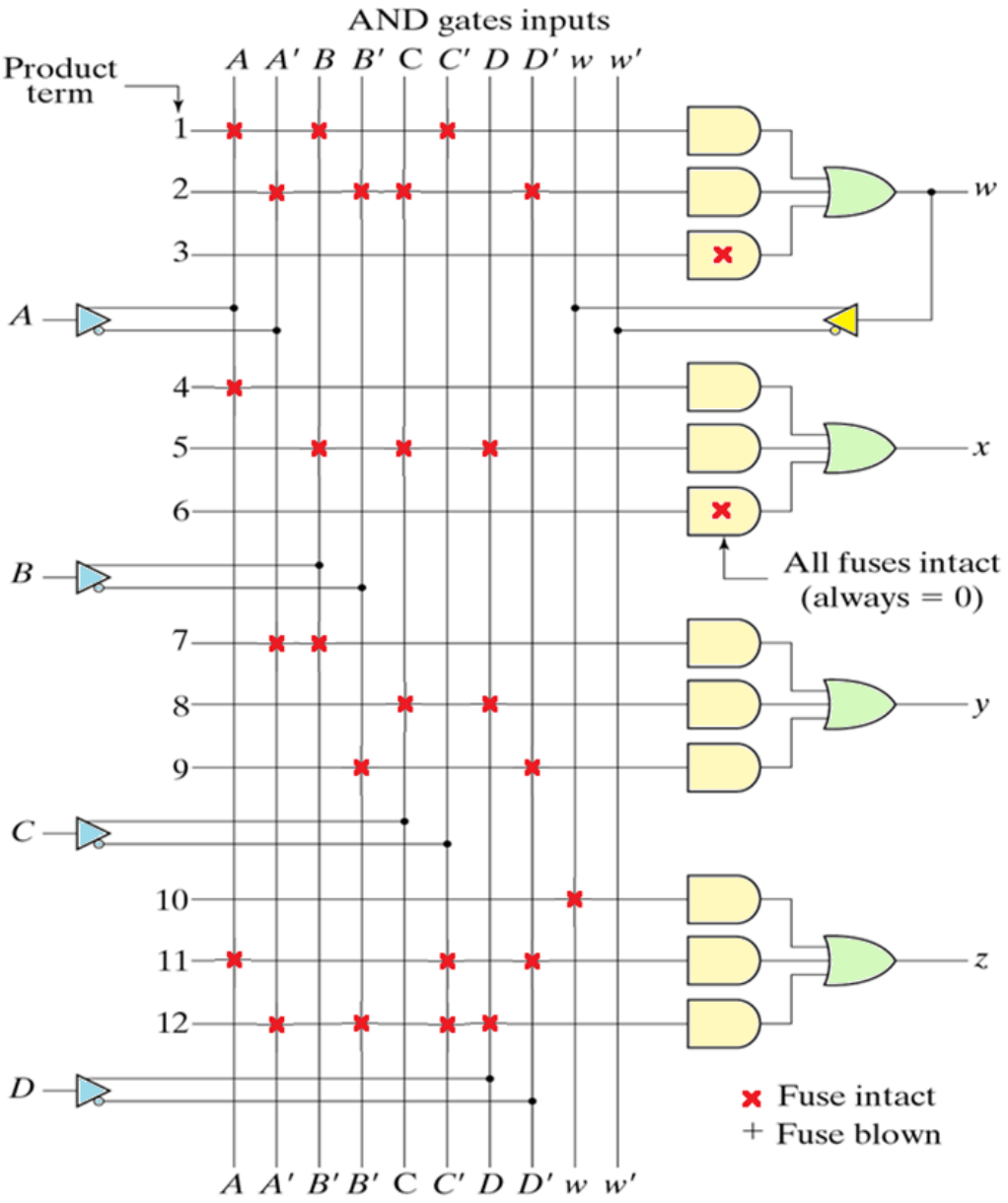
$$=w+AC'D'+A'B'C'D$$

PAL Programming Table

Product Term	AND Inputs A	B	C	D	W	Outputs
1	1	1	0	–	–	$w = ABC'$
2	0	0	1	0	–	$+ A'B'CD'$
3	–	–	–	–	–	
4	1	–	–	–	–	$x = A$
5	–	1	1	1	–	$+ BCD$
6	–	–	–	–	–	
7	0	1	–	–	–	$y = A'B$
8	–	–	1	1	–	$+ CD$
9	–	0	–	0	–	$+ B'D'$
10	–	–	–	–	1	$z = w$
11	1	–	0	0	–	$+ AC'D'$
12	0	0	0	1	–	$+ A'B'C'D$

طراحی مدار با PAL

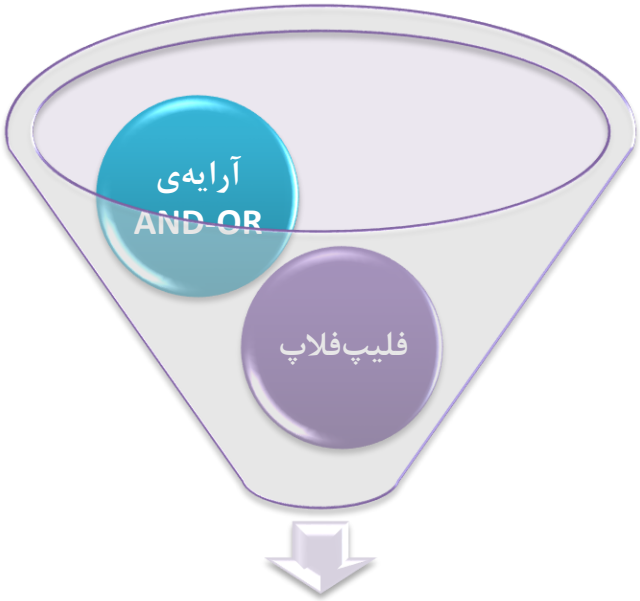
AND Inputs					Outputs
A	B	C	D	W	
1	1	0	–	–	$w = ABC'$
0	0	1	0	–	$+ A'B'CD'$
–	–	–	–	–	
1	–	–	–	–	$x = A$
–	1	1	1	–	$+ BCD$
–	–	–	–	–	
0	1	–	–	–	$y = A'B$
–	–	1	1	–	$+ CD$
–	0	–	0	–	$+ B'D'$
–	–	–	–	1	$z = w$
1	–	0	0	–	$+ AC'D'$
0	0	0	1	–	$+ A'B'C'D$



