

PROGRAMLANABİLİR LOJİK DEVRELER

Programmable Logic Devices- PLD

Programlanabilir kombinasyonel devreler, fiziksel olarak tek bir tüm devre içerisinde birçok VE-VEYA kapıları bulunan ve bunlar arasındaki bağlantıların uygun şekilde programlanmasıyla karmaşık bir lojik fonksiyonun tek bir tümdevre ile gerçekleştirilmesini sağlayan elemanlardır.

Böylece tasarlanacak lojik devrenin daha az tümdevre kullanılarak gerçekleştirilmesi sağlanır.

Programlanabilir kombinasyonel devre, kendi bünyesinde programlanabilir özellikte VE-VEYA kapıları ve programlama işinin yapılmasını sağlayan sigortalar içerir.

Programlama işlemi, giriş değişkenleriyle kapı girişlerine bağlantısını sağlayan sigortaların olduğu gibi bırakılmasıyla sağlanır. Eğer değişkenin kapı girişyle bağlantısı yoksa o değişkene ilişkin sigorta atılır.

Programlanmış bazı PLD elemanlarını tekrar programlamak mümkündür. Ancak bazıları tekrar programlanamazlar.

Bir kez programlanabilen : PROM, PAL, PLA
Birden fazla programlanabilen : EPROM, EEPROM

Programlanabilir Lojik Devre Elemanları

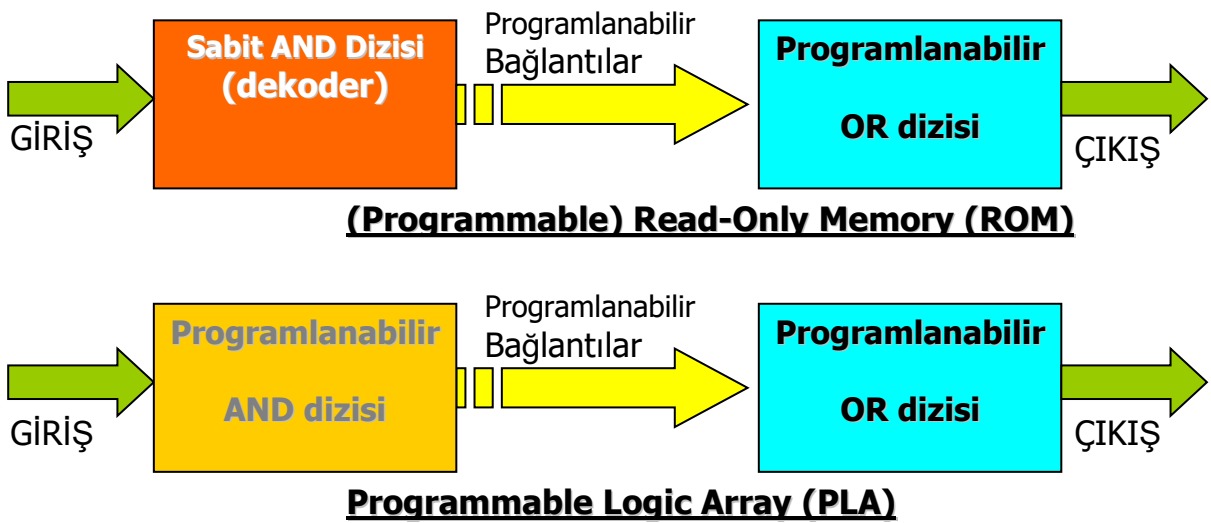
PROM- Programlanabilir yalnızca okunabilir bellek (Programmable Read Only Memory)

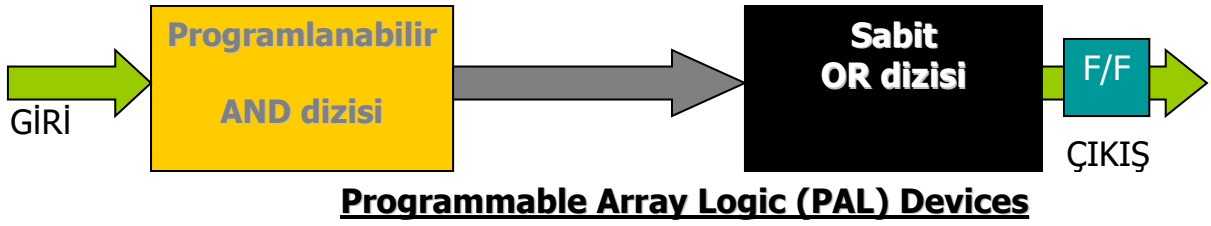
PAL- Programlanabilir Diziler (Programmable Array Logic)

PLA- Programlanabilir Lojik Diziler (Programmable Logic Arrays)

PSA- Programlanabilir Ardışıl Diziler (Programmable Sequential Arrays)

Programlanabilir Lojik Elemanlar (PLD)





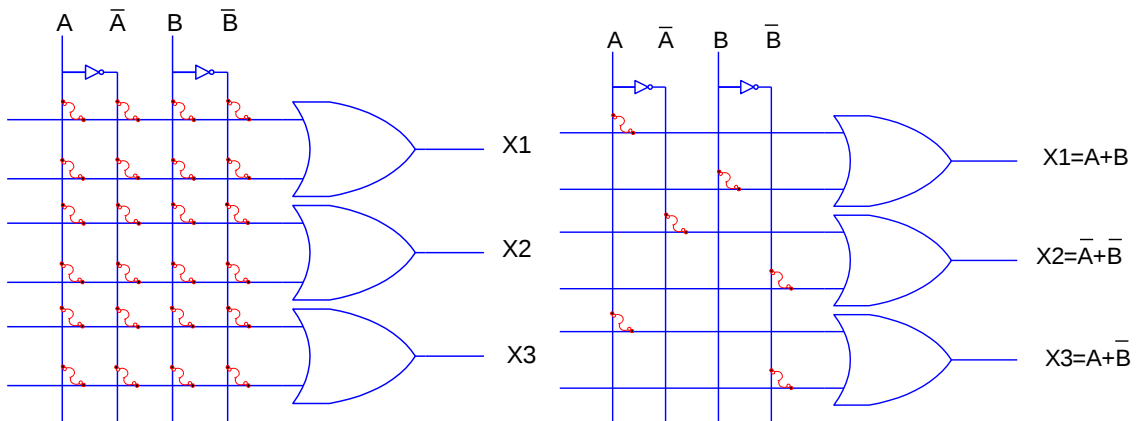
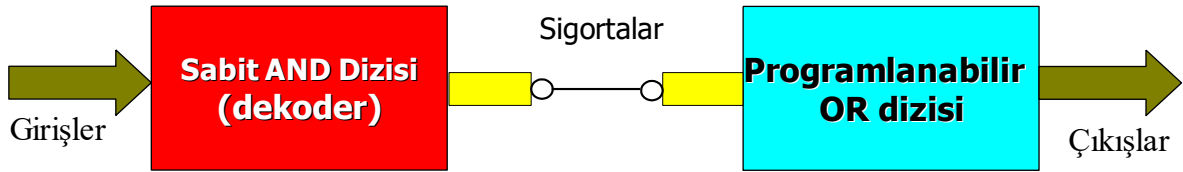
Programmable Read Only Memory (PROM)

PROM, sabit VE kapı dizisi ve sigortalarla programlanabilen VEYA kapı dizisi içeren bir entegre devredir.

Girişler önce VE kapılarına uygulanmakta ve bunların çıkışları VEYA kapılarına programlanarak verilmektedir.

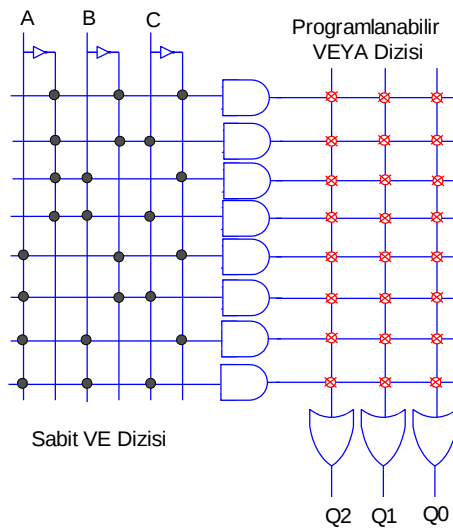
Programlanmadan önce VE kapı dizisindeki tüm çıkışlar VEYA kapısı girişlerine bağlıdır.

Programlama, bağlantı olmayacak VE kapısı çıkışıyla VEYA kapısı girişi arasındaki ilgili sigortayı attırmaktır.



