



1. HAFTA

BLM320

BİLGİSAYAR MİMARİSİ

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU

sgorgunoglu@karabuk.edu.tr

KBUZEM

Karabük Üniversitesi

Uzaktan Eğitim Uygulama ve Araştırma Merkezi

Konu Başlıkları

1. Sayısal mantık devreler
2. Sanal Devre Simulatörleri
3. Sayısal mantık devreler
4. Kaydedici yapısı
5. Hafıza yapısı
6. Kaydediciler arası veri transferi ve Mikroişlemler
7. Bellek, Aritmetik, Lojik ve Kaydırma Mikroişlemleri
8. Temel Bilgisayar Tasarımı
9. Temel Bilgisayar Komut Seti
10. Temel Bilgisayar Mikroişlemler ve Denetim Fonksiyonları
11. Programlama

Kaynaklar

- Mano, M.Morris, Computer System Architecture, Printice Hall, New Jersey, 1993
- Mano, M.Morris, Bilgisayar Sistemleri Mimarisi, Literatür yayıncılık (çeviri),İstanbul, 2002
- Eşref Adalı Mikroişlemciler Mikrobilgisayarlar, Sistem Yayıncılık, 1992
- Mano, M.Morris, Digital Design, Printice Hall
- <http://www.cizgi-tagem.org/e-kutuphane/topic.aspx?id=964&p=3>

1. Giriş

Mikrobilgisayar mikroişlemci, bellek ve giriş/çıkış elemanlarından oluşan bir yapıdır. Mikroişlemci, mikrobilgisayarın merkezi işlem birimi olarak çalışır. Mikroişlemci program komut kodlarının, bellekten alınıp getirilerek kodunun çözülmesi ve çalıştırılıp giriş/çıkış birimlerine yansıtılması işlevini yerine getirmektedir. En basit şekli ile bir mikroişlemcinin mimari yapısı kaydediciler, aritmetik-mantık birimi ve sistemi eşzamanlı çalıştırıp kontrol eden kontrol biriminden meydana gelmektedir.

Bilgisayar mimarisi , Bellek, ALU, kaydediciler, I/O birimi yol arabirimi, kontrol birimi gibi birimlerin tasarlanması, çalışma prensiplerinin öğrenilmesi, verilerin bu birimler arasında taşınması gibi işlemlerin nasıl yapıldığının öğrenilmesi ve bu birimlerin programlanması ile ilgilenir. Bu birimlerin tasarlanması ve donanımsal olarak gerçekleştirilmesi için sanal devre simülatörlerinden yararlanılabilir.








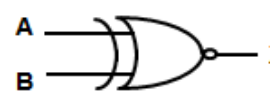
Bilgisayar mimarisinin öğrenilmesinde önemli sorunlardan birisi programlanabilen yapıyı öğrencilere tanımlayıp göstermekteki zorluktur. Dijital devre tasarım ve simülatörleri bu zorlukları aşmamız açısından büyük kolaylıklar sağlamaktadır. Bu simülatörler bize temel mikroişlemci yapısının en küçük elemanlarına kadar simülasyon yapmamıza olanak sağlamaktadır. Bu simülasyonlar sayesinde yeni tasarımların oluşumunda hem maliyetten hem de zamandan büyük kazanç sağlanmaktadır. Simülasyon yapılmadan oluşturulan sistemlerde hata ve arıza bulma konusunda da zorluklar çekilmektedir. Sistem tasarımı simüle edilerek gerekli görülen değişiklikler kolaylıkla yapılmaktadır. Sanal devre simülatörü olarak Logisim kullanılacaktır. Logisim sayısal mantık devrelerini tasarlamak ve simüle etmek için eğitimsel bir araçtır. Basit araç çubuğu arayüzü ve inşa ettiğiniz devrelerin simülasyonu ile, mantık devrelerine bağlı en temel kavramları öğrenmeyi kolaylaştırmak için yeterince basittir.

Küçük alt devrelerden büyük devreler inşa etme ve tek bir fare sürüklemesi ile kablo paketleri çizme yeteneği ile Logisim, eğitimsel amaçlar için tüm CPU' ları tasarlamak ve simüle etmek için kullanılmaktadır. Bilgisayar yapısını öğrenebilmek için temel sayısal devreleri iyi bilmek gerekmektedir.

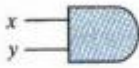
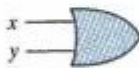


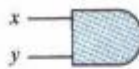


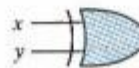
Sayısal Mantık Devreler

Kapı (Geçit) devreleri

Kapılar AND, OR, XOR gibi kapı devreleridir. Aşağıdaki tabloda temel kapılar ve doğruluk tabloları verilmiştir.

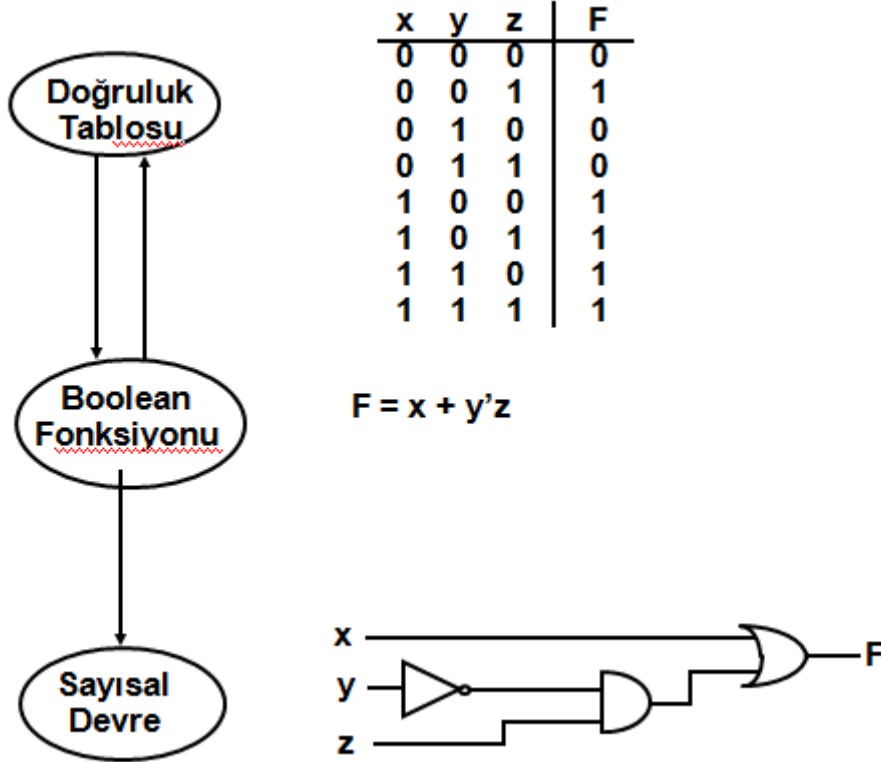
Name	Symbol	Function	Truth Table															
AND		$X = A \cdot B$ or $X = AB$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$X = A + B$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
I		$X = A$	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	
Buffer		$X = A$	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	X	0	0	1	1									
A	X																	
0	0																	
1	1																	
NAND		$X = (AB)'$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$X = (A + B)'$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR Exclusive OR		$X = A \oplus B$ or $X = A'B + AB'$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR Exclusive NOR or Equivalence		$X = (A \oplus B)'$ or $X = A'B' + AB$	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Boolean cebiri ve Lojik Kapılar