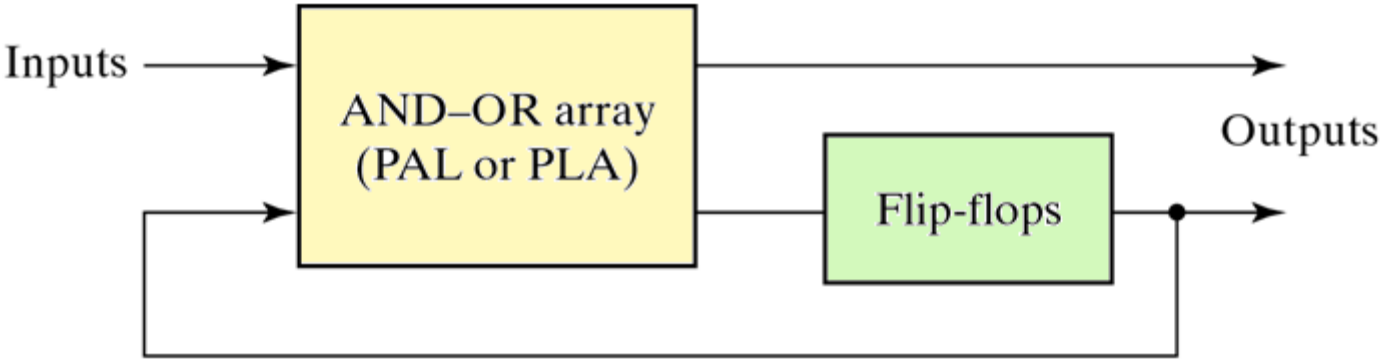
فهرست مطالب

- [x] مقدمه ای بر حافظه ها
- [x] حافظه ی RAM
- [x] تشخیص و تصحیح خطا
- [x] حافظه ی ROM
- [x] منطق برنامه پذیر
- [x] PROM، PLA و PAL
- [ ] CPLD و FPGA

PLDی ترتیبی (Sequential PLD)



SPLD (Sequential PLD)



معرفی چند اصطلاح

SPLD

می‌تواند به عنوان سرنام عبارت‌های زیر استفاده گردد:

