AMASYA ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

SAYISAL ELEKTRONİK

DERS NOTU

Doç. Dr. Ünal KURT

Arş. Gör. Ayşe AYDIN YURDUSEV

ÖNSÖZ

Bu kitapçıkta, Amasya Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği bölümünün 3. Yarıyılında yer alan Sayısal Elektronik Dersi, teorik ders notları bulunmaktadır.

Dersin içeriğindeki uygulama kısmını içeren bölüm ise ayrı bir kitapçık olarak kullanımınıza sunulmuştur.

Bu kitapçık, ders hakkında genel bilgileri içermekle beraber, dersin başarılı olarak tamamlanması için tek başına yeterli değildir. Kaynakçada belirtilen kitaplar da ders kitabı olarak kullanılmaktadır.

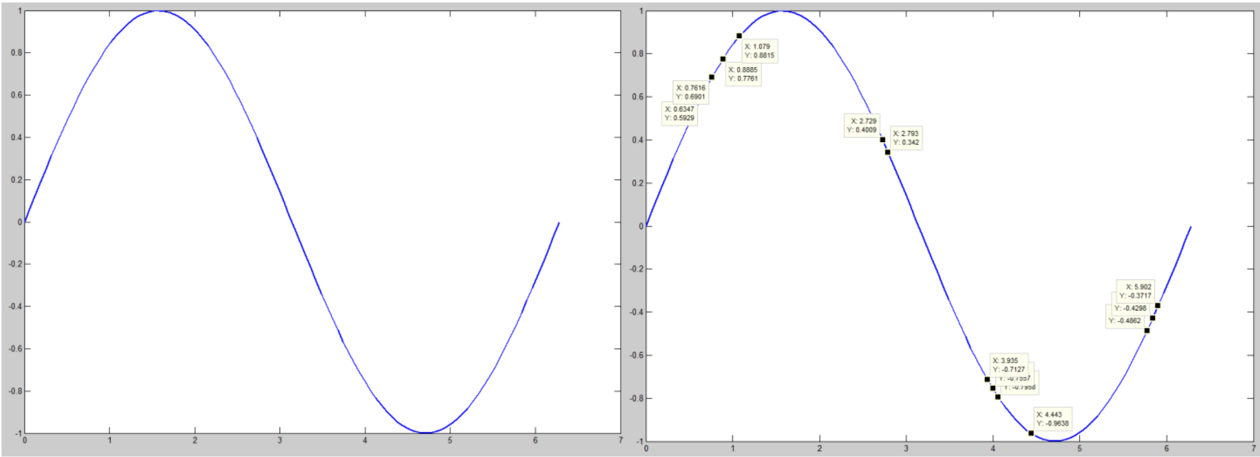
İÇİNDEKİLER

ŞEKİLLER LİSTESİ

1. ANALOG VE SAYISAL SİSTEMLER

1.1 ANALOG SİSTEMLER VE SİNYALLER

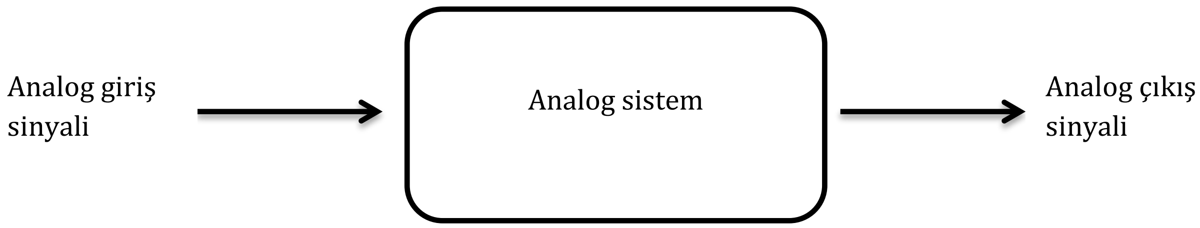
Sinyal zaman, uzay ya da başka bir değişkene göre değişiklik gösteren fiziksel niceliktir. (Başkent Üniversitesi) Analog sinyaller devamlı bir zamanda ilerleyen ve zamana göre değişken bir değer alan sinyallerdir. Analog bir sinyal zamana göre sonsuz değer alır. Analog sistemler ise analog bir sinyal ile işlem yapan fiziksel yapılardır.



ŞEKİL 1: ANALOG SİNYAL

Şekil 1'de bir analog sinyal gösterilmektedir. Bu sinyal analog olduğu için sinyalin istediğimiz bir noktasındaki bir değeri bize gösterebilir.

Günlük yaşamda etkin olan tüm fiziksel olaylar zamana göre sonsuz değer aldığından analogdur. Örneğin sıcaklık birden 20 °C'den 21 °C'ye çıkmaz. Bir eğri biçiminde sürekli olarak artarak nihai dereceye ulaşır.

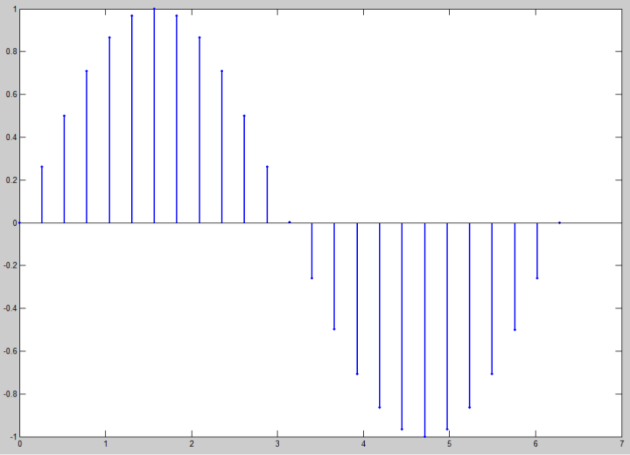


ŞEKİL 2: ANALOG SİSTEM

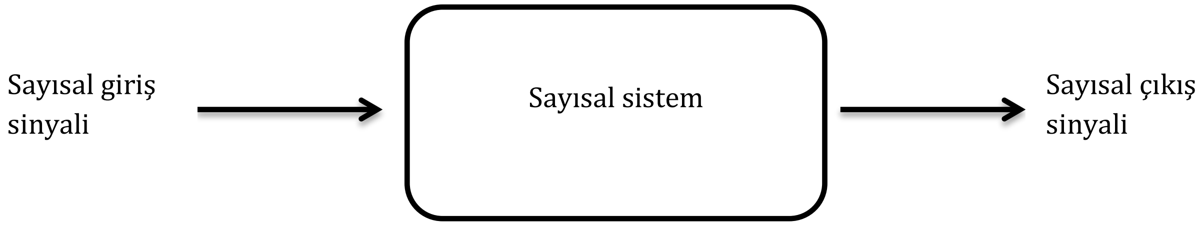
1.2 SAYISAL SİSTEMLER VE SİNYALLER

Sayısal sinyaller, analog sinyaller gibi sürekli değillerdir. Belli bir zaman ya da uzay değişkeninin belli bir değerine göre bir değer alırlar. Şekil 3'te bir sayısal sinyal görünmektedir.

Sayısal sinyallerle fiziksel olarak işlem yapan ortamlar ise sayısal sistem'dir.



ŞEKİL 3: SAYISAL SİNYAL



ŞEKİL 4: SAYISAL SİSTEM

Günümüz elektronik teknolojisinde sinyallerin çoğu sayısal olarak işlem görmektedir. Sayısal sistemlerin analog sistemlere göre bazı avantajları vardır. Bunlar;

- Sayısal devrelerin tasarımı daha kolaydır.
- Sayısal sistemlerde bilgi saklaması kolaydır
- Sayısal sistemler, analog sistemlere göre daha fazla istenilen sonucu verir. Birden çok sayısal devreyi birbirine bağlamak kolaydır.
- Sayısal sistemler gürültüden etkilenmezler.
- Sayısal sistemler donanımı değiştirilmeden tekrar tekrar programlanabilir.
- Sayısal sistemlerde hata ayıklamak daha kolaydır.

1.3 ANALOG VE SAYISAL SİSTEMLER

İnsanların işleyebildiği tüm sinyaller koklama, görme, işitme, dokunma analog sinyallerdir. Fakat günümüzde elektronik teknolojisi analog sinyallerle değil digital (sayısal) sinyallerle(yukarıda saydığımız avantajlardan dolayı) çalışır. Bu durumda analog ve sayısal sistemleri aynı ortamda kullanmak gereklidir. Böyle bir sistem için öncelikli olarak sinyallerin birbiri formuna dönüştürülmesi gerekmektedir. Kısaca özetlersek analogdan digital sinyale dönüştürme işlemi için:

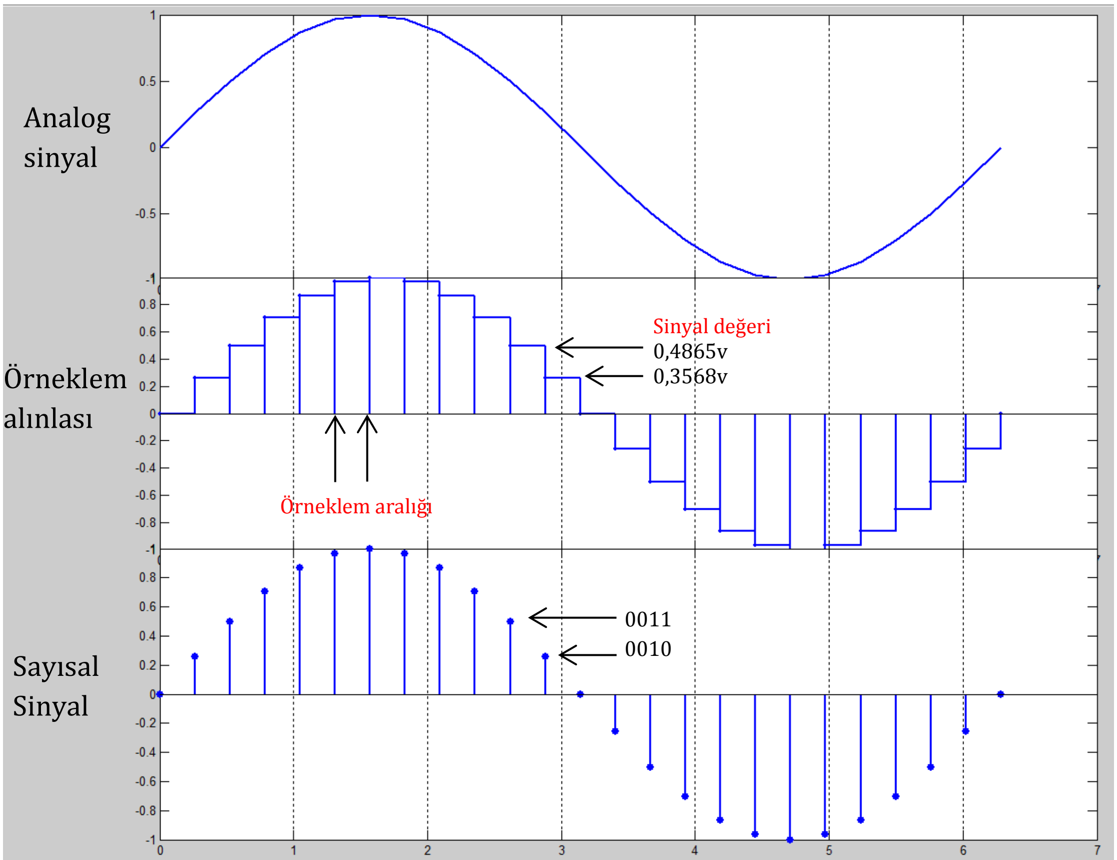
a. Bir analog sinyal öncelikle küçük örneklem aralıklarına bölünür.
b. Her bir örneklem için o örneklem değerine denk gelen sinyal değeri alınır.
c. Bu sinyal değeri sayısal olarak kodlanır.

Şekil 5'de bir periyottuk analog sinüzoidal dalganın sayısal sinüzoidal dalgaya dönüştürmesi görülmektedir.

Analog sinyali Dijital sinyale dönüştürme işlemi bazı önemli faktörler vardır. Nyquist kriteri olarak bilinen duruma göre, bir sinyalin örneklem frekansı, sinyalin kendi frekansından en az 2 katı olmalıdır (Nyquist frekansı). Örneğin 50Hz'lik bir dalga için örnekleme frekansı en az 100Hz olmalıdır. Fakat uygulamada Nyquist frekansı da yeterli olmaz. Gerçekten düzgün ve analog sinyale yakın bir dijital sinyal için örnekleme frekansı yüksek olmalıdır (10 katı-100 katı…). Böylece örnekleme aralığı küçülür.

Analog sinyali dijitale dönüştürme işlemi için bir diğer önemli faktör ise bölüntü seviyeleridir (Quantum seviyesi). Bölüntü seviyeleri sinyal değerinin dijital karşılığını verir. 2^n şeklinde hesaplanır. Örneğin 3 dijital çıkışlı bir çevirici $2^3=8$ ayrık bölüntü seviyesine sahiptir.

Bu işlemi yapan cihazlara ADC (Analogue to Digital Converter), bu işlemin tersini yapan, dijital sinyali analog sinyale dönüştürme işlemi yapan cihazlara DAC (Digital to Analogue Converter) denir.



Şekil 5: Analog Sinyalde Dijital Sinyale Dönüştürme

Günümüzde kullanılan elektronik sistemler analog ve dijital sistemlerin bir karmasıdır.

Giriş ve çıkış sinyalleri analog sinyal olmasına rağmen, sinyalin işlenmesi, saklanması dijital olarak gerçekleştirilir.

Şekil 6: Analog ve Dijital Sistem

2 SAYI SİSTEMLERİ VE KODLAR

2.1 SAYI SİSTEMLERİ

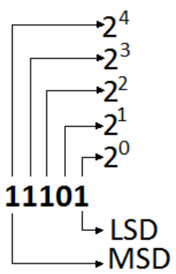
Dijital sistemler iki seviye ile çalışırlar. Bunlar lojik 1 ve lojik 0'dır (var veya yok). Bu yüzden bu sistemler 2 tabanlı sayı sistemlerini kullanılırlar. 2^1 (binary), 2^3(octal), 2^4 (hexadecimal) dijital sistemlerde en çok kullanılan sayı sistemleridir.

2.1.1 DECIMAL (ONLUK) SAYI SİSTEMİ

Onluk sayı sistemi günlük hayatta kullandığımız sistemdir. ...10^3 10^2 10^1 10^0 şeklinde dizilirler.

2.1.2 BINARY (İKİLİ) SAYI SİSTEMİ

İkili sayı sistemi iki tane rakam kullanılan sistemdir. Bunlar 0 ve 1 dir ve her biri bit olarak adlandırılır. Elektronikte var ya da yok, açık ya da kapalı gibi kesin yargı içeren sistemlerin temsilinde kullanılır. Örneğin 5 bitlik bir sayı $(11101)_2$ 'i ele alırsak;



$$(11101)_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$(11101)_2 = 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

$$(11101)_2 = (29)_{10} = 29$$

İki tabanlı sistemlerde sayılar en sağdaki basamaktan başlayarak 2 üzeri 0,1,2,3,...8,9... şeklinde değer alırlar. En sağdaki basamak sadece 20 yani 1 değerine sahip olduğu için LSD (least significant digit) ya da LSB (least significant bit)(en düşük değerli bit) adını alır. Sayının en solundaki basamak ise en yüksek değerli olduğu için MSD-MSB (most significant dijit (bit)) (en yüksek değerli bit) adını alır. Eğer sayımız tam sayı değilse, virgülden sonraki basamak soldan başlayarak 2 üzeri -1,-2,-3....-8,-9... şeklinde değer alırlar. Kısaca özetlersek

$b_n b_{n-1} \ldots b_2 b_1 b_0$, $b_{-1} b_{-2} \ldots b_{m-1} b_m$ şeklindeki bir sayının onluk karşılığı:

$$B = \underbrace{b_n 2^n + b_{n-1} 2^{n-1} + \cdots + b_2 2^2 + b_1 2^1 + b_0 2^0}_{\text{Tam sayı kısmı}}, \underbrace{b_{-1} 2^{-1} + b_{-2} 2^{-2} + \cdots b_{m-1} 2^{m-1} + b_m 2^m}_{\text{Kesirli sayı kısmı}}$$

İkili sayı sistemleri bilgisayarlarda genellikle sayısal değeri ifade etmek için, adres belirtmek için, komut kodu belirtmek için kullanılırlar.