Programlanmamış

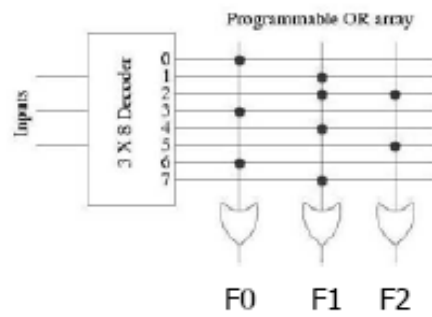
Programlanmış



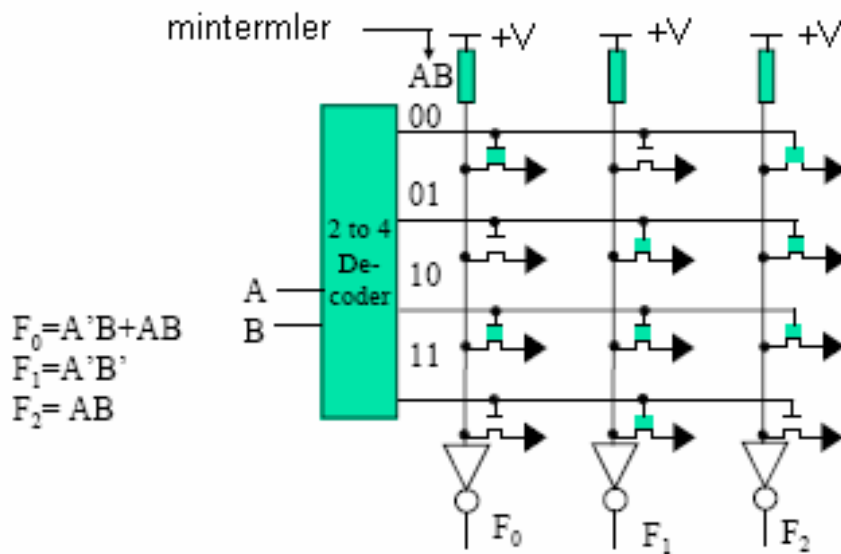
Örnek1:

I0 I1 I2 F0 F1 F2

0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	1	0



Örnek 2:



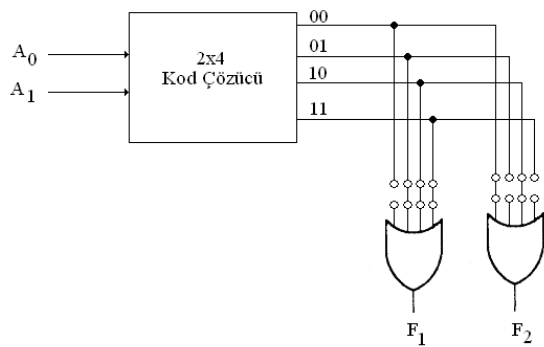
Örnek: İki girişli, iki çıkışlı bir kombinezonal devre PROM kullanılarak tasarlanmak istenmektedir. Boole fonksiyonunun minterimleri aşağıdaki gibi verilmiştir.

$$F_1(A_1, A_0) = \sum(1, 2, 3)$$

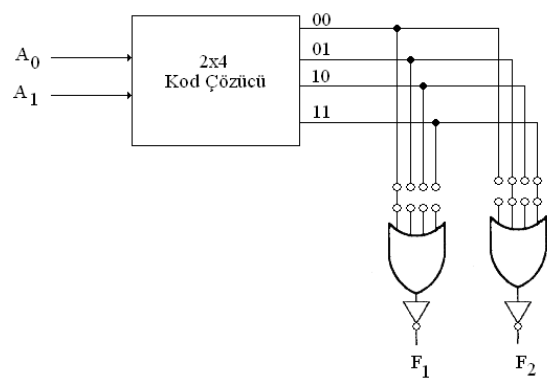
$$F_2(A_1, A_0) = \sum(0, 2)$$

Boole fonksiyonu ile belirlenmiş ifadede iki giriş (n) (A_1, A_0) ve iki çıkış (m) (F_1, F_2) olduğuna göre $2^n \times m$, yani 4×2 'lik bir ROM'a ihtiyaç vardır. Aynı zamanda girişte 2×4 'lük bir kod çözücü olmalıdır. Doğruluk tablosu oluşturmak hangi minterimlerin toplama katılacağını görmek açısından kolaylık sağlar. Şekilde görüldüğü üzere F_1 ve F_2 için minterimlerden faydalanılarak doğruluk tabloları oluşturulmuştur. F_1 için toplamaların çarpımı ifadesinde kullanılan minterimler 1, 2 ve 3 olduğuna göre kod çözücünün 01, 10 ve 11 kodlu çıkışları ROM'un çıkışındaki sigortalı VEYA kapısına giriş olarak bağlanmalıdır. Kullanılmayan minterim ise sigortalı VEYA kapısının girişindeki sigorta atılarak toplama katılması engellenir. F_2 için toplamaların çarpımı ifadesinde kullanılan minterimler 0 ve 2 olduğuna göre kod çözücünün 00 ve 10 kodlu çıkışları ROM'un çıkışındaki sigortalı VEYA kapısına giriş olarak bağlanmalıdır. Kullanılmayan minterimler, 01 ve 11 ise sigortalı VEYA kapısının girişindeki sigortalar atılarak toplama katılması engellenir.

A_1	A_0	F_1	F_2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	0



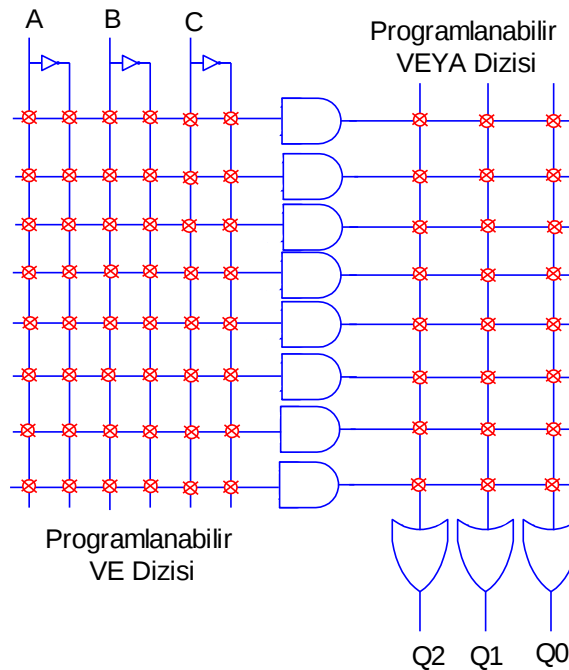
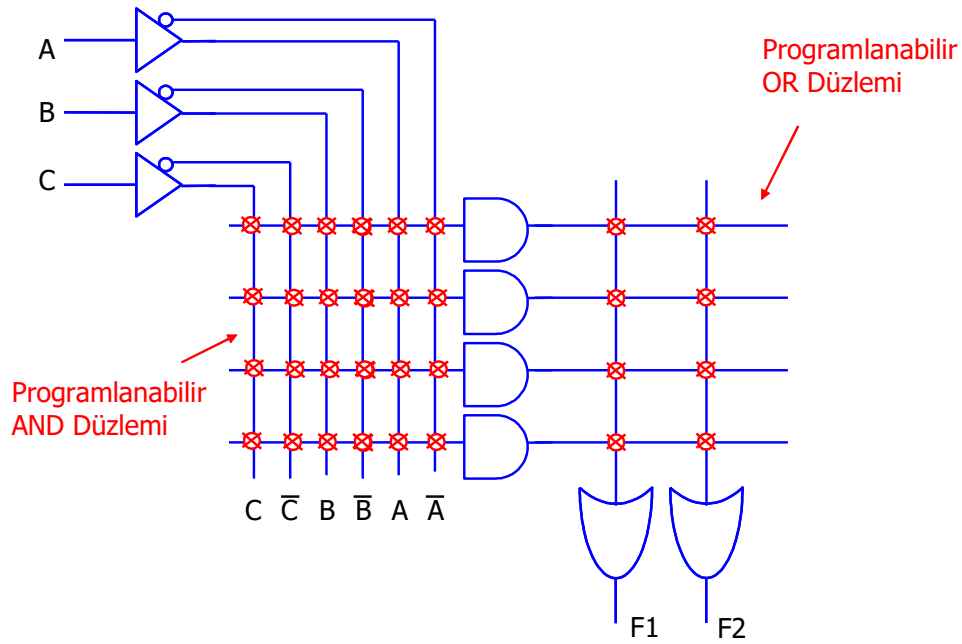
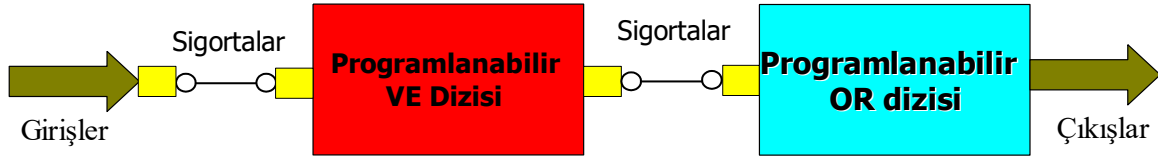
VE-VEYA kapıları ile ROM



VE-VEYA-EVİRME kapıları ile ROM

Programmable Logic Array (PLA)

PLA, Programlanabilen VE kapı dizisiyle programlanabilen VEYA kapı dizisi içermektedir. Hem VE dizisi öncesi hem de VEYA dizisi öncesi programlamak için sigortalar vardır.



