



مدارهای منطقی برنامه پذیر

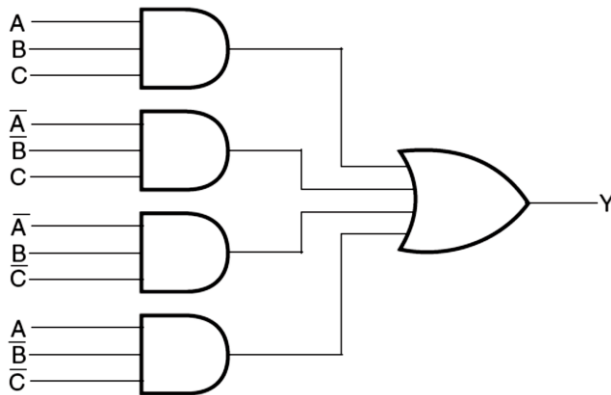
استاد درس: دکتر هادی گرایلو

فصل اول (مبانی سخت افزار)

Programmable Logic Devices (PLD)



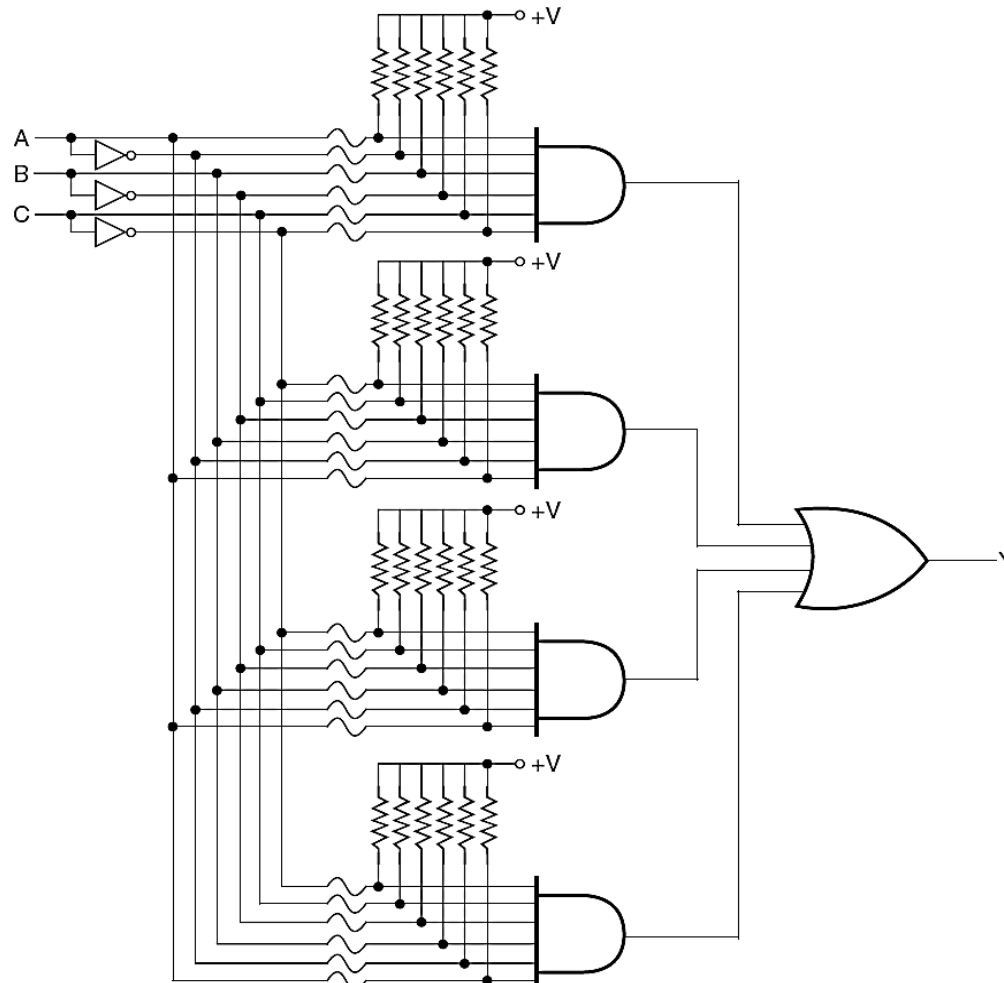
- یک سیستم الکترونیکی دیجیتال در حالت کلی از سه بخش تشکیل شده است: حافظه، میکروپروسسور، و عناصر منطقی.
- تعریف و مقایسه ادوات منطق ثابت (FPLD) و ادوات منطق برنامه پذیر (PLD) از دو دیدگاه وظیفه و اتصالات داخلی
- ادوات PLD -> یک بار برنامه پذیر -> چندبار برنامه پذیر -> برنامه ریزی درون-سیستمی (ISP)
- مثال: مدار تمام جمع کننده
- از نوع منطق ثابت است



Programmable Logic Devices (PLD)



• در مقابل ...



Programmable Logic Devices (PLD)



- در مقابل ...
- معادل مدار قبلی است (همان تعداد I/O)
- از نوع منطق برنامه پذیر است
- استفاده از آنتی فیوز
- برای پیاده‌سازی هر تابع بولی سه متغیره که حداکثر چهار مینترم داشته باشد، مناسب است
- دیدگاه یک آرایه AND قابل برنامه‌ریزی در ورودی و یک گیت OR ثابت در خروجی

Programmable Logic Devices (PLD)



- به طور خلاصه مزایا و معایب هر کدام از دو نوع منطق ...
- زمان لازم از مرحله طراحی تا مرحله تولید (انبوه) و ساخت ادوات منطق ثابت معمولاً بسیار بیشتر از زمان مربوط به ادوات منطق برنامه‌پذیر است (در حد چندماه یا حتی یکسال).
- فرآیند طراحی، اعمال تغییرات، و بررسی عملکرد مدار در شرایط واقعی در مورد ادوات منطق برنامه‌پذیر دارای زمان کوتاه‌تر و هزینه بسیار کمتری است. علت این امر، وجود نرم‌افزارهای طراحی نسبتاً ارزان و طبیعت برنامه‌پذیر بودن ادوات مذکور است.
- در مورد PLD ها، کاربران می‌توانند هر از چندگاهی مدار را مطابق میل یا نیاز خود تغییر داده یا تصحیح کنند. PLD ها در طول مراحل طراحی، انعطاف زیادی از خود نشان می‌دهند.
- ادوات منطق ثابت در کاربردهایی که نیاز به تولید انبوه است و نیز کاربردهایی که بالاترین میزان کارائی مورد نیاز است، مناسب‌تر هستند.

Programmable Logic Devices (PLD)



- ادوات (افزاره های) PLD:
- **معیارهای دسته بندی:** معماری داخلی، ظرفیت منطقی، و نوع (یا تکنولوژی) برنامه ریزی
- **انواع ادوات (افزاره های) PLD:**
 - ROM های برنامه پذیر (PROM)،
 - آرایه منطقی برنامه پذیر (یا PLA)،
 - منطق آرایه برنامه پذیر (یا PAL)،
 - منطق آرایه عمومی (یا GAL)،
 - ادوات منطقی برنامه پذیر پیچیده (یا CPLD)،
 - آرایه گیت های قابل برنامه ریزی در محل (یا FPGA).

Programmable Logic Devices (PLD)



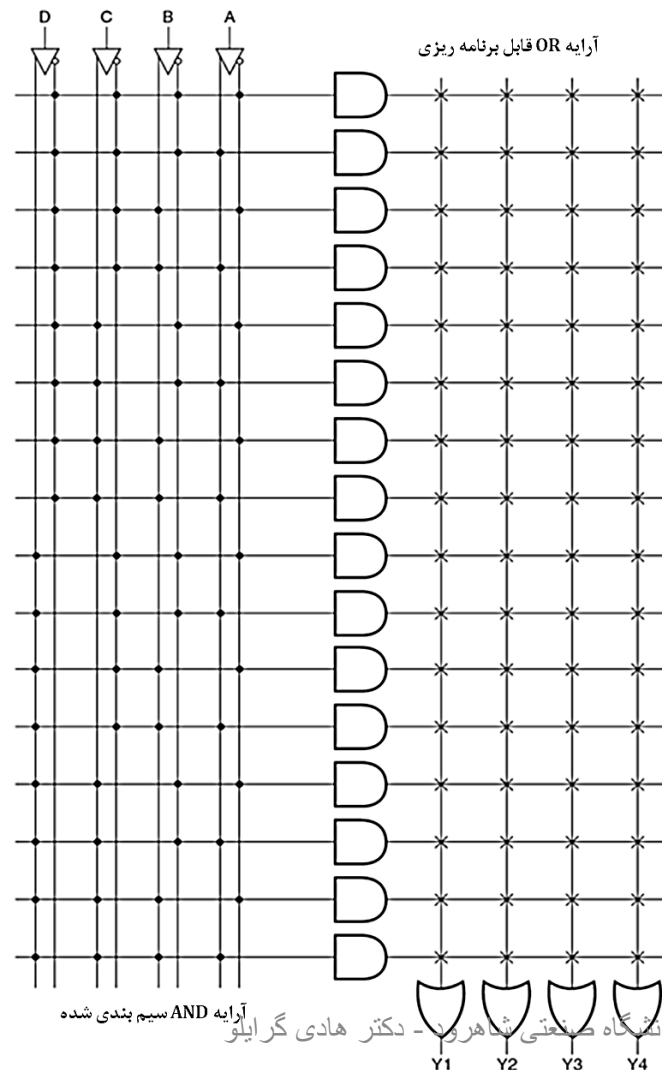
• ROM های برنامه پذیر (PROM):

- یک PROM در حقیقت یک «حافظه فقط خواندنی قابل برنامه ریزی» است
- اگر به آنها به عنوان «حافظه» نگاه شود، دارای n خط ورودی (به نام خطوط آدرس) و m خط خروجی (به نام خطوط داده) جهت ذخیره 2^n کلمه m بیتی می باشند.
- اگر آنها را به عنوان PLD مورد نظر قرار دهیم، می توانیم از آنها برای پیاده سازی m تابع ترکیبی مختلف استفاده کنیم که هر کدام، تابعی از n متغیر می باشند.
- شکل کلی یک ROM که شامل n ورودی و m خروجی باشد، دارای 2^n گیت AND سیم بندی شده در ورودی و m گیت OR برنامه پذیر در خروجی می باشد. هر گیت AND دارای n ورودی و هر گیت OR دارای 2^n ورودی می باشد. هر گیت OR قادر به تولید یک تابع بولی دلخواه n -متغیره است؛ بنابراین، این ROM تعمیم یافته را می توان برای تولید m تابع بولی دلخواه n -متغیره مورد استفاده قرار داد. **به مثال توجه شود**

Programmable Logic Devices (PLD)



• مثال:



Programmable Logic Devices (PLD)



- مثال...

- معماری داخلی یک PROM را نشان می‌دهد که چهار خط ورودی، یک آرایه سیم-بندی شده از 16 گیت AND، و یک آرایه برنامه‌پذیر از 4 گیت OR دارد.
- آی‌سی‌های PROM، EPROM، و EEPROM را می‌توان به کمک پروگرامرهای استاندارد برنامه‌ریزی کرد
- یکی از معایب اصلی PROMها عدم استفاده بهینه از ظرفیت منطقی آنها است.
- دیگر معایب PROMها عمدتاً شامل مصرف توان بالا و عدم اطمینان از صحت انجام مبادلات آسنکرون است. همچنین، این ادوات قادر به پیاده‌سازی مدارات ترتیبی نمی‌باشند زیرا درون خود شامل فلیپ فلاپ نمی‌باشند.

