VE, VEYA, TÜMLEME lojik işlemleri

- Boole fonksiyonları ile tanımlanırlar.
- İki girişli 16 mümkün olan fonksiyondan 3 tanesini gösterirler.

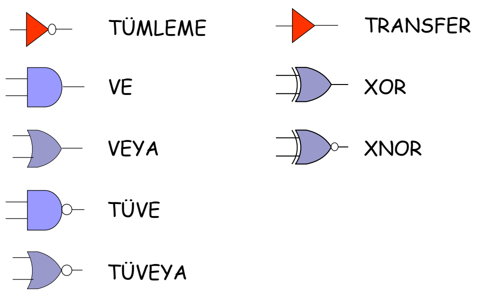
x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

x	y	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1

Diğer Lojik İşlemler

- Bazı iki değişkenli Boole fonksiyonları
  - Sabit fonksiyonlar: $F_0 = 0$ and $F_{15} = 1$
  - VE fonksiyonu: $F_1 = xy$
  - VEYA fonksiyonu: $F_7 = x + y$
  - Dışlayıcı VEYA fonksiyonu (XOR):
    - $F_6 = x'y + xy' = x \oplus y$ (x or y, fakat ikisi birden değil)
  - Eşitlik fonksiyonu (XNOR):
    - $F_9 = xy + x'y' = (x \oplus y)'$ (x y ye eşittir)
  - TÜVEYA (NOR) fonksiyonu :
    - $F_8 = (x + y)' = (x \downarrow y)$ (Not-OR)
  - TÜVE (NAND) fonksiyonu:
    - $F_{14} = (x\, y)' = (x \uparrow y)$ (Not-AND)

Lojik Kapı Sembolleri

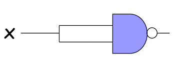


Evrensel Kapı

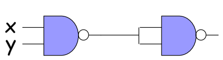
- TÜVE ve TÜVEYA kapıları evrenseldir.
- Bütün Boole fonksiyonları üç lojik işlem kullanılarak ifade edilebilirler:
  - VE, VEYA, TÜMLEME
- TÜVE ve TÜVEYA kapıları da bu üç işlemi gerçekleyebilirler.

x	y	(xy)'	x'	y '	(x' y')'
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	1

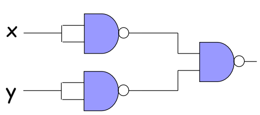
TÜVE Kapısı



$(x\,x)' = x'$ ➔ TÜMLEME

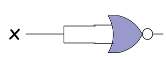


$[\,(x\,y)'\,]' = x\,y$ ➔ VE

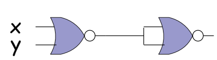


$(x'\,y'\,)' = x + y$ ➔ VEYA

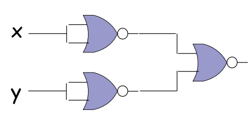
TÜVEYA Kapısı



$(x + x)' = x'$ ➔ TÜMLEME



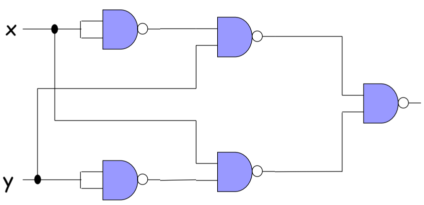
$[\,(x + y)'\,]' = x + y$ ➔ VEYA



$(x' + y'\,)' = x \cdot y$ ➔ VE

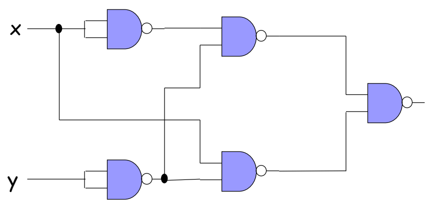
Örnek 1

- $F_1 = x' y + xy'$



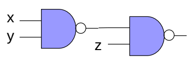
Örnek 2

- $F_2 = x' y' + xy'$

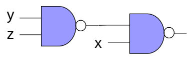


Çok Girişli Kapılar

- VE ve VEYA kapıları:
  - Değişme ve birleşme özelliği vardır.
  - Giriş sayısını artırmakta sorun yok.
- TÜVE ve TÜVEYA kapıları
  - Değişme özelliği vardır, ancak birleşme özelliği yoktur.
  - Giriş sayısını artırmak kolay değil.
- Örnek: TÜVE kapıları
  - $((x\ y)'z)' \neq (x(yz)')'$
  - $((xy)'z)' = ((x'+y')z)' = xy+z'$



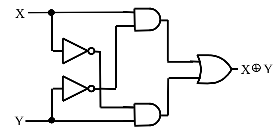
  - $(x\ (yz)')' = x'+yz$



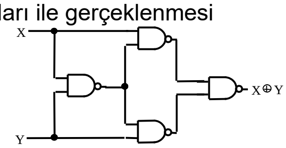
Dışlayıcı VEYA (XOR) Kapısının Gerçeklenmesi

- Çarpımlar toplamı şeklinde ifade edilirse:

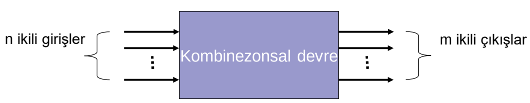
$x \oplus y = x'y + xy'$



- Sadece TÜVE Kapıları ile gerçeklenmesi



Kombinezonsal Devreler



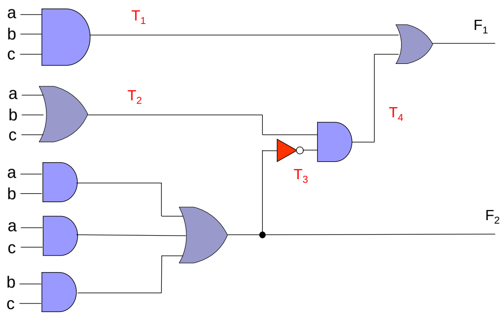
- n ikili girişle → 2^n mümkün giriş kombinezonu
- Her giriş kombinezonu için mümkün bir çıkış değeri var.
  - Doğruluk tablosu
  - Boole fonksiyonu