Örnek: 1

$$F1(A, B, C) = \sum m(3, 5, 6, 7)$$

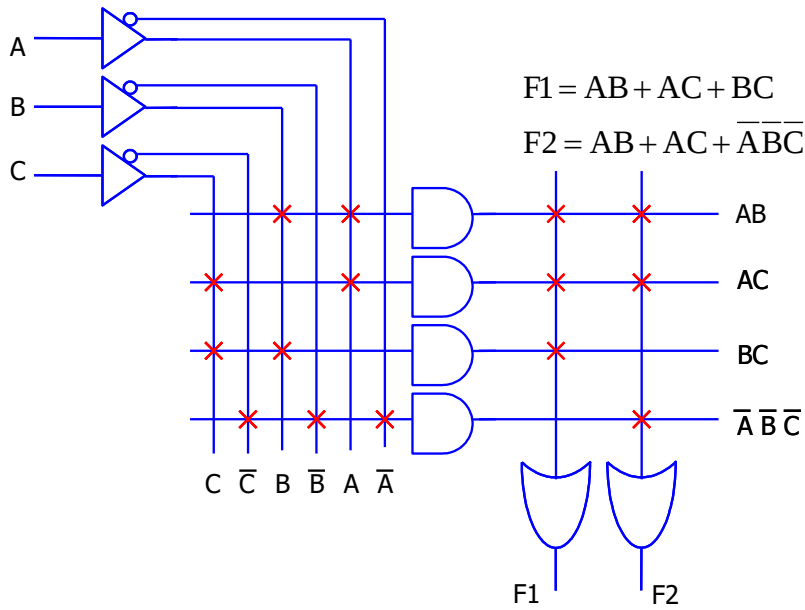
$$F2(A, B, C) = \sum m(0, 5, 6, 7)$$

AB \ C	0	1	
00			
01		1	→ B C
11	1	1	→ A B
10		1	→ A C

AB \ C	0	1	
00	1		→ $\bar{A} \bar{B} \bar{C}$
01			
11	1	1	→ A B
10		1	→ A C

$$F1 = AB + AC + BC$$

$$F2 = AB + AC + \bar{A} \bar{B} \bar{C}$$

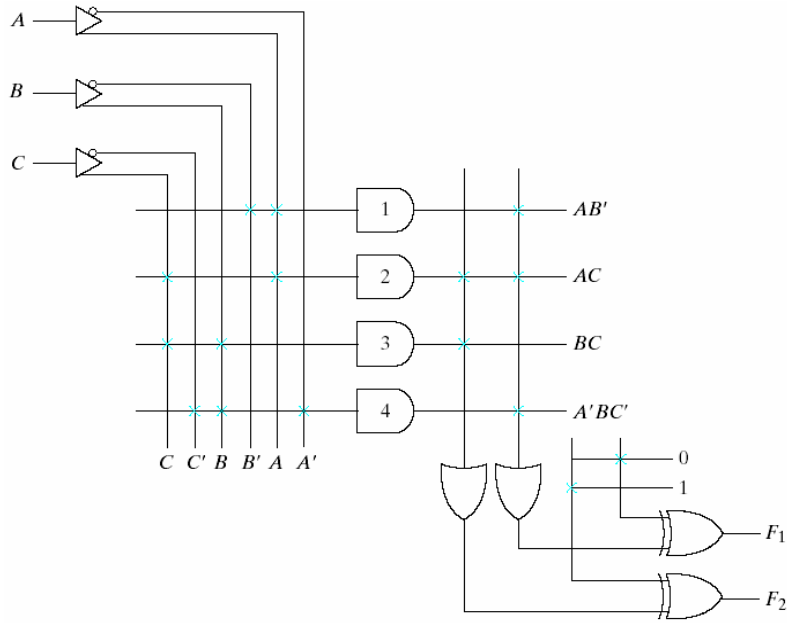


Örnek: 2

$$F1 = AB' + AC + A'BC'$$

$$F2 = (AC + BC)'$$

XOR kapıları ile çıkışlar yada değerleri alınabilmektedir.



PLA Programming Table

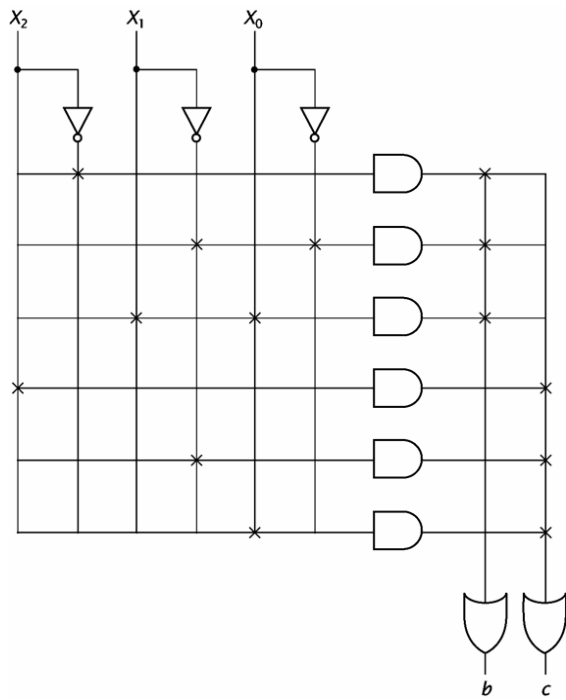
		Inputs			Outputs	
		A	B	C	(T) F_1	(C) F_2
AB'	1	1	0	–	1	–
AC	2	1	–	1	1	1
BC	3	–	1	1	–	1
A'BC'	4	0	1	0	1	–

Örnek 3:

- Girişler ve deęilleri AND kapılarına uygulanabilecek şekilde düzenlenir.
- X işareti ilgili girişin AND kapısına uygulandığını gösterir.
- AND kapıları çıkışları OR kapılarına uygulanır. OR kapılarının çıkışları devre çıkışlarını oluşturur

$$b = X2' + X1'X0' + X1X0$$

$$c = X2 + X1' + X0$$



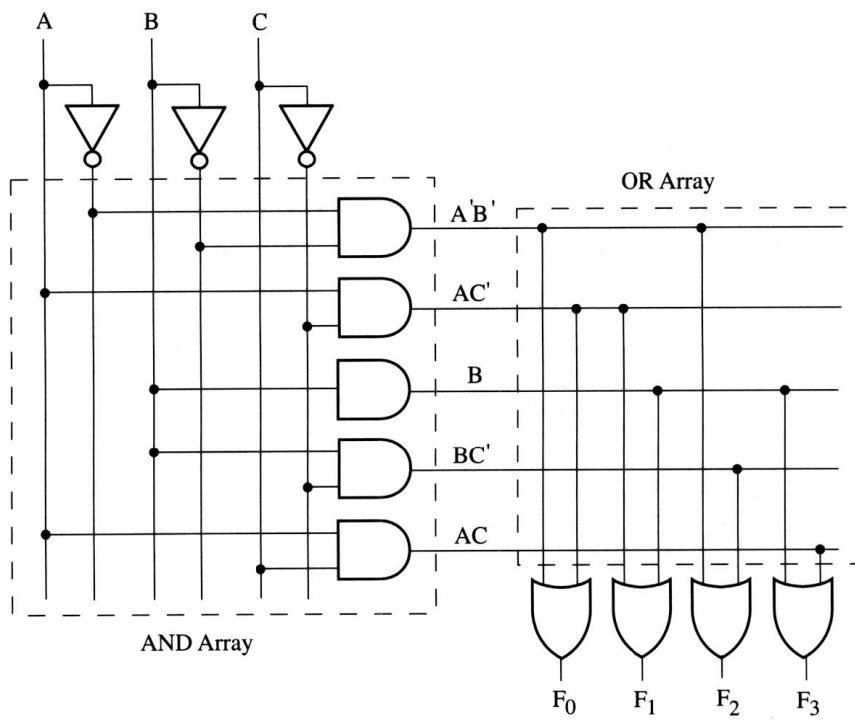
Örnek: 4

$$F_0 = A'B' + AC'$$

$$F_1 = B + AC'$$

$$F_2 = A'B' + BC'$$

$$F_3 = AC + B$$



PLA Tablosu

Product Term	Inputs			Outputs			
	A	B	C	F_0	F_1	F_2	F_3
$A'B'$	0	0	-	1	0	1	0
AC'	1	-	0	1	1	0	0
B	-	1	-	0	1	0	1
BC'	-	1	0	0	0	1	0
AC	1	-	1	0	0	0	1

