

مبحث هفتم

حافظه و منطق برنامه پذیر Memory and Programmable Logic

Presented by Ali Maleki Spring Semester, 2009

تعریف حافظه و منطق برنامه پذیر

✓ حافظه (Memory)

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

✓ منطق برنامه پذیر (Programmable Logic)

قطعه‌ای با امکان آرایش دادن سخت افزار بر اساس بیت‌های ذخیره شده

فهرست مطالب

مقدمه‌ای بر حافظه‌ها

حافظه‌ی RAM

تشخیص و تصحیح خطا

حافظه‌ی ROM

منطق برنامه پذیر

PAL و PLA ، PROM

FPGA و CPLD

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 بیت‌ها

✓ 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 ($k=2^{10}$) ، میگا ($M=2^{20}$) ، گیگا ($G=2^{30}$)

✓ حافظه‌ی فرار (Volatile Memory):

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

✓ حافظه‌ی غیر فرار (Nonvolatile Memory):

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

حافظه با دسترسی تصادفی (RAM)

✓ علت نامگذاری RAM:

در هر زمان، به هر مکان حافظه می توان دسترسی داشت.
 زمان لازم برای انتقال اطلاعات به/از هر مکان حافظه یکسان است.

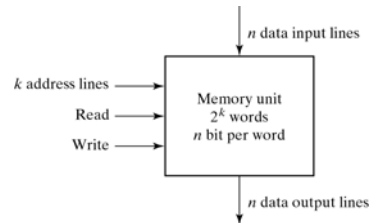
RAM در قیاس با حافظه با دسترسی ترتیبی (sequential access memory) مطرح می گردد.

به عنوان نمونه ای از حافظه با دسترسی ترتیبی می توان به حافظه از نوع نوار مغناطیسی اشاره نمود.

RAM یک حافظه ی فرار است.

RAM

✓ نمودار بلوکی حافظه ی RAM



RAM

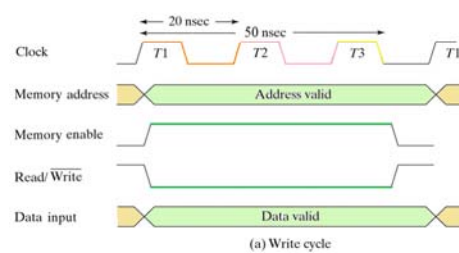
✓ محتوای یک حافظه ی 16 * 1024

Memory address		Memory content
Binary	decimal	
000000000	0	1011010101011101
000000001	1	1010101110001001
000000010	2	0000110101000110
⋮	⋮	⋮
111111101	1021	1001110100010100
111111110	1022	0000110100011110
111111111	1023	110111000100101

RAM

نمودار زمانی سیکل نوشتن حافظه

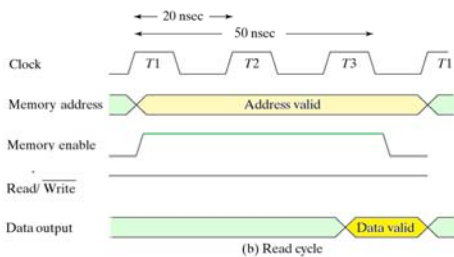
زمان سیکل (cycle time): زمان لازم برای تکمیل نوشتن در یک حافظه
 در این مثال، زمان سیکل را ۵۰ نانوثانیه فرض کنید.