Kombinezonsal devrelerin analizi

- Analiz: bir devrenin gerçeklediği Boole fonksiyonunun bulunması
  - Bir lojik devre veriliyor
  - Bulunması gerekenler
    1. Boole fonksiyonu
    2. Doğruluk tablosu
    3. Devrenin işlevi hakkında bilgi

Boole Fonksiyonunun Bulunması

Örnek



Örnek: Boole Fonksiyonunun Bulunması

- Gösterilen noktaların Boole fonksiyonları
  - $T_1 = abc$
  - $T_2 = a + b + c$
  - $F_2 = ab + ac + bc$
  - $T_3 = F_2' = (ab + ac + bc)'$
  - $T_4 = T_3T_2 = (ab + ac + bc)' (a + b + c)$
  - $F_1 = T_1 + T_4$

$= abc + (ab + ac + bc)' (a + b + c)$

$= abc + ((a' + b')(a' + c')(b' + c')) (a + b + c)$

$= abc + ((a' + a'c' + a'b' + b'c')(b' + c')) (a + b + c)$

$= abc + (a'b' + a'c' + a'b'c' + a'b' + a'b'c' + b'c' + b'c') (a + b + c)$

Örnek: Doğruluk tablosunun elde edilmesi

$F_1 = a \oplus b \oplus c$

$F_2 = ab + ac + bc$

elde toplam

a	b	c	T_1	T_2	T_3	T_4	F_2	F_1
0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	1	0	1
0	1	0	0	1	1	1	0	1
0	1	1	0	1	0	0	1	0
1	0	0	0	1	1	1	0	1
1	0	1	0	1	0	0	1	0
1	1	0	0	1	0	0	1	0
1	1	1	1	1	0	0	1	1

Tam toplayıcı (TT)

İşaretli Sayıların Gösterilmesi

- Pozitif ve negatif sayıları ayırt etmek için ikili sayının en yüksek anlamlı bitine bakılır.
  - "0" ise pozitif
  - "1" ise negatif
- 8 bit ile gösterilebilecek pozitif sayılar 0000 0000 ile 0111 1111 yani 0 ile + 127 arasında değişecektir.
- Negatif sayıların gösteriminde 2'ye tümleme yöntemi kullanılır.
  - Pozitif bir sayının 2'ye tümleyeni hesaplandığında o sayının negatif gösterilimi elde edilmiş olur.
- Bir sayının 2'ye tümleyenini elde etmek için
  - Sayı 1'e tümlenir, yani 0'lar 1, 1'ler 0 yapılır.
  - 1'e tümlenmiş sayıya 1 eklenir.

Negatif Sayılara Örnekler

8 bitlik 5_{10} sayısı 5 mod 256 olarak düşünülebilir.

-5_{10} mod 256 = 256-5 mod 256=251 mod 256

```
  1 0000 0000
– 0 0000 0101
  0 1111 1011
    Negatif sayı
```

```
                 0000 0101
1'e tümleme      1111 1010
1 ekleme       +         1
2'ye tümleme     1111 1011
                 Negatif sayı
```


Negatif Sayılara Örnekler

5_{10} mod 256 = 256-(-5) mod 256= 256-251 mod 256

```
  1 0000 0000
– 0 1111 1011
  0 0000 0101
    Pozitif sayı
```

```
                 1111 1011
1'e tümleme      0000 0100
1 ekleme       +         1
2'ye tümleme     0000 0101
                 Pozitif sayı
```

İkili Sayıların Uzatılması

- Bazı durumlarda daha az bit ile ifade edilen bir sayıyı daha büyük bir yere yazmak ya da daha uzun bir sayı ile işleme sokmak gerekebilir.
- Bu durumda sayı uzatılır.
- İşaretsiz sayılar: Sayının başına gerektiği kadar sıfır '0' eklenir.
  - Örnek: 4 bitlik 3_{10}: 0011 8 bitlik 3_{10}: 0000 0011
- İşaretli sayılar: Sayının başına sayının işareti gerektiği kadar eklenir. Buna işaret uzatma denir.
  - Örnek: 4 bitlik 3_{10}: 0011 8 bitlik 3_{10}: 0000 0011
  - Örnek: 4 bitlik -7_{10}: 1001 8 bitlik -7_{10}: 1111 1001

İkili Matematik

- Elde ile bir bit uzunluklu toplama
- Birden fazla bit uzunluklu toplama
- Borç ile bir bit uzunluklu çıkartma
- Birden fazla bit uzunluklu çıkartma
- Çarpma

Elde ile bir bit uzunluklu toplama

Toplanacak iki basamak (X,Y), elde girişi (Z) kullanılarak toplama yapıldığında aşağıdaki toplam (S) ve elde çıkışı (C) elde edilir:

Elde girişi (Z) 0 ise:

Z	0	0	0	0
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 0	0 1	0 1	1 0

Elde girişi (Z) 1 ise:

Z	1	1	1	1
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 1	1 0	1 0	1 1

İşaretsiz sayıların toplanması

Elde	00000		01100	
X	01100	12	10110	22
Y	+10001	+17	+10111	+23
Toplam	11101	29	101101	45

- Not: En düşük anlamlı basamağın elde girişi her zaman '0' dır.
- n-bitlik iki sayı toplandığında sonuç n+1-bitliktir.

İşaretli sayıların toplanması

Elde	0010		0100	
X	1101	-3	0011	3
Y	+0001	+1	+0010	+2
Toplam	1110	-2	0101	5

Sonuç negatif

Sonuç pozitif

İşaretli sayıların toplanması

Elde	11110		11100	
X	1101	-3	0011	3
Y	+1111	-1	+1110	-2
Toplam	11100	-4	10001	1

Sonuç negatif

İhmal edilir

Sonuç pozitif

İhmal edilir

Elde	1000		0000	
X	0100	4	1010	-6
Y	+0101	+5	+1101	-3
Toplam	1001	9	10111	-9

Sonuç negatif midir? İhmal edilir Sonuç pozitif midir?