Programmable Logic Devices (PLD)



- **مثال:** فرض کنید بخواهیم تابع منطقی زیر را که دارای دو خروجی است به کمک PROM تولید کنیم

$$F_1(A, B, C) = \Sigma 0, 2 \quad (9.1)$$

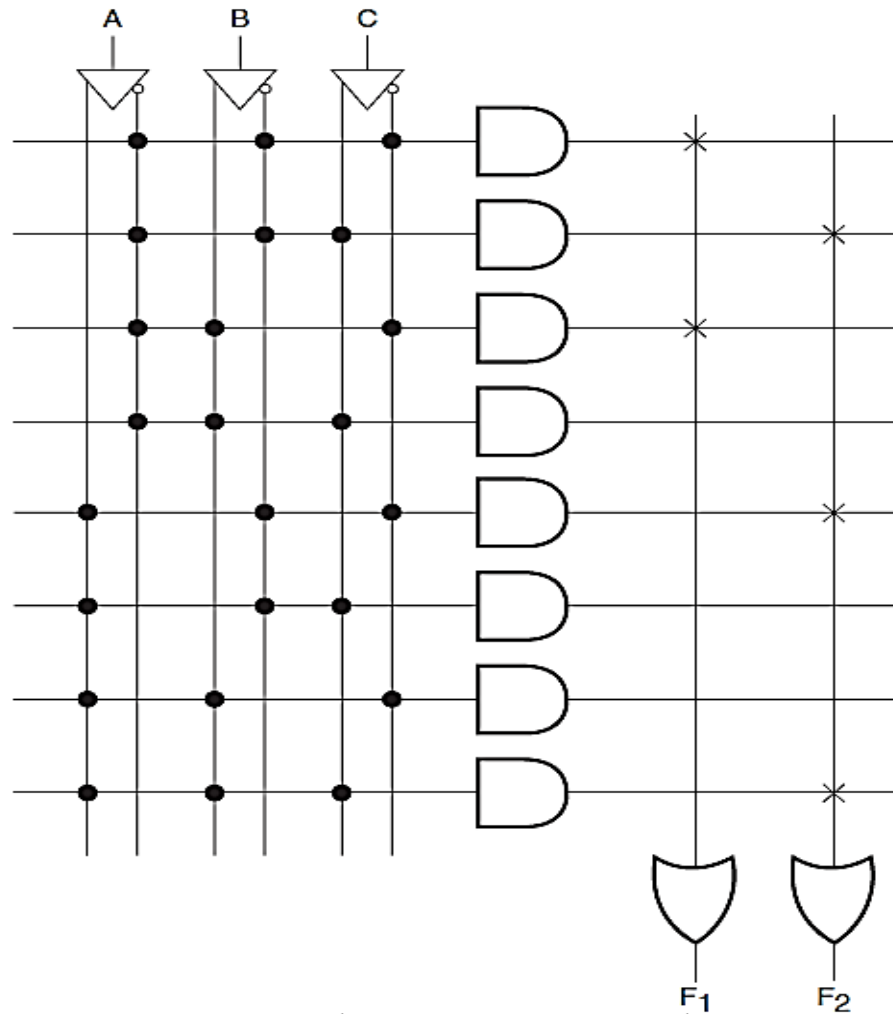
$$F_2(A, B, C) = \Sigma 1, 4, 7 \quad (9.2)$$

- **حل:** پیاده‌سازی این تابع بولی مستلزم استفاده از یک PROM با اندازه‌ی 8×2 است.

Programmable Logic Devices (PLD)



• مثال ...



Programmable Logic Devices (PLD)



- قابل توجه است که مثال فوق فقط جهت نمایش نحوه‌ی استفاده از PROM ها در پیاده‌سازی یک تابع بولی بوده و در عمل برای پیاده‌سازی توابع ساده‌ای مانند تابع فوق، از PROM ها استفاده نمی‌شود بلکه از این افزاره‌ها برای پیاده‌سازی توابع بسیار پیچیده استفاده می‌شود. به بیان دیگر می‌توان گفت PROM ها برای پیاده‌سازی توابعی که تعداد زیادی حالات «بی‌اهمیت» دارند مناسب نمی‌باشد؛ در این مواقع بهتر است از PLA ها یا PAL ها استفاده شود.

Programmable Logic Devices (PLD)



- **مثال:** اندازه‌ی یک PROM برای پیاده‌سازی هر یک از مدارات زیر را تعیین کنید:
 - (الف) یک ضرب‌کننده دودویی که دو عدد ورودی چهار بیتی را در هم ضرب می‌کند،
 - (ب) یک مالتی پلکسر 8 به 1 دوتایی با ورودی‌های انتخاب مشترک،
 - (ج) یک جمع‌کننده/تفریق‌کننده BCD تک رقمی دارای یک ورودی کنترلی جهت تعیین نوع عمل.

Programmable Logic Devices (PLD)



• حل:

– (الف) در اینجا تعداد ورودی‌ها برابر 8 است. نتیجه‌ی حاصلضرب نیز 8 بیتی است. بنابراین اندازه‌ی PROM برابر با $2^8 \times 8 = 256 \times 8$ می‌باشد.

– (ب) در این جا تعداد ورودی‌ها برابر است با $8 + 8 + 3 = 19$ (عدد 3 نماینده‌ی ورودی‌های کنترلی است). تعداد خروجی‌ها نیز برابر 2 است. بنابراین اندازه‌ی PROM برابر خواهد بود با $2^{19} \times 2 = 512K \times 2$.

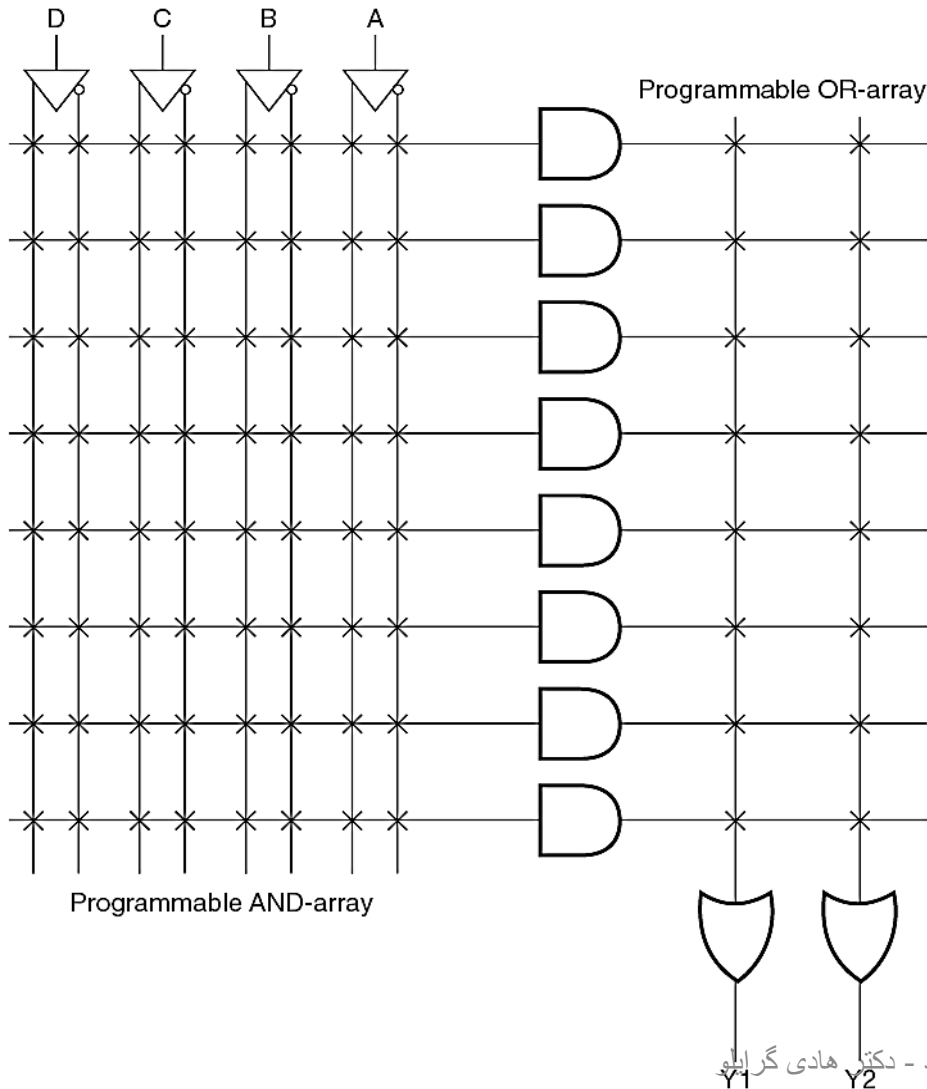
– (ج) تعداد ورودی‌ها برابر است با 4 (عدد اول) + 4 (عدد دوم) + 1 (رقم نقلی ورودی) + 1 (ورودی کنترلی) = 10. تعداد خروجی‌ها نیز برابر است با 4 (تعداد بیت‌های خروجی جمع یا تفریق) + 1 (رقم نقلی یا قرض) = 5. بنابراین اندازه‌ی PROM برابر خواهد بود با $2^{10} \times 5 = 1024 \times 5 = 1K \times 5$.

Programmable Logic Devices (PLD)



- آرایه منطقی برنامه پذیر (PLA):
 - یک PLA شامل یک آرایه AND قابل برنامه‌ریزی در ورودی و یک آرایه OR قابل برنامه‌ریزی در خروجی است.
 - موارد تفاوت آن با PROM ها :
 - به جای داشتن آرایه AND سیم بندی شده، شامل آرایه AND قابل برنامه ریزی است.
 - در یک PROM با m ورودی، تعداد گیت‌های AND همواره برابر 2^m می‌باشد اما در PLA، تعداد گیت‌های AND مربوط به آرایه قابل برنامه‌ریزی (در همان حالت داشتن m ورودی) معمولاً بسیار کمتر از 2^m است.
 - حسن: در PLA از ظرفیت منطقی موثرتر از PROM استفاده می‌شود.
 - عیب: مهمترین عیب PLA داشتن دو مجموعه از فیوزهای قابل برنامه‌ریزی است که موجب دشوارتر شدن ساخت PLA و همچنین برنامه‌ریزی و تست آن می‌شود.

Programmable Logic Devices (PLD)

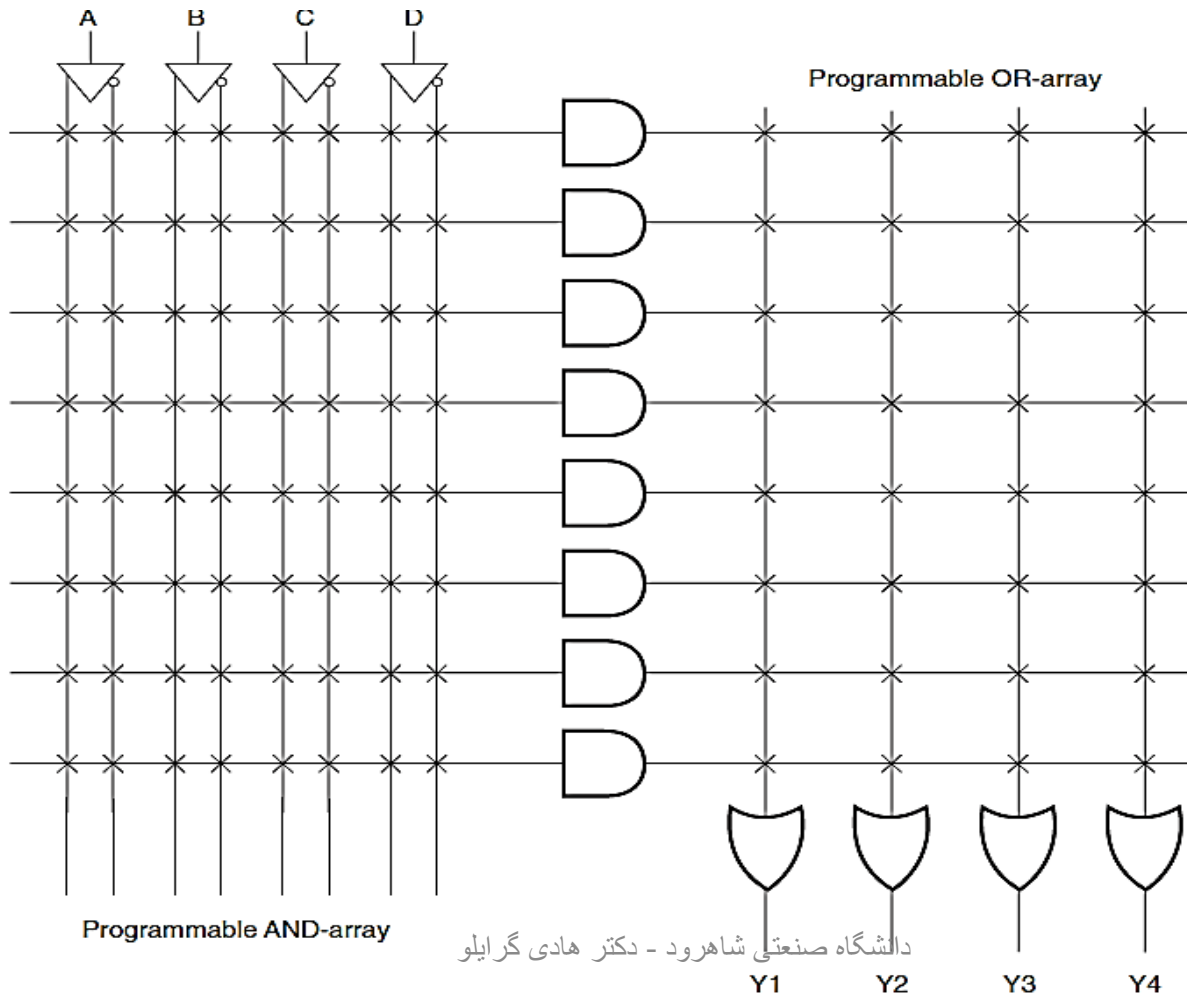


– **مثال:** یک PLA که شامل 4 خط ورودی،
یک آرایه قابل برنامه‌ریزی از 8 گیت AND
در ورودی، و یک آرایه قابل برنامه‌ریزی از
دو گیت OR در خروجی است

Programmable Logic Devices (PLD)