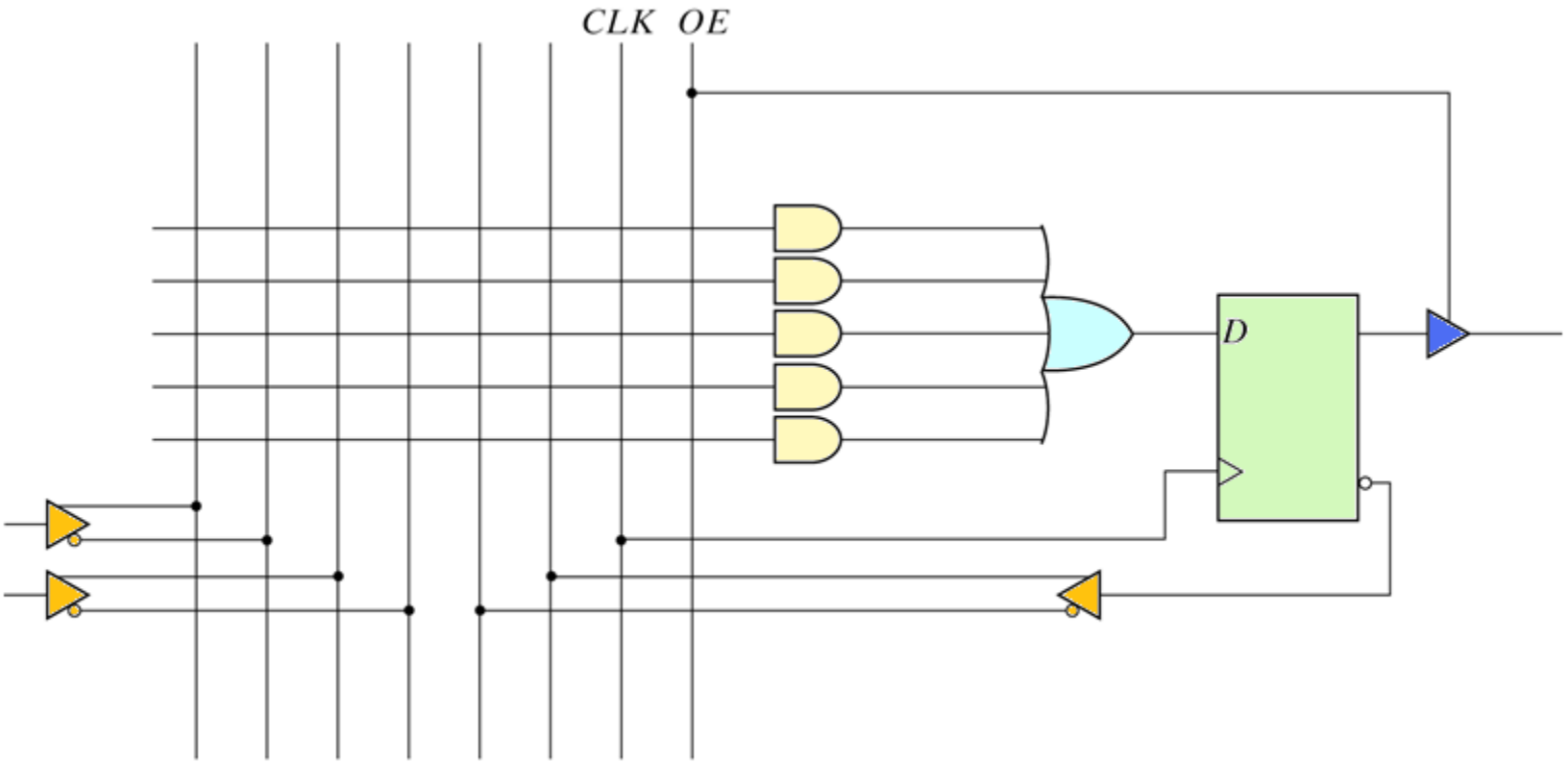
Simple PLD , Sequential PLD

Macrocell

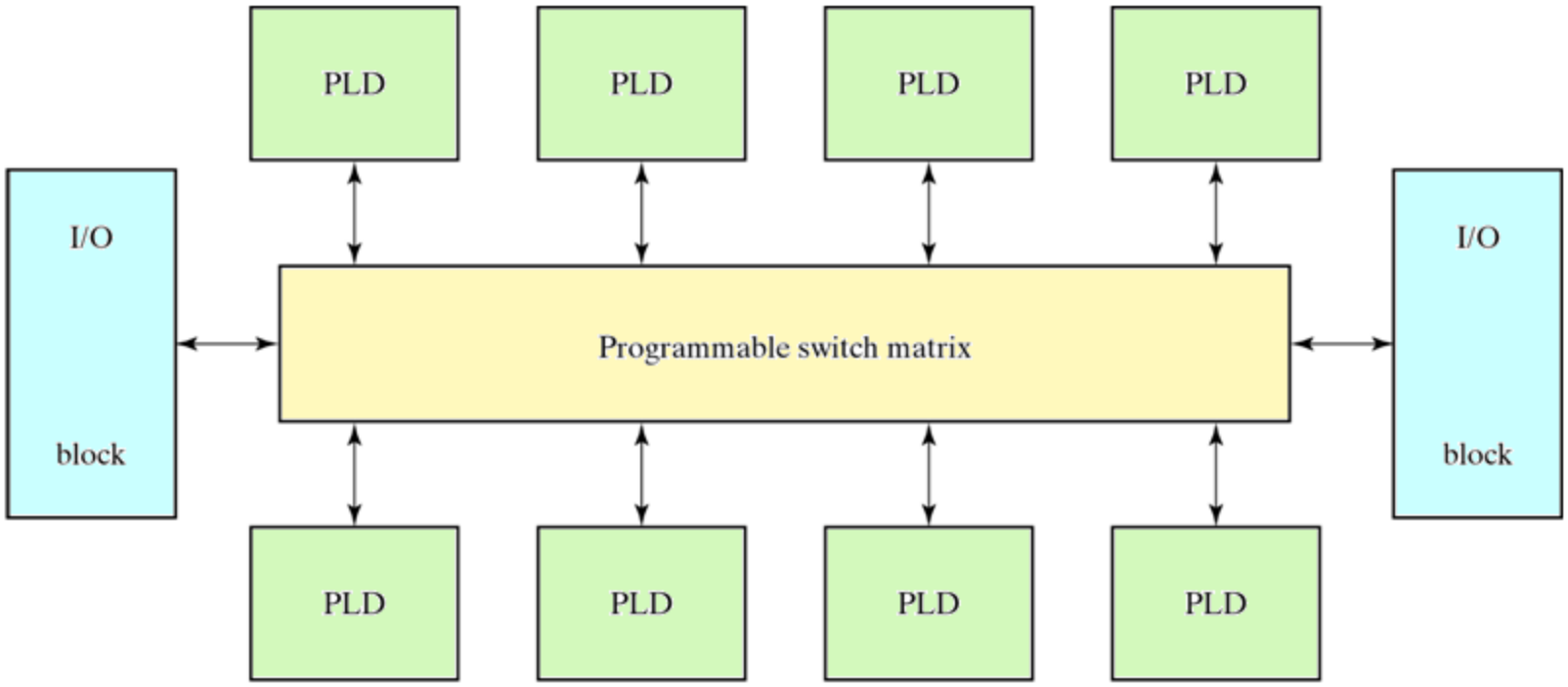
هر یک از بخش‌های پایه‌ی تشکیل دهنده‌ی SPLD را ماکروسِل گویند.