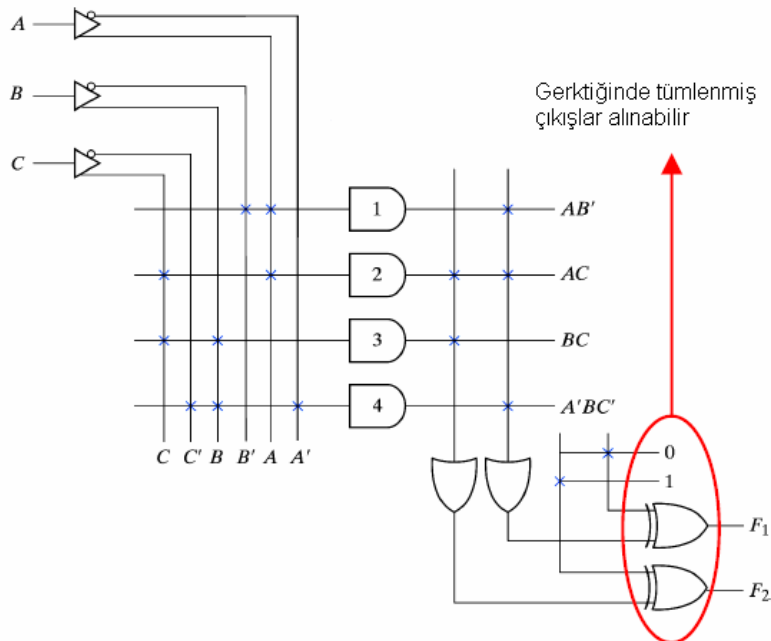
Örnek:5

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

$$F_2 = (AC + BC)'$$

PLA Tablosu

Product Term		Inputs			Outputs (T) (C)	
		A	B	C	F_1	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	-

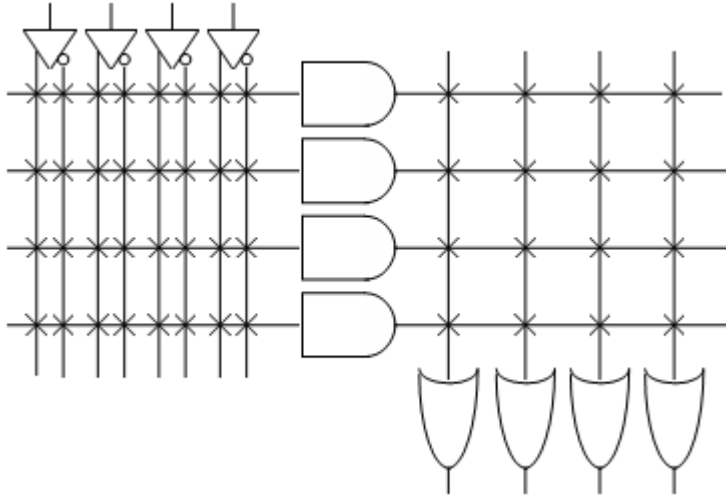


Örnek: 6

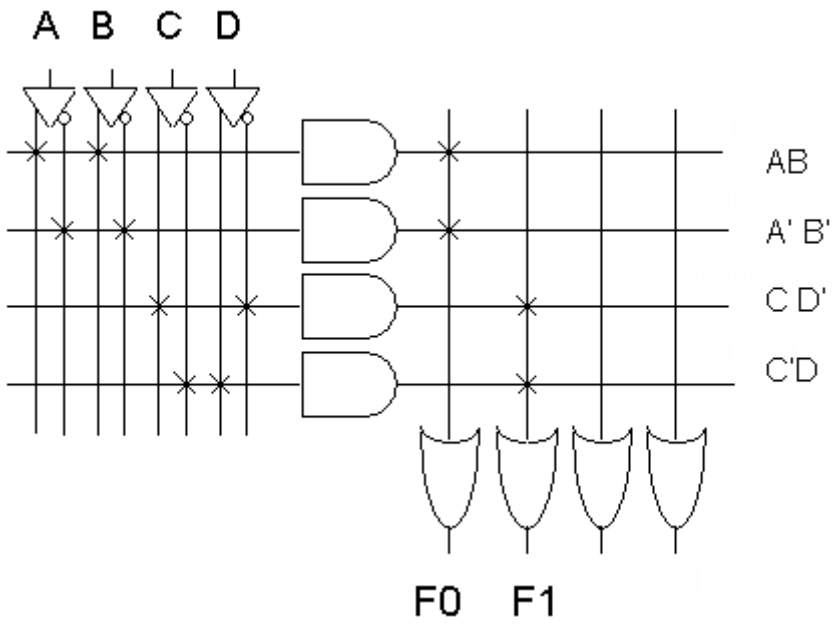
Aşağıdaki fonksiyonları PLA kullanarak gerçekleştiriniz

