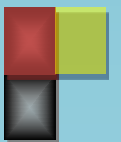
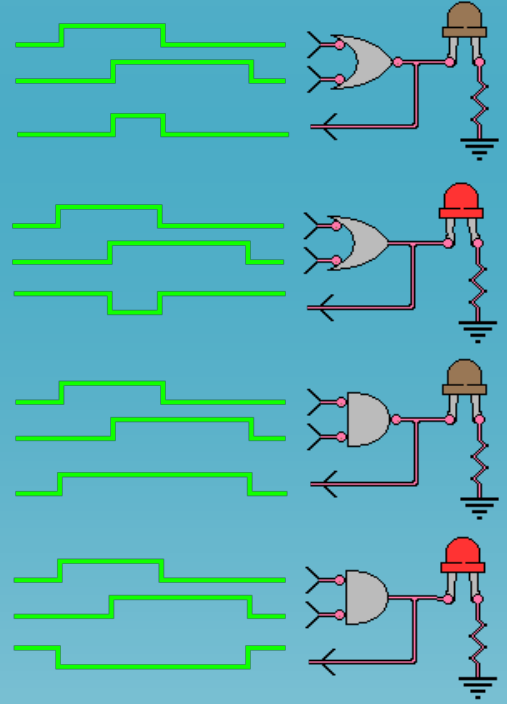
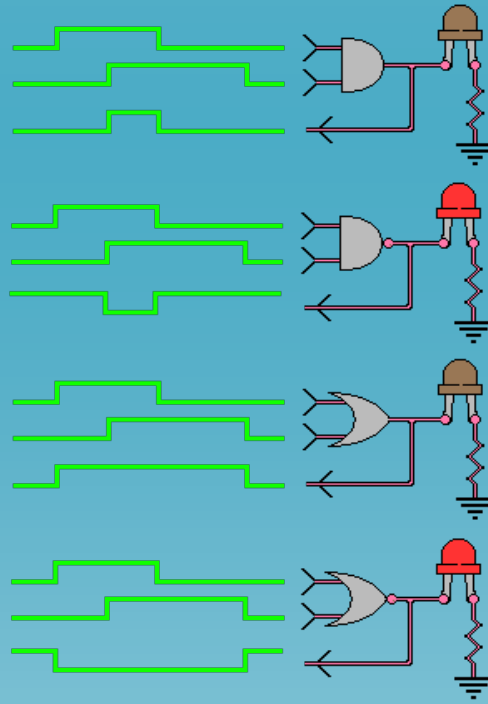
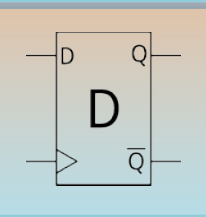
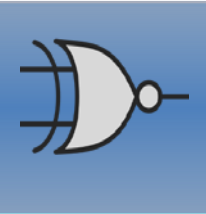
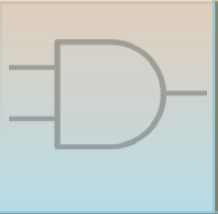


2015

SAYISAL ELEKTRONİK

Yrd. Doç. Dr. Mustafa Engin

Yrd. Doç. Dr. Dilşad Engin



EGE ÜNİVERSİTESİ
EGE MESLEK YÜKSEKOKULU

SAYISAL ELEKTRONİK

(DERS NOTU)



HAZIRLAYANLAR

Yrd.Doç.Dr. MUSTAFA ENGİN Yrd.Doç.Dr. DİLŞAD ENGİN

İZMİR 2015

İÇİNDEKİLER

SAYISAL VE ANALOG ÇOKLUKLAR	6
GİRİŞ	6
İKİLİK SAYILAR VE MANTIK DÜZEYLERİ	8
Dalga Biçimleri	10
SAYISAL DALGALAR	10
ZAMANLAMA DİYAGRAMI	11
Veri İletimi.....	11
SAYI SİSTEMLERİ VE SAYISAL KODLAR	13
ONLUK SAYILAR.....	13
İKİLİK SAYILAR	14
İkilik - Onluk Dönüştürme	15
Onluk - İkilik Dönüştürme	16
Ondalık Sayılar	18
İKİLİK ARİTMETİK	20
İkilik Toplama	20
İkilik Çıkarma.....	21
İkilik Çarpma.....	21
İkilik Bölme	22
İşaretli Sayıların Gösterimi	23
2-tümleyen sistemi	24
İşaretli Sayılarla aritmetik İşlemler	25
ONALTILIK SAYILAR	31
İkilik- Onaltılık Dönüşüm.....	32
Onaltılık-İkilik Dönüşüm.....	32
Onaltılık-Onluk Dönüşüm	33
Onaltılık Toplama	34
2-tümleyen yöntemiyle onaltılık çıkarma	35
İKİLİK KODLU ONLUK SAYILAR (BCD).....	36
İKO Toplama	37
SAYISAL KODLAR.....	37
Gray Kodu	38
Gray kodu arttırmalı enkoder.....	40
3-Artı Kod	40
TEMEL MANTIK İŞLEMLERİ	41

GİRİŞ	41
DEĞİL (NOT) İŞLEMİ	41
VE (AND) İŞLEMİ	42
VEYA (OR) İŞLEMİ	46
VED (NAND) GEÇİDİ	49
VED Geçidi uygulamaları	50
VEYAD (NOR) GEÇİDİ	52
ÖZEL VEYA GEÇİDİ (EXCLUSIVE-OR)	54
ÖZEL VEYA DEĞİL GEÇİDİ (EXNOR GATE)	55
TÜMDEVRE LOJİK AİLELERİ	58
PROBLEMLER	59
BOOLEAN KANUNLARI VE DEMORGAN TEOREMİ	61
GİRİŞ	61
BOOLEAN TOPLAMA	61
BOOLEAN ÇARPMA	61
BOOLEAN ARİTMETİĞİNİN KANUNLARI VE KURALLARI	62
Boolean Aritmetiğinin Kanunları	62
Boolean Kuralları	63
DE MORGAN TEOREMİ	67
ÇARPIMLARIN TOPLAMI	68
Toplamların Çarpımı (TÇ)	69
BİRLEŞİK MANTIK DEVRELERİNİN TASARIMI	70
3-Değişkenli Karnaugh Haritasının Kullanımı	70
4-Değişkenli Karnaugh haritası	71
TASARIM ÖRNEĞİ	77
Sorular	79
TOPLAYICI VE KARŞILAŞTIRICI	81
TOPLAYICILAR	81
PARALEL TOPLAYICILAR	82
KARŞILAŞTIRICILAR	83
MSI Karşılaştırmacılar	86
KODÇÖZÜCÜLER VE KODLAYICILAR	90
GİRİŞ	90
KOD ÇÖZÜCÜLER (DECODERS)	90
Temel İkili Kodçözücü	90

74156 ve 74138 3-e-8 Kodçözümler.....	91
Dört Bitlik İkili Kodçözümler	93
74154 4-E-16 Kodçözümler.....	95
İKO/Onluk Kodçözümler	95
KODLAYICILAR (ENKODERS).....	97
Onludan İKO'ya kodlayıcı	97
MULTİPLEXER DEMULTİPLEXER.....	100
GİRİŞ	100
MULTİPLEXER (VERİ SEÇİCİLER).....	100
Veri Seçici / Multiplexer Uygulamaları	103
DEMULTİPLEXER	105
74154 'ün Demultiplexer olarak kullanılması	106
74156 ve 74138 'in Demultiplexer olarak kullanılması.....	107
EŞLİK BİTİ ÜRETECİ	107
MULTİPLEXER UYGULAMALARI	108
TÜMDEVRE VERİ YAPRAKLARI	113
7400 DÖRTLÜ İKİ GİRİŞLİ VED GEÇİDİ.....	114
4011 DÖRTLÜ İKİ GİRİŞLİ CMOS VED GEÇİDİ	115
74LS02 DÖRTLÜ İKİ GİRİŞLİ VEYAD GEÇİDİ.....	116
4001 DÖRTLÜ İKİ GİRİŞLİ CMOS VEYAD GEÇİDİ	117
74LS04 ALTILI DEĞİL GEÇİDİ	118
74LS08 DÖRTLÜ İKİ GİRİŞLİ VE GEÇİDİ	119
74LS20 İKİLİ DÖRT GİRİŞLİ VED GEÇİDİ	120
74LS32 DÖRTLÜ İKİ GİRİŞLİ VEYA GEÇİDİ	121
74LS86 DÖRTLÜ EXOR GEÇİDİ.....	121
74LS283 (74LS83) 4 BİT TAM TOPLAYICI	122
74HC85 4 BİT BÜYÜKLÜK KARŞILAŞTIRICI.....	125
74LS138 3-8 KODÇÖZÜCÜ/VERİ DAĞITICI.....	127
74HC154 4-16 KODÇÖZÜCÜ/VERİ DAĞITICI	129
74HCT147 10 GİRİŞLİ 4 BİT ÇIKIŞLI YÜKSEK GİRİŞ ÖNCELİKLİ KODLAYICI.....	131
74LS151 8 GİRİŞLİ VERİ SEÇİCİ	133
74LS74A DUAL D TİPİ FLİP-FLOP	134
74HCT75 DÖRTLÜ TUTUCU.....	136
LM555 Timer	138

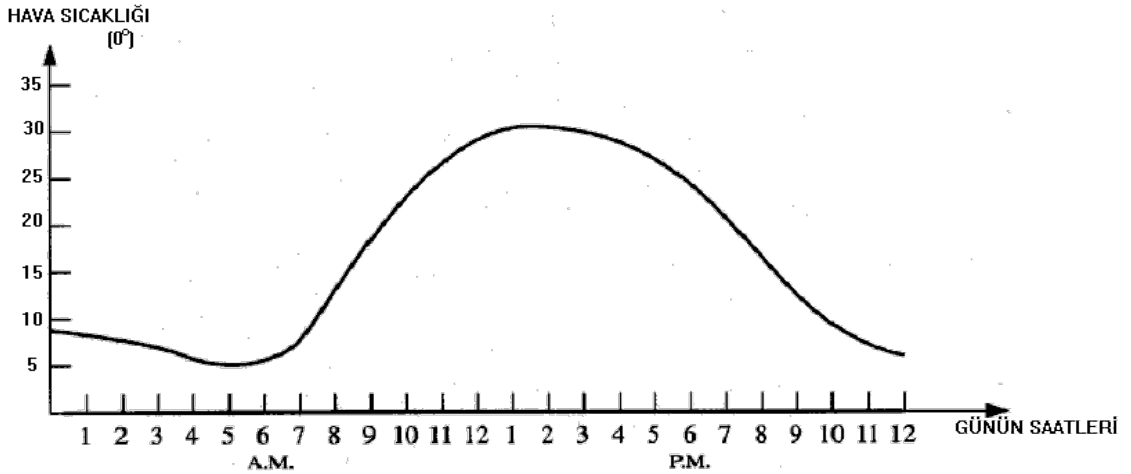


SAYISAL VE ANALOG ÇOKLUKLAR

GİRİŞ

Elektronik devreleri ilgilendikleri çokluklara göre dijital (sayısal) ve analog (örneksel) olmak üzere iki ana gruba ayırabiliriz. Analog bir çoklukta değer değişimi sürekli ve kesintisiz iken, sayısal bir çokluğun değişimi kesiklidir ve ayrık (*discrete*) değerlerden oluşur.

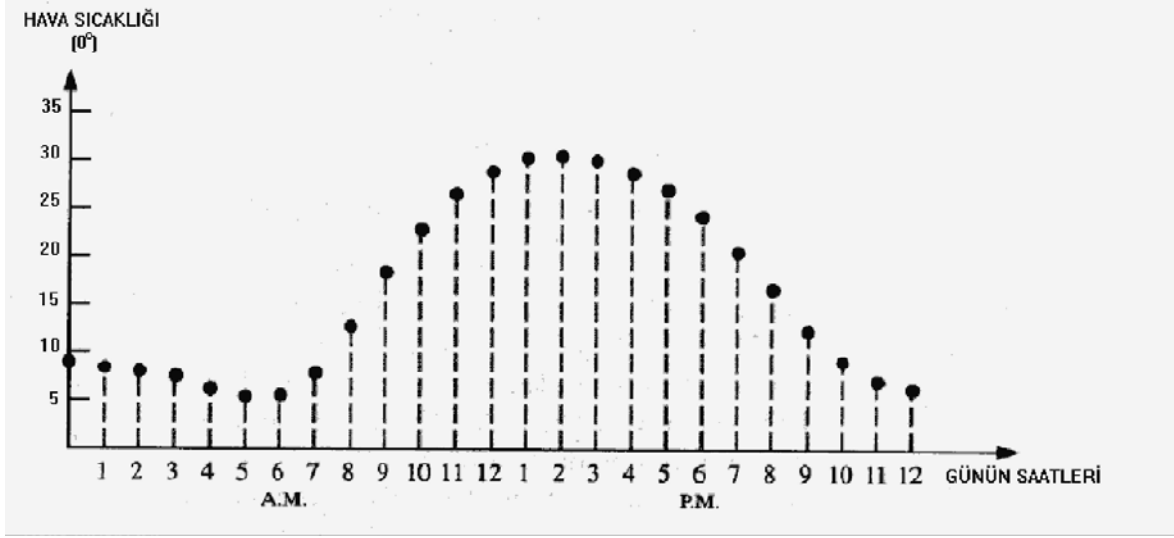
Bu tanıımı açmak için bir yaz günündeki ısı değişimini ele alalım. Havanın sıcaklığı birdenbire örneğin 27°C'den 28°C'ye çıkmaz, bu iki derece arasında sonsuz sayıdaki bütün değerleri alarak değişir. Bu değişimin grafiğini çizdiğimizde şekil-1.1'deki gibi kesintisiz ve sürekli bir eğri elde ederiz. Analog büyüklüklere diğer örnekler, zaman, basınç, uzaklık ve sestir.



Şekil 1-1 Analog işaret örneği

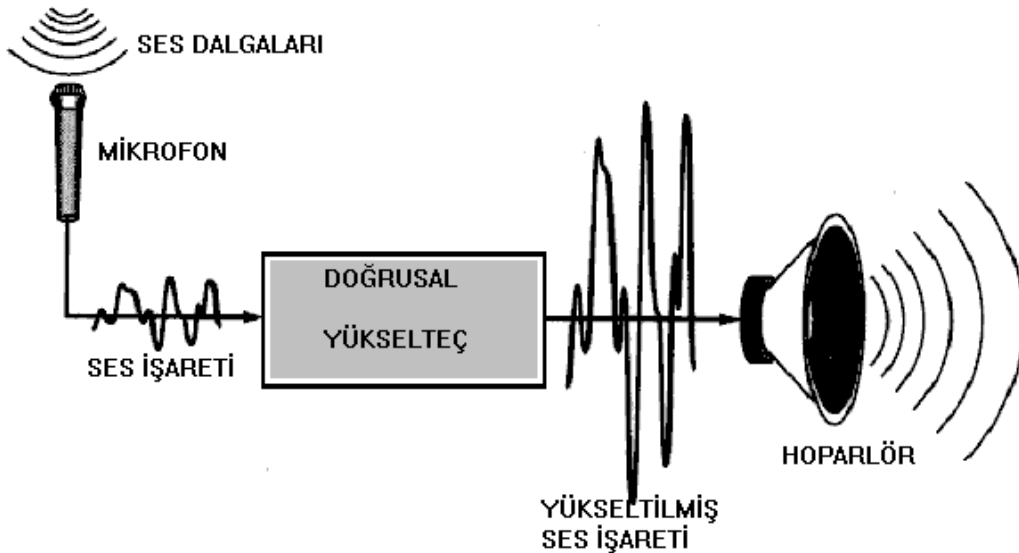
Diğer bir yöntem olarak ısıyı sürekli gözlemek yerine saat başlarında ölçerek şekil-1.2'deki gibi örnekleyebiliriz. Bu grafik henüz bir dijital gösterim değildir ama dönüşüm işleminin büyük kısmı tamamlanmıştır. Her örnek değer dijital bir kodla belirlendiğinde analog-dijital dönüşüm tamamlanmış olur. Elektronikte dijitalin analoga göre belirgin üstünlükleri vardır. En başta dijital bilgi analog bilgidan daha etkin ve güvenli olarak işlenebilir ve iletilebilir. Ayrıca bilginin saklanması gerektiğinde dijital bilginin büyük bir üstünlüğü vardır. Örneğin müzik

dijitalleştirildiğinde, çok daha yoğun biçimde depolanıp büyük bir hassasiyetle yeniden üretilir ve analog biçime dönüştürülebilir.

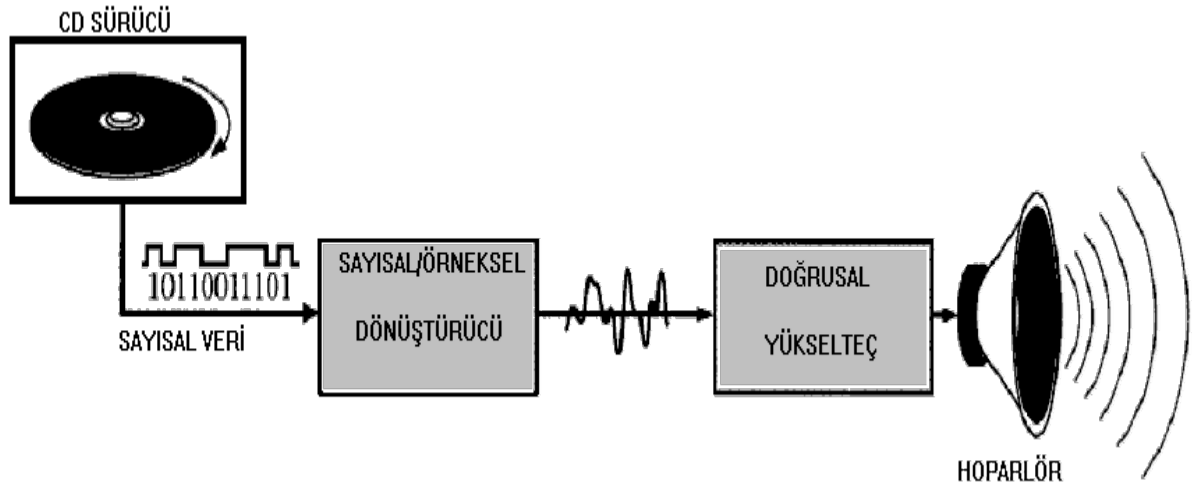


Şekil 1-2 Sayısal işaret örneği.

Analog bir elektronik sisteme örnek olarak bir anons devresini verebiliriz. Analog doğal ses dalgaları mikrofona yardımıyla ses işareti denilen küçük analog gerilimlere dönüştürülür. Bu gerilim sesin genliği ve frekansı ile değişir ve yükselteç ile güçlendirildiğinde de bu özelliklerini yitirmez. Yükselteç yardımıyla yeterince güçlendirilen ses işareti hoparlöre uygulanarak yeniden ses dalgalarına dönüşmesi sağlanmış olur. Dijital ve analog işaretlerin birlikte kullanıldığı bir sisteme en tanınmış örnek CD çalardır. Şekil-1.3'teki basitleştirilmiş diyagram temel ilkeyi göstermektedir.



Şekil 1-3 Analog işaretin işlenmesi.

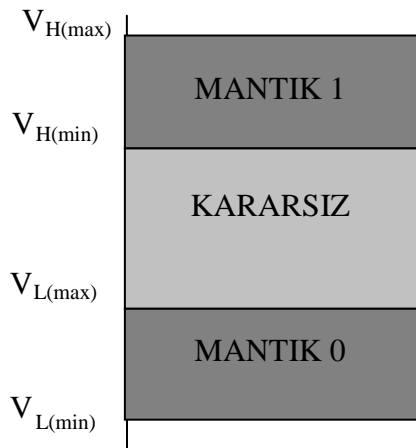


Şekil 1-4 Sayısal işaretin işlenmesi.

İKİLİK SAYILAR VE MANTIK DÜZEYLERİ

Sayısal elektronikte olası yalnız iki durum vardır: YÜKSEK yada DÜŞÜK. Bu iki durum akım şiddetleri, yanık yada sönmük lambalar, açık yada kapalı anahtarlar olarak yada en yaygın biçimiyle iki değişik gerilim değeri ile gösterilirler. Dijital sistemlerde kod dediğimiz ve bu iki durumun kombinasyonlarından oluşan diziler, sayıları, simgeleri, alfabetik karakterleri ve diğer bilgi türlerini göstermekte kullanılırlar. Bu iki durumlu sayı sistemine İKİLİK (BINARY) denir ve bu sistem 0 ve 1'den başka sayı içermez.

İkili sistemde kullanılan iki sayı yani 1 ve 0, *BIT* olarak adlandırılırlar. 1 ve 0 ları göstermek için kullanılan gerilim aralıklarına mantık düzeyi denir. $V_{H(max)}$ ile $V_{H(min)}$ arasında kalan gerilim değerleri mantık 1, $V_{L(max)}$ ile $V_{L(min)}$ arasında kalan gerilim değerleri de mantık 0 bitini ifade eder. $V_{H(min)}$ ile $V_{L(max)}$ sınırları arasında kalan gerilim değerleri belirsizlik ifadesidirler ve iki düzey arasında gerekli tampon aralığını sağlarlar.

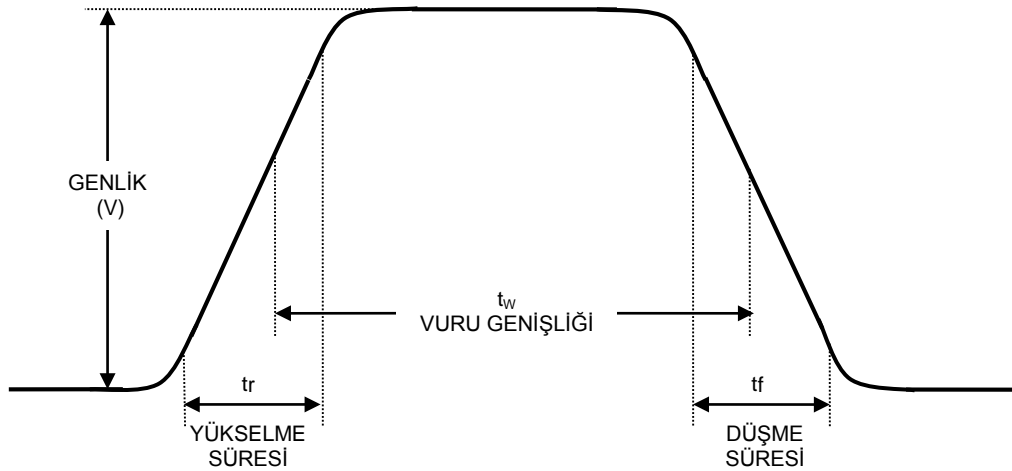


Şekil 1-5 Mantık değerlerin gerilim seviyeleri.

Sayısal dalga biçimleri YÜKSEK ve DÜŞÜK arasında gidip gelen gerilimlerden oluşurlar. Sık karşılaşılan temel kavramları açıklayalım;

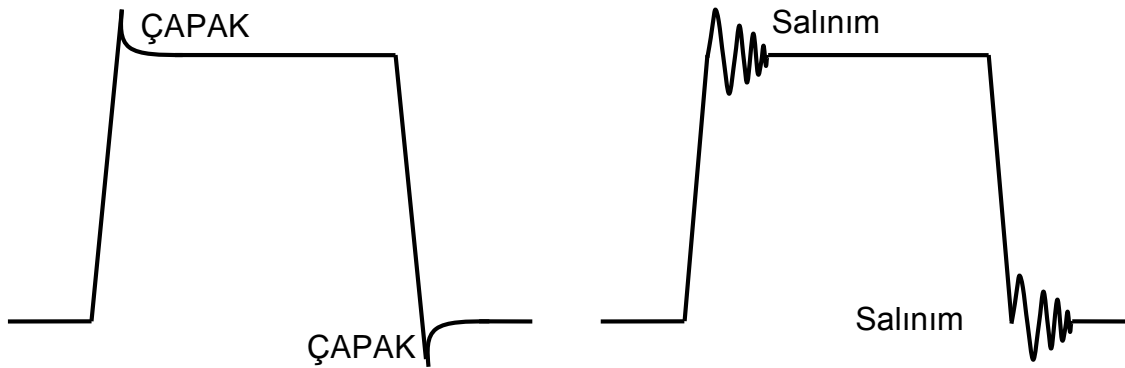
VURU (PULSE)

İki kenarı bulunan ve durumlar arası gidip gelen sayısal işaretin her bir adımına vuru denir. Vuruyu oluşturan kenarlardan birincisine **yükselen (önder) kenar** (*rising or leading edge*), ikincisine ise **düşen (izleyen) kenar** (*falling or trailing edge*) denir. Bir vurunun ideal olması için durum geçişlerinin sıfır sürede gerçekleşmesi gerekir ve bu duruma uygulamada hiçbir zaman ulaşılamaz. Vurunun DÜŞÜK'ten YÜKSEK'e geçmesi için gereken zamana **yükselme süresi** (*rising time*), tersi için gereken süreye de **düşüş süresi** (*fall time*) denir. Bu süreler vurunun tepe genliğinin 10% ve 90% değerleri arasında ölçülür. **Vuru genişliği** genliğin 50% değerleri arasındaki süre ile ölçülür.



Şekil 1-6 Vurunun özellikleri.

Çapak ve çınlama (overshoot, undershoot and ringing), istenmeyen ama genellikle oluşan bu bozulmalardan ilki olan çapak, devrenin yada ölçme aletinin sığasal etkisi nedeniyle oluşur ve normal değerleri kısa süreli aşan gerilim sıçramalarına neden olur. Vurunun yükselen ve düşen kenarlarında oluşan çınlama aslında küçük bir salınımdır ve devredeki kapasitans ile endüktanstan kaynaklanır. Çınlama, çapak bileşenlerini de içerir ve kısa sürede söner.



Şekil 1-7 Vuruda istem dışı oluşan salınım ve çapaklar.

DALGA BİÇİMLERİ

Sayısal sistemlerde karşılaşılan çoğu dalga biçimi vuru dizilerinden oluşmuştur ve periyodikliklerine göre adlandırılırlar. Eğer dalga biçimi belirli bir aralıkta kendini yineliyorsa periyodik vuru olarak adlandırılır. frekans, *hertz* olarak yinelenme hızıdır. Şekil-1.8'de periyodik olan bir sayısal işaret, Şekil-1.9'da ise periyodik olmayan bir sayısal işaret gösterilmiştir. Periyodik dalgada tüm vuruların periyotları eşittir, periyodik olmayan işaretle ise her vurunun periyodu farklıdır. Sayısal sistemlerde her iki işaret türünde kullanılır ve birbirine göre üstünlüğü yoktur. Bir dalga biçiminin sıklığı periyodu ile ters orantılıdır. Sıklık ile periyot arasındaki bağıntıyı şu eşitliklerle gösterebiliriz:

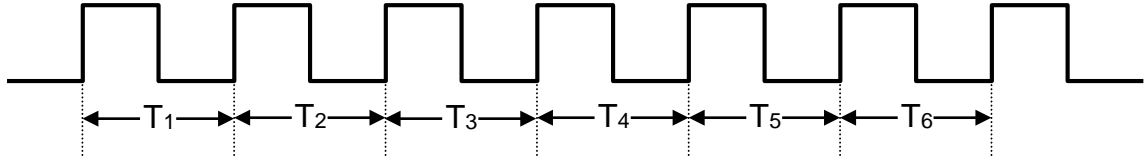
$$f = 1 / T = T^{-1}$$

$$T = 1 / f = f^{-1}$$

Periyodik bir dalga biçiminin önemli bir diğer özelliği de **görev süresidir (duty cycle)**. Görev süresi, vuru (t_w) genişliğinin periyoda olan oranının yüzdelik ifadesidir ve şu şekilde gösterilir:

$$\text{Görev süresi \% (D)} = \frac{t_w}{T} \times 100$$

Şekil-1.8'de periyodik dalganın görev süresi %50'dir. Periyodik olmayan dalganı ise görev süresi her vuru için farklıdır. Bu tür işaretlerin görev süresi hesaplanamaz. İstenildiğinde belirli bir kısmının görev süresi hesaplanabilir.



$$\text{Periyot} = T_1 = T_2 = T_3 = T_4 = T_5 = T_6$$

$$\text{Frekans} = \frac{1}{T}$$

Şekil-1.8 Periyodik dalga şekli.



Şekil-1.9 Periyodik olmayan dalga şekli.

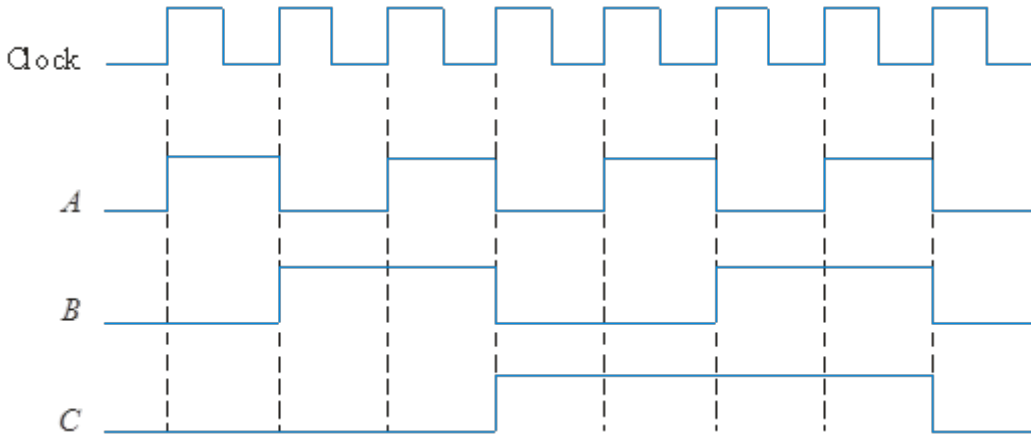
SAYISAL DALGALAR

Sayısal sistemlerde işlenen bilgiler bit dizilerini temsil eden sayısal işaret biçimleri olarak üretilir ve iletilir. İşaretin YÜKSEK olması ikilik 1 verisini, DÜŞÜK olması da ikilik 0 bilgisini gösterir. Sıralı bitlerin her birisi, *bit süresi* denilen belirli bir zaman aralığını kaplar.

Saat (clock): Çoğu sayısal sistemde bütün dalga biçimleri saat denilen temel bir işarette eş zamanlanırlar. Saat; vuruları arasındaki süre bir bit süresine eşit olan ve periyodik dalga biçimli bir işarettir.

ZAMANLAMA DİYAGRAMI

Zamanlama diyagramı, bütün dalga biçimlerinin zamana göre ilişkilerini ve birbirlerine göre nasıl değiştiklerini gösteren bir grafikdir ve çok sayıda sayısal işareti içerebilir. Bu diyagramlar yardımıyla bir bakışta bütün dalga biçimlerinin durumları (YÜKSEK yada DÜŞÜK) ve diğerleri ile ilişkileri görülebilir. Şekil-1.10'da da dört dalga biçimi içeren bir zamanlama diyagramının lojik analizördeki görüntüsü verilmiştir.



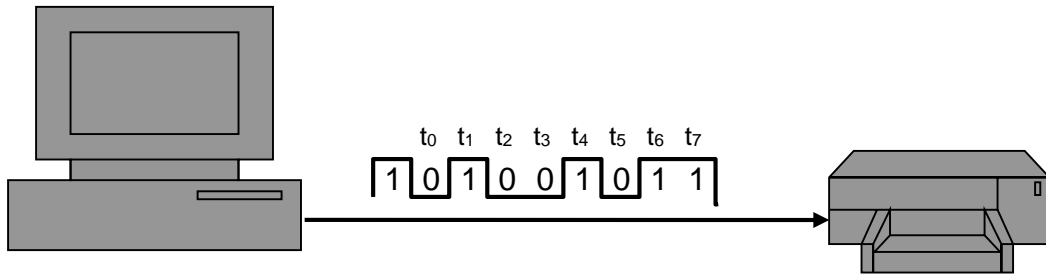
Şekil-1.10 Zamanlama diyagramı.

VERİ İLETİMİ

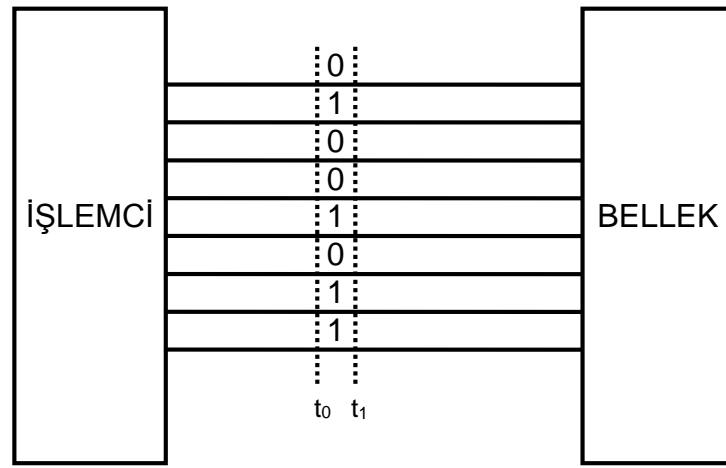
Bir tür bilgi taşıyan bit gruplarına **veri** (*data*) denir. Bir işlem gerçekleştirilebilmesi için, dijital dalga biçimlerinden oluşan ikilik verinin sistemler arasında iletilmesi gereklidir. Veri iletimi seri ve paralel olmak üzere iki türlü yapılmaktadır.

Seri iletimde bitler bir iletken hat üzerinden ardarda gönderilirler. Bu iletim türüne örnek olarak bilgisayardan yazıcı/basıcıya olan veri akışını verebiliriz.

Paralel iletimde ise küçük veri paketleri aynı anda ayrı iletkenler üzerinden gönderilirler. Her bit için bir hat gerekli olduğundan daha masraflıdır ama bir iş süresinde gönderilebilen bit sayısı paralel hat sayısı kadar fazla olduğundan hızı çok daha yüksektir. Bu tür iletime örnek bilgisayarın mikroişlemcisi ile bellek arasındaki veri akışını verebiliriz.



a. Seri veri iletimi



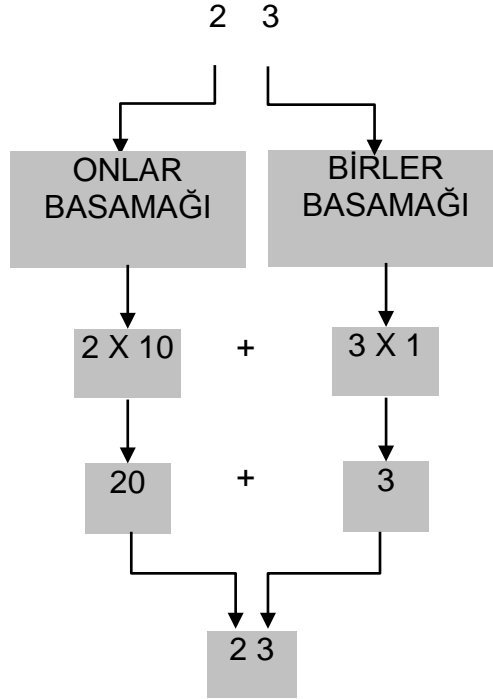
b. Paralel veri iletimi

Şekil 1-11 Seri ve paralel veri iletimi.

BÖLÜM 2

SAYI SİSTEMLERİ VE SAYISAL KODLAR

ONLUK SAYILAR



Onluk sayı sisteminde kullanılan her rakam (0 - 9) belli bir çokluğu gösterir. Buna karşın basamak değerleri değişik olduğundan gerekli basamaklara gerekli rakamları koyarak istediğimiz her çokluğu ifade edebiliriz. 9'a kadar olan çoklukları bir basamakta gösterebiliriz. Eğer dokuzdan yüksek bir değeri belirtmemiz gerekirse bir yada daha fazla basamak ekleyebiliriz. Kullanılan 10 değişik rakam olduğundan on-tabanlı sistem de denilir. Onluk 23 sayısını çarpanlarına ayırılım; 3 rakamının ağırlığı 1'dir , 2 rakamı bu basamakta 10 ağırlığındadır. Her rakam bulunduğu basamağın *ağırlığına* bağlı bir değer gösterir. Onluk sistemde basamakların ağırlığı en sağ basamakta $10^0=1$ den başlar ve sola doğru 10 un pozitif kuvvetlerini alarak artar.

..... 10^5 10^4 10^3 10^2 10^1 10^0

Kesirli sayılarda da basamakların ağırlığı sağa doğru 10'un negatif kuvvetleri ile azalır.

$$.....10^1 10^0 . 10^{-1} 10^{-2} 10^{-3}$$

İKİLİK SAYILAR

İkilik sayı sistemi ile de dilediğimiz çokluğu gösterebiliriz. Yalnızca iki rakam içerdiği için onluk sistemden daha basittir. İkilik sistemde yalnızca iki rakam bulunduğu için iki-tabanlı sayı sistemi olarak da adlandırılır.

Onluk sistemde sayarken sıfırdan başlar ve dokuza dek tek basamakla gideriz. Kullanabileceğimiz rakamlar bitince bir basamak artırır ve en küçük rakamı (1) bu basamağa koyarak saymayı sürdürürüz. Yeni basamaktaki rakamı artırarak bütün kombinasyonları bitirip 99 a gelince bir basamak daha artırıp devam ederiz.

ONLUK SAYI	İKİLİK SAYI			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Çizelge 2-1 4 bit ikilik sayıların onluk karşılıkları.

Yalnızca iki değişik rakam bulunması dışında ikilik sistemdeki sayma işlemi de aynı yapıdadır. Saymaya başlayalım: 0, 1. İki rakamı da kullandık. Şimdi basamak arttırmalıyız. 10, 11... bir basamak daha 100, 101, 110, 111. Şimdi dördüncü basamağa gerek duyuyoruz sonra beş, altı. Görüldüğü gibi aynı çokluğu belirtmek için ikilik sistemde onluk sistemden daha fazla basamak gerekmektedir.

İkilik sistemde belli bir sayıda basamakla gösterilebilecek en büyük onluk sayı şu şekilde hesaplanır:

En büyük onluk sayı = $2^n - 1$

Burada n kullanılan bit sayısıdır. Örnek olarak beş bitlik bir ikilik sayı ile gösterilebilecek en yüksek onluk değeri hesaplayalım:

$$2^5 - 1 = 32 - 1 = 31$$

Altı bitle gösterilebilecek en yüksek değer de;

$$2^6 - 1 = 64 - 1 = 63$$

olarak bulunur.

İKİLİK - ONLUK DÖNÜŞTÜRME

Bir ikilik sayının onluk eşdeğeri, her basamaktaki bitin, o basamağın ağırlığıyla çarpılıp, sonra bütün çarpımların toplanmasıyla bulunur. En sağdaki bit en az önemli bit (*least significant bit - LSB*), en soldaki bit ise en önemli bit (*most significant bit - MSB*) olarak adlandırılır. LSB'nin ağırlığı, $2^0 = 1$ dir. MSB'nin ağırlığı ise sayının boyuna bağlıdır.

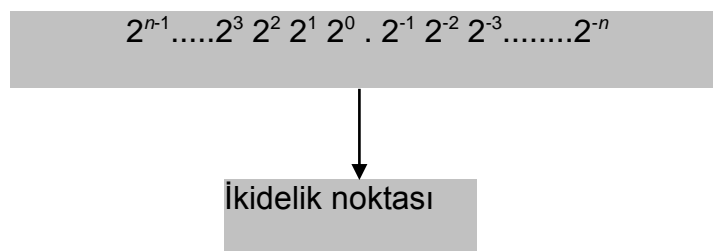
ÖRNEK 2.1:

10100101 ikilik sayının onluk karşılığını bulun.

Çözüm: Her bitin ağırlığı belirlenir, değeri bir olan bitler ağırlığı ile çarpılır ve çarpımlar toplanarak onluk karşılığı elde edilir.

Ağırlığı	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
İkilik sayı	1	0	1	0	0	1	0	1
=	1 x 128	+	1 x 32		+	1 x 4	+	1 x 1
=	128	+	32		+	4	+	1
=	165							

İkilik sistemde kesirli sayılar da gösterilebilir. Burada da aynı onluk sistemdeki gibi ondalık noktasının sağına doğru azalan negatif kuvvetler ile basamak ağırlığı düşer.



ÖRNEK 2.2:

101.00101 ikilik sayının onluk karşılığını bulun.

Çözüm: Her bitin ağırlığı belirlenir, değeri bir olan bitler ağırlığı ile çarpılır ve çarpımlar toplanarak onluk karşılığı elde edilir.

Ağırlığı	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
İkilik sayı	1	0	1	.	0	0	1	0	1
=	4×1	+	1×1	+			0.125×1	+	0.03125×1
=	4	+	1	+			0.125	+	0.03125
=	5.15625								

ONLUK - İKİLİK DÖNÜŞTÜRME**Ağırlıklar toplamı yöntemi:**

Verilen onluk sayının ikilik karşılığını bulmada kullanılan yöntemlerin ilki *ağırlıklar toplamı*dır. Bu yöntemde, verilen onluk sayının değerini verecek bit grubu belirlenir. İkilik sistemde basamakların değerleri ya sıfırdır yada basamak ağırlığına eşittir. Buradan yola çıkarak dönüştürülecek onluk sayının değerinden küçük en büyük ağırlığa sahip olan basamağa 1 yazılır. Geri kalan miktar için de aynı işlem yapılır. Onluk sayının tam değerine ulaşana dek işlem sürdürülerek dönüşüm tamamlanır.

Örneğin onluk 9 sayısı ikilik ağırlıklar toplamı olarak şöyle gösterilebili; Bulunan ağırlıkları taşıyan basamaklara 1 diğerlerine sıfır yazarak,

$$9=2^3+2^0,$$

burada bulunmayan 2^2 , 2^1 ağırlıklarının çarpanı sıfır, olanların çarpanı 1 olarak yazıldığında ikilik karşılığı elde edilir.

$$9=1001$$

Şimdi de 47 sayısını ikilik olarak yazalım;

İkilik sistemdeki basamak ağırlıkları 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048,... olarak sıralanır. Burada 47 sayısına sığabilen en küçük basamak 6. sıradaki 32 dir. Buna göre bulunacak ikilik eşdeğer altı basamaklı olacaktır. Geriye

$$47 - 32 = 15 \text{ kalır.}$$

Buna sığan en büyük basamak 4. sıradaki 8'dir. Geriye kalan;

$$15 - 8 = 7 \text{ dir}$$

Sırasıyla 3., 2. ve 1. basamaklar da 4, 2 ve 1 ağırlıklarıyla doldurulur. Ağırlıklarının

kullandığımız ikilik basamaklara 1 diğerlerine sıfır yazarsak 47'nin ikilik karşılığı

$$(101111)_2$$

olarak elde edilir.

Örnek 2. 3:

Aşağıda verilen onluk sayıları ikiliye dönüştürün.

a. $12 = 8 + 4 = 2^3 + 2^2$ **1100**

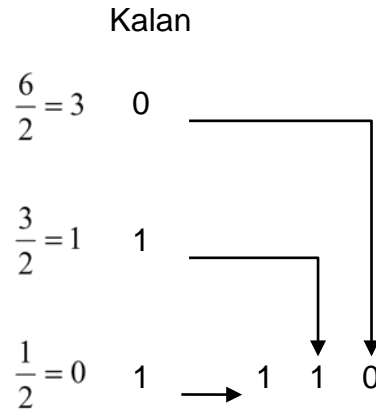
b. $25 = 16 + 8 + 1 = 2^4 + 2^3 + 2^0$ **11001**

c. $58 = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1$ **111010**

d. $82 = 64 + 16 + 2 = 2^6 + 2^4 + 2^1$ **1010010**

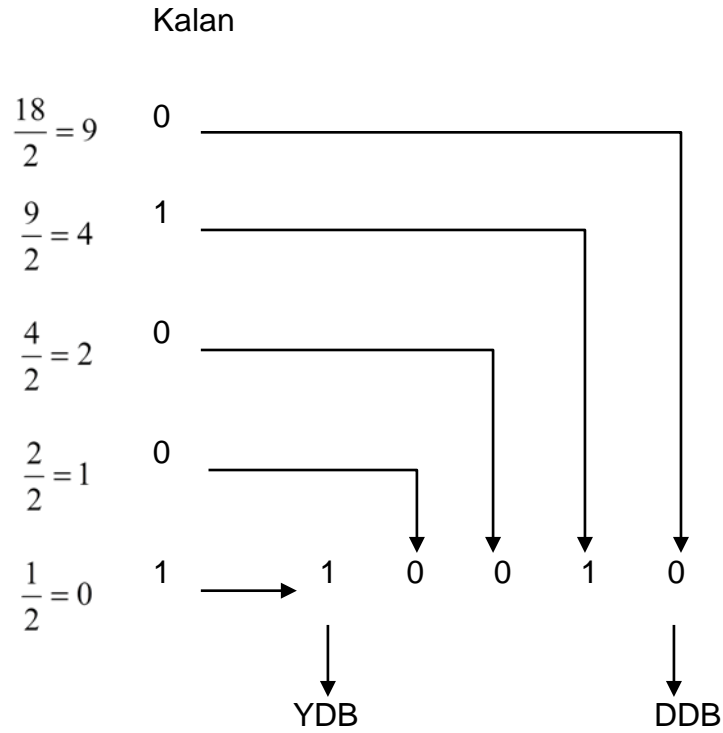
Sürekli 2' ye bölme yöntemi:

Onluk tamsayıları ikiliğe dönüştürmede kullanılan başka bir yöntem de tekrar tekrar ikiye bölmekten oluşan *sürekli 2' ye bölme yöntemi*dir. Bu yöntemde onluk sayı bölümün tamsayı kısmı 0 çıkana dek ikiye bölünür. Her bölmeden sonra **kalan** ikilik sayıyı oluşturur. Örnek olarak 6'yı bu yöntemle ikilik olarak yazalım.



Örnek 2.4:

18'i 2'ye bölme yöntemi ile ikiliğe dönüştürün.



ONDALIKLI SAYILAR

Ağırlıklar toplamı yöntemi

Yöntem kesirli onluk sayılara da uygulanabilir. Örnek olarak 0.625 sayısını ikilik olarak yazalım:

$$0.625 = 0.5 + 0.125 = 2^{-1} + 2^{-3}$$

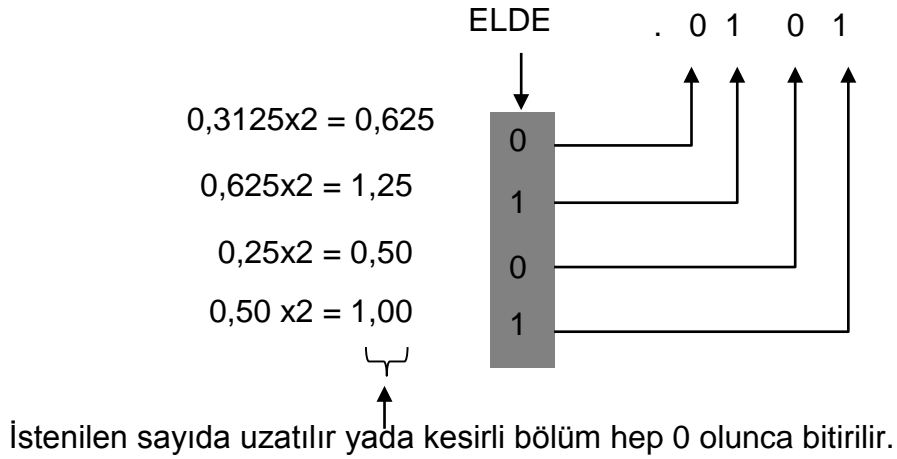
2^{-1} ve 2^{-3} basamaklarına 1 yazarak $0.625 = 0.101$ bulunur.