Sequential PLD

✓ مدار منطقی یک ماکروسل



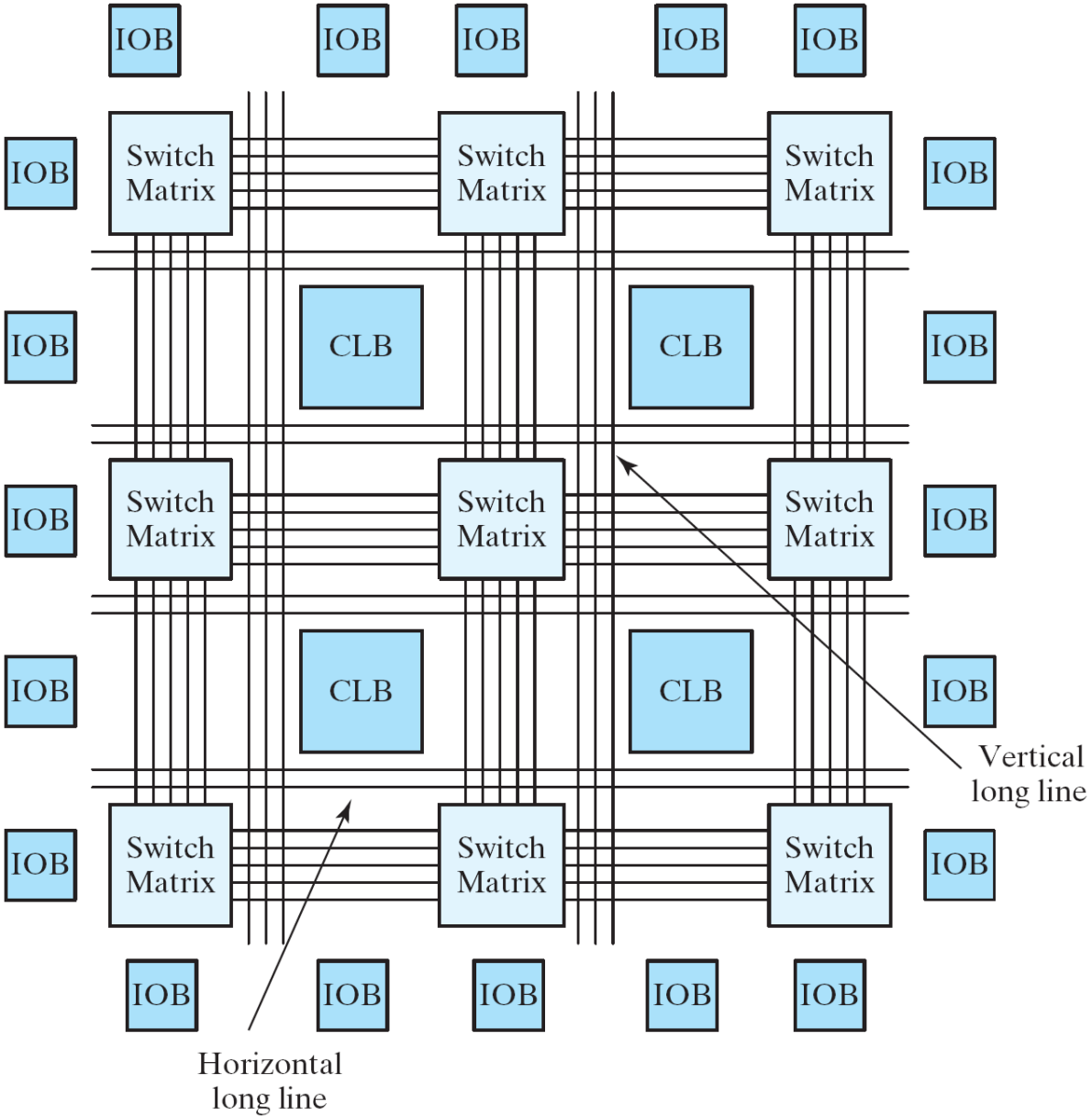
پیکربندی CPLD



FPGA

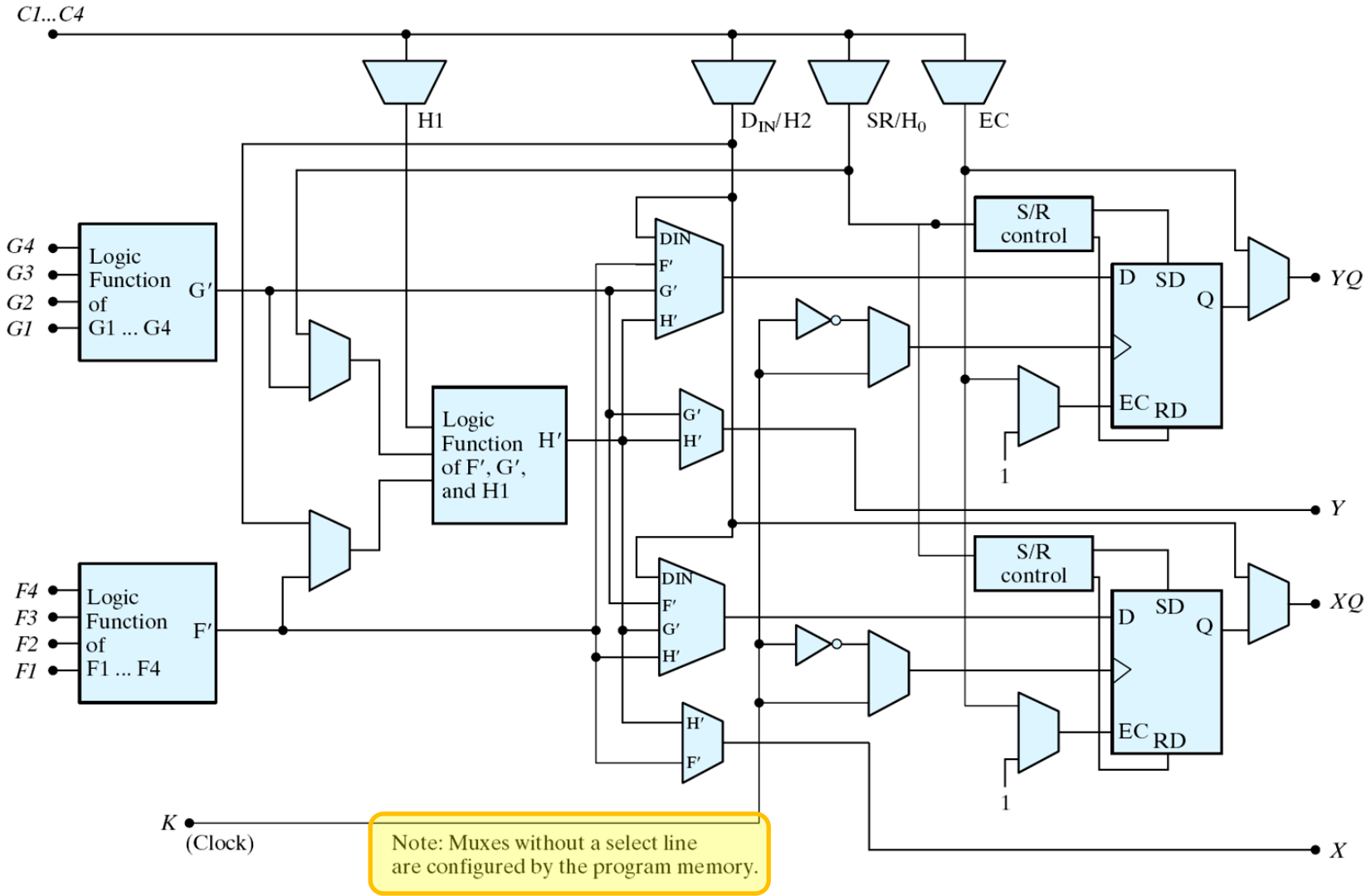
✓ تکامل FPGAهای شرکت Xilinx :