RAM

نمودار زمانی سیکل خواندن حافظه

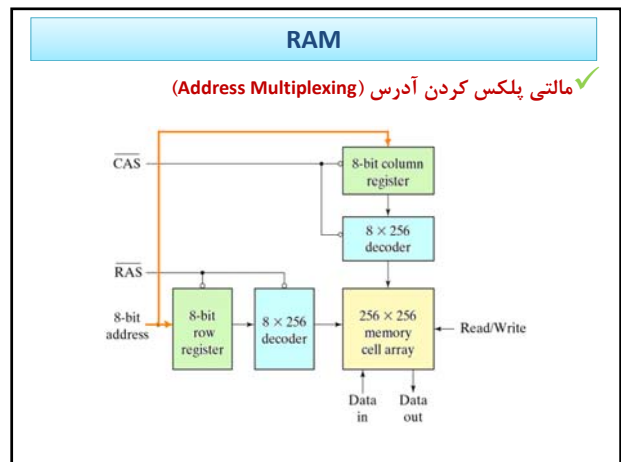
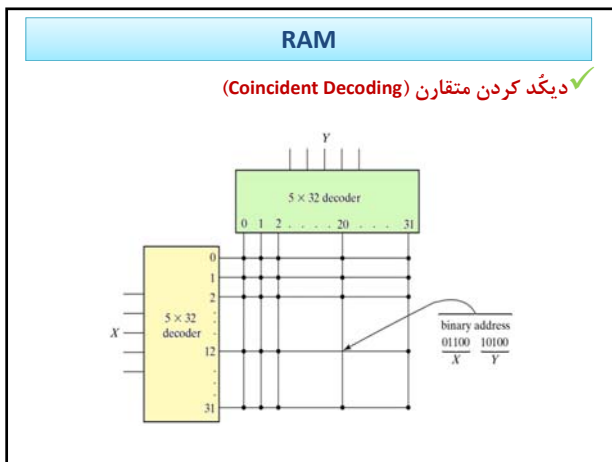
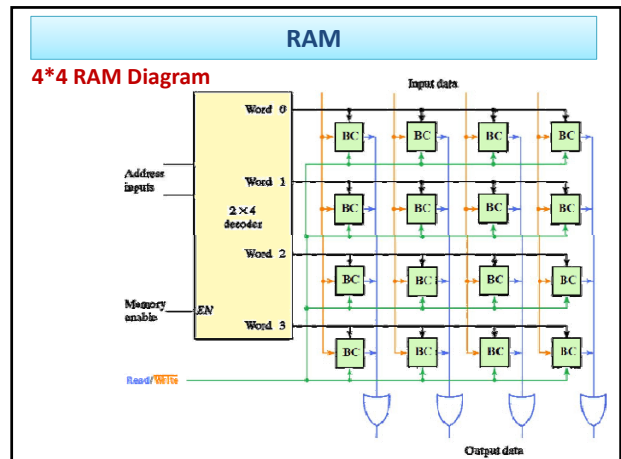
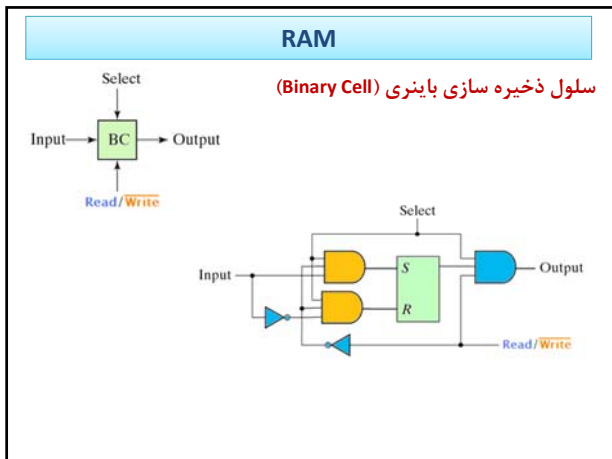
زمان دستیابی (access time): زمان لازم برای انتخاب یک کلمه و خواندن آن از حافظه.
 در این مثال، زمان دستیابی را ۵۰ نانوثانیه فرض کنید.



RAM

✓ انواع RAM:

- SRAM یا RAM استاتیکی (Static RAM)
- DRAM یا RAM دینامیک (Dynamic RAM)



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 همینگ و بیت توازن

کد همینگ (Hamming Code)

کد هشتم بیتی 11000100 را در نظر بگیرید. در کد همینگ، 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

$P_1 = \text{XOR of bits}(3,5,7,9,11)=0$
 $P_2 = \text{XOR of bits}(3,6,7,10,11)=0$
 $P_4 = \text{XOR of bits}(5,6,7,12)=1$
 $P_8 = \text{XOR of bits}(9,10,11,12)=1$

In memory, 001110010100

کد همینگ (Hamming Code)