Sürekli 2 ile çarpma yöntemi

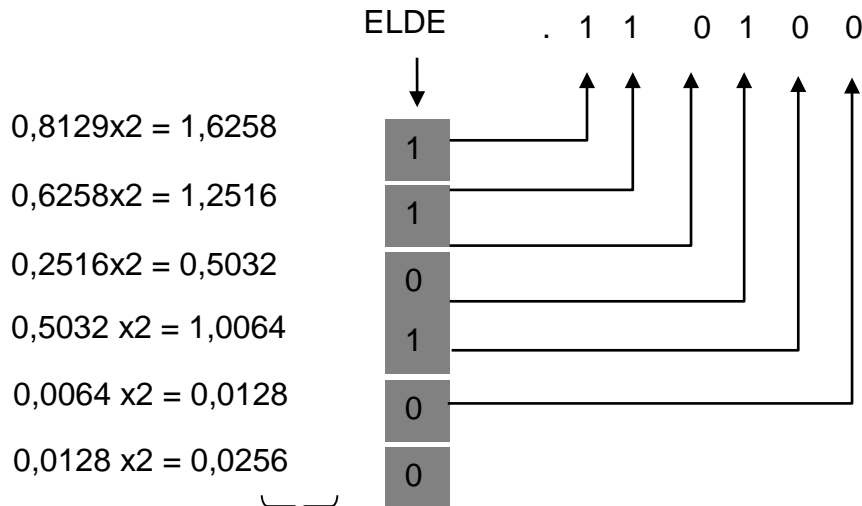
Bu yöntem kesirli sayıların dönüşümünde de küçük değişikliklerle kullanılır. İlk önce verilen kesirli sayı ikiyle çarpılarak sonucun ondalıklı bölümü yeniden ikiyle çarpılır. Bu işleme kesirli kısım sıfırlanana dek yada istenildiği kadar devam edilir. En sonunda sonuçların tamsayılarına bakılır. Taşınan basamakların yada eldelerin oluşturduğu ikilik bit dizisi aranan ikilik karşılığı oluşturur.

Örnek 2.5:

0,3125 sayısını ikilik tabana dönüştürün.

**Örnek 2.6:**

0,8129 sayısını ikilik tabana dönüştürün.



İstenilen sayıda uzatılır yada kesirli bölüm hep 0 olunca bitirilir. Bu örnekte devam edildiğinde arka arkaya 7 sıfırdan sonra en az anlamlı bit 1 olmaktadır. Bu kadar yüksek hassasiyet istenmediği durumda kesirli kısmın sıfıra en yakın olduğu aşamada, yani 4. basamaktan sonra işlem bırakılır.

Örnek 2.7:

45,8129 sayısını ikilik tabana dönüştürün.

Sayı tam ve kesirli olmak üzere iki kısımdan oluşmaktadır. Dönüşüm ayrı ayrı yapılır, işlem sonunda birleştirilir. Sürekli 2'ye bölme yöntemi ile tamsayı kısmı dönüştürülür ve yukarıdaki örnekte yapıldığı gibi, sürekli 2 ile çarpma yöntemi ile de kesirli kısmın dönüşümü yapılır.

$$45_{10} = 101101_2$$

ve

$$0,8129_{10} \approx 0,1101_2$$

$$45,8129 \approx 101101,1101_2$$

İKİLİK ARİTMETİK

Bütün bilgisayarlarda ve çoğu diğer dijital sistemlerde ikilik aritmetik kullanılır. Dijital sistemleri anlayabilmek için ikilik toplama, çıkarma, çarpma ve bölme işlemlerini bilmek gerekir.

İKİLİK TOPLAMA

İkilik sayı bitlerini toplamanın dört temel kuralı şunlardır:

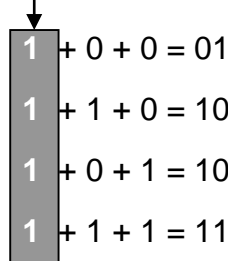
$$0 + 0 = 0 \quad \text{toplam 0, elde 0}$$

$$0 + 1 = 1 \quad \text{toplam 1, elde 0}$$

$$1 + 0 = 1 \quad \text{toplam 1, elde 0}$$

$$1 + 1 = 10 \quad \text{toplam 0, elde 1}$$

ELDE BİTİ



$$\begin{array}{l} 1 + 0 + 0 = 01 \\ 1 + 1 + 0 = 10 \\ 1 + 0 + 1 = 10 \\ 1 + 1 + 1 = 11 \end{array}$$

İlk üç kuralda sonuç toplam tek bitten oluşmaktadır. Son kuralda ise biri elde biri de toplam olmak üzere iki bit vardır. İkilik sayılar toplandığında elde (varsa) bir soldaki basamağa eklenir.

Örnek 2.8:

$$(11) + (01) = ?$$

Sağ sütunda toplam $1 + 1 = 0$ ve elde 1 olarak bulunur. Elde orta sütun toplamına katılır ve $1 + 1 + 0 = 0$ ve elde 1 sonucu bulunur. Bu yeni elde de sol sütun toplamına eklenerek $1 + 0 + 0 = 1$ olarak toplamın son biti de yazılır. Elde 1 olduğunda üç bitlik bir toplama yapılması gerekir. Bu durumda toplama kuralları şu şekilde uygulanır:

$$\begin{array}{r}
 \begin{array}{c} 1 \\ 0 \end{array} \leftarrow \begin{array}{c} 1 \\ 1 \end{array} \leftarrow \begin{array}{c} 1 \\ 1 \end{array} \\
 + \begin{array}{c} 0 \\ 0 \end{array} \quad \begin{array}{c} 0 \\ 0 \end{array} \quad \begin{array}{c} 1 \\ 1 \end{array} \\
 \hline
 \begin{array}{c} 1 \\ 0 \\ 0 \end{array}
 \end{array}$$

İKİLİK ÇIKARMA

İkilik çıkarmanın dört kuralı şöyle sıralanır:

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

$$10 - 1 = 1$$

Borç alındığı için aslında sonuç -1 'dir.

İkilik aritmetikte 0 dan 1 çıkarılırken bir soldaki basamaktan borç alınması gerekir. Borç 0'dan 1'i çıkarmak gerektiğinde alınır. Soldaki basamaktan borç alınınca çıkarma yapılan sütunun değeri 10_2 olur. Böylece buraya ikinci kural uygulanabilir.

Örnek 2.9:

101 'den 011 'i çıkarın.

$$\begin{array}{r}
 \begin{array}{ccc} 1 & 0 & 1 \\ - & 0 & 1 & 1 \end{array} \\
 \hline
 \begin{array}{ccc} 0 & 1 & 0 \end{array}
 \end{array}$$

İkinci kolonda 0'dan 1 çıkmaz, böyle durumlarda onluk sistemde olduğu gibi bir soldaki kolondan borç alınır. En soldaki kolonda da bu olay gerçekleşir ise çıkan sonuç negatif olur.

İKİLİK ÇARPMA

İkilik çarpmanın dört temel kuralı vardır:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

İkilik sayıların çarpımı onluk sayılarınki ile aynı biçimdedir. Basamaklar birer birer çarpılır elde edilen ara toplamlar bir sola kaydırılarak yazılır. Bu ara toplamların

toplamı çarpımı verir.

Örnek 2.10:

$$\begin{array}{r} 11 \\ \times 11 \\ \hline 11 \end{array}$$

$$\begin{array}{r} \text{Ara} \\ \text{çarpımlar} \left\{ \begin{array}{r} 11 \\ \times 11 \\ \hline 11 \\ + 11 \\ \hline 1001 \end{array} \right. \end{array}$$

$$\begin{array}{r} \text{Ara} \\ \text{çarpımlar} \left\{ \begin{array}{r} 111 \\ \times 101 \\ \hline 111 \\ 000 \\ + 111 \\ \hline 100011 \end{array} \right. \end{array}$$

İKİLİK BÖLME

İkilik sayılarda bölme onluk sayılardakiyle aynı biçimdedir.

$$\begin{array}{r} 110 \overline{) 11} \\ \underline{11} \\ 000 \end{array}$$

$$\begin{array}{r} 110 \overline{) 10} \\ \underline{10} \\ 010 \\ \underline{10} \\ 000 \end{array}$$

1 VE 2 TÜMLEYEN KAVRAMLARI

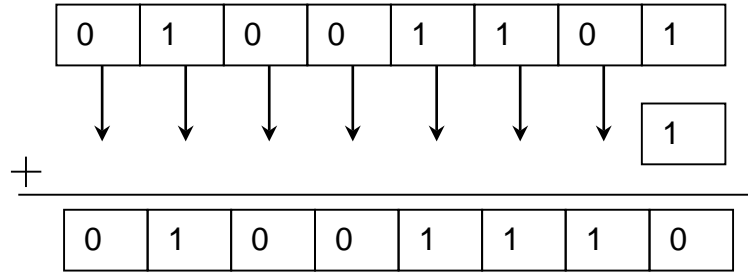
İkilik sayılarda tümleyen kavramı önemlidir, çünkü negatif sayıların gösterimini sağlar ve 2-tümleyen, bilgisayarların negatif sayılarla işlem yapabilmesi için kullanılır.

İkilik sayının 1-tümleyenin bulunması

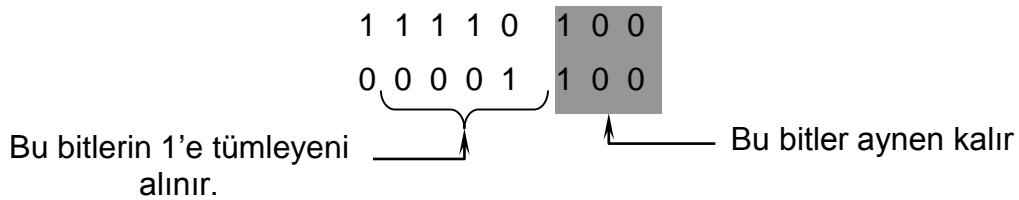
1	0	1	1	0	0	1	0
↓	↓	↓	↓	↓	↓	↓	↓
0	1	0	0	1	1	0	1

İkilik bir sayının 1-tümleyeni bütün bitleri ters çevrilerek kolayca bulunur.

İkili sayının 2-tümleyeninin bulunması



İkili sayıların 2-tümleyeni, 1-tümleyeninin LSB sine 1 eklenerek bulunur. İkili bir sayının 2-tümleyenini bulmak için kullanılan başka bir yöntem de şudur: En sağ bittten (LSB) başlanarak sola doğru bütün bitler ilk 1 de dahil olmak üzere aynen yazılır. Geri kalan bitlerin 1-tümleyeni alınır.

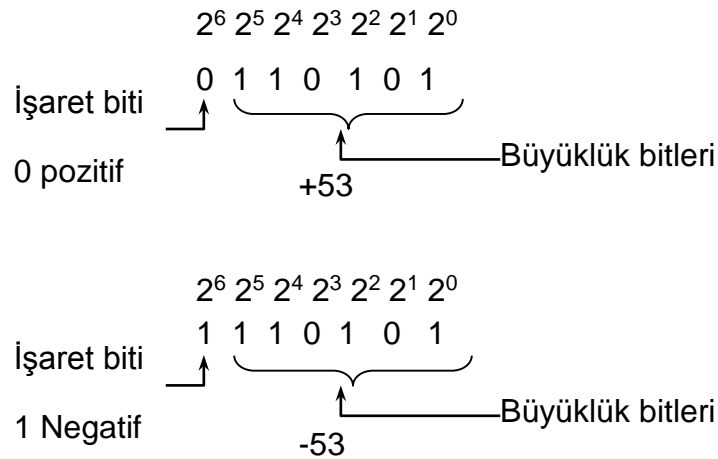


İŞARETLİ SAYILARIN GÖSTERİMİ

İkili sayı sisteminde negatif sayıları belirtmek için ek bir simge yoktur. Bu aritmetikte sayının işaretini belirtmek için sayı değerine ek bir bit kullanılır. İşaretlili sayıların ikili olarak yazılmasında işaret-büyüklik, 2-tümleyen ve 1-tümleyen sistemleri kullanılır. İkili bir sayının ensol biti (MSB), sayının pozitif mi negatif mi olduğunu belirten işaret bitidir. Bu bit "0" pozitif, "1" negatif olduğunu göstermektedir.

İşaret-Büyüklik Sistemi

İşaretlili bir sayı bu sistemde gösterildiğinde en sol bit işaret, geri kalan bitler de büyüklik için kullanılır. Büyüklik, bildiğimiz ikili sistem ile gösterilir. Örnek olarak onluk + 53 ve -53 sayısı bu sistemde aşağıda gösterildiği gibi yazılır. + 53 ile - 53 arasındaki tek gösterim farkı işaret bitidir. Çünkü büyüklik bitleri standart ikili sistemdedir.



İşaret-büyüklük sisteminde gösterilen bir negatif sayının büyüklük bitleri, sayının pozitif karşılığı ile aynıdır ama bundan farklı olarak işaret biti 1 dir.

1-tümleyen sistemi

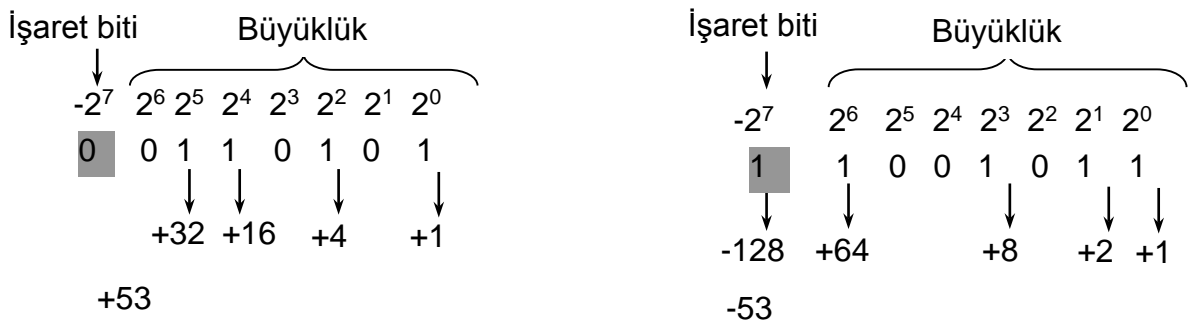
Bu sistemde pozitif sayılar aynı işaret-büyüklük sistemindeki gibi gösterilir. Negatif sayılar ise pozitif karşılığın 1-tümleyeni ile gösterilir. Örneğin - 53, + 53 sayısının 0110101 1-tümleyeni 1001010 olarak gösterilir.

Bu sistemde pozitif sayılar ağırlıklar toplamına göre değerlendirilir. Negatif sayılarda ise işaret bitinin ağırlığına negatif değer verilir ve diğer bit ağırlıkları bu değerle toplandıktan sonra sonuca 1 eklenir.

2-TÜMLEYEN SİSTEMİ

Pozitif sayıların gösterimi bu sistemde de aynıdır. Negatif sayılar ise pozitif karşılığın 2-tümleyeni olarak gösterilir. Örnek olarak yine -53 sayısını verelim. Bu sistemde +53 0110101 olarak, -53 ise 1001011 olarak gösterilir.

2-tümleyen sisteminde pozitif ve negatif sayıların onluk değerleri, 1 bulunan basamakların ağırlıkları toplanarak bulunur. Bilgisayarlarda işaretli sayıların işlenmesinde en yaygın kullanılan sistem 2-tümleyendir.

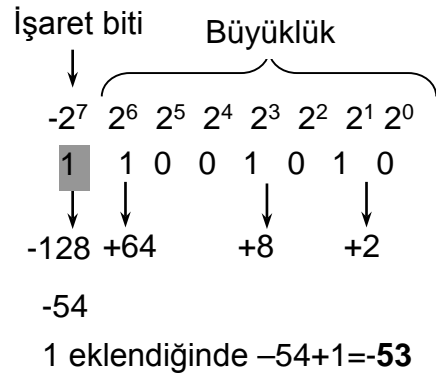
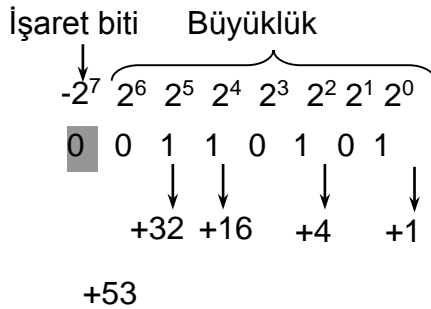


Bu örneklere bakarak, 2-tümleyen sisteminin 1-tümleyen sistemine yeğlenme nedenlerinden birini söyleyebiliriz: Sayının pozitif yada negatif olmasından bağımsız olarak, yalnızca basamak ağırlıklarının toplanmasıyla onluk değer bulunabilmektedir. İşaret-büyüklik sisteminde ağırlıkların toplanması ve işaret bitinin denetimi olmak üzere iki aşama vardır. 1-tümleyen sisteminde bunlara ek olarak eğer sayı negatifse toplama 1 eklenmesi gerekir. Ayrıca 1-tümleyen sisteminde sıfırın iki ayrı karşılığı vardır. 00000000 ve 11111111. Çoğu bilgisayar sistemlerinde 2-tümleyen sisteminin yeğlenme ve kullanılma nedeni aritmetik işlemlerin bu ortamda daha kolay yapılmasıdır.

Örneklere sekiz bitlik sayıların kullanılmasının nedeni, 8-bit grublamanın bilgisayar sistemlerinde standart olmasıdır. Bu gruplar **bayt** olarak adlandırılmışlardır. Sekiz bit kullanılarak 256 değişik sayı gösterilebilir. İki bayt birleştirilerek 16 bitlik bir kod elde edildiğinde 65,536 değişik sayı gösterilebilir. 32 bitlik bir dizi ile 4.295×10^9 ayrı sayı elde edilebilir. Elde edilebilecek en yüksek birleşim sayısı 2^n eşitliğiyle bulunur. 2-tümleyen sisteminde n -bitle gösterilebilecek sayı aralığı,

$$-(2^{n-1}) \text{ ile } +(2^{n-1}-1)$$

sınırları ile belirlidir. Örneğin dört bitle 2-tümleyen sistemde $-(2^3) = -8$ ile $2^3 - 1 = +7$ arası sayılar gösterilebilir. Benzer biçimde sekiz bitle -128 ile +127, onaltı bitle de -32,768 ile +32,767 sayı belirlenebilir.



İŞARETLİ SAYILARLA ARİTMETİK İŞLEMLER

Bilgisayar ve mikroişlemci temelli sistemlerde işaretli sayılar için en yaygın gösterim 2-tümleyen sistemi olduğundan, bu bölümdeki örnekler tümüyle bu sistemi kapsamaktadır. Anlatılacak işlemler gerekirse diğer gösterim sistemlerine de uygulanabilir.

Toplama

Toplamada **artan** ve **arttıran** adlı iki sayı toplanarak **toplam** ve **elde** olmak üzere iki sonuç elde edilir. İki ikilik sayı toplandığında oluşabilecek dört durum vardır:

1. Her iki sayı pozitifdir.

2. Pozitif sayı negatif sayıdan büyüktür.
3. Negatif sayı pozitif sayıdan büyüktür.
4. Her iki sayı negatiftir.

Örnek 2.11:

Aşağıdaki toplama işlemlerini yapın.

- a. Her iki sayı pozitif

$$\begin{array}{r}
 01011101 \quad 93 \\
 + 00011011 \quad + 27 \\
 \hline
 01111000 \quad 120
 \end{array}$$

Toplam pozitifdir ve ikilik olarak doğrudur (gerçektir).

- b. Pozitif sayı negatif sayıdan büyük

Sonuçta çıkan elde gözardı edilir ve böylece sonuç pozitif ve ikilik olarak gerçek olur.

$$\begin{array}{r}
 01011101 \quad 93 \\
 + 11011000 \quad + -40 \\
 \hline
 100110101 \quad 53
 \end{array}$$

Oluşan elde atılır. →

- c. Negatif sayı pozitif sayıdan büyük

$$\begin{array}{r}
 00011101 \quad 29 \\
 + 11011000 \quad + -40 \\
 \hline
 11110101 \quad -11
 \end{array}$$

Toplam negatiftir ve 2-tümleyen olarak gerçektir.

- d. Her iki sayı da negatif

$$\begin{array}{r}
 11011101 \quad -35 \\
 + 11011000 \quad + -40 \\
 \hline
 110110101 \quad -75
 \end{array}$$

Oluşan elde atılır. →

Çıkan elde gözardı edildiğinde çıkan sonuç negatiftir ve 2-tümleyen olarak gerçektir. Bilgisayarlarda negatif sayılar 2-tümleyen formunda saklanır ve görüldüğü gibi toplama işlemi çok basittir: *İki sayı toplanır ve elde çıkarsa gözardı edilir.*

$$\begin{array}{r}
 01011101 \\
 + 01011000 \\
 \hline
 10110101
 \end{array}
 \quad
 \begin{array}{r}
 93 \\
 + 88 \\
 \hline
 181
 \end{array}$$

İşaret biti hatalı \longrightarrow $\underbrace{10110101}$

Büyüklik hatalı \longleftarrow $\underbrace{10110101}$

Taşma Durumu İki sayı toplandığında çıkan sonucu gösterebilmek için gereken bit sayısı toplanan sayılardaki bit adedini aşarsa, yanlış işaret biti ile belirlenen taşma durumu oluşur. Taşma yalnızca her iki sayı da pozitif yada negatifse oluşur. Taşma durumunu 8-bitlik bir örnekle gösterelim:

181 sayısını ikilik olarak gösterebilmek için sekiz bit gereklidir. Sayılarda yedişer büyüklik biti olduğu için (birişer tanesi işaret biti) sonucun işaret bitine taşmayı gösteren bir elde gelir. İşaret biti çıkması gerekenden farklı ise, işaret biti ve büyüklik bitleri hatalıdır. Bu hata taşma ile oluşabilir veya iki negatif sayının toplanmasında oluşur. İki den fazla sayıyı toplamak ancak bu sayıları ikiye ikiye sırayla toplamakla olur. İlk iki sayı toplandıktan sonra bunların toplamına üçüncü sayı, yeni toplama da dördüncü sayı eklenerek işlem sürdürülür.

Örnek 2.12:

01000100, 00011011, 00001110 ve 00010010 sayılarını toplayın.

$$\begin{array}{r}
 01000100 \\
 + 00011011 \\
 \hline
 01011111 \\
 + 00001110 \\
 \hline
 01101101 \\
 + 00010010 \\
 \hline
 01111111
 \end{array}
 \quad
 \begin{array}{r}
 68 \\
 + 27 \\
 \hline
 95 \\
 + 14 \\
 \hline
 109 \\
 + 18 \\
 \hline
 127
 \end{array}$$

İlk iki sayı toplanır

Üçüncü sayı toplama eklenir

dördüncü sayı toplama eklenir

ÇIKARMA

Çıkarma, toplamının özel bir durumudur. Örneğin +9 dan (**eksilen**) +6 yı (**eksilten**) çıkarmakla, -6 ile +9 u toplamak aynı sonucu verir. Çıkarma işlemi, *eksiltenin işareti değiştirilip eksilen ile toplanmasıyla* gerçekleştirilir. Çıkarma işleminin sonucuna **fark** denilir. Pozitif yada negatif bir sayının işareti, sayının **2-tümleyeni alınarak değiştirilir**. Örnek olarak pozitif 00000100 (+4) sayısının 2-tümleyeni alınınca elde edilen 11111100 sayısı -128+64+32+16+8+4 = -4 onluk değerini verir. Diğer bir örnek olarak 11101101 (-19) sayısının 2-tümleyeni alırsak bulacağımız 00010011 sayısı 16+2+1 = 19 onluk değerini verir. İşaretleli iki sayıyı çıkarmak için eksiltenin 2-tümleyeni alınır ve varsa elde biti gözardı edilir.

Örnek 2.13:

Aşağıdaki çıkarma işlemlerini yapın.

a. $00001000 - 00000011 = 00001000 + 11111101 = 100000101$

Elde atılır ve sonuç: $00000101, 8 - 3 = 8 + (-3) = 5$

b. $00001100 - 11110111 \quad 12 - (-9) = 12 + 9 = 21$

c. $11100111 - 00010011 \quad -25 - (+19) = -25 + (-19) = -44$

d. $10001000 - 11100010 \quad -120 - (-30) = -120 + 30 = -90$

ÇARPMA

Çarpma işlemindeki sayılar, çarpılan, çarpan ve çarpım olarak adlandırılırlar. Çoğu bilgisayarda çarpma işlemi de çıkarma gibi toplama işlemi kullanılarak yapılır. Doğrudan toplama ve kısmi çarpımlar, toplama kullanılarak çarpma yapma da kullanılan temel yöntemlerdir.

Doğrudan toplama yönteminde çarpılan, çarpanın sayısına eşit sayıda kendisiyle toplanır. Örneğin yukarıdaki çarpma $8 + 8 + 8 = 24$ olarak gerçekleştirilir.

$$\begin{array}{r} 3 \\ \text{Çarpıla} \\ n \text{ } \underline{\hspace{1cm}} \\ \times 8 \text{ Çarpan} \end{array}$$

Bu yöntemin sakıncası, çarpanın büyümesi durumunda işlem süresinin çok uzamasıdır. Eğer örneğin 350 ile 75' i çarpmak istersek 350 kendisi ile 75 kez toplanmalıdır. Çarpma işlemini ifade ederken kere terimini kullanmamızın nedeni de budur herhalde.

Kısmi çarpımlar yöntemi daha tanıdık gelecektir çünkü elle çarpma yaparken hep kullandığımız yöntemin ta kendisidir. Çarpılan sayı çarpanın en sağ basamağından (enaz önemli bit - LSB) başlanıp sola doğru her basamağıyla birer birer çarpılır. Bu çarpımların sonucuna kısmi çarpım denir ve her biri sola doğru bir basamak kaydırılarak sırayla toplanır.

Çarpımın işareti çarpan ve çarpılanın işaretlerine bağlıdır.

- İşaretler aynı ise çarpım pozitifdir.
- İşaretler farklı ise çarpım negatiftir.

İkili sayılar çarpılırken gerçek (tümlenmemiş) durumda olmalıdırlar.

Örnek 2.14:

İşareti 01001101 (çarpılan) ve 00000100 (çarpan) sayılarını çarpın.

$$\begin{array}{r}
 01001101 \\
 + 01001101 \\
 \hline
 10011010 \\
 + 01001101 \\
 \hline
 11100111 \\
 + 01001101 \\
 \hline
 100110100
 \end{array}$$

Her iki sayı da pozitif ve tümlenmemiş durumda olduğuna göre çarpım pozitif olacaktır. Çarpanın onluk değeri 4 olduğuna göre çarpılan dört kez kendisi ile toplanır.

Çarpan ve çarpılanın işaret bitlerinin aynı olup olmadığına bakılır. Bütün negatif sayılar tümlenmemiş durumda olmalıdır. Bilgisayar sistemlerinin çoğunda negatif sayılar 2-tümleyen olarak saklandığından negatif sayıları gerçek ikilik duruma getirmek için 2-tümleyen işlemi kullanılarak dönüşüm yapılır. Enaz önemli bit LSB den başlayarak kısmi çarpımlar yapılır. Çarpan biti 1 olduğunda kısmi çarpım çarpılana eşit olur. Çarpan biti 0 iken kısmi çarpım sonucu sıfırdır. Her kısmi çarpım sola doğru bir kaydırılarak yazılır.

Çıkan her yeni kısmi çarpım, önceki kısmi çarpımlar toplamına eklenerek son çarpım bulunur. Başlangıçta belirlenen işaret negatifse çarpımın 2-tümleyeni alınır. Pozitif sonuçlar için çarpım gerçektir. İşaret biti çarpıma eklenir.

Örnek 2.15:

İşaretili 01010011 (çarpılan) ve 11000101 (çarpan) ikilik sayılarını çarpın.

1010011	Çarpılan
X 0111011	Çarpan
<u>1010011</u>	1. bitin çarpımı
+ 1010011	2. bitin çarpımı
<u>11111001</u>	Ara toplam
+ 0000000	3. bitin çarpımı
<u>011111001</u>	Ara toplam
+ 1010011	4. bitin çarpımı
<u>1110010001</u>	Ara toplam
+ 1010011	5. bitin çarpımı
<u>100011000001</u>	Ara toplam
+ 1010011	6. bitin çarpımı
<u>1001100100001</u>	Ara toplam
+ 0000000	7. bitin çarpımı
<u>1001100100001</u>	Sonuç (Çarpım)

Çarpılanın işaret biti 0 çarpanın işaret biti 1 olduğu için sonucun işaret biti 1 (negatif)

olacaktır. Çarpanı gerçek duruma getirmek için 2-tümleyeni alınır.

1001100100001 2-tümleyeni 0110011011111 olur.

Başlangıçta sonucun işaret biti 1 olarak belirlendiğine göre çarpımın 2-tümleyeni alınır ve işaret biti eklenir. 1001100100001 sayısının 2-tümleyeni 0110011011111 olur. İşaret biti de eklenince işaretli çarpım sonucu

10110011011111 olur.

Bölme

Bölme işlemindeki sayılar **bölen**, **bölünen** ve **bölüm** olarak adlandırılırlar. Bölme işlemi bilgisayarlarda çıkarma işlemi kullanılarak yapılır. Çıkarma işlemi toplama ile yapıldığına göre bölme işlemi de toplama ile yapılabilir. Bölme işleminin sonucu, bölüm, bölünenin içinde kaç tane bölen olduğunu belirtir. Yani bölen, bölünenden, bölüm sayısı kez çıkarılabilir. Örnek olarak 21 sayısını 7 sayısına bölelim;

$$21 - 7 = 14 - 7 = 7 - 7 = 0$$

Bu basit örnekte, sıfır kalan elde edilene dek bölen bölünenden üç kez çıkarılmıştır. Buna göre bölüm 3'tür.

Bölümün işareti bölen ve bölünenin işaretlerine bağlıdır.

- işaretler aynı ise bölüm pozitifdir.
- işaretler farklı ise bölüm negatiftir.

İki ikilik sayı bölünürken her iki sayı da gerçek (tümlememiş) durumda olmalıdır. Bölme işlemi yapılırken şu sıra izlenir;

- Bölen ve bölünen sayıların işaretlerinin aynı olup olmadığına bakılarak sonucun işaretinin ne olacağı belirlenir. Ayrıca bölüm yazacı sıfırlanır.
- 2-tümleyen toplama kullanılarak bölen bölünenden çıkarılıp ilk kısmi kalan bulunur ve bölüme 1 eklenir. Eğer bu kısmi kalan pozitifse 3. aşamaya geçilir. Eğer sonuç sıfır yada negatifse bölme tamamlanmıştır.
- Bölen kısmi kalandan çıkarılarak bölüme 1 eklenir. Eğer sonuç pozitifse işlem sürdürülür. Sonuç sıfır yada negatifse bölme tamamlanmıştır.

Örnek 2.16:

01100100 sayısını 00011001 sayısına bölün.

$$\begin{array}{rcl}
 01100100 & \text{Bölünen} & \\
 + 11100111 & \text{Bölenin 2-ye tümlenmiş hali} & \\
 \hline
 01001011 & \text{1. ara kalan} &
 \end{array}$$

Bölüm kaydedicisi 1 artırılır.

$$00000000+1=00000001$$

$$\begin{array}{rcl}
 01001011 & \text{1. ara kalan} & \\
 + 11100111 & \text{Bölenin 2-ye tümlenmiş hali} & \\
 \hline
 00110010 & \text{2. ara kalan} &
 \end{array}$$

Bölüm kaydedicisi 1 artırılır.

$$00000001+1=00000010$$

$$\begin{array}{rcl}
 00110010 & \text{2. ara kalan} & \\
 + 11100111 & \text{Bölenin 2-ye tümlenmiş hali} & \\
 \hline
 00011001 & \text{3. ara kalan} &
 \end{array}$$

Bölüm kaydedicisi 1 artırılır.

$$00000010+1=00000011$$

$$\begin{array}{rcl}
 00011001 & \text{3. ara kalan} & \\
 + 11100111 & \text{Bölenin 2-ye tümlenmiş hali} & \\
 \hline
 00000000 & \text{4. ara kalan} &
 \end{array}$$

Bölüm kaydedicisi 1 artırılır.

$$00000011+1=0000100$$

Her iki sayı da pozitif olduğuna göre bölüm pozitif olacaktır. Bölümün yazılacağı kaydedici 00000000 durumuna getirilir.

2-tümleyen toplama yöntemi ile (eldelerin atıldığını unutmadan) bölen bölünenden çıkarılır:

$$\text{Bölüm}=00000100$$

ONALTILIK SAYILAR

Onaltılık sayı sisteminin tabanı onaltıdır, yani onaltı ayrı karakter (basamak) içerir. Sayısal sistemlerin çoğunda ikilik veriler dört ve katları sayıda bit içeren gruplar olarak işlendiğinden onaltılık sistemin kullanılması çok uygun olmaktadır. Çünkü her onaltılık basamak 4-bitlik bir ikilik sayıya karşılık gelmektedir. Bu sistemde kullanılan karakterlerin on tanesi nümerik altı tanesi ise alfabetiktir. A, B, C, D, E, ve F harfleri kullanılarak yazılan sayılar başlangıçta garip gelebilir ama aslında bütün sayı sistemleri bir dizi simgeden başka bir şey değildir. Bu simgelerin hangi çokluğu belirttiğini öğrendikten sonra simgelerin biçimlerinin bir önemi yoktur.

F'ye kadar saydıktan sonra saymaya nasıl devam edilecek? Aynı ikilik ve onluk sistemlerdeki gibi bir basamak eklenir ve bütün simgeler sırayla yeniden sayılır:

E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30,

İki onaltılık basamakla FF_{16} yani onluk 255 sayısına dek sayılır, daha büyük sayılar için daha fazla basamak eklenir. Örneğin 100_{16} onluk 256'ya eşittir. Dört basamakla yazılabilen en büyük sayı $FFFF_{16}$ yani onluk 65,535'tir.

İKİLİK- ONALTILIK DÖNÜŞÜM

İkilik bir sayının onaltılık sisteme çevrilmesi çok kolaydır. Sayı en sağ bittten başlanarak dört bitlik gruplara ayrılır ve her bir grup, karşılığı olan onaltılık basamakla gösterilir.

Örnek 2.17:

1100101001010111_2 ve 111111000101101001_2 ikilik sayılarını onaltılık olarak yazın.

Birinci sayıdaki rakam adedi 4'ün katı olduğu için herhangi bir ek işleme gerek kalmadan dönüşüm yapılır. İkinci sayıda ise soldaki grubu dört bite tamamlamak için sola doğru sıfır(lar) eklenir.

1 1 0 0 1 0 1 0 0 1 0 1 0 1 1 1
C A 5 7

Sonradan eklenen sıfırlar

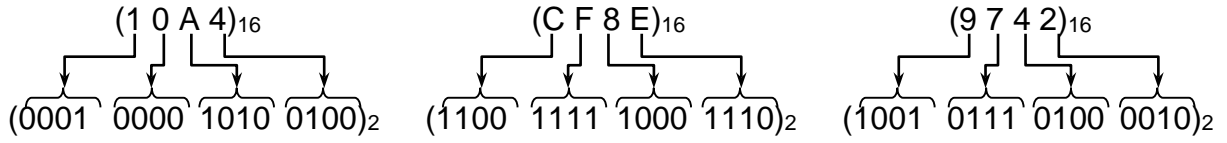
0 0 1 1 1 1 1 1 0 0 0 1 0 1 1 0 1 0 0 1
3 F 1 6 9

ONALTILIK-İKİLİK DÖNÜŞÜM

Onaltılık bir sayıyı ikilik olarak yazmak için ikilik-onaltılık dönüşüm işlemi tersine çevrilir yani her bir onaltılık simgenin yerine dört bitlik ikilik karşılığı yazılır.

Örnek 2.18:

$10A4_{16}$, $CF8E_{16}$ ve 9742_{16} onaltılık sayılarını ikilik olarak yazın.



En soldaki sıfırlar atılır.

$$(100010100100)_2$$

$$(1100111110001110)_2$$

$$(1001011101000010)_2$$

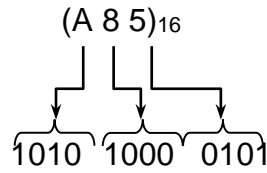
Sayının ikilik eşiti yazılırken soldaki sıfırların çıkarılması uygun olur.

ONALTILIK-ONLUK DÖNÜŞÜM

Onaltılık bir sayının onluk sisteme çevrilme yollarından birincisi, önce ikilik oradan da onluk sisteme geçmektir.

Örnek 2.19:

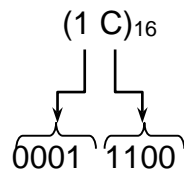
$1C_{16}$ ve $A85_{16}$ sayılarını onluk sistemde gösterin.



$$(101010000101)_2$$

$$=1X2^{11}+0X2^{10}+1X2^9+0X2^8+1X2^7+0X2^6+0X2^5+0X2^4+0X2^3+1X2^2+0X2^1+1X2^0$$

$$=2693_{10}$$



En soldaki sıfırlar atılır.

$$(11100)_2$$

$$=1X2^4+1X2^3+1X2^2+0X2^1+0X2^0$$

$$=28_{10}$$

Diğer bir yöntem de, onaltılık sayının her basamağındaki sayının onluk değerinin o basamağın ağırlığıyla çarpılıp bu çarpımların toplanmasıdır.

Örnek 2.20:

$E5_{16}$ ve $B2F8_{16}$ sayılarını onluk sisteme dönüştürün.

$$E5_{16} = (E \times 16) + (5 \times 1) = (14 \times 16) + (5 \times 1) = 224 + 5 = 229_{10}$$

$$\begin{aligned} B2F8_{16} &= (B \times 96) + (2 \times 256) + (F \times 16) + (8 \times 1) \\ &= (11 \times 4096) + (2 \times 256) + (15 \times 16) + (8 \times 1) \\ &= 45,056 + 512 + 240 + 8 = 45816_{10} \end{aligned}$$

1.1.1.1 ONLUK-ONALTILIK DÖNÜŞÜM

Onluk sayıyı sürekli 16'ya bölerek kalanlardan onaltılık sayı elde edilebilir. İlk kalandan en az önemli sayı DDS (LSD), sonuncudan da en önemli sayı YDS (MSD) elde edilir. Tamsayısı sıfır olunca işlem biter.

ONALTILIK TOPLAMA

Onaltılık sayılarla toplama basamak değerlerinin 0 ile 15 arasında değiştiği akılda tutulursa doğrudan doğruya onluk sistemdeki gibi yapılabilir. Toplama yaparken şu kurallara uyulmalıdır.

Her zaman bütün sayıların onluk değerlerini gözönüne alın. Örneğin

$$5_{16} = 5_{10} \text{ ve } C_{16} = 12_{10}$$

İki sayının toplamı 15_{10} yada daha azsa, buna karşılık gelen onaltılık sayıyı bu sütunun toplamı olarak alta yazın. Eğer bu iki dijitin toplamı 15_{10} 'ten büyükse, 16_{10} 'dan büyük olan kısmını toplam olarak sütunun altına yazın ve soldaki sütuna elde 1 taşıyın.

Örnek 2.21:

Aşağıdaki verilen sayıları toplayın.

a. $23_{16} + 16_{16} = ?$

sağ sütun: $3_{16} + 6_{16} = 3_{10} + 6_{10} = 9_{10} = 9_{16}$

sol sütun: $2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16}$

toplam: 39_{16}

b. $2B_{16} + 84_{16} = ?$

sağ sütun: $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$

sol sütun: $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

toplam: AF_{16}

c. $DF_{16} + AC_{16} = ?$

sağ sütun: $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10} = 1B_{16}$ (elde var)

sol sütun: $1_{16} + D_{16} + A_{16} = 1_{10} + 13_{10} + 10_{10} = 24_{10} = 18_{16}$

toplam: $18B_{16}$

2-TÜMLEYEN YÖNTEMİYLE ONALTILIK ÇIKARMA

Onaltılık bir sayı ikilik bir sayıyı gösterebildiğine göre, ikilik bir sayının 2-tümleyenini de gösterebilir. Örneğin 11001001_2 onaltılık karşılığı $C9_{16}$ 'dır. Bu ikilik sayının 2-tümleyeni 00110111 onaltılık olarak yazılırsa 37_{16} elde edilir. Daha önceden de bildiğimiz gibi bir sayının 2-tümleyeni, toplama işlemi kullanarak çıkarma yapmamızı sağlar. Bu yöntem onaltılık sayılarda da geçerlidir.

Örnek 2.22:

a. $84_{16} - 2A_{16}$ işlemini yapın.

$2A_{16} = 00101010_2$ sayısının 2-tümleyeni alınır $11010110_2 = D6_{16}$ ve 84_{16} ile toplanır.

$$\begin{array}{r}
 84_{16} \\
 + \quad D6_{16} \\
 \hline
 \text{Elde atılır} \leftarrow 1 \quad 5A_{16}
 \end{array}$$

Oluşan elde atılır kalan sonuç gerçektir.

b. $C3_{16} - 0B_{16}$ işlemini yapın.

$0B_{16} = 00001011_2$ sayısının 2-tümleyeni alınır $11110101_2 = F5_{16}$ ve $C3_{16}$ ile toplanır.

$$\begin{array}{r}
 C3_{16} \\
 + \quad F5_{16} \\
 \hline
 \text{Elde atılır} \leftarrow 1 \quad B8_{16}
 \end{array}$$

Oluşan elde atılır kalan sonuç gerçektir.

c. $A5_{16} - B8_{16}$ işlemini yapın.

$B8_{16} = 10111000_2$ sayısının 2-tümleyeni alınır $01001000_2 = 48_{16}$ ve $A5_{16}$ ile

toplanır.

$$ED_{16} = 11101101_2 \rightarrow 00010011_2 = -13_{16}$$

$$\begin{array}{r} A_{16} \\ + 48_{16} \\ \hline \text{Elde yok} \leftarrow 0 \quad ED_{16} \end{array}$$

Elde yoktur ve sonuç ikiye tümlenmiş haldedir. Sonucun ikiye tümleyeni alınır ve önüne eksi işareti konur.

İKİLİK KODLU ONLUK SAYILAR (BCD)

İKO her onluk basamağı bir ikilik kodla göstermenin bir yoludur. İKO sisteminde yalnızca on kod grubu bulunduğu için onluk ve İKO sistemleri arasında dönüşüm çok kolaydır. Onluk sistemde okuyup yazmayı sayıcişvediğimiz için İKO ikilik sistemlerle aramızda çok uygun bir ara aşamadır. En yaygın olarak kullanıldığı yerler tuş takımları ve dijital göstergelerdir.

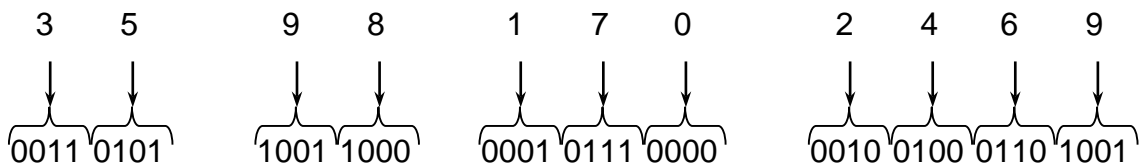
8421 kodu

8421 bir tür İKO'dur. İkilik kodlu onluk demek, 0'dan 9'a kadar olan sayılardan her birinin, ikilik bir kodla gösterilmesi demektir. 8421 adlandırması dört bitin ikilik ağırlıklarını belirtir ($2^3, 2^2, 2^1, 2^0$). 8421 kodu ile onluk sistem arasında dönüşümün kolaylığı, bu sistemin en büyük üstünlüğüdür. Bütün bilinmesi gereken, aşağıdaki tabloda gösterilmiştir. 8421 kodu dört bitten oluşur, bu bitler ile 16 değer kodlanabilir fakat BCD (İKO) bunlardan 10 tanesini kullanır. Diğerleri geçersiz kodlardır. Onluk sayıyı İKO olarak kodlamak için kolayca onluk sayının rakamları ayrı ayrı dörder bit olarak kodlanır. Soldaki sıfırlar asla atılmaz.

ONLUK	0	1	2	3	4	5	6	7	8	9
İKO	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Örnek 2.23:

35, 98, 170, 2469 sayılarını İKO olarak kodlayın.



İKO TOPLAMA

İKO nümerik bir koddur aritmetik işlemler yapılabilir. Toplama yapılırken bazı kurallara uyulmalıdır. Bu kuralları şöyle sıralayabiliriz.

- İKO sayıları ikilik toplamının kurallarını uygulayarak yapın.
- Eğer 4 bit toplam 9 ve 9'dan küçük ise sonuç doğrudur.
- Eğer 4 bit toplam 9'dan büyük ise İKO sonuç elde etmek için toplama 6 (0110) eklenir. Bu işlem sonrası elde oluşur ise bir soldaki dörtlülük gruba aktarılır.

Örnek 2.24:

Aşağıdaki toplama işlemlerini yapın.

$$\begin{array}{r}
 1001 \\
 + 0100 \\
 \hline
 1101 \\
 + 0110 \\
 \hline
 0011
 \end{array}$$

Geçersiz İKO sayı
6 ekle doğru sayıyı bul.
Geçerli İKO sayı

0001 0011
↓ ↓
1 3

Oluşan elde bir sola aktarılır.

$$\begin{array}{r}
 0011 \\
 + 0100 \\
 \hline
 0111
 \end{array}$$

$$\begin{array}{r}
 0010 \quad 0011 \\
 + 0001 \quad 0101 \\
 \hline
 0011 \quad 1000
 \end{array}$$

$$\begin{array}{r}
 1000 \quad 0110 \\
 + 0001 \quad 0011 \\
 \hline
 1001 \quad 1001
 \end{array}$$

$$\begin{array}{r}
 0100 \quad 1101 \quad 0000 \\
 + 0100 \quad 1001 \quad 0111 \\
 \hline
 1001 \quad 0110 \quad 0111
 \end{array}$$

$$\begin{array}{r}
 1001 \\
 + 1001 \\
 \hline
 0010 \\
 + 0110 \\
 \hline
 1000
 \end{array}$$

Elde oluştu yanlış sonuç.
6 ekle doğru sayıyı bul.
Geçerli İKO sayı

0001 1000
↓ ↓
1 3

Oluşan elde bir sola aktarılır.

SAYISAL KODLAR

Şu ana kadar öğrendikleriniz dışın bir çok kod vardır. Bunların bazıları nümerik bazıları ise alfanümerik kodlardır. Nümerik kodlarla sadece sayılar kodlanabilir, alfanümerik kodlarla ise semboller, komutlar ve sayılar kodlanabilir. Bu bölümde

İKO'ya göre daha az kullanılan nümerik kodlardan Gray (Yansıtılmış) kodu ve 3-artı kodlarını öğreneceksiniz. Alfannümerik kodlardan en yaygın kullanılan ASCII kodunu inceleyeceğiz.

GRAY KODU

Gray kodu rakamların ağırlığı olmayan ve aritmetik olmayan bir koddur. En önemli özelliği bir durumdan diğer duruma geçerken sadece bir bit değeri değiştirebilir. Bu özelliği bazı uygulamalarda özellikle bir durumdan diğer duruma geçerken birden fazla bitin değişmesi ile oluşan hatalı veya şüpheli sonucun istenmediği uygulamalarda kullanılmasına olanak sağlamıştır. Mil dönme hareketinin belirleyen arttırmalı kodlayıcılarda bu kod kullanılmaktadır. Tablo-2.3'te ikilik, onluk ve gray kodu karşılıkları gösterilmiştir. İkilik sayılarda olduğu gibi gray kodu da istenilen bit adedi ile kodlanabilir.