معماری پایه‌ی قطعات Spartan شرکت Xilinx



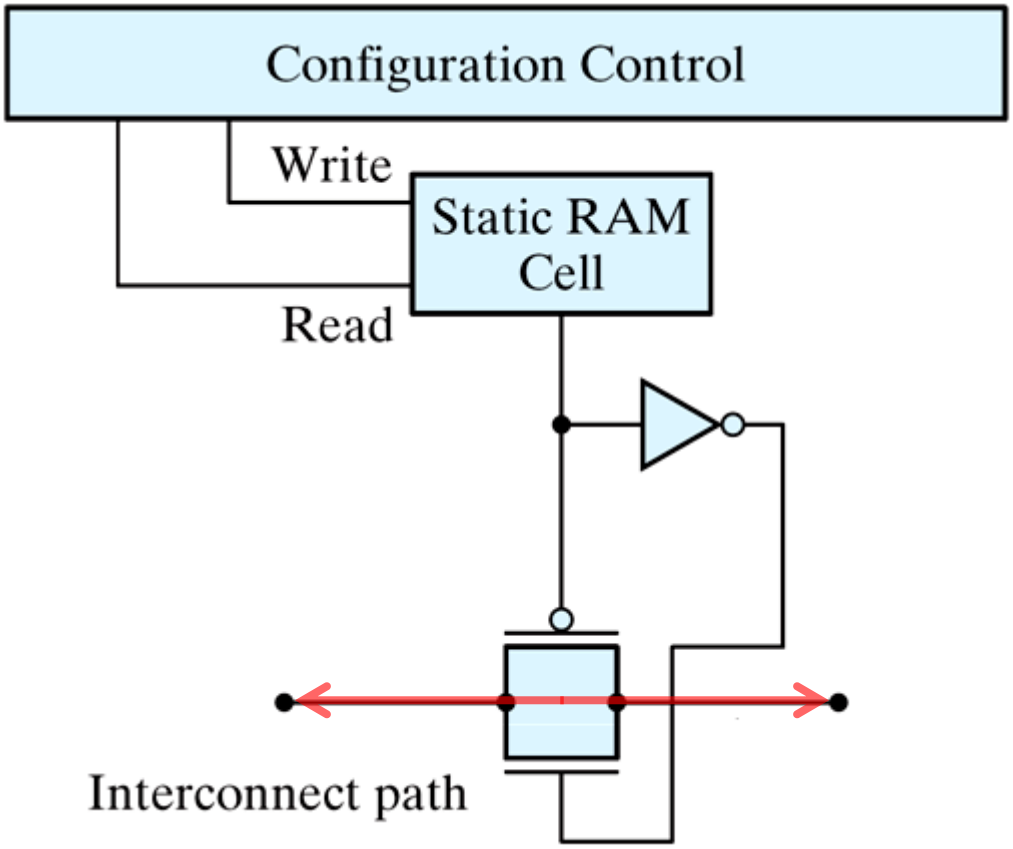
معماری پایه‌ی قطعات Spartan شرکت Xilinx

بلوک منطقی قابل پیکربندی



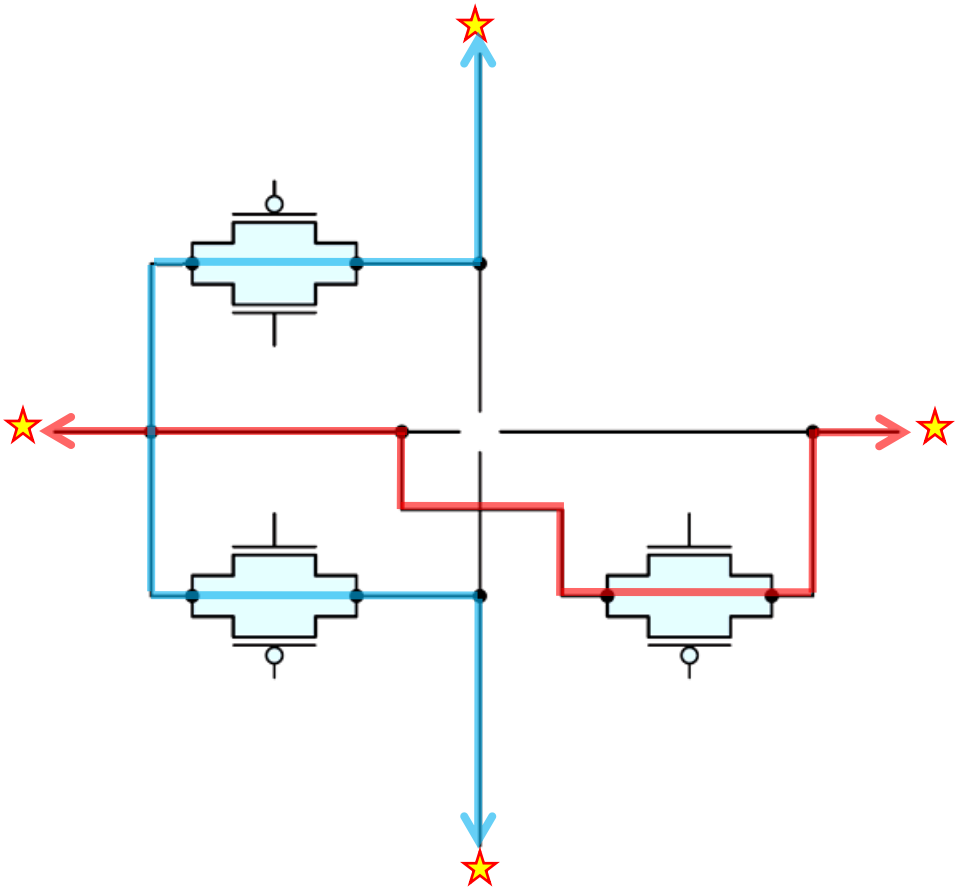
معماری پایه‌ی قطعات Spartan شرکت Xilinx

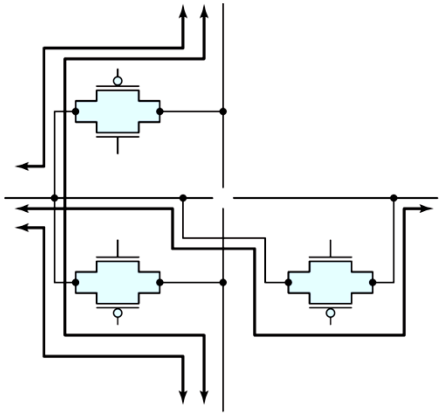
✓ سلول RAM کنترل‌کننده‌ی گیت انتقال PIP



نقطه‌ی اتصال قابل برنامه‌ریزی (Programmable Interconnect Point-PIP)