تشخیص و تصحیح خطا با استفاده از کد همینگ

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

Check Bits:

$$C_1 = \text{XOR of bits (1,3,5,7,9,11)}$$

$$C_2 = \text{XOR of bits (2,3,6,7,10,11)}$$

$$C_4 = \text{XOR of bits (4,5,6,7,12)}$$

$$C_8 = \text{XOR of bits (8,9,10,11,12)}$$

$$C = C_8 C_4 C_2 C_1$$

$C = 0$: No Error

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

کد همینگ (Hamming Code)

مثال ۱: با فرض استفاده از کد همینگ، 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:

$$0 0 1 1 1 0 0 1 0 1 0 0 \quad C_1 = \text{XOR of bits (1,3,5,7,9,11)} = 0$$

$$0 0 1 1 1 0 0 1 0 1 0 0 \quad C_2 = \text{XOR of bits (2,3,6,7,10,11)} = 0$$

$$0 0 1 1 1 0 0 1 0 1 0 0 \quad C_4 = \text{XOR of bits (4,5,6,7,12)} = 0$$

$$0 0 1 1 1 0 0 1 0 1 0 0 \quad C_8 = \text{XOR of bits (8,9,10,11,12)} = 0$$

$C = 0000$

بنابراین خطایی رخ نداده است.

کد همینگ (Hamming Code)

مثال ۲: با فرض استفاده از کد همینگ، 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:

$$1 0 1 1 1 0 0 1 0 1 0 0 \quad C_1 = \text{XOR of bits (1,3,5,7,9,11)} = 1$$

$$1 0 1 1 1 0 0 1 0 1 0 0 \quad C_2 = \text{XOR of bits (2,3,6,7,10,11)} = 0$$

$$1 0 1 1 1 0 0 1 0 1 0 0 \quad C_4 = \text{XOR of bits (4,5,6,7,12)} = 0$$

$$1 0 1 1 1 0 0 1 0 1 0 0 \quad C_8 = \text{XOR of bits (8,9,10,11,12)} = 0$$

$C = 0001$

خطایی در بیت اول رخ داده است.

کد همینگ (Hamming Code)

مثال ۳: با فرض استفاده از کد همینگ، 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:

$$0 0 1 1 0 0 0 1 0 1 0 0 \quad C_1 = \text{XOR of bits (1,3,5,7,9,11)} = 1$$

$$0 0 1 1 0 0 0 1 0 1 0 0 \quad C_2 = \text{XOR of bits (2,3,6,7,10,11)} = 0$$

$$0 0 1 1 0 0 0 1 0 1 0 0 \quad C_4 = \text{XOR of bits (4,5,6,7,12)} = 1$$

$$0 0 1 1 0 0 0 1 0 1 0 0 \quad C_8 = \text{XOR of bits (8,9,10,11,12)} = 0$$

$C = 0101$

خطایی در بیت پنجم رخ داده است.

تصحیح خطای یگانه، تشخیص خطای دوگانه

کد همینگ می‌تواند یک خطا 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 $C \neq 0$ و $P=1$ باشد یک خطا رخ داده است که قابل تصحیح است.

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

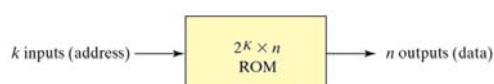
اگر $C=0$ و $P=1$ باشد یک خطا در محل بیت P_{13} رخ داده است.