ONLUK	İKİLİ	GRAY	ONLUK	İKİLİ	GRAY
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Tablo-2.3

1.1.1.2 İKİLİK-GRAY DÖNÜŞÜM

Gray kodunun hafızamızda kalması zordur. Onun yerine gerektiğinde ikilikten dönüşüm yapmak daha kolaydır. Gray kodunda aritmetik işlem yapılamaması da bunu zorunlu hale getirmektedir. Aşağıdaki kurallar uygulanarak dönüşüm yapılır.

- En yüksek değerli bit (MSB), en soldaki bit, ikilik ile Gray'de aynıdır.
- Soldan sağa giderken yan yana olan iki bit topla ve gray biti olarak yaz. Oluşan eldeyi at.

ÖRNEK 1: 10110₂ sayısını gray koduna dönüştürün.

Adım 1: MSB'yi ikilikten aynen Gray'e yaz.

1	0	1	1	0	İKİLİK
↓					
1					GRAY

Adım 2: Soldan başlayarak ilk iki biti topla, graye soldan ikinci bit olarak yaz.

1 + 0	1	1	0	İKİLİK
↓				
1	1			GRAY

Adım 3: Soldan başlayarak ikinci ve üçüncü biti topla, graye soldan üçüncü bit olarak yaz.

1	0 + 1	1	0	İKİLİK
	↓			
1	1	1		GRAY

Adım 4: Soldan başlayarak üçüncü ve dördüncü biti topla, graye soldan dördüncü bit olarak yaz.

1	0	1 + 1	0	İKİLİK
		↓		
1	1	1	0	GRAY

Adım 5: Soldan başlayarak dördüncü ve beşinci biti topla, graye soldan beşinci bit olarak yaz.

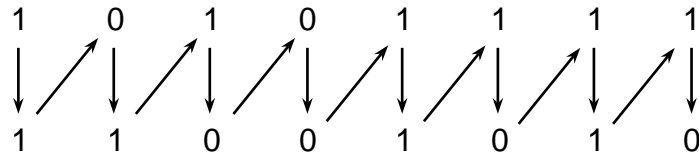
1	0	1	1 + 0	İKİLİK
			↓	
1	1	1	0	1
				GRAY

GRAY-İKİLİ DÖNÜŞÜM

İkilik-gray dönüşümüne benzer bir kural uygulanarak dönüşüm gerçekleştirilir. Dönüşümün kurallarını aşağıdaki gibi özetleyebiliriz.

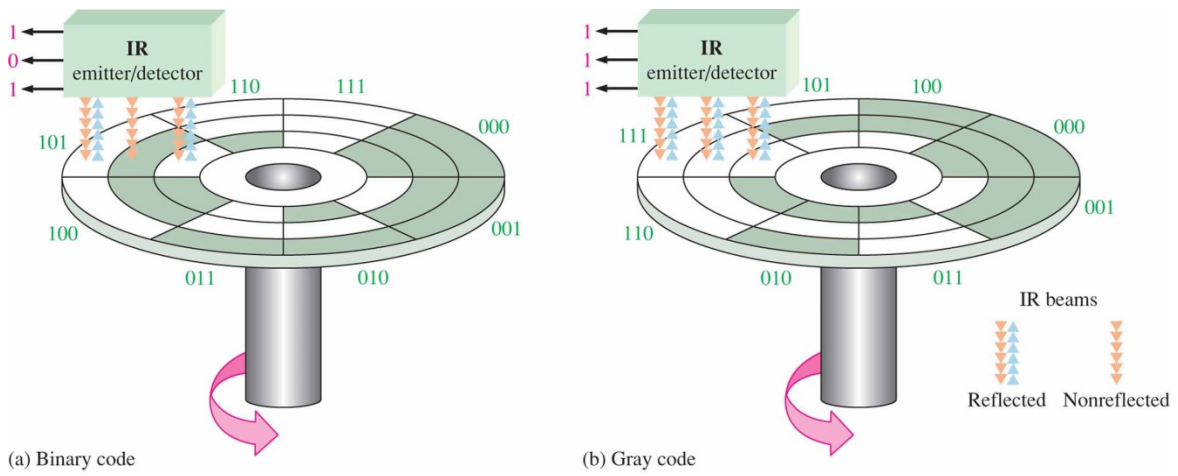
- En yüksek değerli bit (MSB), en soldaki bit, ikilik ile grayde aynıdır.
- İkiliğin en soldaki biti ile Gray kodunun soldan ikinci biti toplanarak ikiliye ikinci bit olarak yazılır. Elde varsa atılır.

Örnek: Aşağıdaki dönüşümü yapın.



GRAY KODU ARTTIRMALI ENKODER

Şekil-2.1’de 3 bitlik arttırmalı enkoderin prensip çizimleri ikilik ve gray olarak verilmiştir. İkilik olanda siyahtan beyaza geçişler birden fazla bitte aynı anda olduğundan hatalı değer elde edilmesi olasılığı yüksektir. Gray kodlu olan enkoder de ise geçiş aynı anda sadece 1 bittedir. Beyaz bölgelerde çıkıştan mantık 0 alınır, siyah bölgelerden mantık 1 alınır.



Şekil-2.1

3-ARTI KOD

İKO koda benzer bir koddur, sadece İKO’nun 4 bitlik ikilik kodlanmış sayısına 3 eklenerek elde edilir. Bitlerin pozisyonundan kaynaklanan herhangi bir ağırlıkları yoktur. Tablo-2.4’te 15’e kadar İKO ve 3-ARTI kodlamalar verilmiştir. Bu kod sisteminde hiçbir zaman tüm bitler aynı anda sıfır olmaz. En büyük özelliği budur.

ONLUK	İKO	3-ARTI	ONLUK	İKO	3-ARTI
0	0000	0011	8	1000	1011
1	0001	0100	9	1001	1100
2	0010	0101	10	0001 0000	0100 0011
3	0011	0110	11	0001 0001	0100 0100
4	0100	0111	12	0001 0010	0100 0101
5	0101	1000	13	0001 0011	0100 0110
6	0110	1001	14	0001 0100	0100 0111
7	0111	1010	15	0001 0101	0100 1000

Tablo-2.4



TEMEL MANTIK İŞLEMLERİ

GİRİŞ

Mantık temel anlamıyla belli koşullar sağlandığında belli bir önermenin doğru olduğunu söyleyen bir bilim dalıdır. Örneğin, "Işık yanıyor" önermesi ya doğrudur yada yanlış. "Ampul bozuk değil" ve "düğme açık" önermeleri de aynı şekildedir. Günlük yaşamda karşılaşılan çoğu sorun, işlem ve durum önerme yada mantık işlevi (logic function) olarak ifade edilebilir. Bunlar doğru/yanlış yada evet/hayır durumları olduğuna göre iki durum (1 ve 0) içeren dijital devreler bu durumlara rahatlıkla uygulanabilir.

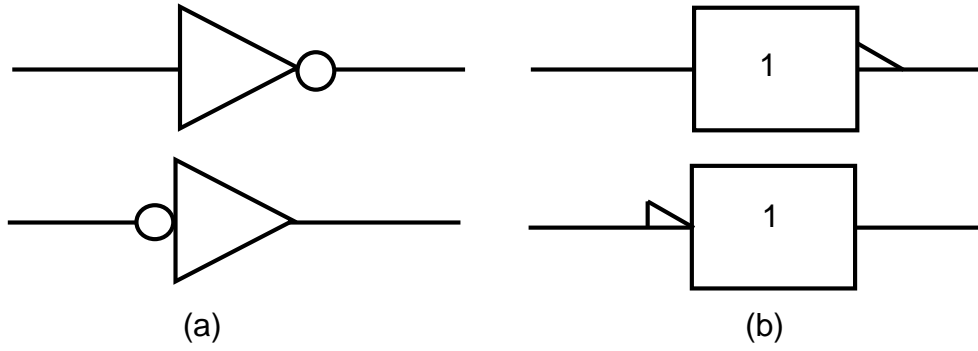
DEĞİL (NOT) İŞLEMİ

DEĞİL işlemi bir mantık değerini tersine dönüştürür. Giriş YÜKSEKse çıkış DÜŞÜKTür. Eğer giriş DÜŞÜKse çıkış YÜKSEK olur. DEĞİL işlemi evirici (inverter) denilen dijital mantık devresi ile gerçekleştirilir. Evirici mantık sembolü Şekil-3.1'de gösterilmiştir. Şekil-3.1(a)'da üçgen yapılı sembolü (b)'de ise kare yapılı sembolü verilmiştir. Üçgen yapıda değil deliği önce çıkışa konmuştur, bunun anlamı çıkışın etkin durumunun DÜŞÜK seviye olduğudur. Girişe konan delik ise girişin etkin durumunun DÜŞÜK seviye olduğunu belirtir. Her iki sembolü bilmek ileriki çalışmalarınızda yarar sağlayacaktır.

DEĞİL geçidinin doğruluk tablosu Tablo-3.1'de verilmiştir. Giriş YÜKSEK ise çıkış DÜŞÜK, giriş DÜŞÜK ise çıkış YÜKSEKTir.

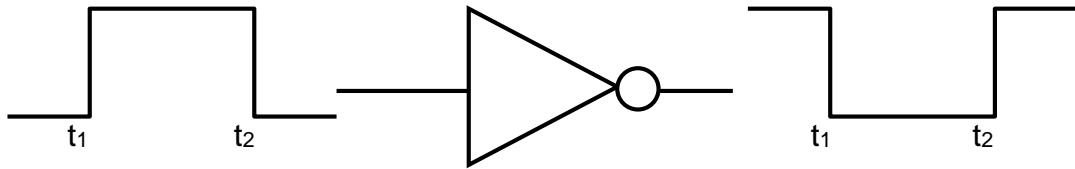
Giriş	Çıkış
DÜŞÜK (0)	YÜKSEK (1)
YÜKSEK (1)	DÜŞÜK (0)

Tablo-3.1 DEĞİL geçidinin doğruluk tablosu.



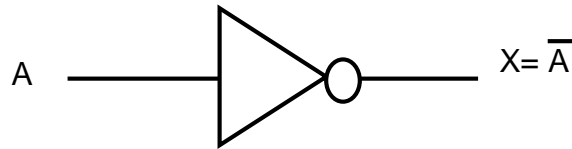
(Şekil-3.1 DEĞİL geçidinin mantık sembolleri)

DEĞİL geçidi girişine bir vuru uygulanırsa bu vurunun tersi çıkıştan elde edilir. Şekil-3.2’de giriş ve çıkış vuruları verilmiştir.



Şekil-3.2 Değil geçidini girişine uygulanan vuru ve çıkışından elde edilen vuru.

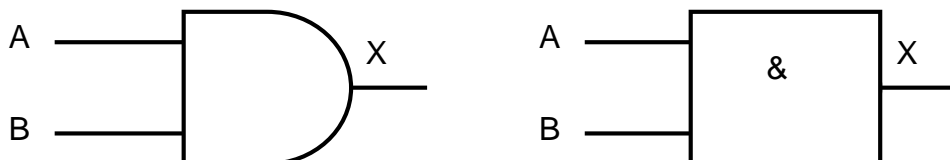
DEĞİL geçidi girişine uygulanan ikili sayının 1-e tümleyenini alır. Girişine A uygulanırsa çıkışından A’nın değili elde edilir.



Şekil-3.3 DEĞİL mantık işlevi

VE (AND) İŞLEMİ

VE işlemi, ancak bütün girişleri YÜKSEK olduğunda çıkışı YÜKSEK olan bir işlemdir. Girişlerden en az biri DÜŞÜK olursa çıkış DÜŞÜK olur. Bu mantık işlem, VE geçidi (AND gate) denilen mantık devresiyle gerçekleştirilir. Şekil-3.4’te VE geçidinin kullanılan mantık gösterimleri verilmiştir.



Şekil-3.4 VE geçidinin simgeleri

VE geçidinin doğruluk tablosu Tablo-3.2’de verilmiştir. Bu tabloda sadece iki girişli bir VE geçidinin değerleri verilmiştir. Üç girişli VE geçidinin doğruluk tablosu ise

Tablo-3.3'de verilmiştir. Bu iki tabloyu kullanarak daha fazla sayıda girişi bulunan VE geçidinin doğruluk tablosunu elde edebilirsiniz.

Şekil-3.5'te iki girişli bir VE geçidinin girişine verilen işaretler uygulanmıştır. Çıkışı doğruluk tablosundan yararlanılarak şekilde görüldüğü gibi çizilmiştir. Giriş işaretleri ile çıkış işaretinin aynı zaman aralıkları için çizilmesine o elemanın zamanlama diyagramı adı verildiğini birinci bölümde öğrenmiştiniz. Zamanlama diyagramı yeterli sayıda girişi olan osilaskoplardan veya lojik analizörlerden görüntülenebilir.

GİRİŞLER		ÇIKIŞ
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Tablo-3.2 VE geçidinin doğruluk tablosu.

Doğruluk tablosu yapılırken bulunması gereken toplam kombinasyon sayısı şu formülle bulunabilir.

$$N=2^n$$

Bu formülde;

N =toplam kombinasyon sayısı.

n =giriş sayısı.

İki girişli eleman için $N=2^2=4$

Üç girişli eleman için $N=2^3=8$

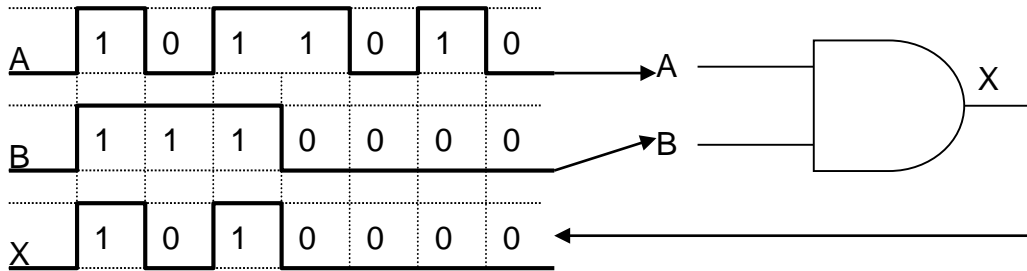
Dört girişli eleman için $N=2^4=16$

Şekil-3.5'te iki girişli VE geçidi girişlerine A ve B işaretleri uygulanmıştır. Çıkış işareti her zaman aralığında iki giriş işaretine bakılarak ve doğruluk tablosu kullanılarak elde edilir. t_1 zaman aralığında A ve B girişine gelen her iki işaret YÜKSEKtir, doğruluk tablosuna bakıldığında bu kombinasyon VE geçidinin girişlerine geldiğinde çıkışın YÜKSEK olduğu görülür. Çıkış işareti bu zaman aralığında YÜKSEK olarak çizilir. t_2 zaman aralığında A işareti DÜŞÜK B işareti YÜKSEKtir, doğruluk tablosu bu durum için çıkışın DÜŞÜK olduğunu göstermektedir, çıkış DÜŞÜK olarak çizilir. t_3 zaman aralığında A ve B işaretlerinin her ikisi YÜKSEKtir bu durumda çıkış YÜKSEK olacaktır. t_4 zaman aralığında A YÜKSEK, B DÜŞÜK seviyededir. Tablo bu durum için çıkışın DÜŞÜK seviyede olması gerektiğini göstermektedir. Bu zaman aralığı için çıkış DÜŞÜK

olarak çizilir. t_5 zaman aralığında A ve B işaretlerinin her ikisi DÜŞÜK seviyededir. Tabloda çıkış için DÜŞÜK seviye gösterildiğinden çıkış DÜŞÜK olarak çizilir.

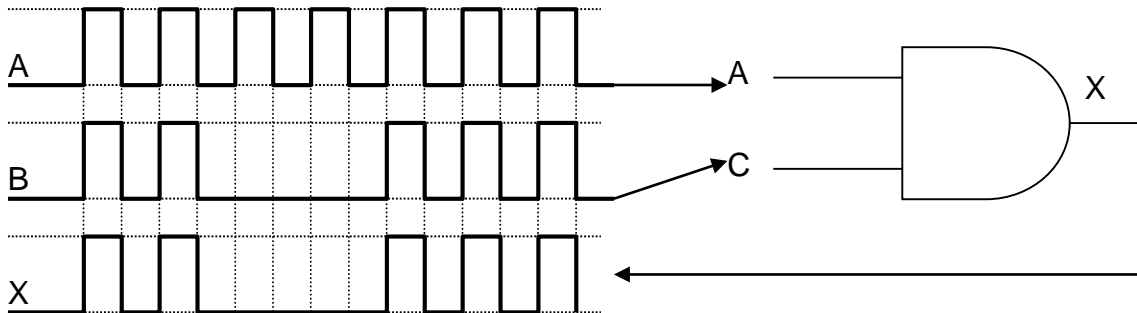
GİRİŞLER			ÇIKIŞ
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Tablo-3.3 Üç girişli VE geçidinin doğruluk tablosu.

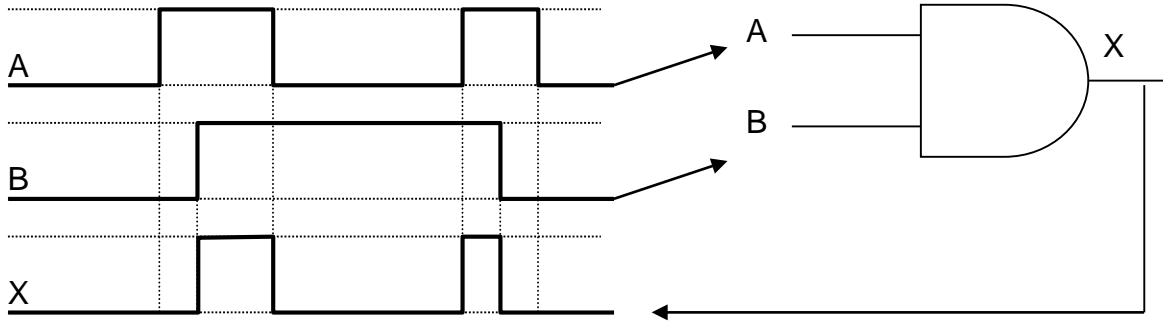


Şekil-3.5

Şekil-3.6'daki zamanlama diyagramında yine iki girişli bir VE geçidi girişine A olarak adlandırılan bir saat işareti, B olarak adlandırılan bir kontrol işareti uygulanmıştır. A saat işareti B kontrol işaretinin izin verdiği zaman aralıklarında çıkışa aktarılır, diğer zamanlarda çıkış DÜŞÜK seviyede kalır.



Şekil-3.6



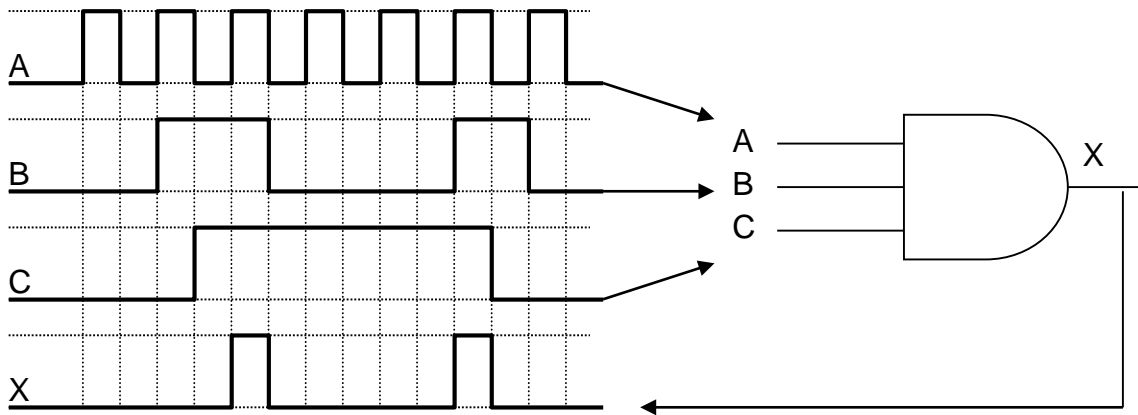
Şekil-3.7

Şekil-3.7'deki zamanlama diyagramında giriş işaretlerinin durum değiştirmeleri birbirleri ile uyumlu değildir. Bu tür durumla karşılaştığınızda işaretlerin durum değiştirme anları sizin zamanlama çizgilerini çizdiğiniz yerler olacaktır. Zaman dilimleri eşit olmayacaktır. Çıkıştan elde edilen işaret de periyodik olmayacaktır.

Şekil-3.8'de 3 girişli VE geçidinin zamanlama diyagramı görülmektedir. Diğerlerinden farkı giriş sayısının fazla olması ve buna bağlı olarak zaman çizgilerinin fazla olmasıdır.

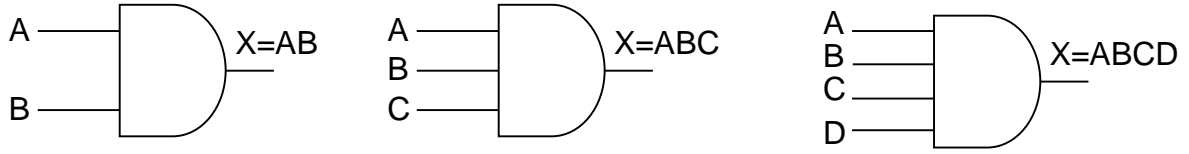
VE geçidinin mantık işlevi doğruluk tablosu incelendiğinde ikili çarpma ile aynı olduğu görülür. Kısaca **VE geçidi ikili çarpma** işlemi yapar diyebiliriz. Çıkış fonksiyonunu şöyle yazılabilir:

$X=A \bullet B$ nokta genellikle kullanılmaz, $X=AB$ olarak yazılır.



Şekil-3.8

Giriş sayısı artıktıkça çarpımdaki harf sayısı da artacaktır. Şekil-3.9'da iki, üç, dört girişli VE geçidinin çıkış fonksiyonları yazılmıştır.

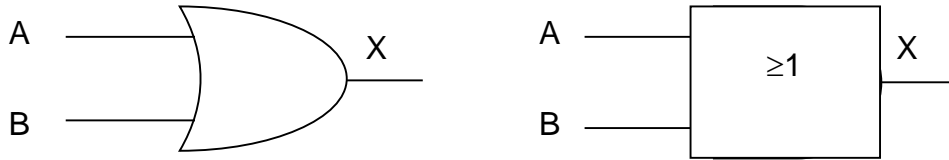


Şekil-3.9 VE geçidinde giriş sayısı artıktıkça çıkış fonksiyonundaki terim

VEYA (OR) İŞLEMİ

VEYA işlemi, yalnızca bütün girişleri DÜŞÜK olduğunda çıkışı DÜŞÜK olan bir işlemdir. Girişlerden en az biri YÜKSEK olursa çıkış YÜKSEK olur. Bu mantık işlemi, **VEYA geçidi** (*or gate*) denilen mantık devresiyle gerçekleştirilir. VEYA geçidinin kullanılan her iki sembolü Şekil-3.10'da verilmiştir. Her iki sembolü derste kullanabilirsiniz.

Şekil-3.11'de ise VEYA geçidinin girişlerine olası giriş seviyeleri uygulandığında oluşacak mantık seviyeleri gösterilmiştir. Bu geçidin çıkışının YÜKSEK olabilmesi için girişlerden birinin YÜKSEK olması yeterlidir. Çıkış ancak tüm girişler DÜŞÜK ise DÜŞÜK'tür. Tablo-3.4'te doğruluk tablosu verilmiştir.

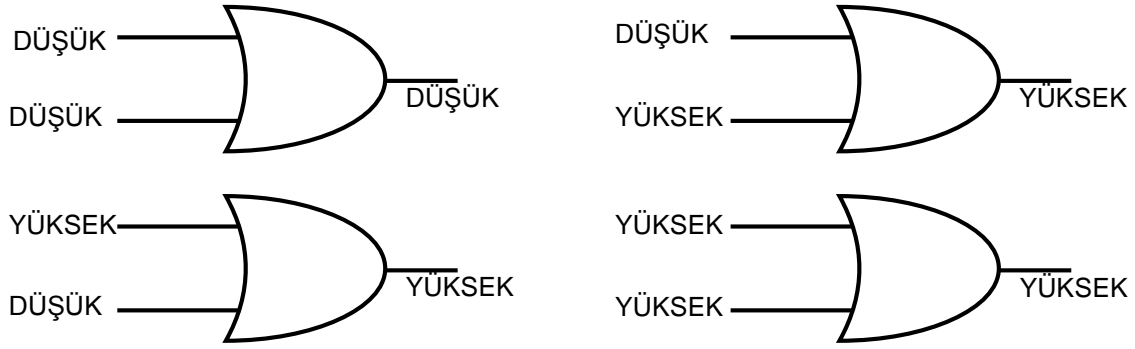


Şekil-3.10 VEYA geçidinin simgeleri.

İki VEYA geçidinin doğruluk tablosu Tablo-3.4'te verilmiştir. Giriş sayısı artıktıkça kombinasyon sayısı VE geçidinde olduğu gibi artacaktır.

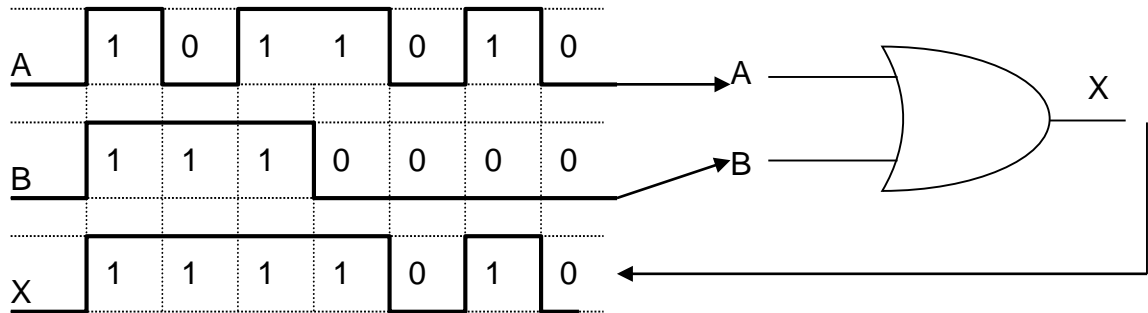
GİRİŞLER		ÇIKIŞ
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Tablo-3.4 VEYA geçidinin doğruluk tablosu.



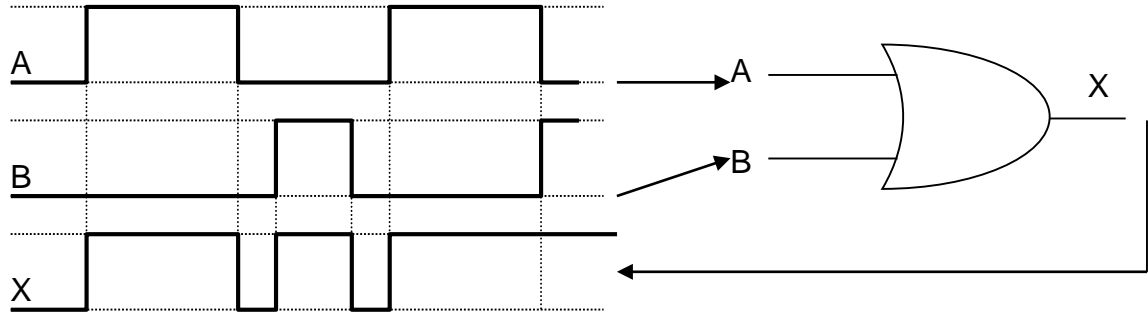
Şekil-3.11

Şekil-3.12’de iki girişli VE geçidi girişlerine A ve B işaretleri uygulanmıştır. Çıkış işareti her zaman aralığında iki giriş işaretine bakılarak ve doğruluk tablosu kullanılarak elde edilir. t_1 zaman aralığında A ve B girişine gelen her iki işaret YÜKSEKTİR, doğruluk tablosuna bakıldığında bu kombinasyon VEYA geçidinin girişlerine geldiğinde çıkışın YÜKSEK olduğu görülür. Çıkış işareti bu zaman aralığında YÜKSEK olarak çizilir. t_2 zaman aralığında A işareti DÜŞÜK B işareti YÜKSEKTİR, doğruluk tablosu bu durum için çıkışın YÜKSEK olduğunu göstermektedir, çıkış YÜKSEK olarak çizilir. t_3 zaman aralığında A ve B işaretlerinin her ikisi DÜŞÜKTÜR, bu durumda çıkış DÜŞÜK olacaktır. t_4 zaman aralığında A YÜKSEK, B DÜŞÜK seviyededir. Tablo bu durum için çıkışın YÜKSEK seviyede olması gerektiğini göstermektedir. Bu zaman aralığı için çıkış YÜKSEK olarak çizilir.

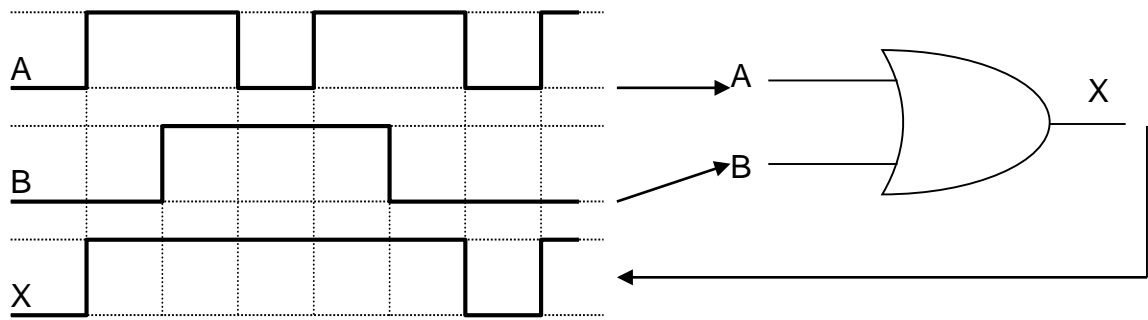


Şekil-3.12

VEYA geçidi girişine gelen işaretlerin YÜKSEK seviyelerini toplamak amacıyla kullanılır. Şekil-3.13’teki devre bu uygulamaya bir örnektir.

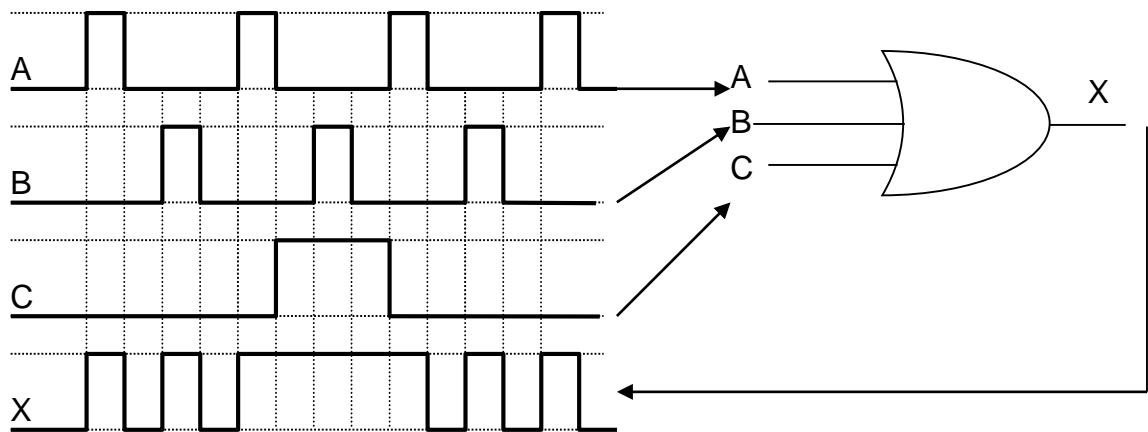


Şekil-3.13



Şekil-3.14

Şekil-3.14'te VEYA geçidinin girişlerine farklı işaretler uygulandığında elde edilen çıkış işaretine başka bir örnek verilmiştir. Şekil-3.15'te üç girişli VEYA geçidi girişlerine gösterilen A, B, C işaretleri uygulanmış ve çıkışından gösterilen işaret elde edilmiştir. Giriş sayısının artması zaman çizgilerinin sayısını arttırmak dışında çıkış işaretinin çizim kurallarını değiştirmemiştir.

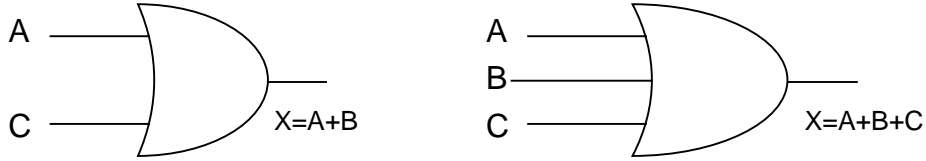


Şekil-3.15

VEYA geçidi boolean toplama işlemi yapar. İkili toplama boolean toplamadan farklıdır. Boolean toplamada $1+1=1$ 'dir, oysa ikili toplamada bu toplamdan elde olur ve toplam 10_2 olur. VEYA geçidinin çıkış fonksiyonunu şöyle yazabiliriz:

$$X=A+B$$

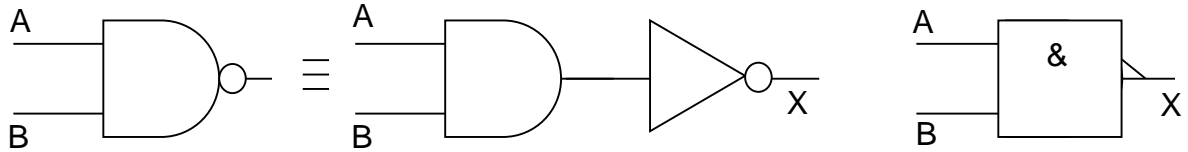
Giriş sayısı arttıkça toplama giriş sayısı kadar ekleme yapılacaktır. Şekil-3.16'da iki, üç, ve dört girişli VEYA geçitlerinin çıkış fonksiyonları verilmiştir.



Şekil-3.16

VED (NAND) GEÇİDİ

VED geçidi en yaygın olarak kullanılan bir geçittir, bunun nedeni üniversal geçit olmasıdır. Bu geçit ile diğer mantık işlemleri yapılabilir. Bu tür geçitlere üniversal geçit adı verilir. VED geçidi işlev olarak daha önce tanıdığınız VE ve DEĞİL geçitlerinin birleşimidir. Şekil-3.17'de eşdeğer devresi ile kullanılan iki farklı sembolü gösterilmiştir. VED geçidinin çıkışı sadece her iki giriş YÜKSEK olduğunda DÜŞÜK olur. Diğer durumlarda geçidin çıkışı YÜKSEK tir. Tablo-3.5'te VED geçidinin doğruluk tablosu verilmiştir. Dikkat edilirse VE geçidinin tam tersi bir işlem gerçekleştirir.



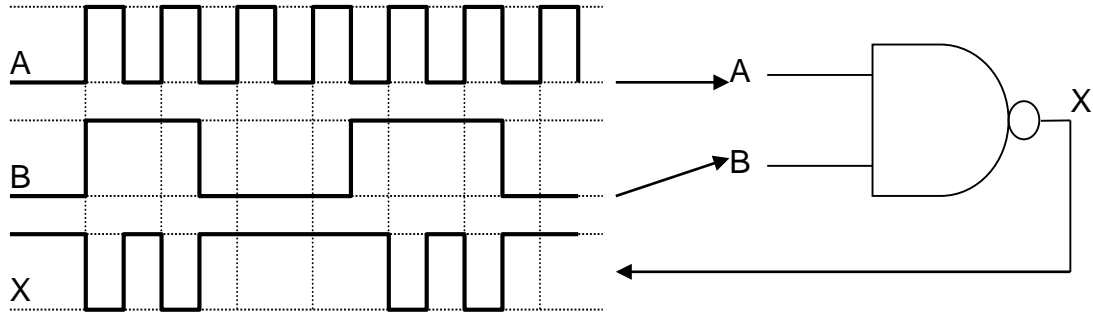
Şekil-3.17 VED geçidinin simgeleri.

Şekil-3.18'de A ve B gibi iki işaret VED geçidi girişlerine uygulandığında X çıkışı doğruluk tablosu kullanılarak çizilebilir. Dikkat edilmesi gereken nokta her iki giriş 1 ise çıkış 0, aksi durumlarda çıkış 1 olur.

GİRİŞLER		ÇIKIŞ
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

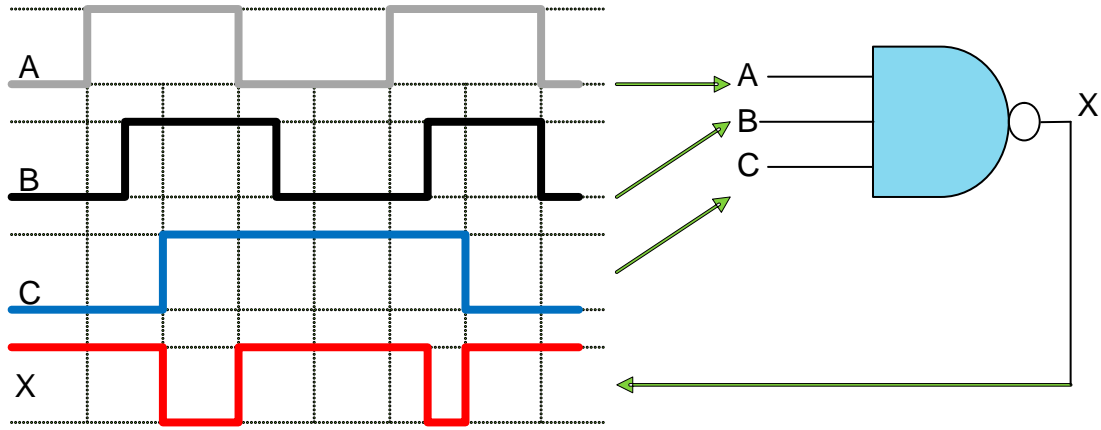
Tablo-3.5 VED geçidinin doğruluk tablosu

VED geçidi ikiden fazla girişe sahip olabilir çıkış için kural yine değişmez. Şekil-3.19'da bu duruma bir örnek verilmiştir.



Şekil-3.18

VED geçidi negatif VEYA geçidi olarak kullanılabilir. VEYA geçidinin özelliği girişlerden biri YÜKSEK ise çıkış YÜKSEKTİR. Her iki giriş DÜŞÜK ise çıkış DÜŞÜKTÜR. VED geçidinin çıkışındaki delik geçit üzerinden girişe aktarılır ise geçidin tipi değişir ve Şekil-3.20'deki görüntü elde edilir.



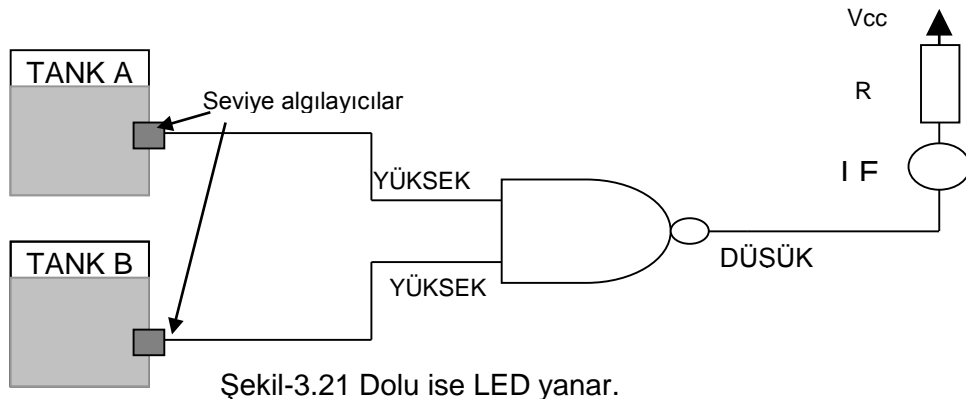
Şekil-3.19



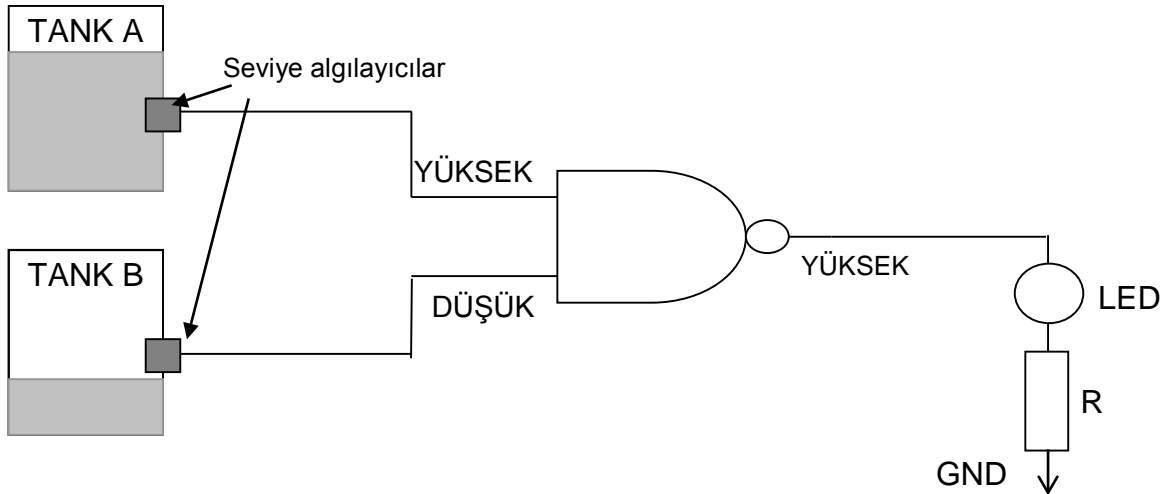
Şekil-3.20 VED geçidinin kullanılan iki farklı simgesi

VED GEÇİDİ UYGULAMALARI

Şekil-3.21'deki seviye algılayıcılar kullanılarak tank A ve tank B'deki sıvı seviyesinin %25'in altına düşüp düşmediği belirlenecektir. Her iki tankta veya herhangi birinde seviye istenen seviyenin altına düştüğünde LED sönecektir.

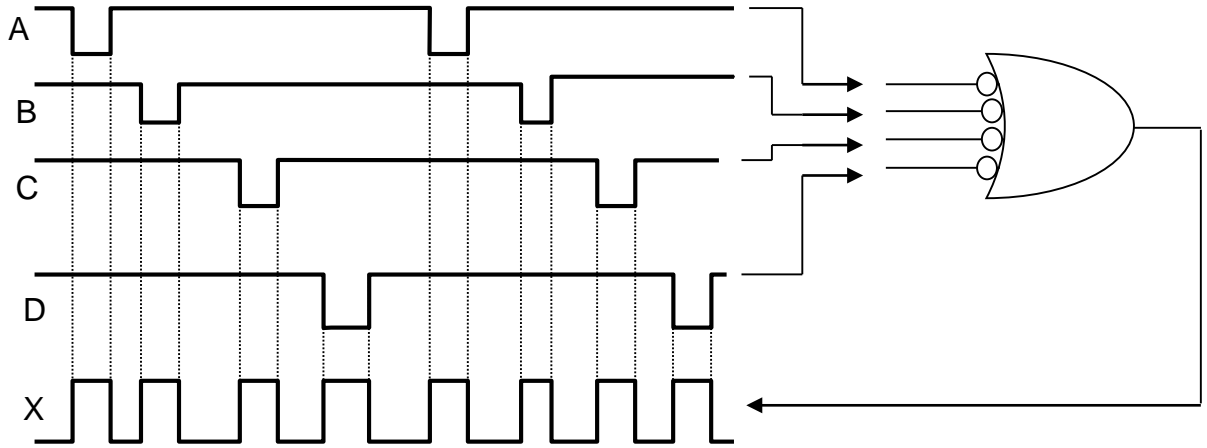


Aynı sistemde dolu durumda LED'in yanması yerine herhangi birinde %25'in altına düştüğünde uyarı LED'inin yanmasını istiyorsak devrenin Şekil-3.22'de olduğu gibi değiştirilmesi gerekir.



Şekil-3.23'te 4 girişli VED geçidi girişine verilen işaretler uygulanmıştır, çıkış işareti aynı şekilde gösterilmiştir. Giriş sayısı arttıkça çıkış dalga şeklini çizmek zorlaşır. Fakat VED geçidinin fonksiyonu uygulandığında iki girişli VED geçidi ile aynı kolaylıkla çıkış işareti çizilebilir.

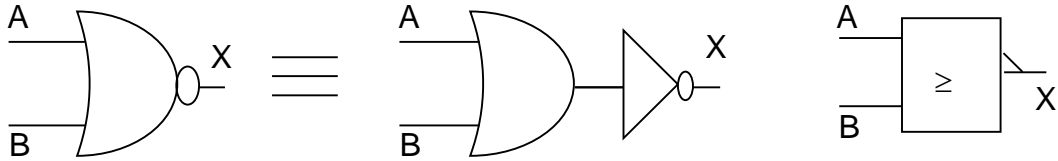
$$X = \overline{AB} = \overline{A} + \overline{B}$$



Şekil-3.23 4 girişli VED geçidine uygulanan giriş dalga şekilleri ve çıkış dalga şekli.

VEYAD (NOR) GEÇİDİ

VEYAD kelimesi VEYA ile DEĞİL kelimelerinin birleşimidir, adı gibi VEYAD geçidi VEYA geçidi ile DEĞİL geçidinin birleşimi ile elde edilir. VED geçidi gibi universal geçittir. Kullanılan sembolleri ve eşdeğeri şekil-3.24'te gösterilmiştir.



Şekil-3.24 VEYAD geçidinin simgeleri.

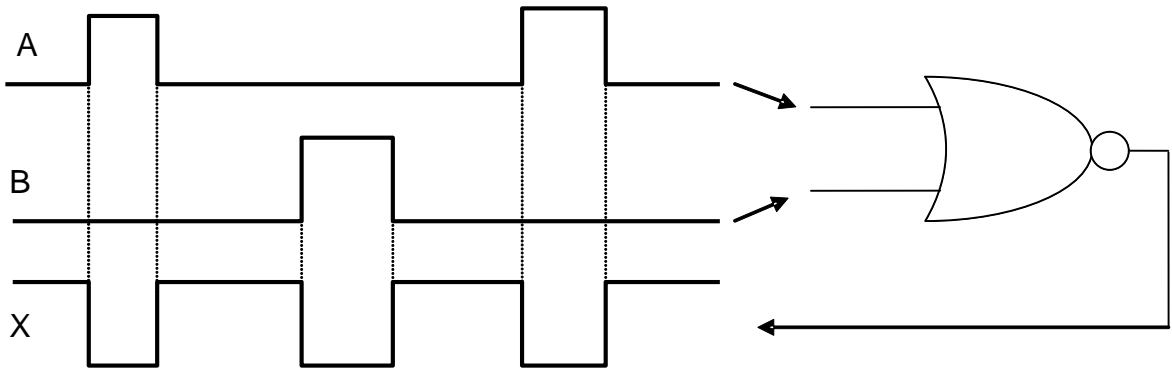
Tablo-3.6'da VEYAD geçidinin doğruluk tablosu verilmiştir. Bu geçidin çıkışının YÜKSEK olabilmesi için mutlaka her iki girişin DÜŞÜK olması gerekir. Diğer tüm giriş birleşimlerinde çıkış DÜŞÜKTÜR. Şekil-3.25'te iki girişli VEYAD geçidin girişlerine A ve B işaretleri uygulanmış ve X çıkışındaki işaret elde edilmiştir. X çıkışı çizilirken öncelikle giriş işaretlerinin değişim anlarından zaman çizgileri çizilir. Bu zaman çizgilerinde giriş işaretlerinin birleşimi doğruluk tablosundan bulunur ve çıkış işareti olarak çizilir.