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = xy$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Boolean Cebiri

Dijital devrelerin tasarımında Boolean cebiri önem arzeder. Boolean cebiri ile ifade edilen fonksiyonların devre şemaları AND, OR ve inverter (I) kullanılarak çizilebilir.

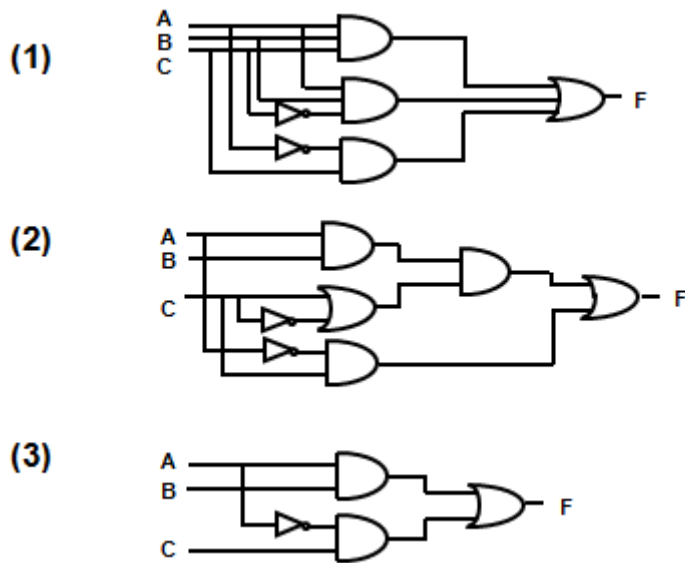


Boolean cebirin özellikleri

- | | |
|--------------------------------|------------------------------|
| 1 $x + 0 = x$ | 2 $x \cdot 0 = 0$ |
| 3 $x + 1 = 1$ | 4 $x \cdot 1 = x$ |
| 5 $x + x = x$ | 6 $x \cdot x = x$ |
| 7 $x + x' = 1$ | 8 $x \cdot x' = 0$ |
| 9 $x + y = y + x$ | 10 $xy = yx$ |
| 11 $x + (y + z) = (x + y) + z$ | 12 $x(yz) = (xy)z$ |
| 13 $x(y + z) = xy + xz$ | 14 $x + yz = (x + y)(x + z)$ |
| 15 $(x + y)' = x'y'$ | 16 $(xy)' = x' + y'$ |
| 17 $(x')' = x$ | |

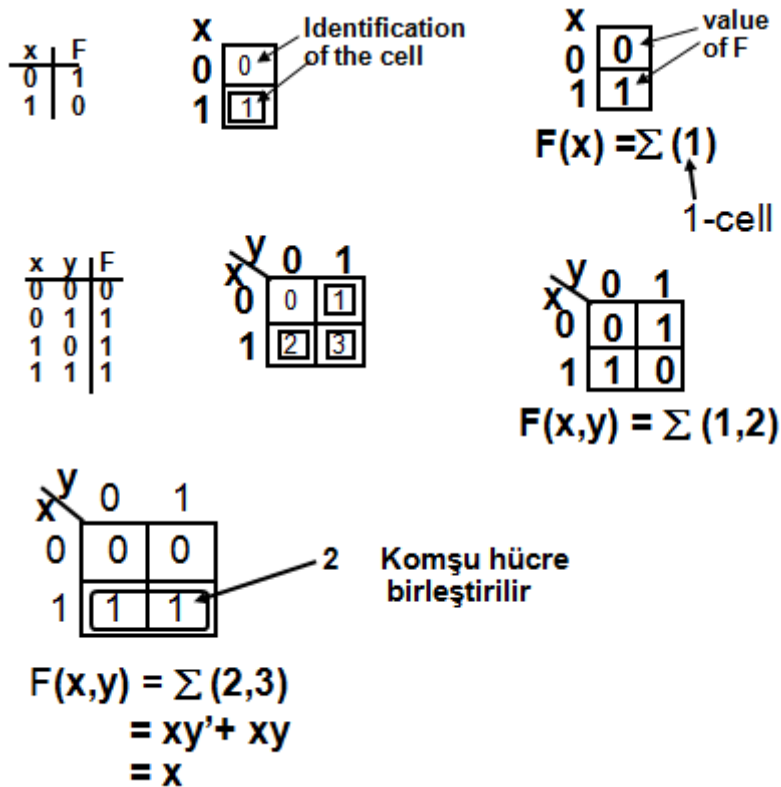
Örnekler

$$\begin{aligned}
 F &= ABC + ABC' + A'C & (1) \\
 &= AB(C + C') + A'C & (2) \\
 &= AB \cdot 1 + A'C \\
 &= AB + A'C & (3)
 \end{aligned}$$



Karnaugh haritaları

Sayısal devreleri basitleştirmek, sadeleştirmek için kullanılır.