2.1.3 OCTAL (SEKİZLİ) SAYI SİSTEMİ

Sekizli sayı sisteminde 8 adet sayı kullanılır bunlar 0 ila 7 arasındadır (0 ve 7 dahil). Genel formülü:

$$O = b_n 8^n + b_{n-1} 8^{n-1} + \cdots + b_2 8^2 + b_1 8^1 + b_0 8^0 \ , \qquad b_{-1} 8^{-1} + b_{-2} 8^{-2} + \cdots b_{m-1} 8^{m-1} + b_m 8^m$$

2.1.4 HEXADECIMAL (ONALTILI) SAYI SİSTEMİ

Hexadecimal sayı sistemi sayıyı onaltı tabanında yazar. 0'dan 9'a kadar rakamları ve A,B,C,D,E,F harflerini kullanır. Günümüz bilgisayarlarında sıkça kullanılırlar, büyük rakamları yazmak için az sayıda basamak yeterlidir. Genel ifadesi:

$$H = b_n 16^n + b_{n-1} 16^{n-1} + + b_1 16^1 + b_0 16^0 \ , \qquad b_{-1} 16^{-1} + b_{-2} 16^{-2} + \cdots b_{m-1} 16^{m-1} + b_m 16^m$$

2.2 SAYI SİSTEMLERİ ARASINDAKİ DÖNÜŞÜMLER

Kullanılan bilgisayar sistemi ya da elektronik sistemin kod yazılışı genellikle binary, octal veya hexadecimal olarak tasarlanır. Günlük hayatta ise onluk sistem kullanılmaktadır. Bu durumda sayı sistemleri arasında dönüşüm yapılması gerekmektedir.

DECIMAL	BINARY	OCTAL	HEXADECIMAL
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

TABLO 1: 0-15 ARASI SAYILARIN KARŞILAŞTIRILMASI

Tablo 1'de elektronik ve bilgisayar işlemlerinde sıkça kullanılan ilk 15 sayının ondalık, ikilik, sekizlik ve onaltılık dönüşümleri görülmektedir.

2.3 ON TABANINDA SAYILARIN İKİLİ, SEKİZLİ ve ONALTILI SİSTEME DÖNÜŞTÜRÜLMESİ

a. İkili Sayılara Dönüştürme

On tabanındaki sayı ikiye bölünür. Kalan not edilerek bölüm tekrar ikiye bölünür. Bölüm sıfır olana kadar devam edilir. Sonuç MSB'den LSB'ye doğru yazılır.

Bölünen	**Bölüm**	**Kalan**	
621	310	**1**	**←LSB**
310	155	**0**	
155	77	**1**	
77	38	**1**	
38	19	**0**	
19	9	**1**	
9	4	**1**	
4	2	**0**	
2	1	**0**	
1	0	**1**	**←MSB**

$(621)_{10}=(1001101101)_2$

Kesirli sayılarda ise sayının virgülden sonraki kısmı 2 ile çarpılır, çarpımın tam sayı kısmı not edilerek, sonuç sıfıra yaklaşana kadar tekrarlanır, bu durumda bize yaklaşık sonucu verir. İlk tam sayıdan başlanarak sayı yazılır.

Çarpılan	**Çarpım**	**Tam Sayı**
0,35	0,70	**0**
0,70	1,40	**1**
0,40	0,80	**0**
0,80	1,60	**1**
0,60	1,20	**1**
0,20		

$(0{,}35)_{10} \cong (0{,}01011)_2$

Örneğin 521,65 sayısının binary yazılımı:

Tam sayı kısmı

Bölünen	**Bölüm**	**Kalan**	
521	260	**1**	**←LSB**
260	130	**0**	
130	65	**0**	
65	32	**1**	
32	16	**0**	
16	8	**0**	
8	4	**0**	
4	2	**0**	
2	1	**0**	
1	0	**1**	**←MSB**

Kesirli kısım

Çarpılan	**Çarpım**	**Tam Sayı**
0,65	1,30	**1**
0,30	0,60	**0**
0,60	1,20	**1**
0,20		

$(521{,}65)_{10} \cong (1000001001{,}101)_2$

b. Sekizli Sayılara Dönüştürme

Onlu bir sayısı sekizli bir sayıya dönüştürmek için ikili sayı sistemdeki kurallar uygulanır. Yalnız burada kesirli sayılar 8 ile çarpılır ve tam sayılar 8'e bölünür.

Bölünen	Bölüm	Kalan	
621	77	**5**	**←LSB**
77	9	**5**	
9	1	**1**	
1	0	**1**	**←MSB**

$(621)_{10}=(1155)_8$

Çarpılan	Çarpım	Tam Sayı
0,35	2,80	**2**
0,80	6,40	**6**
0,40	3,20	**3**
0,20		

$(0,35)_{10}\cong(0,263)_8$

Tam sayı kısmı

Bölünen	Bölüm	Kalan	
521	65	**1**	**←LSB**
65	8	**1**	
8	1	**0**	
1	0	**1**	**←MSB**

Kesirli kısım

Çarpılan	Çarpım	Tam Sayı
0,65	5,20	**5**
0,20		

$(521,65)_{10}\cong(1011,5)_8$

c. Onaltılı Sayılara Dönüştürme

Onlu bir sayıyı onaltı tabanında yazmak için tam sayılar 16 ile bölünür ve kesirli sayılar 16 ile çarpılır.

Bölünen	Bölüm	Kalan	
621	38	**D**	**←LSB**
38	2	**6**	
2	0	**2**	**←MSB**

$621=(26D)_{16}$

Çarpılan	Çarpım	Tam Sayı
0,35	5,60	**5**
0,60	9,60	**9**
0,60	9,60	**9**
Kendini tekrarladığı için işlem bırakılır		

$(0,35)\cong(0,599)_{16}$

Tam sayı kısmı				Kesirli kısım			
Bölünen	Bölüm	Kalan		Çarpılan	Çarpım	Tam Sayı	
521	32	**9**	**←LSB**	0,65	10,40	**A**	↓
32	2	**0**		0,40	6,40	**6**	
2	0	**2**	**←MSB**	0,40	6,40	**6**	↓
				Kendini tekrarladığı için işlem bırakılır			

$(521{,}65)_{10} \cong (209{,}A66)_{16}$

2.2.2 İKİ TABANINDA SAYILARIN, ONLU, SEKİZLİ ve ONALTILI SİSTEME DÖNÜŞTÜRÜLMESİ

a. İkili sayıların onlu sisteme dönüştürülmesi

Daha önce değinildiği gibi, ikili sayılar onlu sisteme basamak değerlerine göre 2 üzeri değerlerle çarpılarak elde edilirler.

$$B = b_n 2^n + b_{n-1} 2^{n-1} + \cdots + b_2 2^2 + b_1 2^1 + b_0 2^0 \ , \qquad b_{-1} 2^{-1} + b_{-2} 2^{-2} + \cdots b_{m-1} 2^{m-1} + b_m 2^m$$

Örneğin $(100110{,}101)_2$ sayısı için;

$100110 = 0x2^0 + 1x2^1 + 1x2^2 + 0x2^3 + 0x2^4 + 1x2^5 = 2+4+32 = 38$

$0{,}101 = 1x2^{-1} + 0x2^{-2} + 1x2^{-3} = \frac{1}{2} + \frac{0}{4} + \frac{1}{8} = 0{,}625$

$(100110{,}101)_2 = (38{,}625)$

b. İkili sayıların sekizli sisteme dönüştürülmesi

İkili sayıları sekizli sayılara dönüştürmek tam sayı kısmında sağdan sola doğru, kesirli kısımda soldan sağa doğru olacak şekilde sayı üçer bitlik gruplar halinde ayrılır. Son gruplar üç basamaktan azsa sıfır eklenerek üçe tamamlanır ve her bir gruba karşılık gelen octal değer yazılır. Not: 3 basamaklı binary sayılar için octal değer decimal değerle aynıdır.

Tam sayı kısmı	Kesirli sayı kısmı
gruplandırma yönü ←	→gruplandırma yönü
01011000101011010111000,	10010111010001010100101001
001 011 000 101 011 010 111 000,	100 101 110 100 010 101 001 010 010
1 3 0 5 3 2 7 0,	4 5 6 4 2 5 1 2 1
13053270,	456425121

Sonuç olarak:

$(0101100010101101011100 0,\ 100101110100010101001010 01)_2=(13053270,456425121)_8$

c. İkili sayıların onaltılı sisteme dönüştürülmesi

İkili sayıların onaltılı sayılara dönüşmesi için sayılar dörtderli şekilde gruplanır. Karşılılarına onaltılık sayı yazılır.

Tam sayı kısmı	Kesirli sayı kısmı
gruplandırma yönü ←	→gruplandırma yönü
010110001010110101110 00,	1001011101000101010010 1001
0010 1100 0101 0110 1011 1000,	1001 0111 0100 0101 0100 1010 0100
2 C 5 6 B 8,	9 7 4 5 4 A 4
2C56B8,	97454A4

$(01011000101011010111000,\ 100101110100010101001010 01)_2=(2C56B8,97454A4)_{16}$

2.2.3 SEKİZ TABANINDA SAYILARIN İKİLİ, ONLU ve ONALTILI SİSTEME DÖNÜŞTÜRÜLMESİ

a. Sekizli sayıların onlu sisteme dönüştürülmesi

Herbir basamak rakam basamağının değeriyle çarpılır. Bu değerler tam sayı kısmı için 8 üzeri0,1,2.. şeklinde artarken, kesirli kısım için 8 üzeri -1,-2… şeklinde azalır.

$$O = b_n 8^n + b_{n-1}8^{n-1} + \cdots + b_2 8^2 + b_1 8^1 + b_0 8^0\ , \qquad b_{-1}8^{-1} + b_{-2}8^{-2} + \cdots b_{m-1}8^{m-1} + b_m 8^m$$

$(472,36)_8=4x8^2+7x8^1+2x8^0$, $3x8^{-1}+6x8^{-2}$=256+56+2 , 0,375+0,09375 ≅ 314, 47

b. Sekizli sayıların ikili sisteme dönüştürülmesi

İkili sayıların sekizli sayılara dönüştürülmesinin tersi yapılır. Her bir basamak için binary karşılık yazılır.

$(314,47)_8$=> 3=011 1=001 4=100, 4=100 7=111

$(314,47)_8=(011001100,100111)_2$

c. Sekizli sayıların onaltılı sisteme dönüştürülmesi

Sekizli sayıları onaltı sayıya çevirmek için öncelikle 2.2.3 b'de gösterilen şekilde , iki sisteme dönüştürüp sonrada ikili sistemden onaltılı sisteme dönüştürülür.

İkili sayıların sekizli sayılara dönüştürülmesinin tersi yapılır. Her bir basamak için binary karşılık yazılır.

İkili sayıların onaltılı sayılara dönüşmesi için sayılar dörtderli şekilde gruplanır. Karşılılarına onaltılık sayı yazılır.

2.2.4 ONALTI TABANINDA SAYILARIN İKİLİ, ONLU ve SEKİZLİ SİSTEME DÖNÜŞTÜRÜLMESİ

a. Onaltılı sayıların onlu sisteme dönüştürülmesi

$$H = b_n 16^n + b_{n-1} 16^{n-1} + +b_1 16^1 + b_0 16^0 \ , \qquad b_{-1} 16^{-1} + b_{-2} 16^{-2} + \cdots b_{m-1} 16^{m-1} + b_m 16^m$$

Formülü ile her sayı 16'nın üsleriyle çarpılır ve kesirli ksıım için her basamak 16'nın üstlerine bölünür.

$(D8C,B53)_{16}$= $Dx16^2+8x16^1+Cx16^0$, $Bx16^{-1}+5x16^{-2}+3x16^{-3}$

=3328+128+12 , 0,6875+0,019+0,0007

=3468, 7072

b. Onaltılı sayıların ikili sisteme dönüştürülmesi

Her bir sayı için o sayının ikili karşılığı 4 bit olarak yazılır. Sonra bu rakamlar yan yana getirilir.

$(BAE,02F)_{16}$

B	A	E,	0	2	F
1011	1010	1110	0000	0010	1111

$(BAE,02F)_{16}$=(101110101110,000000101111)

c. Onaltılı sayıların sekizli sisteme dönüştürülmesi

Onaltılı sayı önce ikili sayıya dönüştürülür. Sonra ikili kod sekizli koda dönüştürülür.

Her bir sayı için o sayının ikili karşılığı 4 bit olarak yazılır. Sonra bu rakamlar yan yana getirilir.

$(BAE,02F)_{16}$

B	A	E,	0	2	F
1011	1010	1110	0000	0010	1111

$(BAE,02F)_{16}$=(101110101110,000000101111)

İkili sayıları sekizli sayılara dönüştürmek tam sayı kısmında sağdan sola doğru, kesirli kısımda soldan sağa doğru olacak şekilde sayı üçer bitlik gruplar halinde ayrılır. Son gruplar üç basamaktan azsa sıfır eklenerek üçe tamamlanır ve her bir gruba karşılık gelen octal değer yazılır. Not: 3 basamaklı binary sayılar için octal değer decimal değerle aynıdır.

Tam sayı kısmıKesirli sayı kısmı

Tam sayı kısmı	**Kesirli sayı kısmı**
gruplandırma yönü ←	→gruplandırma yönü
101110101110,	000000101111
101-110-101-110,	000-000-101-111
5-6-5-6,	0-0-5-7
5656,	0057

Sonuç olarak: $(101110101110,000000101111)_2=(5656, \quad 0057)_8$

2.2.5 SAYI SİSTEMLERİ SORULARI

Doğum tarihinizi gün ay yıl olarak onluk sayı olarak yazınız. Ör: 19 Mayıs 1991 için: 190591

Bu sayıyı aşağıdaki sistemlere dönüştürünüz.

a) İkili sayı sistemine
b) Sekizli sayı sistemine
c) Onaltılı sayı sistemine

Doğum tarihi sayınızı iki ile çarpıp üçe bölünüz. Bu yeni sayıyı aşağıdaki sistemlere dönüştürünüz.

d) İkili sayı sistemine
e) Sekizli sayı sistemine
f) Onaltılı sayı sistemine

Aşağıdaki alıştırmaları tamamlayınız.

g) $(ABCD,DEF)_{16} = (\quad)_8$
h) $(7654,321)_8 = (\quad)_{16}$
i) $(1001,101)_{16} = (\quad)_8$
j) $(10025,5)_8 = (\quad)_{16}$
k) $(A15F,D)_{16} = (\quad)_8$
l) $(6593,1)_8 = (\quad)_{16}$

2.3 SAYI SİSTEMLERİNDE HESAPLAMA

2.3.1 İKİLİ SAYI SİSTEMİNDE TOPLAMA ve ÇIKARMA

İkili sayı sisteminde toplama kuralları aşağıdaki gibidir:

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 0 \end{array} \ \leftarrow \text{elde } 1$$

Örneğin;

$$\begin{array}{r} 1010 \\ +\ 0010 \\ \hline 1100 \end{array} \qquad \begin{array}{r} 11101 \\ +\ 11001 \\ \hline 110110 \end{array} \qquad \begin{array}{r} 1001 \\ 1011 \\ +\ 0010 \\ \hline 10110 \end{array} \qquad \begin{array}{r} 0011 \\ 1010 \\ +\ 1101 \\ \hline 11010 \end{array}$$

İkili sayı sisteminde çıkarma kuralları şöyledir:

$$\begin{array}{r} 0 \\ -\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 10 \\ -\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ -\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ -\ 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ -\ 1 \\ \hline 1 \end{array} \ \leftarrow \text{borç } 1$$

Sadece çıkarma kuralları uygulanarak yapılan çıkarma işlemine doğrudan çıkarma işlemi denir. Örneğin:

$$\begin{array}{r} 11010 \\ -\ 0110 \\ \hline 10100 \end{array} \qquad \begin{array}{r} 110101 \\ -\ 11001 \\ \hline 11100 \end{array}$$

2.3.2 TÜMLEYEN ARİTMETİĞİ

a. ”r” tümleyen aritmetiği ve çıkartma

Herhangi bir r tabanında n basamaklı N sayısı için r tümleyeni: r^n-N ‘dir.