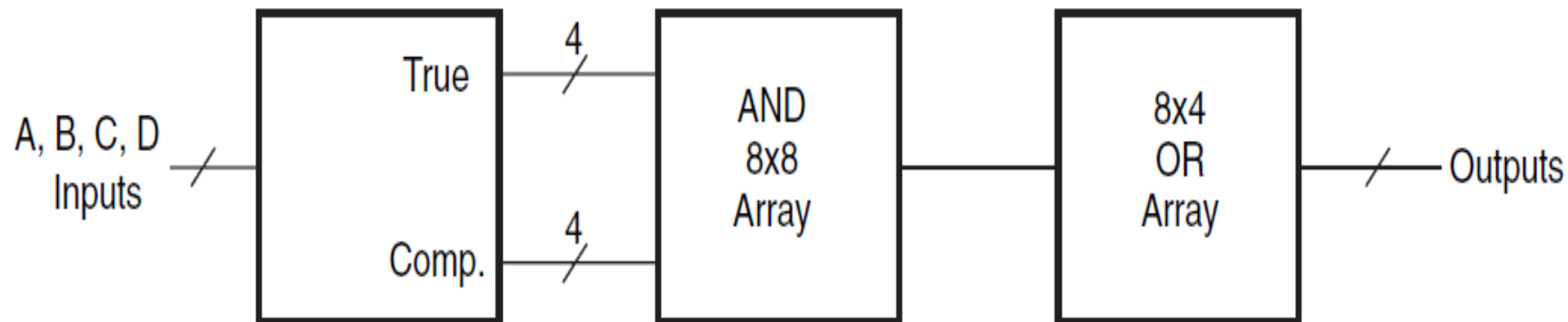
- معماری داخلی یک PLA شامل چهار خط ورودی، هشت خط حاصلضرب، و چهار خط خروجی



Programmable Logic Devices (PLD)



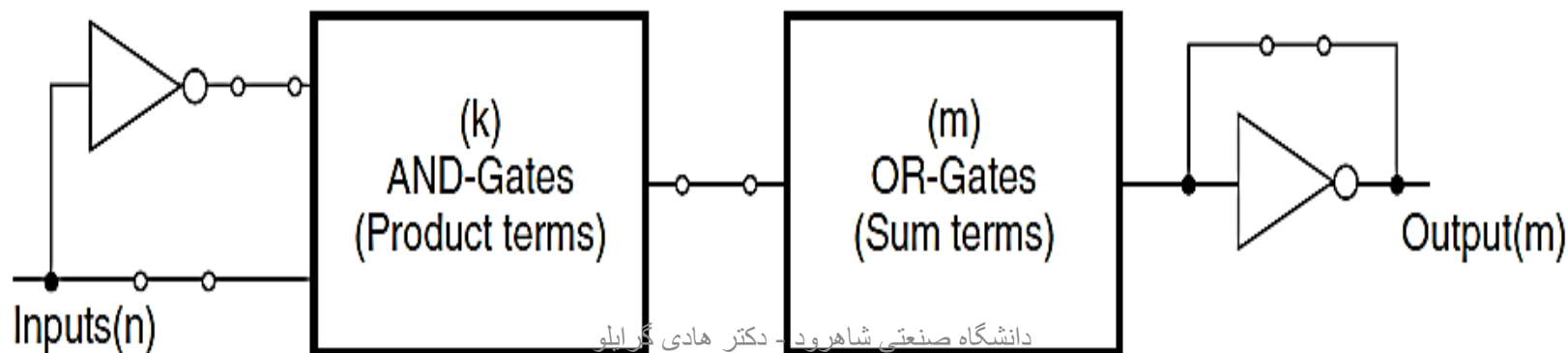
- این شکل را می توان به صورت شماتیکی مطابق با شکل زیر نمایش داد.



Programmable Logic Devices (PLD)



- افزاره‌های PLA معمولاً در خروجی گیت‌های OR دارای معکوس‌کننده (گیت NOT) هستند تا علاوه بر تولید توابع بولی به شکل (یا قالب) AND-OR بتوانند قالب AND-INVERT را نیز تولید کنند.
- دیاگرام بلوکی کلی از یک افزاره‌ی PLA که دارای n ورودی، m خروجی، و k جمله حاصلضرب (مینترم) است. پارامتر n تعداد متغیرهای ورودی، m تعداد گیت‌های AND و n تعداد گیت‌های OR را نشان می‌دهند. تعداد ورودی‌های هر گیت OR برابر با k و تعداد ورودی‌های هر گیت AND برابر با 2^n می‌باشد. تعداد اتصالات قابل برنامه‌ریزی در PLA برابر با $2kn + km$ می‌باشد حال آن که اگر یک ROM با همین تعداد خط ورودی و همین تعداد خط خروجی داشته باشیم دارای $2^n \times m$ اتصال قابل برنامه‌ریزی خواهد بود.



Programmable Logic Devices (PLD)



- در هنگام استفاده از یک PLA بهتر است ابتدا تابع بولی مورد نظرمان را تا حد امکان ساده کنیم طوری که کمترین تعداد مینترم ها را در بر بگیرد و بنابراین در تعداد گیت های AND صرفه جویی شود. آن نوع ساده سازی که منجر به کاهش تعداد متغیرها در هر مینترم شود، در اینجا سودمند نخواهد بود زیرا در PLA در هر حال برای هر گیت AND از تمام متغیرهای ورودی استفاده شده است و ناچاراً باید از تمامی متغیرها در هر جملهی مینترم استفاده شود.
- **مثال:** نحوه ی استفاده و برنامه ریزی یک PROM و یک PLA را جهت پیاده سازی تابع بولی مربوط به یک تمام جمع کننده ی دودویی نشان دهید.
- **پاسخ:** جدول صحت یک تمام جمع کننده و عبارت بولی متناظر با خروجی های جمع (S) و رقم نقلی خروجی (C_0)

Programmable Logic Devices (PLD)



• مثال ...

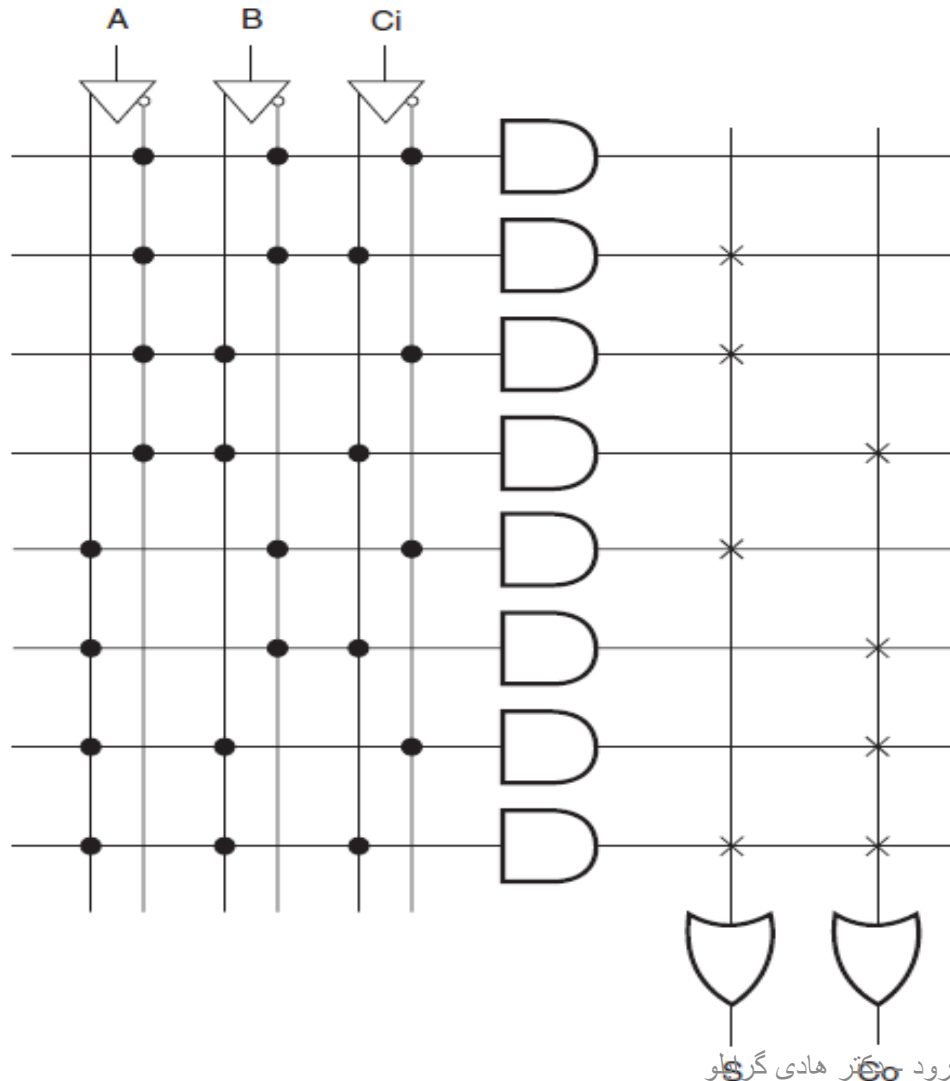
$$S = \Sigma 1, 2, 4, 7 \quad (9.3)$$

$$C_o = \Sigma 3, 5, 6, 7 \quad (9.4)$$

Table 9.1 Truth table for example 9.2.

A	B	Carry-in (C_i)	Sum (S)	Carry-out (C_o)
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

Programmable Logic Devices (PLD)



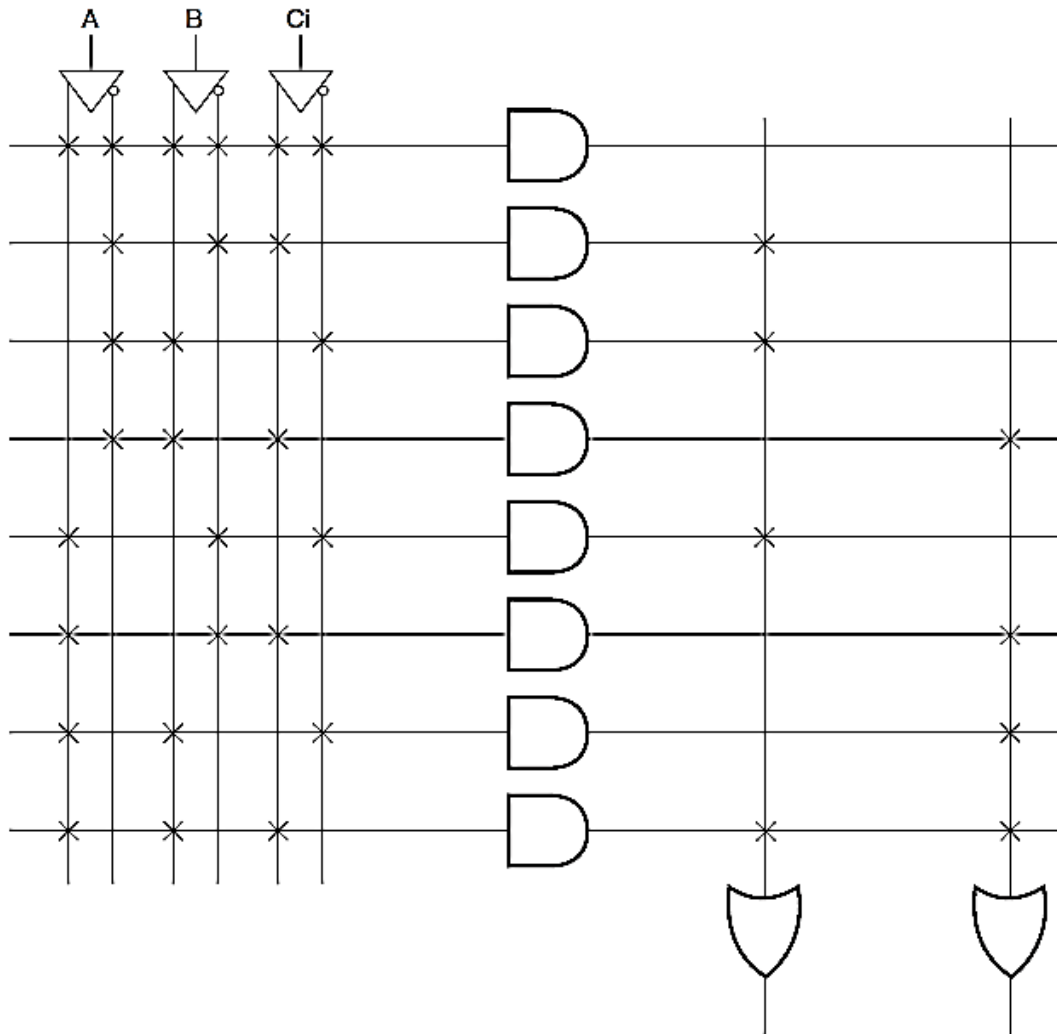
- پیاده سازی روی یک PROM با اندازه 8×2

Programmable Logic Devices (PLD)



- در مورد استفاده از PLA، اگر بخواهیم ساده‌سازی انجام دهیم ملاحظه خواهیم کرد که عبارت تابع بولی S را نمی‌توان بیش از وضع موجود ساده‌سازی کرد اما در مورد تابع بولی C_0 می‌توان به سه مینترم (در مقایسه با وضع موجود که چهار مینترم دارد) کاهش داد. این کاهش همراه با کاهش تعداد متغیرها در مینترم‌ها خواهد بود که عملاً این کاهش تعداد متغیرها در پیاده‌سازی روی PLA هیچ سودمندی به همراه نخواهد داشت (زیرا موجب کاهش تعداد گیت‌های AND نشده است).