$$F0 = A B + A' B'$$

$$F1 = C D' + C' D$$



Programlanmamış



Programlanmış

Örnek:7

$$F1 = A B C$$

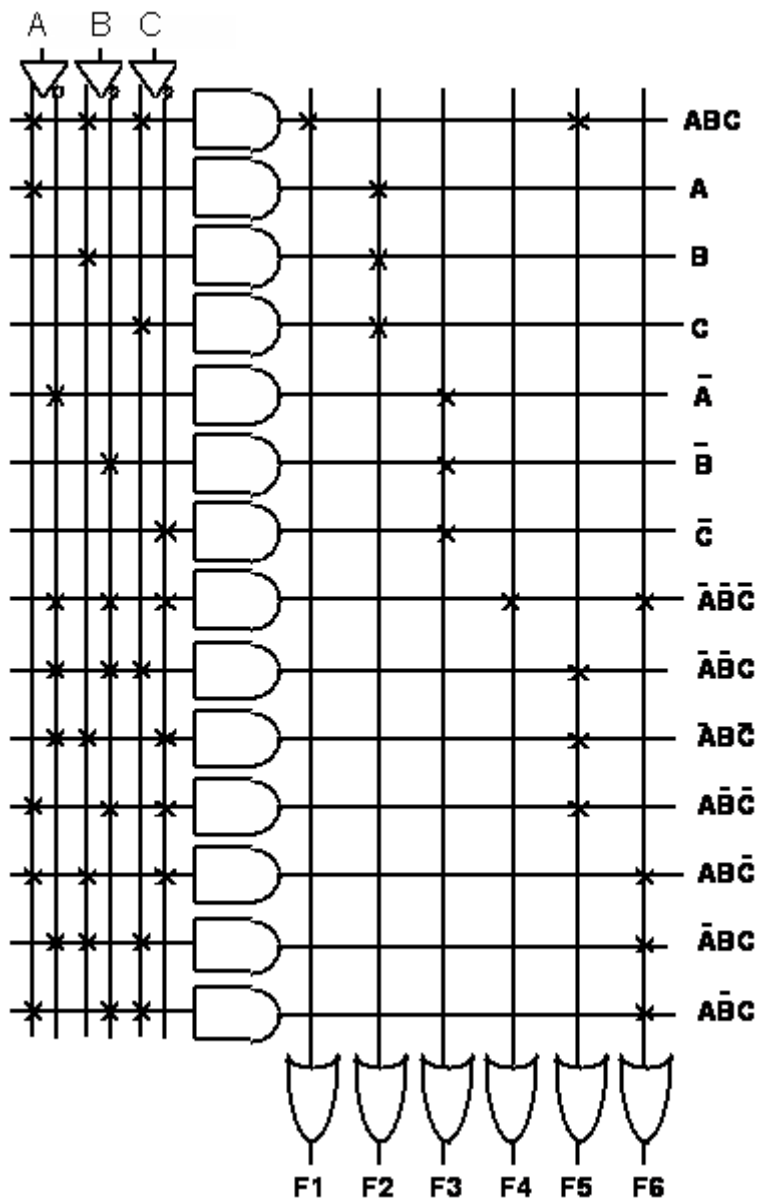
$$F2 = A + B + C$$

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

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

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

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

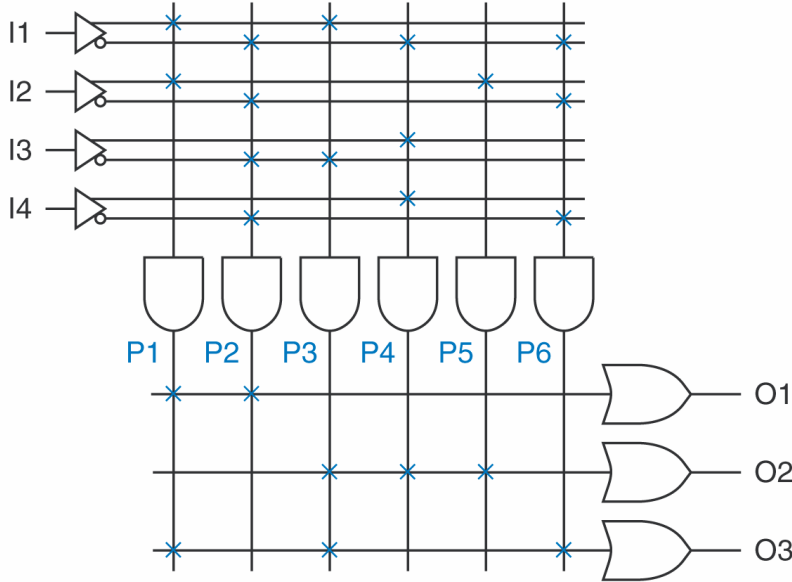


Örnek: 8

$$O1 = I1 \cdot I2 + I1' \cdot I2' \cdot I3' \cdot I4'$$

$$O2 = ?$$

$$O3 = ?$$



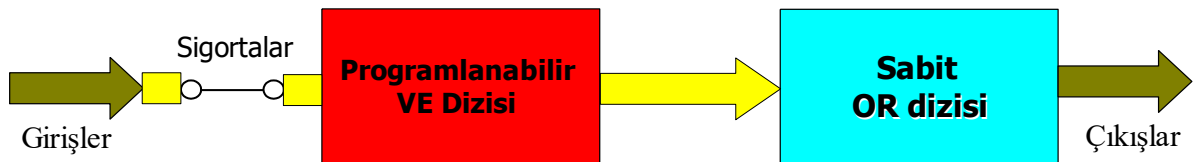
4X3 PLA

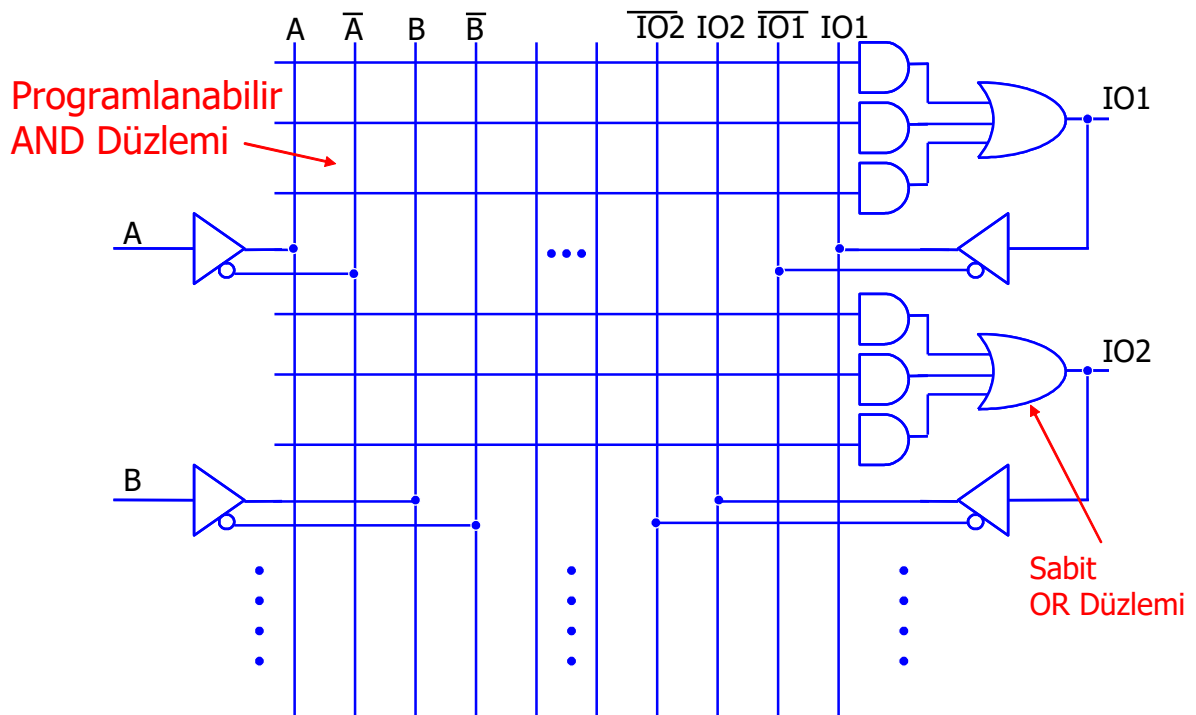
Programmable Array Logic (PAL)

Programlanabilen VE kapı dizisiyle Sabit VEYA kapı dizisi şeklinde bir yapıya sahiptir.

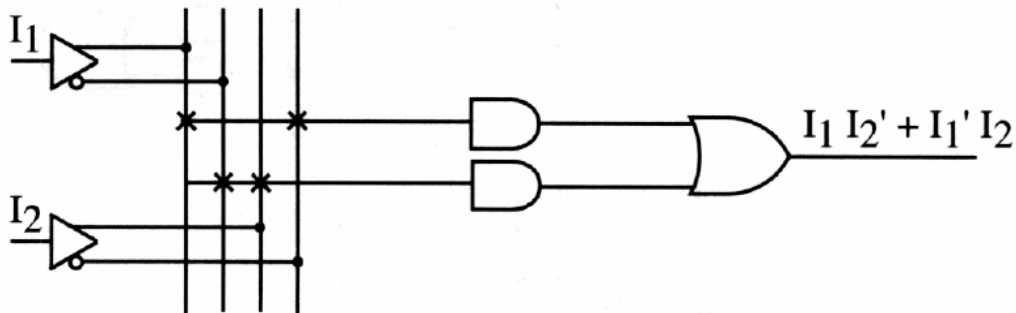
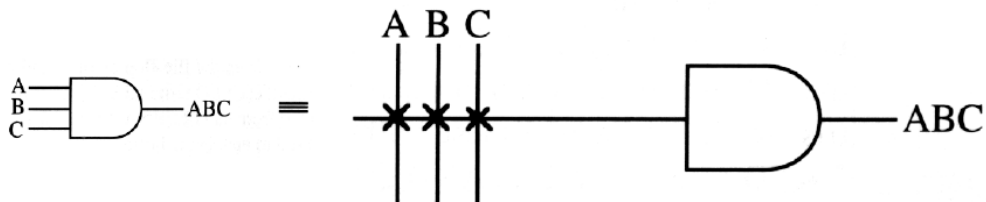
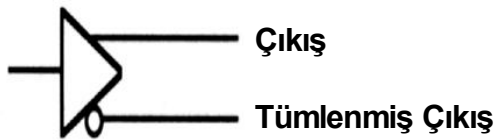
Sadece VE kapı dizisi programlanabilmektedir. Girişler VE kapısına verilmeden önce programlanmaktadır.

Bu elemanlarla çarpımların toplamı şeklinde verilmiş ifadelerin indirgenmiş şekilleri gerçekleşir.