Örneğin 67 sayısı için: r=10 tabanı n=2 basamaklı

10^2-67=100-67=33

$(25683)_{10}$ => r tümleyeni=10^5-25683=74317

$(10010101)_2$ => r tümleyeni=2^8-10010101=100000000-10010101=1101011

Kesirli sayılar için basamak sayısı 0 kabul edilir:

(0,101)2 =>r tümleyeni=20-0,101=1-0,101=0,011

R tümleyeni ile çıkarma işleminde bir sayıyı diğerinden çıkartmak yerine, bir sayı olduğu gibi alınır, diğer sayının ise tümleyeni alınarak toplama işlemi yapılır. Çünkü toplama işlemi çıkarma işleminden daha kolaydır. Bu durumda sonuç iki şekilde yorumlanır:

i. Sonucun en büyük basamağında elde 1 varsa sonuç pozitiftir.
ii. Sonucun en büyük basamağında (en sol) elde yoksa sonuç negatiftir. Bu sonucun r tümleyeni alınır ve başına eksi işareti konulur.

Örneğin

1) $(562)_{10}-(345)_{10}$=?
   345'in 10 tümleyeni: 10^3-345=1000-345=655
   562+655=1217 sayının en sol basamağında elde 1 olduğu için sonuç:+217
   $(562)_{10}-(345)_{10}=(217)_{10}$

2) $(611)_{10}-(877)_{10}$=?
   877'nin 10 tümleyeni: 10^3-877=123
   611+123=0734 sayını en sol basamağında elde 1 olmadığı için sonuç negatif.
   Sonucun 10 tümleyeni alınır:
   734'ün 10 tümleyeni: 10^3-734=266
   Sonuç: -266
   $(611)_{10}-(877)_{10}=(-266)_{10}$

3) $(1001)_2-(1011)_2$=?

   1011'in 2 tümleyeni=24-1011=10000-1011=0101

   $(1001)_2+(0101)_2=(01110)_2$ önünde elde 1 yok

   Sonucun 2 tümleyeni alınır:

   2^4-1110=10000-1110=0010

   $(1001)_2-(1011)_2=-(0010)_2$

4) $(11101)_2$-$(10110)_2$=?

10110' ın 2 tümleyeni:

2^5-110=100000-10110=1010

$(11101)_2$+$(1010)_2$=$(100111)_2$ elde 1 var. Sonuç pozitif

$(11101)_2$-$(10110)_2$=$(00111)_2$

b. "r-1" tümleyen aritmetiği ve çıkarma

Herhangi bir r tabanında n basamaklı N sayısı için r-1 tümleyeni için

r^n-N-1 formülü kullanılır.

Örneğin 10 tabanındaki 3 basamaklı 256 sayısının r-1=10-1=9 tümleyeni için

10^3-256-1= 1000-256-1=999-256=743

Not: dikkat ederseniz 9 tümleyende her bir basamak kendi tümleyeniyle toplanırsa sonuç 9 elde edilir.

Bir sayı kesirli ise o sayının tam sayı kısmı n basamaklı, kesirli kısmı m basamaklı oldun. Bu sayının r-1 tümleyeni: r^n-r^{-m}-N'dir.

0,423 sayısı için: n=0 m=3 r=10 9 tümleyeni:

10^0-10^{-3}-0,423=1-0,001-0,423= 0,576

Not: tümleyen ve tümlenen sayının toplamı: 0,99… şeklindedir.

r-1 tümleyeni yöntemiyle çıkarma işlemi yapmak için, ilk sayının kendisiyle ikinci sayının r-1 tümleyeni toplanır. Sonuca göre:

i. İşlem sonucunda elde 1 varsa sonuç pozitiftir. Bu sonuca 1 eklenir ve sonuç aynen yazılır.
ii. İşlem sonucunda elde 1 yoksa bu sonuç negatiftir. Sonucun r-1 tümleyeni alınır ve önüne eksi işareti konularak yazılır.

Örneğin:

a) $(945)_{10}$-$(158)_{10}$=

158'in 9 tümleyeni: 841

945+841=1786, elde 1 var, sonuç pozitif

786+1=787

$(945)_{10}$-$(158)_{10}$=$(787)_{10}$

b) $(321)_{10}$-$(432)_{10}$=

432'nin 9 tümleyeni :567

321+567=0888, elde 1 yok sonuç negatif

888'in 9 tümleyeni 111

$(321)_{10}$-$(432)_{10}$=$(-111)_{10}$

c) $(11001)_2$-$(00101)_2$=

00101'in 1 tümleyeni: 11010

11001+11010=110011 elde 1 var sonuç pozitif

10011+1=10100

$(11001)_2$-$(00101)_2$=$(10100)_2$

d) $(10001)_2$-$(11001)_2$=$(?)_2$

11001'in 1 tümleyeni: 00110

10001+00110=010111, elde 1 yok sonuç negatif;

10111'in 1 tümleyeni: 01000

$(10001)_2$-$(11001)_2$=$(-01000)_2$

2.3.3 İKİLİ SAYI SİSTEMİNDE ÇARPMA ve BÖLME

İkili sayı sisteminde çarpma kuralları aşağıdaki gibidir.

$$\begin{array}{r} 0 \\ x\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ x\ 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ x\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ x\ 1 \\ \hline 1 \end{array}$$

Çarpma işlemi onluk sistemde olduğu gibi yapılır:

$$\begin{array}{ccccc} & & 1 & 0 & 1 \\ & x & & 1 & 0 \\ \cline{2-5} & & 0 & 0 & 0 \\ + & 1 & 0 & 1 & \\ \hline & 1 & 0 & 1 & 0 \end{array} \qquad\qquad \begin{array}{cccccccc} & & & & 1 & 1 & 1 & 0 \\ & & & x & & 1 & 0 & 1 \\ \cline{4-8} & & & & 1 & 1 & 1 & 0 \\ & & & 0 & 0 & 0 & 0 & \\ + & & 1 & 1 & 1 & 0 & & \\ \hline & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

İkili sayı sisteminde bölme işlemi de onluk sistemle aynı temellere dayanmaktadır.

$$\begin{array}{cccccc|ccc} & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ \cline{7-9} - & 1 & 0 & 1 & & & 1 & 1 & 0 \\ \cline{1-5} & 0 & 1 & 0 & 1 & & & & \\ & - & 1 & 0 & 1 & & & & \\ \cline{2-5} & & 0 & 0 & 0 & & & & \end{array}$$

2.3.4 İKİLİ SAYI SİSTEMİNDE HESAPLAMA SORULARI

I. Aşağıdaki toplama işlemlerini yapınız.

a)
$$\begin{array}{rl} & (1\ 0\ 0\ 1\ 0)_2 \\ & (0\ 1\ 1\ 1\ 0)_2 \\ + & (1\ 0\ 1\ 1\ 1)_2 \\ \hline \end{array}$$

b)
$$\begin{array}{rl} & (1\ 1\ 0\ 1\ 1)_2 \\ & (0\ 1\ 0\ 1\ 0)_2 \\ + & (0\ 0\ 1\ 1\ 0)_2 \\ \hline \end{array}$$

c)
$$\begin{array}{rl} & (1\ 1\ 0\ 1\ 1)_2 \\ & (0\ 1\ 0\ 1\ 0)_2 \\ & (1\ 0\ 1\ 1\ 1)_2 \\ & (0\ 1\ 0\ 0\ 0)_2 \\ + & (0\ 0\ 1\ 1\ 0)_2 \\ \hline \end{array}$$

d)
$$\begin{array}{rl} & (0\ 0\ 0\ 0\ 1)_2 \\ & (0\ 0\ 0\ 1\ 1)_2 \\ & (0\ 0\ 1\ 1\ 1)_2 \\ & (0\ 1\ 1\ 1\ 1)_2 \\ + & (1\ 1\ 1\ 1\ 1)_2 \\ \hline \end{array}$$

II. Aşağıdaki işlemleri doğrudan çıkarma yöntemi kullanarak yapınız

a)
$$\begin{array}{rl} & (1\ 0\ 0\ 1\ 0)_2 \\ - & (0\ 0\ 1\ 1\ 1)_2 \\ \hline \end{array}$$

b)
$$\begin{array}{rl} & (1\ 0\ 1\ 0)_2 \\ - & (0\ 1\ 0\ 1)_2 \\ \hline \end{array}$$

c)
$$\begin{array}{rl} & (1\ 0\ 1\ 0)_2 \\ - & (0\ 1\ 1\ 1)_2 \\ \hline \end{array}$$

III. Aşağıdaki işlemler r tümleyeni yöntemiyle yapınız. İşlemin <u>her bir basamağını</u> gösteriniz.

a)
$$\begin{array}{rl} & (1\ 0\ 0\ 1\ 0)_2 \\ - & (0\ 1\ 0\ 1\ 1)_2 \\ \hline \end{array}$$

b)
$$\begin{array}{rl} & (1\ 0\ 0\ 1\ 0)_2 \\ - & (1\ 1\ 0\ 1\ 1)_2 \\ \hline \end{array}$$

IV. Aşağıdaki çıkarma işlemlerini r-1 tümleyeni kullanarak yapınız. İşlemin <u>her bir basamağını</u> gösteriniz.

a)
$$\begin{array}{rl} & (1\ 1\ 1\ 0)_2 \\ - & (1\ 0\ 1\ 1)_2 \\ \hline \end{array}$$

b)
$$\begin{array}{rl} & (1\ 0\ 1\ 0\ 1\ 0)_2 \\ - & (1\ 1\ 0\ 0\ 1\ 1)_2 \\ \hline \end{array}$$

2.4 KODLAR

Sayısal elektronikte ve bilgisayarda sıkça kullanılan diğer bir kavram kodlardır. Kodlar kullanıcıları tarafından okunabilen değer taşıyıcılardır. Sadece rakamlarla kodlanmış kodlara nümerik kodlar, hem rakam hem karakter (noktalama, harf) ile kodlananlara ise alfa numerik kodlar denir.

2.4.1 NÜMERİK KODLAR

A. BCD (Binary Coded Decimal)

Onluk sistemdeki her bir basamağın ikilik sistemde karşılığının bulunmasıyla yazılan koddur. Basamak ağırlığı 8421 şeklindedir.
Örnek 1) $(73)_{10}$ -> 7:0111 ve 3:0011 kodlar yan yana getirilir. BCD kodu: $(01110011)_{BCD}$
Örnek 2) $(010111010001)_{BCD}$ =0101 1001 0001 = 591

B. GRAY KODU

Gray kodunda basamak değeri yoktur. Her bir sayı artışı için sadece 1 bit değişir. İkili sayının Gray koduna dönüşümü için MSB'nin önüne 0 konur ve sayılar soldan başlanarak 2 bitler halinde toplanarak yazılır.

Örnek 1) $(10011)_2$ sayısını Gray kodu: $(01010)_{GRAY}$

0	1	0	0	1	1
	0+1	1+0	0+0	0+1	1+1
	1	1	0	1	0

Gray kodu ikili sayıya çevirmek için ise en soldaki basamak direk alınır. Bu sayı ile sonraki sayılar toplanır.

Örnek 2) $(01010)_{GRAY}$ =$(10011)_2$

1	1	0	1	0
1	1+1	0+0	0+1	1+0
1	0	0	1	1

C. +3 KODU

Onluk sistemdeki sayıyı ikili sistemle kodlamak için kullanılır. Her basamak 3 ile toplanır ve basamağın ikili kodu yazılır. +3 kodundan onluk sisteme dönüşüm için her bir 4 bit binary sayıdan 3 çıkarılır.

Örnek 1) 64 sayısı için: 6+3=10 4+3=7
1010 0111 $=(10100111)_{+3}$

Örnek 2) $(10100111)+3$ kodu için 1010-0011=0111=7 0111-0011=0100=4
Sonuç $(74)_{10}$

D. EŞİTLİK KODU

Kod gönderiminde hata olup olmadığını anlamak için kullanılır. Tek bittir. Çift eşitlik kodunda gönderideki 1lerin sayısı çifte tamamlanır (çift 1 varsa kod 0, tek bir varsa kod 1). Tek eşitlik kodunda ise 1lerin sayısı teke indirilir (çift 1 varsa kod 1, tek bir varsa kod 0).

Örneğin 011101110110000101101000101000010110010 kodunda **çift** sayıda 1 bulunur.

Bu durumda çift eşitlik kodu: 0 0111011101100001011010001010000010110010

Tek eşitlik kodu: 1 0111011101100001011010001010000010110010

E. AİKEN KODU

Basamak değeri 2421 olan koddur. 0-4 arasındaki sayılar ikili kod sistemine göre yazılırken 5-9 arasındaki sayılar bunların simetriğidir. Örneğin 5 ile 4, 6 ile 3 simetriktir.

Sayı	Aiken Kodu			Sayı
0	0000	⋘ ⋙ simetrik	1111	9
1	0001		1110	8
2	0010		1101	7
3	0011		1100	6
4	0100		1011	5

Örneğin 8 sayısını Aiken kodu ile kodlarsak;

2 4 2 1 ←basamak değeri
1 1 1 0

F. 5’te 2 KODU

Her onlu sayı için içerisinde sadece 2 tane 1 bulunduran 5 basamaklı kod sistemidir. Basamak ağırlığı 74210 şeklinde ilerler. 0 sayısının temsili için 11 değerini (11000), 2,4,7 sayısı içinde fazladan sıfır basamağını kullanır.

Örneğin 7 sayısının 5’te 2 kodu:

7	4	2	1	0	←basamak değeri
1	0	0	0	1	

8 sayısının 5’te 2 kodu:

7	4	2	1	0	←basamak değeri
1	0	0	1	0	

3 sayısının 5’te 2 kodu:

7	4	2	1	0	←basamak değeri
0	0	1	1	0	

2.4.2 ALFANÜMERİK KODLAR

A. ASCII KODU

Bu kodda sadece rakamsal değerler değil harfler de yazılabilir. 7 bitlik bu kodun onaltılı karşılı bulunarak bilgi kodlanır. ASCII kodu 7 bit olmasına rağmen ikili, onlu ve onaltılı sisteme kolay çevrilmesi açısından önüne 0 konur.