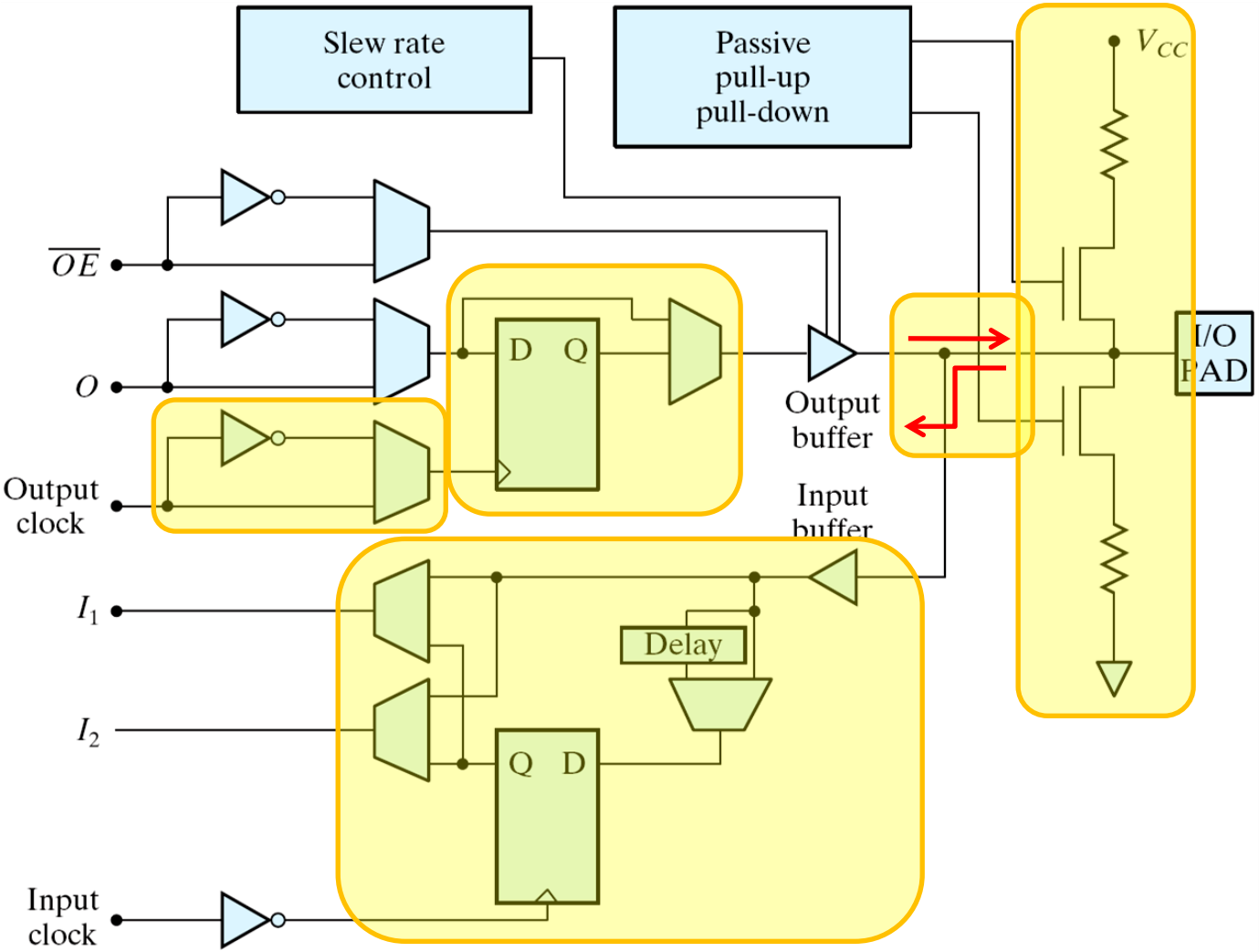
✓نقطه‌ی اتصال قابل برنامه‌ریزی





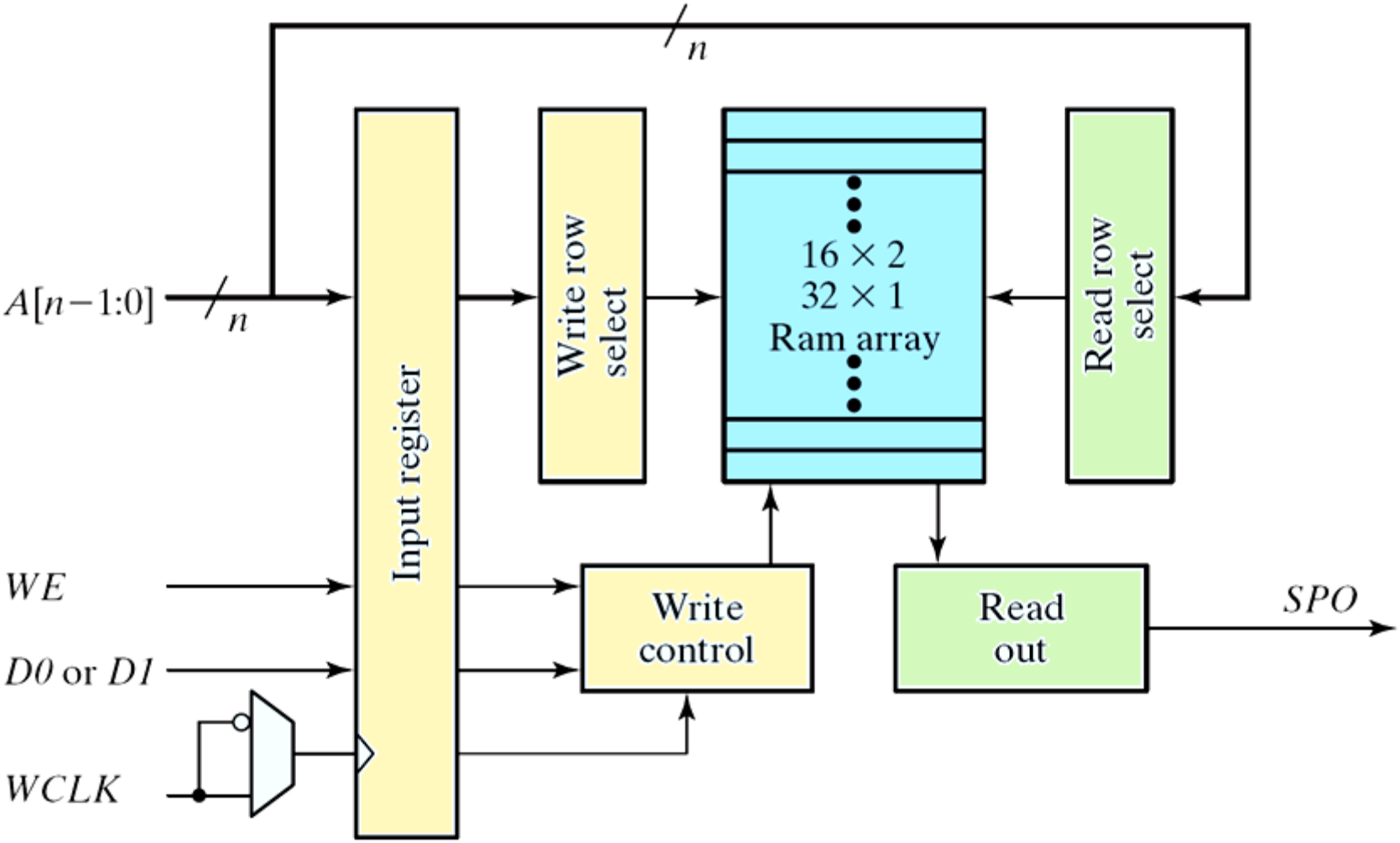
بلوک ورودی/خروجی (I/O Block)

✓بلوک ورودی/ خروجی



معماری پایه‌ی قطعات Spartan شرکت Xilinx

✓ سلول RAM توزیع شده با استفاده از جدول جستجو (LUT)



FPGA های خانواده‌ی Spartan XL شرکت Xilinx

Table 7.7
Attributes of the Xilinx Spartan XL Device Family

Spartan XL	XCS05/XL	XCS10/XL	XCS20/XL	XCS30/XL	XCS40/XL
System Gates[1]	2K–5K	3K–10K	7K–20K	10K–30K	13K–40K
Logic Cells[2]	238	466	950	1,368	1,862
Max Logic Gates	3,000	5,000	10,000	13,000	20,000
Flip-Flops	360	616	1,120	1,536	2,016
Max RAM Bits	3,200	6,272	12,800	18,432	25,088
Max Avail I/O	77	112	160	192	224

[1] 20–30% of CLBs as RAM.
[2] 1 Logic cell = four-input lookup table + flip-flop.

FPGA های خانواده ی Spartan II شرکت Xilinx

Table 7.8
Spartan II Device Attributes

Spartan II FPGAs	XC2S15	XC2S30	XC2S50	XC2S100	XC2S150	XC2S200
System Gates[1]	6K–15K	13K–30K	23K–50K	37K–100K	52K–150K	71K–200K
Logic Cells[2]	432	972	1,728	2,700	3,888	5,292
Block RAM Bits	16,384	24,576	32,768	40,960	49,152	57,344
Max Avail I/O	86	132	176	196	260	284

[1] 20–30% of CLBs as RAM.
[2] 1 Logic cell = four-input lookup table + flip-flop.

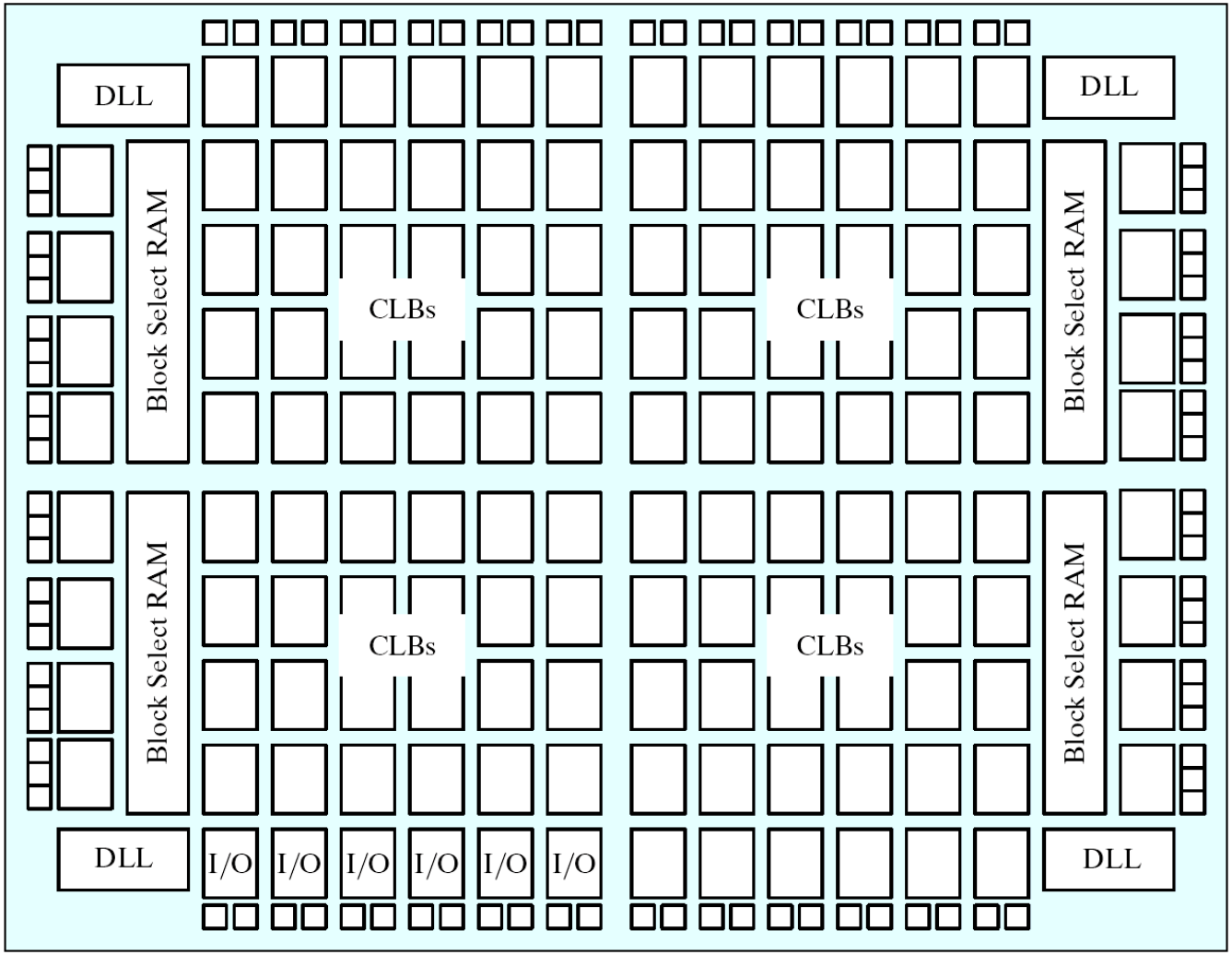
FPGA های خانواده ی Spartan II شرکت Xilinx