Programmable Logic Devices (PLD)



• پیاده سازی روی PLA

Programmable Logic Devices (PLD)



- **مثال:** دو عدد دودویی A_1A_0 و B_1B_0 در دست داریم. یک مقایسه‌گر اندازه را طراحی و روی یک PLA مناسب پیاده‌سازی نمایید. در این مدار، چهار خروجی باید نسبت به عدد A_1A_0 تولید شوند: «نامساوی»، «مساوی»، «کوچکتر» و «بزرگتر».
- **پاسخ:** جدول صحت مدار به صورت زیر می‌باشند:

Table 9.2 Function table for example 9.3.

A_1	A_0	B_1	B_0	Output 1 $A=B$	Output 2 $A \neq B$	Output 3 $A < B$	Output 4 $A > B$
0	0	0	0	1	0	0	0
0	0	0	1	0	1	1	0
0	0	1	0	0	1	1	0
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	1	0	0	0
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	0	1
1	0	1	0	1	0	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	1	0	1
1	1	0	1	0	1	0	1
1	1	1	0	0	1	0	1
1	1	1	1	1	0	0	0

Programmable Logic Devices (PLD)



- پاسخ... عبارات بولی چهار خروجی به صورت زیر می باشند:

$$\text{Output 1 (equal to)} = \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} \quad (9.5)$$

Output 2 (not equal to)

$$\begin{aligned} = & \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot B_0 + \overline{A_1} \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} + \overline{A_1} \cdot \overline{A_0} \cdot B_1 \cdot B_0 + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot B_1 \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} \\ & + A_1 \cdot \overline{A_0} \cdot \overline{B_1} \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot B_0 + A_1 \cdot A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot \overline{B_0} \end{aligned} \quad (9.6)$$

Output 3 (less than)

$$= \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot B_0 + \overline{A_1} \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} + \overline{A_1} \cdot \overline{A_0} \cdot B_1 \cdot B_0 + \overline{A_1} \cdot A_0 \cdot B_1 \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot B_0 \quad (9.7)$$

Output 4 (greater than)

$$= \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot \overline{A_0} \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot \overline{B_0} \quad (9.8)$$

Programmable Logic Devices (PLD)



• پاسخ... جداول کارنو به صورت زیر می باشند:

A ₁ A ₀		00	01	11	10
B ₁ B ₀	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

(a)

A ₁ A ₀		00	01	11	10
B ₁ B ₀	00	0	1	1	1
	01	1	0	1	1
	11	1	1	0	1
	10	1	1	1	0

(b)

A ₁ A ₀		00	01	11	10
B ₁ B ₀	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	10	1	1	0	0

(c)

A ₁ A ₀		00	01	11	10
B ₁ B ₀	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

(d)

Programmable Logic Devices (PLD)



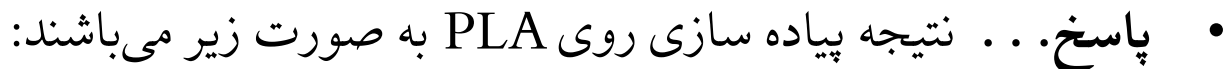
• پاسخ... عبارات ساده شده به صورت زیر می باشند:

$$\text{Output 1 (equal to)} = \overline{A_1}.\overline{A_0}.\overline{B_1}.\overline{B_0} + \overline{A_1}.A_0.\overline{B_1}.B_0 + A_1.A_0.B_1.B_0 + A_1.\overline{A_0}.B_1.\overline{B_0} \quad (9.9)$$

$$\text{Output 2 (not equal to)} = \overline{A_1}.B_1 + A_1.\overline{B_1} + \overline{A_0}.B_0 + A_0.\overline{B_0} \quad (9.10)$$

$$\text{Output 3 (less than)} = \overline{A_1}.B_1 + \overline{A_1}.\overline{A_0}.B_0 + \overline{A_0}.B_1.B_0 \quad (9.11)$$

$$\text{Output 4 (Greater than)} = A_1.\overline{B_1} + A_1.A_0.\overline{B_0} + A_0.\overline{B_1}.\overline{B_0} \quad (9.12)$$



- بررسی عبارات مینیمم شده نشان می‌دهد در کل تعداد دوازده عبارت «مختلف» (یا حاصلضرب) وجود دارد لذا استفاده از یک PLA با دوازده گیت AND برای پیاده‌سازی این مدار کافی خواهد بود. همچنین به دلیل وجود چهار تابع، تنها نیاز به چهار گیت OR در خروجی داریم.

Programmable Logic Devices (PLD)

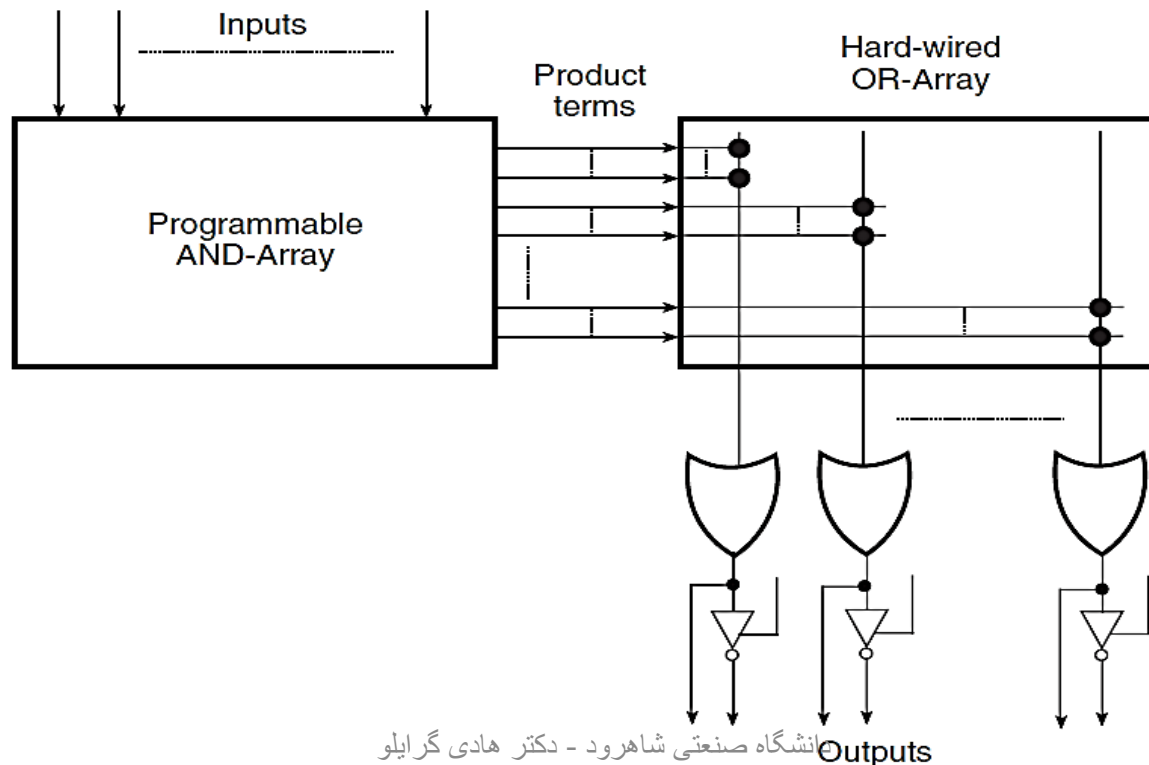


- منطق آرایه برنامه پذیر (PAL):
 - یک PAL شامل یک آرایه AND قابل برنامه‌ریزی در ورودی و یک آرایه OR ثابت در خروجی می‌باشد.
 - در PAL نیز، مشابه PLA، تعداد گیت‌های AND قابل برنامه‌ریزی معمولاً کمتر از تعداد لازم برای تولید تمام مینترم‌های ممکن از متغیرهای ورودی است. در PAL، آرایه OR ثابت بوده و خروجی‌های گیت‌های AND بطور مساوی بین گیت‌های OR تقسیم می‌شوند.
 - علت اصلی ثابت شدن آرایه‌ی OR در این افزاره نسبت به افزاره‌ی PAL این است که در بسیاری از کاربردهای از تمام اتصالات برنامه‌پذیر موجود در آرایه‌ی OR مربوط به افزاره‌ی PLA استفاده نمی‌شود لذا به منظور استفاده‌ی موثرتر از منابع افزاره، آرایه‌ی OR ثابت طراحی شده و نام آن، PAL انتخاب شد.
 - این نام، علامت تجاری ثبت شده توسط شرکت Advanced Micro Devices Inc. می‌باشد. البته افزاره‌های PAL قابلیت انعطاف کمتری نسبت به افزاره‌های PLA دارند.

Programmable Logic Devices (PLD)



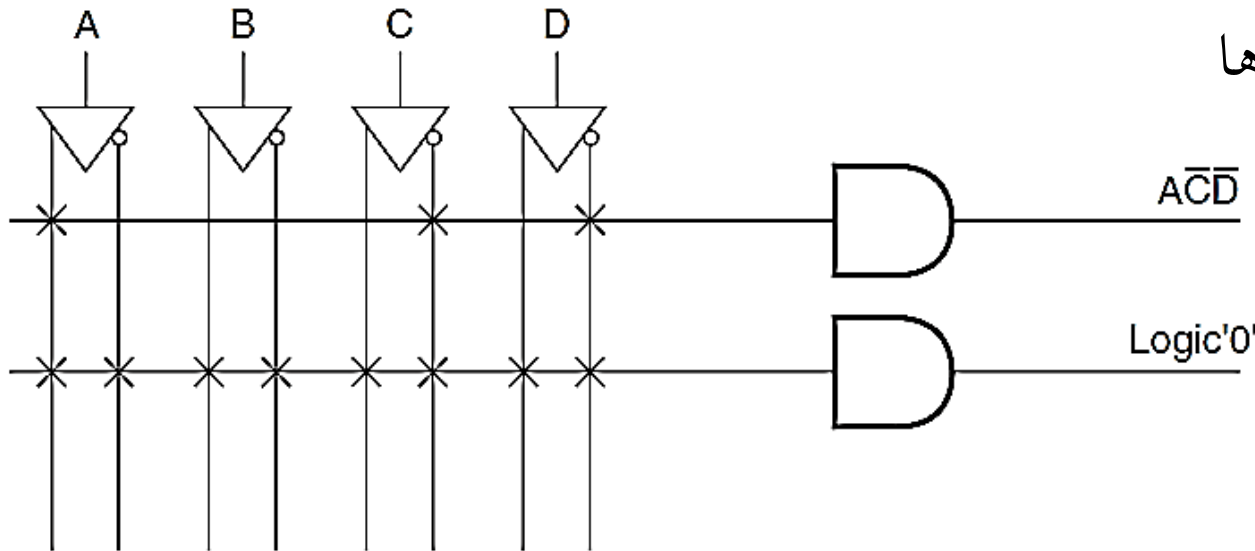
- طرح بلوکی از معماری کلی PAL
 - هر گیت AND یک مینترم مشخص (تعیین شده توسط کاربر) که شامل ترکیب خاصی از متغیرها و متمم‌هایشان است، تولید می‌کند.



Programmable Logic Devices (PLD)



- مثالی از تولید مینترمها



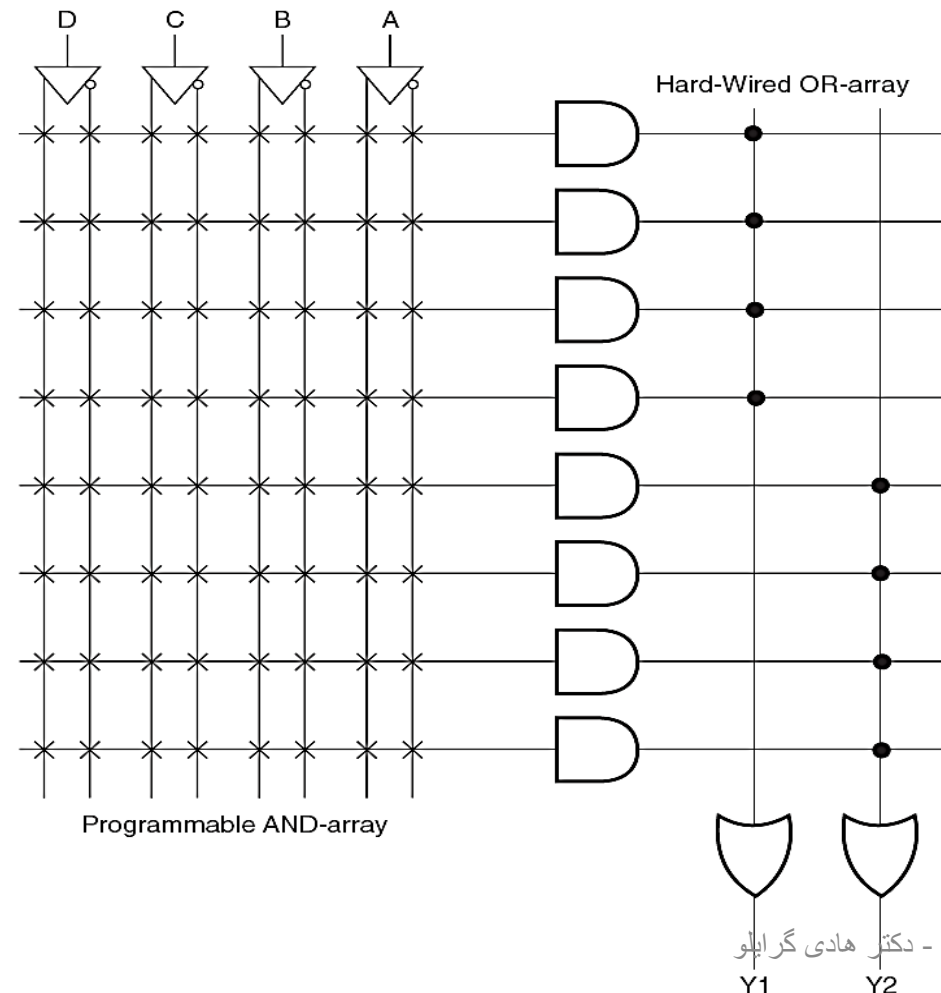
- هر گیت OR تنها از طریق «زیرمجموعه‌ای» از آرایه‌های AND موجود تغذیه می‌شود و نه تمام آرایه‌های موجود. بنابراین، هر تابع بولی که توسط یکی از گیت‌های OR تولید می‌شود حداکثر تعداد مینترم‌های مشخص و محدودی خواهد داشت که این تعداد توسط تعداد آرایه‌های AND متصل به (یا تغذیه کننده‌ی) آن گیت OR تعیین می‌شود.