VEYAD geçidi üç girişli veya daha fazla girişli olarak da üretilir. Giriş sayısının artması çıkış kuralını değiştirmez. Çıkış sadece tüm girişler DÜŞÜK olduğunda YÜKSEKtir, diğer durumlarda DÜŞÜKTÜR. Şekil-3.26'da üç girişli bir VEYAD geçidinin girişine uygulanan A, B, C işaretleri ve bu işaretlere bağlı olarak elde edilen X işareti gösterilmiştir.

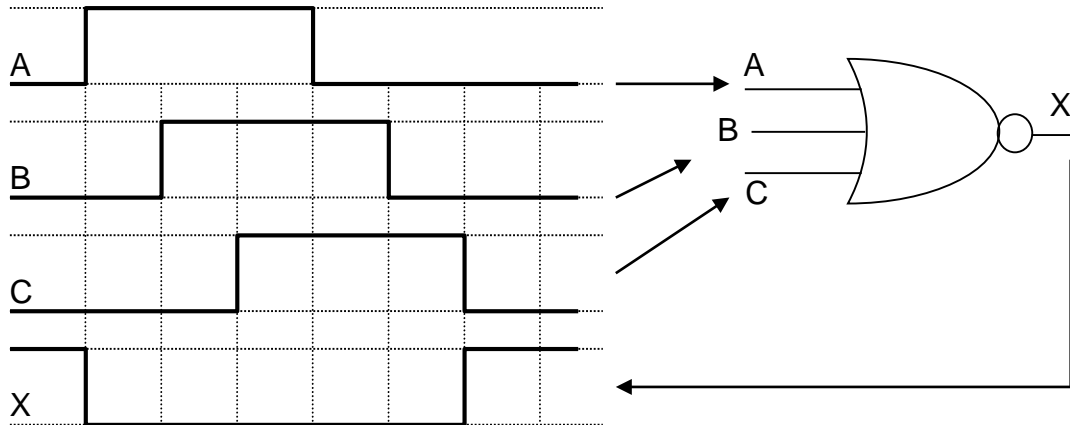
GİRİŞ		ÇIKIŞ
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Tablo-3.6 VEYAD geçidi doğruluk tablosu

VEYAD geçidi girişleri terslenmiş VE geçidi olarak kullanılabilir. Çıkıştaki delik geçit girişlerine alındığında geçidin şekli değişir. Şekil-3.27’de gösterilen şekil elde edilir. Bu sembole girişleri terslenmiş VE geçidi veya negatif VE geçidi adı verilir. Şekil-3.28’de dört girişli negatif VE geçidi girişine A, B, C, D işaretleri uygulanmıştır. Negatif VE geçidi VEYAD geçidi ile aynı özelliğe sahip olduğundan aynı kurallar kullanılarak çıkış işareti X çizilir.



Şekil-3.25 VEYAD geçidi zamanlama diyagramı.



Şekil-3.26



Şekil-3.27 VEYAD geçidinin simgeleri.

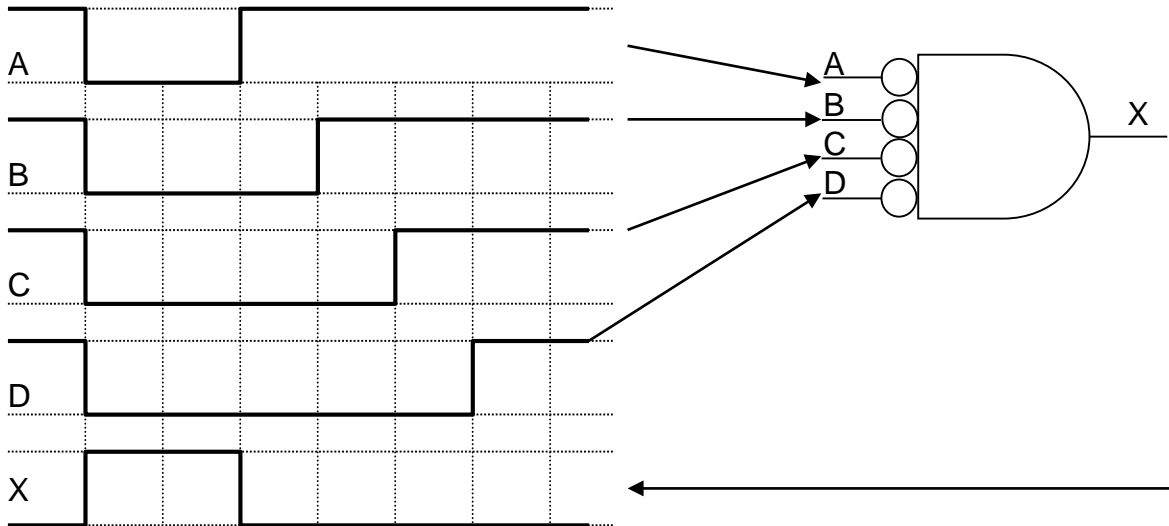
VEYAD geçidinin çıkış fonksiyonunu yazalım;

$$X = \overline{A + B} = \overline{A} \bullet \overline{B}$$

Giriş sayısı arttıkça çarpım uzayacaktır.

$$X = \overline{A + B + C} = \overline{A} \bullet \overline{B} \bullet \overline{C}$$

$$X = \overline{A + B + C + D} = \overline{A} \bullet \overline{B} \bullet \overline{C} \bullet \overline{D}$$

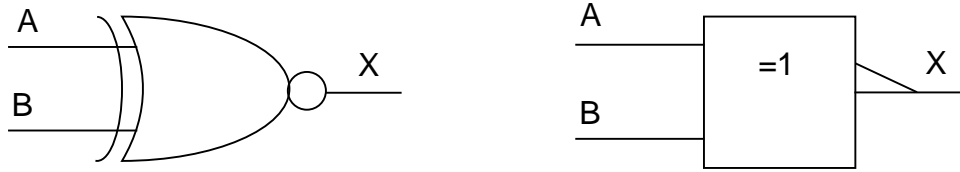


Şekil-3.28

ÖZEL VEYA GEÇİDİ (EXCLUSIVE-OR)

Bu geçit aslında diğer geçitlerin kombinasyonundan oluşur. Bir çok uygulamada kullanılması bu geçidin bağımsız bir sembolü olmasına neden olmuştur. Kullanılan standart sembolleri Şekil-3.29'da gösterilmiştir. Bu geçit sadece iki girişli olarak üretilir, birden fazla girişli elde etmek için diğer geçitler kullanılarak yapılabilir.

Şekil-3.31’de kullanılan standart sembolleri ve girişine gelebilecek olası giriş seviyeleri ve buna karşılık gelen çıkış seviyeleri gösterilmiştir.



Şekil-3.31 (EXNOR) ÖZEL VEYA DEĞİL geçidinin sembolleri.

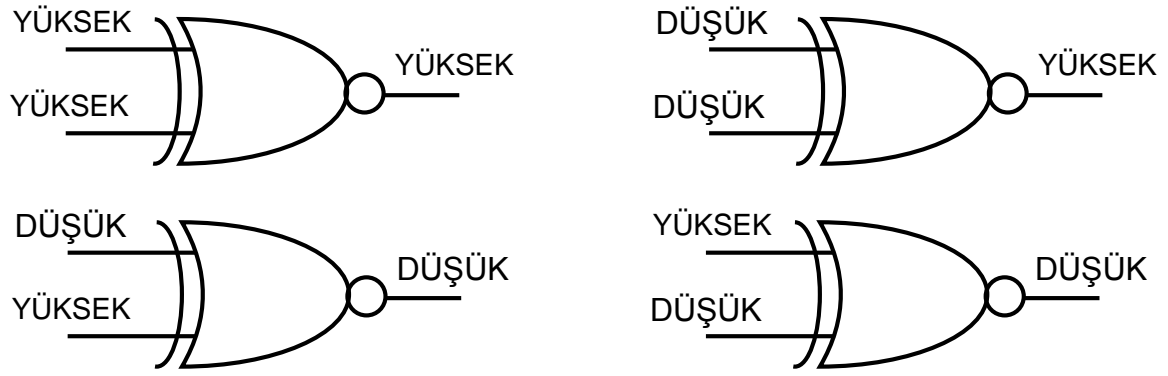
ÖZEL VEYA DEĞİL geçidi geçidinin çıkışının YÜKSEK olabilmesi için girişine gelen işaretlerin seviyelerinin aynı olması gerekir. Diğer durumlarda çıkış seviyesi DÜŞÜK olacaktır. ÖZEL VEYA DEĞİL geçidinin doğruluk tablosu Tablo-3.8’de gösterilmiştir. Girişleri aynı olduğunda YÜKSEK çıkış verdiği için EŞ geçit olarak ta adlandırılır.

GİRİŞ		ÇIKIŞ
A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

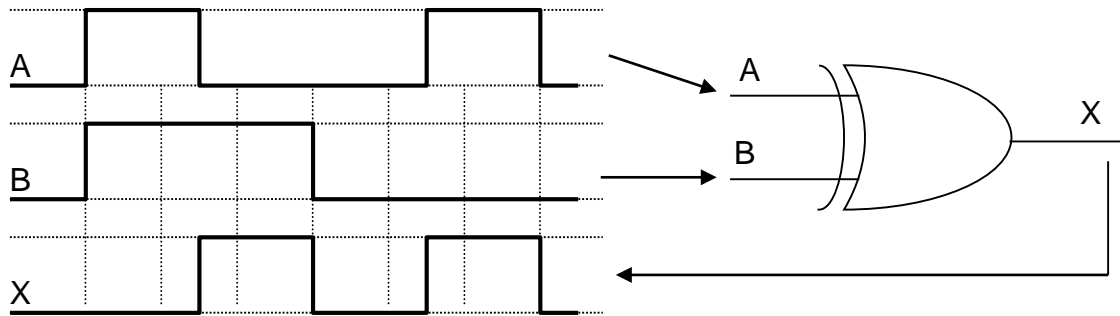
Tablo-3.8 ÖZEL VEYA DEĞİL geçidinin doğruluk tablosu

Şekil-3.33’te EXOR geçidinin girişlerine uygulanan işaretler ve bu işaretlere göre geçit çıkışı çizilmiştir. Çıkış çizilirken yine diğer geçitlerde olduğu gibi önce zaman çizgileri çizilir ve daha sonra doğruluk tablosu kullanılarak çıkış seviyeleri her bir zaman aralığı için çizilir.

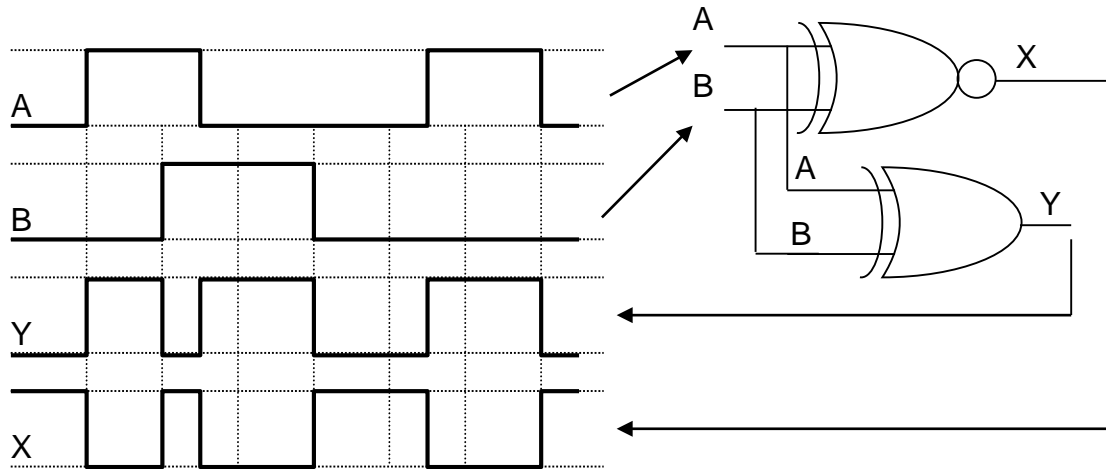
Şekil-3.34’te ise XOR ve XNOR geçitlerinin girişlerine aynı işaretler uygulanmış ve her ikisi için çıkış işaretleri çizilmiştir. Çıkış işaretlerinin birbirinin tersi olduğuna dikkat edin. XOR geçidinin diğer bir kullanım alanı ise iki ikili bitin toplanmasıdır. Şekil-3.35’te iki adet ikilinin toplanmasına örnek gösterilmiştir. Oluşan elde başka bir devre tarafından belirlenmelidir. XOR geçidi sadece toplama işlemini yerine getirecektir.



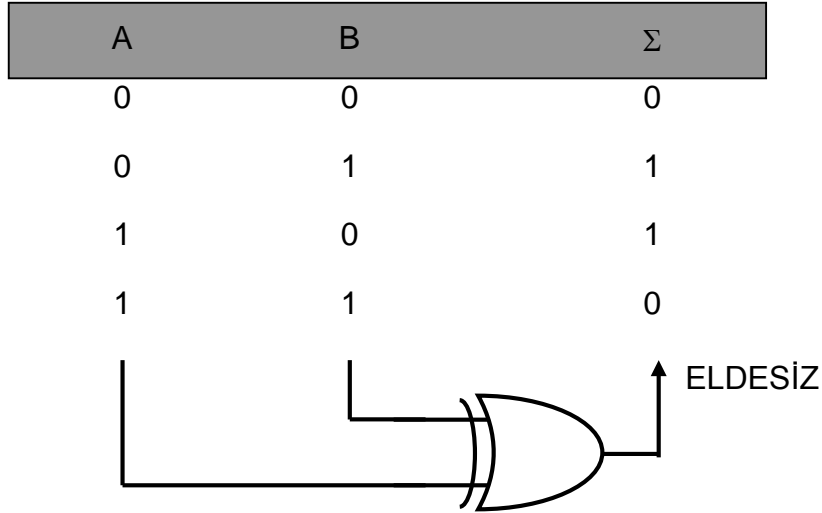
Şekil-3.32 EXNOR geçidinin girişine gelen seviyelere göre oluşan çıkış seviyeleri.



Şekil-3.33



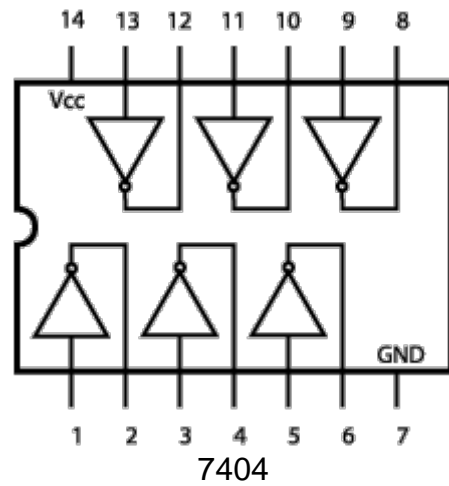
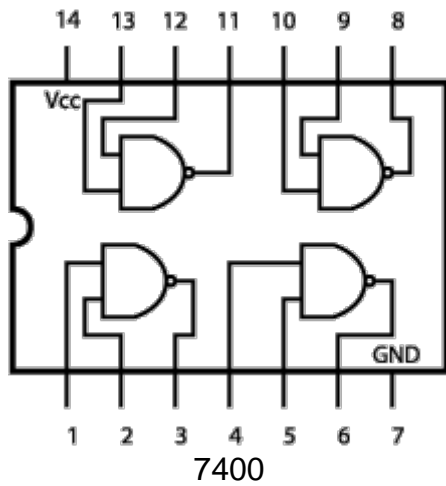
Şekil-3.34

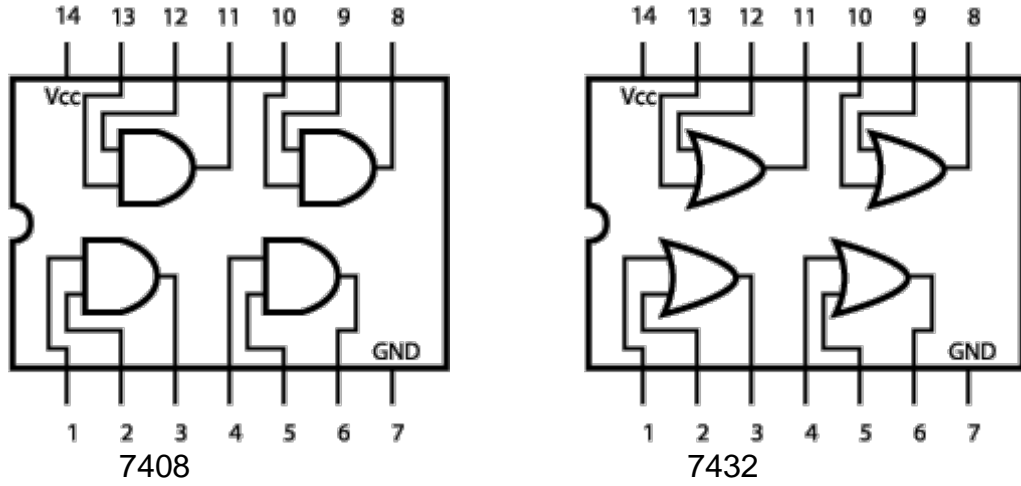


Şekil-3.35 EXOR geçidi ikilik eldesiz toplayıcı olarak kullanılabilir.

TÜMDEVRE LOJİK AİLELERİ

Geçitler tümdevre olarak üretilirler ve tüm devre içerisine birden fazla geçit yerleştirilebilir. Üretim teknolojisi geçitlerin işlevlerini değiştirmez fakat güç tüketimi, giriş ile çıkış arasındaki zaman gecikmesi gibi özellikleri değişir. Yaygın olarak kullanılan iki tür üretim teknolojisi vardır. Bunlardan birincisi TTL (bipolar transistors) ailesi, diğeri ise CMOS ailesidir. TTL ailesi 0-5 Volt ile beslenir, hatta son yıllarda 3 Voltluk olan tümdevreler üretilmektedir. CMOS ailesi ise 6 V ile 18 V arası değerlerdeki besleme gerilimi ile beslenebilir.

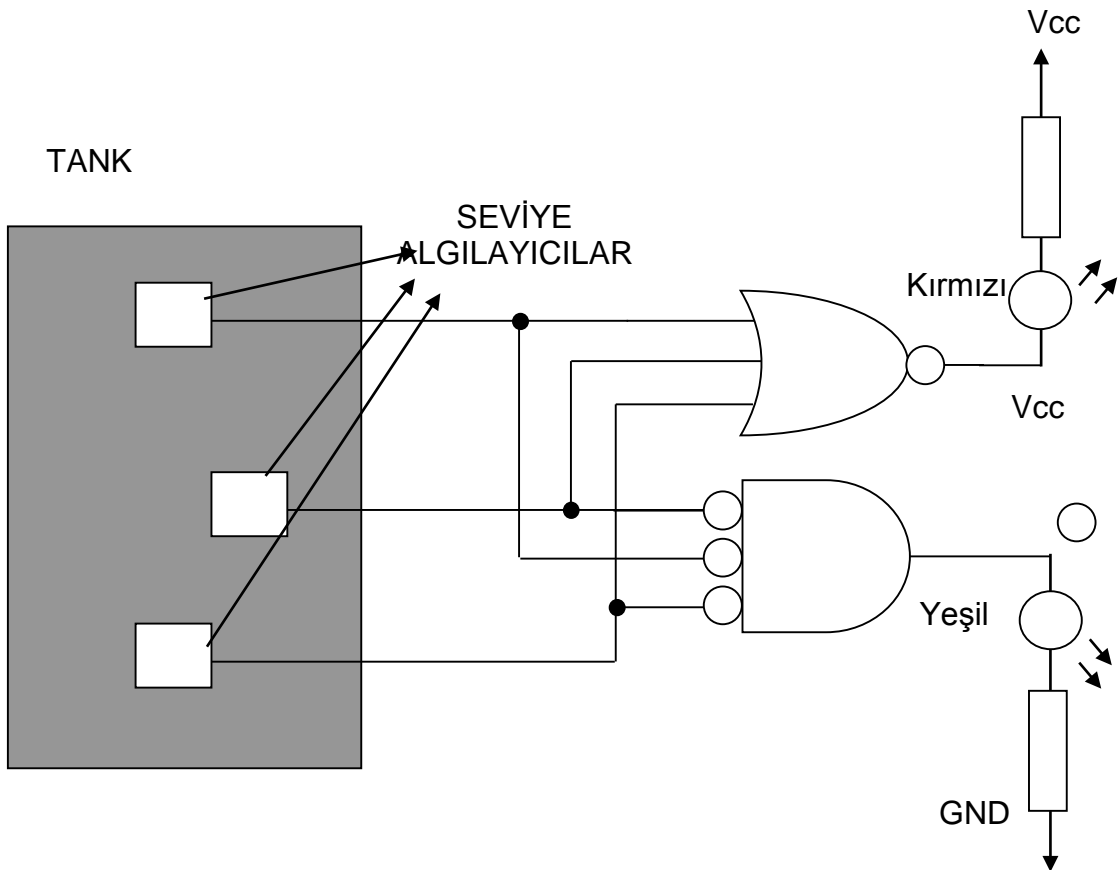




Şekil-3.36 En çok kullanılan mantık tümdevreleri.

PROBLEMLER

1. Şekil-3.36'daki devrede kırmızı LED'in yanması için gerekli giriş birleşimini belirleyin.



Şekil-3.36 VEYAD geçidi uygulaması.

2. Şekil-3.36'daki devrede yeşil LED'in yanması için gerekli giriş birleşimini belirleyin.



BOOLEAN KANUNLARI VE DEMORGAN TEOREMİ

GİRİŞ

Boolean aritmetiği ikilik sayı sisteminin matematiğidir. Matematikte gördüğünüz işlemlerin tamamı boolean aritmetiğinde de vardır. Fakat değişkenlerin alabileceği sadece iki değer vardır. Değişken olarak sadece “1” ve “0” yada doğru ve yanlış vardır. Boolean aritmetiğinin 12 adet kanunu vardır, genellikle harfler kullanılarak bu kanunlar ifade edilir.

Boolean aritmetiğinde kullanılan sembollerin bazıları matematikteki ile aynı olmasına rağmen farklı anlam taşıyabilir. Harfin üzeri çizili ise bunun anlamı DEĞİL demektir, bazı kaynaklarda üst çizgi yerine ' kesme işareti kullanılır ($A' = \overline{A}$). + işareti VEYA işlemini, • işareti VE işlemini gösterir. Nokta işareti genellikle kullanılmaz, $A \bullet B$ yerine AB kullanılır.

BOOLEAN TOPLAMA

Üçüncü konuda gördüğünüz gibi boolean toplama VEYA işlemine eşittir. Basit olarak toplanmanın kuralını şöyle özetleyebiliriz.

$$0+0=0$$

$$0+1=1$$

$$1+0=1$$

$$1+1=1$$

Boolean aritmetiğinde toplama toplama dahil olan literallerin toplamıdır. Sadece VEYA işlemidir.

$$A+B$$

$$A+B'$$

$$A+B+C'$$

$$A'+B+C+D'$$

Toplam giriş literallerinden en az biri 1 olduğunda 1, aksi halde 0'dır.

BOOLEAN ÇARPMA

Daha önce tanımladığımız gibi boolean çarpma VE işlemine eşdeğerdir. Kurallarını şöyle özetleyebiliriz.

$$0.0=0 \quad 0.1=0 \quad 1.0=0 \quad 1.1=1$$

Boolean aritmetiğinde çarpım giriş literallerinin çarpımıdır.

$$A.B \quad A.B' \quad A.B.C \quad AB'CD'$$

Çarpım literallerden biri sıfır olduğunda 0, tümü 1 olduğunda 1'dir.

BOOLEAN ARİTMETİĞİNİN KANUNLARI VE KURALLARI

Matematikte işlemler için çok iyi kurallar ve kanunlar geliştirilmiştir. Bu kural ve kanunlar boolean aritmetiğine uygulanabilir.

BOOLEAN ARİTMETİĞİNİN KANUNLARI

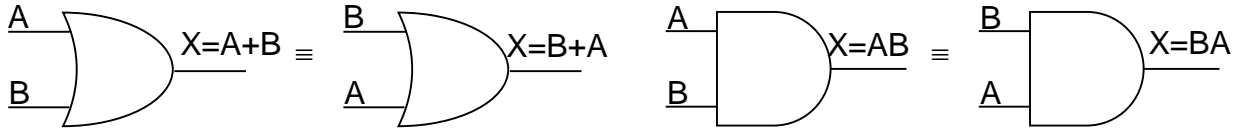
Boolean aritmetiğinin değişme, birleşme ve dağılma özellikleri vardır.

Değişme özelliği

VEYA işleminde veya VE işleminde girişlerin sırası önemli değildir.

$$A+B=B+A \quad \text{veya} \quad AB=BA$$

Şekil-4.1'de geçitler üzerinde değişme özelliği gösterilmiştir.



Şekil-4.1 Değişme özelliği.

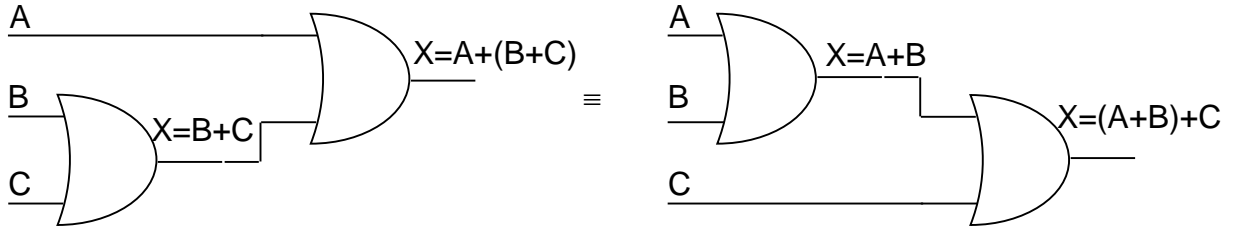
Birleşme özelliği

$$A+(B+C)=(A+B)+C$$

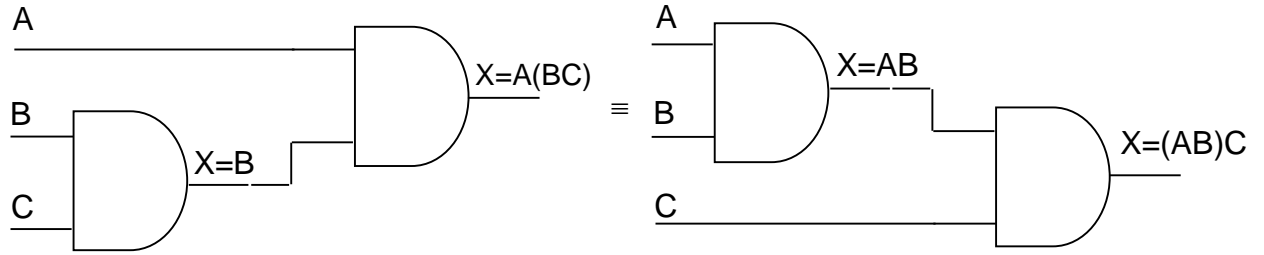
Bu özellik ikiden fazla değişkenin VEYAlanmasının sırasının önemli olmadığını gösterir. Aynı özellik VElemede de geçerlidir.

$$A(BC)=(AB)C$$

Şekil-4.2'de toplamının birleşme özelliği geçitlere uygulanmıştır. Şekil-4.3'de ise çarpmanın birleşme özelliği geçitlere uygulanmıştır.



Şekil-4.2 Toplamada Birleşme özelliği.



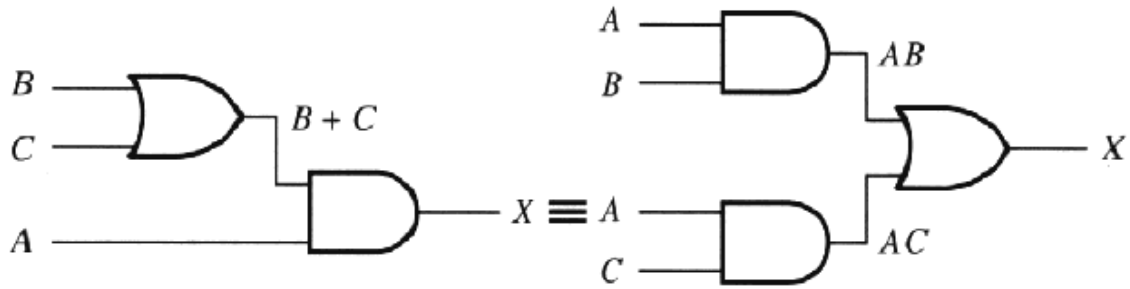
Şekil-4.3 Çarpmada birleşme özelliği

Dağılma özelliği

Üç değişkenli yazılmış hali aşağıdadır.

$$A(B+C) = AB + AC$$

Bu kanun birçok değişkeni VEYAladıktan sonra tek bir değişkenle VElemek ile bir değişkeni tüm diğer değişkenlerle ayrı ayrı VEledikten sonra VEYAlamak eşdeğer işlemlerdir. Şekil-4.4'te dağılma özelliğinin geçitlere uygulanması gösterilmiştir.



Şekil-4.4 Dağılma özelliğinin uygulaması.

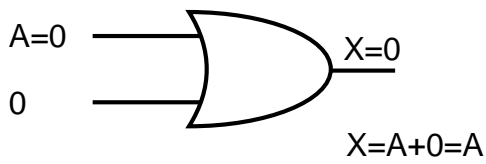
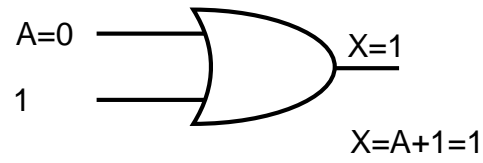
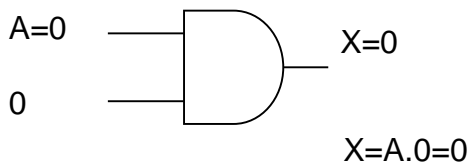
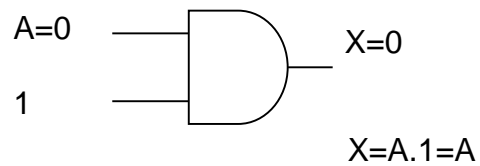
BOOLEAN KURALLARI

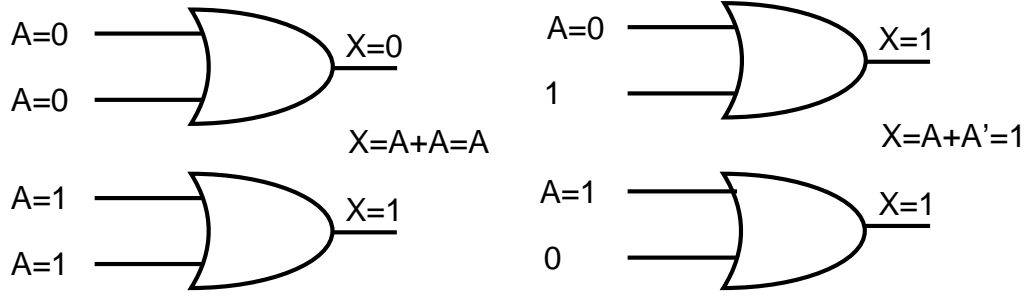
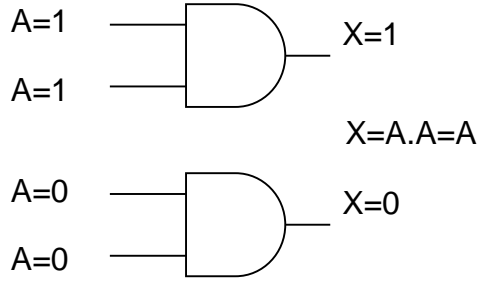
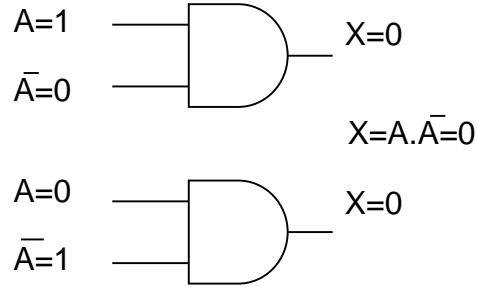
Boolean kuralları boolean eşitliklerini sadeleştirmek için kullanılır. 12 adet olan kuralların özeti Tablo-4.1'de gösterilmiştir.

1.	$A+0=A$
2.	$A+1=1$
3.	$A \cdot 0=0$
4.	$A \cdot 1=A$
5.	$A+A=A$
6.	$A+A'=1$
7.	$A \cdot A=A$
8.	$A \cdot A'=0$
9.	$\overline{\overline{A}} = A$
10.	$A+AB=A$
11.	$A + \overline{A}B = A + B$
12.	$(A+B)(A+C)=A+BC$

Tablo-4.1 Boolean kuralları.

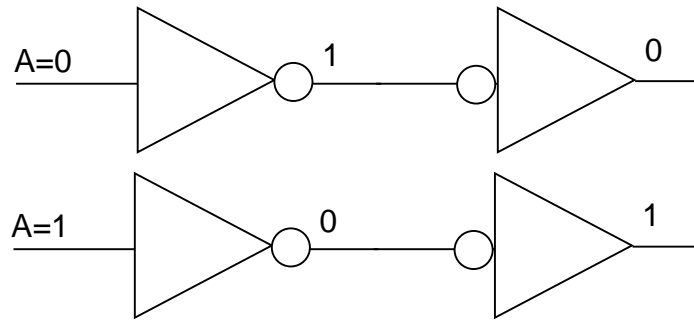
Şekil-4.5'de 1-8 numaralı boolean kurallarının geçitlere uygulanması gösterilmiştir.

KURAL 1 $A+0=A$ **KURAL 2 $A+1=1$** **KURAL 3 $A \cdot 0=0$** **KURAL 4 $A \cdot 1=A$** **KURAL 5 $A+A=A$** **KURAL 6 $A+A'=1$**

**KURAL 7 $A.A=A$** **KURAL 8 $A.A'=0$** 

Şekil-4.5 1-8 nolu kuralların geçitlere uygulanması.

Şekil-4.6'da 9. kuralın geçide uygulanması gösterilmiştir. Bu kurala göre değişkenin değilinin değil kendisidir. Bir değişken iki kez DEĞİL geçidi geçidinden geçirildiğinde kendisi elde edilir. Bu işlem size çok gereksiz gibi gelebilir fakat yaygın şekilde kullanılır. Özellikle dijital işareti güçlendirmek için kullanılır. Dijital işaret gerilim olarak yükseltilmez, sadece akım olarak yükseltilir.



Şekil-4.6 KURAL 9'un geçitler ile gösterimi

KURAL 10: $A+AB=A$ Bu kural dağılma özelliği kullanılarak açıklanabilir.

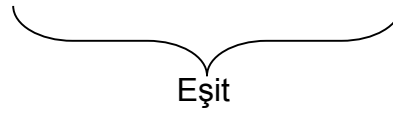
$$A+AB = A(1+B) \quad \text{Dağılma özelliği.}$$

$$= A.1 \quad \text{Kural 2}$$

$$= A \quad \text{Kural 4}$$

Tablo ile doğrulayalım.

A	B	AB	A+AB
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1



Eşit

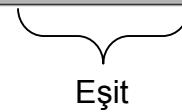
KURAL 11:

Diğer kuralları kullanarak doğruluğunu gösterelim;

$$\begin{aligned}
 A + \overline{AB} &= A + B \\
 A + \overline{AB} &= (A + AB) + \overline{AB} && \text{Kural 10} \\
 &= (AA + AB) + \overline{AB} && \text{Kural 7,} \\
 &= AA + AB + \overline{AA} + \overline{AB} && \text{Kural 8} \\
 &= (A + \overline{A})(A + B) && \text{Paranteze alın} \\
 &= 1.(A + B) && \text{Kural 6} \\
 &= A + B && \text{Kural 4}
 \end{aligned}$$

Doğruluk tablosu ile de gösterelim;

A	B	\overline{AB}	$\overline{A+AB}$	$\overline{A+B}$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1



Eşit

KURAL 12 : $(A + B)(A + C) = A + BC$

$$\begin{aligned}
 (A + B)(A + C) &= AA + AC + AB + BC && \text{Dağılma özelliği} \\
 &= A + AC + AB + BC && \text{Kural 7} \\
 &= A(1 + C) + AB + BC && \text{Dağılma özelliği} \\
 &= A.1 + AB + BC && \text{Kural 2} \\
 &= A(1 + B) + BC && \text{Dağılma özelliği} \\
 &= A.1 + BC && \text{Kural 2} \\
 &= A + BC && \text{Kural 4}
 \end{aligned}$$

Doğruluk tablosu ile de gösterelim;

A	B	C	A+B	A+C	(A+B)(A+C)	BC	A+BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

DE MORGAN TEOREMİ

Şöyle açıklanabilir;

$$\overline{XY} = \overline{X} + \overline{Y} \quad (1)$$

$$\overline{X + Y} = \overline{X} \bullet \overline{Y} \quad (2)$$

1 nolu eşitliği özetleyecek olursak; çarpımın değil de değişkenlerin değillerinin toplamına eşittir. 2 nolu eşitliği özetlersek; toplamın değil de değişkenlerin değillerinin çarpımına eşittir.

De Morgan Teoremi ikiden fazla değişkenler için de geçerlidir.

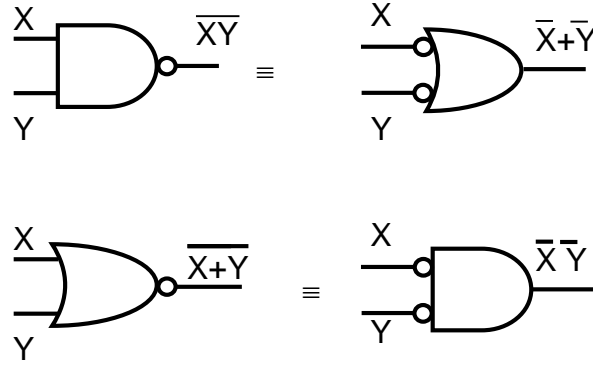
$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$

$$\overline{WXYZ} = \overline{W} + \overline{X} + \overline{Y} + \overline{Z}$$

$$\overline{X + Y + Z} = \overline{X} \overline{Y} \overline{Z}$$

$$\overline{W + Y + X + Z} = \overline{W} \overline{Y} \overline{X} \overline{Z}$$

De Morgan teoremlerinin geçitlere uygulanışı şekil-4.7'de gösterilmiştir.



Şekil-4.7 De Morgan teoreminin geçitler üzerinde uygulanması.

ÇARPIMLARIN TOPLAMI

Boolean aritmetiğinde çarpımı VE işleminin yaptığını daha önce belirtmiştik. İki veya daha fazla değişkenin çarpımı VE geçidi ile gerçekleştirilebilir. İki değişkenin çarpımı AB, üç değişkenin çarpımı ABC ve dört değişkenin çarpımı ABCD şeklinde açıklanabilir. Toplama işlemi ise VEYA geçitleri ile gerçekleştirilir. Çarpımların toplamı şeklindeki eşitlikler ise önce VE geçidi kullanılarak çarpma yapılır ve sonra birden fazla çarpım VEYA geçidi kullanılarak toplanır. Örnek olarak aşağıdaki eşitlikleri verebiliriz;

$$AB + BCD$$

$$ABC + DEC$$

$$\overline{A}BC + D\overline{E}FG + AEG$$

$$ABC + \overline{A}BC + ABC$$

ÇT ifadelerde değişkenlerin üzerindeki değil çizgileri birleşik çizilemez. Örneğin aşağıdaki ifadeler birbirine eşit değildir.

$$\overline{A} \overline{B} \overline{C} \neq \overline{ABC}$$

Tüm eşitlikler ÇT olarak ifade edilebilir. Boolean kanunları kullanılarak ÇT olmayan eşitlikler ÇT haline dönüştürülebilirler.

ÖRNEK:

$$A(AB + CD) = AB + ACD$$

TOPLAMLARIN ÇARPIMI (TÇ)

Toplamların çarpımı şeklindeki ifadeler birden fazla VEYA işleminin sonucunu VE işlemine tabi tutulmuş halidir.

Örnek:

$$\begin{aligned} &(A + B)(B + C + D) \\ &(A + B + C)(D + E + F) \\ &(A + \bar{B} + C)(D + C + \bar{F} + G)(A + \bar{F} + \bar{G}) \end{aligned}$$

TÇ ifadelerin bazılarında çarpımlardan biri veya birkaçı tek değişken olabilir.

$$A(B + C + D)(E + F + G)$$

Boolean Kanunlarının Kullanılarak Mantık Fonksiyonlarının Basitleştirilmesi:

Boolean kanunları genellikle karmaşık mantık fonksiyonlarının basitleştirilerek daha az sayıda geçitle veya daha az girişli geçit kullanarak gerçeklemek için kullanılır. Bu işlem için boolean kanunlarının iyi bilinmesi gerekir.

ÖRNEK:

$AB + A(B + C) + B(B + C)$ fonksiyonunu boolean kanunlarını kullanarak en basit hale indirgeyin.

ÇÖZÜM:

Adım 1. Dağılma kanunu 2. ve 3. terimlere uygulanırsa;

$$AB + AB + AC + BB + BC$$

Adım 2. 7 nolu kanun uygulanırsa ($BB=B$)

$$AB + AB + AC + B + BC$$

Adım 3. 3 nolu kanun uygulanırsa ($AB+AB=AB$)

$$AB + AC + B + BC$$

Adım 4. B çarpan parantezine alınırsa.

$$AB + AC + B(1 + C)$$

Adım 5. 2 nolu kanun uygulanırsa.

$$AB + AC + B \bullet 1$$

Adım 6. 4 nolu kanun uygulanırsa.

$$AB + AC + B$$

Adım 7. Birinci ve üçüncü terim B ortak parantezine alınır.

$$AC + B(A + I)$$

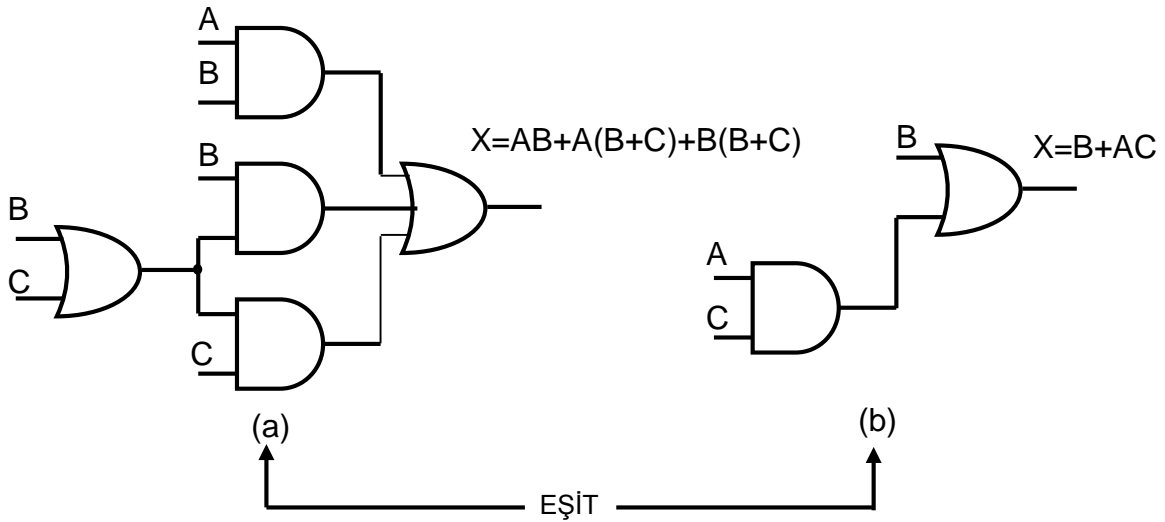
Adım 8. 2 nolu kanun uygulanırsa.

$$AC + B \cdot I$$

Adım 9. 4 nolu kanun uygulanırsa en sade hali elde edilir.

$$AC + B$$

Karmaşık ve sade hallerinin devrelerini birlikte çizersek aradaki fark daha iyi anlaşılabilir. Şekil-4.8'de sade ve karmaşık halleri ayrı ayrı iki devre olarak çizilmiştir. Karmaşık olanda 5 geçit sade olanda ise 2 geçit kullanılmıştır.



Şekil-4.8 Sade ve karmaşık haliyle fonksiyonun devresi.

BİRLEŞİK MANTIK DEVRELERİNİN TASARIMI

Birleşik mantık devresinin çıkışı giriş değişkenleri tarafından belirlenir. Bu tür devrelerin tasarımında öncelikle doğruluk tablosu elde edilir. Doğruluk tablosundan elde edilen ÇT ifade Karnaugh haritasına yerleştirilerek en sade ifade yazılır. Sadeleştirmenin amacı devreyi daha az sayıda geçit kullanarak gerçeklemektir. Karnaugh haritası 4 ve daha az değişken olduğunda kullanışlıdır. Daha fazla değişkene sahip olduğunda Boolean kanunları kullanılarak ifadeler sadeleştirilebilir.

3-DEĞİŞKENLİ KARNAUGH HARİTASININ KULLANIMI

3-değişkenli Karnaugh haritası Şekil-4.9'da gösterilmiştir. (a)'da değişkenler sıralanmış ve ikilik karşılıkları yazılmıştır. Değişkenlere isim olarak A, B, C kullanılmıştır ki bu isimler isteğe göre değiştirilebilir. (b)'de ise her hücrenin adresi yazılmıştır. Bir devrenin doğruluk tablosu elde edildikten sonra kolay bir şekilde devrenin değerleri bu adreslerden yola çıkarak Karnaugh haritasına yerleştirilebilir.

4-DEĞİŞKENLİ KARNAUGH HARİTASI

4-değişkenli Karnaugh haritası Şekil-4.10'da gösterilmiştir. (a)'da değişkenler sıralanmış ve ikilik karşılıkları yazılmıştır. Değişkenlere isim olarak A, B, C, D kullanılmıştır bu isimler isteğe göre değiştirilebilir. (b)'de ise her hücrenin adresi yazılmıştır. Bir devrenin doğruluk tablosu elde edildikten sonra kolay bir şekilde devrenin değerleri bu adreslerden yola çıkarak Karnaugh haritasına yerleştirilebilir.