Table 7.9
Comparison of the Spartan Device Families

Part	Spartan	Spartan XL	Spartan II
Architecture	XC4000 Based	XC4000 Based	Virtex Based
Max # System Gates	5K–40K	5K–40K	15K–200K
Memory	Distributed RAM	Distributed RAM	Block + Distributed
I/O Performance	80 MHz	100 MHz	200 MHz
I/O Standards	4	4	16
Core Voltage	5 V	3.3 V	2.5 V
DLLs	No	No	Yes

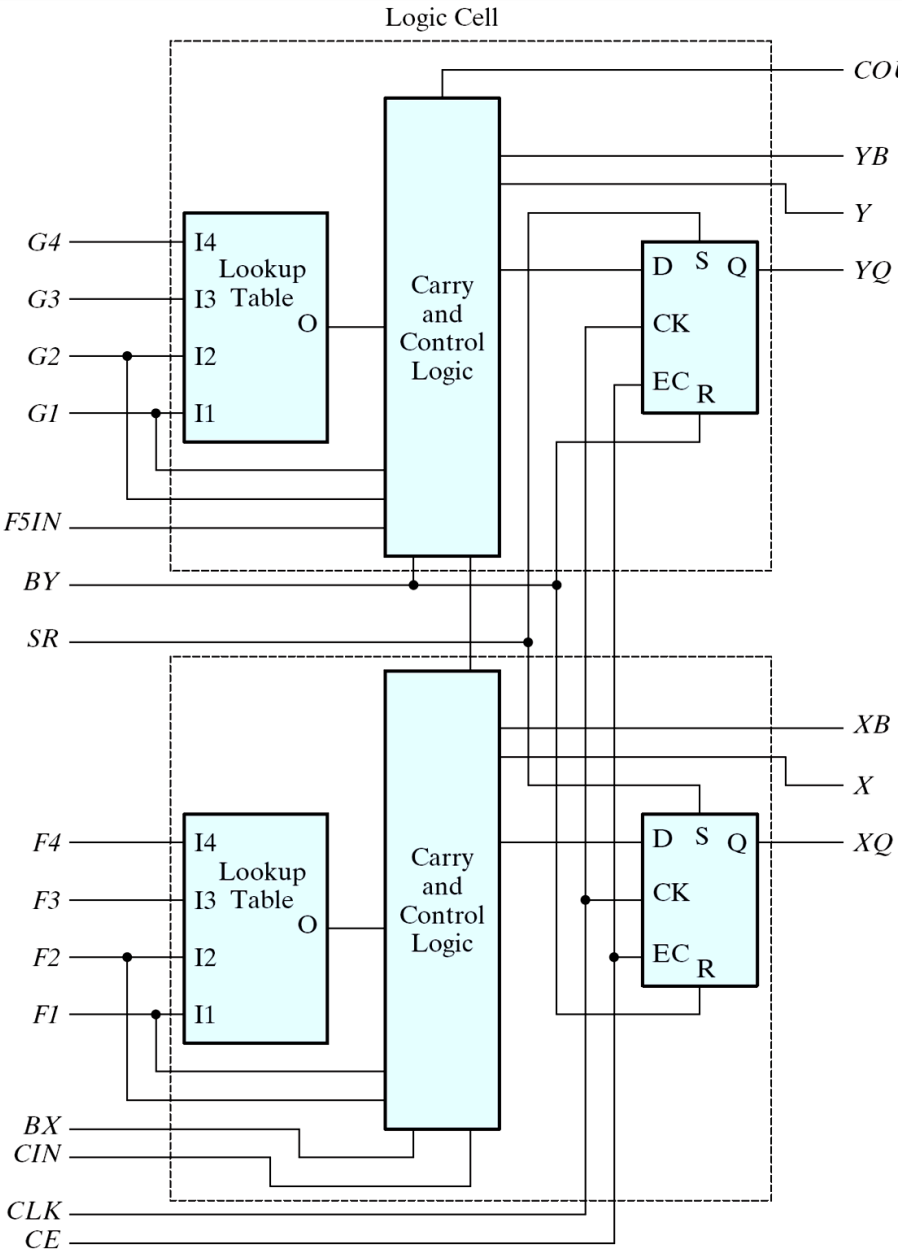
FPGA های خانواده ی Spartan II شرکت Xilinx