•Taşma oluşmuştur. 4-bit ile gösterilebilen en büyük pozitif sayı +7 dir. Daha büyük sayılar 4-bit ile gösterilemez.
•4-bit ile gösterilebilen mutlak değeri en büyük negatif sayı -8 dir. Mutlak değeri daha büyük olan negatif sayılar 4-bit ile gösterilemez.
•Sayıların hangi bit uzunluğu ile gösterileceğine yapılacak işlemlere ve bu işlemler sonucunda ortaya çıkması olası olan sonuçların sınırlarına göre karar verilmelidir.

Borç ile bir bit uzunluklu çıkartma

- **Çıkarma işlemi yapılacak iki basamak (X,Y), borç girişi (Z) kullanılarak çıkarma yapıldığında aşağıdaki fark (S) ve borç çıkışı (B) elde edilir:**
- **Borç girişi (Z) 0 ise:**

Z	0	0	0	0
X	0	0	1	1
- Y	-0	-1	-0	-1
BS	0 0	1 1	0 1	0 0

- **Borç girişi (Z) 1 ise:**

Z	1	1	1	1
X	0	0	1	1
- Y	-0	-1	-0	-1
BS	11	1 0	0 0	1 1

İşaretsiz sayılar ile çıkartma

Borç	00000		00110	
X	10110	22	10110	22
Y	- 10010	-18	- 10011	-19
Fark	00100	4	00011	3

- Not: En düşük anlamlı basamağın borç girişi her zaman '0' dır. Eğer Y>X ise X ve Y yer değiştirilir ve sonucun başına – işareti eklenir.

İşaretli sayılar ile 2'ye tümleme kullanılarak çıkartma

X	3	0011		0011	3
Y	-1	-0001	2'ye tümleyen	+1111	+(-1)
Fark	2			10010	2

Sonuç pozitif

İhmal edilir

X	3	0011		0011	3
Y	-4	-0100	2'ye tümleyen	+1100	+(-4)
Fark	-1			1111	-1

Sonuç negatif

X	3	0011		0011	3
Y	-(-1)	-1111	2'ye tümleyen	+0001	+1
Fark	4			0100	4

Sonuç pozitif

X	1	0001		0001	1
Y	-(-7)	-1001	2'ye tümleyen	+0111	+7
Fark	8			1000	8

Sonuç negatif midir?

X	-5	1011		1011	-5
Y	-4	-0100	2'ye tümleyen	+1100	+(-4)
Fark	-9			10111	-9

Sonuç pozitif midir?

İhmal edilir

•Taşma oluşmuştur. 4-bit ile gösterilebilen en büyük pozitif sayı +7 dir. Daha büyük sayılar 4-bit ile gösterilemez.
•4-bit ile gösterilebilen mutlak değeri en büyük negatif sayı -8 dir. Mutlak değeri daha büyük olan negatif sayılar 4-bit ile gösterilemez.

İkili Çarpma

İkili çarpım tablosu:

0 ∗ 0 = 0 | 1 ∗ 0 = 0 | 0 ∗ 1 = 0 | 1 ∗ 1 = 1

Çarpmayı birden çok bit uzunluklu sayılar ile yapma:

Çarpılan	**1011**
Çarpan	**x 101**
Ara çarpım	**1011**
	0000 -
	1011 - -
Çarpım	**110111**

Kombinezonsal Devrelerin Tasarımı

- Problemin sözle tanımı
  - Sözle tanımlar genellikle tam değildir ve hatalıdır.
  - Yanlış anlama yanlış devre tasarımı ile sonuçlanır.
- Bulmamız gerekenler
  1. Doğruluk tablosu
  2. Boole fonksiyonu
  3. Boole fonksiyonunu gerçekleyen minimal devre

Toplama Devresi

- 2-bitlik sayıların toplanması
  - $Z = X + Y$
  - $X = (x_1\ x_0)$ and $Y = (y_1\ y_0)$
  - $Z = (z_2\ z_1\ z_0)$
- Bitlerin toplanması
  1. $z_0 = x_0 \oplus y_0$

     $e_1 = x_0\ y_0$ (elde)
  2. $z_1 = x_1 \oplus y_1 \oplus e_1$

     $e_2 = x_1\ y_1 + x_1\ e_1 + y_1\ e_1$
  3. $z_2 = e_2$

x_i	y_i	e_i	z_i	e_{i+1}
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

Toplama Devresi

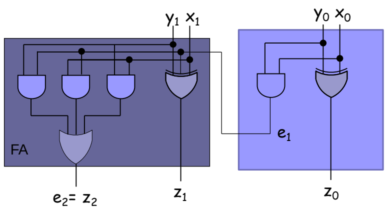
$z_2 = e_2$

$z_1 = x_1 \oplus y_1 \oplus e_1$

$e_2 = x_1\ y_1 + x_1\ e_1 + y_1\ e_1$

$z_0 = x_0 \oplus y_0$

$e_1 = x_0\ y_0$



Tam Toplayıcı

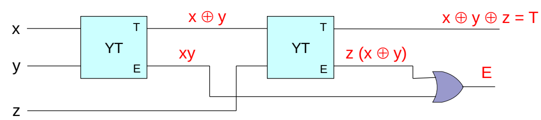
Yarı Toplayıcı

Tam Toplayıcı: Yarı Toplayıcılar ile Gerçekleme

- Toplam
  - $T = x \oplus y \oplus z$
- Elde
  - $E = xy + xz + yz$

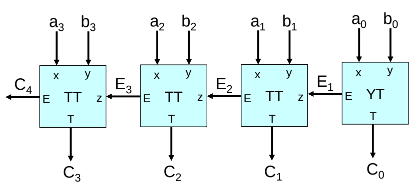
    $= (x + y)\ z + xy$

    $= (x \oplus y)\ z + xy$

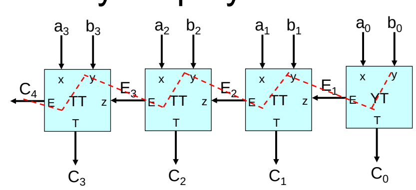


Tamsayı Toplayıcı 1/2

- <u>İkili Toplayıcı</u>:
  - $A = (a_{n-1}, a_{n-2}, \ldots, a_1, a_0)$
  - $B = (b_{n-1}, b_{n-2}, \ldots, b_1, b_0)$
  - $A + B = C = (c_n, c_{n-1}, c_{n-2}, \ldots, c_1, c_0)$
- Basit hal: 4-bit ikili toplayıcı