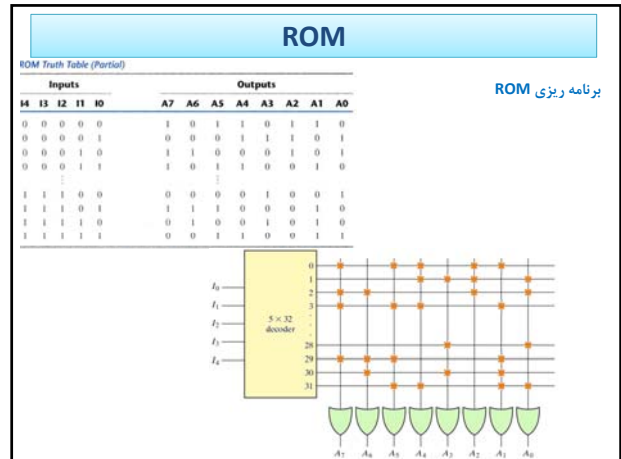
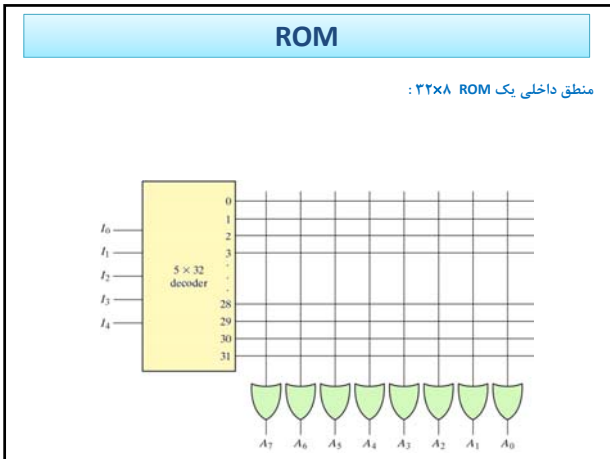
حافظه فقط خواندنی (ROM)

✓ علت نامگذاری ROM:

فقط از این حافظه خوانده می‌شود.

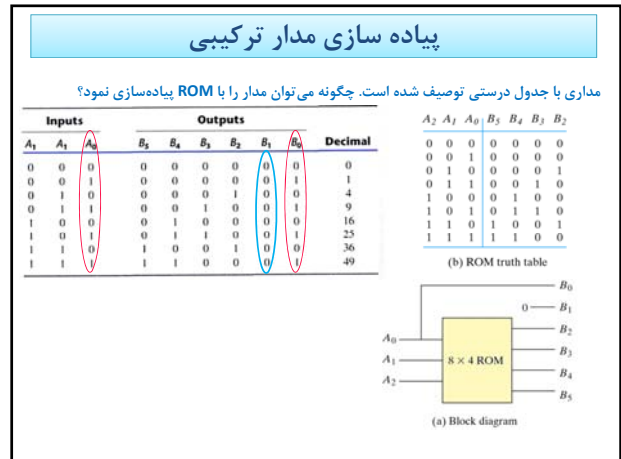
ROM یک حافظه غیرفرار است.





پیاده سازی مدار ترکیبی

ROM از یک دیکدر و تعدادی گیت OR تشکیل شده است.
 با استفاده از دیکدر می توان 2^k مینترم برای k متغیر فراهم نمود.
 بنابراین می توان با ROM، تابع را به صورت جمع مینترمها پیاده سازی نمود.