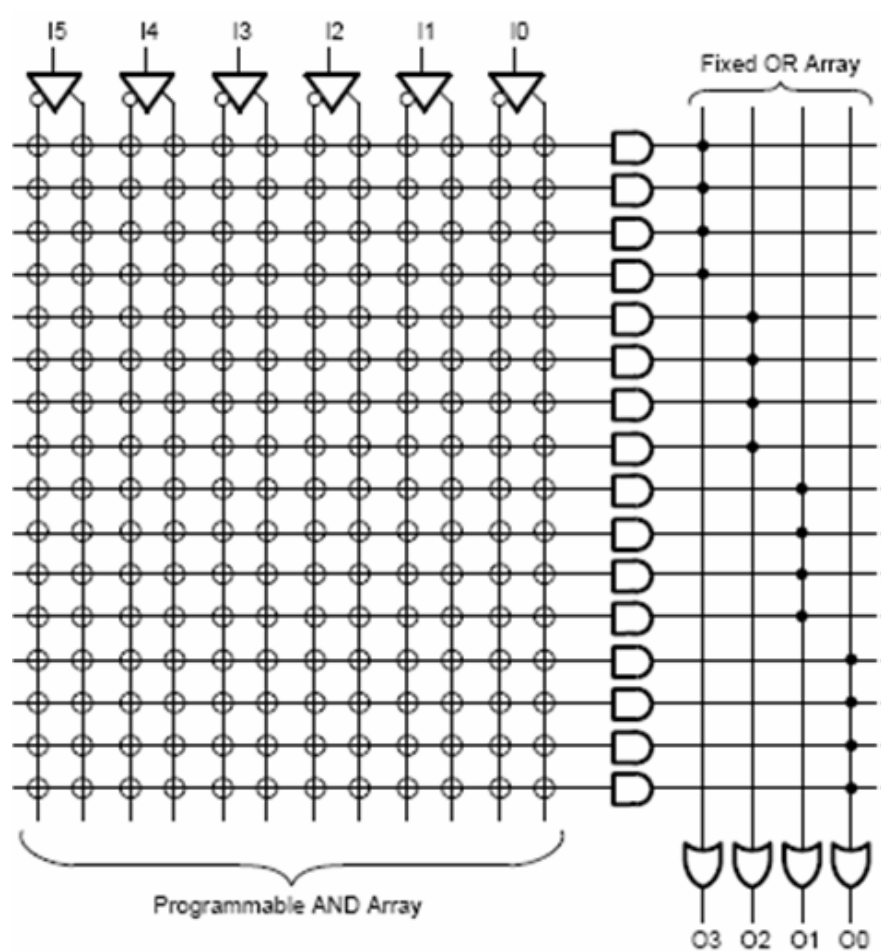
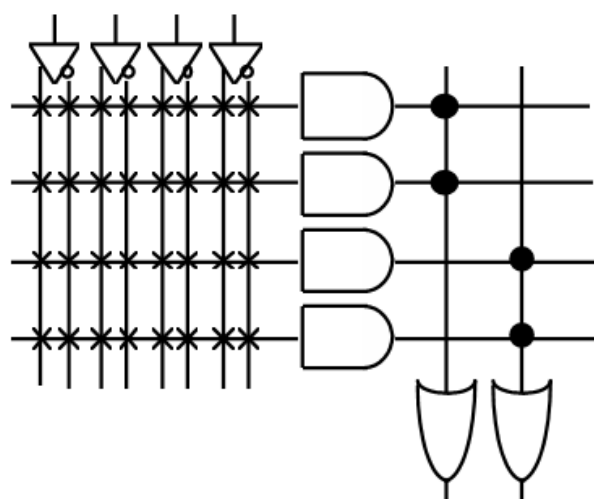




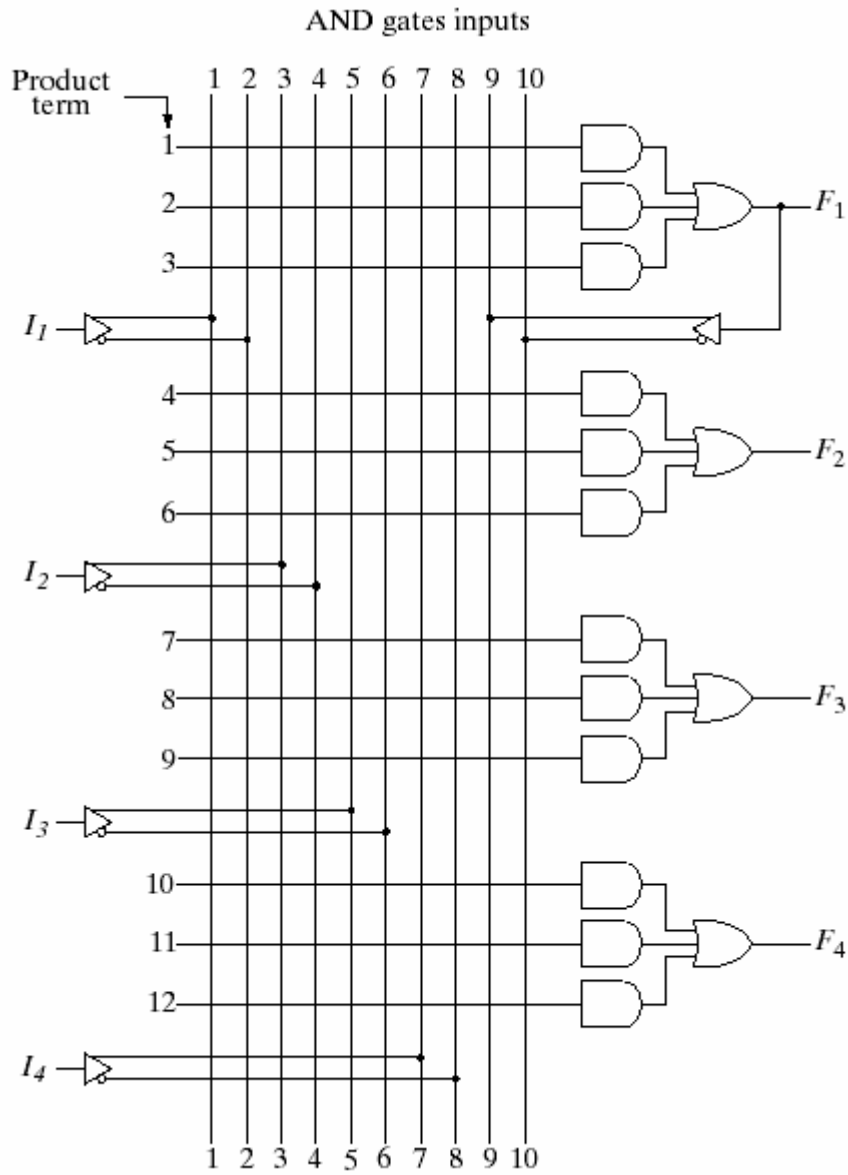
PAL'ın Programlanması



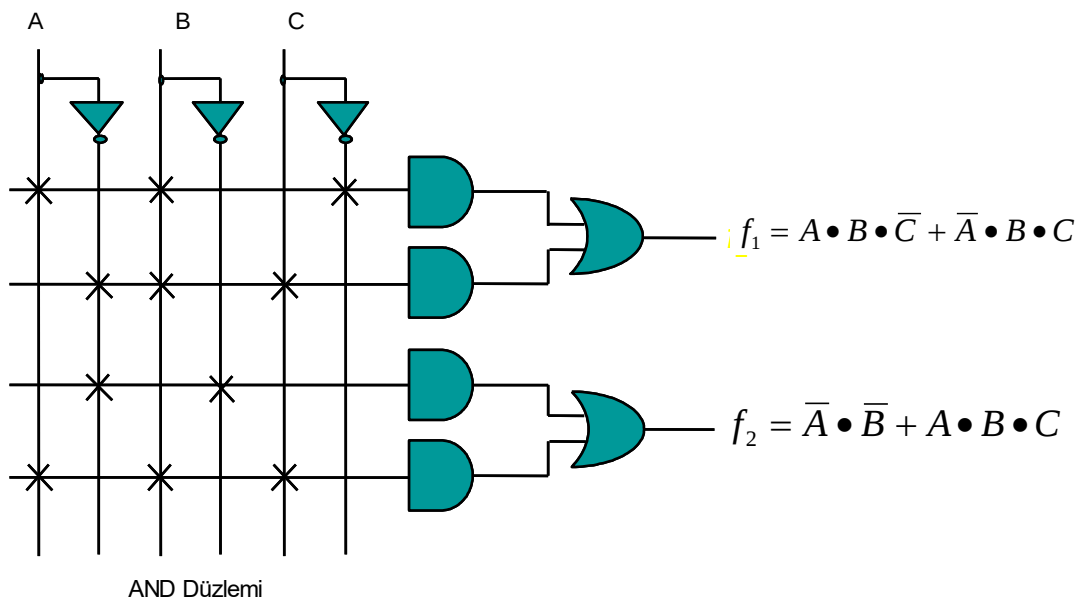
A given column of the OR array has access to only a subset of the possible product terms



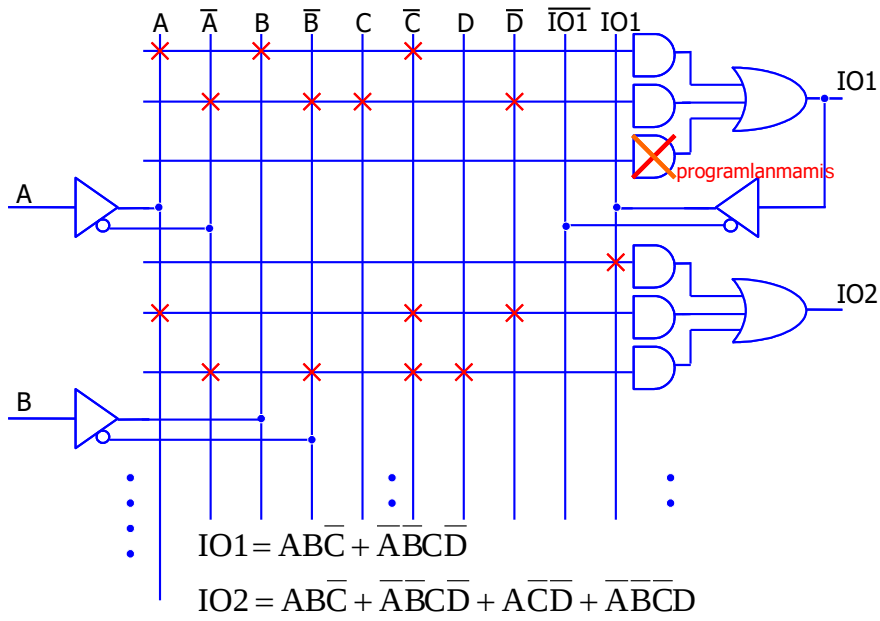
- ⊕ Indicates Programmable Connection
- Indicates Fixed Connection