ASCII KODU							
Karakter	**8-bit**	**Onluk**	**Onaltılık**	**Karakter**	**8-bit**	**Onluk**	**Onaltılık**
boşluk	00100000	032	020	**P**	01010000	080	050
!	00100001	033	021	**Q**	01010001	081	051
"	00100010	034	022	**R**	01010010	082	052
#	00100011	035	023	**S**	01010011	083	053
$	00100100	036	024	**T**	01010100	084	054
%	00100101	037	025	**U**	01010101	085	055
&	00100110	038	026	**V**	01010110	086	056
'	00100111	039	027	**W**	01010111	087	057
(	00101000	040	028	**X**	01011000	088	058
)	00101001	041	029	**Y**	01011001	089	059
*	00101010	042	02A	**Z**	01011010	090	05A
0	00110000	048	030	**`**	01100000	096	060
1	00110001	049	031	**a**	01100001	097	061
2	00110010	050	032	**b**	01100010	098	062
3	00110011	051	033	**c**	01100011	099	063
4	00110100	052	034	**d**	01100100	100	064
5	00110101	053	035	**e**	01100101	101	065
6	00110110	054	036	**f**	01100110	102	066
7	00110111	055	037	**g**	01100111	103	067
8	00111000	056	038	**h**	01101000	104	068
9	00111001	057	039	**i**	01101001	105	069
:	00111010	058	03A	**j**	01101010	106	06A
;	00111011	059	03B	**k**	01101011	107	06B
<	00111100	060	03C	**l**	01101100	108	06C
=	00111101	061	03D	**m**	01101101	109	06D
>	00111110	062	03E	**n**	01101110	110	06E
?	00111111	063	03F	**o**	01101111	111	06F
@	01000000	064	040	**p**	01110000	112	070
A	01000001	065	041	**q**	01110001	113	071
B	01000010	066	042	**r**	01110010	114	072
C	01000011	067	043	**s**	01110011	115	073
D	01000100	068	044	**t**	01110100	116	074
E	01000101	069	045	**u**	01110101	117	075
F	01000110	070	046	**v**	01110110	118	076
G	01000111	071	047	**w**	01110111	119	077
H	01001000	072	048	**x**	01111000	120	078
I	01001001	073	049	**y**	01111001	121	079
J	01001010	074	04A	**z**	01111010	122	07A
K	01001011	075	04B	**{**	01111011	123	07B
L	01001100	076	04C	**|**	01111100	124	07C
M	01001101	077	04D	**}**	01111101	125	07D
N	01001110	078	04E	**~**	01111110	126	07E
O	01001111	079	04F	**silme**	01111111	127	07F

Örneğin: SAYISAL boşluk ELEKTRONIK bilgisinin onaltılık ASCII kodu:

S	53	E	45
A	41	L	4C
Y	59	E	45
I	49	K	4B
S	53	T	54
A	41	R	52
Boşluk	20	O	4F
		N	4E
		I	49
		K	4B

KOD: 53 41 59 49 53 41 20 45 4C 45 4B 54 52 4F 4E 49 4B

Örnek: 01000100010001010101001001010011 ASCII kodundaki bilgi:

0100	0100	0100	0101	0101	0010	0101	0011
4	4	4	5	5	2	5	3
	44		45		52		53
	D		E		R		S

B. EBCDIC KODU

ASCII koduna benzer 8 bitlik bir koddur. Fakat ASCII kadar yaygın değildir.

2.4.3 KODLAMA SORULARI

I. $(853)_{10}$ bilgisini

a) BCD,

b) GRAY,

c) +3,

d) 5’te 2 kodlarına göre kodlayınız.

II. $(1234)_{10}$ bilgisi için yukarıdaki a,b,c,d basamakları tekrarlayınız.

III. 0100111101001011010101010101001100 ACSII kodunun içerdiği bilgi nedir?

IV. Aşağıdaki ASCII onaltılık kod ile gönderilen bilginin içeriğini bulunuz.

53	4F	4E	52	41	4B	49	20	4B	4F	4E	55	20	4B	41	50	49	4C	41	52

3 TEMEL MANTIK İŞLEMLERİ

3.1 LOJİK KAPILAR

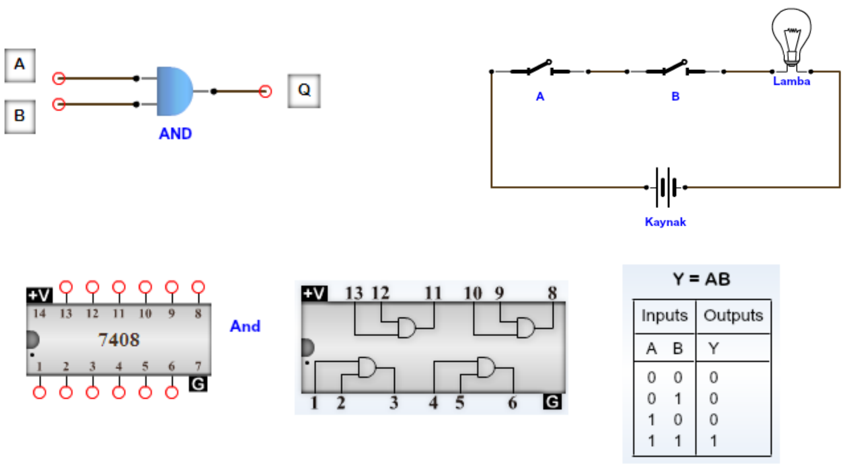
Lojik değişken girişlerle işlem yapıp bu sonucu çıkışına aktaran yapılar lojik kapılardır. Sayısal elektroniğin temelini oluştururlar. En basit, en sade programlama elemanlarıdır. Girişlerindeki bilgi lojik 1 ve lojik 0 olarak adlandırılır. Lojik 1 ve lojik sıfır uygulamada 5v -0v, gerilim var-yok, gerilim yüksek-düşük gibi değerler alırlar.

Lojik kapılar transistör, BJT, MOSFET, diyot v.b. elemanlarla gerçekleştirilirler. Günümüzde entegra haline getirilmişler sıkça kullanılır.

Lojik kapılarla gerçekleştirilen devrelere lojik devreler denir. Başlıca lojik kapılar aşağıda anlatılmıştır.

3.1.1. VE (AND) KAPISI

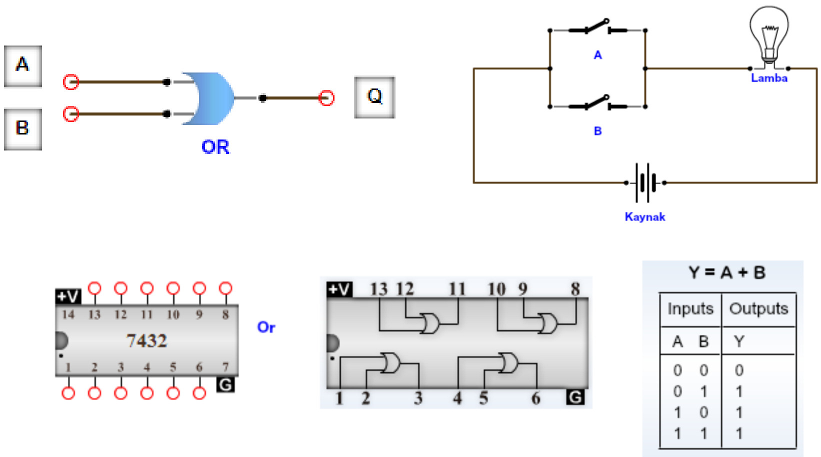
Girişinden uygulanan bilgiye ve (.) işlemi yapan kapıdır. Bu işlem çarpma işlemine benzetile bilinir. Girişlerinin bir tanesinin sıfır olması çıkışı sıfır yapar. Q=A.B ya da Q=AB şeklinde gösterilir. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir.



ŞEKİL 7: VE KAPISI

3.1.2. VEYA (OR) KAPISI

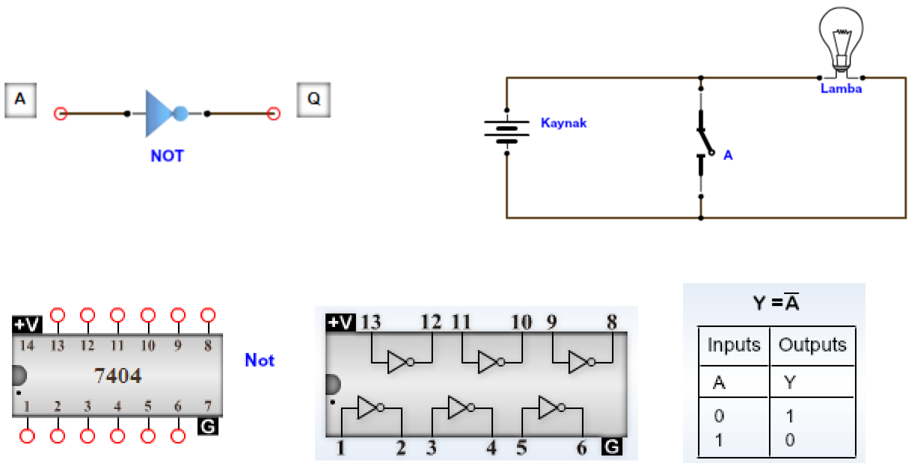
Girişlerinden gelen bilgilere veya işlemi yapar. Toplama işlemine benzer fakat toplamadan farklı olarak girişlerinden 1 tanesine 1 olsa bile çıkışı 1'dir. Q=A+B şeklinde gösterilir. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir.



Şekil 8: VEYA Kapisi

3.1.3. DEĞİL (NOT) KAPISI

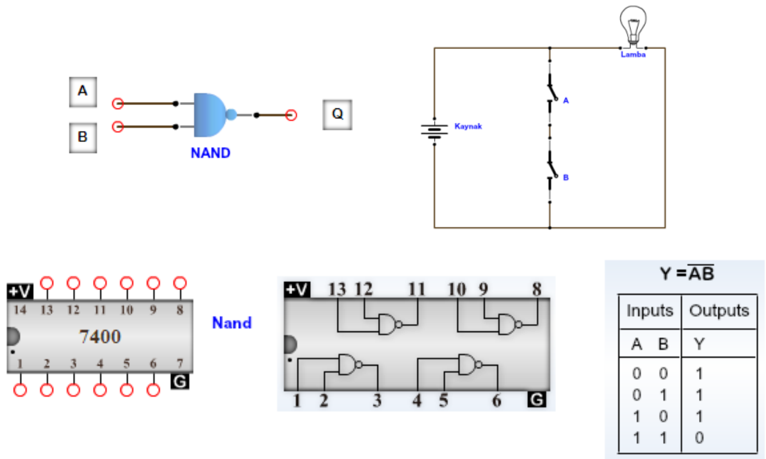
Tek girişli bu kapı, girişin tersini alır. Giriş 1 ise çıkış 0'dır, veya tam tersi. Q=$\bar{A}$ şeklinde gösterilir. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir.



Şekil 9: DEĞİL Kapisi

3.1.4. VEDEĞİL (NAND) KAPISI

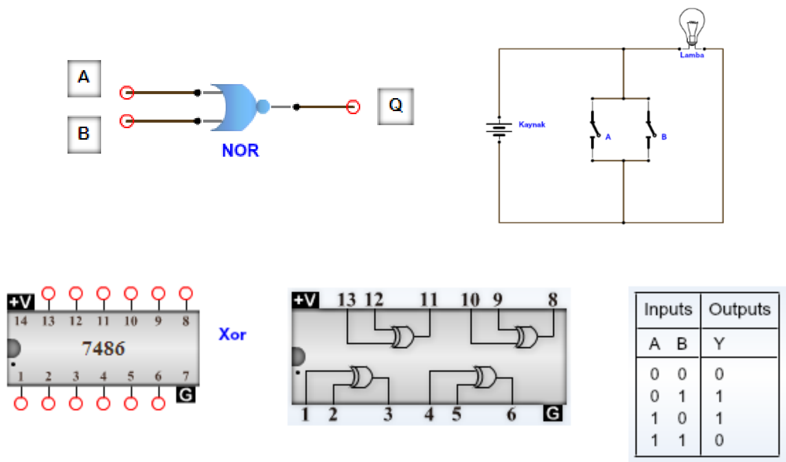
VE kapısının arkasına değil kapısı eklenmiş kapıdır. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir. Q= $\overline{A.B}$ veya Q= $\overline{AB}$ şeklinde gösterilir.



Şekil 10: VEDEĞİL Kapisi

3.1.5. VEYADEĞİL (NOR) KAPISI

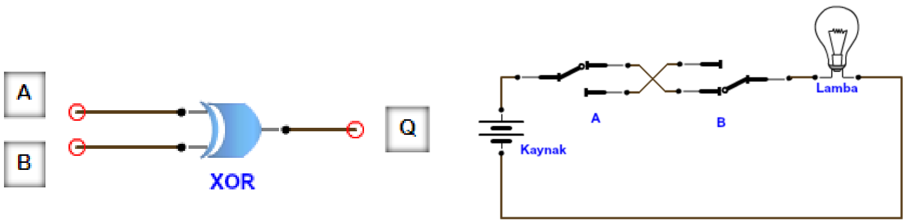
VEYA kapısının arkasına değil kapısı eklenmiş kapıdır. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir. Q=$\overline{A+B}$ şeklinde gösterilir.

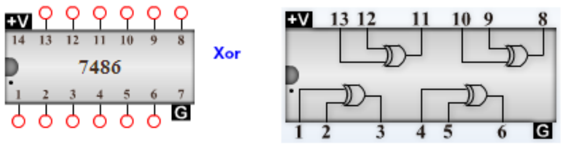


Şekil 11: VEYA DEĞİL Kapisi

3.1.6. ÖZEL VEYA (EX-OR) KAPISI

Bu kapı özel yapılı bir kapıdır. Girişlerinin 0 ya da 1 olmasından bağımsız olarak iki girişi birbirine karşılaştırarak işlem yapar. Girişler birbirinden farklı ise çıkışı 1'dir. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir. Q=A⊕B şeklinde gösterilir.



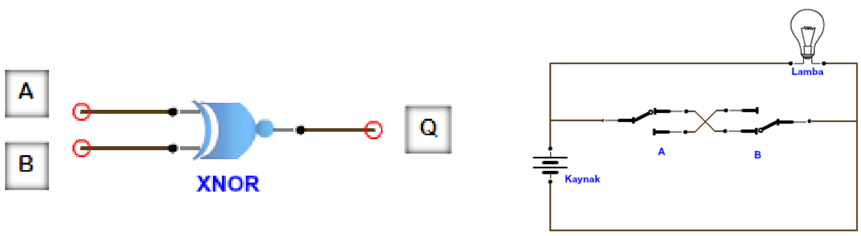


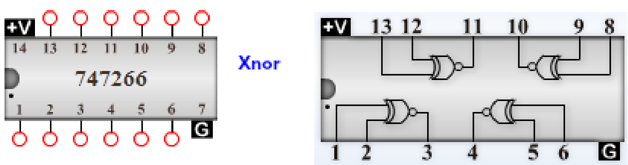
Inputs		Outputs
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

ŞEKİL 12: ÖZEL VEYA KAPISI

3.1.7. ÖZEL VEYA DEĞİL (EX-NOR) KAPISI

ÖZEL VEYA kapısının değillenmiş şeklidir. Girişleri birbirinin aynı ise çıkış 1'dir. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir. Q=$\overline{A \oplus B}$ şeklinde gösterilir.



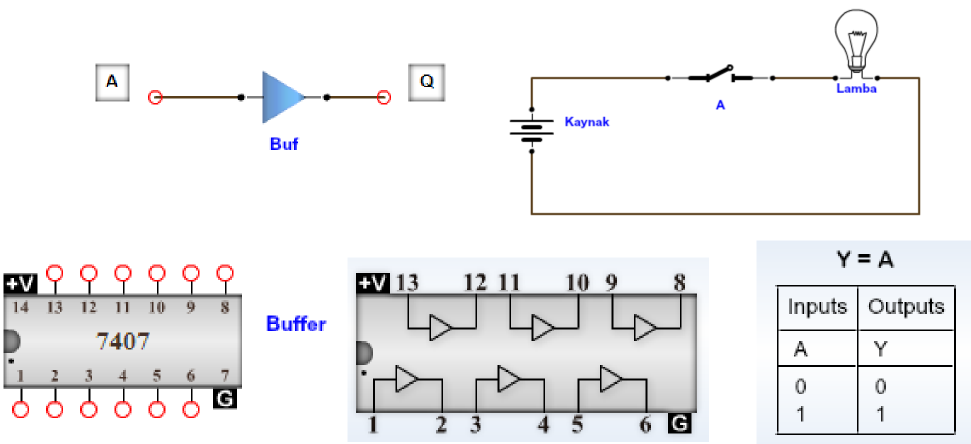


Inputs		Outputs
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

ŞEKİL 13: ÖZEL VEYA DEĞİL KAPISI

3.1.8. TAMPON (BUFLİP-FLOPER) KAPISI

Tampon kapısı aslında bir kapı gibi işlem yapmaz. Girişi olduğu gibi çıkışa aktarır. Devre tasarımında devre girişlerinin eşit sayıda kapıdan geçmesi için kullanılır. Hassas devrelerde kapıların neden olduğu gecikmeyi eşitler. Kapının şematik simgesi, anahtarlarla yapılan temsili devresi, entegre görünümü, entegre iç görünümü ve doğruluk tablosu şekilde gösterilmiştir.