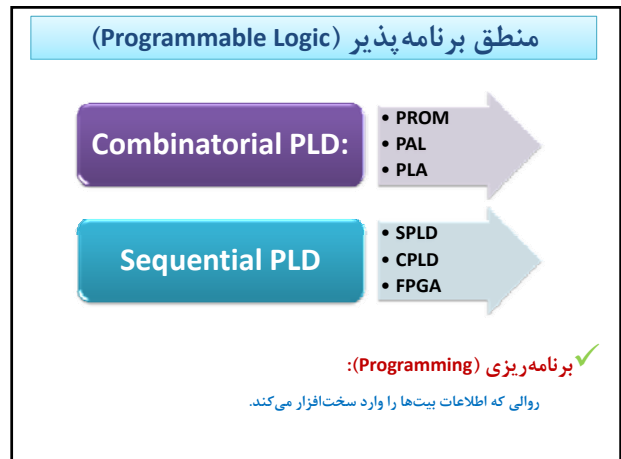


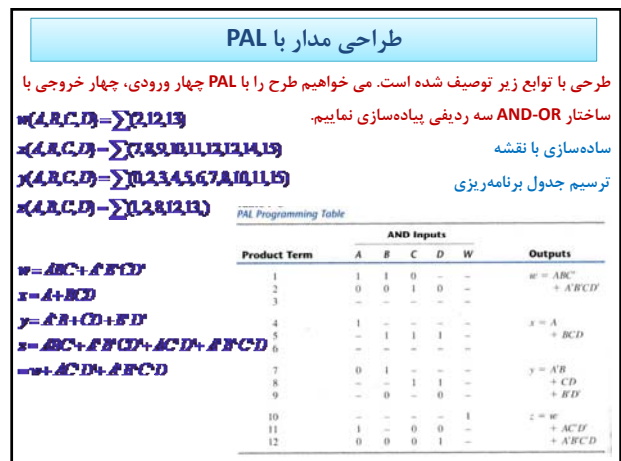
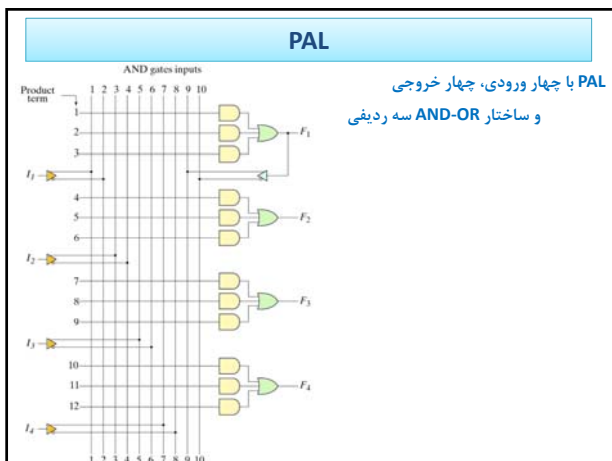
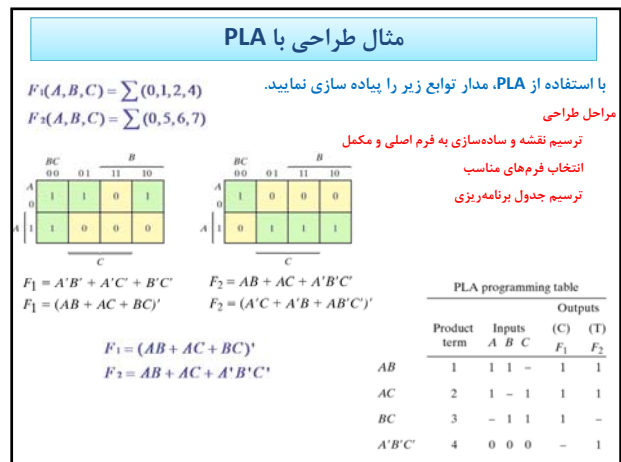
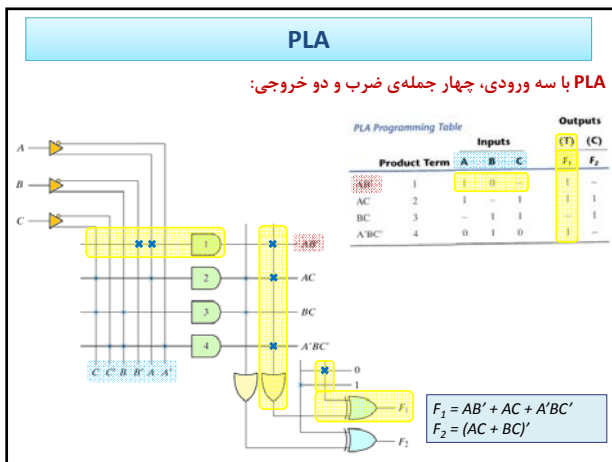
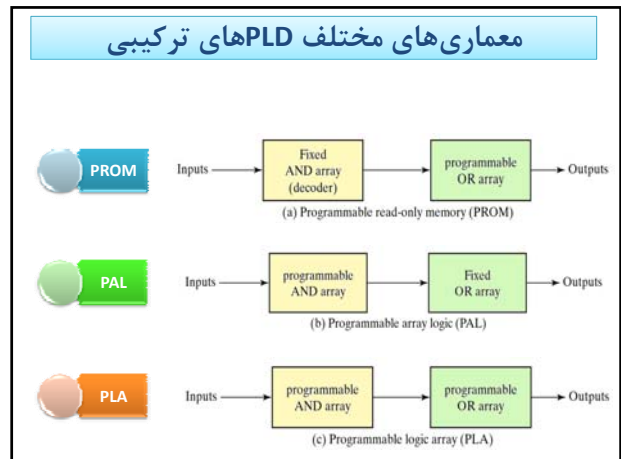
ROM