x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

yz		y			
		00	01	11	10
x	0	0	1	3	2
	1	4	5	7	6

yz		y			
		00	01	11	10
x	0	0	1	0	1
	1	1	0	0	0

$$F(x,y,z) = \sum (1,2,4)$$

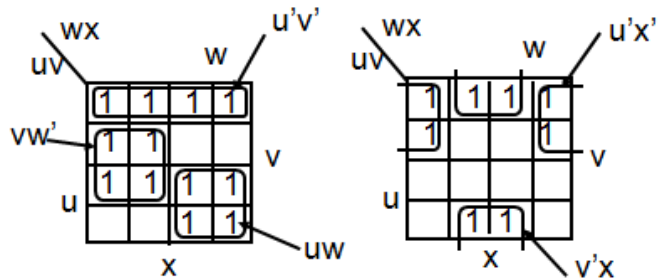
u	v	w	x	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

wx		w			
		00	01	11	10
uv	00	0	1	3	2
	01	4	5	7	6
u	11	12	13	15	14
	10	8	9	11	10

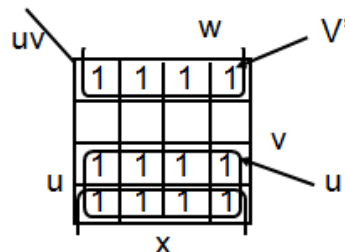
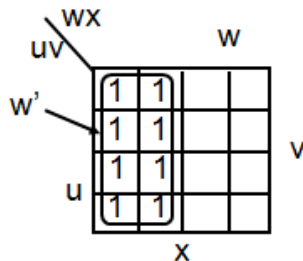
wx		w			
		00	01	11	10
uv	00	0	1	1	0
	01	0	0	0	1
u	11	0	0	0	1
	10	1	1	1	0

$$F(u,v,w,x) = \sum (1,3,6,8,9,11,14)$$

$$\begin{aligned}
 &u'v'w'x' + u'v'w'x + u'v'wx + u'v'wx' \\
 &= u'v'w'(x' + x) + u'v'w(x + x') \\
 &= u'v'w' + u'v'w \\
 &= u'v'(w' + w) \\
 &= u'v'
 \end{aligned}$$



$$\begin{aligned}
 &u'v'w'x' + u'v'w'x + u'vw'x' + u'vw'x + uvw'x' + uvw'x + uv'w'x' + uv'w'x \\
 &= u'v'w'(x' + x) + u'vw'(x' + x) + uvw'(x' + x) + uv'w'(x' + x) \\
 &= u'(v' + v)w' + u(v' + v)w \\
 &= (u' + u)w' = w'
 \end{aligned}$$



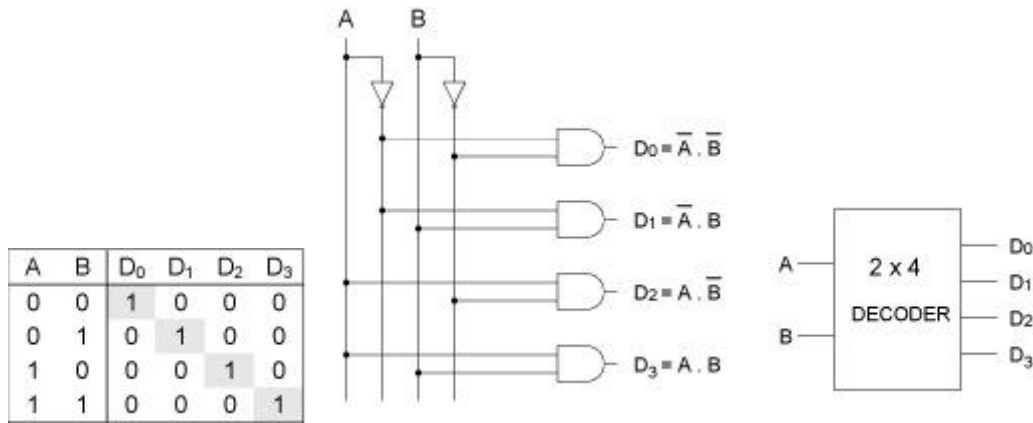
Sayısal mantık devreler

Kod Çözücüler (Decoders)

Sayısal sistemlerde bilgiler ikilik kodlarla tanımlanırlar. n bitlik bir ikilik kod ile 2^n kadar farklı durum tanımlanabilir. Bir kod çözücü, n giriş hattından gelen ikilik bilgileri maximum 2^n kadar farklı çıkış hattına dönüştüren birleşik bir devredir. Bir kod çözücünün n kadar girişi varsa 2^n kadar çıkışı vardır. Kullanılmayan veya dikkate alınmaz durumlar varsa kod çözücü çıkışı 2^n 'den az olacaktır.

İki Bitlik Kod Çözücü

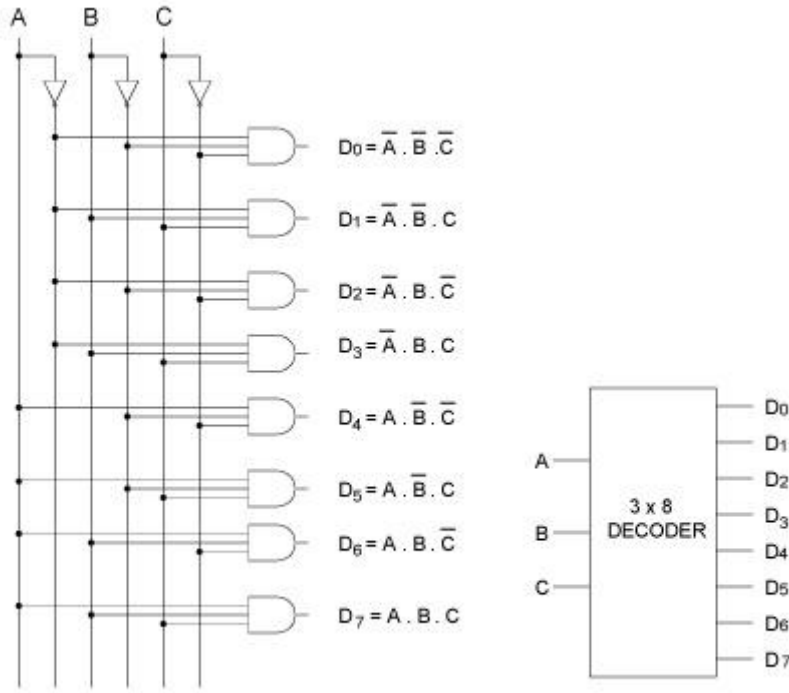
İki bitlik bir kod çözücünün 2 girişi 4 çıkışı vardır. Böyle bir devre için girişlerin durumuna bağlı olarak sadece tek bir çıkış doğru olacaktır. Aşağıda 2x4 Kod çözücünün doğruluk tablosu ,Lojik diyagramı ve sembolü verilmiştir.



Üç Bitlik Kod Çözücü

Üç bitlik kod çözücüde(decoder) üç girişin kodu çözülerek sekiz çıkış üretir. Her çıkış bu üç giriş değişkenine ait bir minimum terimle tanımlanır. Girişlerin durumuna bağlı olarak sadece tek bir çıkış doğrudur.