✓ معماری



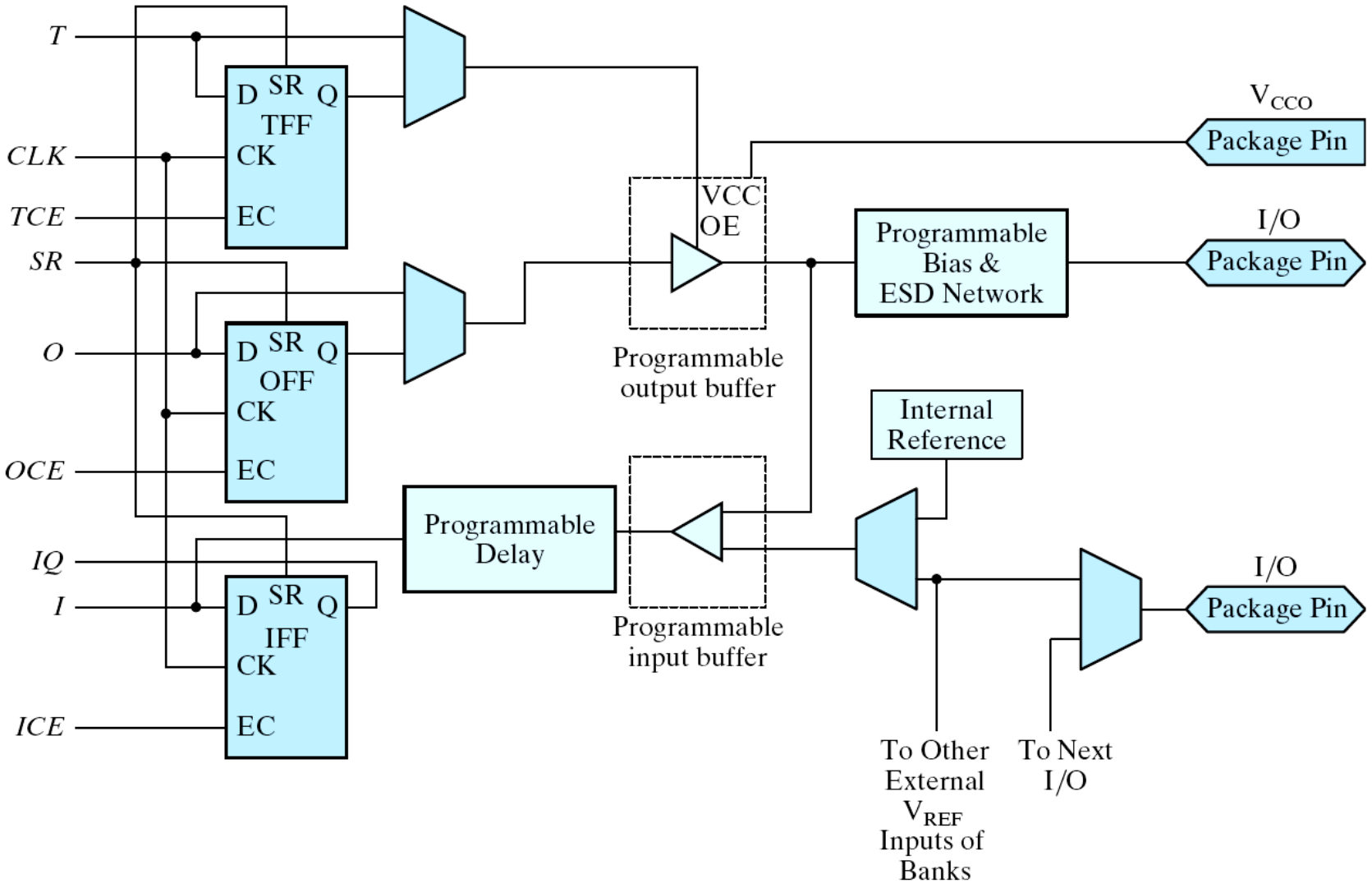
FPGA های خانواده ی Spartan II شرکت Xilinx

Slice



FPGA های خانواده ی Spartan II شرکت Xilinx

I/O Block ✓



FPGA های خانواده ی Virtex شرکت Xilinx

گره تکنولوژی: ۶۵ نانومتر

ولتاژ کاری: ۱ ولت

فرکانس کاری: ۵۵۰ مگاهرتز

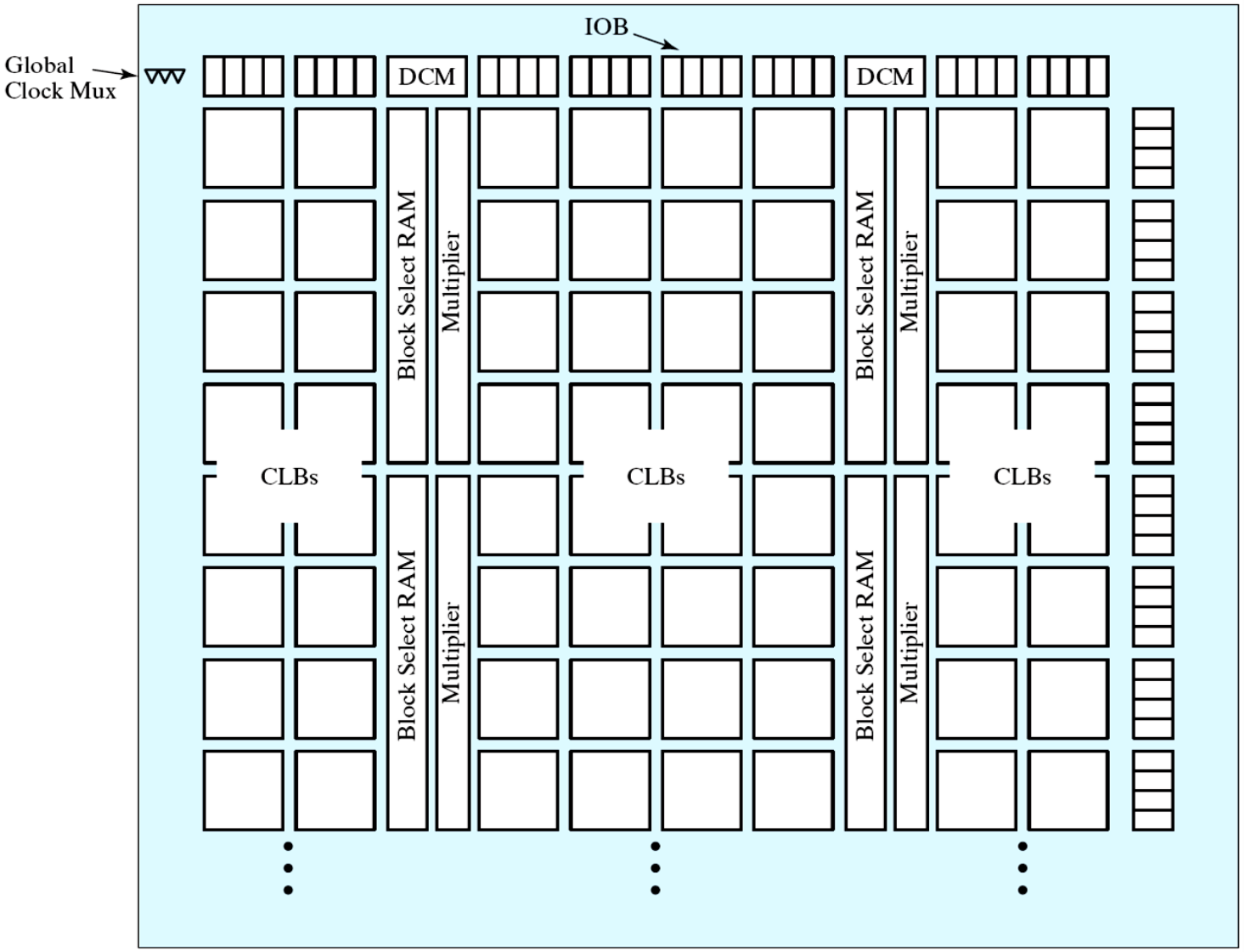
RAM بلوکی: بیش از یک مگابیت

فلیپ فلاپ درونی: ۲۰۰۰۰۰

سلول منطقی: ۳۳۰۰۰۰۰

FPGA های خانواده‌ی Virtex شرکت Xilinx

معماری



DCM: Clock Manager

فهرست مطالب

- مقدمه ای بر حافظه ها
- حافظه ی RAM
- تشخیص و تصحیح خطا
- حافظه ی ROM
- منطق برنامه پذیر
- PROM، PLA و PAL
- CPLD و FPGA