Programmable Logic Devices (PLD)



– **مثال:** یک PAL شامل 4 خط ورودی، یک آرایه از 8 گیت AND در ورودی، و 2 گیت OR در خروجی.

– با مقایسه دو شکل اخیر ملاحظه می شود که در PAL، هر عبارت بولی خروجی شامل تعداد مینترمهای کمتری نسبت به PLA است. بنابراین، یک PAL حالت خاص و ساده-شده‌ای از یک PLA می باشد.



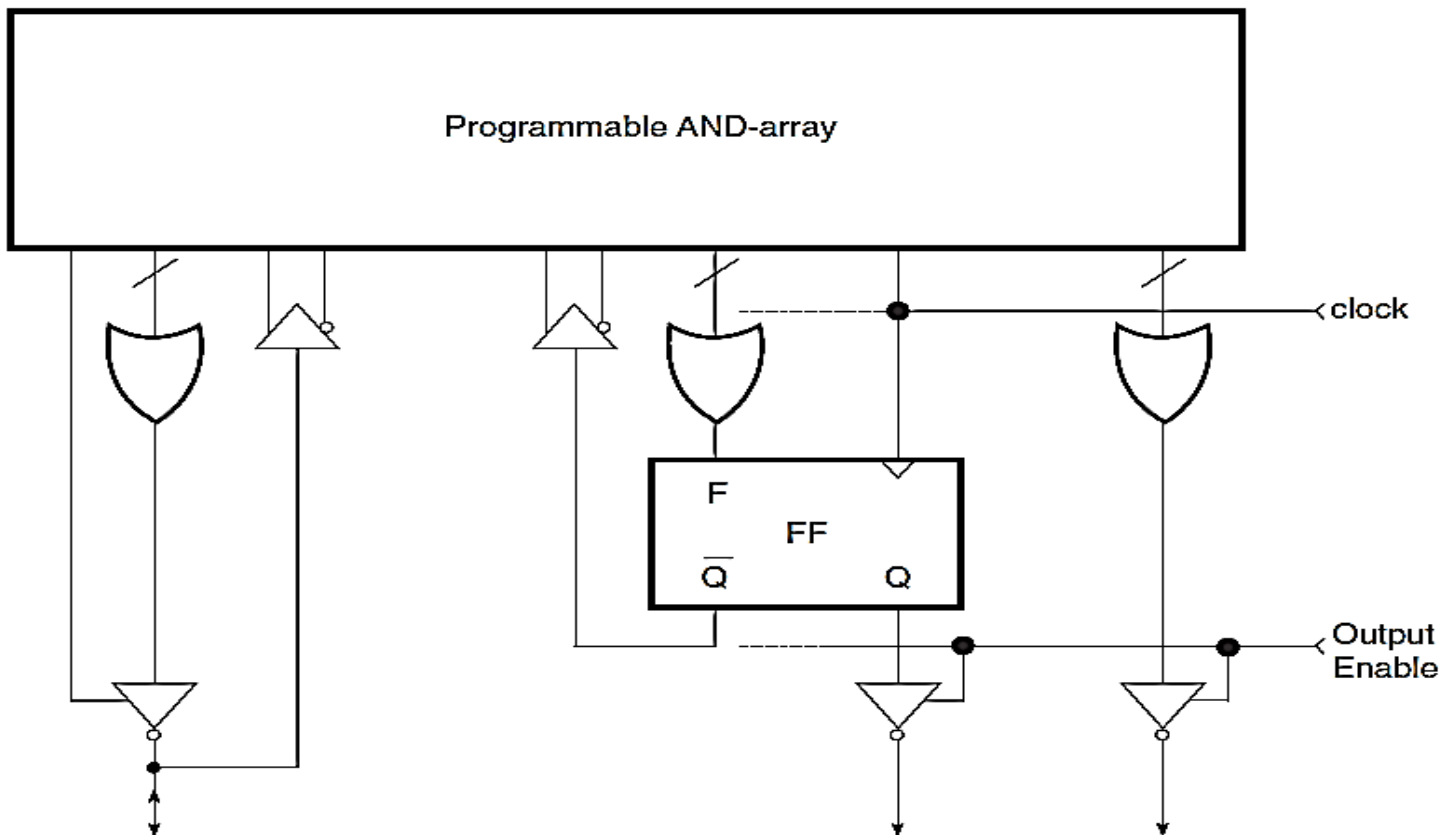
Programmable Logic Devices (PLD)



– در کاربردهای عملی، افزاره‌های PAL خروجی‌های متنوعی تولید می‌کنند.

- یکی همان خروجی OR و خروجی NOR است که گفته شد.
- دیگری امکان تولید خروجی رجیستر شده است. در این حالت، خروجی گیت OR به ورودی D یک فلیپ فلاپ نوع D حساس به لبه‌ی بالارونده یا پایین رونده‌ی کلاک ورودی خود متصل شده است.
- یک نوع امکان دیگر برای خروجی، دوجهته بودن خروجی (یا پین) PAL است. در این حالت، پین مورد نظر از PAL هم به صورت ورودی و هم به صورت خروجی قابل استفاده خواهد بود. با این قابلیت، امکان پس‌خورد یا اعمال مجدد یک مینترم به آرایه‌ی AND قابل برنامه‌ریزی وجود دارد. بدین ترتیب امکان تولید توابع چند-خروجی که برخی مینترم‌های مشخص را به اشتراک و استفاده‌ی مجدد می‌گذارند، وجود خواهد داشت.
- برخی افزاره‌های PAL در خروجی خود پس از هر گیت OR دارای گیت XOR می‌باشند. یکی از ورودی‌های این گیت‌های XOR به گونه‌ای قابل برنامه‌ریزی توسط کاربر است که بتوان یک بافر معکوس‌کننده یا بافر غیرمعکوس‌کننده و یا یک گیت XOR معمولی دو-ورودی در اختیار داشت. این قابلیت در برخی کاربردها مانند کاربردهای مربوط به بیت توازن و کاربردهای حسابی کاربرد دارد.

Programmable Logic Devices (PLD)



Programmable Logic Devices (PLD)



– مثال: جدول عملکرد یک معکوس کننده داده شده است. عبارت بولی هر یک از چهار خروجی (P، Q، R و S) را نوشته و آنها را به کمک جدول کارنو ساده نمایید. سپس این معکوس کننده را روی یک PAL پیاده سازی نمایید.

Table 9.4 Function table in example 9.4.

A	B	C	D	P	Q	R	S
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

Programmable Logic Devices (PLD)



- حل:

AB \ CD	00	01	11	10
00	0	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

(a)

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	0	0	X	X
10	0	0	X	X

(b)

AB \ CD	00	01	11	10
00	0	1	X	0
01	0	1	X	0
11	1	1	X	X
10	1	1	X	X

(c)

AB \ CD	00	01	11	10
00	0	0	X	1
01	1	0	X	0
11	0	1	X	X
10	1	0	X	X

(d)

Programmable Logic Devices (PLD)



– عبارات قبل از ساده سازی کارنو

$$P = \bar{A}.B.\bar{C}.D + \bar{A}.B.C.\bar{D} + \bar{A}.B.C.D + A.\bar{B}.\bar{C}.\bar{D} + A.\bar{B}.\bar{C}.D \quad (9.13)$$

$$Q = \bar{A}.B.\bar{C}.\bar{D} + \bar{A}.B.\bar{C}.D \quad (9.14)$$

$$R = \bar{A}.\bar{B}.C.\bar{D} + \bar{A}.\bar{B}.C.D + \bar{A}.B.\bar{C}.\bar{D} + \bar{A}.B.\bar{C}.D + \bar{A}.B.C.\bar{D} + \bar{A}.B.C.D \quad (9.15)$$

$$S = \bar{A}.\bar{B}.\bar{C}.D + \bar{A}.\bar{B}.C.\bar{D} + \bar{A}.B.C.D + A.\bar{B}.\bar{C}.\bar{D} \quad (9.16)$$

– عبارات پس از ساده سازی کارنو

$$P = B.D + B.C + A \quad (9.17)$$

$$Q = B.\bar{C} \quad (9.18)$$

$$R = B + C \quad (9.19)$$

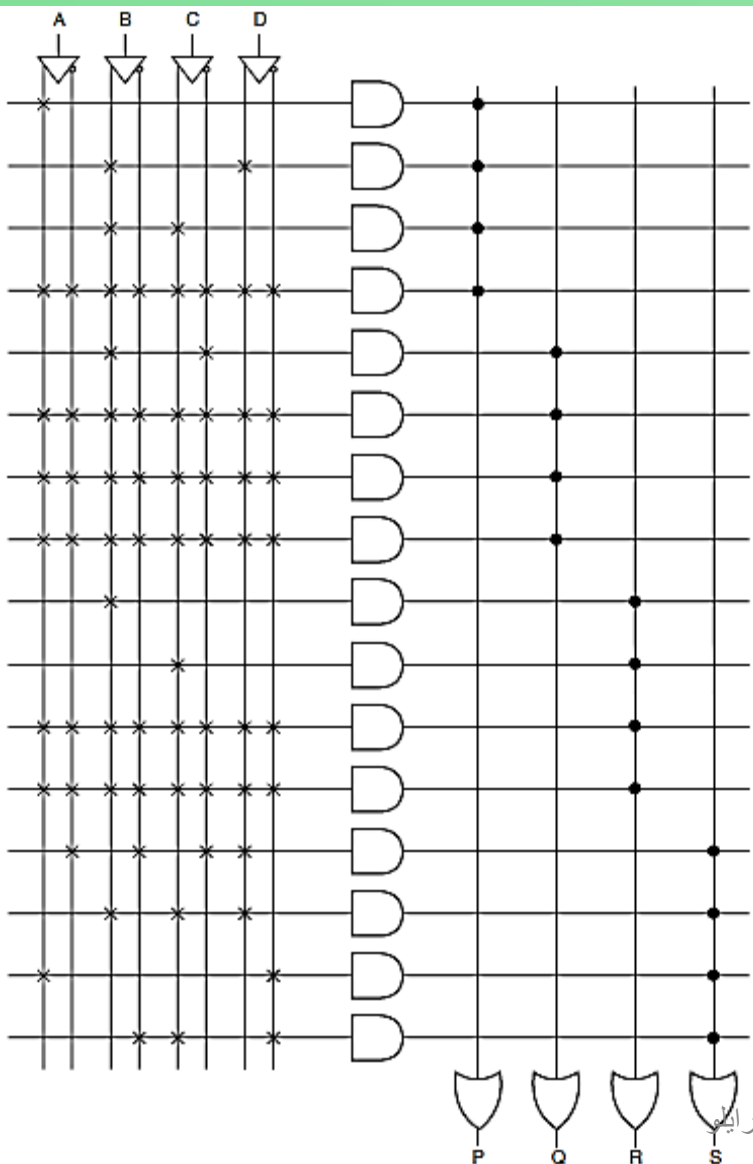
$$S = \bar{A}.\bar{B}.\bar{C}.D + B.C.D + A.\bar{D} + \bar{B}.C.\bar{D} \quad (9.20)$$

Programmable Logic Devices (PLD)



- حال، قدم بعدی انتخاب یک افزارهی PAL مناسب می باشد. چون چهار خروجی داریم پس افزارهی PAL باید حداقل چهار گیت OR در خروجی خود باشد. با توجه به این که (1) هر گیت OR تنها به تعداد محدودی آرایه های AND برنامه پذیرسیم بندی شده است؛ (2) یکی از توابع بولی دارای چهار جمله ی حاصلضرب است (بیشترین تعداد حاصلضرب موجود)؛ و (3) آرایه ی گیت های AND به تعداد مساوی بیم گیت های OR موجود تقسیم می شود؛ بنابراین، آرایه ی گیت های AND می بایست حداقل دارای 16 گیت AND باشد (چهار تابع، هر تابع حداکثر چهار جمله ی حاصلضرب پس $4 \times 4 = 16$).
- از آنجا که چهار متغیر ورودی داریم پس هر گیت AND باید دارای هشت ورودی باشد تا بتواند متغیرهای ورودی را به همراه متمم هایشان دریافت و مینترم مناسب را تولید کند.
- بنابراین، در مجموع می توان گفت نیاز به یک افزارهی PAL داریم که دارای هشت ورودی، 16 گیت AND در آرایه ی AND برنامه پذیر و چهار گیت OR در خروجی خود داشته باشد و در ضمن، هر گیت OR دارای چهار ورودی باشد.