Decimal	A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1



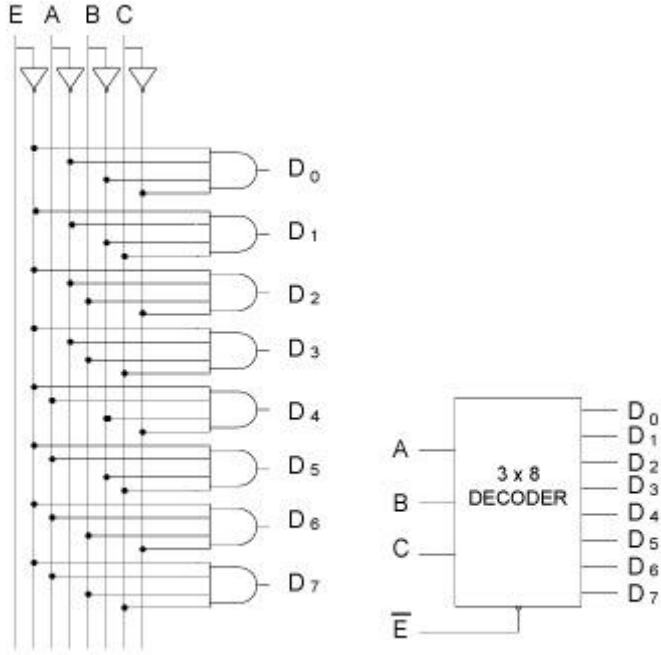
Yetki Girişli Kod Çözücüler

Kod çözücülerin tamamı olmasada büyük bir bölümü bir veya birden çok yetki (enable) girişi içerir. Kod çözücü (decoder) sadece yetkilendiğinde (enable girişine gelen Lojik-1 veya Lojik-0) kod çözme işlemini gerçekleştirir. Diğer bütün durumlarda kod çözücü (decoder) çıkışları Lojik-1 veya Lojik-0 olur.

Not: Yetkilendirme, çalışılan decoder özelliğine göre Lojik-1 veya Lojik-0 da olabilir.

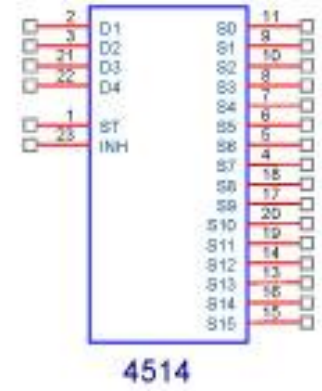
Aşağıda Lojik-0 da yetkilenen 3x8 Decoder'ın doğruluk tablosu verilmiştir.

E	A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
1	X	X	X	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	1	1	1	0	0	0	0	0	0	0	1



Aşağıda yetki giriqli 4x16 Decoder olan 4514 entegresinin doğruluk tablosu ve Lojik sembolü verilmiştir. Inhibit (INH) adlı giriş Decoder için yetkilendirme girişidir. Eğer bu giriş Lojik-1' e çekilmezse kodçözme işlemi gerçekleştirilmez ve bütün çıkışlar Lojik-0 olur. Harici bir giriş olan Strobe (ST) devre içindeki bir Latch'e (Mandal- Veri tutucu kumanda etmektedir. Eğer bu giriş Lojik-1'e çekilmezse diğer girişlerdeki (D,C,B,A) değişim ne olursa olsun eski durum korunacaktır.

ST	INH	D ₄	D ₃	D ₂	D ₁	S ₀	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	S ₈	S ₉	S ₁₀	S ₁₁	S ₁₂	S ₁₃	S ₁₄	S ₁₅
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

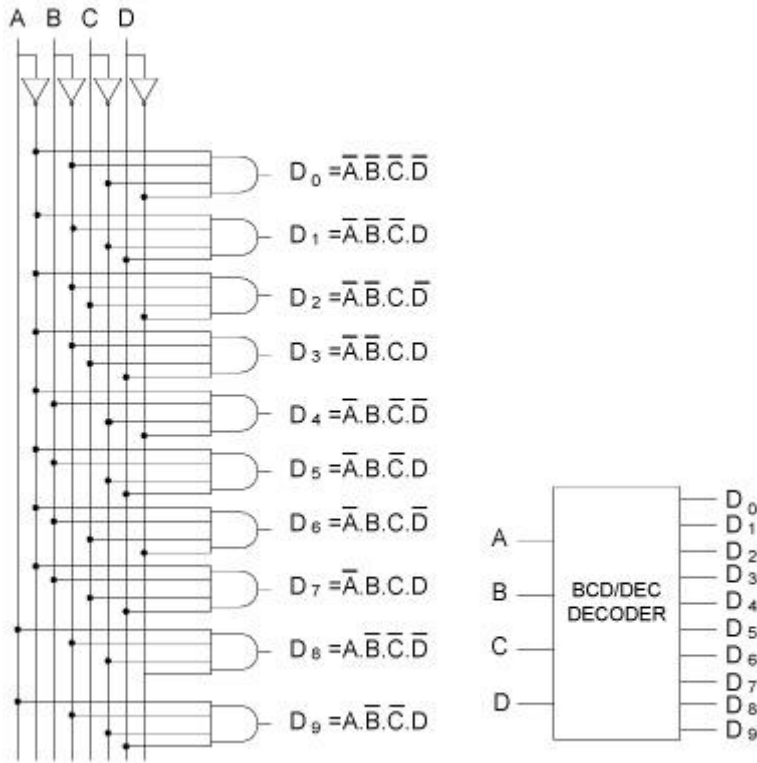


BCD Decimal Kod Çözücü

BCD kodu 0 – 9 arasındaki Decimal(Onluk) sayıların 4-Bitlik Binary(İkilik) karşılıklarının yazılması ile tanımlanmış bir kodlamadır. Bu durumda tasarlanacak kod çözücünün 4 giriş hattı olması, BCD kodu 0-9 arasındaki Decimal(Onluk) sayılar arasında tanımlı olduğundan 10 çıkış hattının olması gerekmektedir. Geri kalan durumlar don't care (dikkate alınmaz durumlar) olarak tanımlanacaktır.

Aşağıda BCD-Decimal Decoderin Lojik diyagramı ve Blok gösterimi verilmiştir.