Örnek: 1



Örnek: 2



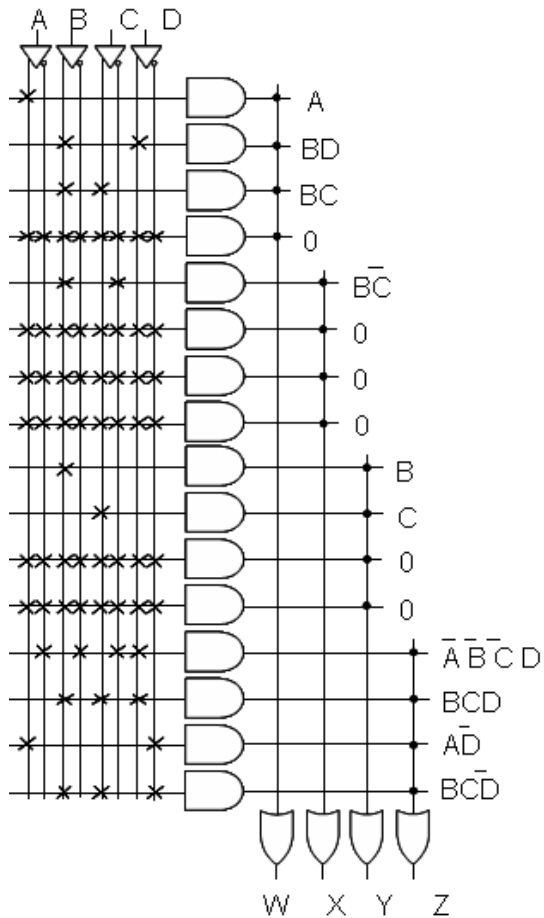
Örnek: 3

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

$$X = B C'$$

$$Y = B + C$$

$$Z = A' B' C' D + B C D + A D' + B C D'$$



Örnek 4:

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

$$x(A,B,C,D) = \Sigma(7,8,9,10,11,12,13,14)$$

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

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

Fonksiyonların sadeleştirilmiş halleri

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

$$x = A + BCD$$

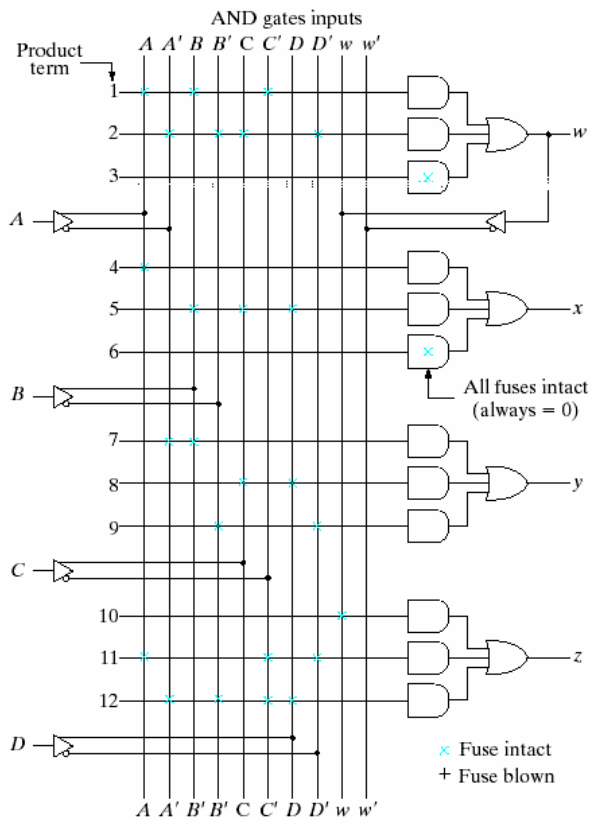
$$y = A'B + CD + B'D'$$

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

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

PAL Tablosu

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



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

$$x = A + BCD$$

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

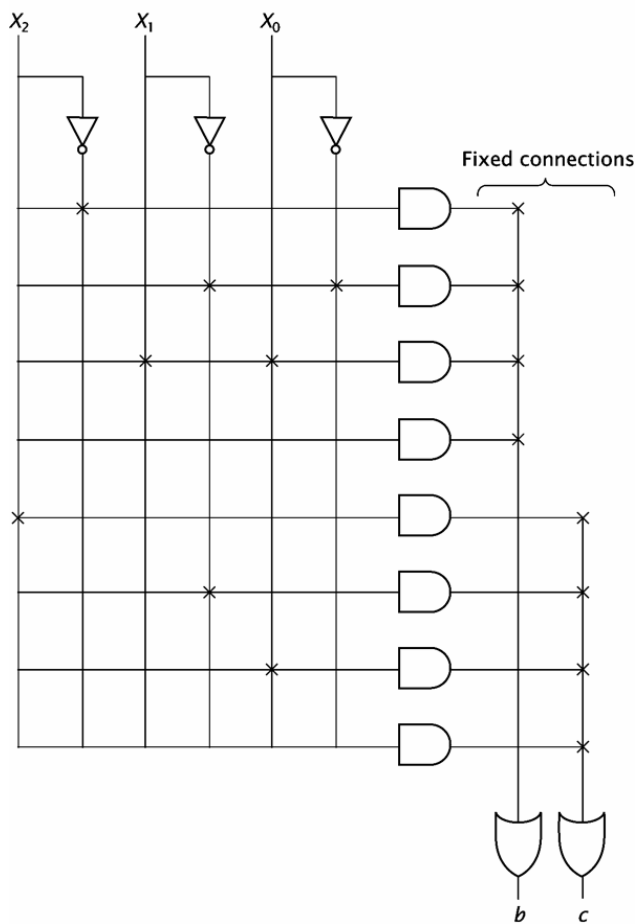
$$z = AC'D' + A'B'C'D$$

Örnek: 5

- OR dizisi sabittir.
- Belirli AND kapıları belirli OR kapılarının girişini oluşturur.
- Örnek:
PLA ile yapılan bir örneğin PAL ile gerçekleştirilmesi