Tamsayı Toplayıcı 2/2



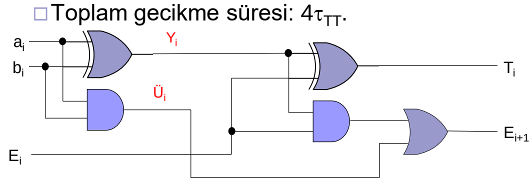
Elde Zincirli Toplama (Ripple-carry adder)

Aşamalı Tasarım Yöntemi

- Elde zincirli toplayıcı tasarımında aşamalı tasarım yöntemi kullandık.
- Klasik tasarımda aşağıdaki haller var.
  - 8 giriş
  - 5 çıkış
  - 2^9 = 512 satırlı beş doğruluk tablosu
  - 9 değişkenli 5 Boole fonksiyonunu indirgemeliyiz.
- Aşamalı Tasarımda
  - Tasarımı daha küçük işlem bloklarına ayırıyoruz.
  - Küçük işlem bloklarını birbirine bağlayarak daha büyük fonksiyonu gerçeklemek istiyoruz.

Elde Yayılımı

- 4-bitlik elde zincirli toplayıcının toplam gecikme süresi nedir?
  - τ_{TT}: bir tam toplayıcının gecikme süresi
  - Kaskat şekilde bağlanmış 4 tam toplayıcı kullanıldı.
  - Toplam gecikme süresi: $4\tau_{TT}$.



$4\tau_{TT} \approx 8\tau_{XOR}$

Hızlı Toplayıcılar

- Elde yayılımı iki sayının toplanmasında hızı sınırlayan sebeptir.
- İki seçenek
  - Düşük gecikmeli kapılar kullanmak.
  - Elde gecikmesini azaltacak şekilde devre karmaşıklığını artırmak.
- Elde öngörülü toplayıcı (carry lookahead adders) ikinci seçeneğe bir örnektir.
  - İki değişken:
    1. $Y_i = a_i \oplus b_i$ – elde yayılımı
    2. $Ü_i = a_i b_i$ – elde üretimi

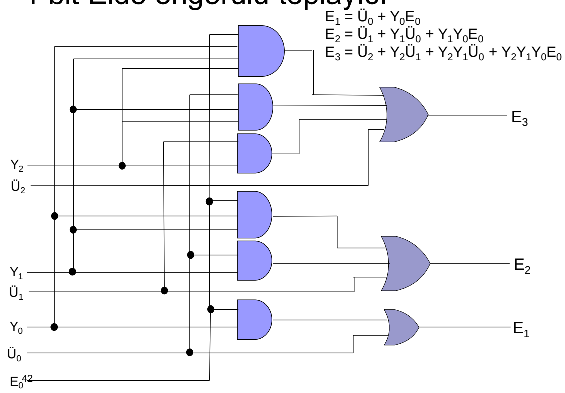
Elde öngörülü toplayıcı

- Toplam ve elde P_i ve G_i cinsinden ifade edilebilir:
  - $T_i = Y_i \oplus E_i$
  - $E_{i+1} = Ü_i + Y_i E_i$
- Neden elde yayılımı ve üretimi?
  - Eğer $Ü_i = 1$ ($a_i = b_i = 1$), yeni bir elde üretilir.
  - Eğer $Y_i = 1$ ($a_i = 1$ or $b_i = 1$), bir önceki basamaktan gelen elde bir sonraki basamağa yayılır.

4-bit Elde öngörülü toplayıcı

- Elde yayılımı ve üretimi işaretlerini kullanarak elde bitleri hesaplanabilir.
  - E_0 = giriş
  - $E_1 = Ü_0 + Y_0 E_0$
  - $E_2 = Ü_1 + Y_1 E_1$
    $= Ü_1 + Y_1(Ü_0 + Y_0 E_0) = Ü_1 + Y_1 Ü_0 + Y_1 Y_0 E_0$
  - $E_3 = Ü_2 + Y_2 E_2 = Ü_2 + Y_2(Ü_1 + Y_1 Ü_0 + Y_1 Y_0 E_0)$
    $= Ü_2 + Y_2 Ü_1 + Y_2 Y_1 Ü_0 + Y_2 Y_1 Y_0 E_0$
  - $Y_0 = a_0 \oplus b_0$ ve $Ü_0 = a_0 b_0$
  - $Y_1 = a_1 \oplus b_1$ ve $Ü_1 = a_1 b_1$
  - $Y_2 = a_2 \oplus b_2$ ve $Ü_2 = a_2 b_2$
  - $Y_3 = a_3 \oplus b_3$ ve $Ü_3 = a_3 b_3$

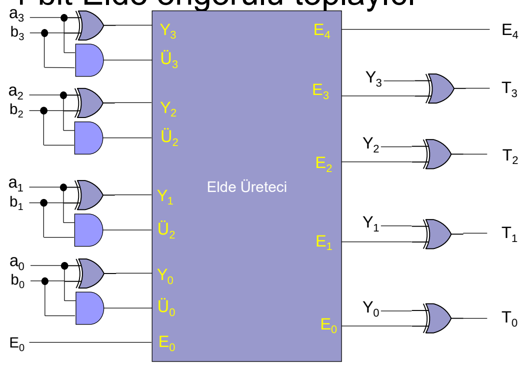
4-bit Elde öngörülü toplayıcı



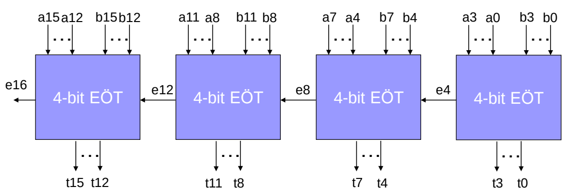
4-bit Elde öngörülü toplayıcı

- Bütün eldeler (E_1, E_2, E_3) iki seviyeli şekilde (VE-VEYA) gerçeklenebilir.
- E_3 E_2 ve E_1 in yayılımını beklemek zorunda değildir.