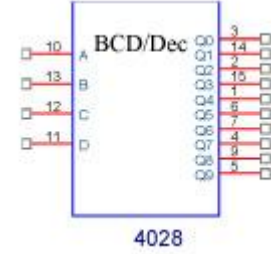
A	B	C	D	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈	D ₉
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	X	X	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X	X	X



BCD Decimal Decoder

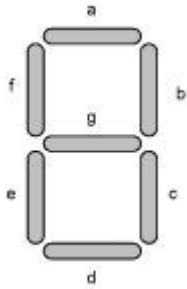
Aşağıda BCD - Decimal Kod çözücü olan 4028 entegresinin doğruluk tablosu ve Lojik sembolü verilmiştir.

D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	1	0
1	1	1	1	1	0	0	0	0	0	0	0	0	1

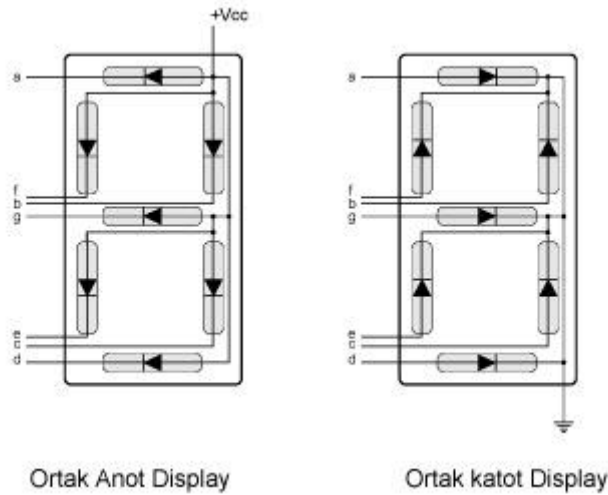


BCD Seven Segment Kod Çözücü

Yedi ayrı Led'in uygun bağlanması ile 0-9 arasındaki sayıları görüntüleyebiliriz. Bu işlemi yapan devre elemanına yedi parçalı gösterge(seven segment display) adı verilir. Display led bağlantılarına göre ortak anot veya ortak katot'lu olabilir.

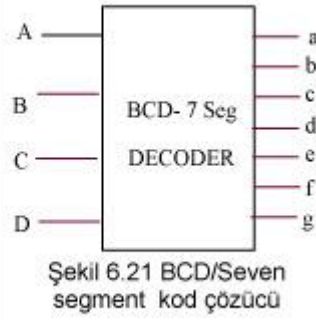


Yukardaki şekilde seven segment display gösterimi verilmiştir.



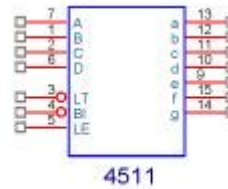
Aşağıda Ortak Katotlu Display için BCD- Seven Segment Decoder doğruluk tablosu verilmiştir.

Dec	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



Aşağıda ortak katotlu display için BCD- Seven Segment Decoder olan 4511 entegresinin doğruluk tablosu ve lojik sembolü verilmiştir.

Inputs							Outputs								Display
LE	$\overline{\text{BI}}$	$\overline{\text{LT}}$	D	C	B	A	a	b	c	d	e	f	g		
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B	
X	0	1	X	X	X	X	0	0	0	0	0	0	0		
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1	
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2	
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3	
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4	
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5	
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6	
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7	
0	1	1	1	0	0	0	0	1	1	1	1	1	1	8	
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9	
0	1	1	1	0	1	0	0	0	0	0	0	0	0		
0	1	1	1	0	1	1	0	0	0	0	0	0	0		
0	1	1	1	1	0	0	0	0	0	0	0	0	0		
0	1	1	1	1	0	1	0	0	0	0	0	0	0		
0	1	1	1	1	1	0	0	0	0	0	0	0	0		
0	1	1	1	1	1	1	0	0	0	0	0	0	0		
1	1	1	X	X	X	X	0	0	0	0	0	0	0		

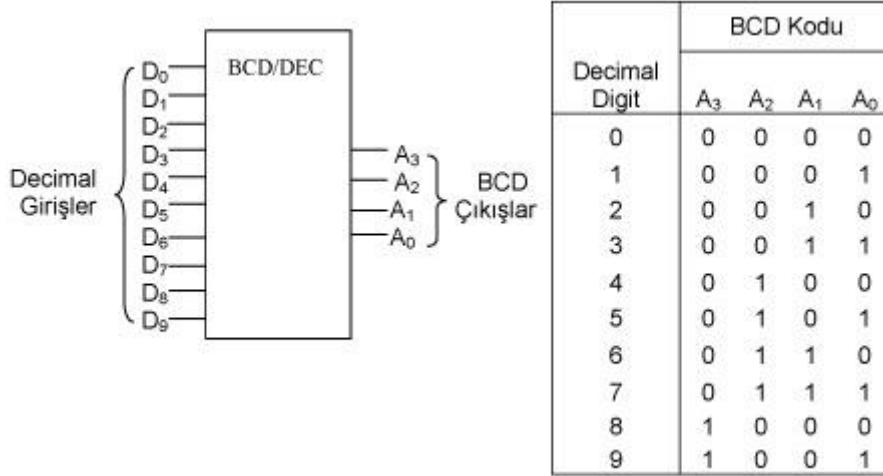


Kodlayıcılar (Encoder)

ENCODER (KODLAYICILAR) Kodlayıcı devre (encoder circuit) kod çözücü devrenin tersi işlemi yapar. **Bu devreler, decimal veya bilinen klasik şekillerdeki bilgileri sayısal devrelerin işlem yapabileceği şekle dönüştürürler.** Bir kodlayıcının (encoder) 2^n (yada daha az) giriş hattı ve n sayıda çıkış hattı üretir.

Decimal - BCD Encoder

Decimal – BCD encoder girişindeki decimal bilgiyi kodlayarak BCD kod karşılığını dört çıkışta gösterir. Aşağıda 10x4 encoder lojik sembolü ve doğruluk tablosu verilmiştir.