		C	
		0	1
AB	00		
	01		
	11		
	10		

(a)

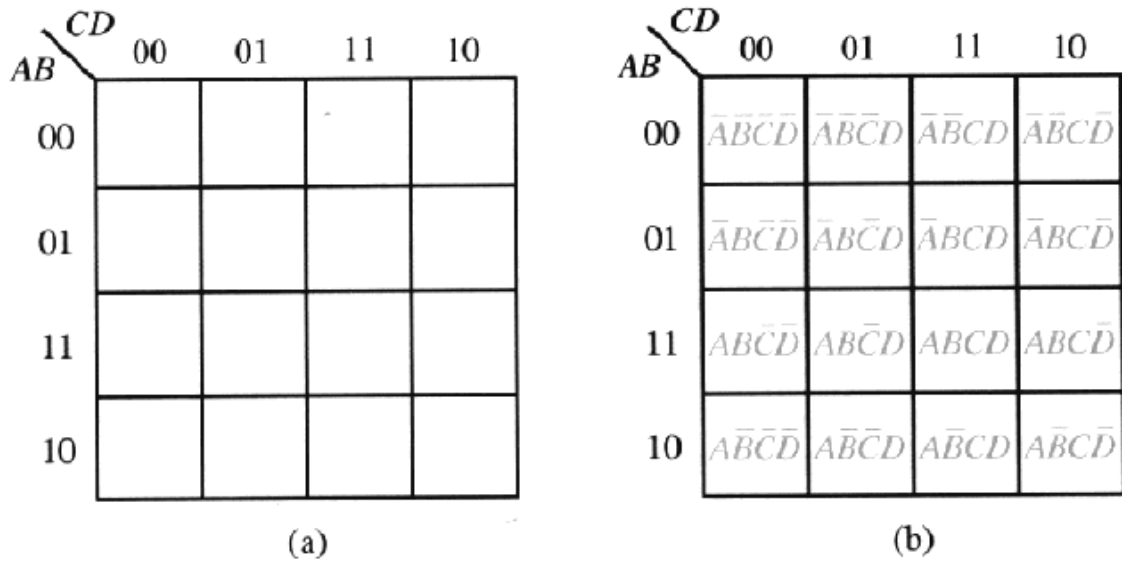
		C	
		0	1
AB	00	$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$
	01	$\bar{A}B\bar{C}$	$\bar{A}BC$
	11	$AB\bar{C}$	ABC
	10	$A\bar{B}\bar{C}$	$A\bar{B}C$

(b)

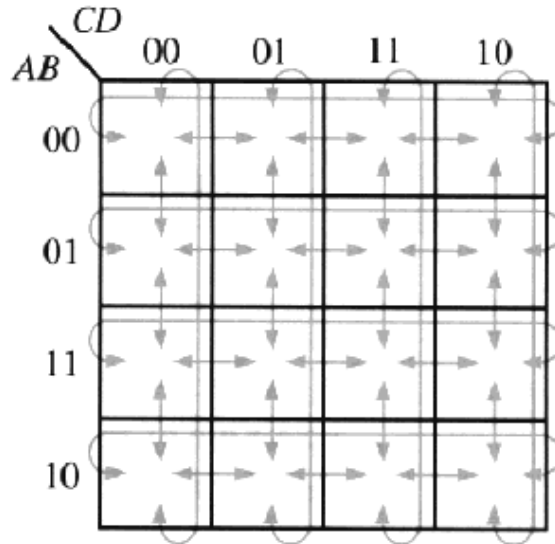
Şekil-4.9 3-değişkenli Karnaugh haritası ve adresleri.

Hücre komşulukları

Karnaugh haritasında hücrelerin komşu olabilmeleri için sadece bir değişkeni farklı olmalıdır. Yerleşim planı tablodaki sıra ile DEĞİL geçidi bu kurala uyacak şekilde düzenlenmiştir. Fiziki olarak her hücre diğerine komşu olarak görünse bile çapraz yerleşmiş hücreler kurala uymadıkları için komşu sayılmazlar. Şekil-4.11'de 4-değişkenli Karnaugh haritasında komşuluklar gösterilmiştir.



Şekil-4.10 4-değişkenli Karnaugh haritası ve hücrelerin adresleri.



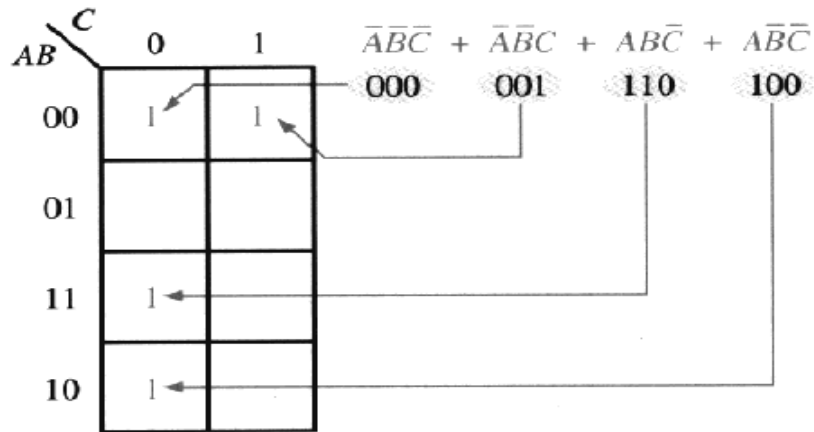
Şekil-4.11 Hücre komşulukları.

Standart Çarpımların Toplamı İfadelerinin Yerleştirilmesi

Standart formdaki ÇT ifadelerde her bir terim bir hücreye yerleştirilir. İfadede yer alan terimlerin yerine 1, yer almayanların bulunması gereken adrese 0 yerleştirilir. Yerleştirme bittikten sonra haritadaki 1 sayısı ile standart formdaki ÇT ifadedeki çarpım terimlerin sayısı eşit olmalıdır. Şekil-4.12'de buna 3-değişkenli haritada örnek verilmiştir.

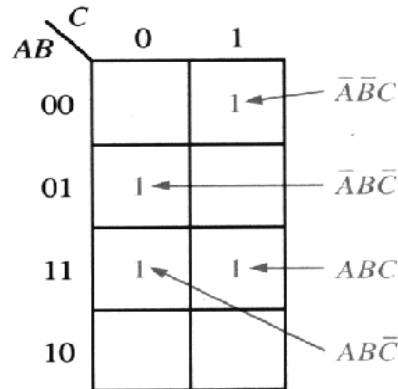
ÖRNEK 1: $A'B'C + A'B'C' + ABC' + AB'C'$ ifadesini haritaya yerleştirin.

ÇÖZÜM: Öncelikle ikilik karşılıklarını yazalım.



Şekil-4.12 Standart formdaki bir ÇT ifadenin karnaugh haritasına yerleştirilmesi.

ÖRNEK 2: $A'B'C + A'BC' + ABC + ABC'$ ifadesini haritaya yerleştirin.



ÇÖZÜM: Öncelikle ikilik karşılıklarını yazalım.

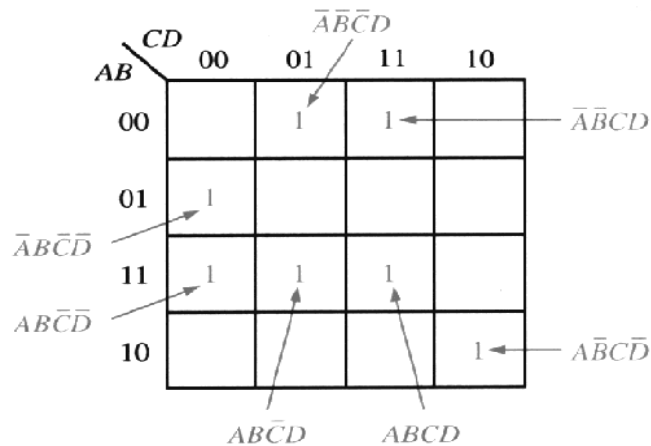
$$A'B'C + A'BC' + ABC + ABC'$$

001 010 111 110

ÖRNEK 3: $\bar{A}\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + AB\bar{C}\bar{D} + A\bar{B}C\bar{D}$

İfadesini Karnaugh haritasına yerleştirin.

ÇÖZÜM:



Standart olmayan ÇT ifadelerinin haritaya yerleştirilmesi.

Standart formda olmayan ÇT ifadeleri öncelikle standart hale getirilir. Boolean kuralları uygulanarak ifadeler genişletilir. Standart olmayan terimde bulunması gereken değişken bu terime eklenir. Ekleme yapılırken standart olmayan terim kayıp terimin kendisi ile değilinin toplamı ile çarpılır. Böylece artık iki terim elde edilir. Bir anlamda sadeleştirmenin tersi yapılır. $ABC+A'C$ ifadesinde standart olmayan terim $A'C$ terimidir, eksik olan değişken ise B değişkenidir. B 'nin kendisi ve B 'nin değili ile bu terim ayrı ayrı çarpılır. $ABC+A'CB+A'CB'$ ifadesi elde edilir. Bu genişletme pratik şekilde harita üzerinde de yapılabilir.

ÖRNEK 1: $A'+AB'+ABC'$ ifadesini haritaya yerleştirin.

ÇÖZÜM: İfade standart olmadığı için standart hale getirelim.

$$A' + AB' + ABC'$$

$$000 \quad 100 \quad 110$$

$$001 \quad 101$$

$$010$$

$$011$$

		C
AB		
	1	1
	1	1
	1	
	1	1

ÇT İFADELERİNİN KARNAUGH HARİTASI İLE SADELEŞTİRİLMESİ

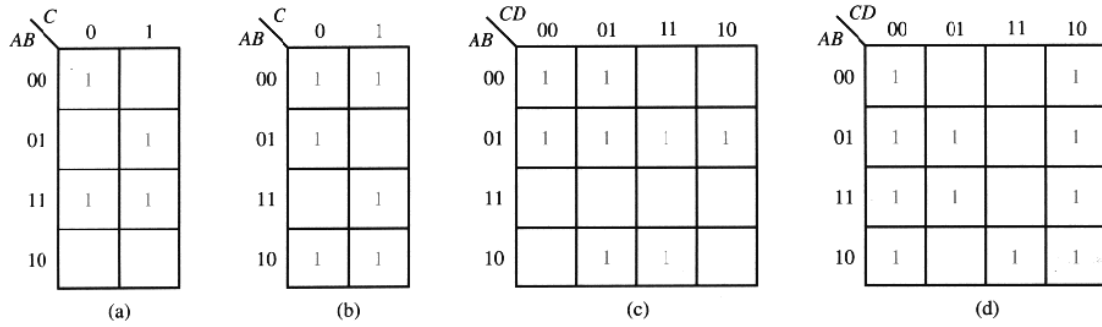
Amaç ÇT ifadeleri en az terime ve en az değişkenli hale getirmektir. Bir önceki kısımda ÇT ifade haritaya yerleştirildi, bu kısımda haritada terimler birleştirilerek tekrar ÇT ifade haline getirilecek. Birleştirmenin yapılabilmesi için öncelikle komşu 1'ler belirlenmeli, komşu 1'ler ikişer dörder veya sekizerli gruplar halinde birleştirilmelidir. Gruplar belirlenirken aşağıdaki kurallara uyulmalıdır.

- Grup 1,2,4,8,16 hücrelerden (içinde 1 olan hücre) oluşabilir.
- Gruptaki her hücre bir veya daha fazla hücre ile komşu olmalı. Gruptaki tüm 1'lerin birbirlerine komşu olması beklenmez.
- Koşul 1'deki kurala uymak koşulu ile grup maksimum sayıda 1'den oluşmalıdır.

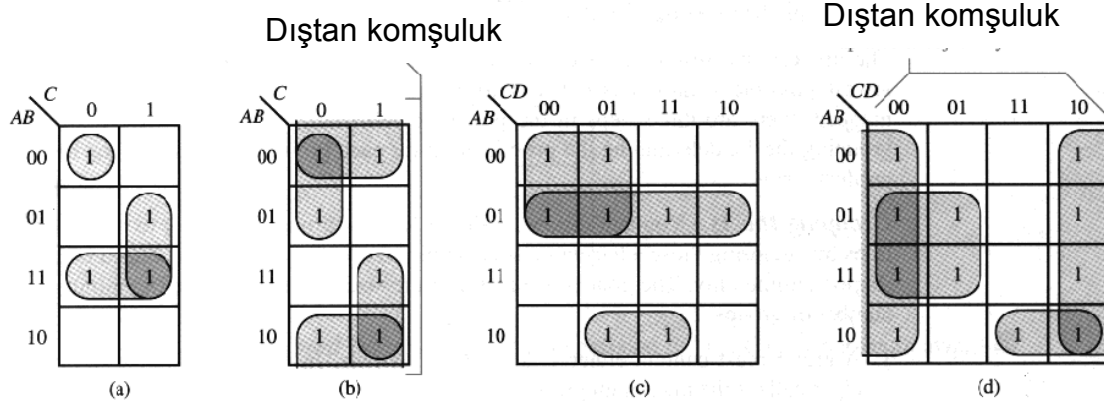
- Karnaugh haritasındaki tüm birler bir grup tarafından içermelidir. İçerilmeyor ise tek başına grup yapılmalıdır.

ÖRNEK:

Bu kurallara uyarak Şekil-4.13'de verilen Karnaugh haritalarındaki grupları belirleyin. Gruplar belirlendikten sonra en az sayıda değişken ile yazılması gerekir. Gruplar doğru belirlendiğinde en az sayıda terim elde edilir. Çünkü her grup bir terimi temsil eder. Grubun genişliğini ise terimdeki değişken sayısını belirler. 4-değişkenli Karnaugh haritasında grup bir adet 1'den oluşuyor ise bu grup 4 değişkenle, 2 adet 1'den oluşuyorsa 3 değişkenle, 4 adet 1'den oluşuyorsa 2 değişkenle, 8 adet 1'den oluşuyorsa 1 değişkenle tanımlanabilir. Eğer Karnaugh haritası tamamı bir grup ise ifadenin değeri 1'dir. 3-değişkenli Karnaugh haritasında ise grup 1 adet 1'den oluşuyor ise bu grup 3 değişkenle, 2 adet 1'den oluşuyorsa 2 değişkenle, 4 adet 1'den oluşuyorsa 1 değişkenle tanımlanabilir. 8 adet 1'den oluşan grup varsa bu ÇT ifadenin değeri 1'e eşittir.



- Fⁱ Aşağıda gösterilenden farklı bir şekilde gruplar oluşturulabilir fakat ifadedeki terim sayısı veya değişken sayısında bir azalma olmaz.



Şekil-4.13

ÖRNEK: Şekil-4.14'te verilen Karnaugh haritalarının en sade ÇT ifadelerini yazın.

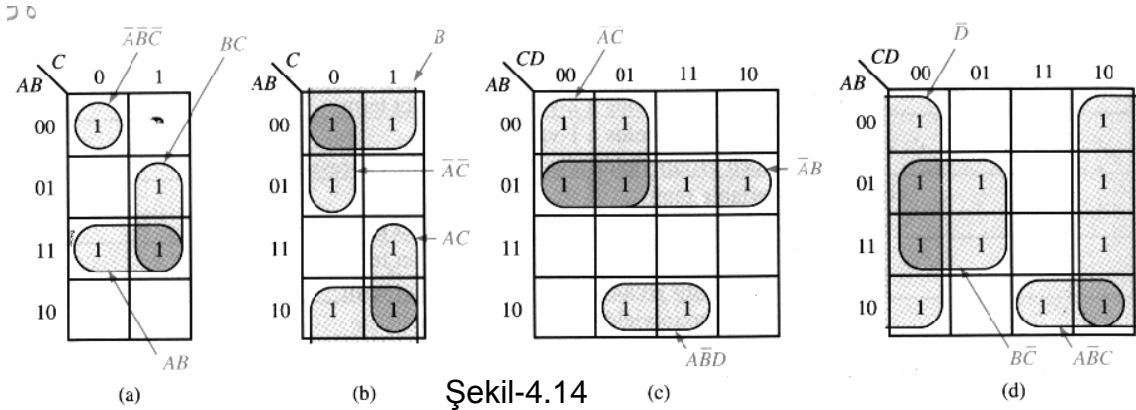
ÇÖZÜM: Grupların terimleri Karnaugh haritası üzerine yazılır ve sonra bu terimler toplanarak ÇT ifade elde edilir.

(a). $AB+BC+A'B'C'$

(b). $B'+A'C'+AC$

(c). $A'B+A'C'+AB'D$

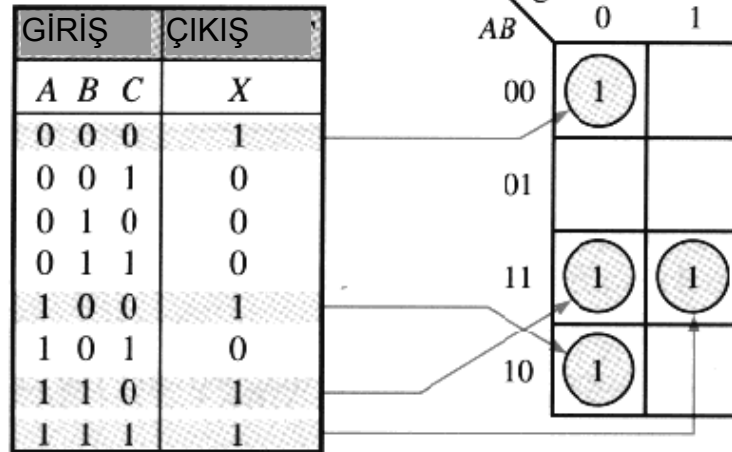
(d). $D'+AB'C+BC'$



DOĞRULUK TABLOSUNDAN KARNAUGH HARİTASINA GEÇİŞ

Doğruluk tablosu bir devrenin değişkenlerinin tüm olasılıklarını ve bu olasılıklara göre devrenin çıkışını belirler. Doğruluk tablosundan çıkışın 1 olduğu satırlar ÇT ifadenin terimlerini oluşturur. Tablodan yazılan ÇT ifade en geniş ifadedir. Tablonun her satırının Karnaugh haritasında bir karşılığı vardır. Daha önce belirlediğimiz adreslere göre tablonun değeri Karnaugh haritasına geçirilebilir. Karnaugh haritasında sadeleştirilerek sade halde yazılabilir. Tasarım yapılırken önce problem tanımlanır daha sonra değişkenler belirlenir. Değişken sayısı oranında doğruluk tablosu yapılır ve tüm giriş olasılıkları bu tabloda yer alır. Çıkışın 1 olması gereken satırlar belirlenir ve 1 yazılır. Buradan Karnaugh haritasına geçirilir ve en sade ÇT ifade elde edilir. En son adımda mantık geçitleri kullanılarak önce devrenin şekli çizilir ve devre kurularak çalıştırılır.

$$X = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$



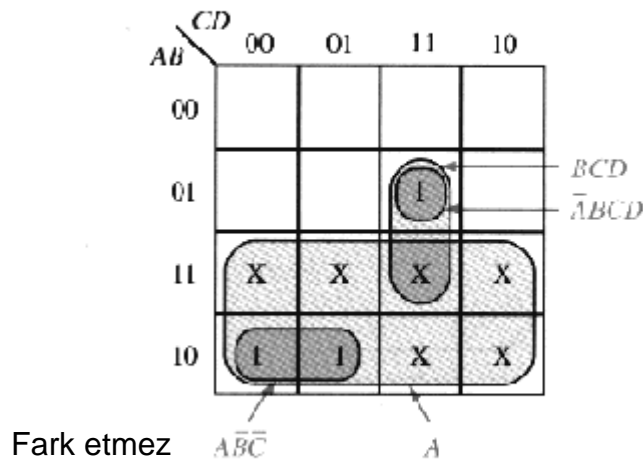
Şekil-4.15

Şekil-4.15'te doğruluk tablosundan Karnaugh haritasına geçişe örnek verilmiştir.

FARK ETMEZ (DON'T CARE) DURUMU

Bazı durumlarda devrenin giriş değişken sayısı fazladır, fakat girişe aslında bu olasılıkların gelmesi mümkün değildir. Bazı olasılıklar girişe gelir ve çıkış üzerinde etkisi vardır. Gelmeyen giriş olasılıkları ise çıkışı etkilemeyecektir fakat doğruluk tablosunda ve Karnaugh haritasında bu giriş olasılıkları bulunacaktır. Bu durumlara *fark etmez durumu* adı verilir ve X ile gösterilir. Bu farketmez durumları gerekirse "1" kabul edilerek diğer "1"lerle gruplanabilir. Şekil-4.16'da bu duruma bir örnek verilmiştir.

GİRİŞ	ÇIKIŞ
ABCD	Y
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	1
1000	1
1001	1
1010	X
1011	X
1100	X
1101	X
1110	X
1111	X



$$\text{Fark etmeziz } Y = A + B' C' + A' B C D$$

$$\text{Fark etmezli } Y = A + B C D$$

Şekil-4.16

TASARIM ÖRNEĞİ

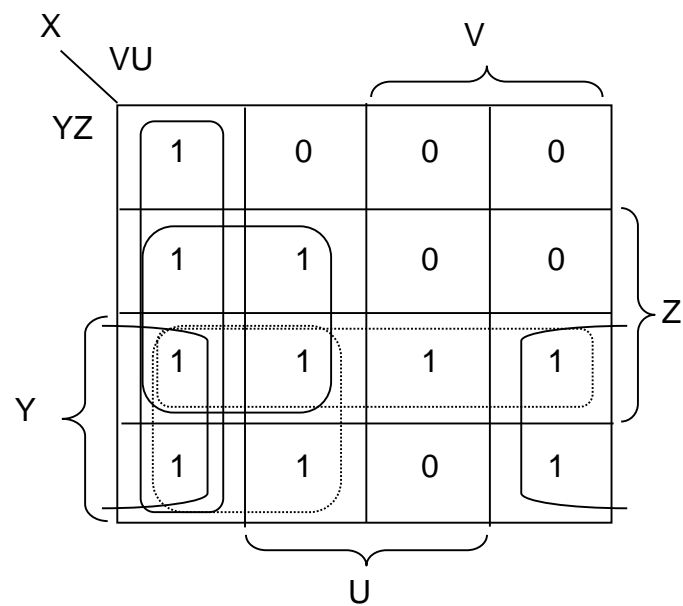
Örnek: A ve B gibi iki bitlik iki sayı birbirine eşit veya A büyük ise devrenin çıkışı "1", diğer durumlarda sıfır olsun. Bu işlemi yapan devreyi en az geçit kullanarak tasarlayın.

Çözüm: $A = A_1 A_0$ ve $B = B_1 B_0$ şeklinde gösterelim ve doğruluk tablosunu oluşturalım.

Karnaugh haritasından elde edilen ifade en sade ifade olduğuna göre devresi bu eşitliğe göre çizilebilir.

A		B		
Y	Z	V	U	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

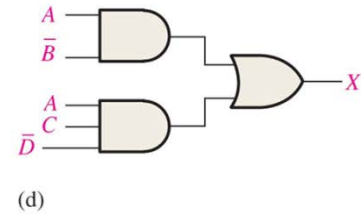
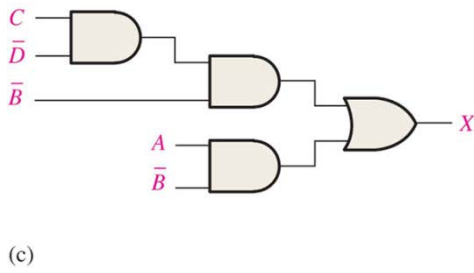
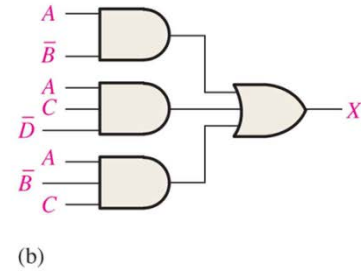
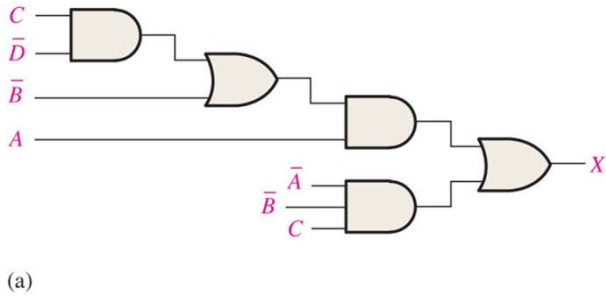
Tablo-4.2



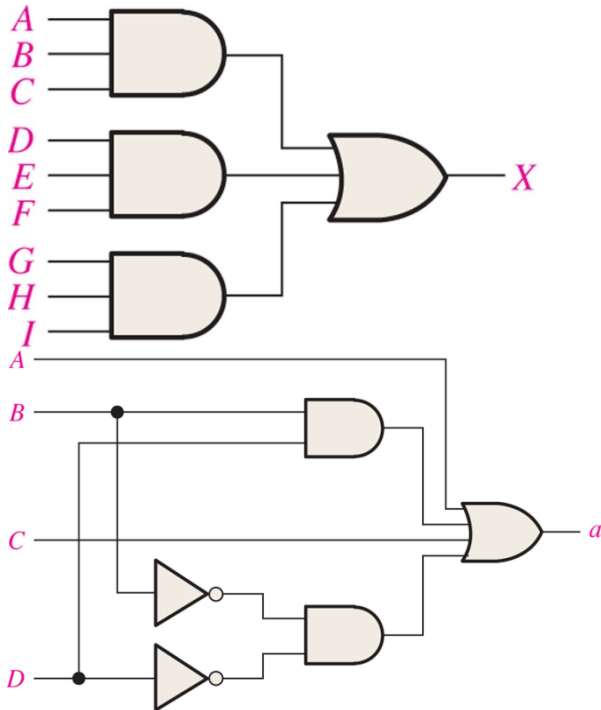
$$X = Y\bar{V} + Z\bar{V} + YZ + \bar{U}\bar{V} + \bar{U}Y$$

Sorular

1. Aşağıda verilen sayısal devrelerin çıkış eşitliklerini çarpımların toplamı olarak yazın?



2. Aşağıda verilen sayısal devrelerin çıkış eşitliklerini çarpımların toplamı olarak yazın?



3. Aşağıda verilen karnough haritasından mantık eşitliğini yazın.

		C	
		0	1
AB	00	1	
	01	1	1
	11		1
	10		

X=.....

		C	
		0	1
AB	00		
	01		1
	11		
	10	1	1

Y=.....

AB	CD			
	00	01	11	10
		1		
		1		1
	1	1	1	1
	1	1	1	1

W=.....

AB	CD			
	00	01	11	10
				1
	1		1	1

Z=.....

BÖLÜM 5

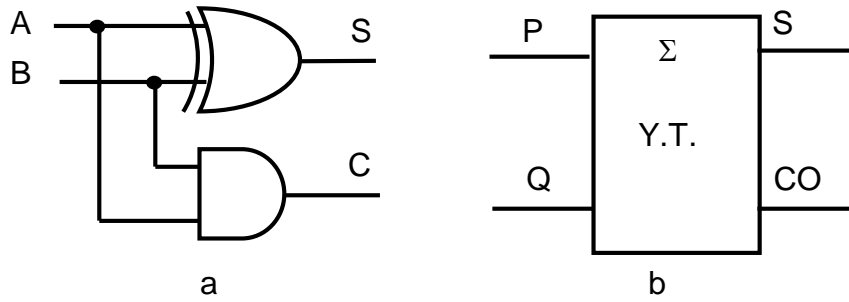
TOPLAYICI VE KARŞILAŞTIRICI

TOPLAYICILAR

Toplayıcı devreler ikilik sayıları toplar ve sonucu yine ikilik olarak çıkışa yazarlar. Elde girişi olmayan toplayıcı devreye yarım toplayıcı, olan devrelere tam toplayıcı adı verilir. Yarım toplayıcı devreyi geçitler kullanarak tasarlayalım. Devrenin yapması gereken işlevi doğruluk tablosunda oluşturalım.

A	B	TOPLAM (S)	ELDE (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tablo-5.1 Yarım toplayıcının doğruluk tablosu.



Şekil-5.1 a. Yarım toplayıcının devresi b. Mantık sembolü.

Toplam çıkışının fonksiyonunu yazacak olursak ifadesi elde edilir. Elde çıkışınınınki ise $C=AB$ 'dir. Yarım toplayıcıyı bir EXOR ve bir VE geçidi ile oluşturulabilir. Şekil-5.1'de yarım toplayıcı devresi gösterilmiştir.

Yarım toplayıcı devresi sadece iki adet bir bitlik ikilik sayının toplanmasında kullanılabilir. İki bitlik sayının toplanması için devrenin elde girişi olan bir devre

tasarlanması gerekir. Elde girişi olan bir toplayıcı devresine tam toplayıcı devresi adı verilir. Tam toplayıcı devresinin doğruluk tablosu Tablo-5.2'de ve devresi Şekil-5.2'de verilmiştir. Tablodan elde çıkışı ve toplam çıkışının eşitliklerini yazalım;

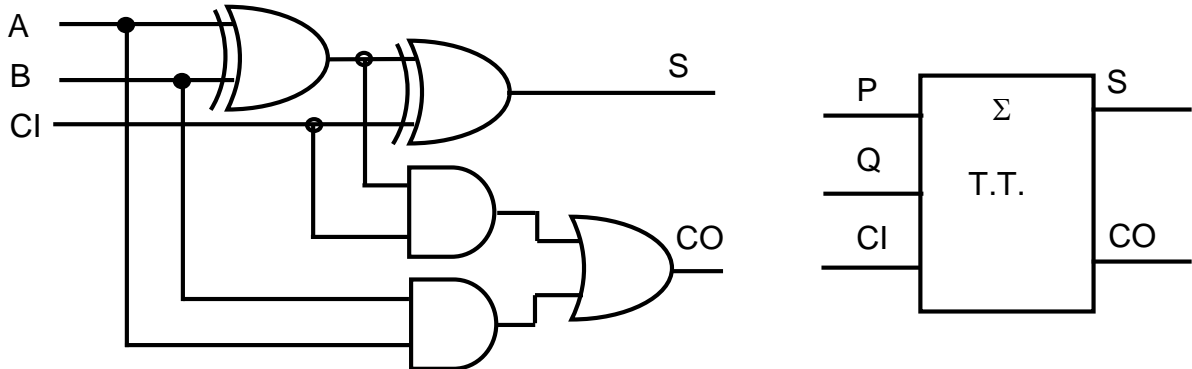
Toplam çıkışının fonksiyonu; $S = A \oplus B \oplus CI$

Elde çıkışının fonksiyonu;

Şekilden de anlaşılacağı gibi iki adet yarım toplayıcıdan tam toplayıcı elde edilebilir.

A	B	CI	S	CO
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

Tablo-5.2 Tam toplayıcı doğruluk tablosu



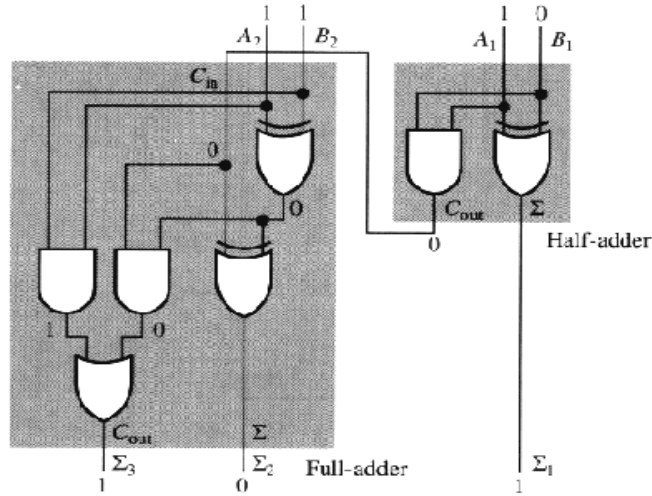
Şekil-5.2 Tam toplayıcının devresi ve mantık simgesi.

PARALEL TOPLAYICILAR

Tam toplayıcı bir bitlik iki sayıyı toplamak için kullanılabilir. Birden fazla bitlik ikilik sayıların toplanmasında tam toplayıcılar paralel bağlanarak kullanılır. Paralel toplayıcılarda en düşük değerli bit önce toplanır ve en son yüksek değerli bit toplanır. Düşük değerli bitlerin toplanmasından oluşan elde bir üst değerdeki sayıların toplanmasında kullanılır. En yüksek değerli bitlerin toplamından oluşan elde toplamın eldesidir. Toplam sonucunda sayının bit sayısı artmıştır. En yüksek değerli bitten oluşan eldeye toplam adı verilir.

Toplayıcılar SSI (Small-Scale Integration) tümdevreler kullanılarak elde edilebileceği gibi MSI (Medium-Scale Integration) tümdevreler kullanılarak da elde edilebilir. MSI tümdevrelerde kullanılan bağlantının azalmasının yanı sıra fiyatları da SSI tümdevrelere oranla daha düşüktür. Bu konunun deneyinde MSI ailesinden 7483 4 bit tam look-ahead-carry toplayıcıyı kullanarak iki adet 4 bitlik sayıyı toplayacaksınız. 74283 ise diğerinin yeni sürümüdür, besleme bacakları standart

hale getirilmiştir. Onun dışında işlevsel olarak farkları yoktur.



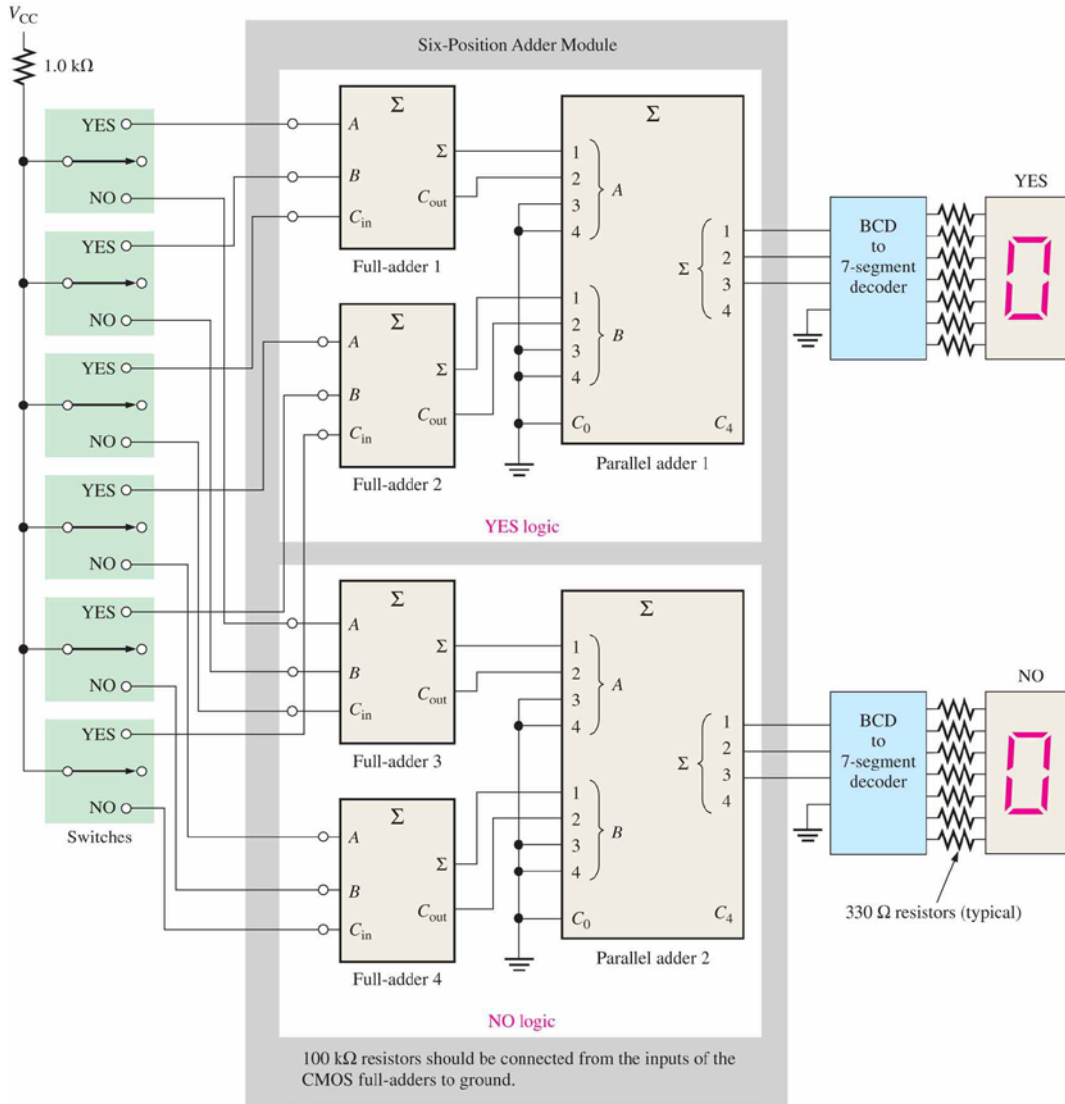
Şekil-5.3 Paralel toplayıcı.

Şekil-5.4'te toplayıcı kullanılarak oy sayım makinesi şeması verilmiştir. Devrenin evet ve hayır sayan ayrı iki göstergesi vardır. Oy verme anahtarları üç konumludur; evet ve hayır konumlarında +5 volt uygulanır. Evet ve hayır oyları farklı iki devre tarafından toplanır.

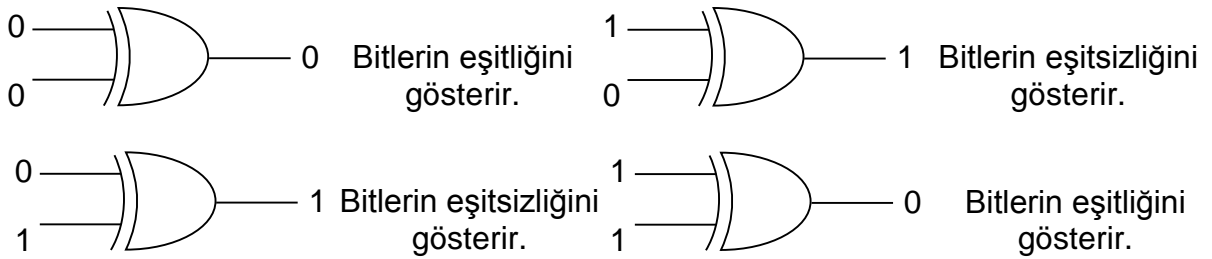
KARŞILAŞTIRICILAR

Karşılaştırmaların temel işlevi iki niceliğin büyüklüğünü karşılaştırarak bu niceliklerin arasındaki ilişkiyi belirlemektir. En basit şekliyle karşılaştırıcı iki sayının eşitliğini denetler. Zıt geçidi (EXOR) temel bir karşılaştırıcı olarak kullanılabilir çünkü bu geçidin çıkışı iki giriş biti eşit değilse "1", girişler eşitse "0"dır. Şekil-5.5'te EXOR geçidinin iki bitlik karşılaştırıcı olarak kullanılması gösterilmektedir.

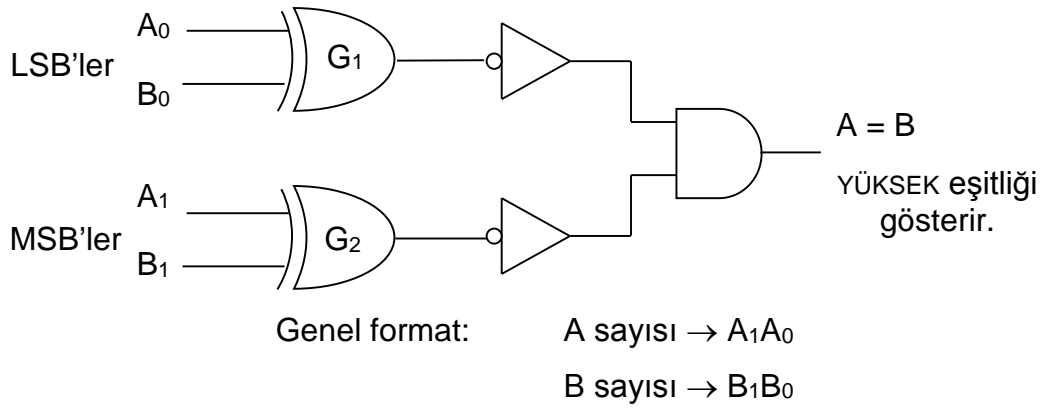
İki bitlik ikilik sayıları karşılaştırmak için ek bir EXOR geçidine gerek vardır. İki sayının en az anlamlı iki biti (LSB) G₁ geçidiyle ve en anlamlı iki biti (MSB) G₂ geçidiyle karşılaştırılır (şekil-5.6). İki sayı eşitse, karşılıklı bitleri eşittir ve EXOR geçitlerinin çıkışları "0"dır. Karşılıklı bitleri eşit değilse, EXOR geçidinin çıkışı "1"dir. İki sayının eşit yada eşit olmadığını belirten *tek* bir çıkış elde edebilmek için, Şekil-5.6'da gösterildiği gibi, iki evirici ve bir VE geçidi kullanılır. Her bir EXOR geçidinin çıkışı değillenir ve VE geçidinin girişine uygulanır. Her bir EXOR geçidinin giriş bitleri eşit olduğunda, sayıların karşılıklı bitleri de eşittir ve VE geçidinin her iki girişi "1" olduğu için çıkışında da "1" görülür. Sayılar eşit değilse, karşılıklı bit setlerinden biri yada her ikisi eşit değildir ve VE geçidinin en az bir girişine "0" geldiği için çıkışında da "0" görülür. Sonuç olarak, VE geçidinin çıkışı "1" ise sayıların eşitliği, "0" ise eşitsizliği belirtilir. Örnek-5.1'de her iki durumda da çalışması anlatılmaktadır. EXOR geçidi ile eviricilerin yerini EXNOR (eş) geçidi almıştır.



Şekil-5.4 Örnek toplayıcı uygulaması.



Şekil-5.5 Temel karşılaştırıcının çalışması.



Şekil-5.6 İki 2-bitlik ikilik sayının karşılaştırmasını yapan devrenin mantık çizgesi.

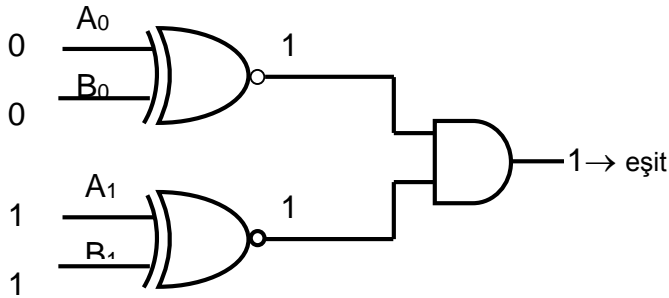
Örnek-5.1 Her bir ikilik sayı setini Şekil-5.7'deki karşılaştırıcının girişlerine uygulayın ve devrenin her noktasının mantık seviyelerini belirleyerek çıkışı bulun.

(a) 10 ve 10

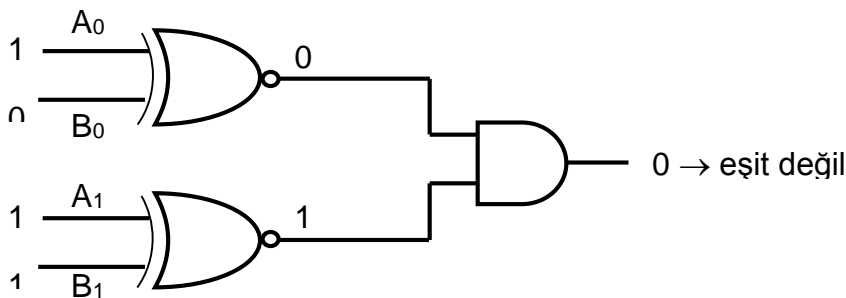
(b) 11 ve 10

Çözüm:

Şekil-5.7(a)'da gösterildiği gibi, 10 ve 10 girişleri için çıkış "1" dir.



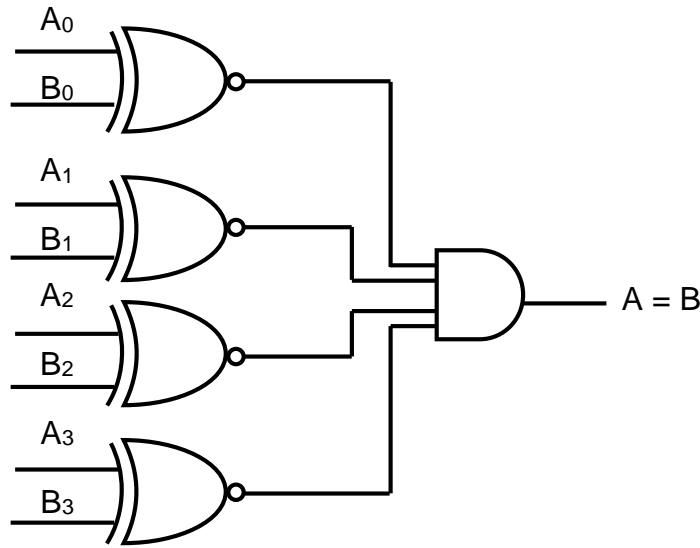
Şekil-5.7(a)



Şekil-5.7(b)

Şekil-5.7(b)'de gösterildiği gibi, 11 ve 10 girişleri için çıkış "0" dır.

İki adet 4-bitlik sayı için Şekil-5.8'de gösterildiği gibi, karşılaştırıcı devresi herhangi bit sayısına genişletilebilir. Bu devreye göre, iki sayı eşitse karşılıklı bitleri de eşittir ve VE geçidinin çıkışı "1" olur.



Şekil-5.8 İki 4-bitlik sayının, $A_3A_2A_1A_0$ ve $B_3B_2B_1B_0$, karşılaştırılmasını gösteren mantık devresi

MSİ KARŞILAŞTIRICILAR

Bazı tümdevre karşılaştırıcıların karşılaştırılan iki sayıdan hangisinin daha büyük olduğunu belirten çıkışları da vardır. Şekil-5.9'daki dört-bitlik karşılaştırıcı mantık simgesinde görüldüğü gibi, A sayısı B sayısından büyük olduğunda ($A > B$) ve A sayısı B sayısından küçük olduğunda ($A < B$) belirten çıkışlar da eklenmiştir.

Bu iki çıkış işlevinin devreye eklenmiş hali Şekil-5.10'da gösterilmiştir. $A > B$ ve $A < B$ işlevlerini yerine getiren mantık devresinin çalışmasını anlayabilmek için iki adet ikilik sayıyı inceleyerek sayıların eşitsizliğini belirleyen durumu açıklayalım.

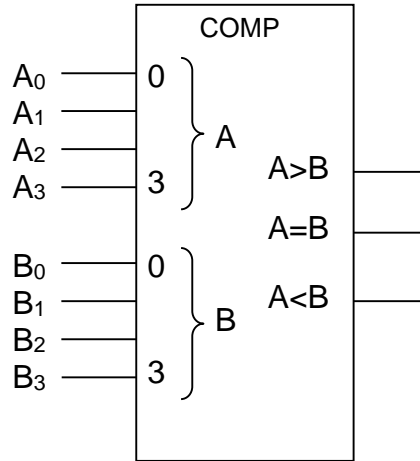
Genel biçimi A sayısı için $A_3A_2A_1A_0$ ve B sayısı için $B_3B_2B_1B_0$ olan iki 4-bitlik sayıyı kullanalım. A ve B sayılarının eşitsizliğini belirlemek için öncelikle her sayıda en önemli bitleri incelemeliyiz. Aşağıdaki koşullar olasıdır:

$A_3 = 1$ ve $B_3 = 0$ ise, A sayısı B sayısından büyüktür.

$A_3 = 0$ ve $B_3 = 1$ ise, A sayısı B sayısından küçüktür.

$A_3 = B_3$ ise, eşitsizlik için bir sonraki bitleri incelemeliyiz.