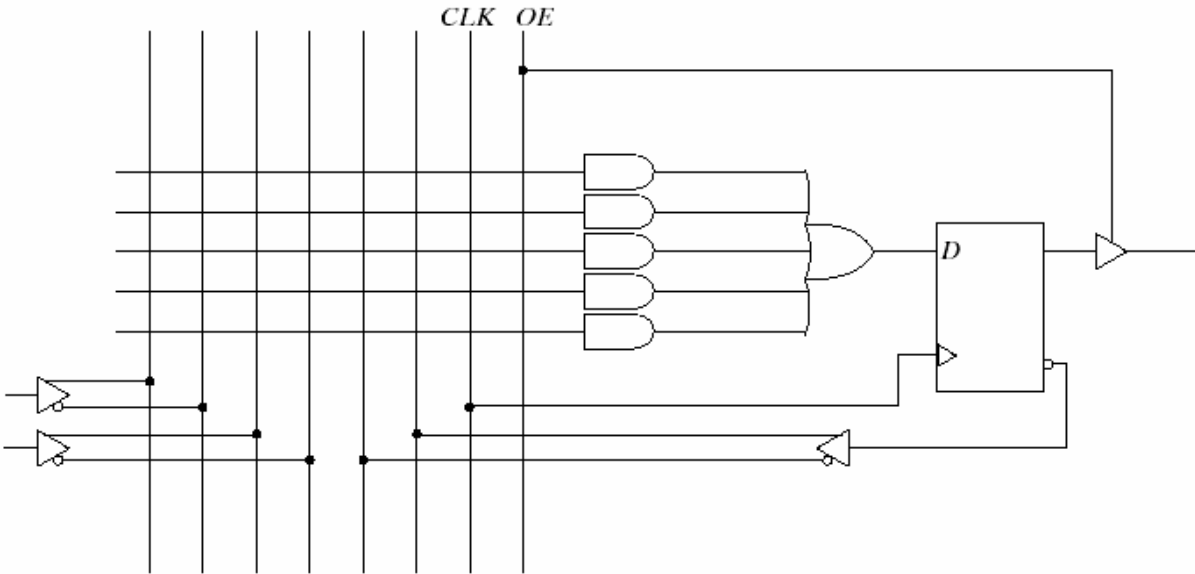
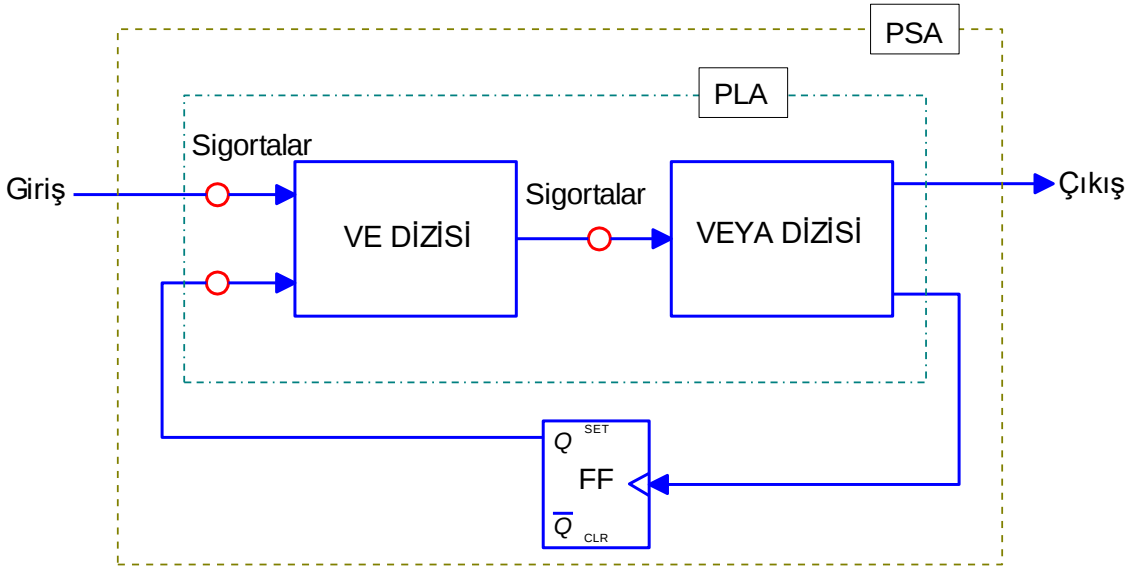
$$b = X_2' + X_1'X_0' + X_1X_0$$

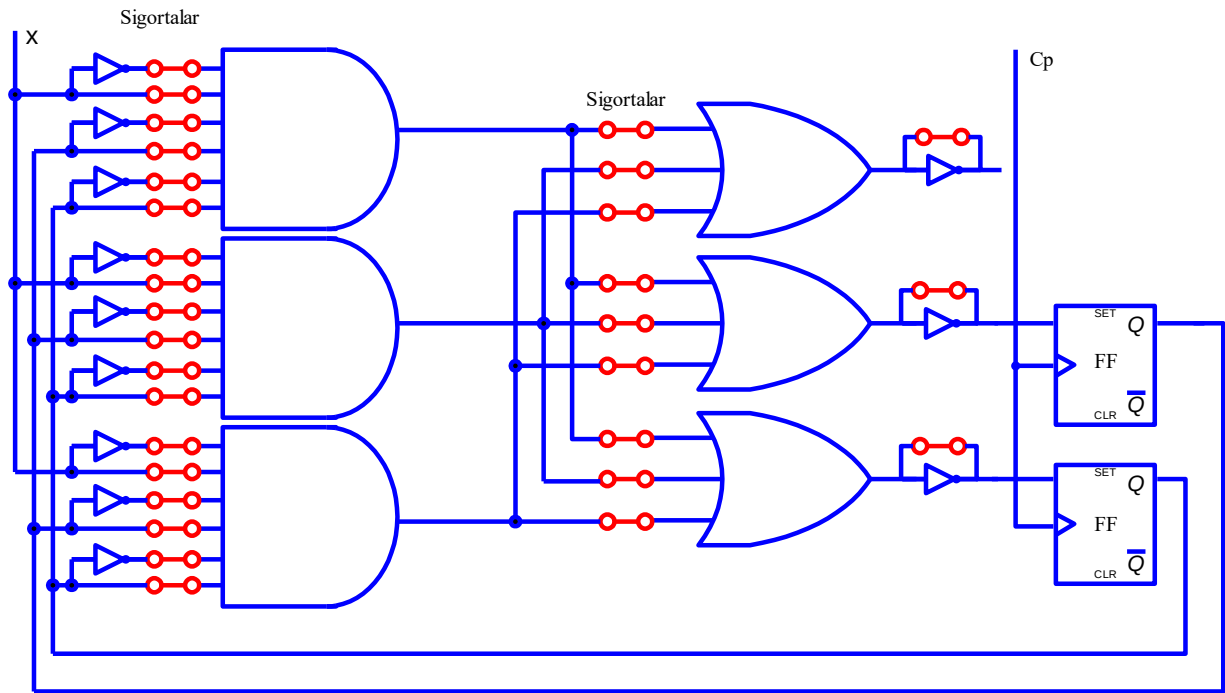
$$c = X_2 + X_1' + X_0$$



Programlanabilir Ardışıl Devreler (Programmable Sequential Arrays -PSA)

Programlanabilir ardışıl devreler, tek bir tüm devre kullanarak ardışıl devrelerin gerçekleştirilmesinde kullanılır. Bu yapıda programlanabilir VE dizisi, programlanabilir VEYA dizisi ve durum saklaması için flip-floplar bulunmaktadır. Kısaca PLA'ya ek olarak PSA'da flip-floplar kullanılmaktadır.



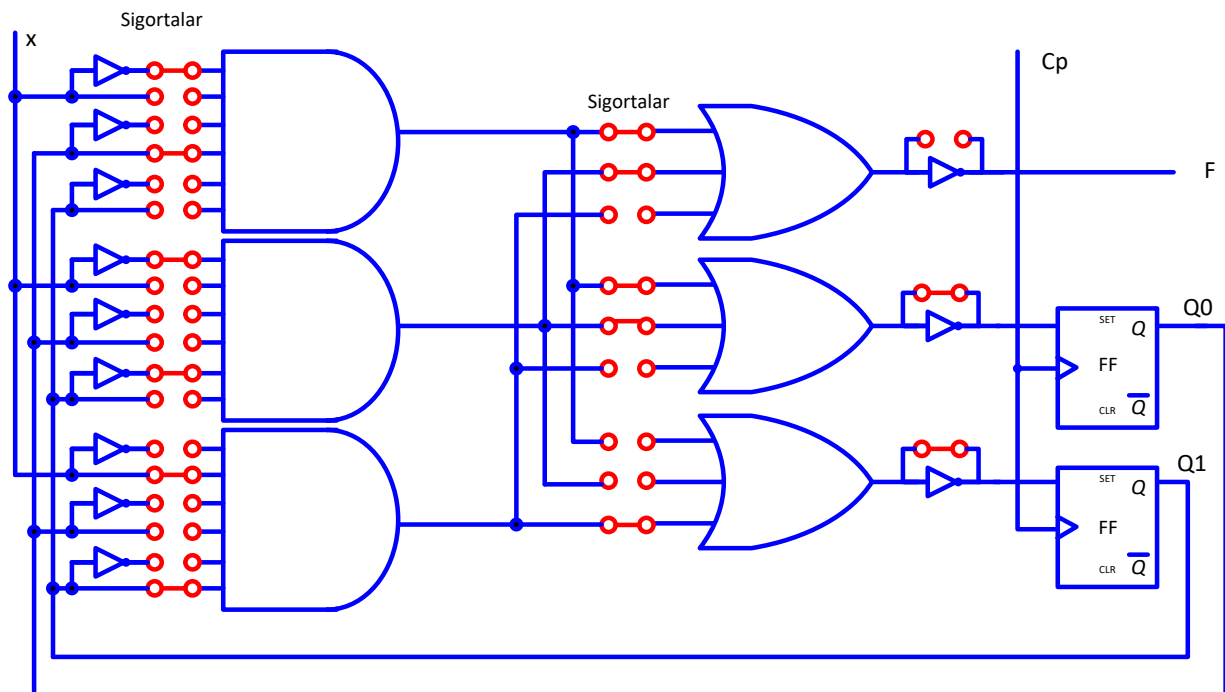


Örnek: 1

$$\overline{F} = Q_0 \overline{X} + \overline{Q_1} \overline{X}$$

$$FF0 = Q_0 \overline{X} + \overline{Q_1} \overline{X}$$

$$FF1 = Q_1 X$$



Örnek: 2

$$Q^+ = D = A'BQ' + AB'Q$$