ŞEKİL 14: TAMPON KAPISI

3.2 LOJİK DİYAGRAM TASARIMI

Sayısal mantıkla bir devre gerçekleştirilmek isteniyorsa öncelikle bir lojik ifade elde edilmelidir. Elde edilen lojik ifade, lojik kapılarla (ve lojik tümleşik yapılarla) gerçekleştirile bilinir.

Örneğin bir binada duman detektörü aktif olmuşsa ve yangın alarmı butonuna basılmışsa acil çıkış lambaları yansın ve asansörler kullanım dışı olsun. Girişlerden yalnızca biri aktif olduğu zaman iş güvenliği ekibine bir uyarı sinyali gönderilsin.

girişler		çıkış	
	Lojik ifade		Lojik ifade
Duman detektörü	A	Lambalar	X
Yangın alarmı	B	Asansör	Z
		Uyarı sinyali	Q

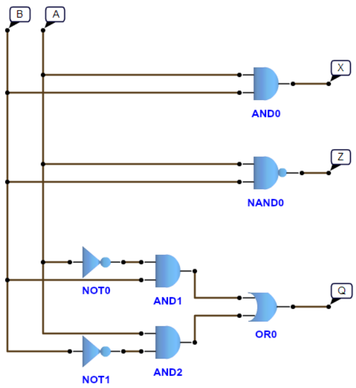
1. Öncelikle problemi (devrede olması gerekenler) çözümlenir.

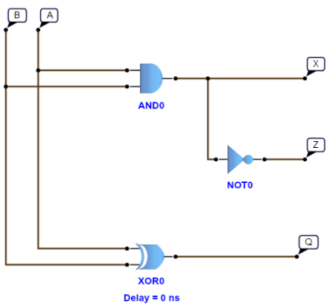
A ve B aktifse X aktif
A ve B aktifse Z pasif
A aktif B pasif veya B aktif A pasif ise Q aktif

2. Problemle ilgili lojik ifade bulunur.

$$X=A.B$$
$$Z=\overline{A.B}$$
$$Q=\overline{A}.B + A.\overline{B}$$

3. Lojik ifade lojik kapılar kullanılarak gerçekleştirilir.

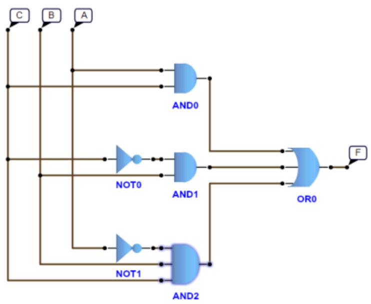




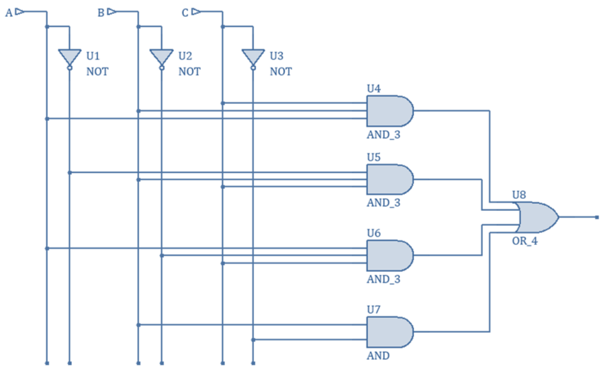
Aynı lojik ifadenin diğer bir gösteriliş şeklide yandaki şekildedir. Bu diyagramın karşılığını lojik ifade şeklinde yazarsak:

$$X=A.B$$
$$Z=\overline{X}$$
$$Q=A \oplus B$$ olur.

Örnek 1) **F= A.C+B.C'+A'BC** ifadesini lojik kapılarla gerçekleştirelim.



Örnek 2) **Q=**$\boldsymbol{ABC + \overline{A}BC + A\overline{B}C + B\overline{C}}$ ifadesini lojik kapılarla gerçekleştirelim.



3.3 LOJİK EŞİTLİKLERİN NAND VE NOR KAPILARIYLA GERÇEKLENMESİ

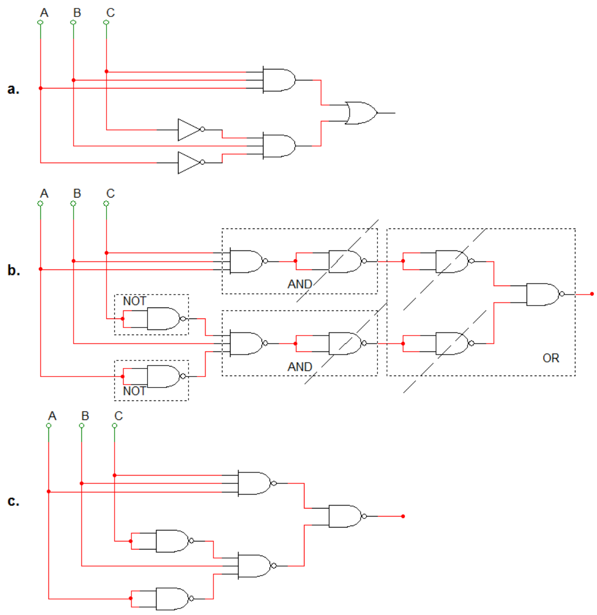
Sayısal elektronikte, her bir kapı farklı entegreler içinde yer aldığından, tek tip kapı ile devre tasarlamak daha ekonomiktir. Bunun için de genellikle VEDEĞİL veya VEYADEĞİL kapıları tercih edilir. Bir VEDEĞİL veya VEYADEĞİL entegresi içinde 4 adet kapı bulunur.

Lojik kapılardan VEDEĞİL VEYADEĞİL kapılarına dönüşüm için aşağıdaki kurallar geçerlidir:

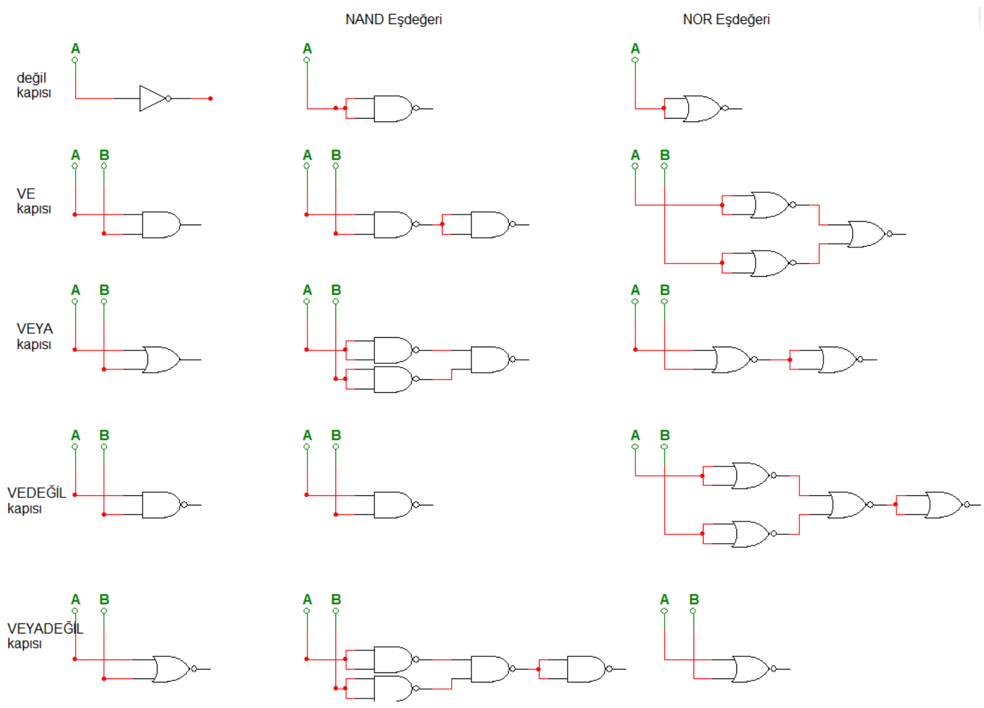
I. VEDEĞİL veya VEYADEĞİL kapıları birleştirilip kullanılırsa bu kapı DEĞİL kapısına dönüşür.
II. İki VEDEĞİL kapısı kullanılarak ve kapısı elde edilir.
III. Kapı dönüşümleri için DeMORGAN kurallarından yararlanılır.

Örnek olarak **ABC+A'BC** devresini **NAND** kapılarıyla gerçekleştiriniz.

a. Devrenin gerekli lojik kapıları kullanılarak çizimi.
b. Her bir kapı için NAND eşdeğerinin yerine konularak devrenin çizimi
c. Devrenin sadeleştirilmesi (değilinin değili kendisidir-artarda iki değil kapısı varsa silinir).



Şekil 15: Örnek NAND kapıları devresi



ŞEKİL 16: LOJİK KAPILARIN NAND VE NOR EŞDEĞERİ

Örneğin VE kapısı VEDEĞİL kapıları ile elde edilmek isteniyorsa:

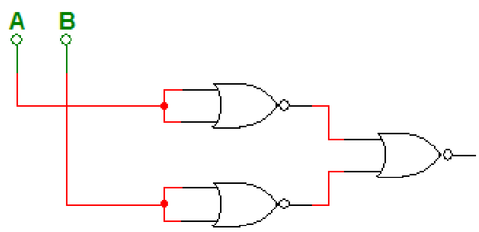
A.B girişlerimiz olsun. Bu girişlere deMorgan kuralını uygular üzerine iki DEĞİL eklersek (bir kapının değilinin değili kendisidir):

$$\overline{\overline{A.B}}$$

Bu değillerden birini parçalara ayırırsak aradaki işaret değişir (VE (.) VEYA (+) olur).

$$\overline{\overline{A}+\overline{B}}$$

VE kapısı VEYADEĞİL kapısına dönüştü:



4 BOOLEAN KANUNLARI

Lojik kapılardan oluşan elektronik devreler belli yöntemlerle daha basit hale getirile bilinir. Matematiksel sadeleşme kuralları olduğu gibi lojik ifadelerin de sadeleşme kuralları vardır. bu kurallardan en çok kullanılanı kendine özgü sistemi, kuralları, eşitlikleriyle Boolean Kanunlarıdır. Bu kurallar girişi ve girişe bağlı çıkışı değiştirmeden sadece devre kısmını sadeleştiren yapılardır.

4.1. TEMEL BOOLEAN KANUNLARI

4.1.1. Temel İfadeler

Boolean matematiğinde VE kapısını çarpma, VEYA kapısını toplama gibi düşünebilir.

a. Toplamada etkisiz eleman (0)

A+0=A

b. Çarpmada etkisiz eleman (1)

A.1=A

c. Toplamada birim eleman (1)

A+1=1

d. Çarpmada yutan eleman (0)

A.0=0

e. Ters eleman

A'veya $\bar{A}$ şeklinde gösterilir. Eğer A=1 ise A'=0

A=0 ise A'=1

Değilinin değili kendisidir. A''=A ya da $\bar{\bar{A}}$=A

f. Toplama ve çarpma işlemleri

A+A'=1

A.A’=0

A+A=A

A.A=A

4.1.2. Sabit kuvvetlilik

A+A+A+A….=A ve A.A.A.A…=A

4.1.3. Değişme kanunu

A+B=B+A ve A.B=B.A

4.1.4. Birleşme kanunu

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

A.(B.C)=(A.B).C=A.B.C

4.1.5. Dağılma kanunu

A.(B+C)=(A.B)+(A.C)

A+(B.C)=(A+B).(A+C)

4.1.6. Yutma kanunu

A+A.B=A

A.(A+B)=A

4.1.7. Basitleştirme kanunu

A+A’.B=A+B

A(A’+B)=A.B

4.1.8. De MORGAN KANUNLARI

$\overline{A.B} = \overline{A} + \overline{B}$ $\overline{A + B} = \overline{A}.\overline{B}$

4.2. LOJİK İFADELERİN BOOLEAN KURALLARI İLE SADELEŞTİRİLMESİ

Önceki bölümde anlattığımız kuralları kullanılarak karmaşık lojik ifadeler daha sade hale getirile bilinir. Böylelikle yapılacak devrede ekonomiklik sağlanırken, devrenin anlaşılması, hata ayıklanması daha kolay hale gelir.

Örnek 1) lojik ifadede kendisi ve değili olan elemanlar yok edilmeye çalışılır.

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

$\overline{A}C(\overline{B} + B) + A\overline{B}$=

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

Örnek 2) A'B'C'+A'B'C+ABC'+AB'C'=

A'B'(C'+C)+AC'(B+B')= A'B'+AC'

Örnek 3) A'B+A+A.B=

B(A'+A)+A=B+A

Örnek 4) AB'+A'B'C=

AB'(C+C')+A'B'C= (üç değişken)

AB'C+AB'C'+A'B'C+AB'C= (birleşme özelliği için ilk ifade tekrarlandı)

B'C(A+A')+AB'(C+C')=

B'C+AB'=

B'(A+C)

Örnek 5) $\overline{A + \overline{BC}}$= (de Morgan)

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

$\overline{A} + BC$

Örnek 6) $\overline{AB + CD}$ de Morgan

$\overline{AB} + \overline{CD}$ de Morgan

$(\overline{A} + \overline{B}).(\overline{C} + \overline{D})$ de Morgan

Örnek 7) $\overline{(\overline{A} + \overline{B} + \overline{C}).(\overline{A} + B + \overline{C}).(A + B + \overline{C})}$=

$\left(\overline{\overline{A} + \overline{B} + \overline{C}}\right) + \left(\overline{\overline{A} + B + \overline{C}}\right) + (\overline{A + B + \overline{C}})$=

$\left(\overline{\overline{A}}.\overline{\overline{B}}.\overline{\overline{C}}\right) + \left(\overline{\overline{A}}.\overline{B}.\overline{\overline{C}}\right) + (\overline{A}.\overline{B}.\overline{\overline{C}})$=

$(ABC) + \left(A\overline{B}C\right) + (\overline{A}\overline{B}C)$

5. KARNOUGH HARİTALARI

Lojik ifadeleri boolean kuralları haricinde, çizelge olarak sadeleştirmek için kullanılır. Doğruluk tabloları çizelge üzerinde işaretlenerek sadeleştirme yapılır. 2,3,4,5... değişkenli olarak devreye göre değişiklik göstermektedir. 2^n hücre içerir. Örneğin 3 değişkenli bir harita 2^3=8 hücre, 4 değişkenli bir harita 2^4=16 hücre içerir.

5.1. ÜÇ DEĞİŞKENLİ KARNOUGH HARİTALARI

Şekildeki gibi 3 değişkenli karnough haritasında değişkenler ABC olası her bir duruma bir hücre gelecek şekilde yerleştirilir. Örneğin denklemde elde edilen ABC’ ifadesi için kırmızı 1 olan hücre

AB’C için mavi 1 olan hücre işaretlenir. AB’C’ için ise yeşil 1 olan hücre işaretlenir. Bu hücrelerin ortak kısımları bu ifadenin sadeleştirilmiş halini verir.