Programmable Logic Devices (PLD)



– معماری افزاره‌ی PLA برنامه‌ریزی شده
جهت پیاده‌سازی توابع خروجی

Programmable Logic Devices (PLD)



- ملاحظه می‌شود که خروجی P تنها سه جمله‌ی حاصلضرب دارد پس باید ورودی چهارم به گیت OR مربوط به این خروجی را برابر با «صفر» منطقی قرار دهیم. برای انجام این کار کافی است تمام ورودی‌ها و متمم‌هایشان را در ورودی گیت AND مربوطه را به این گیت AND وارد کنیم تا حاصل AND این هشت ورودی، «صفر» منطقی ایجاد کند.
- به طریق مشابه، هر جا منطق «صفر» لازم داشته باشیم از این روش استفاده می‌کنیم.

Programmable Logic Devices (PLD)

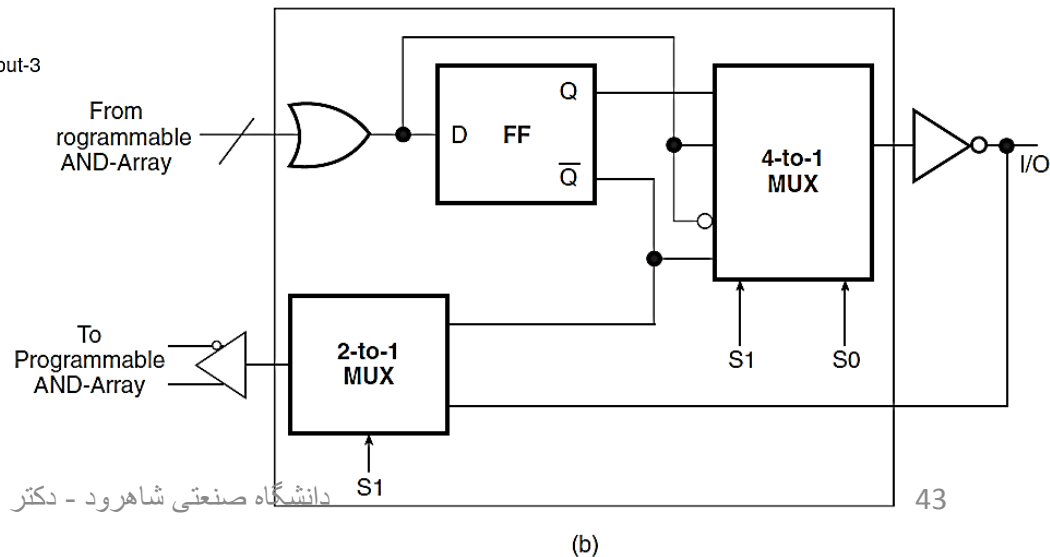
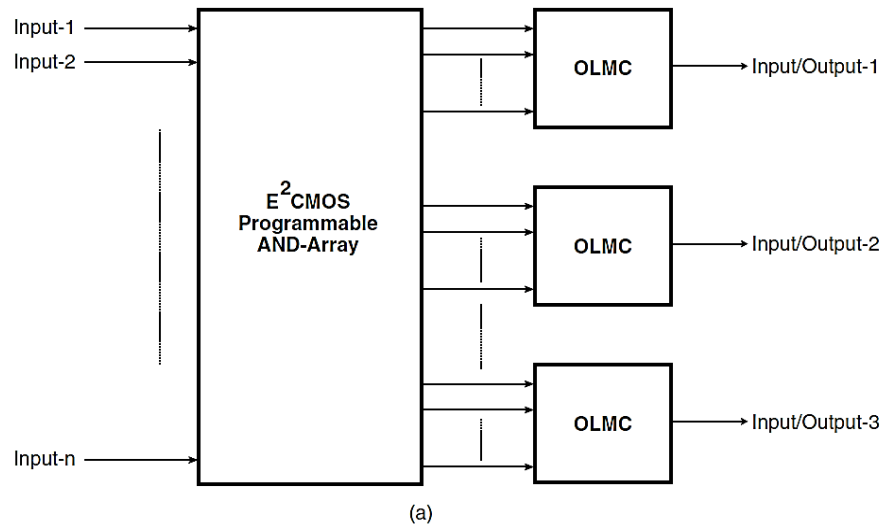


- منطق آرایه عمومی (GAL):
 - یک GAL مشابه PAL بوده و توسط شرکت Lattice Semiconductor اختراع شده است.
 - مهمترین مشخصات یک افزاره‌ی GAL عبارتند از: (1) آرایه‌ی AND با قابلیت برنامه‌ریزی مجدد، (2) آرایه‌ی OR ثابت، (3) منطق خروجی با قابلیت برنامه‌ریزی مجدد.
 - **تفاوت** آن با PAL در این است که (1) برخلاف PAL که آرایه‌ی AND تنها یک بار قابل برنامه‌ریزی است، در اینجا می‌توان به هر تعداد دلخواه آرایه‌ی AND را برنامه‌ریزی کرد. این قابلیت به کمک استفاده از سلول‌های EEPROM در ذخیره‌ی الگوی برنامه‌ریزی امکان‌پذیر شده است. همچنین (2) خروجی منطقی آن نیز مجدداً قابل برنامه‌ریزی است.
 - **حسن** بزرگ آن در این است که هرگونه تغییر و اصلاح قابل اعمال مجدد می‌باشد. یک افزاره مشابه دیگر، PEEL نام دارد که محصول شرکت ICT است.

Programmable Logic Devices (PLD)



- افزاره‌های GAL از ماکروسل‌های منطقی خروجی (OLMC) در خروجی خود استفاده کرده‌اند. این منطق به این افزاره‌ها اجازه می‌دهد دو نوع خروجی تولید کنند: خروجی ترکیبی و خروجی رجیستر شده



Programmable Logic Devices (PLD)



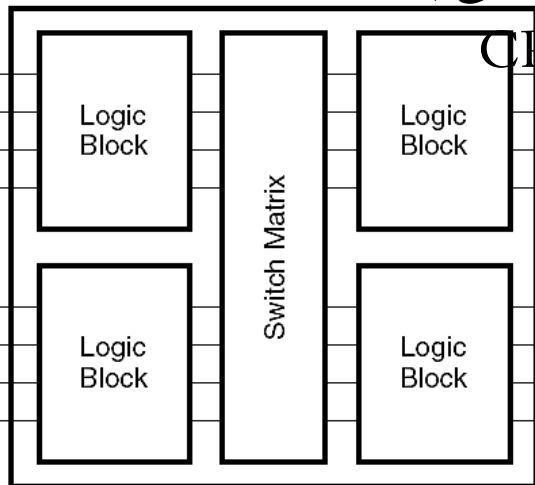
- نمونه OLMC نمایش داده شده در این شکل قادر به پیکره‌بندی جهت تولید چهار نوع خروجی، بسته به ورودی‌های انتخاب‌گر خود، می‌باشد:
 - حالت $S_1S_0=00$: حالت رجیستر شده با خروجی از نوع فعال پایین
 - حالت $S_1S_0=01$: حالت رجیستر شده با خروجی از نوع فعال بالا
 - حالت $S_1S_0=10$: حالت ترکیبی با خروجی از نوع فعال پایین
 - حالت $S_1S_0=11$: حالت ترکیبی با خروجی از نوع فعال بالا
- از بین چهار خروجی مالتی پلکسر 4 به 1، دو ورودی آن خود خروجی از نوع ترکیبی و دو ورودی دیگر، خود خروجی از نوع رجیستر شده هستند. همچنین، از بین دو خروجی ترکیبی یک فعال بالا و دیگری فعال پایین می‌باشد. همین حالت در مورد دو خروجی رجیستر شده نیز وجود دارد. از بین این چهار نوع خروجی بررسی شده، یکی به خروجی مالتی پلکسر راه پیدا می‌کند. انتخاب برعهده‌ی ورودی انتخاب‌گر مالتی پلکسر است. مالتی پلکسر 2 به 1 استفاده شده در شکل جهت تضمین انجام پس‌خورد از خروجی به ورودی آرایه‌ی AND برنامه‌پذیر می‌باشد.

Programmable Logic Devices (PLD)



- ادوات منطقی قابل برنامه‌ریزی پیچیده (CPLD):

- اغلب ادواتی نظیر PLA، PAL، GAL و دیگر ادوات شبیه به PAL را در یک گروه به نام «ادوات منطقی قابل برنامه‌ریزی ساده» یا SPLD دسته‌بندی می‌کنند تا آنها را از دسته دیگری که ساختار بسیار پیچیده‌تری نسبت به SPLD ها دارند، متمایز سازند.
- یک CPLD ممکن است مداری معادل چندین PAL داشته باشد که به کمک تعدادی اتصالات قابل برنامه‌ریزی به یکدیگر متصل شده باشند. مرتبه پیچیدگی یک PAL عموماً در حد چند صد گیت منطقی است، اما در مقابل، یک CPLD ممکن است دارای مرتبه پیچیدگی حدود دهها (و صدها) هزار گیت منطقی باشد.



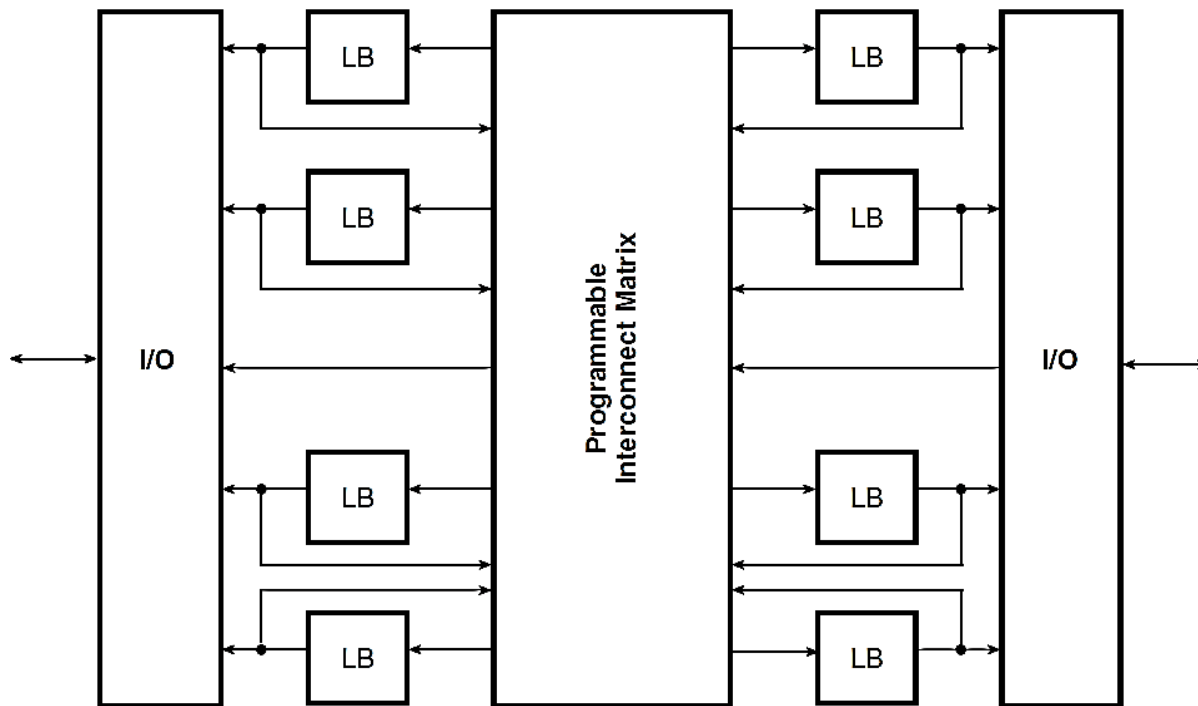
- یک طرح متداول از ساختار (یا معماری) داخلی یک CPLD
- هر یک از چهار بلوک منطقی معادل یک SPLD است
- ماتریس سوئیچ برای اتصال بین بلوکهای منطقی

Programmable Logic Devices (PLD)



• شکل زیر معماری یک نمونه CPLD را نشان می‌دهد:

– ماتریس اتصالات قابل برنامه‌ریزی قادر به ایجاد اتصال بین ورودی/خروجی هر بلوک منطقی به هر بلوک منطقی دیگر است. همچنین پین‌های ورودی/خروجی مستقیماً هم به ماتریس اتصالات راه دارند و هم به بلوکهای منطقی.