4-bit Elde öngörülü toplayıcı

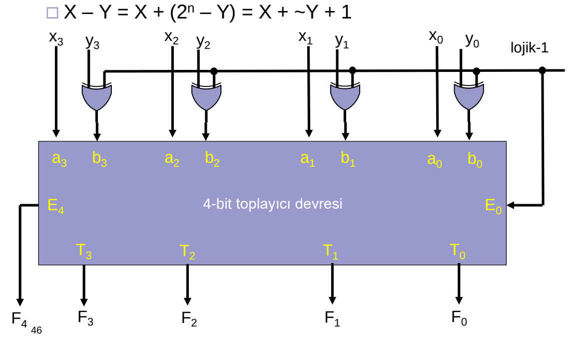


16-bit Melez Toplayıcı



Çıkarma Devresi

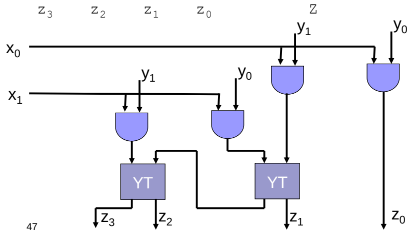
- İkiye tümleyen ile nasıl toplama yaptığımızı hatırlayalım
  - $X – Y = X + (2^n – Y) = X + \sim Y + 1$



İkili Çarpıcı

- 2-bitlik çarpıcı

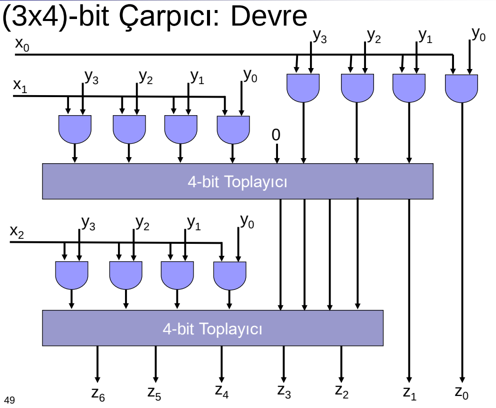
			y_1	y_0	Y
		×	x_1	x_0	X
			$x_0\ y_1$	$x_0\ y_0$	
+		$x_1\ y_1$	$x_1\ y_0$		
	z_3	z_2	z_1	z_0	Z



(3x4)-bit Çarpıcı: Yöntem

				y_3	y_2	y_1	y_0	Y
			×		x_2	x_1	x_0	X
				$x_0\ y_3$	$x_0\ y_2$	$x_0\ y_1$	$x_0\ y_0$	
			$x_1\ y_3$	$x_1\ y_2$	$x_1\ y_1$	$x_1\ y_0$		
+		$x_2\ y_3$	$x_2\ y_2$	$x_2\ y_1$	$x_2\ y_0$			
	z_6	z_5	z_4	z_3	z_2	z_1	z_0	

(3x4)-bit Çarpıcı: Devre



49

mxn-bit Çarpıcılar

- çarpılan: m-bit tamsayı
- çarpan: n-bit tamsayı
- m×n VE kapısı
- (m-1) toplayıcı
  - Her toplayıcı n-bit

50

2'ye Tümleyen Gösterilimli İşaretli Sayıların Çarpımı (1/2)

- 3-bit genlik, 1-bit işaret, toplam 4-bit
- En büyük çarpan +7
- En büyük çarpım sonucu $+49_{10}=0110001_2$
- Bütün sayılar 7-bit ile gösterilmeli

Onluk	İkilik
+0	0000000
+1	0000001
+2	0000010
+3	0000011
+4	0000100
+5	0000101
+6	0000110
+7	0000111

Onluk	2'ye Tümleyen
-0	000000
-1	1111111
-2	1111110
-3	1111101
-4	1111100
-5	1111011
-6	1111010
-7	1111001

51

2'ye Tümleyen Gösterilimli İşaretli Sayıların Çarpımı (2/2)

- 3×(-4)= -12
- $0000011_2 \times 1111100_2$=?

```
      1 1 1 1 1 0 0
    × 0 0 0 0 0 1 1
    ---------------
      1 1 1 1 1 0 0
  +   1 1 1 1 1 0 0
  -----------------
    1 0 1 1 1 0 1 0 0
```

ihmal negatif

Sonucun mutlak değeri

$0001011+1=0001100_2$

- (-5)×(-6)= +30
- $1111011_2 \times 1111010_2$=?

```
                    1 1 1 1 0 1 1
                  × 1 1 1 1 0 1 0
                  ---------------
                    0 0 0 0 0 0 0
                  1 1 1 1 0 1 1
                0 0 0 0 0 0 0
              1 1 1 1 0 1 1
            1 1 1 1 0 1 1
          1 1 1 1 0 1 1
  +     1 1 1 1 0 1 1
  -------------------------------
        1 1 1 0 1 0 1 0 0 1 1 1 1 0
```

ihmal pozitif

52

4-bit Genlik Karşılaştırma Devresi

- İki tamsayının karşılaştırılması: A ve B.
  - A > B ➔ (1, 0, 0) = (x, y, z)
  - A = B ➔ (0, 1, 0) = (x, y, z)
  - A < B ➔ (0, 0, 1) = (x, y, z)
- Örnek: 4-bit karşılaştırıcı
  - A = (a_3, a_2, a_1, a_0) and B = (b_3, b_2, b_1, b_0)
  1. (A = B) olması için
     - bütün $a_i = b_i$ $\quad 0 \le i \le 3$
     - $t_i = (a_i \oplus b_i)'$ $\quad 0 \le i \le 3$
     - y = (A=B) = $t_3 t_2 t_1 t_0$

53

4-bit Karşılaştırma Devresi

2. (A > B) and (A < B) cases
   - A ve B nin en yüksek anlamlı bitleri karşılaştırılır.
     - eğer (a_3 = 1 ve b_3 = 0) ➔ A > B
     - değilse eğer (a_3 = 0 ve b_3 = 1) ➔ A < B
     - değilse ($a_3 = b_3$) a_2 ve b_2 yi karşılaştır.