Bu üç inceleme sayılardaki her bit konumu için geçerlidir. Genel yöntem bir bit konumundaki eşitsizliği en anlamlı bitlerden başlayarak sınamaktır. Bir eşitsizlik bulunduğu, iki sayı arasındaki ilişki belirlenmiştir ve diğer daha az değerli bitlerin durumları *gözardı edilmelidir* çünkü zıt bir eşitsizlik belirtimi ortaya çıkabilir; en değerli sıradaki eşitsizlik ifadesi *önceliklidir*. Bunu açıklamak için A sayısı 0111 ve B sayısı da 1000 olsun. A_3 ve B_3 bitlerinin karşılaştırılması sonucu A sayısının B sayısından küçük olduğu ortaya çıkar, çünkü $A_3 = 0$ ve $B_3 = 1$ 'dir. Ancak A_2 ve B_2 bitleri karşılaştırıldığında $A_2 = 1$ ve $B_2 = 0$ olduğundan $A > B$ ifadesiyle karşılaşıyoruz. Aynı koşul diğer daha az değerli bitler için de geçerlidir. Bu durumda öncelik A_3 ve B_3 bitlerine verilmelidir çünkü uygun eşitsizlik koşulunu bu bitler sağlar.



Şekil-5.9 Eşitsizlik gösterimine sahip 4-bitlik büyüklük karşılaştırıcısının mantık simgesi.

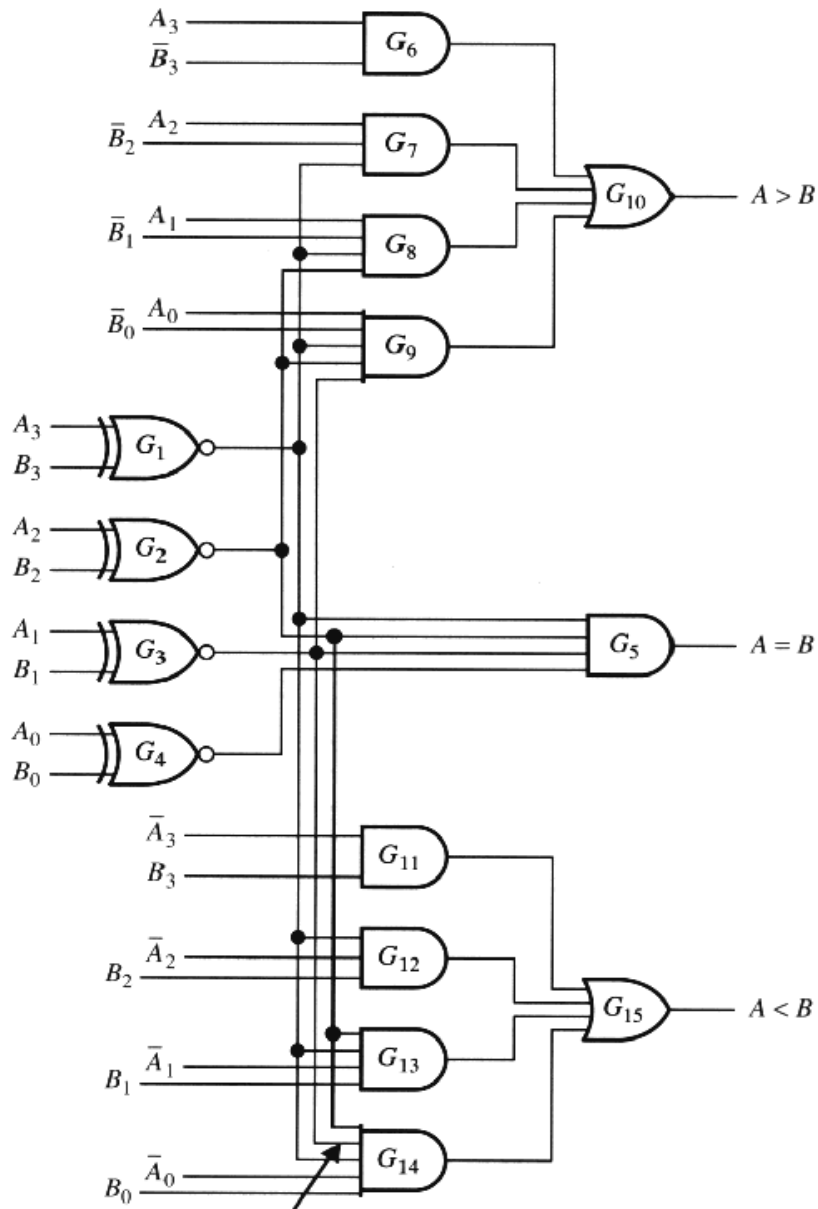
Şekil-5.10 iki 4-bitlik sayının karşılaştırılması ve $A > B$, $A < B$ veya $A = B$ çıkışlarının üretilmesi yöntemini göstermektedir. $A > B$ koşulu G_6 ile G_{10} arası geçitlerle sağlanmaktadır. G_6 geçidi $A_3=1$ ve $B_3=0$ 'ı denetlemektedir ve çıkış fonksiyonu A_3B_3 'tür. G_7 geçidi $A_2=1$ ve $B_2=0$ 'ı denetlemektedir ve çıkış fonksiyonu A_2B_2 'dir. G_8 $A_1=1$ ve $B_1=0$ 'ı denetlemektedir (A_1B_1). G_9 $A_0=1$ ve $B_0=0$ 'ı denetlemektedir (A_0B_0). Bu koşulların hepsi A sayısının B sayısından daha büyük olduğunu gösterir. Tüm bu geçitlerin çıkışları $A > B$ çıkışını elde etmek için G_{10} geçidiyle VEYA'lanır.

G_1 geçidinin çıkışı G_7 , G_8 , G_9 geçitlerinin girişlerine bağlanmıştır. Bu bağlantıyla, A_3 ve B_3 bitlerinde uygun eşitsizlik oluştuğunda ($A_3 < B_3$), diğer daha az değerli bitlerin incelenmesi engellenir. Aynı amaçla G_2 geçidi G_8 ve G_9 'a, G_3 geçidi de G_9 'a bağlanmıştır.

G_{11} ile G_{15} arası geçitler $A < B$ koşulunu sınamaktadır. Her VE geçidi A sayısında 0'ın ve B sayısında 1'in olup olmadığını denetler. VE geçidi çıkışları G_{15} geçidi ile VEYA'lanarak $A < B$ çıkışı elde edilir. Uygun eşitsizlik bulunduğunda daha az değerli bitlerin incelenmesinin engellenmesi yukarıda anlatıldığı gibidir. A sayısındaki dört bit B'deki bitlere eşitse, her bir EXNOR geçidinin çıkışı 1'dir. Bu da $A = B$ çıkışındaki G_5 geçidinin çıkışının "1" olmasını sağlar.

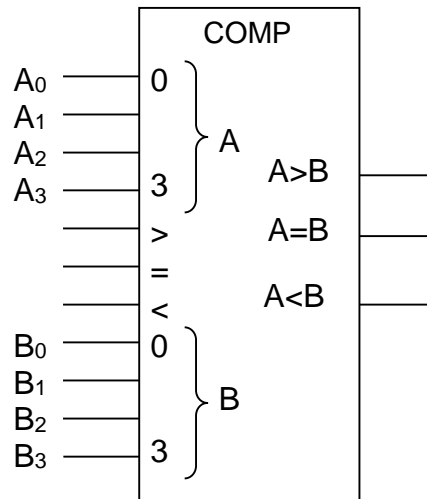
7485 Dört-Bitlik Büyüklük Karşılaştırıcı

7485 MSI teknolojisi ile üretilen büyüklük karşılaştırıcısıdır. Mantık simgesi Şekil-5.11'dedir. (Bu tümdevre hakkındaki teknik bilgiyi EK 1'de veri yaprakları kısmından bulabilirsiniz.)



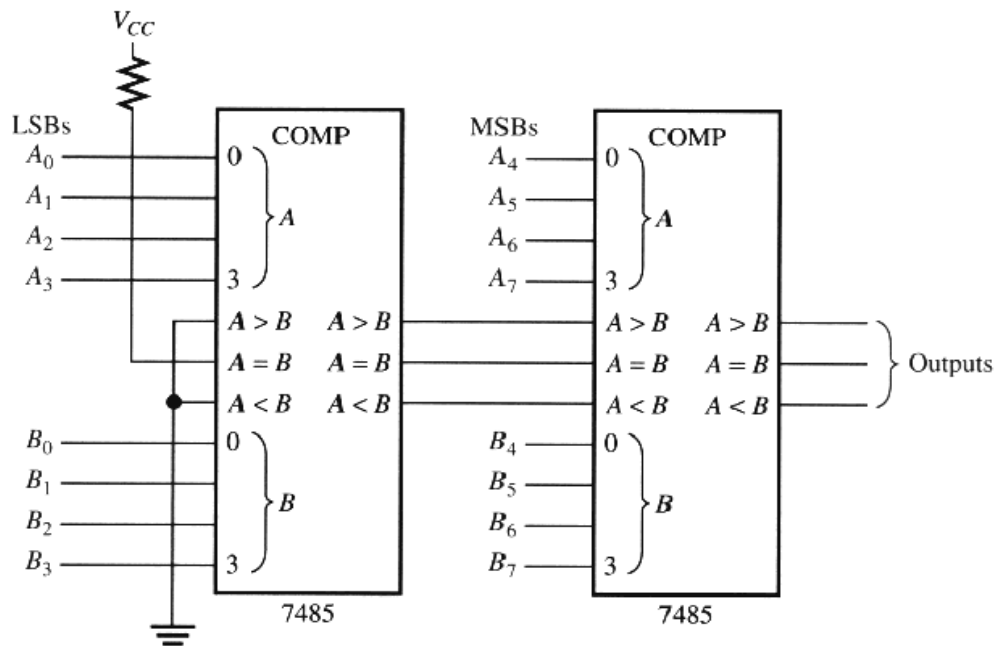
Şekil-5.10 4 bitlik büyüklük karşılaştırıcı

Daha önce bir büyüklük karşılaştırıcısında olduğunu belirttiğimiz giriş ve çıkışlara ek olarak üç adet *kaskat bağlama girişi* de bulunmaktadır. ($<$, $=$, $>$). Bu girişler dört bittten büyük sayıların karşılaştırılabilmesi için birkaç karşılaştırıcının kaskat bağlanabilmesine olanak sağlar. Karşılaştırıcıyı genişletebilmek için $A < B$, $A = B$ ve $A > B$ çıkışları bir sonraki daha değerli karşılaştırıcının karşılık gelen girişlerine bağlanır. En az değerli karşılaştırıcının $=$ girişi YÜKSEK, $<$ ve $>$ girişleri DÜŞÜK seviyeye bağlanmalıdır.



Şekil-5.11 7485 4-bitlik büyüklük karşılaştırıcı tümdevresinin mantık simgesi

Bu tümdevre kullanılarak 4 bitten daha fazla uzunluktaki ikilik sayılar karşılaştırılabilir. Şekil-5.12 8 bitlik iki sayının karşılaştırılmasına örnek verilmiştir.



Şekil-5.12 8 bitlik karşılaştırıcı.

BÖLÜM 6

KODÇÖZÜCÜLER VE KODLAYICILAR

GİRİŞ

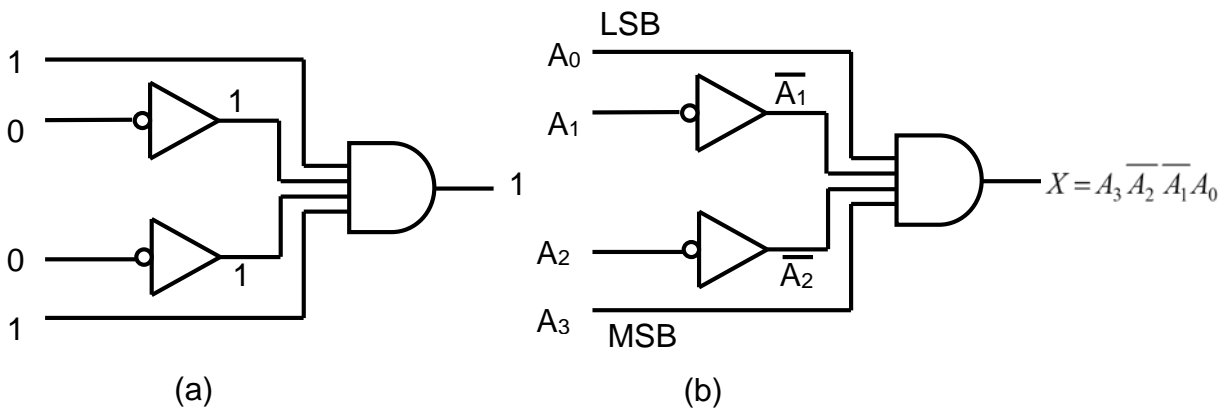
Kodçözücü giriş bitlerinin belirli kombinasyonunun girişe gelip gelmediğini denetler, geldiğinde çıkışını etkin hale getirir. Girişine gelen veriyi bir başka forma dönüştürür. Günlük yaşamda onluk sayı sistemi kullanırız, fakat bilgisayar ikilik sayılar ile işlem yaptığı için bilgisayar kullanmak istediğimizde onludan ikiliye kodlayıcıya gereksinim duyarız. Kodlayıcının kodladığı bilgi tekrar günlük yaşamın parçası olabilmesi için kodçözücü kullanılarak kodu eski haline getirilmelidir.

KOD ÇÖZÜCÜLER (DECODERS)

Kodçözücünün temel işlevi girişlerinde belirli bir bit birleşiminin (kod) olup olmadığını algılamak ve varsa bunu belirli bir çıkış seviyesiyle göstermektir. Genel olarak, kodçözücülerin n girişi varsa bir ile 2^n adet çıkışı olabilir.

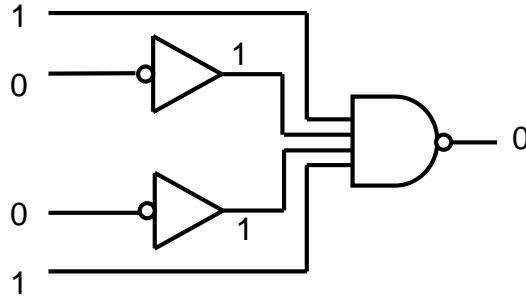
TEMEL İKİLİK KODÇÖZÜCÜ

Bir dijital devrenin girişlerine ikilik 1001 sayısı gelip gelmediğini belirlemek için bir VE geçidi temel kodçözücü elemanı olarak kullanılabilir çünkü bu geçit tüm girişleri YÜKSEK olduğunda YÜKSEK seviye üretir. Dolayısıyla, 1001 ikilik sayısı oluştuğunda VE geçidinin bütün girişlerine YÜKSEK seviye gelebilmesi için ortadaki iki biti (0'lar) değillemek gerekir Şekil-6.1(a)'daki kodçözücü için mantık fonksiyonu Şekil-6.1(b)'de gösterildiği gibidir. Çıkış fonksiyonu $A_0 = 1$, $A_1 = 0$, $A_2 = 0$ ve $A_3 = 1$ girişlere uygulanmadığı sürece "0"dır. A_0 en az değerli bit ve A_3 en değerli bittir.



Şekil-6.1 1001 için etkin-YÜKSEK seviyeli kodçözücü devre.

VE geçidinin yerine VED geçidi kullanıldığında (Şekil-6.2), uygun ikilik kodun varlığını DÜŞÜK seviye ile belirtir.



Şekil-6.2 1001 için etkin-DÜŞÜK seviyeli kodçözücü devre.

Örnek-6.1

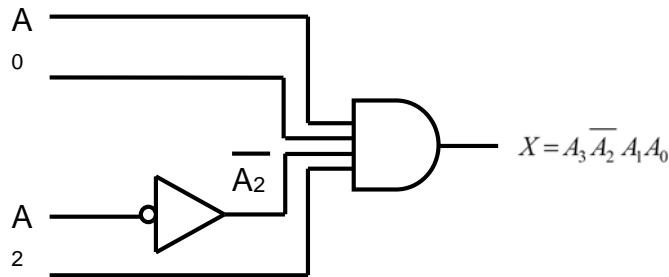
1011 ikilik sayısının kodunun çözülmesi için gerekli etkin-YÜKSEK çıkışlı mantık devresini çizin.

Çözüm:

İkilik sayıdaki sıfır olarak görünen değişkenin değillenmesiyle fonksiyon bulunmuş olur:

$$X = A_3 \overline{A_2} A_1 A_0$$

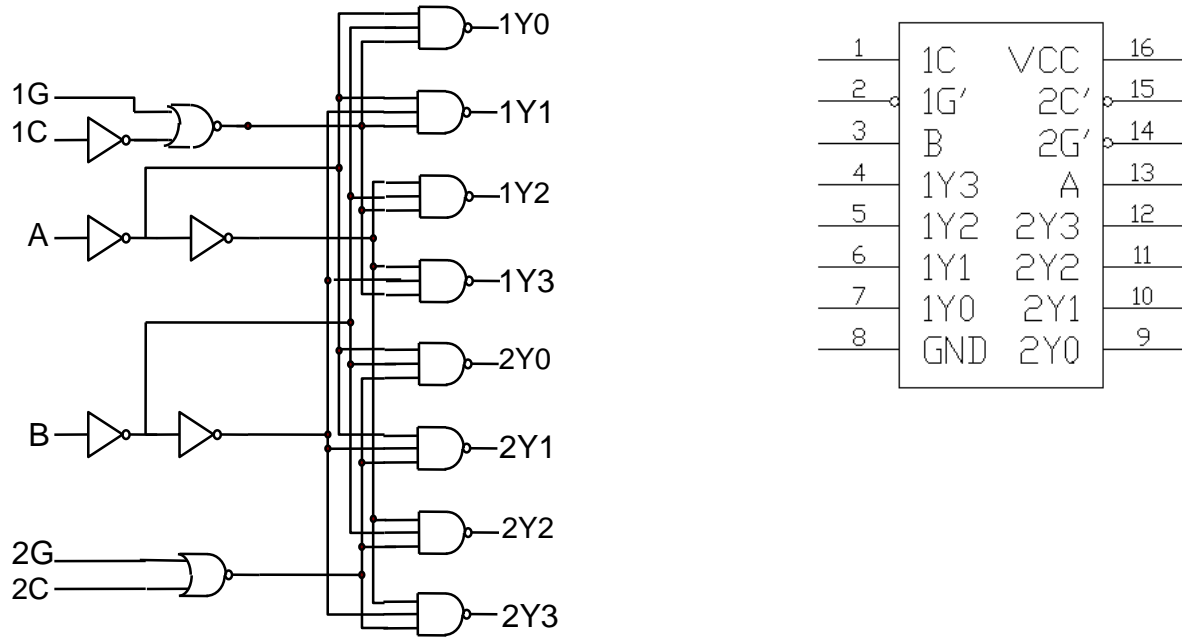
Bu fonksiyon değillenmemiş değişkenler olan A_0 , A_1 , A_3 'ü VE geçidinin girişine doğrudan bağlayarak ve A_2 değişkenini değilleyerek VE geçidinin girişine uygulayarak gerçekleştirilir (Şekil-6.3).



Şekil-6.3 1011 için etkin-YÜKSEK seviyeli kodçözücü devre.

74156 VE 74138 3-E-8 KODÇÖZÜCÜLER

Tümdevre şeklinde üretilmiş kod çözücülerde (MSI) olası tüm çıkışlar tümdevre üzerine yerleştirilir, kullanıcılar istediği çıkışları kullanırlar. 3 bit ikilik kodçözücü 74LS156 tümdevresinin çıkış sayısı $2^3 = 8$ 'dir çıkışı vardır. Bu tümdevrenin doğruluk tablosu Tablo-6.1'de verilmiştir. Tümdevrenin veri yaprağı ekte verilmiştir. Açık kollektör olduğu için çıkış uçlarına yükseğe çekme dirençleri bağlanmalıdır. Aksi halde çalışmaz.



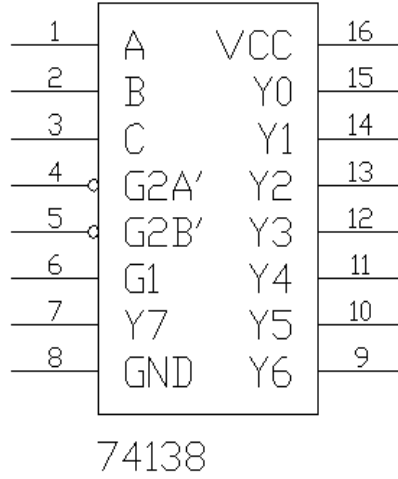
Şekil-6.4 74156 3 bit kodçözücü tümdevresinin mantık sembolü ve iç bağlantı şeması.

74LS156 tümdevresinde A, B girişleri tek, 3. giriş 1C ve 2C olarak iki parçaya bölünmüştür. 1C ve 2C girişleri birleştirilip kullanıldığında 3'e 8 kodçözücü olarak kullanılırken ayrı kullanıldığında iki adet 2'ye 4 kodçözücü olarak çalıştırılabilir. Bacak bağlantısı ve iç yapısı Şekil-6.4'te gösterilmiştir. 1G ve 2G girişleri birinci ve ikinci 2'ye 4 kodçözücülerin izin verme girişleridir. 3'e 8 kodçözücü olarak kullanıldığında bu iki giriş birleştirilerek kullanılır.

74LS138 tümdevresi 74LS156 tümdevresi gibi 3'e 8 kodçözücüdür. Farkı ise 3 adet kontrol girişine sahip olması ve açık kollektör olmaması ve direnç bağlantısına ihtiyaç duymamasıdır. Kontrol girişlerinden iki tanesi düşüğe etkin olurken diğeri YÜKSEKTE etkin olur. Çıkışlar etkin düşüğe göre çalışır. Şekil-6.5'te bu tümdevrenin mantık sembolü verilmiştir. Tablo-6.2'de ise aynı tümdevrenin doğruluk tablosu verilmiştir.

GİRİŞLER				ÇIKIŞLAR							
SEÇME			İZİN	0	1	2	3	4	5	6	7
C	B	A	G	2Y0	2Y1	2Y2	2Y3	1Y0	1Y1	1Y2	1Y3
X	X	X	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1
0	1	0	0	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1
1	0	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	1	1	0	1
1	1	1	0	1	1	1	1	1	1	1	0

Tablo-6.1 3 bit ikilik kodçözücü 74LS156 tümdevresinin doğruluk tablosu.



Şekil-6.5 74138 3 bit kodçözücü tümdevresinin mantık sembolü.

DÖRT BİTLİK İKİLİK KODÇÖZÜCÜ

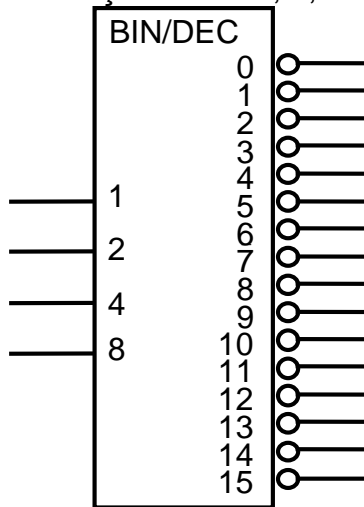
Dört bit için tüm olası kombinasyonların kodunun çözülebilmesi için on altı kod çözücü geçide gerek vardır ($2^4 = 16$). Bu tür bir kod çözücüyü *4-hattan-16-hatta* kod çözücü denir çünkü dört girişi ve on altı çıkışı vardır. On altı ikilik kod sözcüğü ve karşılık gelen mantık fonksiyonları Tablo-6.3'te verilmiştir.

GİRİŞLER					ÇIKIŞLAR							
SEÇME			İZİN		0	1	2	3	4	5	6	7
C	B	A	G ₁	G ₂	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	X	X	1	1	1	1	1	1	1	1	1
X	X	X	0	X	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	0	1	1	1	1	1	1
0	1	0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	1	0	1	1	1
1	0	1	1	0	1	1	1	1	1	0	1	1
1	1	0	1	0	1	1	1	1	1	1	0	1
1	1	1	1	0	1	1	1	1	1	1	1	0

Tablo-6.2 3 bit ikilik kodçözücü 74LS138 tümdevresinin doğruluk tablosu.

Her sayı için etkin-DÜŞÜK çıkış gerekirse, kodçözücü devrenin VED geçitleriyle ve eviricilerle tasarlanması gerekir. Öncelikle her değişkenin kendisi ve değiline gerek olduğundan değişkenlerin değilleri bir defa üretilir ve gerekli olan tüm geçitler için kullanılır. Onaltı ikilik kodun kodunun çözülebilmesi için onaltı VED geçidi gereklidir. Kodçözücünün devresi bu şekilde çizilebilir. Ancak bu şekildeki bir devrenin çizimi

karmaşık olacağı için 4-e-16 kodçözücü için Şekil-6.6'daki mantık simgesi kullanılır. BIN/DEC etiketi ikilik (binary) girişin karşılık gelen onluk (decimal) çıkışı etkin yaptığını belirtir. Giriş etiketleri 1, 2, 4 ve 8 giriş bitlerinin ikilik ağırlıklarını gösterir.



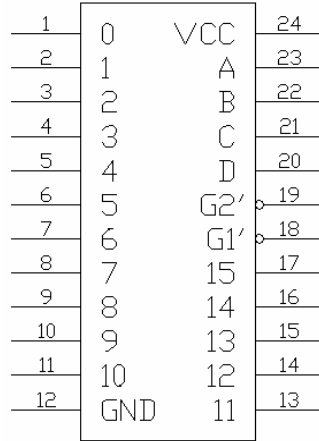
Şekil-6.6 İkiliden onluya kod çözücü.

[illegible]

Tablo-6.3 4-hattan-16-hatta kod çözücü için mantık fonksiyonu ve doğruluk tablosu.

74154 4-E-16 KODÇÖZÜCÜ

74154 TTL MSI bir kodçözücüdür. 74154 4 bit kodçözücü tümdevresinin mantık sembolü Şekil-6.7'de, doğruluk tablosu ise, Tablo-6.4'tedir. Tümdevre, 4 adet giriş, 16 adet çıkış ve iki adet içeriden VE'lenmiş etkin düşükte çalışan kontrol (izin) girişine (G_1 ve G_2) sahiptir. Çıkışlar etkin düşükte çalışır. Kontrol girişlerinin her ikisine birden sıfır gelmediği durumda girişlerin durumu ne olursa olsun çıkışların tamamı "1" seviyesinde kalacaktır. Çıkışlar açık kollektör bağlantı olmadığından çıkışlara direnç bağlantısı gerekmez.



Şekil-6.7 74154 4-e-16 kodçözücü bacak bağlantısı.

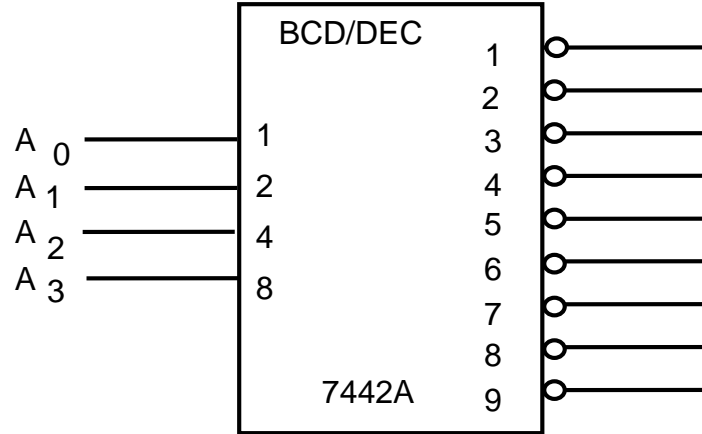
GİRİŞLER						ÇIKIŞLAR															
G_1	G_2	A	B	C	D	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Tablo-6.4 74154 tümdevresinin doğruluk tablosu.

İKO/ONLUK KODÇÖZÜCÜ

İKO/onluk kodçözücü her İKO kod sözcüğünü (8421 kodu) olası on onluk basamak

gösterimlerinden birine çevirir. *4-hattan-10-hatta* kodçözücü olarak da adlandırılır. Bu kodçözücünün iç yapısı 4-e-16 kodçözücüyle temelde aynıdır, ancak farkı 16 yerine 10 adet kodçözücü geçit kullanılmasıdır, çünkü İKO kod 0 ile 9 arasında on adet onluk basamakla gösterilmektedir. On İKO kod sözcüğünün listesi ve karşılık gelen fonksiyonları Tablo-6.5'te verilmiştir. Bu fonksiyonların her biri etkin-DÜŞÜK çıkış üretmek için VED geçitleriyle gerçekleştirilir. Etkin-YÜKSEK çıkış üretmek gerekirse, VE geçitleri kullanılır.



Şekil-6.8 7442A İKO/onluk kodçözücü.

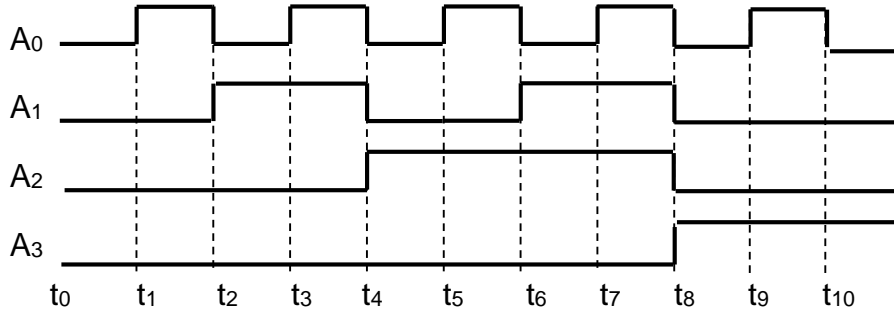
Onluk Basamak	İKO Kod				Mantık Fonksiyonu
	A ₃	A ₂	A ₁	A ₀	
0	0	0	0	0	$\overline{A_3} \overline{A_2} \overline{A_1} \overline{A_0}$
1	0	0	0	1	$\overline{A_3} \overline{A_2} \overline{A_1} A_0$
2	0	0	1	0	$\overline{A_3} \overline{A_2} A_1 \overline{A_0}$
3	0	0	1	1	$\overline{A_3} \overline{A_2} A_1 A_0$
4	0	1	0	0	$\overline{A_3} A_2 \overline{A_1} \overline{A_0}$
5	0	1	0	1	$\overline{A_3} A_2 \overline{A_1} A_0$
6	0	1	1	0	$\overline{A_3} A_2 A_1 \overline{A_0}$
7	0	1	1	1	$\overline{A_3} A_2 A_1 A_0$
8	1	0	0	0	$A_3 \overline{A_2} \overline{A_1} \overline{A_0}$
9	1	0	0	1	$A_3 \overline{A_2} \overline{A_1} A_0$

Tablo-6.5 İKO Kodçözücü fonksiyonları.

Örnek 6.2:

7442A İKO/onluk kodçözücü tümdevresidir. Mantık simgesi Şekil-6.8'de verilmiştir. Şekil-6.9'daki dalga şekilleri 7442A'nın girişlerine uygulandığında çıkış dalga şeklini

çizin.



Şekil-6.9

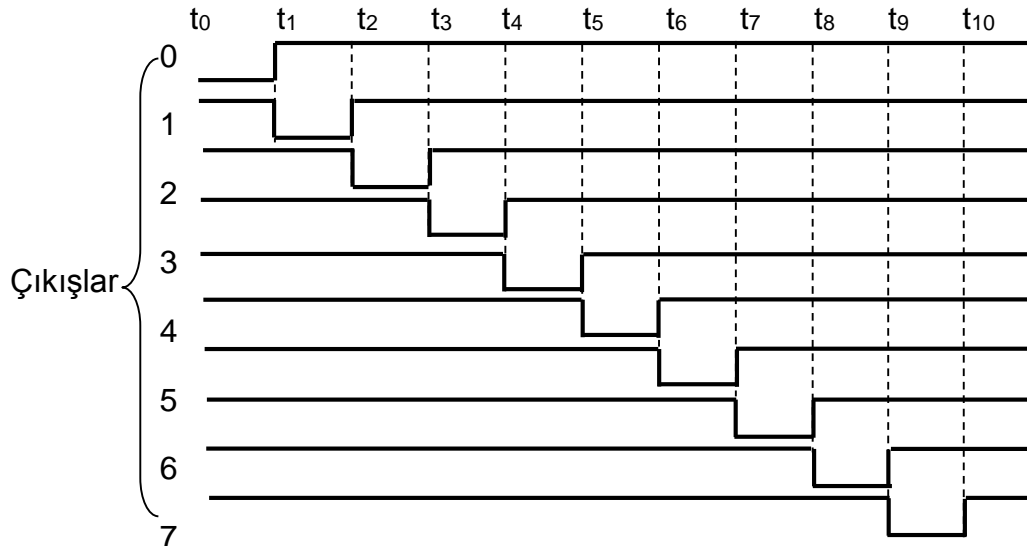
Çözüm:

Çıkış dalga şekilleri Şekil-6.10'da gösterildiği gibidir. Girişler artan sırayla İKO kodlarını ifade ettiğinden çıkışlarda da aynı sırayla karşılık gelen onluk çıkışlar etkin olmaktadır.

Kod çözücüler pratikte mikrobilgisayar sistemlerinde farklı tipteki bellek tümdevreleri kullanıldığında bu tümdevrelerin adres bilgisinin doğru bir şekilde ulaştırılması için kullanılır.

KODLAYICILAR (ENKODERS)

Kodçözücünün tersi işlem yapar. Onluk sayıları İKO'ya veya sekizli sayıları ikiliye, onaltılık sayıları ikiliye dönüştürür.



Şekil-6.10

ONLUDAN İKO'YA KODLAYICI

Bu kodlayıcının 10 adet girişi ve 4 adet çıkışı vardır. Tablo-6.6'da onluk sayıların İKO karşılıkları verilmiştir. Bu devrenin 4 adet çıkışı olacaktır, bu çıkışları başlangıçta diğerlerinden bağımsız düşünerek mantık devresini tasarlayabiliriz. A₃ çıkışının 1 olabilmesi için 8 veya 9 sayılarından biri 1 olmalıdır. Buna göre A₃ 'ün ifadesi şöyle

yazılabilir.

$$A_3=8+9$$

A_2 biti ise 4,5,6,7 sayılarının herhangi birisi girişte etkin olduğu zaman 1 olur. Diğer durumlarda sıfır olur. VEYA işlemi ile tanımlayabiliriz.

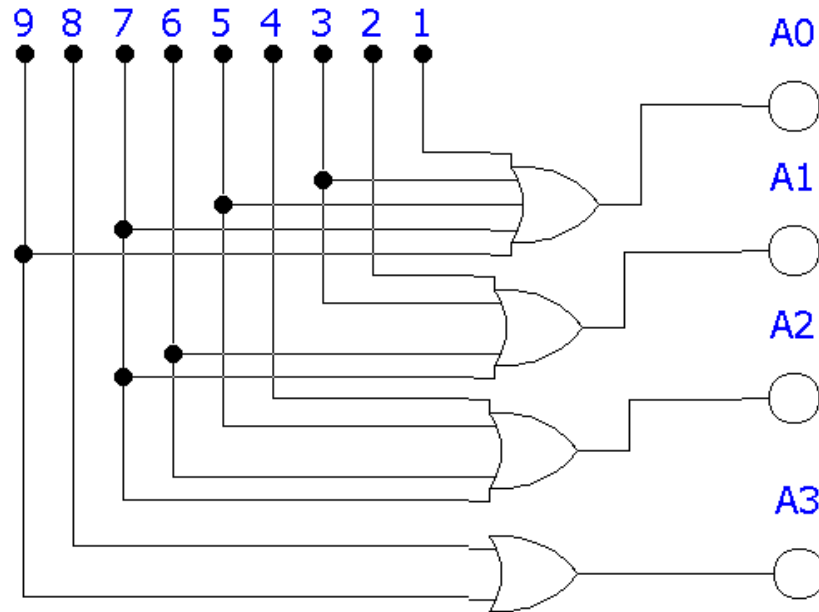
$$A_2=4+5+6+7$$

Aynı yöntemle $A_1=2+3+6+7$ ve $A_0=1+3+5+7+9$ olarak tanımlayabiliriz.

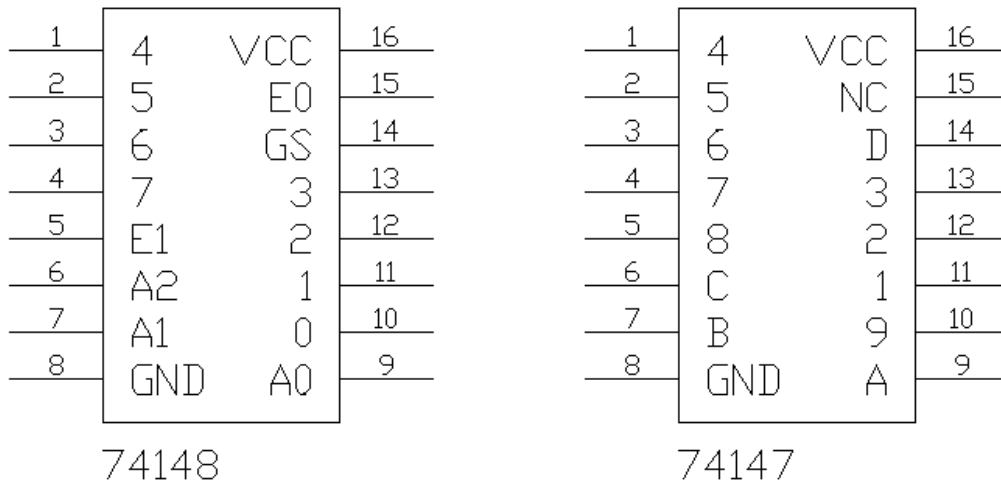
Şekil-6.11'de kodlayıcı mantık geçitleri ile gösterilmiştir. Kodlayıcıların girişi genellikle keyborda bağlıdır, kullanıcının keyboardun birden fazla tuşuna aynı anda basması durumunda çıkışların oluşmasında kargaşa oluşacaktır. Tuşlara öncelik verilmelidir, öncelik genellikle en büyük değerli girişe verilir. Onluk-İKO kodlayıcıda 9'a öncelik verildiğinde devre karmaşık hale gelecektir. Şekil-6.12'de 8 girişli ve yüksek değerli girişi öncelikli 3 bit ikilik çıkışlı 74148'in ve 10 girişli İKO çıkışlı YÜKSEK değerli girişe öncelik veren 74147'nin bacak bağlantıları verilmiştir. Girişler ve çıkışlar DÜŞÜK seviyede etkin olurlar.

	İKO KOD			
ONLU SAYI	A_3	A_2	A_1	A_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Tablo-6.6



ŞEKİL-6.11 Onluk-İKO kodlayıcı.



Şekil-6.12

BÖLÜM 7

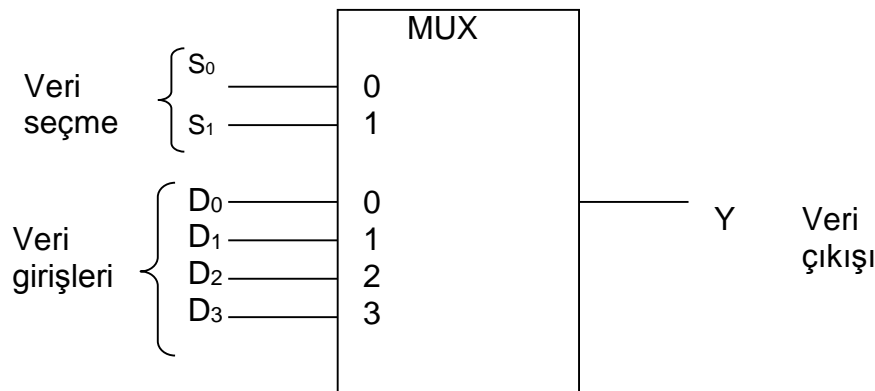
MULTİPLEXER DEMULTİPLEXER

GİRİŞ

Multiplexer veya veri seçiciler her zaman aralığında girişlerindeki verinin birini tek bir çıkışa bağlar. Demultiplexer ise bu işlemin tersini gerçekleştirir. Aynı tümdevre (dekoder) kodçözücü olarak kullanılabilir. Multiplexer kısaca MUX, demultiplexer ise DEMUX olarak adlandırılır. Multiplexer ve demultiplexer tümdevreleri kullanılarak birleşik mantık devreleri tasarlanabilir. Eşlik biti üretici/denetleyici iletişim sistemlerinde oluşan hataları denetlemek için üretilir. Sadece bir bitin değerinin değişmesini algılar. İletilecek olan veri bitleri içerisindeki "1"lerin sayısının tek ya da çift olmasına göre hata denetimi yapılır. Eşlik biti üretici/denetleyici çift (EVEN) veya tek (ODD) eşliğe göre çıkışını etkin yapar. Tümdevreler üzerinde hem tek eşlik biti için hem de çift eşlik biti için çıkış bulunur.

MULTİPLEXER (VERİ SEÇİCİLER)

Multiplexer (MUX) sayısal veriyi birkaç kaynaktan alıp iletim için ortak tek bir yola aktaran bir aygıttır. Temel bir veri seçicide birkaç veri giriş hattı ve tek bir çıkış hattı vardır. Hangi girişin çıkışa aktarılacağını belirleyen birkaç veri seçme hattı da bulunmaktadır. Dört girişli bir veri seçicinin (multiplexer) mantık simgesi Şekil-7.1'de verilmiştir. Burada iki tane veri seçme hattı vardır ki bu iki hatla dört veri hattından hangisinin çıkışa aktarılacağı seçilebilir. Şekil-7.1'e baktığımızda, veri seçme girişlerindeki iki bitlik ikilik kod seçilen veri girişindeki verinin veri çıkışına aktarılmasını sağlayacaktır.



Şekil-7.1 4-giriş 1-çıkış hatlı (4x1) veri seçici/ multiplexer için mantık simgesi.

İkilik 0 ($S_1 = 0$ ve $S_0 = 0$) veri seçme hatlarına uygulanırsa, D_0 girişindeki veri çıkışa aktarılır. İkilik 1 ($S_1 = 0$ ve $S_0 = 1$) veri seçme hatlarına uygulanırsa, D_1 girişindeki

veri çıkışa aktarılır. İkilik 2 ($S_1 = 1$ ve $S_0 = 0$) veri seçme hatlarına uygulanırsa, D_2 girişindeki veri çıkışa aktarılır. İkilik 3 ($S_1 = 1$ ve $S_0 = 1$) veri seçme hatlarına uygulanırsa, D_3 girişindeki veri çıkışa aktarılır. Bu çalışmanın özeti Tablo-7.1'dedir.

Veri Seçme Girişleri		
S_1	S_0	Seçilen veri
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Tablo-7.1 Dört girişli veri seçici için veri seçimi.

Veri seçicinin bu işlemi yapmasını sağlayan mantık devresini inceleyelim. Veri çıkışı *seçilen* veri girişinin durumuna eşittir. O zaman, veri çıkışını veri girişleri ve veri seçme girişleri cinsinden yazabilmeliyiz. Bunu şu şekilde yapabiliriz:

Yalnızca $S_1 = 0$ ve $S_0 = 0$ olduğunda, Y veri çıkışı D_0 veri girişine eşittir:

$$Y = D_0 \overline{S_1} \overline{S_0}$$

Yalnızca $S_1 = 0$ ve $S_0 = 1$ olduğunda, veri çıkışı D_1 veri girişine eşittir:

$$Y = D_1 \overline{S_1} S_0$$

Yalnızca $S_1 = 1$ ve $S_0 = 0$ olduğunda, veri çıkışı D_2 veri girişine eşittir:

$$Y = D_2 S_1 \overline{S_0}$$

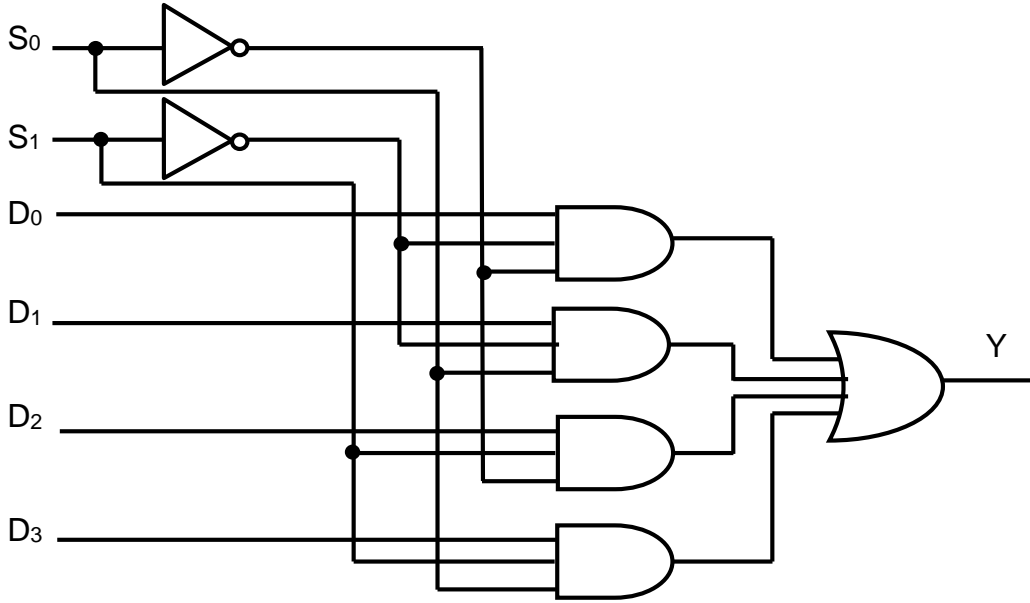
Yalnızca $S_1 = 1$ ve $S_0 = 1$ olduğunda, veri çıkışı D_3 veri girişine eşittir:

$$Y = D_3 S_1 S_0$$

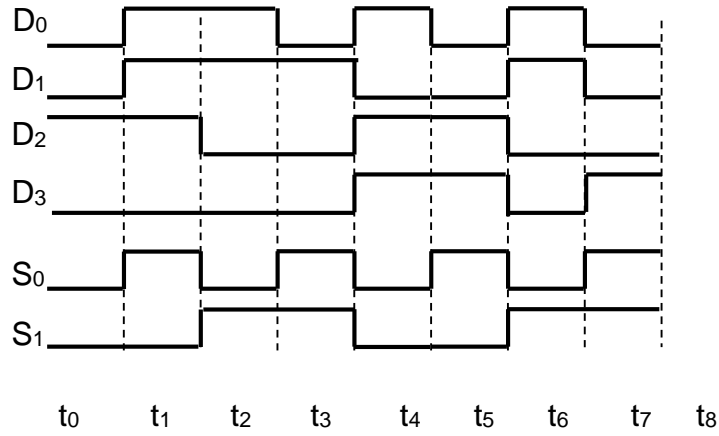
Bu terimler VEYAlandığında veri çıkışının toplam ifadesi elde edilir:

$$Y = D_0 \overline{S_1} \overline{S_0} + D_1 \overline{S_1} S_0 + D_2 S_1 \overline{S_0} + D_3 S_1 S_0$$

Bu eşitliğin mantık geçitleriyle gerçekleştirilmesi için 4 adet 3 girişli VE geçidi, 1 adet 4 girişli VEYA geçidi ve S_1 ve S_0 'ın değillerini almak için iki adet evirici gerekmektedir (Şekil-7.2). Herhangi bir giriş hattından veri seçilebildiğinden bu devreye *veri seçici* de denilmektedir.



Şekil-7.2 Dört girişli veri seçicinin mantık devresi.