Kırmızı-yeşil: AC’ ve yeşil mavi grup AB’ dir. Denklem:

ABC’+AB’C+AB’C’=AC’+AB’= A(B’+C’)

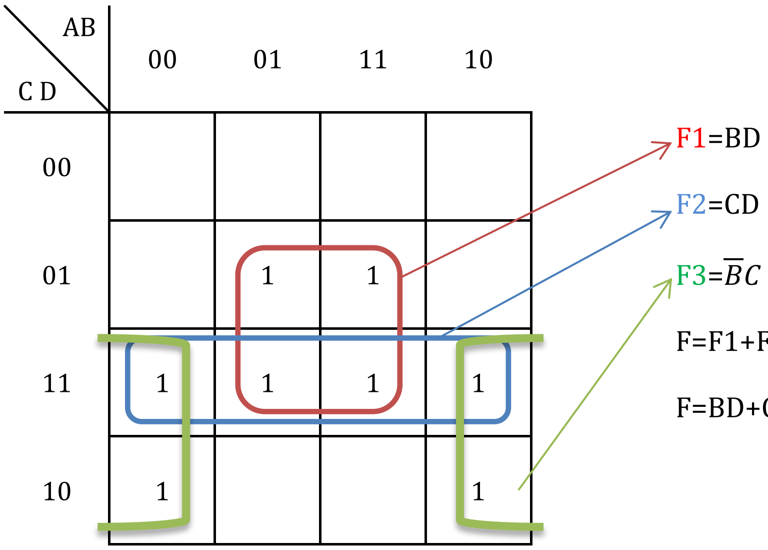
AB \ C	00 $\overline{A}\overline{B}$	01 $\overline{A}B$	11 AB	10 $A\overline{B}$
0 $\overline{C}$			1	1
1 C				1

5.2 DÖRT DEĞİŞKENLİ KARNOUGH HARİTALARI

- Bir karnough haritasında tüm hücreler 1 ise sadeleştirilmiş fonsiyon “1” dir.
- Birbirine komşu olan hücreler ve karşı komşular gruplanabilir.
- Bir hücre birden fazla gruplandırmada kullanılabilir.
- Haritada hedef en büyük grubu oluşturmaktır. 2 tane 2 li grup yerine 1 tane 4 lü grup daha sadedir.

AB \ CD	00	01	11	10
00	$\overline{ABCD}$	$\overline{A}B\overline{CD}$		
01				
11			$ABCD$	
10				$A\overline{B}C\overline{D}$

Örnek 1) F= $\overline{A}\overline{B}CD + \overline{A}BCD + ABCD + A\overline{B}CD + \overline{A}B\overline{C}D + AB\overline{C}D + \overline{A}\overline{B}C\overline{D}$ +A $\overline{B}$ C $\overline{D}$ ifadesini sadeleştiriniz.



F=F1+F2+F3

F=BD+CD+B’C

Örnek 2) Aşağıda doğruluk tablosu verilen ifadeyi Karnough haritasıyla sadeleştiriniz:

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
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

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

F=A’B’+AB+C’D’

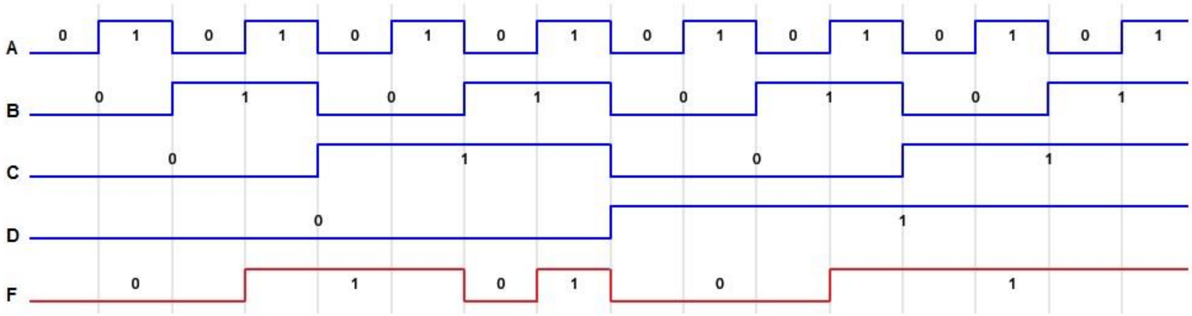
Örnek 3) **F=∑(0,2,4,6,8,10,12,14)** şeklindeki bir minterm ifadesini Karnough haritasıyla sadeleştiriniz.

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

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

F=D

Örnek 4) aşağıdaki dalga şekillerine göre girişleri olan ve çıkışı verilen lojik ifadeyi önce Karnough haritalarıyla sadeleştiriniz ve sonra kapılarla gerçekleştiriniz.



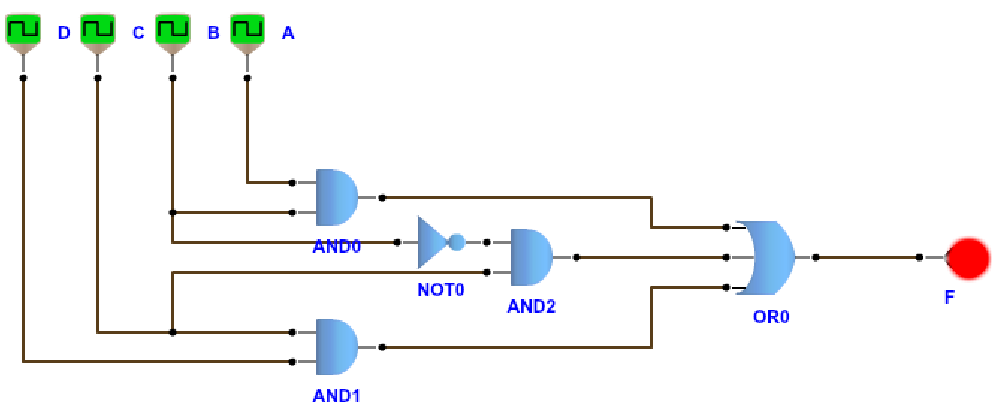
Dikkat bu soruda MSB-D ve LSB-A bitidir.

Çıkışın 1 olduğu durumlar:

D	C	B	A	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	
0	1	1	1	1
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

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

F=CD+BC’+AB



ÇALIŞMA SORULARI (BOOLEAN KURALLARI, KARNOUGHT HARİTALARI, LOJİK DEVRE TASARIMI)

1.a. F=XYZ+XY'Z+XY'Z' ifadesini lojik kapılarla gerçekleştiriniz.

1.b. F=XYZ+XY'Z+XY'Z' ifadesi boolean cebrini kullanarak sadeleştiriniz.

1.c. F=XYZ+XY'Z+XY'Z' ifadesini karnought haritalarıyla sadeleştiriniz.

1.d. F=XYZ+XY'Z+XY'Z' ifadesinin sadeleştirilmiş eşitliğini lojik kapılarla gerçekleştiriniz.

2.a. Boolean cebrinde yutma kanununu ispatlayınız.

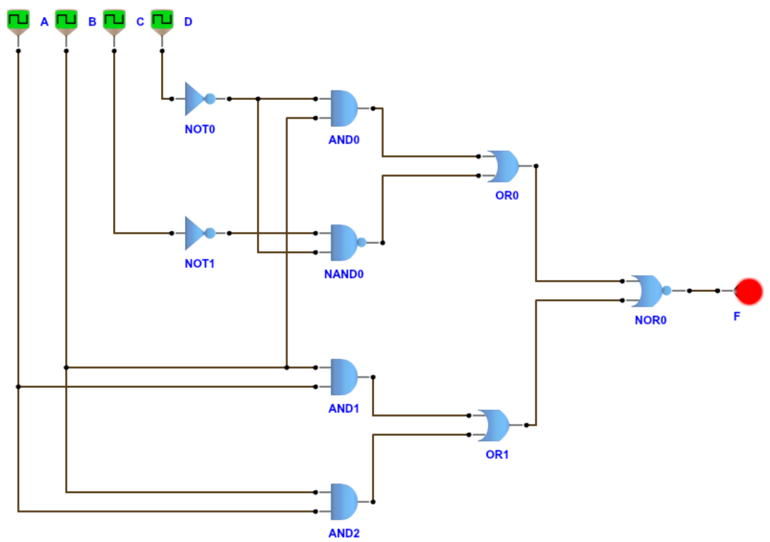
2.b. Boolean cebrinde basitleştirme kanununu ispatlayınız.

3.a. Q=∑(0,1,2,3) mintermini A,B,C,D girişlerini kullanarak yazınız.

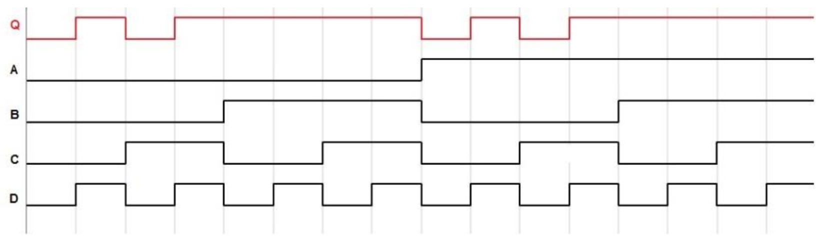
3.b. 2.a şıkkındaki eşitliği Boolean cebrini kullanarak sadeleştiriniz.

3.c. 2.a şıkkındaki eşitliği Karnough haritalarını kullanarak sadeleştiriniz.

4.a. Aşağı şekildeki lojik kapılarla oluşturulan devreyi sadeleştirip tekrar tasarlayınız.



5.a. Q çıkışında istenilen çıkışı veren devreyi en az sayıda kapı kullanarak tasarlayınız.



6. BİLEŞİK MANTIK DEVRELERİ

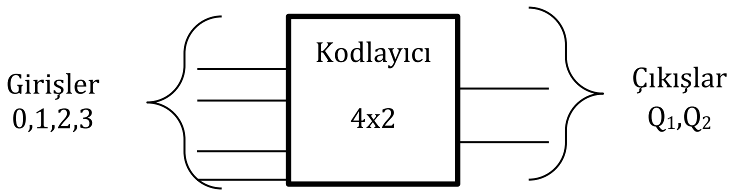
Dijital sistemlerde çıkışların durumu o anki girişlerin durumuyla kararlaştırılan devrelere bileşik (combinational) devreler denir. Çok sayıda girişleri ve/veya çıkışları olabilir. Bu devreler belli amaçlarla gerçekleştirilirler.

6.1. KODLAYICILAR (ENCODERS)

Kodlayıcılar rakam ve kelimeleri ikili sayı sistemine çevirerek makineler için anlamlı kodlar oluştururlar. Örneğin 9 bilgisini girdiğimizde bunun karşılığı binary sayıyı (1001) veren devrelerdir. Hazır IC (Integrated Circuit) (tümleşik entegre) devreleri olduğu gibi (ör. 74147) kapılarla da tasarlanabilir.

Dört girişli bir kodlayıcı devrede 4 tane giriş onluk sistemdeki sayıları (0,1,2,3) temsil eder. Bu sayıların ikili sistemde karşılığı $2^2=4$, 2 çıkışla verilir.

0	1	2	3	Q_2	Q_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1



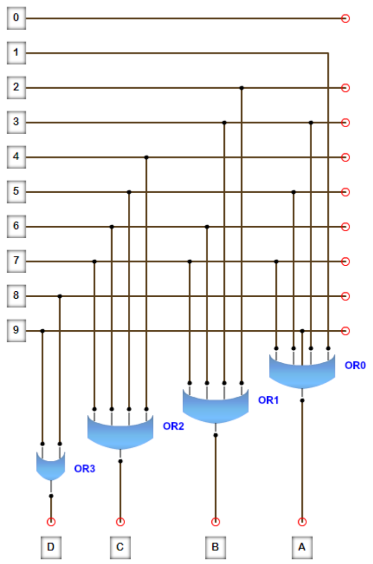
Hazır entegre devrelerde (74147-74148 gibi)öncelikli kodlayıcılar kullanılırlar. Öncelikli kodlayıcılar girişlerinde en yüksek değerlikli bitteki sıfıra göre işlem yaparlar. Örneğin 5 numaralı giriş aktif yapıldığında 5 numaraya 0 aynı anda 6,7,8,9 numaralara 1 (çünkü bu sayılar 5 den daha yüksek değerlidir) girişi uygulandığında çıkış BCD 5'in karşılığı 0101 in 1 tümleyeni 1010 olur. Burada 5den düşük değerlikli girişler önemsizdir (1 veya 0 farketmez).

	D9	D8	D7	D6	D5	D4	D3	D2	D1	Q4	Q3	Q2	Q1
0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	0	1	1	1	0
2	1	1	1	1	1	1	1	0	X	1	1	0	1
3	1	1	1	1	1	1	0	X	X	1	1	0	0
4	1	1	1	1	1	0	X	X	X	1	0	1	1
5	1	1	1	1	0	X	X	X	X	1	0	1	0
6	1	1	1	0	X	X	X	X	X	1	0	0	1
7	1	1	0	X	X	X	X	X	X	1	0	0	0
8	1	0	X	X	X	X	X	X	X	0	1	1	1
9	0	X	X	X	X	X	X	X	X	0	1	1	0

Onlu sistemden BCD'ye dönüştürme işlemi doğruluk tablosu oluşturarak yapalım.

	Çıkışlar			
Girişler	D	C	B	A
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

$A=\sum(1,3,5,7,9)$
$B=\sum(2,3,6,7)$
$C=\sum(4,5,6,7)$
$D=\sum(8,9)$



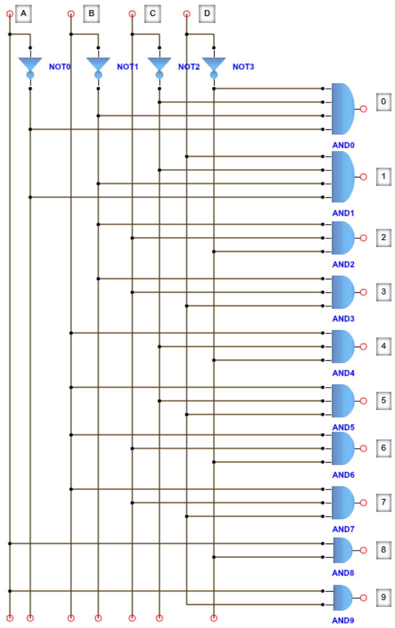
6.2. *KOD ÇÖZÜCÜLER (DECODERS)*

Kod çözücüler kodlayıcı devresini tersini yaparak 2 sayı sistemindeki bilgiyi onluk sisteme çevirir. N sayıda girişi olan bir sistemin çıkışı 2^N sayıdadır.

2x4 kod çözücü doğruluk tablosu:

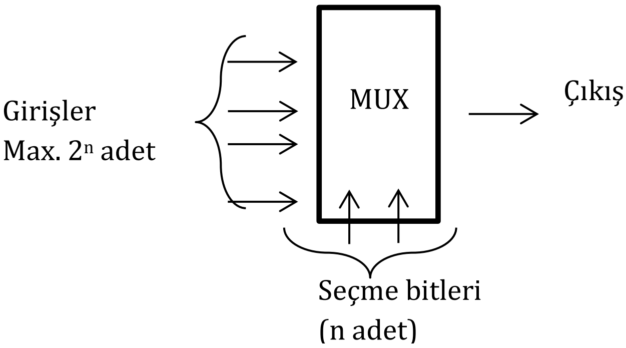
B	A	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

A	B	C	D	Q	LOJİK İFADE
0	0	0	0	0	Q_0=A'B'C'D'
0	0	0	1	1	Q_1=A'B'C'D
0	0	1	0	2	Q_2=B'CD'
0	0	1	1	3	Q_3=B'CD
0	1	0	0	4	Q_4=BC'D'
0	1	0	1	5	Q_5=BC'D
0	1	1	0	6	Q_6=BCD'
0	1	1	1	7	Q_7=BCD
1	0	0	0	8	Q_8=AD'
1	0	0	1	9	Q_9=AD



6.3. *BİLGİ SEÇİCİLER- ÇOKLAYICILAR (MULTIPLEXERS)*

Çok sayıda girişten yalnızca birini seçerek çıkışa aktaran devrelere bilgi seçiciler denir. Bilgi seçiciler çıkış bilgisini seçiciye bağlı olarak yeniler. Giriş sayısı 2^n=N olan bir devrede seçici sayısı n olmalıdır. Seçiciler-çoklayıcılar kısaca MUX diye gösterilirler.



Örneğin 8x1 bilgi seçicide 8 giriş bulunur. Bu girişler seçme bitleriyle seçilerek çıkışa aktarılır. Örneğin çıkışın I5 olması isteniyorsa seçme bitleri 101 olmalıdır.