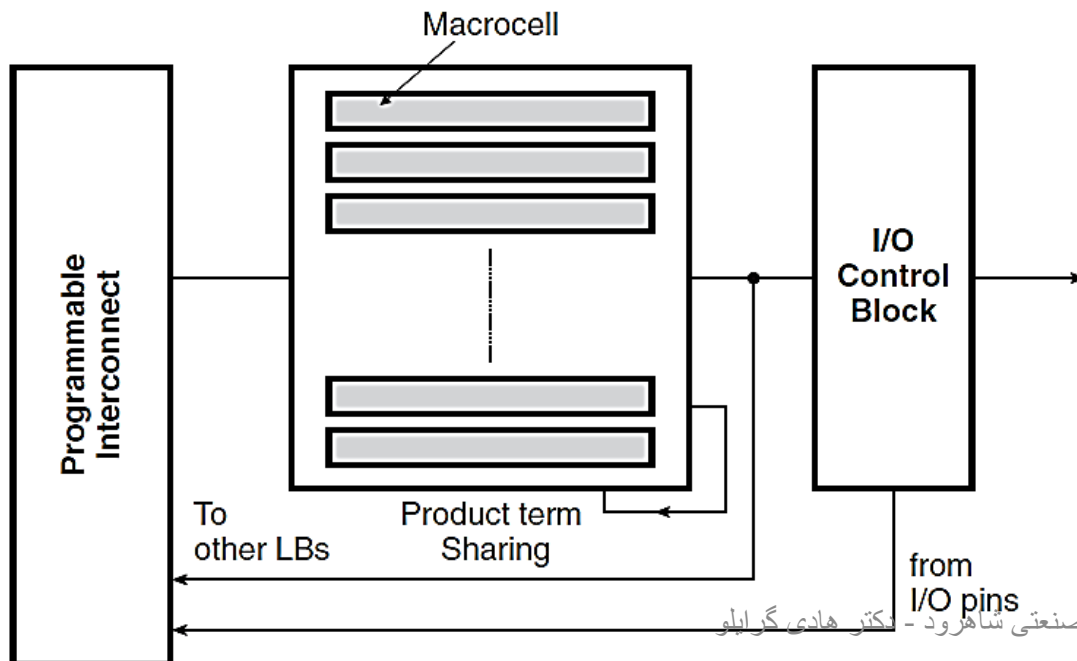
دانشگاه صنعتی شاهرود - دکتر هادی گرایلو
LB : Logic B lock

Programmable Logic Devices (PLD)



• بلوک‌های منطقی می‌توانند شامل واحدهای منطقی کوچکتری به نام «ماکروسل یا درشت-سلول» باشد.

- شکل زیر ساختار یک بلوک منطقی را به همراه اتصالاتی که با ماتریس اتصالات قابل برنامه‌ریزی و بلوک I/O دارد، نمایش می‌دهد.
- هر ماکروسل زیرمجموعه‌ای از یک بلوک منطقی شبیه به افزاره‌های PLD است.
- نوارهای افقی خاکستری داخل بلوک منطقی شامل آرایه‌ای از ماکروسل‌ها می‌باشند.

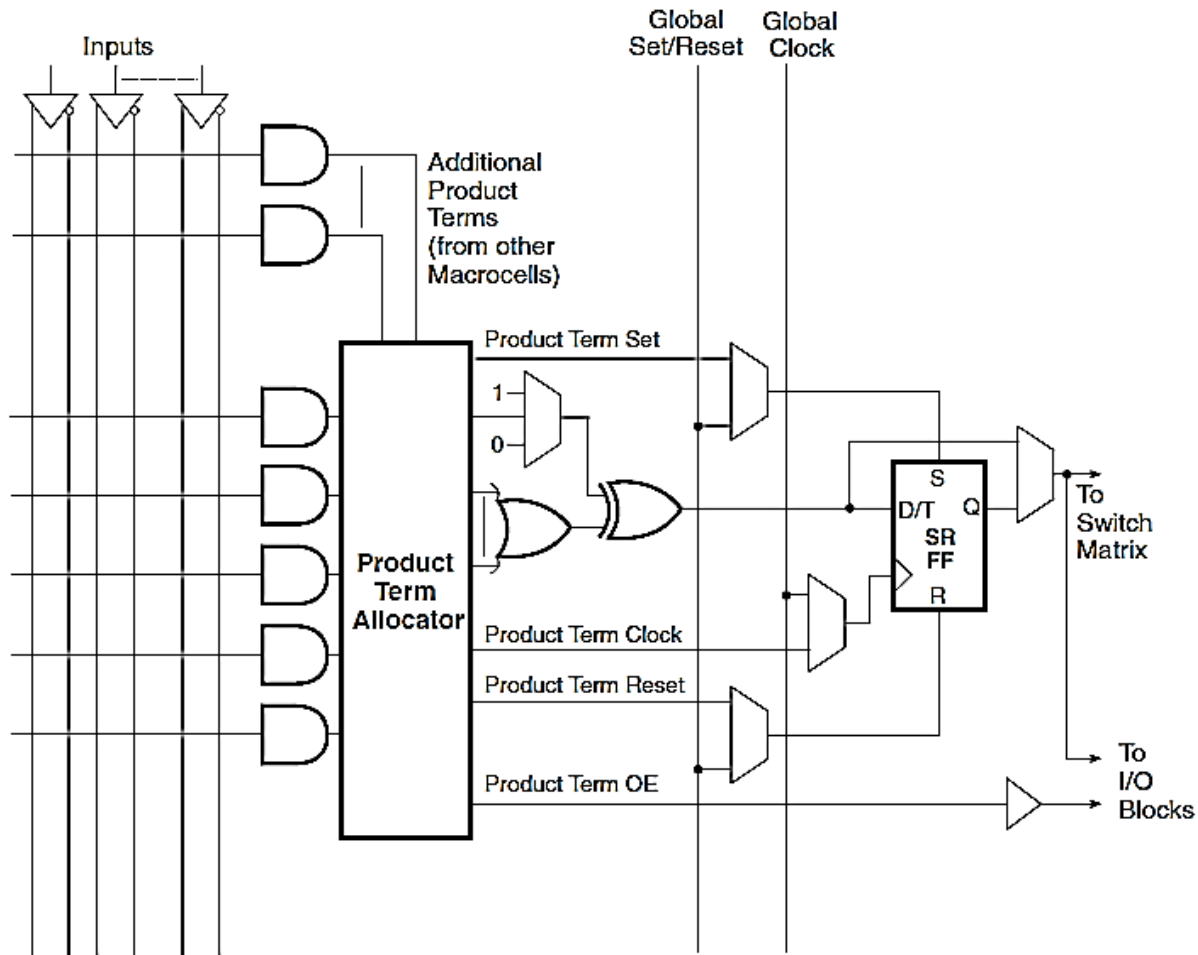


Programmable Logic Devices (PLD)



- معمولاً هر ماکروسل شامل مجموعه‌ای از عبارات حاصلضرب هستند که توسط زیرمجموعه‌ای از آرایه‌های AND برنامه‌پذیر که به نوبه‌ی خود یک خروجی منطقی قابل پیکره‌بندی را تغذیه می‌کنند، تولید می‌شوند. این خروجی منطقی معمولاً شامل یک گیت OR، یک گیت XOR، و یک فلیپ فلاپ است. البته ممکن است این فلیپ فلاپ حذف شده و به اصطلاح «شفاف» باشد.
- همچنین هر گیت OR ممکن است توسط بخشی تا تمام حاصلضرب‌های تولید شده توسط ماکروسل مربوطه (خودی) تغذیه شود. بیشتر CPLDهای امروزی امکان تغذیه‌ی گیت‌های OR توسط حاصلضرب‌های تولید شده توسط دیگر ماکروسل‌ها را نیز فراهم می‌کنند. برای مثال در CPLDهای سری MAX-7000 از شرکت Altera هر گیت OR، علاوه بر امکان تغذیه شدن تا 5 عبارت حاصلضرب تولید شده توسط ماکروسل خودی، امکان تغذیه شدن توسط تا 15 عبارت حاصلضرب مربوط به دیگر ماکروسل‌ها را دارد.
- دیگرام منطقی یک نمونه ماکروسل که در اکثر CPLDهای امروزی استفاده می‌شود در شکل بعد نشان داده شده است.

Programmable Logic Devices (PLD)



Programmable Logic Devices (PLD)



• مهمترین ویژگی های افزاره های CPLD:

- در مقایسه با FPGA ها، CPLD ها به دلیل معماری داخلی ساده تر و برخورداری کمتر از قابلیت انعطاف، دارای ویژگی های زمان بندی قابل پیش بینی هستند؛ به همین دلیل در کاربردهای کنترلی حیاتی و نیز کاربردهایی که نیاز به سطح کارآمدی بالایی دارند، مناسب تر هستند.
- CPLD ها به دلیل هزینه و مصرف توان پایین تر گزینه ای ایده آلی در کاربردهای جابجایی و مبتنی بر باتری نظیر تلفن های همراه و دستیارهای دیجیتال می باشند.
- برخی از مهمترین کاربردهای CPLD ها
 - کاربردهای منطق متصل
 - پیاده سازی طراحی های کنترلی حساس و مهم مانند کنترلرهای گرافیکی، کنترل کش، دریافت/ارسال کننده های آسنکرون (UART)، و کنترلرهای شبکه های محلی (LAN)
 - تلفن های همراه
 - همیارهای دیجیتال
 - کاربردهایی که نیاز مبرم آنها بیشتر به گیت های AND و OR است تا به فلیپ فلاپها
 - کاربردهایی که نیاز به تغییر سریع طراحی و حتی تغییر پیکره بندی سیستم دارند مثلاً در تغییر پروتکل ارتباطی سیستم های مخابراتی

Programmable Logic Devices (PLD)

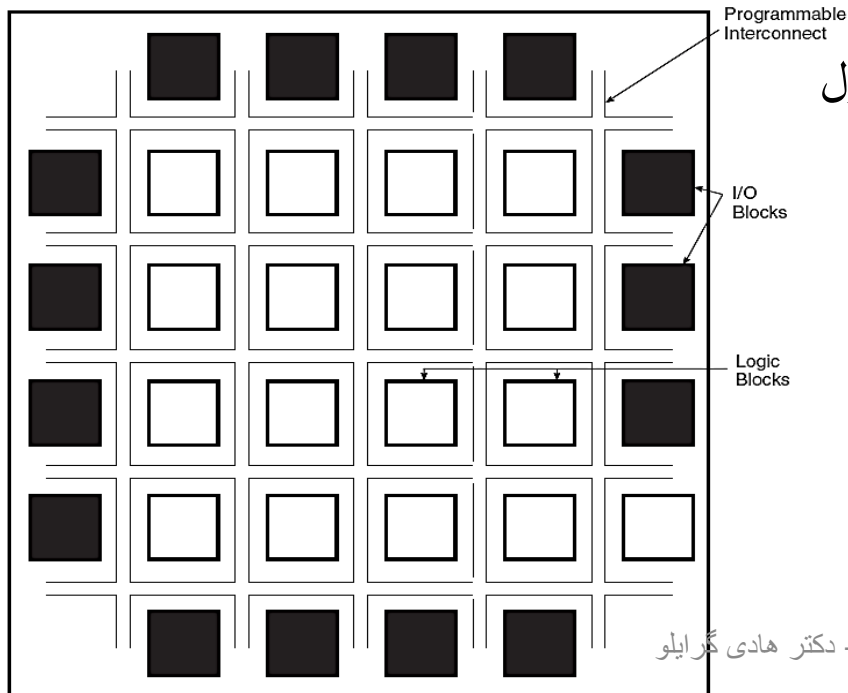


- آرایه گیت‌های قابل برنامه‌ریزی در محل (FPGA):
 - میزان پیچیدگی CPLD ها از حدی بیشتر قابل افزایش نیست. برای رسیدن به ظرفیتهای منطقی بزرگتر از FPGA ها استفاده می‌شود.
 - ایده اصلی در FPGA این است که بلوکهای منطقی را ساده‌تر (به عبارت دیگر، خلوت‌تر) و تعداد آنها را بسیار زیادتر کنند تا به این ترتیب بتوان به ظرفیت منطقی بالایی دست یافت. بنابراین، یک FPGA نیز مشابه CPLD ها، شامل آرایه‌ای از بلوکهای منطقی است که توسط کاربر قابل سازمان‌دهی است.
 - معماری داخلی یک FPGA سه بخش اصلی دارد:
 - آرایه‌ای از بلوکهای منطقی
 - اتصالات داخلی قابل برنامه‌ریزی
 - بلوکهای I/O (ورودی/خروجی)

Programmable Logic Devices (PLD)



- بلوکهای I/O تعیین می‌کنند که هر کدام از پایه‌های همه منظوره FPGA به عنوان ورودی، خروجی، و یا دوجته برنامه‌ریزی شوند.
- بلوکهای منطقی معمولاً شامل تعدادی گیت منطقی یا یک جدول جستجو (یا LUT) هستند که به حداقل یک فلیپ فلاپ متصل‌اند (آنها را تغذیه یا درایو می‌کنند)
- اتصالات داخلی قابل برنامه‌ریزی نیز اتصال بین بلوکهای منطقی و بلوکهای I/O را برقرار و فراهم می‌کنند.
- معماری داخلی یک نمونه FPGA متداول



Programmable Logic Devices (PLD)