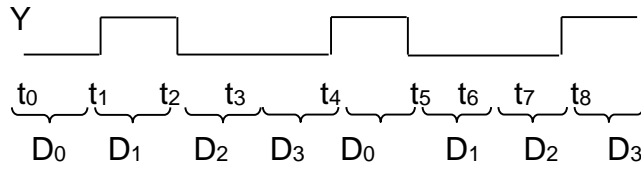
Şekil-7.3

Örnek 7.1

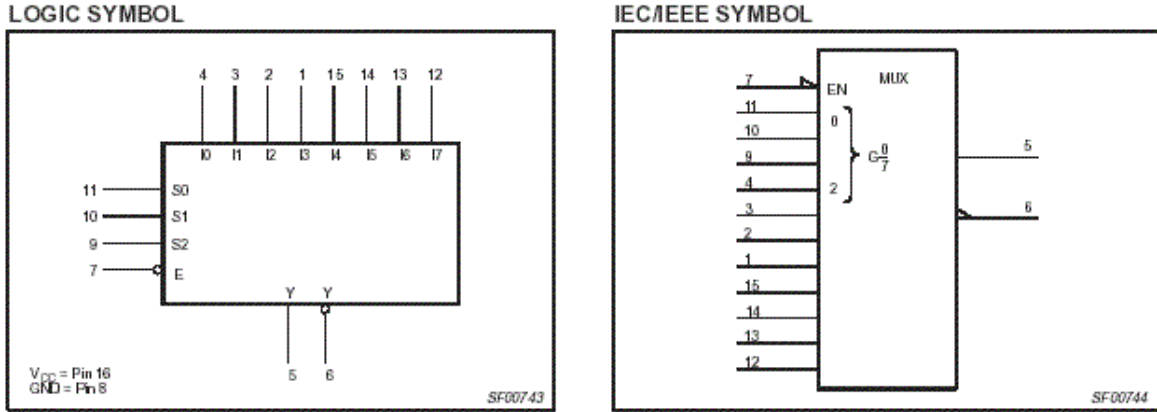
Şekil-7.3'teki veri seçici dalga şekilleri Şekil-7.2'deki veri seçicinin girişlerine uygulanmıştır. Girişlere göre çıkış dalga şeklini çizin.

Çözüm:

Her aralıkta veri seçme hatlarının ikilik durumları hangi verinin seçileceğine karar verir. Veri seçme girişlerinin 00, 01, 10, 11, 00, 01, 10, 11,... şeklinde yinelenen kodlar şeklinde devam ettiğine dikkat edin. Çıkış dalga şekli Şekil-7.4'te görülmektedir.



Şekil-7.4



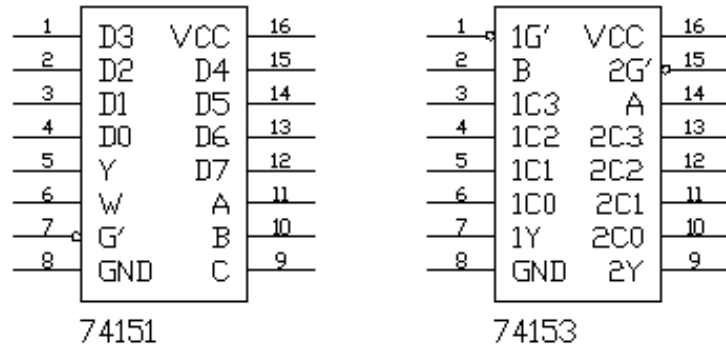
Şekil-7.5 74151 multiplexer tümdevresinin mantık gösterimi.

Şekil-7.5'te deneyde kullanılacak olan 74151 multiplexer tümdevresinin mantık gösterimi verilmiştir. 8 veri girişi ve 3 adet veri seçimi yapan denetim girişi vardır. Veri çıkışının kendisi (Y) ve değili (Y':W) tümdevre üzerinde bulunur, ayrıca izin verme girişi G' (EN: enable) vardır. Şekil-7.6'da 8-1 veri seçici içeren 74151 ve 4-1 çift veri seçici içeren 74153 tümdevrelerinin bacak bağlantıları verilmiştir. Burada D veya I olarak adlandırılan girişler veri girişleridir. A, B, C girişleri veri seçme hatlarıdır.

VERİ SEÇİCİ / MULTİPLEXER UYGULAMALARI

Mantık fonksiyon üretici Veri seçici/multiplexer'ın bir uygulama alanı toplamaların çarpımı şeklinde birleşimsel mantık fonksiyonlarının üretilmesidir. Bu şekilde kullanıldığında, aygıt ayırık geçitlerin yerini alabilir, kullanılacak tümdevrelerin sayısını azaltabilir ve tasarımdaki değişikliklerin daha kolay yapılabilmesine olanak sağlar.

74151A sekiz-girişli veri seçici/multiplexer değişkenler veri seçme girişlerine bağlanır ve her bir veri girişi doğruluk tablosundaki o fonksiyon için gerekli mantık seviyesine kurularsa, üç değişkenli herhangi bir mantık ifadesini belirtmek için kullanılabilir. Örneğin, fonksiyon değişken birleşimi $\overline{A_2} A_1 \overline{A_0}$ olduğunda "1" ise, 2 girişi (010 ile seçilen) YÜKSEK seviyeye bağlanır. Veri seçme hatlarına bu özel değişken birleşimi uygulandığında, bu 'YÜKSEK seviye' çıkışa aktarılır.



Şekil-7.6 Multiplexer tümdevreleri.

Örnek 7.2

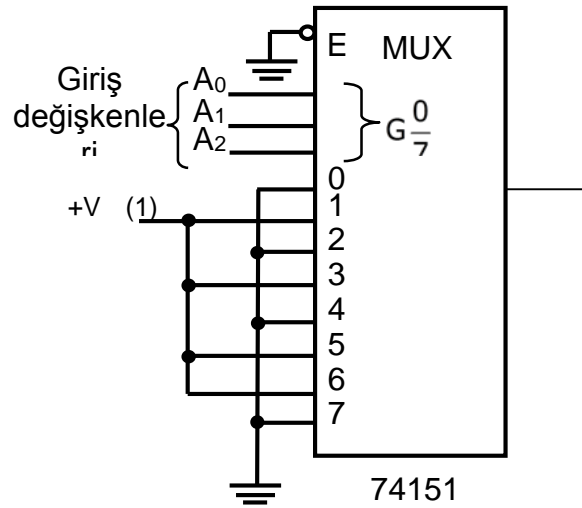
Tablo-7.2’de tanımlanan mantık fonksiyonunu 74151A multiplexer ile gerçekleyin. Bu yöntemi ayırık mantık geçitleriyle yapılan tasarım ile karşılaştırın.

Çözüm:

Doğruluk tablosundan Y’nin 001, 011, 101 ve 110 giriş değişken birleşimleri için “1” olduğunu görüyoruz. Diğer bütün birleşimler için Y “0”dır. Bu fonksiyonun veri seçici ile tasarlanabilmesi için, yukarıda belirtilen her bir birleşim tarafından seçilen veri girişinin YÜKSEK seviyeye bağlanması gerekir. Şekil-7.7’de gösterildiği gibi diğer bütün veri girişleri DÜŞÜK seviyeye bağlanmalıdır. Bu fonksiyonun mantık geçitleriyle tasarlanması için dört tane 3-girişli VE geçidi, bir tane 4-girişli VEYA geçidi ve üç eviriciye gerek vardır.

Girişler			Çıkış
A ₂	A ₁	A ₀	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

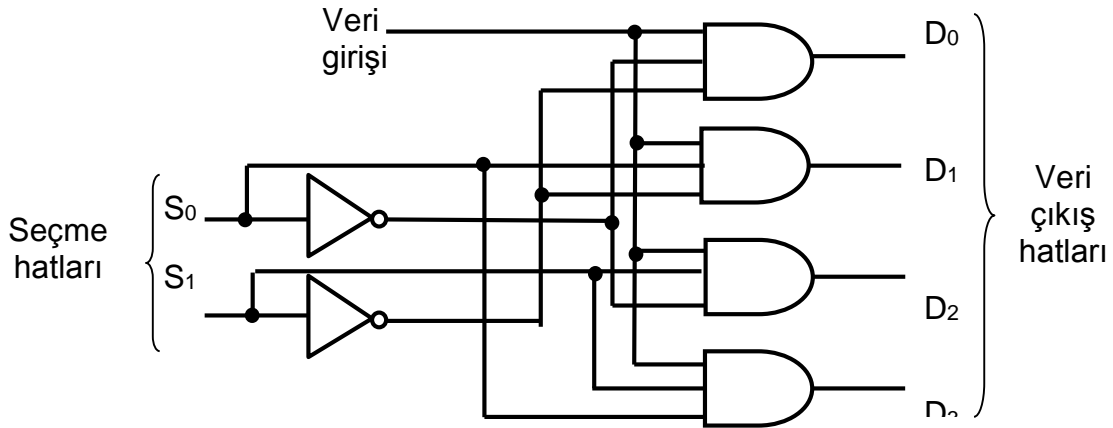
Tablo-7.2



Şekil-7.7 Üç-değişkenli fonksiyon üretici olarak veri seçici.

DEMÜLTİPLEXER

Demultiplexer (DMUX) temel olarak multiplexer'ın yaptığı işin tersini yapar. Bir hattan veriyi alır ve belirli sayıdaki çıkış hatlarından birine aktarır. Bu nedenle, bu ayyığa veri çoklayıcı yada veri dağıtıcı denilir.

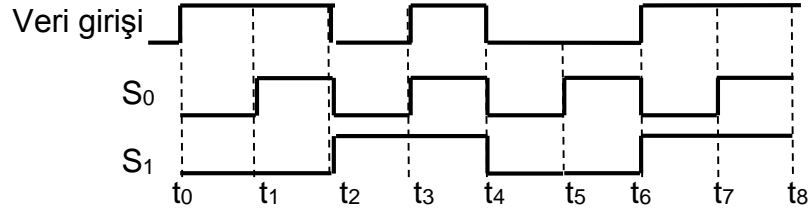


Şekil-7.8 1-hattan-4-hatta demultiplexer

Şekil-7.8'de 1-hattan-4-hatta veri dağıtıcı devresi görölmektedir. Veri-giriş hattı tüm VE geçitlerinin girişlerine gitmektedir. İki veri-seçme hattı her seferinde yalnız bir geçide izin vermektedir ve veri-giriş hattındaki veri seçilen geçitten ilgili veri-çıkış hattına aktarılır.

Örnek 7.3

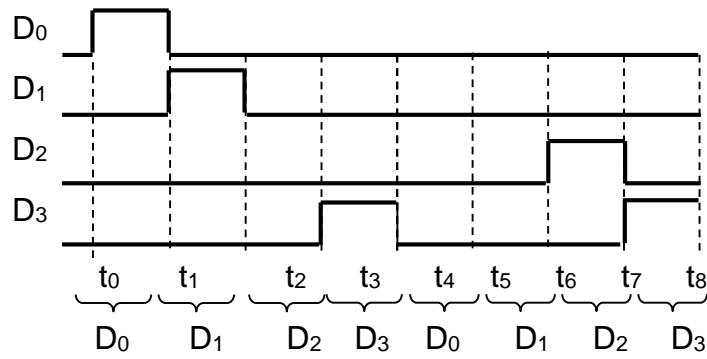
Seri veri-giriş dalga şekli ve veri-seçme girişleri Şekil-7.9'da verilmiştir. Şekil-7.8'deki demultiplexer için veri-çıkış dalga şeklini çizin



Şekil-7.9

Çözüm:

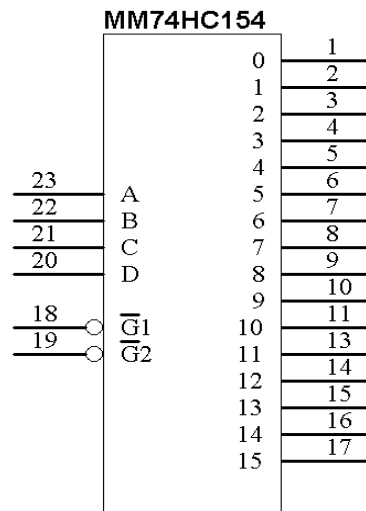
Seçme hatları ikilik sayma düzeninde gitmektedir, dolayısıyla birbirini izleyen her bir giriş Şekil-7-10'daki gibi sırayla D₀, D₁, D₂, D₃ 'e yönlendirilir.



Şekil-7.10

74154 'ÜN DEMULTİPLEXER OLARAK KULLANILMASI

74154'ü 4-hattan-16-hatta kodçözücü olarak kullanmıştık. Bu aygıt ve diğer kodçözücüler demultiplexer uygulamalarında da kullanılabilir. Demultiplexer olarak kullanıldığında bu tümdevrenin mantık simgesi Şekil-7-11'de gösterilmektedir.



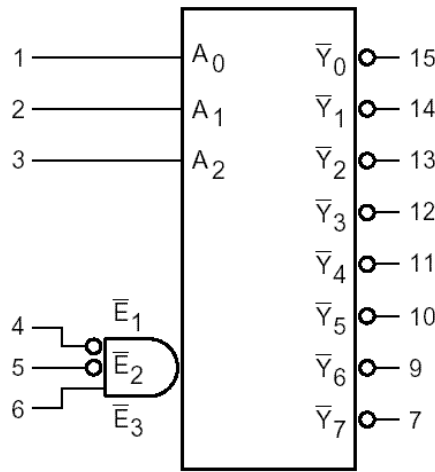
Şekil-7.11 74154'ün demultiplexer olarak kullanılması.

Demultiplexer uygulamalarında giriş hatları veri seçme hatları olarak kullanılır. İzin girişlerinden biri veri-giriş hattı olarak kullanılırken diğer izin girişi de DÜŞÜK tutularak

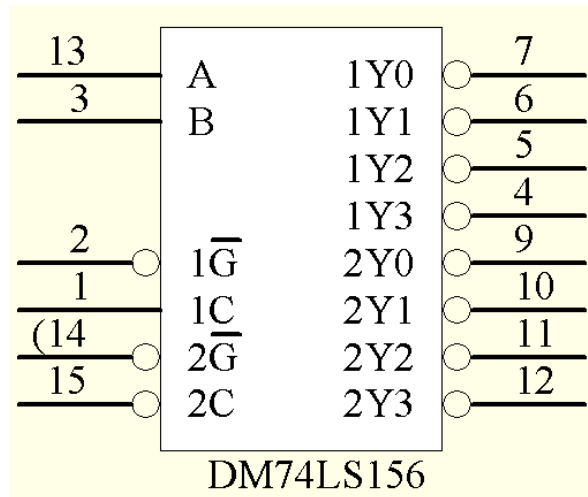
negatif-VE geçidine izin verilir.

74156 VE 74138 'İN DEMULTİPLEXER OLARAK KULLANILMASI

Demultiplexer olarak 74156 ve 74138 tüm devresi kullanılmaktadır. Kodçözücü olarak da kullanılan bu tüm devrelerin mantık gösterimleri Şekil-7.12 ve 13'tedir. 74156 tümdevresi dual 4×1 demultiplexer tümdevresidir. 1C girişi birinci veri girişi, 2C ikinci veri girişi olarak, 1G girişi birinci izin girişi 2G girişi ikinci izin girişi olarak kullanılır. Denetim girişleri A_0 ve A_1 her iki demultiplexer için ortaktır. 74138 tümdevresi 1×8 demultiplexer olarak kullanılabilir, daha önce bu tümdevre kodçözücü olarak kullanılmıştı. Demultiplexer olarak kullanmak için E_3 girişi veri girişi olarak kullanılır, E_1 , E_2 girişleri izin girişleri olarak kullanılır. A, B, C girişleri denetim için kullanılır. Çıkışlar etkin DÜŞÜKte çalışır.



Şekil-7.13 74138 demultiplexer tümdevresinin mantık gösterimi.



Şekil-7.12 74156 demultiplexer tümdevresinin mantık gösterimi

EŞLİK BİTİ ÜRETECİ

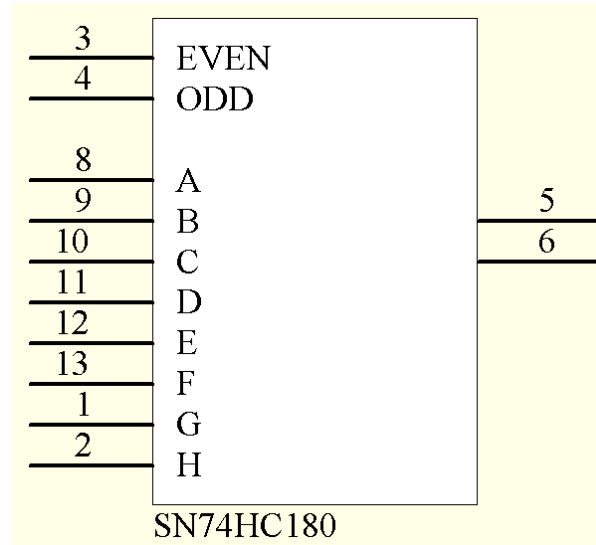
Eşlik biti üretici/denetleyicisi olarak 74180 tümdevresi kullanılacaktır. Bu tümdevrenin mantık gösterimi Şekil-7.14'te, doğruluk tablosu da Tablo-7.3'te verilmiştir. Girişteki tek ve çift girişleri daha önceki üretcin çıkışından gelen eşlik

biti içindir. Bu tümdevreler veri iletişim sistemlerinde kullanılır.

Girişler			Çıkışlar	
Girişteki 1'lerin sayısı	Tek	Çift	Σ Tek	Σ Çift
Çift	1	0	1	0
Tek	1	0	0	1
Çift	0	1	0	1
Tek	0	1	1	0
X	1	1	0	0
X	0	0	1	1

Tablo-7.3 74180 tümdevresinin doğruluk tablosu

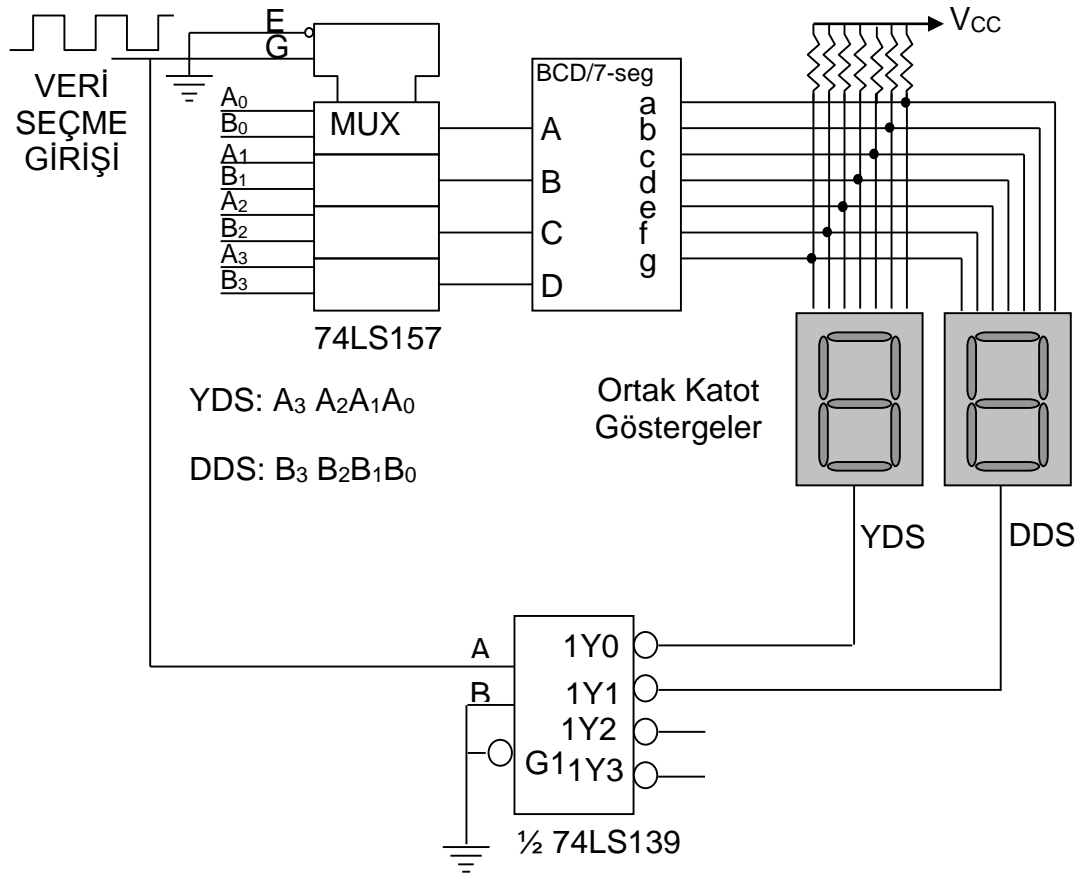
Eğer girişteki tek ve çift girişleri kullanılmıyorsa kullanılan tek veya çift eşlik bitine bağlı olarak uygun bağlantı yapılmalıdır. İletişim sistemlerinde verici kısmında bu girişlerin her ikisi de kullanılmaz, alıcı kısmında ise seçilen eşlik bitine bağlı olarak tek veya çift girişlerden biri kullanılır diğerine tabloya göre uygun bağlantı yapılır.



Şekil-7.14

MULTİPLEXER UYGULAMALARI

Şekil-7.15’de multiplexer ve dekoder uygulaması gösterilmiştir. Kodçözücü/sürücü tümdevreleri genellikle bu şekilde kullanılır. Bu devrede iki gösterge zaman paylaşımli olarak aynı Kodçözücü/sürücü tümdevre kullanılarak sürülmektedir. İki veya daha fazla gösterge bu yöntemle sürülebilir. 74139’un yarısı kullanılmıştır, sayıların katotlarını sürmek için kullanılır. Multiplexer ise gelen A verisini YDS’da B verisini DDS’da gösterir.



Şekil-7.5 Multiplexer uygulama devresi.

1. kadardır?

TÜMDEVRE VERİ YAPRAKLARI

GİRİŞ

Bu bölümde derste ve deneyde kullanılan tümdevre veri yapraklarının özeti verilmiştir. Ayrıntılı bilgi

<http://ics.nxp.com/products/gates/>

http://www.ti.com/lscs/ti/logic/logic.page?DCMP=TIHomeTracking&HQS=Other+O+T+home_p_logic

adreslerinden alınabilir.

7400 DÖRTLÜ İKİ GİRİŞLİ VED GEÇİDİ

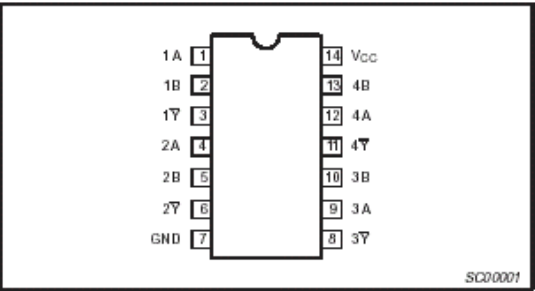
Quad 2-input NAND gate	74ALS00A
------------------------	----------

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS00A	4.0ns	1.0mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^{\circ}C$ to $+70^{\circ}C$	
14-pin plastic DIP	74ALS00AN	SOT27-1
14-pin plastic SO	74ALS00AD	SOT108-1
14-pin plastic SSOP Type II	74ALS00ADB	SOT337-1

PIN CONFIGURATION

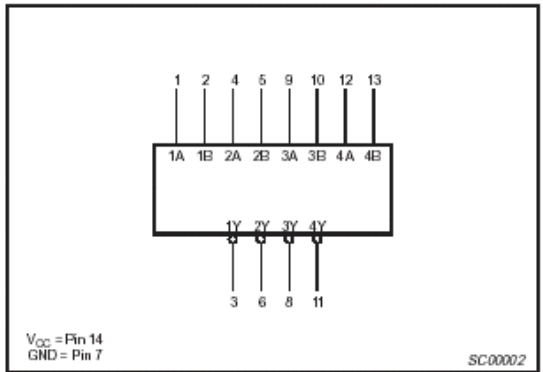


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

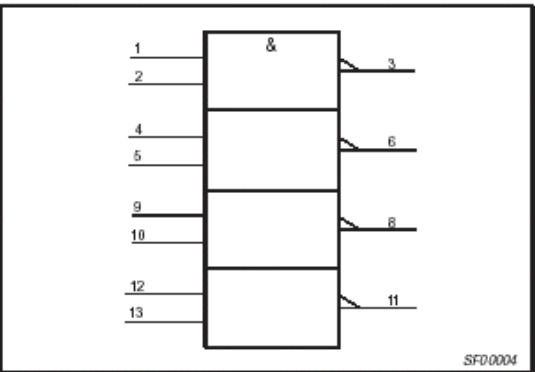
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA, nB	Data inputs	1.0/1.0	20 μ A/0.1mA
nY	Data output	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

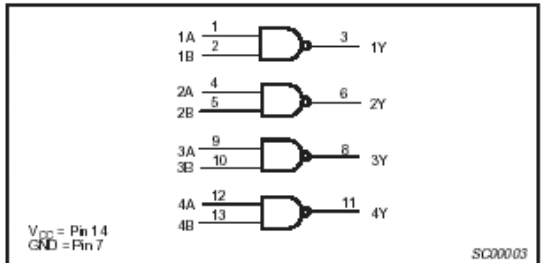
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

INPUTS		OUTPUT
nA	nB	nY
H	H	L
L	X	H
X	L	H

H = High voltage level
L = Low voltage level
X = Don't care

4011 DÖRTLÜ İKİ GİRİŞLİ CMOS VED GEÇİDİ

Quadruple 2-input NAND gate

HEF4011B gates

DESCRIPTION

The HEF4011B provides the positive quadruple 2-input NAND function. The outputs are fully buffered for highest noise immunity and pattern insensitivity of output impedance.

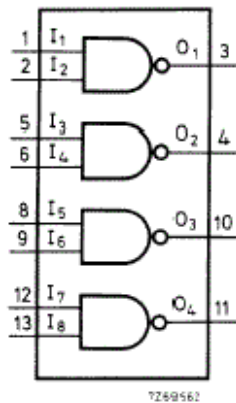


Fig.1 Functional diagram.

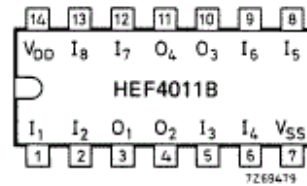


Fig.2 Pinning diagram.

- HEF4011BP(N): 14-lead DIL; plastic
(SOT27-1)
- HEF4011BD(F): 14-lead DIL; ceramic (cerdip)
(SOT73)
- HEF4011BT(D): 14-lead SO; plastic
(SOT108-1)
- (): Package Designator North America

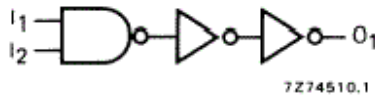


Fig.3 Logic diagram (one gate).

74LS02 DÖRTLÜ İKİ GİRİŞLİ VEYAD GEÇİDİ

Quad 2-input NOR gate

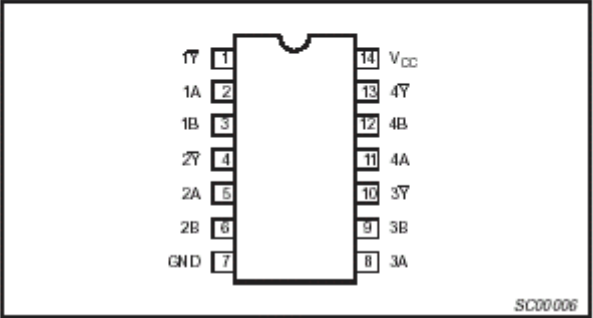
74ALS02

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS02	4.0ns	1.0mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE V _{CC} = 5V ±10%, T _{amb} = 0°C to +70°C	
14-pin plastic DIP	74ALS02N	SOT27-1
14-pin plastic SO	74ALS02D	SOT108-1

PIN CONFIGURATION

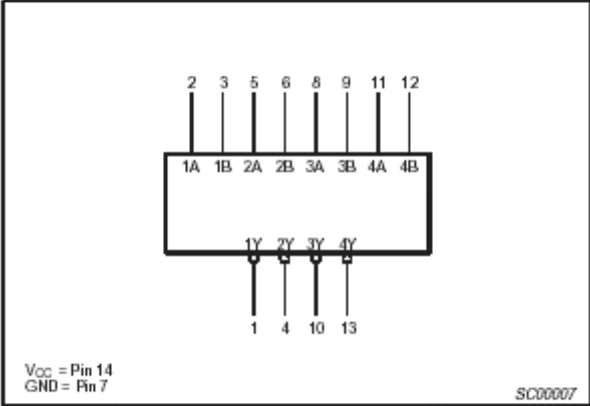


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

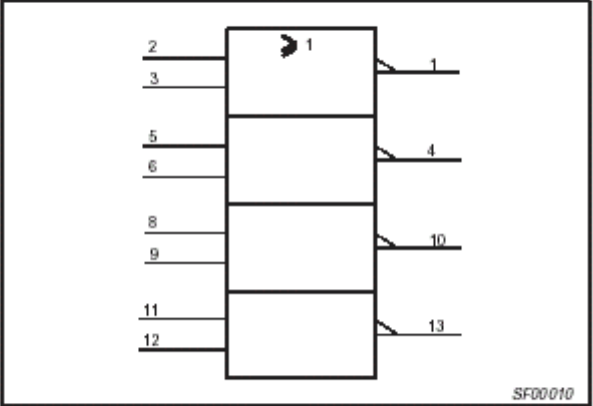
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA, nB	Data Inputs	1.0/1.0	20µA/0.1mA
nY	Data output	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20µA in the High state and 0.1mA in the Low state.

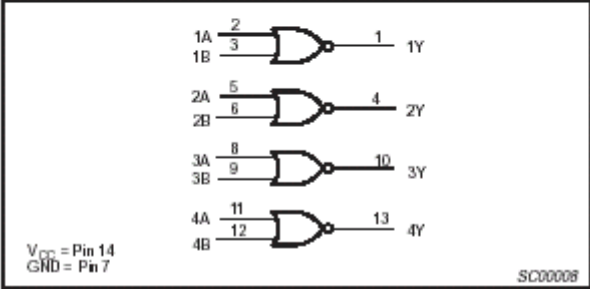
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

INPUTS		OUTPUT
nA	nB	nY
H	H	L
L	X	H
X	L	H

H = High voltage level
L = Low voltage level
X = Don't care

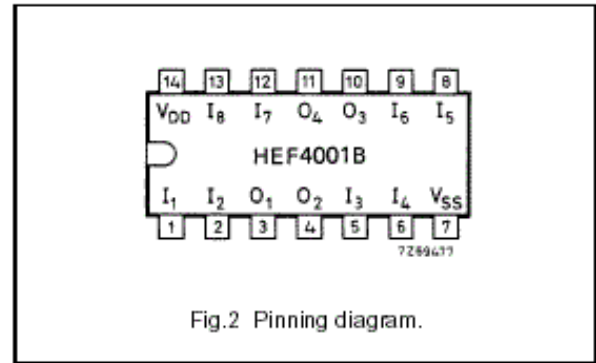
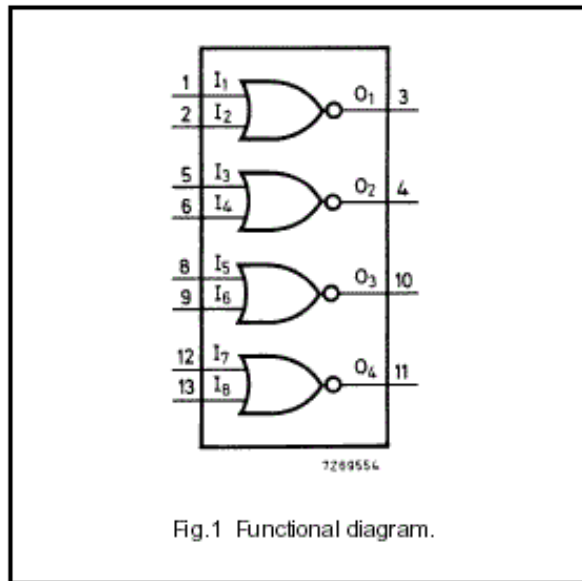
4001 DÖRTLÜ İKİ GİRİŞLİ CMOS VEYAD GEÇİDİ

Quadruple 2-input NOR gate

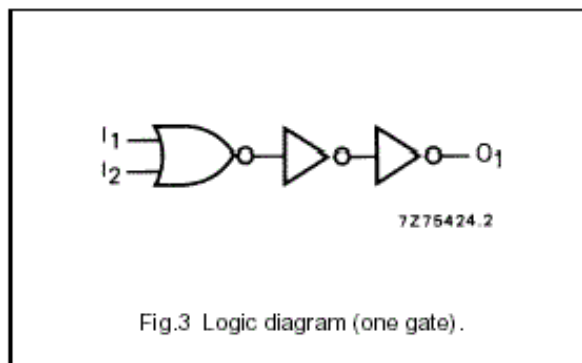
HEF4001B gates

DESCRIPTION

The HEF4001B provides the positive quadruple 2-input NOR function. The outputs are fully buffered for highest noise immunity and pattern insensitivity of output impedance.



- HEF4001BP(N): 14-lead DIL; plastic (SOT27-1)
- HEF4001BD(F): 14-lead DIL; ceramic (cerdip) (SOT73)
- HEF4001BT(D): 14-lead SO; plastic (SOT108-1)
- (): Package Designator North America



74LS04 ALTILI DEĞİL GEÇİDİ

Hex inverter

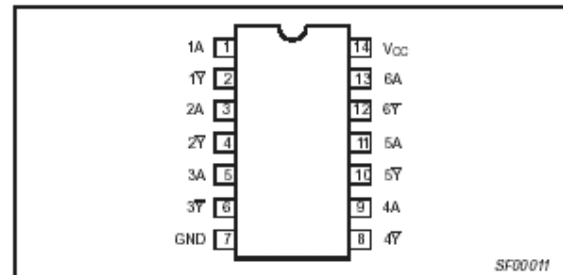
74ALS04B

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS04B	3.5ns	2.0mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^{\circ}C$ to $+70^{\circ}C$	
14-pin plastic DIP	74ALS04BN	SOT27-1
14-pin plastic SO	74ALS04BD	SOT108-1
14-pin plastic SSOP Type II	74ALS04BDB	SOT337-1

PIN CONFIGURATION

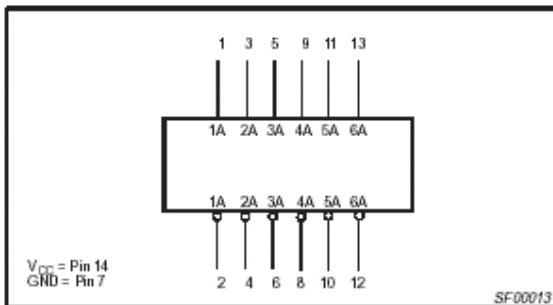


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

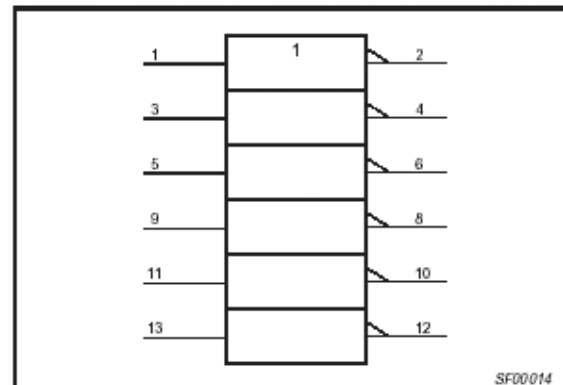
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA	Data Input	1.0/1.0	20 μ A/0.1mA
nY	Data output	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

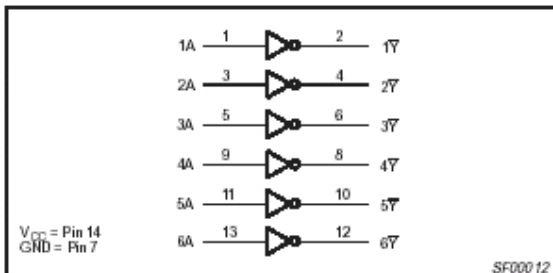
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

INPUT	OUTPUT
nA	nY
L	H
H	L

H = High voltage level
L = Low voltage level

74LS08 DÖRTLÜ İKİ GİRİŞLİ VE GEÇİDİ

Quad 2-input AND gate

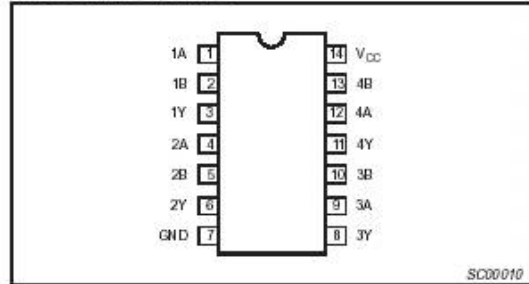
74ALS08

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS08	5.0ns	1.8mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^{\circ}C$ to $+70^{\circ}C$	
14-pin plastic DIP	74ALS08N	SOT27-1
14-pin plastic SO	74ALS08D	SOT108-1
14-pin plastic SSOP Type II	74ALS08DB	SOT337-1

PIN CONFIGURATION

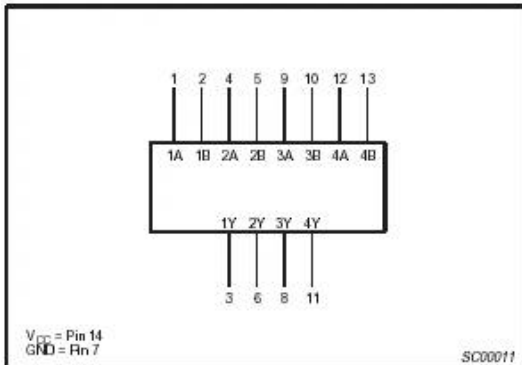


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

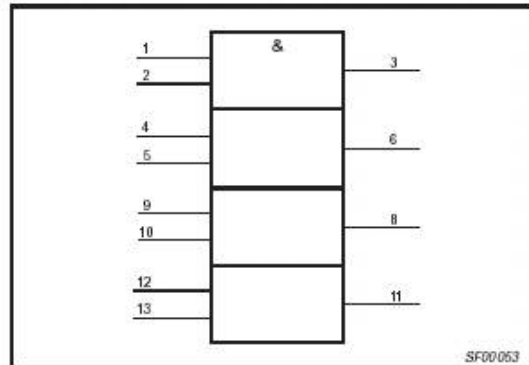
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA, nB	Data inputs	1.0/1.0	20 μ A/0.1mA
nY	Data outputs	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

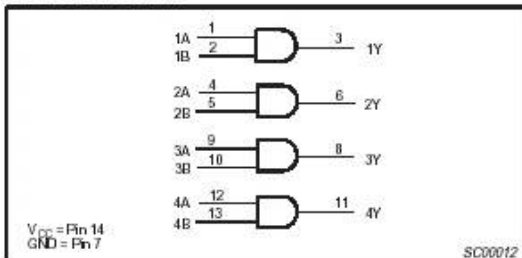
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

INPUTS		OUTPUT
nA	nB	nY
H	H	L
L	X	H
X	L	H

H = High voltage level
L = Low voltage level
X = Don't care

74LS20 İKİLİ DÖRT GİRİŞLİ VED GEÇİDİ

Dual 4-input NAND gate

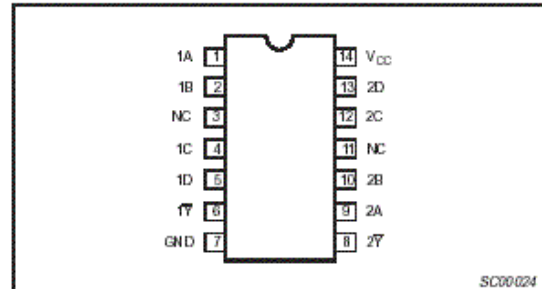
74ALS20A

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS20A	4.5ns	0.65mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^{\circ}C$ to $+70^{\circ}C$	
14-pin plastic DIP	74ALS20AN	SOT27-1
14-pin plastic SO	74ALS20AD	SOT108-1

PIN CONFIGURATION

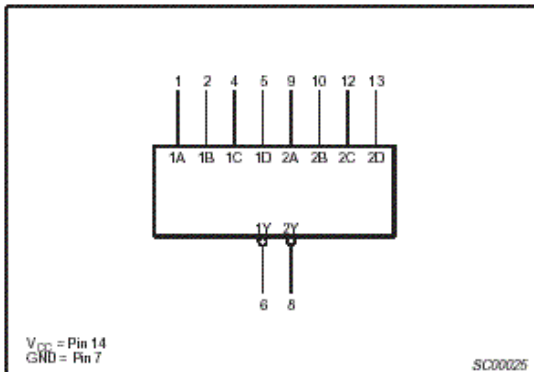


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

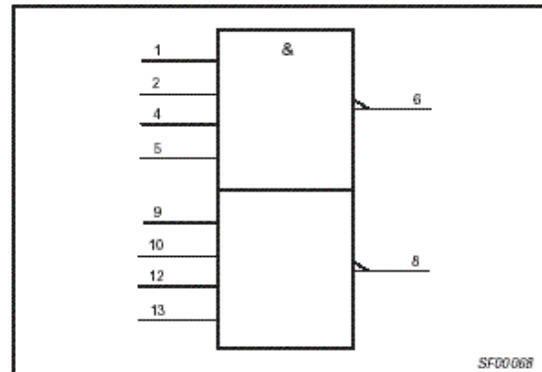
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA, nB, nC, nD	Data inputs	1.0/1.0	20 μ A/0.1mA
nY	Data outputs	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

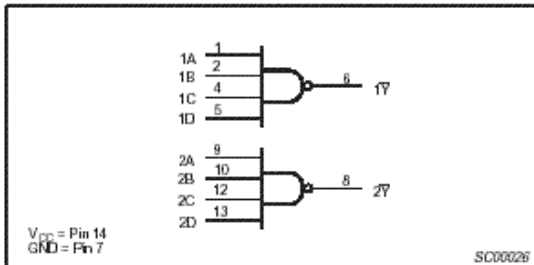
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

INPUTS				OUTPUT
nA	nB	nC	nD	nY
H	H	H	H	L
L	X	X	X	H
X	L	X	X	H
X	X	L	X	H
X	X	X	L	H

H = High voltage level
L = Low voltage level
X = Don't care

74LS32 DÖRTLÜ İKİ GİRİŞLİ VEYA GEÇİDİ

Quad 2-input OR gate

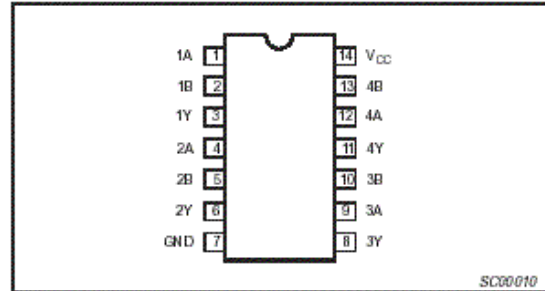
74ALS32

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS32	5.0ns	2.3mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^\circ\text{C to } +70^\circ\text{C}$	
14-pin plastic DIP	74ALS32N	SOT27-1
14-pin plastic SO	74ALS32D	SOT108-1
14-pin plastic SSOP Type II	74ALS32DB	SOT337-1

PIN CONFIGURATION

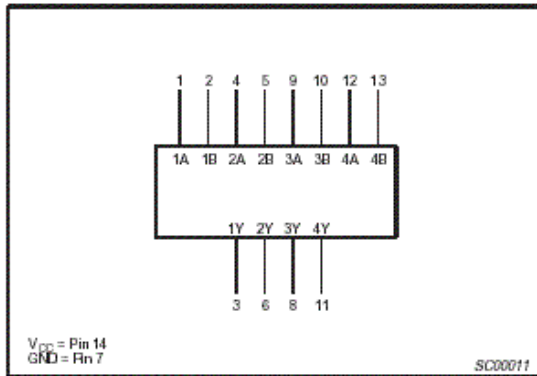


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

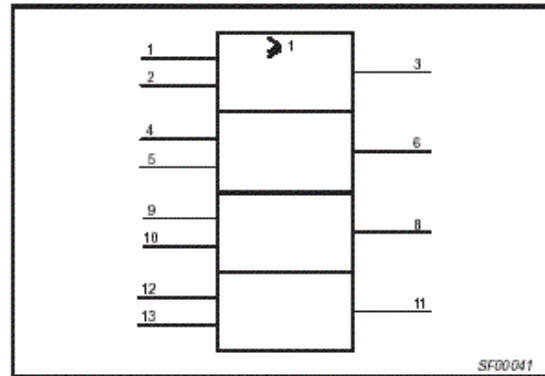
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA, nB	Data inputs	1.0/1.0	20 μ A/0.1mA
nY	Data output	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

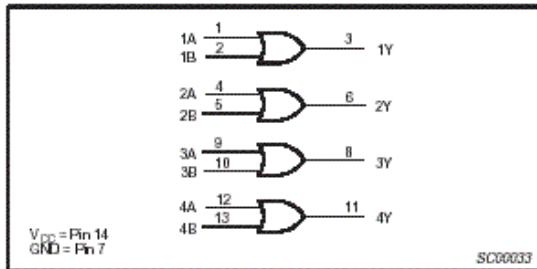
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

INPUTS		OUTPUT
nA	nB	nY
H	X	H
X	H	H
L	L	L

H = High voltage level
L = Low voltage level
X = Don't care

74LS86 DÖRTLÜ EXOR GEÇİDİ

Quad 2-input exclusive-OR gate

74ALS86

DESCRIPTION

The 74ALS86 contain four independent 2-input Exclusive-OR gates. A common application is a true/complement element. If one input is held Low, the signal on the other input will be reproduced in true form at the output. If one input is held High, the signal on the other input will be reproduced inverted at the output.