Çıkışları Lojik ifade doğruluk tablosundan yazmak istersek;

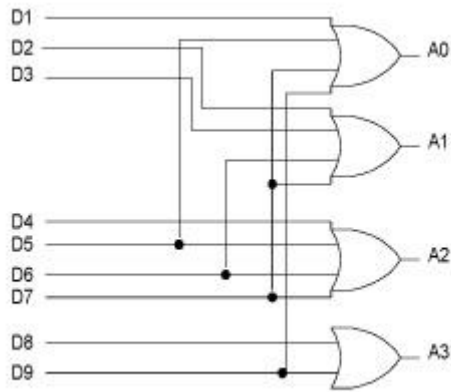
$$A_3 = D_8 + D_9$$

$$A_2 = D_4 + D_5 + D_6 + D_7$$

$$A_1 = D_2 + D_3 + D_6 + D_7$$

$$A_0 = D_1 + D_3 + D_5 + D_7 + D_9$$

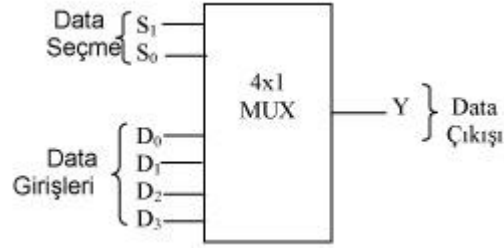
Aşağıda Decimal-BCD Encoder' in Lojik diyagramı verilmiştir.



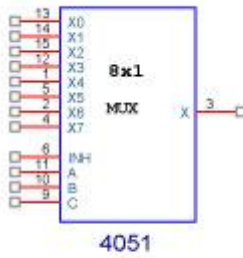
Çoğullayıcılar (Multiplexers)

Çoğullama çok sayıdaki bilginin, daha az sayıda kanal veya hat üzerinden iletilmesi anlamına gelir. Sayısal çoğullayıcı, birçok giriş hattının birisinden gelen ikilik bilgileri seçen ve tek bir çıkış hattına yönlendiren birleşik bir devredir. Belli bir girişin seçilmesi bir dizi seçme hattı ile kontrol edilir. Bir çoğullayıcı için 2^n sayıda giriş hattı varsa hangi girişin seçileceğini belirleyen n kadar seçme hattı vardır.

Data Inputs	Select	Data Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



Bazı durumlarda kod çözücülerde olduğu gibi Multiplexler içinde çalışmayı kontrol eden bir yetkilendirme (enable) girişi bulunabilir. Multiplex (çoğullayıcı) ancak yetkilendirildiğinde çoğullama işlemini gerçekleştirir. Yetkilendirme girişi, iki veya daha fazla sayıda Multiplex' in birleştirilerek daha çok sayıda girişli bir sayısal çoğullayıcının elde edilmesinde kullanılır. Aşağıda yetki girişli bir Multilex olan 4051 entegresinin doğruluk tablosu ve Lojik sembolü verilmiştir. Yetkilendirme girişi Inhibit (INH) olarak adlandırılmıştır.

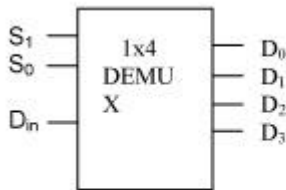


Input States				Output
INH	C	B	A	X
0	0	0	0	X_0
0	0	0	1	X_1
0	0	1	0	X_2
0	0	1	1	X_3
0	1	0	0	X_4
0	1	0	1	X_5
0	1	1	0	X_6
0	1	1	1	X_7
1	X	X	X	*

Bilgi Dağıtıcılar (Demultiplexers)

Demultiplex (Bilgi Dağıtıcı) tek bir hattın bilgi alan ve bu bilgiyi olası 2^n sayıda çıkış hattından birisi üzerinden ileten bir devredir. Belli bir çıkış hattının seçimi n kadar çıkış hattının durumları tarafından kontrol edilir.

Aşağıda iki seçme hattı ve dört çıkış hattı olan bir DEMUX' un doğruluk tablosu ve lojik sembolü verilmektedir.

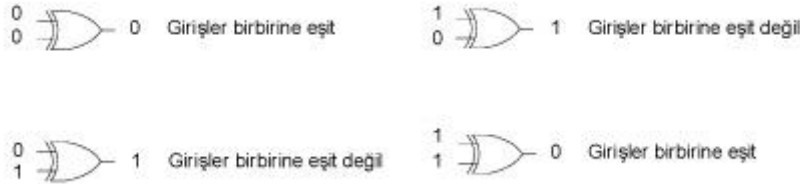


S_1	S_0	D_0	D_1	D_2	D_3
0	0	D_{in}	0	0	0
0	1	0	D_{in}	0	0
1	0	0	0	D_{in}	0
1	1	0	0	0	D_{in}

Karşılaştırıcılar (Comparators)

Karşılaştırma işlemi girişindeki sayısal bilgilerden birinin diğerine göre büyük,küçük veya eşit olma durumlarının belirlenmesidir.

En temel karşılaştırıcı devreleri Özel-Veya (Xor) kapılarıdır. Bir Özel-Veya kapısının girişleri farklı iken çıkış Lojik-1,girişleri aynı iken çıkış Lojik-0'dır. Şekilde veya kapısı ile temel karşılaştırma işlemini göstermektedir.



Özel-Veya kapısı ile girişlerindeki iki bitlik bilginin eşit olup olmadığı görülür. Ancak bir karşılaştırıcının eşitlik durumu ile birlikte bilginin küçük veya büyük olması durumlarını göstermesi beklenir.

Örnek:

Bir bitlik A ve B verilerini karşılaştıran bu karşılaştırma sonunda $A > B$, $A = B$, $A < B$ durumlarını gösteren devreyi tasarlayınız.

Devreye ait doğruluk tablosu aşağıdaki gibi olacaktır. Doğruluk tablosu yardımı ile çıkışlar yazılırsa;

Girişler		Çıkışlar		
A	B	$A > B$	$A = B$	$A < B$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

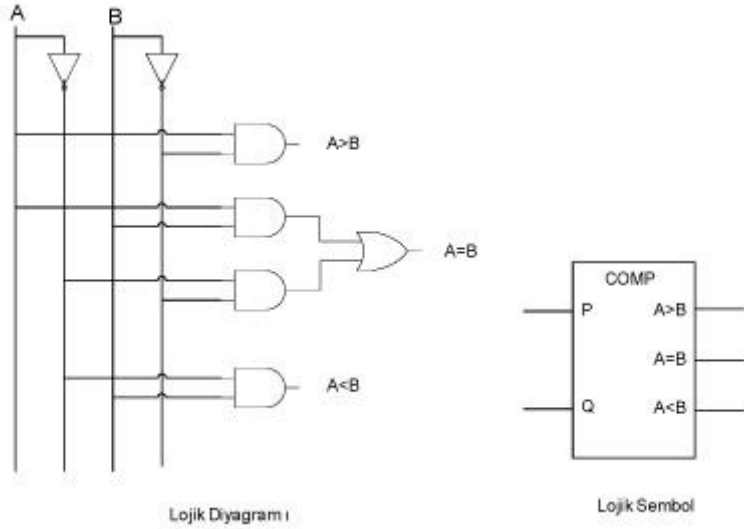
$$(A > B) = A \cdot \bar{B}$$

$$(A = B) = \bar{A} \cdot \bar{B} + A \cdot B = \bar{A} \oplus \bar{B}$$

$$(A < B) = \bar{A} \cdot B$$

olacaktır. Böyle bir devre ile bir bitlik iki verinin $A > B$, $A = B$, $A < B$ durumları belirlenecektir.