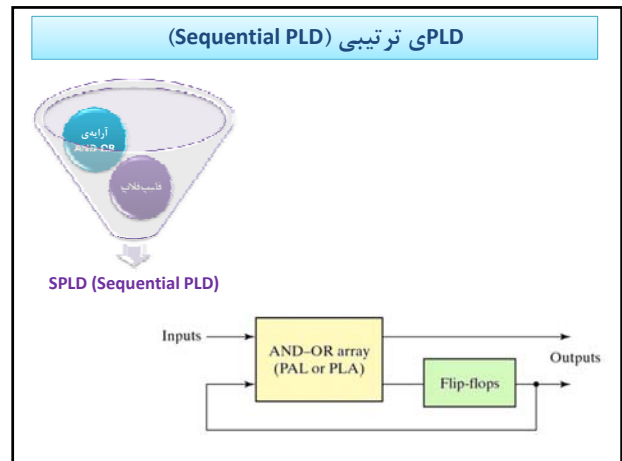
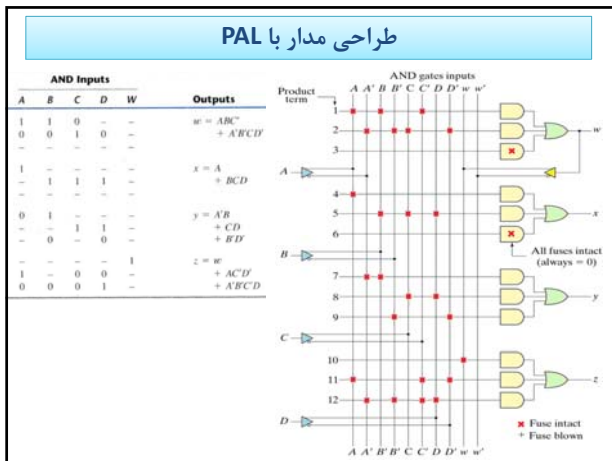
انواع ROM:

ROM	
PROM	(Programmable ROM)
EPROM	(Erasable Programmable ROM)
EEPROM or E ² PROM	(Electrically Erasable Programmable ROM)

PROM نوعی قطعه منطقی قابل برنامه ریزی (PLD) است.







معرفی چند اصطلاح

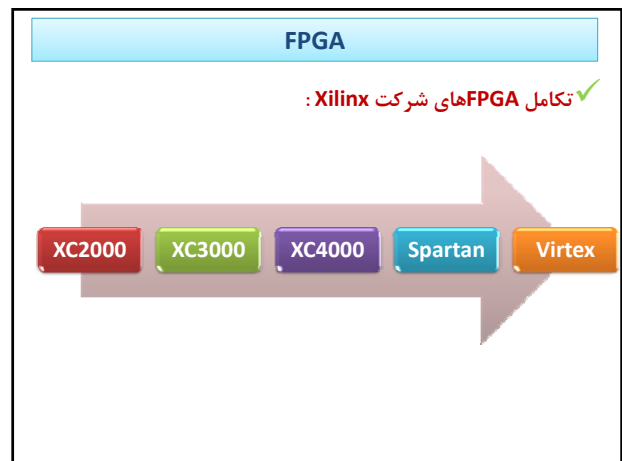
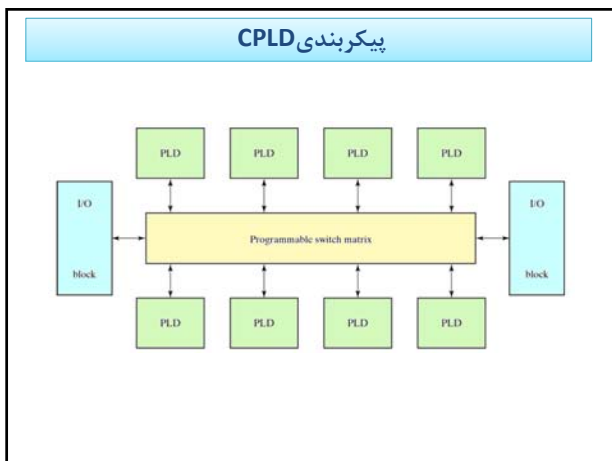
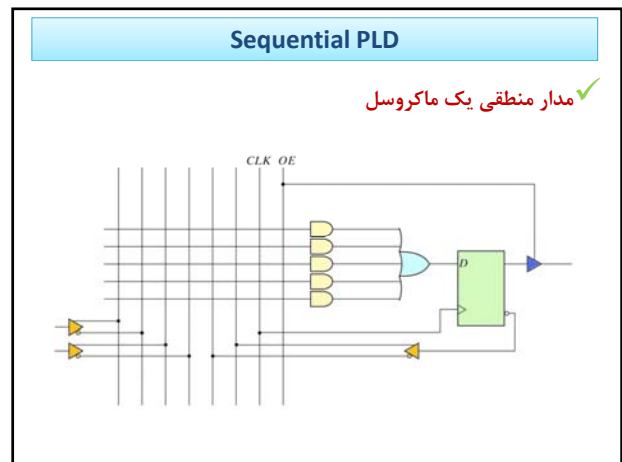
SPLD