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS86	6.0ns	3.9mA

ORDERING INFORMATION

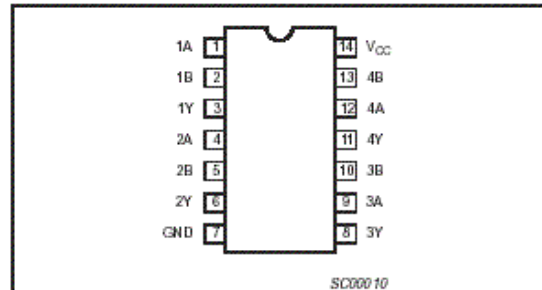
DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^{\circ}C \text{ to } +70^{\circ}C$	
14-pin plastic DIP	74ALS86N	SOT27-1
14-pin plastic SO	74ALS86D	SOT108-1

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

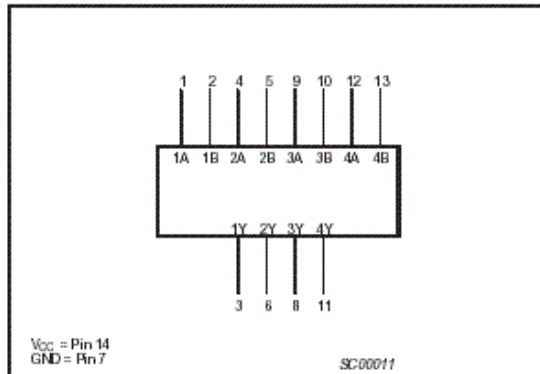
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
nA, nB	Data inputs	1.0/1.0	20 μ A/0.1mA
nY	Data outputs	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

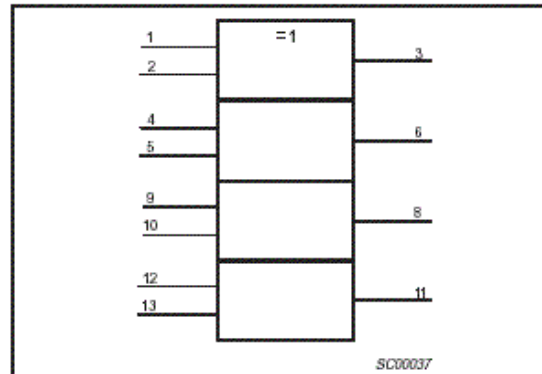
PIN CONFIGURATION



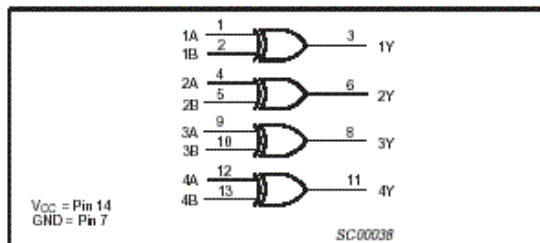
LOGIC SYMBOL



IEC/IEEE SYMBOL



LOGIC DIAGRAM



FUNCTION TABLE

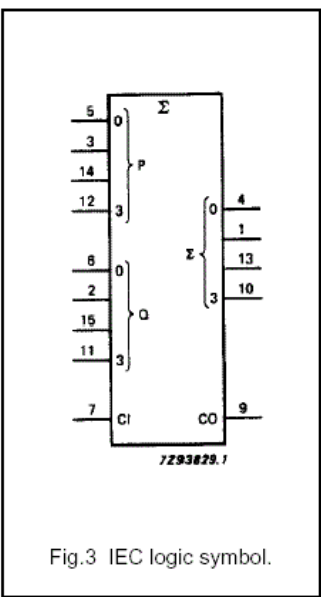
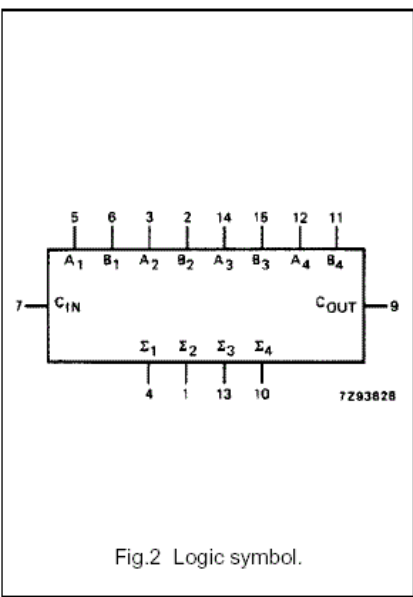
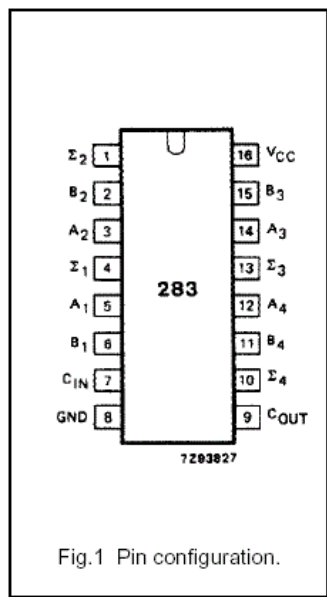
INPUTS		OUTPUT
nA	nB	nY
L	L	L
L	H	H
H	L	H
H	H	L

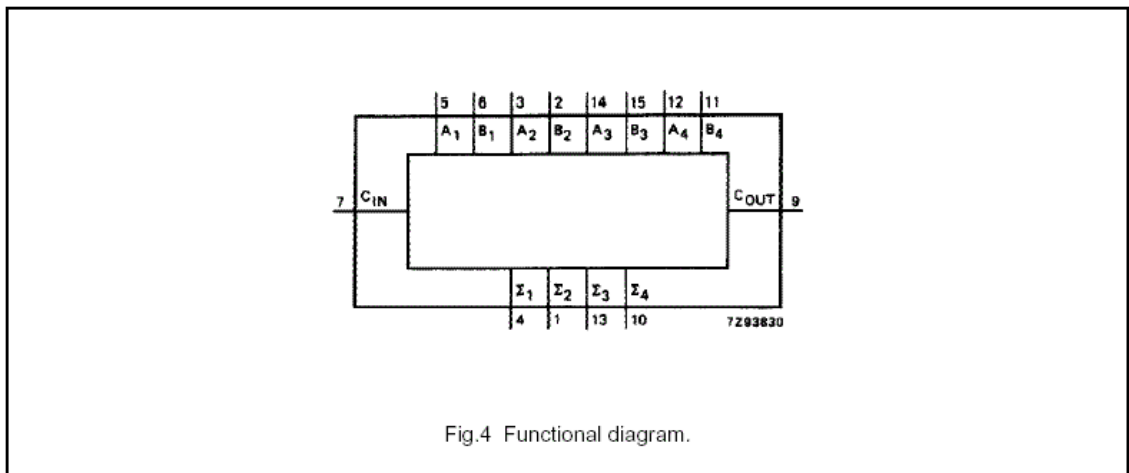
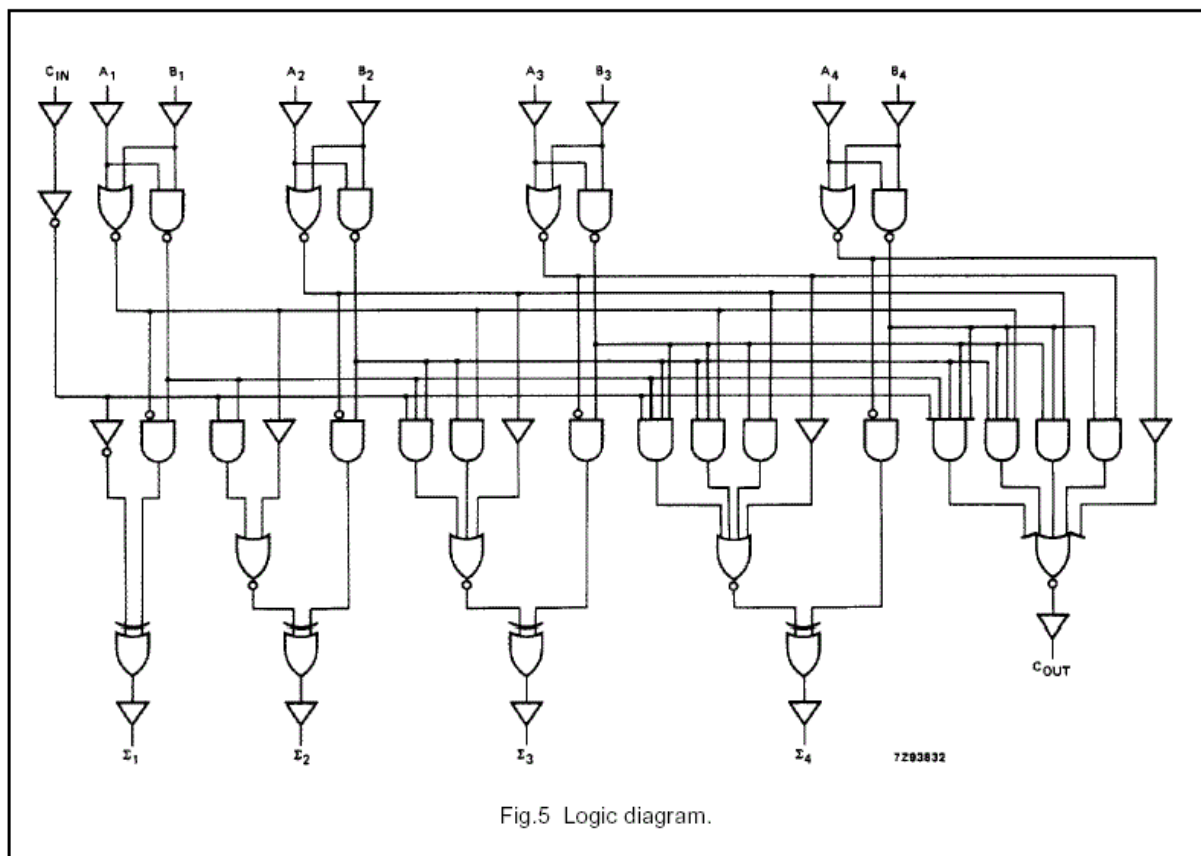
H = High voltage level
L = Low voltage level

74LS283 (74LS83) 4 BİT TAM TOPLAYICI

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
4, 1, 13, 10	Σ_1 to Σ_4	sum outputs
5, 3, 14, 12	A_1 to A_4	A operand inputs
6, 2, 15, 11	B_1 to B_4	B operand inputs
7	C_{IN}	carry input
8	GND	ground (0 V)
9	C_{OUT}	carry output
16	V_{CC}	positive supply voltage





FUNCTION TABLE

PINS	C _{IN}	A ₁	A ₂	A ₃	A ₄	B ₁	B ₂	B ₃	B ₄	Σ ₁	Σ ₂	Σ ₃	Σ ₄	C _{OUT}	EXAMPLE ⁽²⁾
logic levels	L	L	H	L	H	H	L	L	H	H	H	L	L	H	
active HIGH	0	0	1	0	1	1	0	0	1	1	1	0	0	1	(3)
active LOW	1	1	0	1	0	0	1	1	0	0	0	1	1	0	(4)

Note

1. H = HIGH voltage level
L = LOW voltage level
2. example
1001
1010

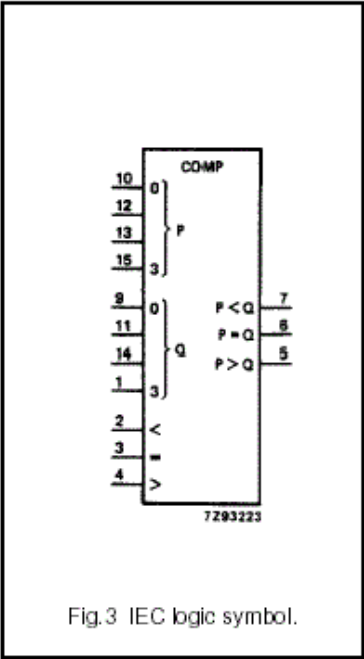
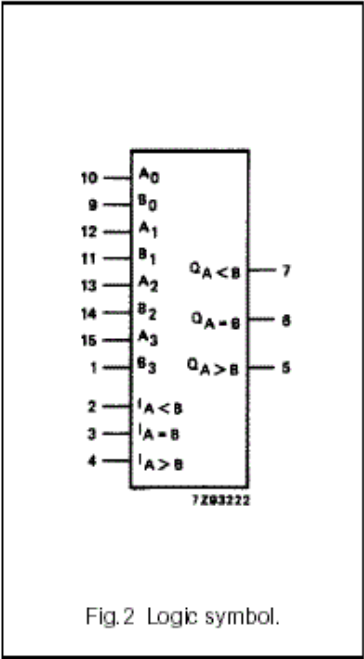
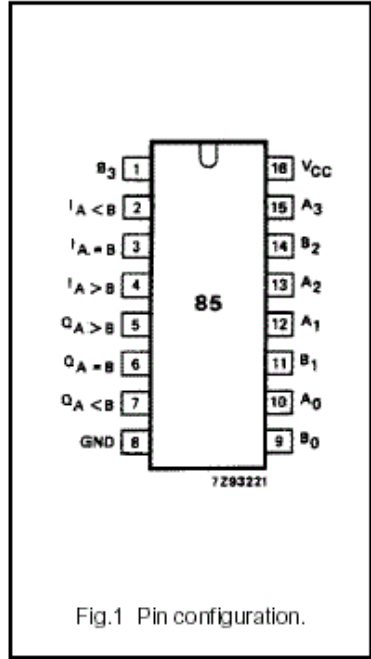
10011

74HC85 4 BİT BÜYÜKLÜK KARŞILAŞTIRICI

4-bit magnitude comparator	74HC/HCT85
----------------------------	------------

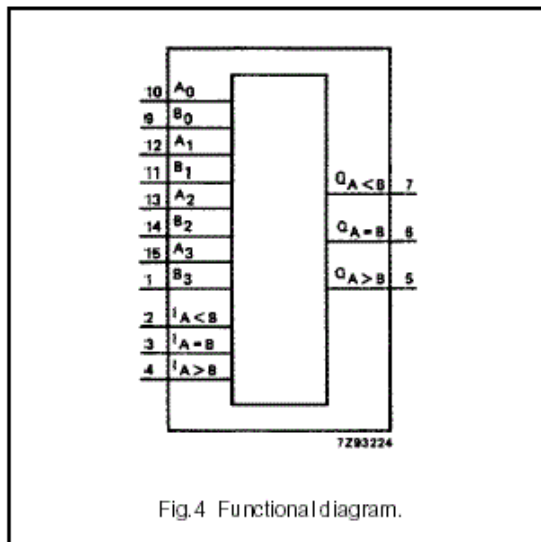
PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
2	$I_{A<B}$	A < B expansion input
3	$I_{A=B}$	A = B expansion input
4	$I_{A>B}$	A > B expansion input
5	$Q_{A>B}$	A > B output
6	$Q_{A=B}$	A = B output
7	$Q_{A<B}$	A < B output
8	GND	ground (0 V)
9, 11, 14, 1,	B_0 to B_3	word B inputs
10, 12, 13, 15	A_0 to A_3	word A inputs
16	V_{CC}	positive supply voltage



4-bit magnitude comparator

74HC/HCT85



APPLICATIONS

- Process controllers
- Servo-motor control

FUNCTION TABLE

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
A ₃ , B ₃	A ₂ , B ₂	A ₁ , B ₁	A ₀ , B ₀	I _{A>B}	I _{A<B}	I _{A=B}	Q _{A>B}	Q _{A<B}	Q _{A=B}
A ₃ >B ₃	X	X	X	X	X	X	H	L	L
A ₃ <B ₃	X	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ >B ₂	X	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ <B ₂	X	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ >B ₁	X	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ <B ₁	X	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ >B ₀	X	X	X	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ <B ₀	X	X	X	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	L	L	H	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	H	L	L	H	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	X	X	H	L	L	H
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	H	H	L	L	L	L
A ₃ =B ₃	A ₂ =B ₂	A ₁ =B ₁	A ₀ =B ₀	L	L	L	H	H	L

Notes

1. H = HIGH voltage level
L = LOW voltage level
X = don't care

74LS138 3-8 KODÇÖZÜCÜ/VERİ DAĞITICI

1-of-8 decoder/demultiplexer

74ALS138

FEATURES

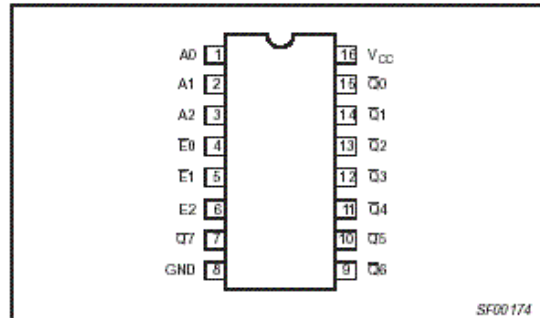
- Demultiplexing capability
- Multiple Input enable for easy expansion
- Ideal for memory chip select decoding

DESCRIPTION

The 74ALS138 decoder accepts three binary weighted inputs (A0, A1, A2) and when enabled, provides eight mutually exclusive, active-Low outputs ($\overline{Q}0 - \overline{Q}7$). The device features three Enable inputs; two active-Low ($\overline{E}0, \overline{E}1$) and one active-High (E2). Every output will be High unless $\overline{E}0$ and $\overline{E}1$ are Low and E2 is High. This multiple enable function allows easy parallel expansion of the device to 1-of-32 (5 lines to 32 lines) decoder with just four 74ALS138s and one inverter. The device can be used as an eight output demultiplexer by using one of the active-Low Enable Inputs as the data input and the remaining Enable Inputs as strobes. Enable inputs not used must be permanently tied to their appropriate active-High or active-Low state.

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS138	12.0ns	4.0mA

PIN CONFIGURATION



SF00174

ORDERING INFORMATION

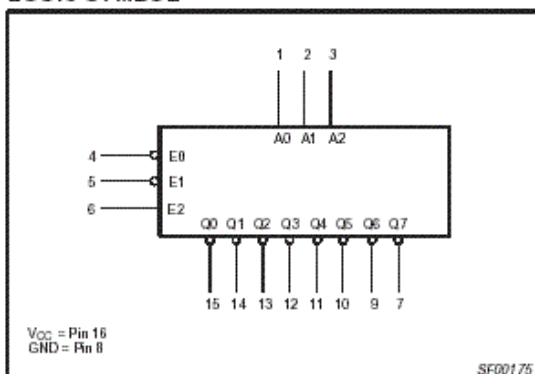
DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^\circ C \text{ to } +70^\circ C$	
16-pin plastic DIP	74ALS138N	SOT38-4
16-pin plastic SO	74ALS138D	SOT109-1

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
A0 – A2	Address Inputs	1.0/1.0	20 μ A/0.1mA
$\overline{E}0, \overline{E}1$	Enable Inputs (active-Low)	1.0/1.0	20 μ A/0.1mA
E2	Enable Input (active-High)	1.0/1.0	20 μ A/0.1mA
$\overline{Q}0 - \overline{Q}7$	Data outputs (active-Low)	50/33	1.0mA/20mA

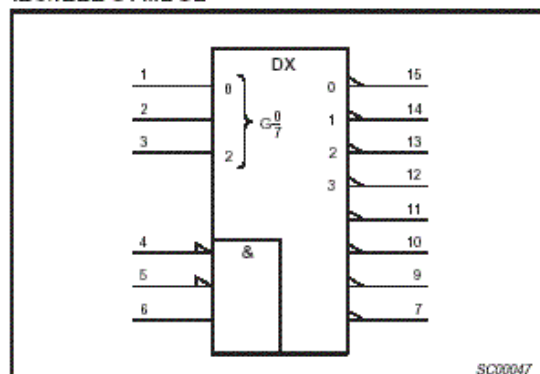
NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

LOGIC SYMBOL



SF00175

IEC/IEEE SYMBOL

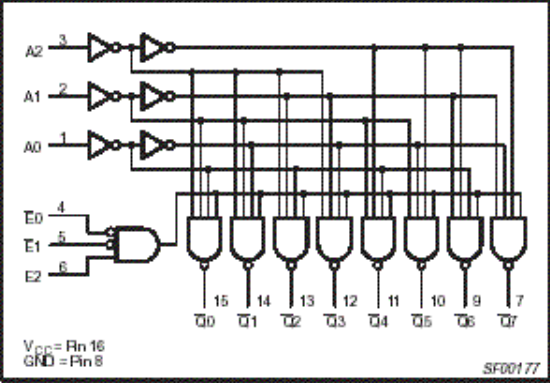


SC00047

1-of-8 decoder/demultiplexer

74ALS138

LOGIC DIAGRAM



FUNCTION TABLE

INPUTS						OUTPUTS							
E0	E1	E2	A0	A1	A2	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = High voltage level
L = Low voltage level
X = Don't care

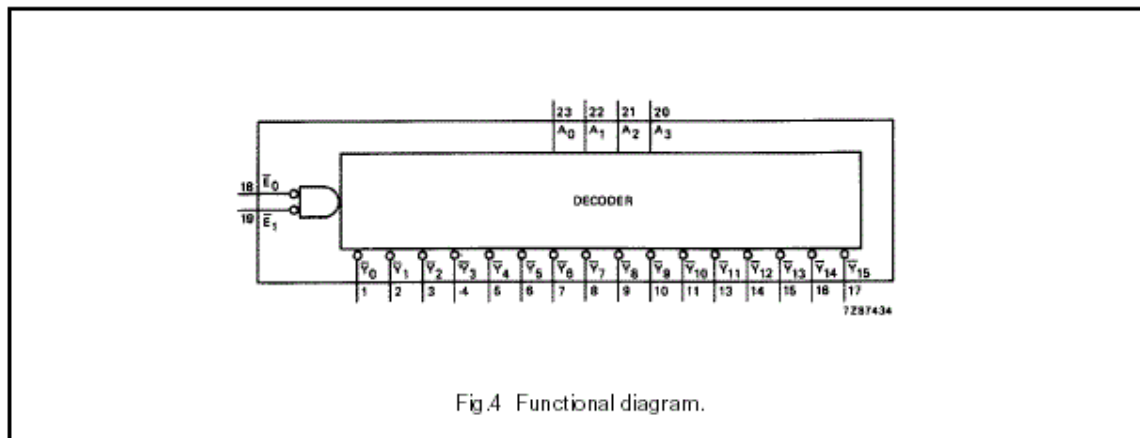
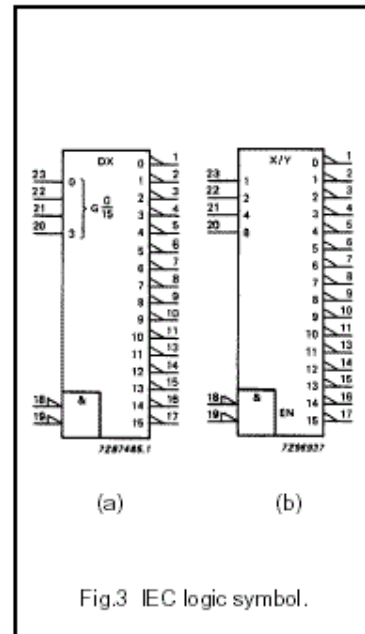
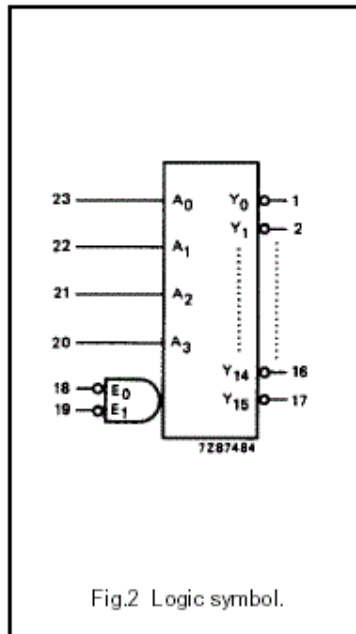
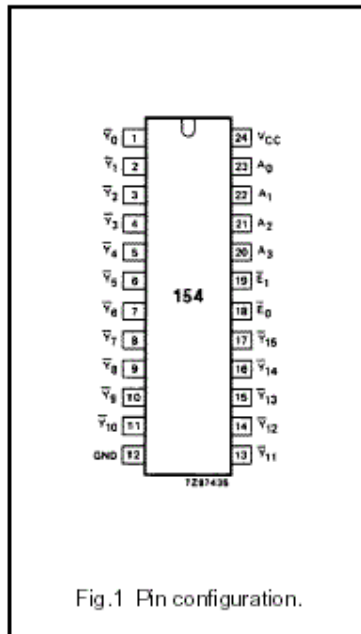
74HC154 4-16 KODÇÖZÜCÜ/VERİ DAĞITICI

4-to-16 line decoder/demultiplexer

74HC/HCT154

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17	\bar{Y}_0 to \bar{Y}_{15}	outputs (active LOW)
18, 19	\bar{E}_0, \bar{E}_1	enable inputs (active LOW)
12	GND	ground (0 V)
23, 22, 21, 20	A_0 to A_3	address inputs
24	V_{CC}	positive supply voltage



4-to-16 line decoder/demultiplexer

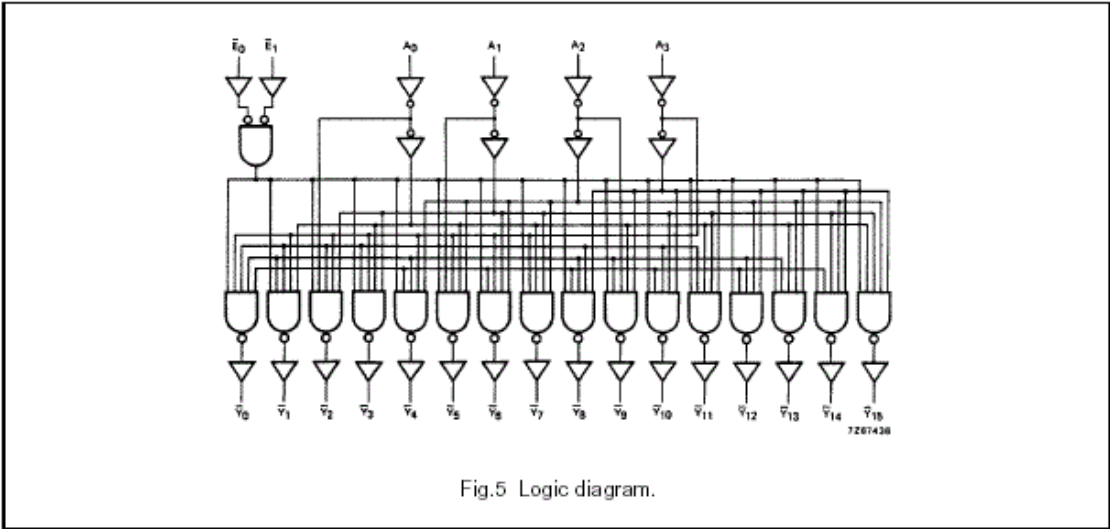
74HC/HCT154

FUNCTION TABLE

		INPUTS				OUTPUTS															
\overline{E}_0	\overline{E}_1	A_0	A_1	A_2	A_3	\overline{Y}_0	\overline{Y}_1	\overline{Y}_2	\overline{Y}_3	\overline{Y}_4	\overline{Y}_5	\overline{Y}_6	\overline{Y}_7	\overline{Y}_8	\overline{Y}_9	\overline{Y}_{10}	\overline{Y}_{11}	\overline{Y}_{12}	\overline{Y}_{13}	\overline{Y}_{14}	\overline{Y}_{15}
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

Note

- 1. H = HIGH voltage level
- L = LOW voltage level
- X = don't care



74HCT147 10 GİRİŞLİ 4 BİT ÇIKIŞLI YÜKSEK GİRİŞ ÖNCELİKLİ KODLAYICI

10-to-4 line priority encoder

74HC/HCT147

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
8	GND	ground (0 V)
9, 7, 6, 14	\bar{Y}_0 to \bar{Y}_3	BCD address outputs (active LOW)
11, 12, 13, 1, 2, 3, 4, 5, 10	\bar{A}_0 to \bar{A}_9	decimal data inputs (active LOW)
15	n.c.	not connected
16	V_{CC}	positive supply voltage

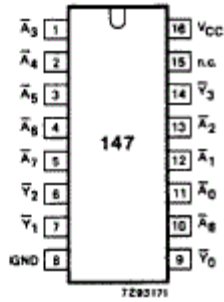


Fig.1 Pin configuration.

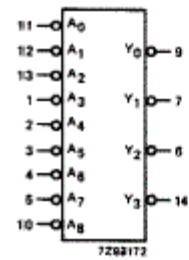


Fig.2 Logic symbol.

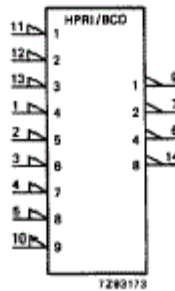
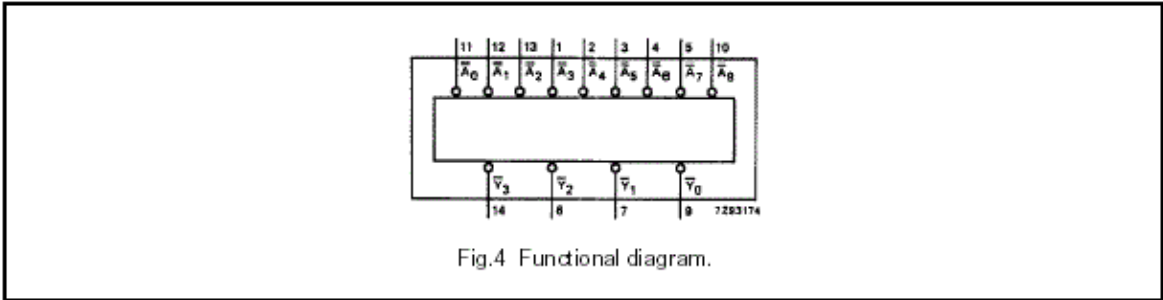


Fig.3 IEC logic symbol.



FUNCTION TABLE

INPUTS									OUTPUTS			
$\overline{A_0}$	$\overline{A_1}$	$\overline{A_2}$	$\overline{A_3}$	$\overline{A_4}$	$\overline{A_5}$	$\overline{A_6}$	$\overline{A_7}$	$\overline{A_8}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	L
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	L
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

Notes

- 1. H = HIGH voltage level
- L = LOW voltage level
- X = don't care

74LS151 8 GİRİŞLİ VERİ SEÇİCİ

8-input multiplexer

74ALS151

FEATURES

- 8-to-1 multiplexing
- On chip decoding
- Multi-function capability
- Complementary outputs
- See 74ALS251 for 3-State version

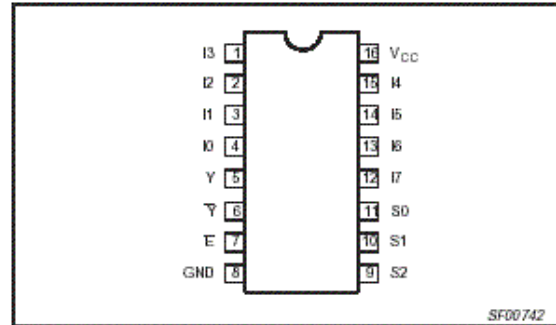
DESCRIPTION

The 74ALS151 is a logic implementation of a single 8-position switch with the switch position controlled by the state of three select (S_0 , S_1 , S_2) inputs. True (Y) and complementary (\bar{Y}) outputs are both provided.

The enable (\bar{E}) is active-Low. When \bar{E} is High, Y output is Low and the \bar{Y} output is High regardless of all other inputs.

TYPE	TYPICAL PROPAGATION DELAY	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS151	8.0ns	8.0mA

PIN CONFIGURATION



ORDERING INFORMATION

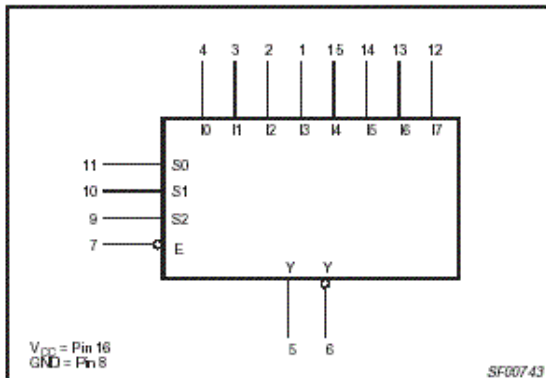
DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$, $T_{amb} = 0^\circ C \text{ to } +70^\circ C$	
16-pin plastic DIP	74ALS151N	SOT38-4
16-pin plastic SO	74ALS151D	SOT109-1

INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

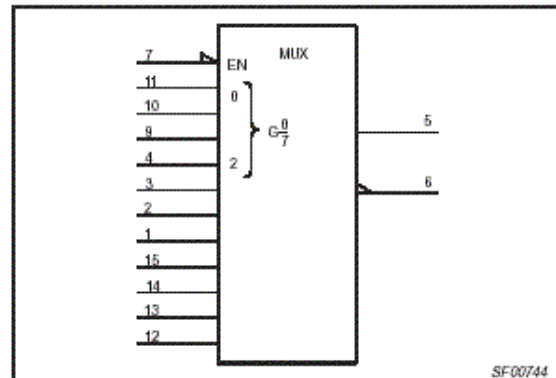
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
I0 – I7	Data Inputs	1.0/1.0	20 μ A/0.1mA
S0 – S2	Select Inputs	1.0/1.0	20 μ A/0.1mA
\bar{E}	Enable Input (active-Low)	1.0/1.0	20 μ A/0.1mA
Y, \bar{Y}	Data outputs	130/240	2.6mA/24mA

NOTE: One (1.0) ALS unit load is defined as: 20 μ A in the High state and 0.1mA in the Low state.

LOGIC SYMBOL

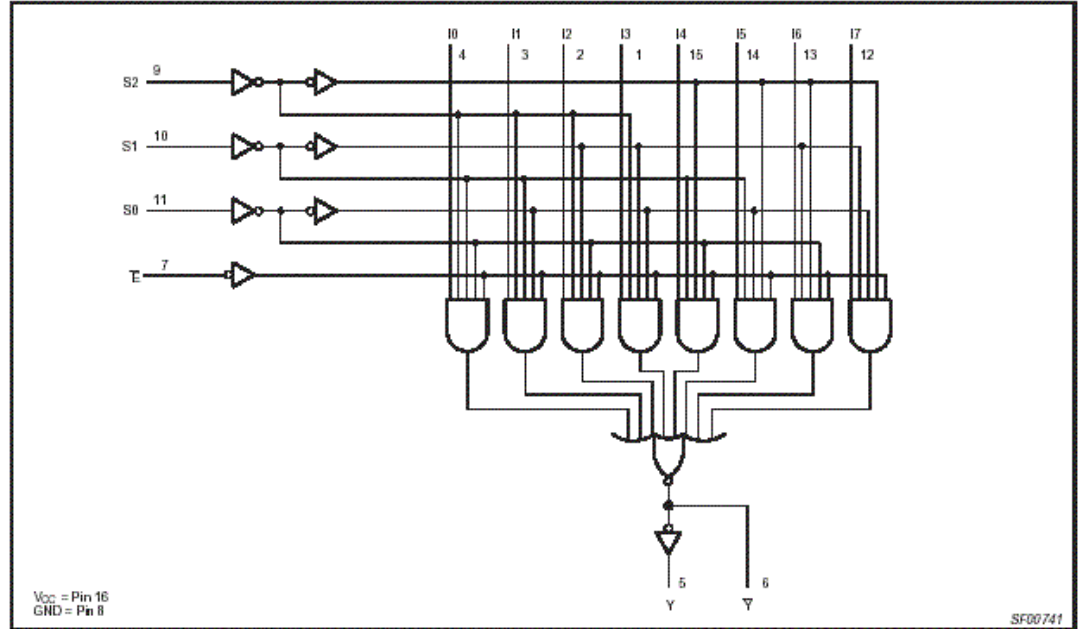


IEC/IEEE SYMBOL



8-input multiplexer	74ALS151
---------------------	----------

LOGIC DIAGRAM



FUNCTION TABLE

INPUTS				OUTPUTS	
S2	S1	S0	E	Y	\overline{Y}
X	X	X	H	L	H
L	L	L	L	I0	$\overline{I_0}$
L	L	H	L	I1	$\overline{I_1}$
L	H	L	L	I2	$\overline{I_2}$
L	H	H	L	I3	$\overline{I_3}$
H	L	L	L	I4	$\overline{I_4}$
H	L	H	L	I5	$\overline{I_5}$
H	H	L	L	I6	$\overline{I_6}$
H	H	H	L	I7	$\overline{I_7}$

H = High voltage level
L = Low voltage level
X = Don't care

74LS74A DUAL D TİPİ FLİP-FLOP

Dual D-type flip-flop with set and reset

74ALS74A

DESCRIPTION

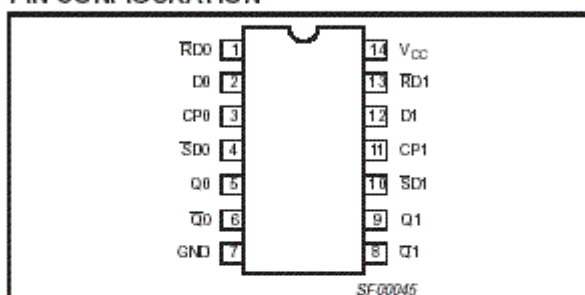
The 74ALS74 is a dual positive edge-triggered D-type flip-flop featuring individual data, clock, set, and reset inputs; also true and complementary outputs. Set (\overline{SD}) and reset (\overline{RD}) are asynchronous active-Low inputs and operate independently of the clock input. When set and reset are inactive (High), data at the D input is transferred to the Q and \overline{Q} outputs on the Low-to-High transition of the clock. Data must be stable just one setup time prior to the Low-to-High transition of the clock for predictable operation. Clock triggering occurs at a voltage level and is not directly related to the transition time of the positive-going pulse. Following the hold time interval, data at the D input may be changed without affecting the levels of the output.

TYPE	TYPICAL f_{MAX}	TYPICAL SUPPLY CURRENT (TOTAL)
74ALS74A	150MHz	3.0mA

ORDERING INFORMATION

DESCRIPTION	ORDER CODE	DRAWING NUMBER
	COMMERCIAL RANGE $V_{CC} = 5V \pm 10\%$ $T_{amb} = 0^\circ C \text{ to } +70^\circ C$	
14-pin plastic DIP	74ALS74AN	SOT27-1
14-pin plastic SO	74ALS74AD	SOT108-1
14-pin plastic SSOP Type II	74ALS74ADB	SOT337-1

PIN CONFIGURATION

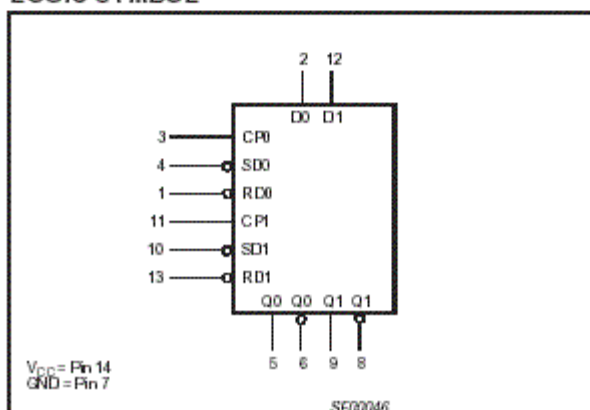


INPUT AND OUTPUT LOADING AND FAN-OUT TABLE

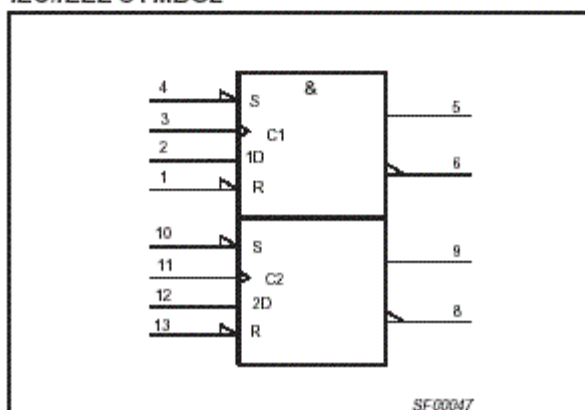
PINS	DESCRIPTION	74ALS (U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
D0, D1	Data inputs	1.0/2.0	20 μA /0.2mA
CP0, CP1	Clock inputs (active rising edge)	1.0/2.0	20 μA /0.2mA
$\overline{SD0}$, $\overline{SD1}$	Set inputs (active-Low)	2.0/4.0	40 μA /0.4mA
$\overline{RD0}$, $\overline{RD1}$	Reset inputs (active-Low)	2.0/4.0	40 μA /0.4mA
Q0, Q1, $\overline{Q0}$, $\overline{Q1}$	Data outputs	20/80	0.4mA/8mA

NOTE: One (1.0) ALS unit load is defined as: 20 μA in the High state and 0.1mA in the Low state.

LOGIC SYMBOL



IEC/IEEE SYMBOL

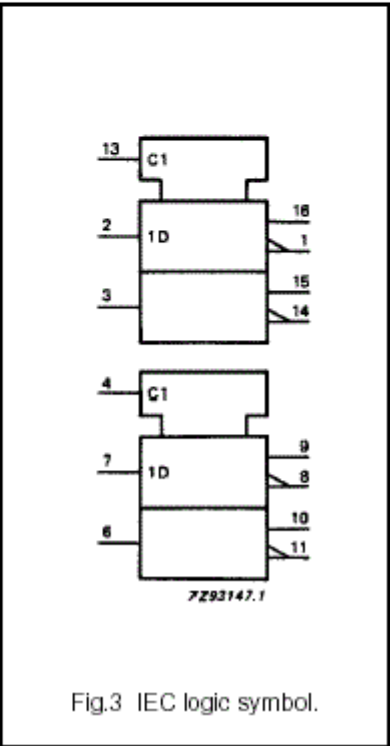
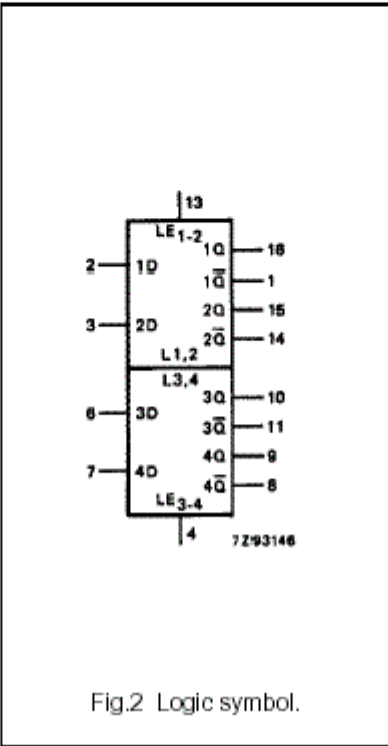
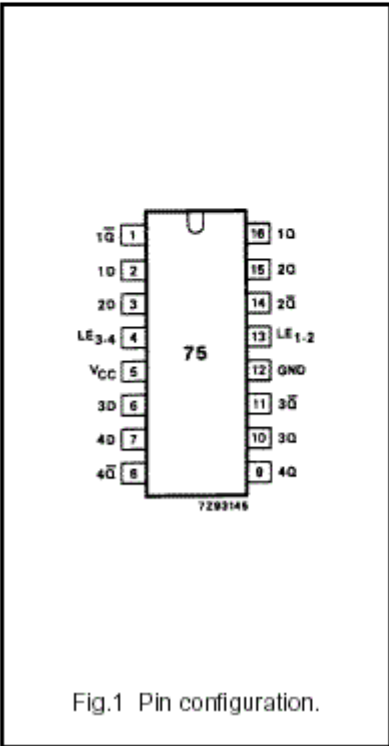


74HCT75 DÖRTLÜ TUTUCU

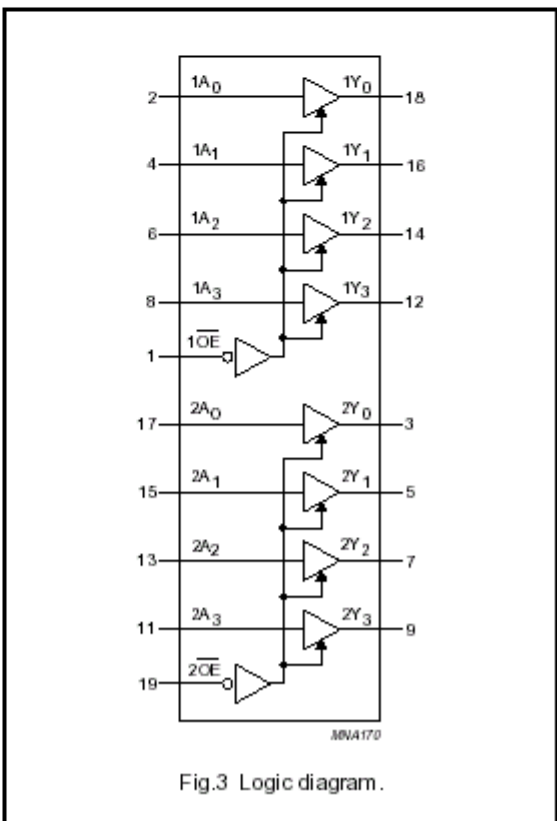
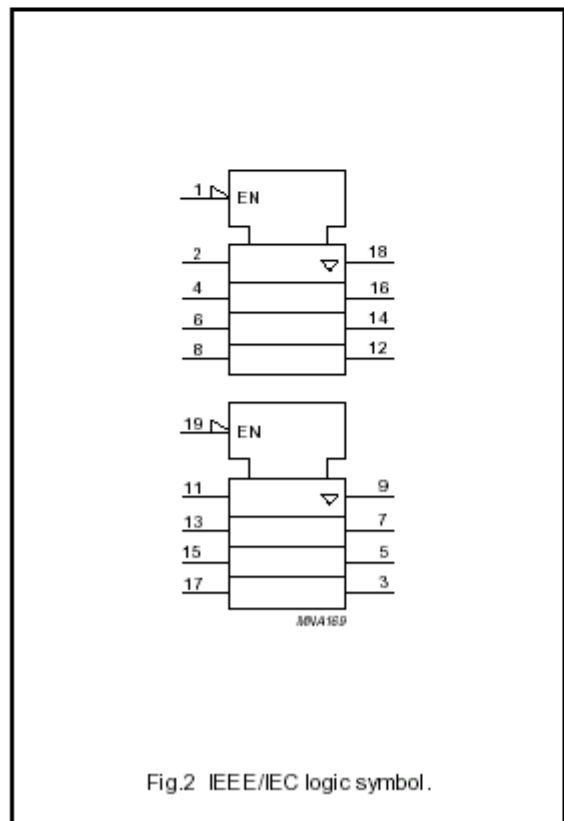
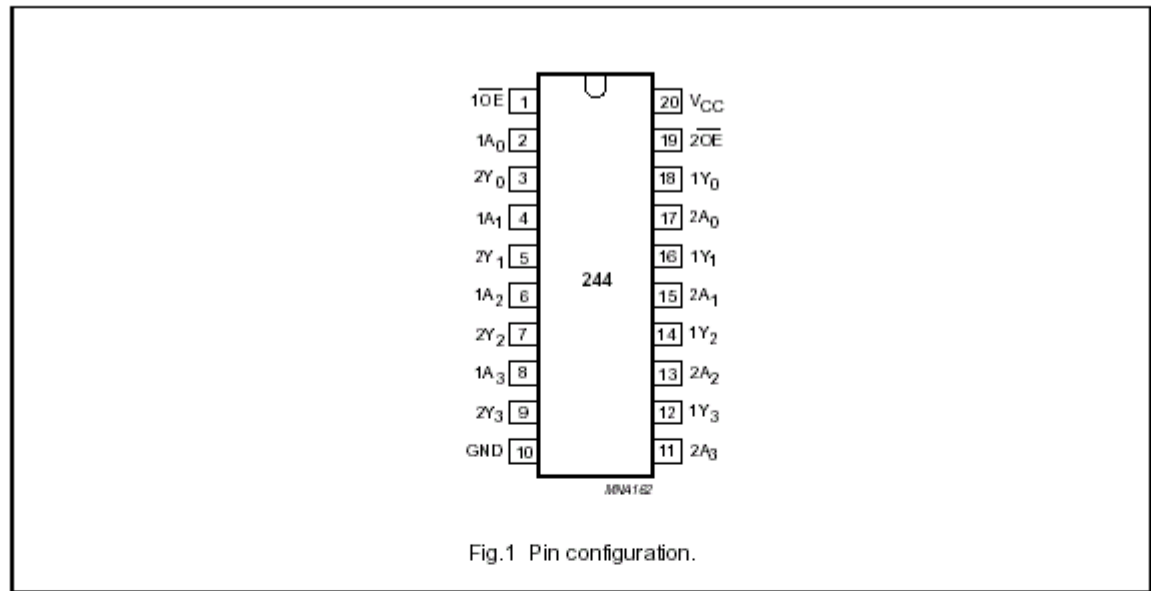
Quad bistable transparent latch	74HC/HCT75
---------------------------------	------------

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 14, 11, 8	1Q to 4Q	complementary latch outputs
2, 3, 6, 7	1D to 4D	data inputs
4	LE ₃₋₄	latch enable input, latches 3 and 4 (active HIGH)
5	V _{CC}	positive supply voltage
12	GND	ground (0 V)
13	LE ₁₋₂	latch enable input, latches 1 and 2 (active HIGH)
16, 15, 10, 9	1Q to 4Q	latch outputs



Octal buffer/line driver; 3-state	74AHC244; 74AHCT244
-----------------------------------	---------------------



LM555 Timer

Connection Diagram