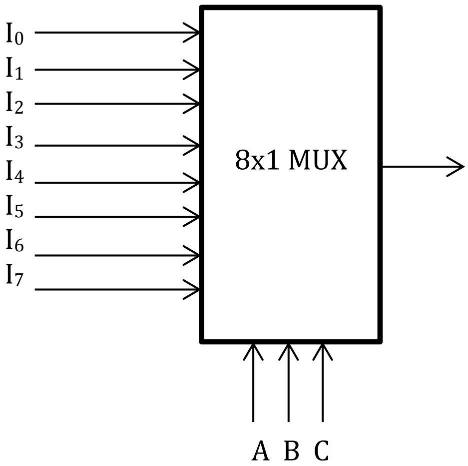
Örnek: 8x1 MUX devresini A,B,C seçme bitlerini kullanarak gerçekleyiniz.

Öncelikle doğruluk tablosu ve çıkış fonksiyonu bulunur.

Seçme bitleri			Çıkış
A	B	C	Q
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

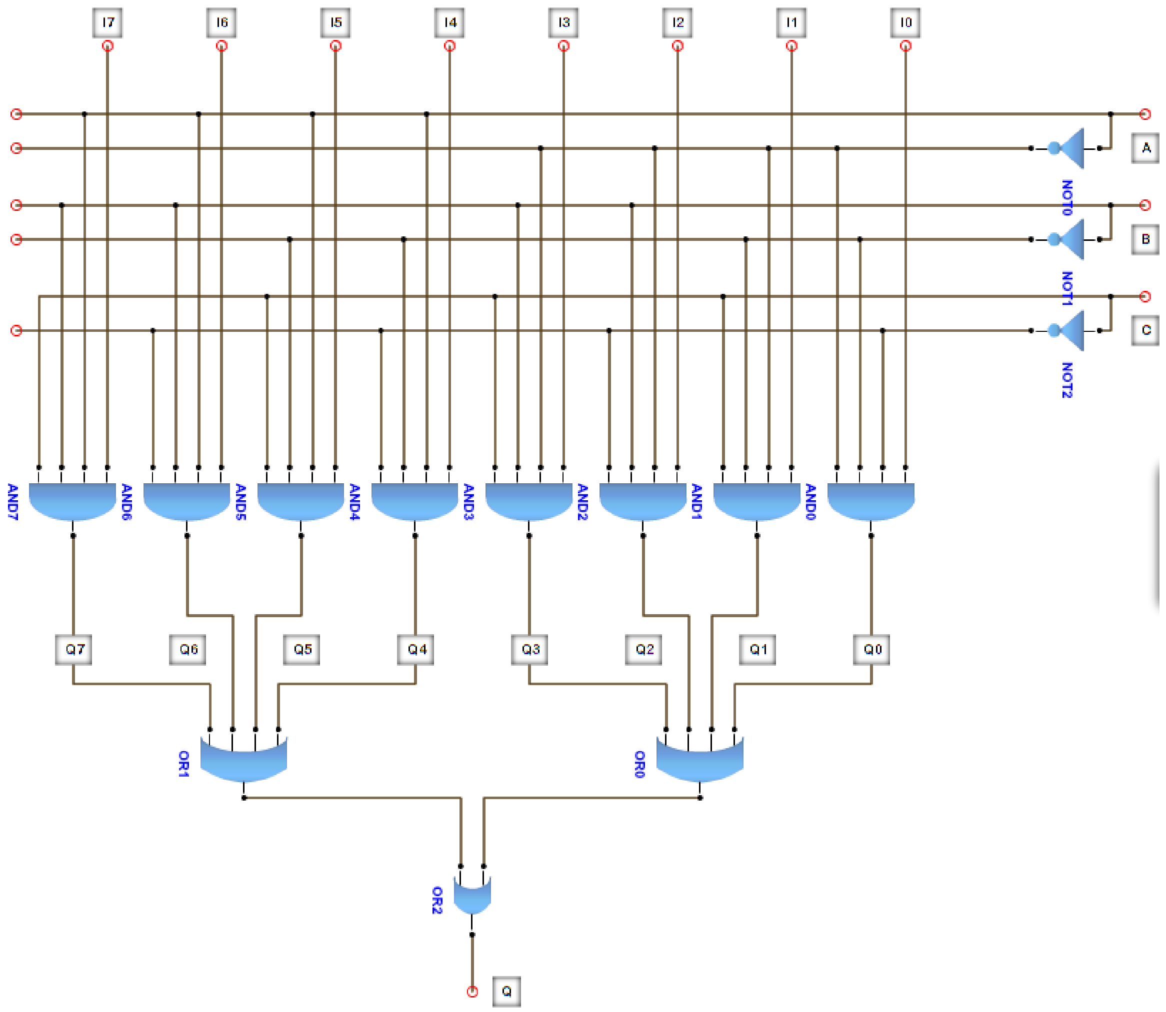
Q=I_0A'B'C'+I_1A'B'C+I_2A'BC'+I_3A'BC+I_4AB'C'+I_5AB'C+I_6ABC'+I_7ABC

Sonra devre şema halinde çizilir.



$Q=I_0A'B'C'+I_1A'B'C+I_2A'BC'+I_3A'BC+$
$I_4AB'C'+I_5AB'C+I_6ABC'+I_7ABC$

İstenirse bu devrenin eşitliği kapılar kullanılarak çizilir.

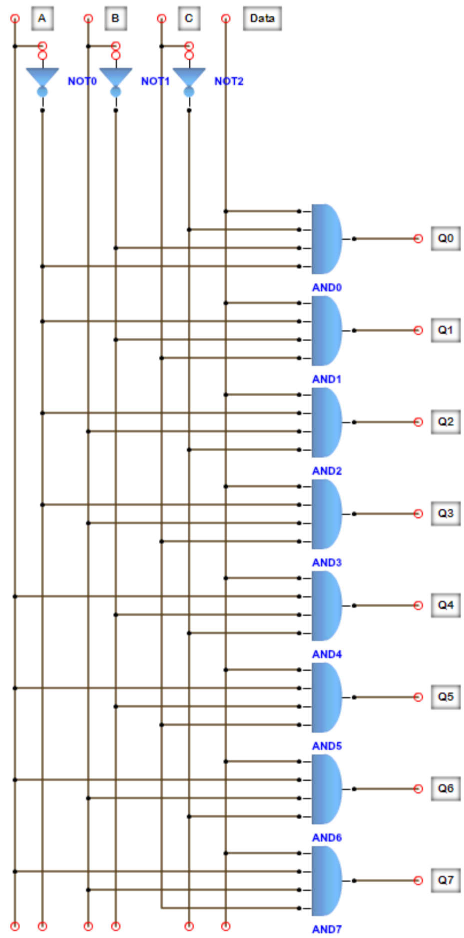


6.4. BİLGİ DAĞITICILAR (DEMULTIPLEXERS)

DEMUX bir kaynaktan aldığı tek bir bilgiyi seçme bitleri yardımıyla birden çok çıkışa aktarabilir.

1'den 8'e bilgi dağıtıcının tasarımı aşağıdaki gibi gerçekleştirilir.

Giriş	Seçme Bitleri			Çıkışlar							
Data	A	B	C	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
I	0	0	0	0	0	0	0	0	0	0	1
I	0	0	1	0	0	0	0	0	0	1	0
I	0	1	0	0	0	0	0	0	1	0	0
I	0	1	1	0	0	0	0	1	0	0	0
I	1	0	0	0	0	0	1	0	0	0	0
I	1	0	1	0	0	1	0	0	0	0	0
I	1	1	0	0	1	0	0	0	0	0	0
I	1	1	1	1	0	0	0	0	0	0	0



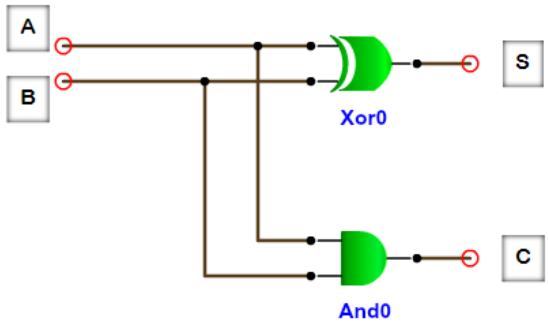
6.5. ARİTMETİK İŞLEMDEVRELERİ

Toplama, çıkarma, çarpa ve bölme yapan dijital devrelere aritmetik işlem devreleri adı verilmektedir. Bu devrelerden bazı hazır halde IC entegre olarak bulunmaktadır.

Toplayıcı devresi

İki bitlik sayının toplanmasını yarım toplayıcı, üç bitlik sayının toplanması ise tam toplayıcı devresiyle gerçekleştirilir. Toplama işleminde toplam ve elde çıkışları elde edilir. Örneğin 1+1 işleminin sonucu toplam 0 elde 1'dir.

Yarım toplayıcı (Half Adder-HA)



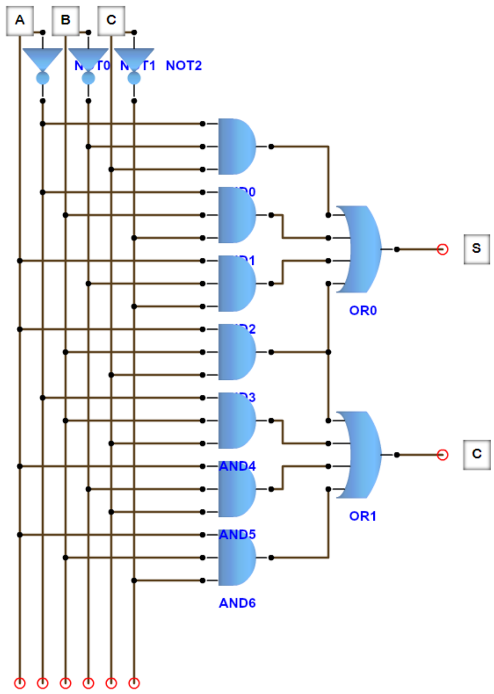
A	B	Toplam Sum -S	Elde Carry-C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

S=A’B+AB’

C=AB

Yarım toplayıcı devresinde toplama ve elde çıkışı vardır. 2 bitlik sayıları toplayabilir. Toplama kuralları geçerlidir. Girişler 1 ve 1 olduğu zaman çıkışında elde 1 olur.

Tam Toplayıcı (Full Adder-FA)

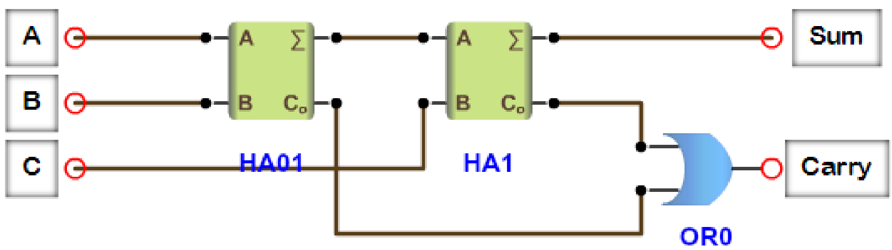


A	B	C	Toplam Sum -S	Elde Carry-C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S=A’B’C+A’BC’+AB’C’+ABC

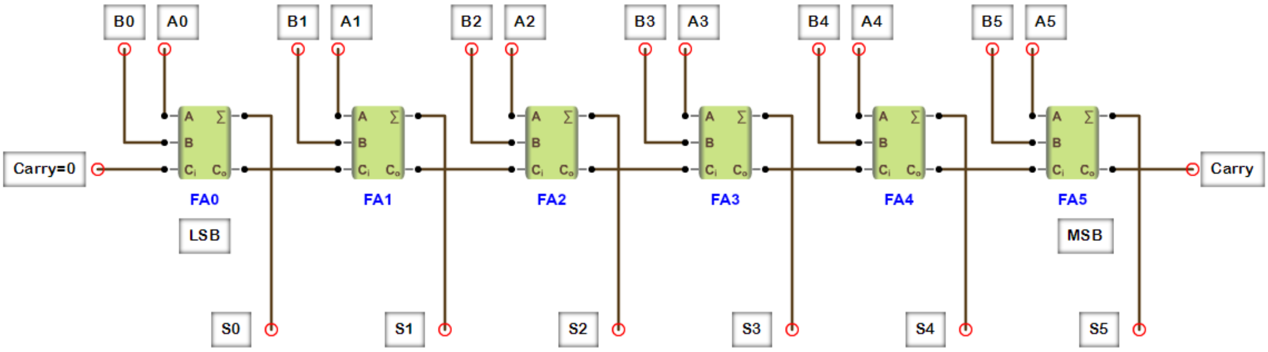
C=A’BC+AB’C+ABC’+ABC

Tam toplayıcı devresini gerçekleştirmenin diğer bir yolu ise iki yarım toplayıcı kullanmaktır. İlk yarım toplayıcının çıkışı diğerinin girişine verilir. Yani önce 2 bit toplanır ve toplamına 1 bit daha eklenerek 3 bit toplanmış olur.



Paralel Toplayıcı

Çok bitli sayıları toplayabilen yapıdır. Hesap makineleri ve bilgisayarlar tarafından kullanılırlar. Tam toplayıcı (Full Adder-FA) devrelerinin birbirine paralel bağlanmasıyla elde edilir. her bir tam toplayıcı için iki bit girişlerden gelen A ve B sayılarıdır. 3. Bit ise önceki tam toplayıcının elde çıkışıdır.



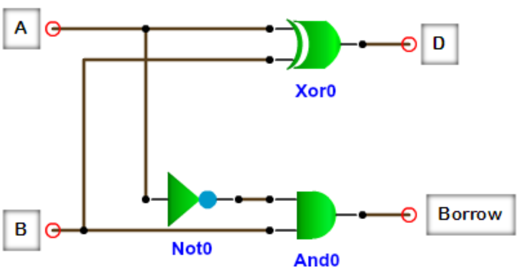
Yarım Çıkarıcı Devre

İki bitlik sayıyı çıkaran devredir. Fark (difference-D) ve borç (borrow-B) çıkışı verir.

A	B	Fark D	Borç B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

D=A’B+AB’

B=A’B



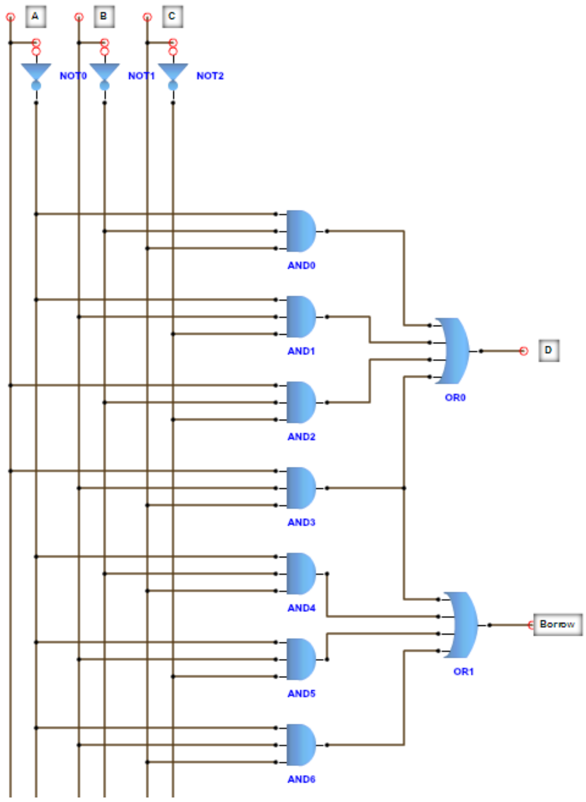
Tam çıkarıcı devre

Üç bitlik sayıyı çıkaran devredir. Düşük değerli basamak tarafından 1 borç alındığını varsayarak işlem yapar.