x = (A>B) = $a_3 b_3' + t_3 a_2 b_2' + t_3t_2 a_1 b_1' + t_3t_2t_1 a_0 b_0'$

z = (A<B) = $a_3' b_3 + t_3 a_2' b_2 + t_3t_2 a_1' b_1 + t_3t_2t_1 a_0' b_0$

y = (A=B) = $t_3 t_2 t_1 t_0$

54

4-bit Karşılaştırma Devresi

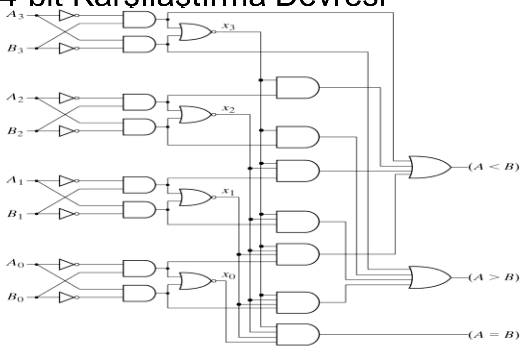
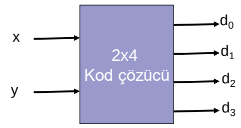


Fig. 4-17 4-Bit Magnitude Comparator

55

Kod Çözücü

- n-bitlik bir kod ile
  - 2^n kodlanmış bilgi gösterilebilir.
  - Bir kod çözücü n ikili girişi 2^n çıkışa dönüştüren kombinezonsal devredir.



x	y	d_0	d_1	d_2	d_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- $d_0 = x'y'$
- $d_1 = x'y$
- $d_2 = xy'$
- $d_3 = xy$

56

2-den-4'e Kod Çözücü

- Aktif çıkış 0 olabilir.
- Devrenin çalışmasını kontrol etmek için bir de izin (enable) girişi olabilir.

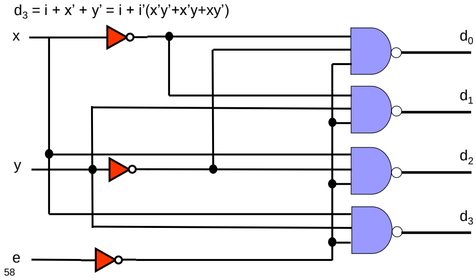
i	x	y	d_0	d_1	d_2	d_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

- $d_0 = i + x + y = i + i'(x'y+xy'+xy)$
- $d_1 = i + x + y' = i + i'(x'y'+xy'+xy)$
- $d_2 = i + x' + y = i + i'(x'y'+x'y+xy)$
- $d_3 = i + x' + y' = i + i'(x'y'+x'y+xy')$

57

İzin Girişli 2-den-4'e Kod Çözücü

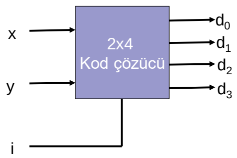
$d_0 = i + x + y = i + i'(x'y+xy'+xy)$
$d_1 = i + x + y' = i + i'(x'y'+xy'+xy)$
$d_2 = i + x' + y = i + i'(x'y'+x'y+xy)$
$d_3 = i + x' + y' = i + i'(x'y'+x'y+xy')$

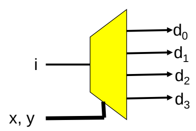


58

Kod Çözücü/Veri Dağıtıcı

- Veri dağıtıcı
  - Bir hattan bilgiyi alır ve 2^n çıkıştan birine yönlendirir.
  - Hangi çıkışın veriyi alacağını gösteren n seçim girişi vardır.

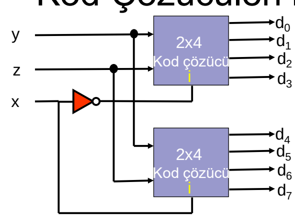




x = 0 ve y = 0 ise d_0 = i
x = 0 ve y = 1 ise d_1 = i
x = 1 ve y = 0 ise d_2 = i
x = 1 ve y = 1 ise d_3 = i

59

Kod Çözücüleri Birleştirme

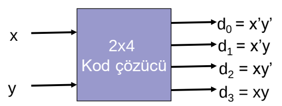


x y z	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
0 0 0	0	0	0	0	0	0	0	1
0 0 1	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	0	1	0	0
0 1 1	0	0	0	0	1	0	0	0
1 0 0	0	0	0	1	0	0	0	0
1 0 1	0	0	1	0	0	0	0	0
1 1 0	0	1	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0

60

Kod Çözücünün Gerçeklemede Kullanılması

- Kod çözücü n giriş için 2^n çarpım terimini verir.

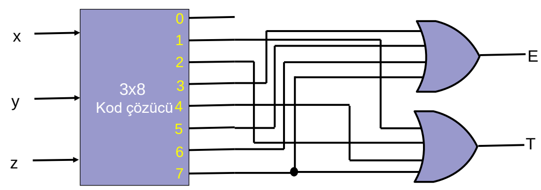


- nx2^n Kod çözücü ve VEYA kapıları kullanarak çarpımlar toplamı ifade kullanılarak gösterilen n değişkenli bütün Boole fonksiyonları gerçeklenebilir.

61

Örnek

- Tam Toplayıcı
  - $E = xy + xz + yz = \Sigma(3, 5, 6, 7)$
  - $T = x \oplus y \oplus z = \Sigma(1, 2, 4, 7)$



62

Kodlayıcı

- Giriş sayısı: 2^n
- Çıkış sayısı: n
- Çıkışlarda giriş değerine bağlı olarak kod üretilir
- Örnek: n = 2

d_0	d_1	d_2	d_3	x	y
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

63

Öncelikli Kodlayıcı

- Kodlayıcıdaki problemler:
  - Girişlerden bir anda sadece bir tanesi aktif olabilir.
  - Aynı anda birden fazla giriş aktif olduğunda çıkış tanımsızdır.
- Öncelikli Kodlayıcı:
  - Girişler arasında öncelik tanımlanır.

d_0	d_1	d_2	d_3	x	y	v
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

64

4-bit Öncelikli Kodlayıcının Karnaugh Diyagramı

d_0d_1 \ d_2d_3	00	01	11	10
00	X	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