Aşağıda bir-bitlik büyüklük karşılaştırıcının Lojik diyagramı ve sembolü verilmiştir.



Aritmetik Üniteler

Toplama, çıkarma, çarpma ve bölme gibi aritmetik işlemleri yapan sayısal devrelere aritmetik devreler adı verilir. Sayısal sistemlerde temel aritmetik işlemler toplama ve çıkarma işlemidir. Çarpma işlemi tekrarlanan toplama, bölme işlemi ise tekrarlanan çıkarma işlemi ile tanımlanır.

Toplayıcı Devreler (Adders)

Sayısal devreler için toplama işlemini gerçekleştiren devrelere toplayıcılar (adders) adı verilir. Aşağıda Binary (ikilik) sayıların toplamına ilişkin temel kurallar verilmiştir.

				Elde(Carry)	Sonuç(Sum)
0	+	0	=	0	0
0	+	1	=	0	1
1	+	0	=	0	1
1	+	1	=	1	0

Not: Toplama işlemi sonunda oluşan eldenin işlem sonucunun en yüksek değerlikli basamağı olduğu unutulmamalıdır.

Yarım Toplayıcı (Half Adder)

Bir bitlik iki veriyi toplayan devrelere yarım toplayıcı (half adder) adı verilir. Bir yarım toplayıcının birer bitlik iki veri girişi için iki giriş, toplam ve oluşan eldenin gösterimi için iki tane çıkışı vardır. Aşağıda bir yarım toplayıcının tasarımı anlatılmıştır;

Bir bitlik iki veri P Ve Q ile adlandırırsak tasarlanacak devrenin iki binary sayının toplanması işlemini gerçekleştirmesi istenir. Toplama işleminin gösterimi için sonuç (Sum -S) ve elde (Carry -C) olmak üzere iki tane çıkış olması gerekir.

Doğruluk tablosu yardımı ile çıkışları yazmak istersek;

P	Q	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = \bar{P} \cdot Q + P \cdot \bar{Q} = P \oplus Q$$

$$C = P \cdot Q$$

ifadeleri elde edilir.

Not : Çıkışlara ait Lojik ifadeyi elde ederken Her bir çıkışa ait olan minimum terimin yazıldığı görülmelidir.

Aşağıda bir yarım toplayıcının (Half Adder) Lojik diyagramı ve sembolü verilmiştir.



Tam Toplayıcı (Full Adder)

İkinci temel tür toplayıcı derelere tam toplayıcı (full adder) adı verilir. Üç bitlik verilerin toplanması işlemini gerçekleştiren devrelerdir. Devrenin toplama işlemi için üç giriş, sonucun gösterimi için iki tane çıkışı vardır. Girişlerden ikisi toplama işlemi yapılacağı iki veriyi gösterirken diğer giriş düşük değerlikli basamaktan oluşan elde girişi içindir. Aşağıda bir tam toplayıcının doğruluk tablosu verilmiştir;

Doğruluk tablosunda ;

Cin – Bir önceki işlemde oluşan elde

Cout – Toplama işlemi sonrasında oluşan eldeyi göstermektedir.

Doğruluk tablosundan çıkışlara ait Lojik ifadeler ise ;

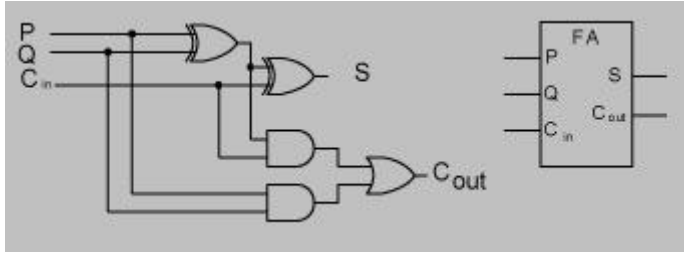
P	Q	C _{in}	C _{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = C_{in} \oplus (P \oplus Q)$$

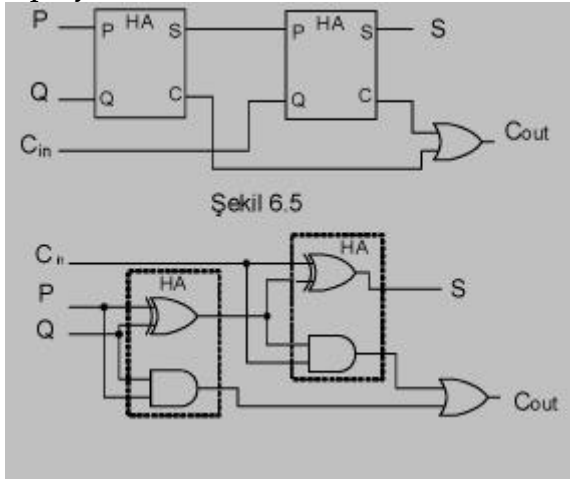
$$C_{out} = P \cdot Q + C_{in} \cdot (P \oplus Q)$$

ifadeleri elde edilir.

Aşağıda bir tam toplayıcının lojik diyagramı ve sembolü verilmiştir;



Aşağıda iki yarım toplayıcı ve harici bir VEYA kapısı kullanılarak elde edilmiş tam toplayıcı devresi verilmiştir.



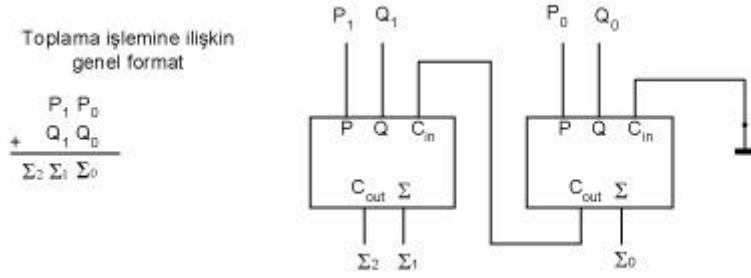
Paralel Toplayıcılar (Parallel Adders)

Bir n-bitlik paralel toplayıcı n tane tam toplayıcının birbirine paralel bağlanması ile elde edilebilir. Her bir tam toplayıcının elde çıkışı (Cout) daha yüksek değerli toplayıcının elde girişine bağlanır. Böylece düşük değerlikli basamakların toplamından oluşan elde (Cout) bir sonraki toplamı yapılacak basamaklara etki edebilecektir.