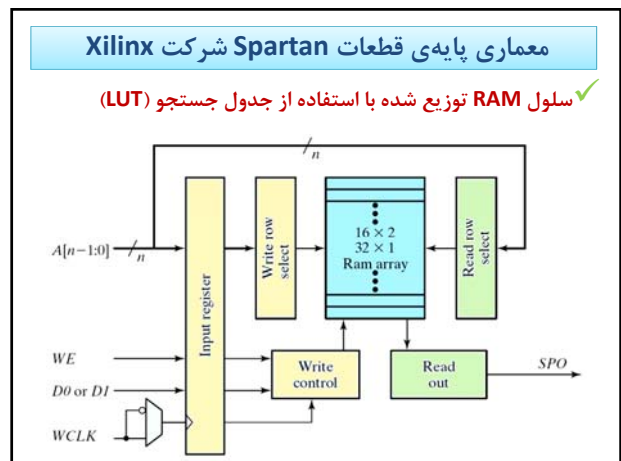
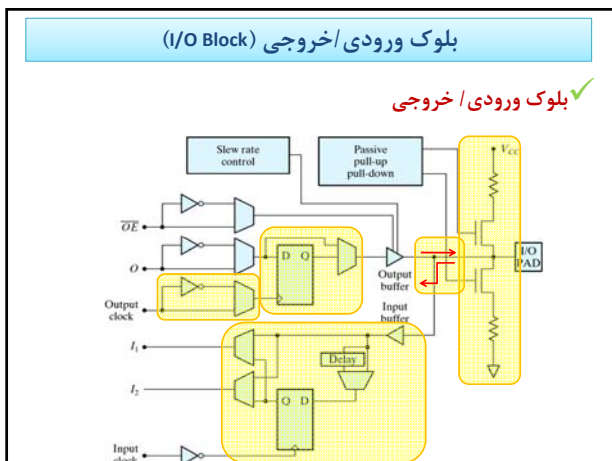
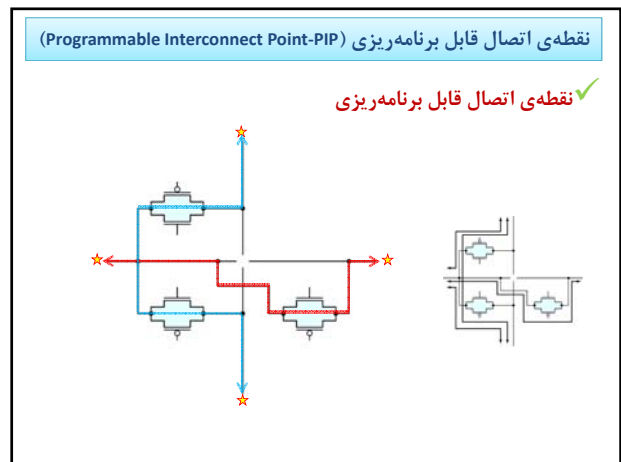
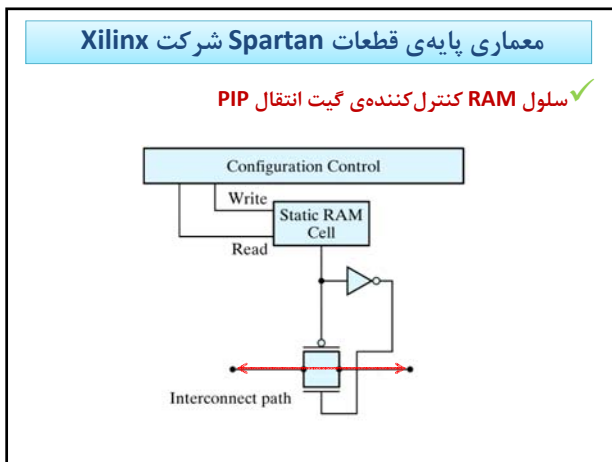
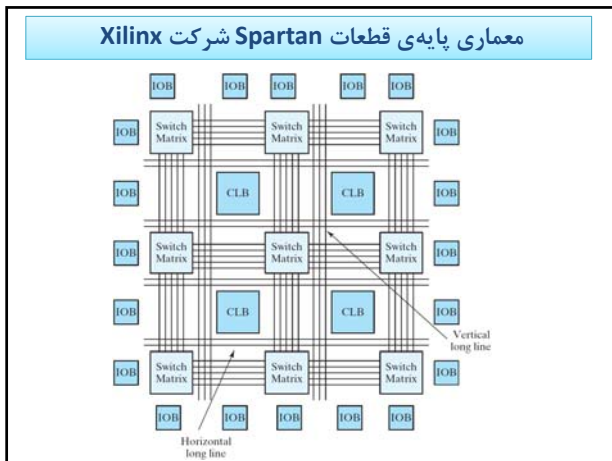
می تواند به عنوان سرنام عبارت های زیر استفاده گردد:

Simple PLD , Sequential PLD

Macrocell

هر یک از بخش های پایه ی تشکیل دهنده ی SPLD را ماکروسل گویند.





Xilinx های خانواده ی Spartan XL شرکت Xilinx

Table 7.7
Attributes of the Xilinx Spartan XL Device Family

Spartan XL	XC505/XL	XC510/XL	XC520/XL	XC530/XL	XC540/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.

Xilinx های خانواده ی Spartan II شرکت Xilinx

Table 7.8
Spartan II Device Attributes

Spartan II FPGAs	XC2515	XC2530	XC2550	XC25100	XC25150	XC25200
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.

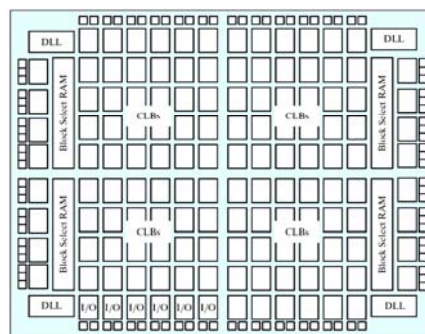
Xilinx های خانواده ی Spartan II شرکت Xilinx

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

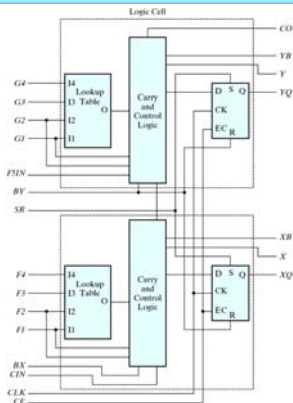
Xilinx های خانواده ی Spartan II شرکت Xilinx

معماری ✓



Xilinx های خانواده ی Spartan II شرکت Xilinx

Slice ✓



Xilinx های خانواده ی Spartan II شرکت Xilinx

I/O Block ✓