– $x = d_2 + d_3$

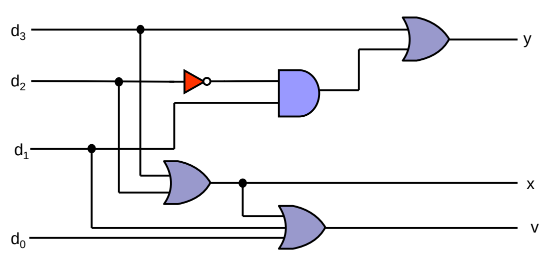
d_0d_1 \ d_2d_3	00	01	11	10
00	X	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

– $y = d_3 + d_1 d_2'$

65

4-bit Öncelikli Kodlayıcı Devresi

– $x = d_2 + d_3$
– $y = d_1d_2' + d_3$
– $V = d_0 + d_1 + d_2 + d_3$



66

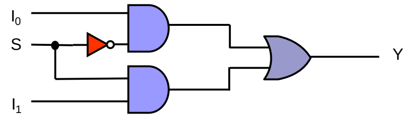
Veri Toplayıcılar (Multiplexers - MUX)

- Birçok girişinden birindeki veriyi tek çıkışına aktarır.
- Seçme girişleri n → n = ?
- Örnek: 2-den-1'e MUX
  - 2 giriş hattı I_0, I_1
  - 1 çıkış hattı Y
  - 1 seçim hattı S

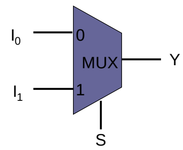
S	Y
0	I_0
1	I_1

$Y = S'I_0 + SI_1$

2-den-1'e Veri Toplayıcı



- Özel Sembol



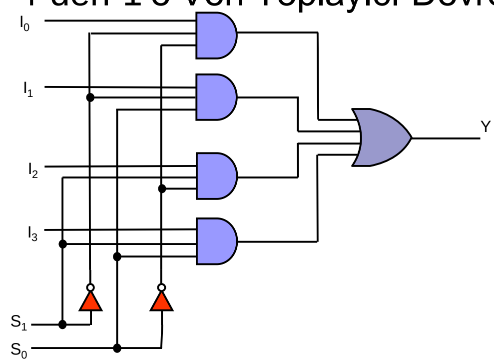
4-den-1'e Veri Toplayıcı

- 4 giriş hattı: I_0, I_1, I_2, I_3
- 1 çıkış hattı: Y
- 2 seçim hattı: S_1, S_0.

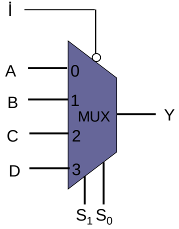
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$Y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$

4-den-1'e Veri Toplayıcı Devresi



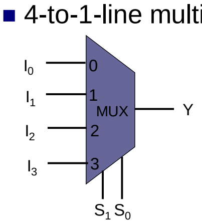
İzin Girişli Veri Toplayıcı



İ	S	Y
1	XX	0
0	00	A
0	01	B
0	10	C
0	11	D

Veri Toplayıcılar Kullanılarak Tasarım

- 4-to-1-line multiplexer



- S_1 → x
- S_0 → y
- $S_1'S_0' = x'y'$,
- $S_1'S_0 = x'y$,
- $S_1S_0' = xy'$,
- $S_1S_0 = xy$

- $Y = S_1'S_0' I_0 + S_1'S_0 I_1 + S_1S_0' I_2 + S_1S_0 I_3$.
- $Y = x'y' I_0 + x'y I_1 + xy' I_2 + xyI_3$

Veri Toplayıcılar Kullanılarak Tasarım

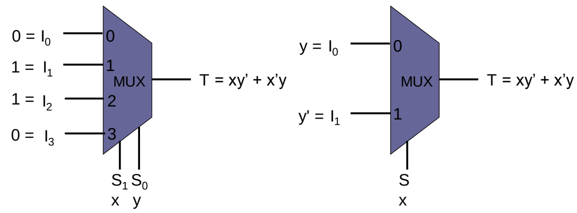
- n değişkenli Boole fonksiyonunu m seçim girişli MUX ile gerçeklemek için
  - $F(x_1, x_2, ..., x_n)$

1. Boole fonksiyonu doğruluk tablosu il gösterilir.
2. İlk m değişken ($x_1, x_2, ..., x_m$) seçim girişlerine uygulanır
3. Bu m değişkenin her kombinezonu için çıkış değeri geri kalan n-m değişkenin ($x_{m+1}, x_{m+2}, ..., x_n$) cinsinden bulunur.
   - $0, 1, x_{m+1}'x_{m+2}'\ldots x_n', x_{m+1}'x_{m+2}'\ldots x_n, \ldots, x_{m+1}x_{m+2}\ldots x_n$,
4. Bu fonksiyonlar veri girişlerine doğru sırada uygulanır.

73

Veri Toplayıcılar Kullanılarak Tasarım

- Örnek: $T = \Sigma(1, 2)$



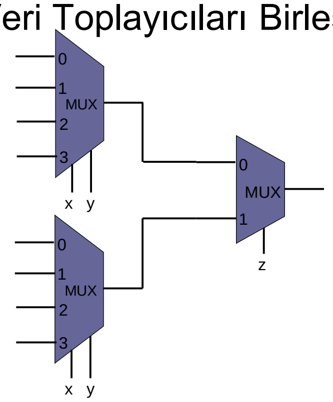
74

Veri Toplayıcılar Kullanılarak Tasarım

- $F(x, y, z) = \Sigma(1, 2, 6, 7)$
  - $F = x'y'z + x'yz' + xyz' + xyz$
- Bir seçim girişli MUX ile
  - $Y = S'I_0 + SI_1$
  - $S=x$, $I_0 = y'z + yz'$, $I_1 = y$
- İki seçim girişli MUX ile
  - $Y = S_1'S_0' I_0 + S_1'S_0 I_1 + S_1S_0' I_2 + S_1S_0 I_3$
  - $S_1=x$, $S_0=y$, $I_0 = z$, $I_1 = z'$, $I_2 = 0$, $I_3 = 1$
- Üç seçim girişli MUX ile
  - $Y = S_2'S_1'S_0' I_0 + S_2'S_1'S_0 I_1 + S_2'S_1S_0' I_2 + S_2'S_1S_0 I_3 + S_2S_1'S_0' I_4 + S_2S_1'S_0 I_5 + S_2S_1S_0' I_6 + S_2S_1S_0 I_7$
  - $S_2=x$, $S_1=y$, $S_0=z$, $I_0=0$, $I_1=1$, $I_2=1$, $I_3=0$, $I_4=0$, $I_5=0$, $I_6=0$, $I_7=1$