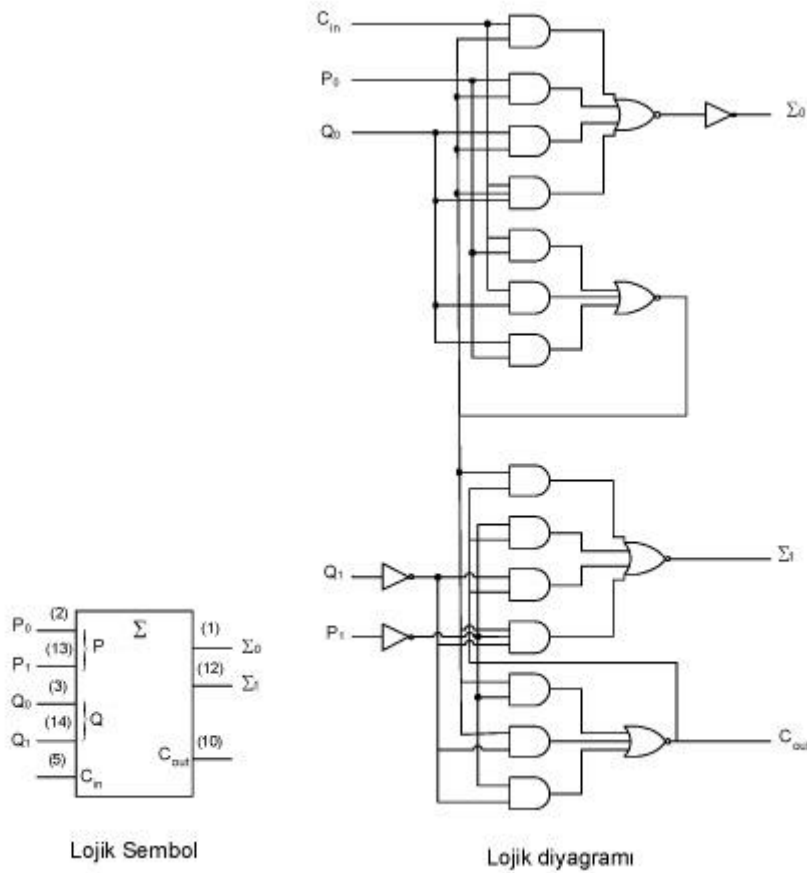
İki Bit Paralel Toplayıcı

Böyle bir devre ile iki bitlik verilerin toplama işlemi gerçekleştirilir. İki bitlik iki verinin toplanmasını sağlamak için iki tam toplayıcıya ihtiyaç vardır. Toplam işlemi en düşük değerlikli bitlerin toplamı ile başlayacaktır. Bu toplam işleminden oluşan elde (0 veya 1) bir sonraki toplama işlemine eklenmek zorundadır. İki bitlik P ile Q verilerinin toplanması ile işlemi açıklayalım;

Aşağıda iki tam toplayıcının paralel bağlanması ile elde edilmiş iki-bit paralel toplayıcı devresi ve sembolü verilmiştir. En yüksek değerlikli bitlerin toplamından oluşan elde toplama sonucunun en yüksek basamağıdır.

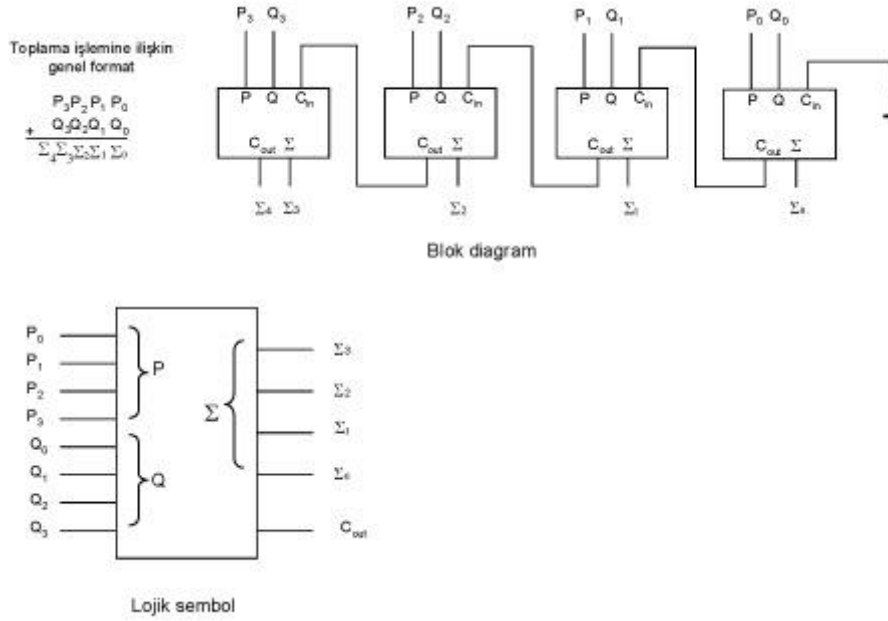


Not: En düşük değerlikli basamakların toplamına hiçbir zaman bir elde girişi olmadığından birinci tam toplayıcının Cin girişi toprağa (Lojik-0) bağlanmıştır.



Dört Bit Paralel Toplayıcı

Aşağıda dört-bit paralel toplayıcının blok diyagramı ve sembolü verilmiştir. Toplama işlemi için önce en düşük değerlikli bitler en sağdaki tam toplayıcı girişlerine uygulanır. Bütün tam toplayıcıların elde çıkışları (Cout) bir sonraki toplama işleminin yapılacağı tam toplayıcının elde girişlerine (Cin) bağlanmıştır.

**Örnek:**

İki tane 7482 İki-bit paralel toplayıcı kullanarak Dört-bit paralel toplayıcı elde ediniz.

Çözüm:

İki 7482 kullanarak dört-bit paralel toplayıcının elde edilmesi şekil 5.11'de gösterilmiştir. Düşük değerlikli iki bit birinci paralel toplayıcı girişlerinde toplanır. Yüksek değerlikli iki bitin toplamı ise ikinci paralel toplayıcıda gerçekleştirilir. Bu toplamdan oluşan elde toplama işleminin en yüksek basamağı olur.