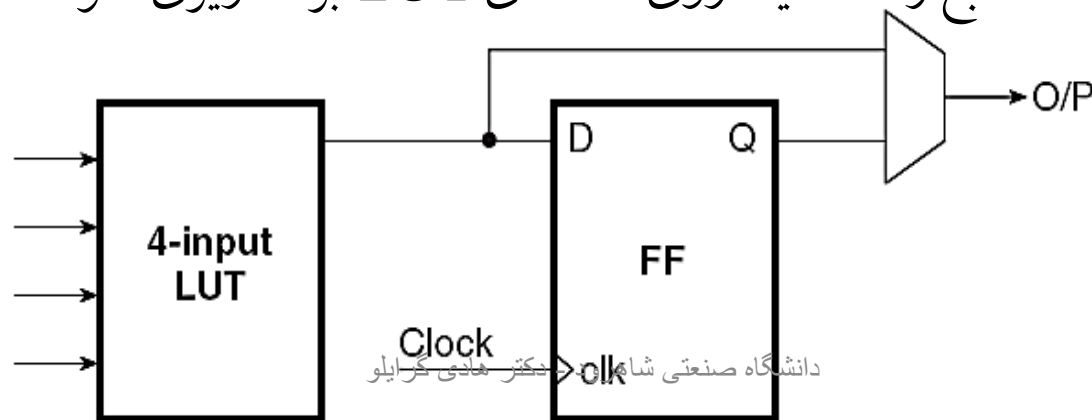


- تفاوت اساسی بین افزاره‌های CPLD و FPGA در معماری داخلی آنها است. مهمترین ویژگی یا شاخصه‌ی معماری یک افزاره‌ی CPLD تعداد نسبتاً کمی از آرایه‌های منطقی برنامه‌پذیر از نوع جمع حاصلضرب‌ها (SOP) است که تعداد کمی از فلیپ‌فلاپ‌های کلاک‌دار را تغذیه می‌کنند. این ویژگی موجب کاهش قابلیت انعطاف اما افزایش دقت و قابلیت پیش‌بینی در مشخصه‌های زمان‌بندی افزاره‌های CPLD می‌شود (همین امر موجب جذابیت آنها در کاربردهای کنترلی شده است).
- در طرف مقابل، ویژگی یا شاخصه‌ی غالب در معماری افزاره‌های FPGA شامل (1) اتصالات برنامه‌پذیر و (2) ساده‌تر بودن بلوک‌های منطقی قابل پیکره‌بندی می‌باشد. بلوک‌های منطقی در افزاره‌های FPGA می‌توانند از حد سادگی یک ماکروسل در افزاره‌های PLD تا بزرگتر و پیچیده‌تر تغییر کنند اما در هر صورت میزان پیچیدگی آنها به هیچ وجه به حد پیچیدگی بلوک‌های منطقی موجود در افزاره‌های CPLD نمی‌رسد. این امر موجب افزایش قابلیت انعطاف و متنوع‌تر شدن نوع طراحی‌هایی (در مقایسه با CPLD) می‌شود که می‌توان با افزاره‌های FPGA انجام داد.

Programmable Logic Devices (PLD)



- بلوکهای منطقی در FPGAها، برخلاف CPLDها، بسیار ساده تر هستند.
- یک نمونه بلوک منطقی ساده (شکل زیر) شامل یک جدول جستجوی (LUT) چهار-ورودی است که خروجی آن یک فلیپ فلاپ کلاک دار را تغذیه می کند.
- خروجی این بلوک منطقی می تواند هر یک از دو نوع رجیستر شده و رجیستر نشده (برگرفته از خروجی LUT) باشد. انتخاب نوع خروجی برعهده ی مالتی پلکسر است. یک جدول جستجو (LUT) چیزی جز یک آرایه ی حافظه به عرض یک بیت نمی باشد. در این حافظه، خطوط آدرس همان ورودی های وارد شده به بلوک منطقی بوده و خروجی داده ی حافظه نیز خروجی LUT محسوب می شود. از یک LUT با n ورودی می توان برای تحقق هر تابع منطقی n -ورودی استفاده نمود. برای این کار کافی است جدول صحت تابع را مستقیماً روی حافظه ی LUT برنامه ریزی نمود.

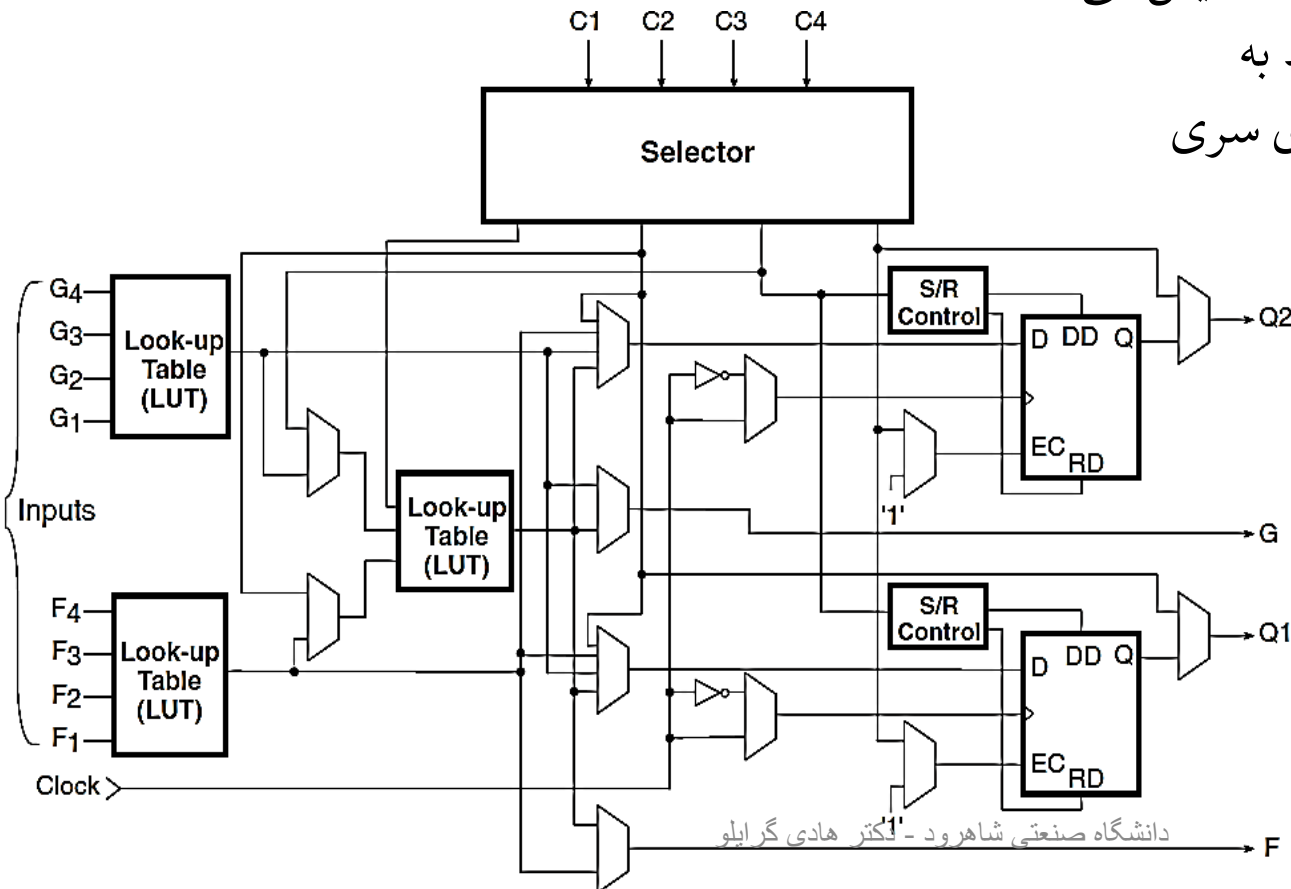


Programmable Logic Devices (PLD)



– برای برقراری امکان تحقق توابع منطقی پیچیده‌تر، بلوک‌های منطقی می‌توانند شامل تعداد بیشتری LUT و فلیپ فلاپ باشند. شکل زیر معماری یک نمونه از چنین بلوک‌های منطقی را نمایش می‌دهد.

– این معماری مربوط به بلوک منطقی افزاره‌های سری XC4000 از شرکت Xilinx می‌باشد.



Programmable Logic Devices (PLD)

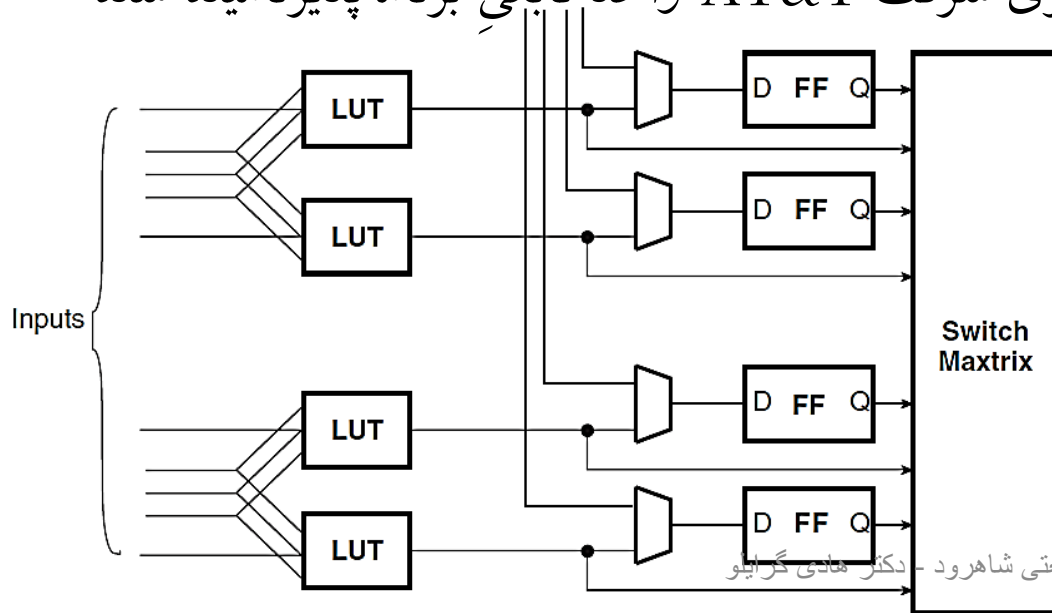


- این بلوک منطقی دارای دو LUT چهار-ورودی است که از طریق ورودی‌های بلوک منطقی تغذیه می‌شوند. در کنار این دو LUT از یک LUT سوم نیز به منظور افزایش تنوع توابع مورد نظر جهت پیاده‌سازی استفاده شده است. در این بلوک منطقی از دو فلیپ فلاپ استفاده شده است.

- یک معماری مشابه دیگر که مبتنی بر LUT است متعلق به شرکت AT&T

- این معماری از چندین LUT و فلیپ فلاپ تشکیل شده است. این معماری مربوط به یک بلوک منطقی است که از سوی شرکت AT&T واحد تابعی برنامه‌پذیر نامیده شده است. این بلوک منطقی را

می‌توان یا به صورت چهار LUT
چهار-ورودی یا دو LUT پنج-
ورودی و یا یک LUT شش-
ورودی پیکره‌بندی کرد.



Programmable Logic Devices (PLD)



– مهمترین ویژگیها و تفاوتهای بین CPLD ها و FPGA ها که ما باید در انتخاب آنها برای کاربرد مورد نظرمان به آنها توجه کنیم

- در CPLD ها معماری داخلی دارای قابلیت انعطاف کمتری است؛ لذا، مشخصات زمانی آنها قابل پیش‌بینی‌تر است.
- به دلیل قابل پیش‌بینی‌تر بودن مشخصات زمانی CPLD ها، آنها مناسب کاربردهای کنترلی حساس و دیگر کاربردهایی هستند که نیازمند سطح کارایی بالایی می‌باشند.
- CPLD ها دارای مصرف توان و قیمت کمتری نسبت به FPGA ها می‌باشند؛ لذا، مناسب کاربردهای قابل حمل و مبتنی بر باتری هستند.
- ظرفیت منطقی FPGA ها و در نتیجه، قدرت مانور آنها بیشتر از CPLD ها است. FPGA های امروزی دارای ظرفیت حدود چند ده میلیون گیت هستند.
- برخی FPGA های امروزی شامل پردازشگرهای داخلی، حافظه‌های داخلی بزرگ، سیستمهای مدیریت کلاک، و پشتیبانی از بسیاری از تکنولوژیهای سیگنال‌دهی بین ادوات می‌باشند. این ویژگیها باعث افزایش تنوع کاربردهای FPGA ها در مقایسه با CPLD ها می‌گردد.
- FPGA ها در بسیاری از کاربردها و عمدتاً در ذخیره و پردازش داده، پردازش سیگنالهای دیجیتال، اندازه‌گیری، و مخابرات استفاده دارند.
- FPGA ها مشابه با CPLD ها، حتی زمانی که روی برد لحیم شده باشند نیز قابل برنامه‌ریزی هستند اما این برنامه ریزی معمولاً فرار بوده و هر بار که برق سیستم وصل می‌شود، و یا هر بار که نیاز به اعمال تغییرات و یا تغییر کاربری باشید، باید دوباره برنامه‌ریزی شوند.

Programmable Logic Devices (PLD)



برای تعجیل در فرج حضرت ولی عصر (عج) صلوات