75

Veri Toplayıcıları Birleştirme



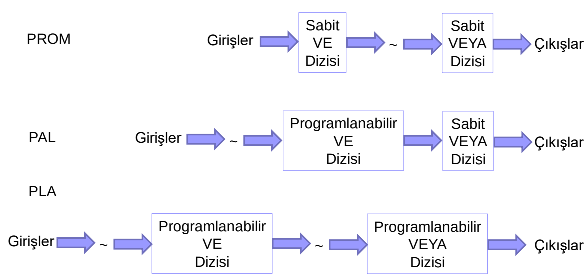
76

Programlanabilir Lojik Elemanlar - Programmable Logic Devices (PLD's)

- Programlanabilir lojik elemanlar VE ve VEYA kapı dizilerinden oluşurlar. Kapı dizileri özel bir bağlantı şekli oluşturmak amacıyla anahtarlar ile kontrol edilirler.
- Bu derste üç çeşit PLD inceleyeceğiz.
  1. Programlanabilir Salt Okunabilir Bellek - Programmable Read Only Memory (PROM)
  2. Programmable Logic Array (PLA)
  3. Programmable Array Logic (PAL)

77

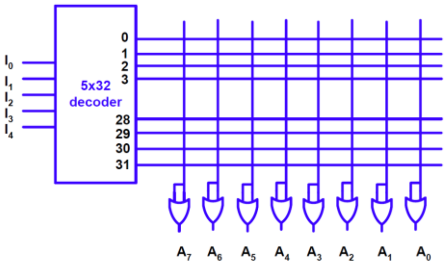
Programlanabilir Lojik Elemanlar



78

Salt Okunabilir Bellek - Read Only Memory (ROM)

- **ROM ikili bilginin saklanabildiği ve güç kaynağı kesilse bile bilgiyi koruyan bir elemandır.**
- **ROM bir kod çözücü ve bir sabit VEYA kapısı dizisinden oluşur.**



32x8 bitlik ROM'un iç yapısı

ROM Kullanarak Kombinezonsal Devre Tasarımı

- Boole fonksiyonunun doğrudan gerçeklenmesi
  - Fonksiyonu indirgemeye gerek yok. Bütün çarpım terimleri üretiliyor.
- Yeniden programlama aynı cihaz ile farklı Boole fonksiyonlarının gerçeklenebilmesine olanak sağlar.

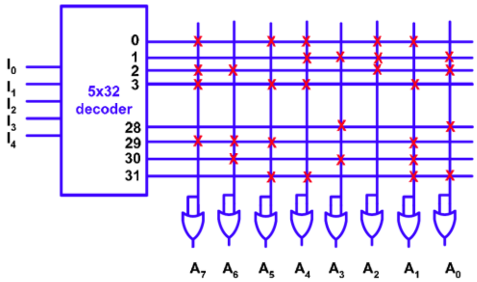
ROM Kullanarak Kombinezonsal Devre Tasarımı

- ROM ile gerçeklenecek olan Boole fonksiyonunun doğruluk tablosu kapalı olan anahtarların yerlerini gösterir.

Girişler					Çıkışlar							
I_4	I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
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

ROM Kullanarak Kombinezonsal Devre Tasarımı

- X bağlantının olduğunu yani lojik-1'i gösterir. X olmaması bağlantının olmadığını yani lojik-0'ı gösterir.



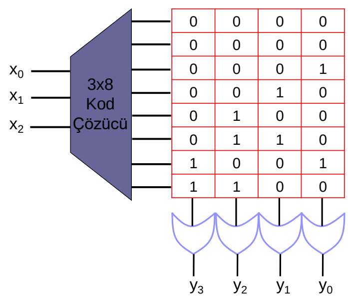
Örnek

- 3-bitlik girişindeki sayının karesini bulan devreyi ROM kullanarak tasarlayınız.
- Giriş bit uzunluğu, çıkış bir uzunluğu ve çıkışlara ait doğruluk tablosu bulunacak.
- Bu devrede 3-bitlik giriş ve 6 bitlik-çıkış vardır. $7^2 = 49 = 110001_2$.
- Doğruluk Tablosu:

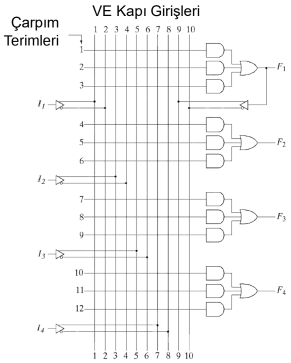
x_2	x_1	x_0	y_5	y_4	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

Örnek

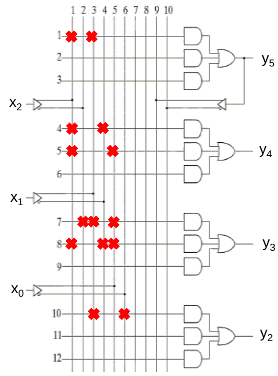
- Doğruluk tablosundan $y_0 = x_0$ ve $y_1 = 0$ olarak bulunur.
- Gerekli olan ROM un büyüklüğü 8 X 4 dür.



Programmable Array Logic (PAL)



PAL ile Tasarım



$y_5 = x_2x_1$

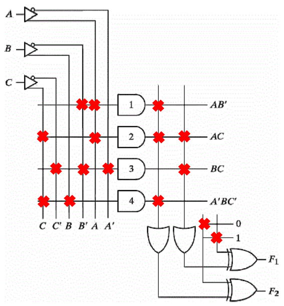
x_0 \ x_2x_1	00	01	11	10
0				1
1			1	1

$y_4 = x_2x_1' + x_2x_0$

$y_3 = x_2'x_1x_0 + x_2x_1'x_0$

$y_2 = x_1x_0'$

Programmable Logic Array (PLA)



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