A	B	C	Fark D	Borç Borrow-B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

D=A’B’C+A’BC’+AB’C’+ABC

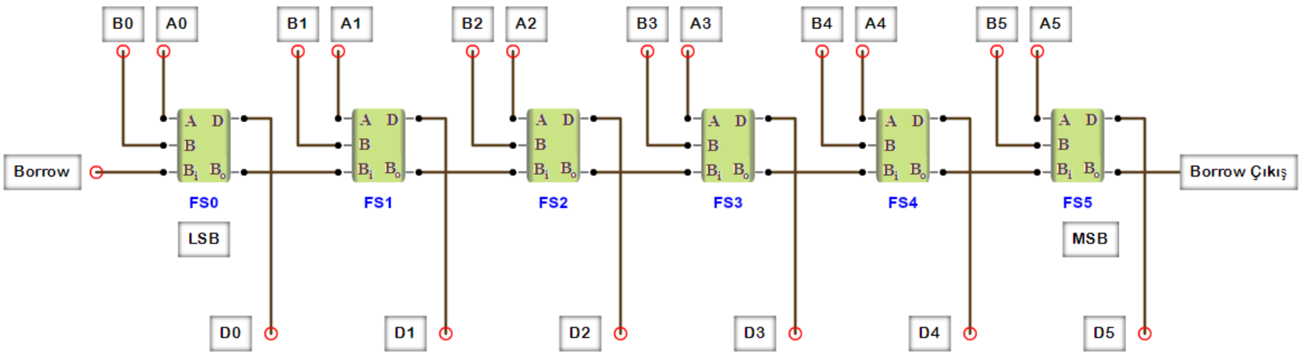
B=A’B’C+A’BC’+A’BC+ABC



Tam çıkarıcı devre, tam toplayıcı devrenin iki yarım toplayıcıdan gerçekleştirildiği gibi iki yarım çıkarıcıdan gerçekleştire bilinir.

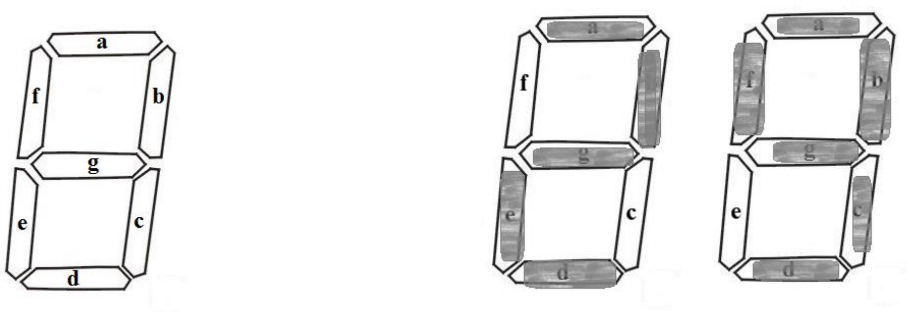
Paralel çıkarıcı devre

Yüksek bitli sayıları çıkarmak için kullanılan devredir. Tam çıkarıcı devrelerin (Full Subtractor-FS) birbirine paralel bağlanmasıyla elde edilir. Devrenin sonundaki borç çıkışı "1" ise sonuç negatif, "0" ise sonuç pozitiftir.



6.6. *KOD ÇEVİRİCİ DEVRELERİ*

Sayısal elektronikte, bilgisayarlarda birden fazla kod kullanıldığı önceki bölümlerde anlatılmıştı. Bu kodlar arası dönüştürme yapan devrelere kod çevirici devreler denir. Örneğin BCD den +3 koduna, 8421 kodundan gray koduna gibi. En çok kullanılan kod çeviriciden bir tanesi de binaryden 7 segmentli displaye kod çeviren devredir. 7 segment göstergede 7 ayrı led bulunur. Bu ledler a ile g arasında isim alırlar. 0'dan 9'a kadar olan sayılar ve harfler bu segmentle yazılabilir.



Örneğin, ortak katotlu bir göstergene LED'lerin katotları (-) uca bağlıdır. Yanmaları için anot ucuna (+) yani "1" uygulamak gerekir bu durumda 2 rakamı için a,b,g,e,d LED'leri, 9 rakamı için a,b,c,d,g,f LED'leri yanacaktır. Ortak anot 7 segment göstergelerde ise yanan değil sönen LED'lere göre işlem yapılmaktadır. Yani girişleri "0" yapılmalıdır. Bazı 7 segmentle göstergeler ise girişi olmadığı durumlarda tüm LED'leri yakar. Bu durumda yazılmak istenen sayı-harf için gerekli LED'ler söndürülmelidir. Örneğin 2 için c,f, 9 için e LED'i söndürülmelidir. Aksi belirtilmediği takdirde, sayısal elektronikte 7 parçalı göstergeden bahsedilirken, çıkış verilmediği durumda tüm LED'leri sönük olan çeşit display kullanılmaktadır.

Decimal	**Binary**				**Gösterge**						
	A	**B**	**C**	**D**	**a**	**b**	**c**	**d**	**e**	**f**	**g**
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	X	X	X	X	X	X	X
.	.				.						
.	.				.						
.	.										
15	1	1	1	1	X	X	X	X	X	X	X

a=A'B'C'D'+A'B'CD'+A'B'CD+A'BC'D+A'BCD+AB'C'D'+AB'C'D

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

Kırmızı grup : CD
Mavi grup : A
Yeşil grup : BD
Turuncu grup : $\overline{BD}$
a=A+BD+B'D'+CD

ÇALIŞMA SORULARI (BİLEŞİK MANTIK DEVRELERİ)

1) Z=ABC+A’BC+A’B’C’+AB’ fonsiyonunu

a) 4x1 MUX seçme bitleri A ve B ile,
b) 2x1 MUX seçme biti A ile gerçekleyiniz.

2) Hexadecimal sayı sisteminden binary sayı sistemine kodlayıcı (encoder) devresini tasarlayınız.

3) 1’den 10’a bilgi dağıtıcı (DEMUX) devresini 4 seçme biti (A,B,C,D) ile tasarlayınız.

4) iki bitlik sayının karesini alan aritmetik işlem devresini tasarlayınız.

5) Decimalden 7 bölmeli göstergeye kod çeviren devre için

a) b,c,d,e,f,g LED’lerinin lojik ifalerini Karnough haritaları yardımıyla bulunuz.

b) lojik kapılar kullanarak tasarımını gerçekleştiriniz.

7. MULTİVİBRATÖRLER

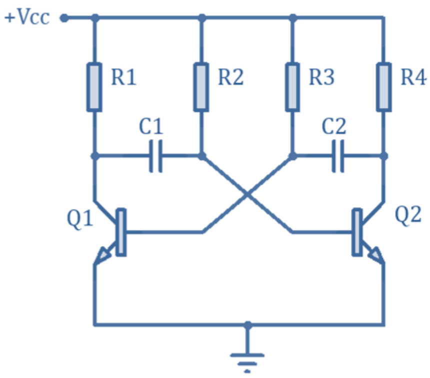
Dijital devrelerin çalışmasında 1 ve 0'ların temel olduğundan bahsetmiştik. Lojik 1'ler ve 0'lar uygulamada yüksek gerilim-düşük gerilim, gerilim var-yok gibi algılanırlar. Çoğunda lojik 1 +5v gibi bir durumken, lojik 0 ise 0v gibi bir durumu temsil eder. Buradan da anlaşıla bilineceği gibi lojik devreler kare dalga (veya dikdörtgen dalga) ile çalışırlar. Yani gerilim ya 0 olmalı ya 5 v olmalı, sinüzoidal dalga gibi 0 ila 5 volt arasında dalgalanmamalıdır.

Dijital devrelerin çalışması için kare veya dikdörtgen dalga üreten devrelere multivibratör adı verilir. Genellikle transistör yardımıyla veya entegre yardımıyla gerçekleştirilir.

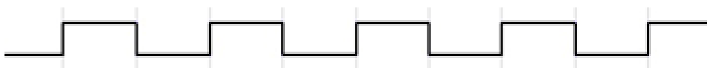
- Serbest çalışan (astable) multivibratör
- Tek kararlı (monostable) multivibratör
- Çift kararlı (bistable) multivibratör

Olmak üzere 3 çeşit multivibratör vardır.

7.1. *SERBEST ÇALIŞAN (ASTABLE) MULTİVİBRATÖR*



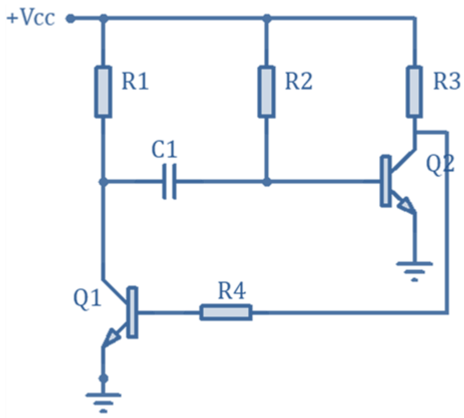
Astable multivibratörler belirli aralıkla sürekli durum değiştiren multivibratörlerdir. Başlangıçta bir transistör (Q1)iletimde birinin yalıtımda (Q2) olduğunu varsayarsak iletimde olan transistör (Q1), diğer transistörün beyzine bağlı kapasitörü (C1) şarj eder. Bu kapasitör (C1) şarj olduğunda beyzine gerilim alan transistör iletime geçer (Q2). Bu anda diğer transistöre (Q1) bağlı olan kapasitör (C2) transistör üzerinden (Q2) deşarj olur. Böylece Q1 yalıtıma geçer. Bu işlem sürekli tekrarlanır.



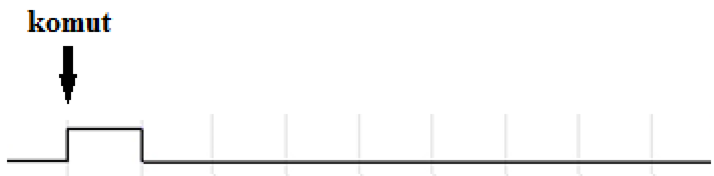
ŞEKİL 17: ELEMANLARIN BİRBİRİNE EŞİT OLDUĞU DURUMDA ASTABLE MUL. ÇIKIŞI

7.2. TEK KARARLI (MONOSTABLE) MULTİVİBRATÖR

Devreye dışarıdan müdahale edilmediği (komut) müddetçe Q1 yalıtımda,Q2 iletimdedir. Dışarıdan müdahale ile Q1 iletime geçirilirse, kapasitör deşarj olana kadar Q2 yatılımda kalır. C1 şarj olduğunda ise Q2 iletime geçer. Tekrar Q1'e müdahale edilene kadar devre bu durumunu korur.

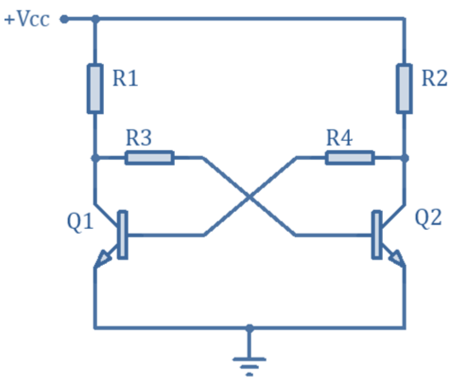


ŞEKİL 18:MONOSTABLE MULTİVİBRATÖR



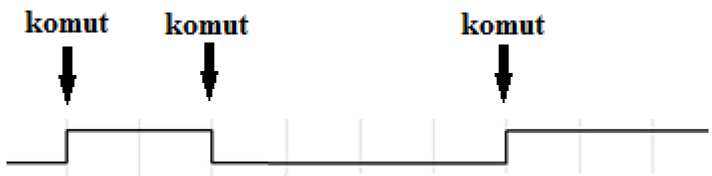
ŞEKİL 19: MONOSTABLE MULTİVİBRATÖR ÇIKIŞI

7.3. ÇİFT KARARLI (BİSTABLE) MULTİVİBRATÖR



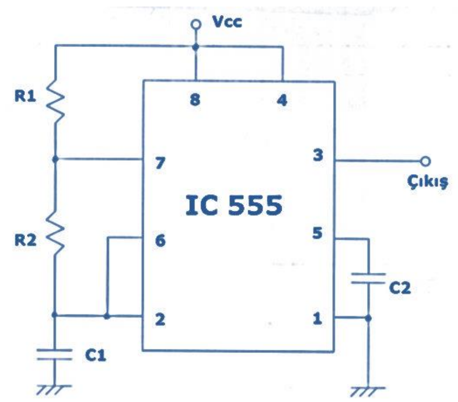
ŞEKİL 20:BİSTABLE MULTİVİBRATÖR

Bu devre temel hafıza biriminin temelini oluşturur. Dışarıdan müdahale olmadığı sürece transistörler durumunu korur. Biri iletimde, diğeri yalıtımdadır. Bu devrede 2 transistör hiçbir zaman aynı durumda olmaz. Dışarıdan bir müdahale (komut) ile transistörler durum değiştirir. Yatılımda olana iletime, iletimde olan yalıtıma geçer. Ve tekrar müdahale edilene kadar durumlarını korurlar.



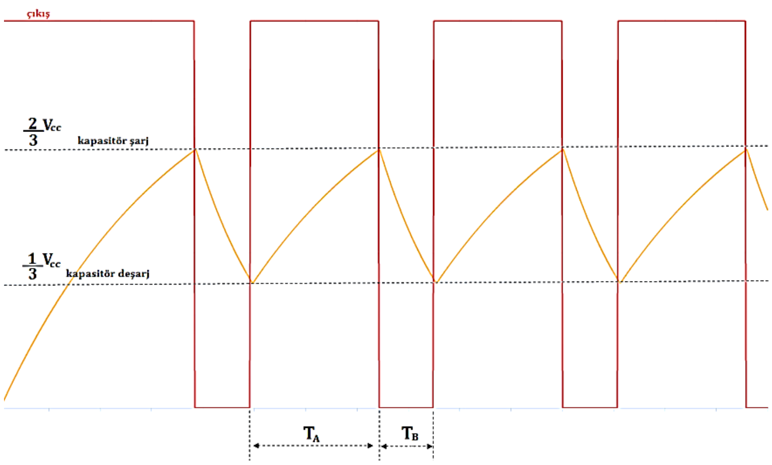
ŞEKİL 21:BİSTABLE MULTİVİBRATÖR ÇIKIŞI

7.4. IC 555 ENTEGRESİ İLE YAPILAN ASTABLE VE MONOSTABLE MULTİVİBRATÖR DEVRELERİ



ŞEKİL 22: IC555 ENTEGRESİ İLE ASTABLE MULTİVİBRATÖR

IC 555 ile yapılan astable multivibratör devresinin çalışması C_1 kapasitörü ile bağlantılıdır. C_1 kapasitörü R_1 ve R_2 dirençleri üzerinden V_{cc} geriliminin 2/3'üne kadar şarj olur ($E_{C1}=\frac{2}{3}V_{CC}$). Bu durumda entegrenin çıkışından sinyal alınır. C_1 kapasitörü V_{cc} geriliminin 1/3'üne kadar deşarj olur. Bu durumda entegrenin çıkışından alınan sinyal düşer.



ŞEKİL 23: IC555 İLE YAPILAN ASTABLE MULTİVİBRATÖR ÇIKIŞI

IC555 entegresi ile yapılan astable multivibratör devresinin çıkış periyodu aşağıdaki formüllerle bulunabilir.

$$T_A = 0{,}7(R_1+R_2)C_1$$

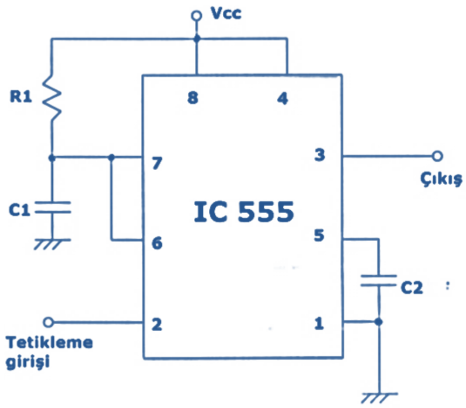
$$T_B = 0{,}7R_2C_1$$

$$T=T_A+T_B$$

$$T=0{,}7(R_1+2R_2)C_1$$

$$f=1/T$$

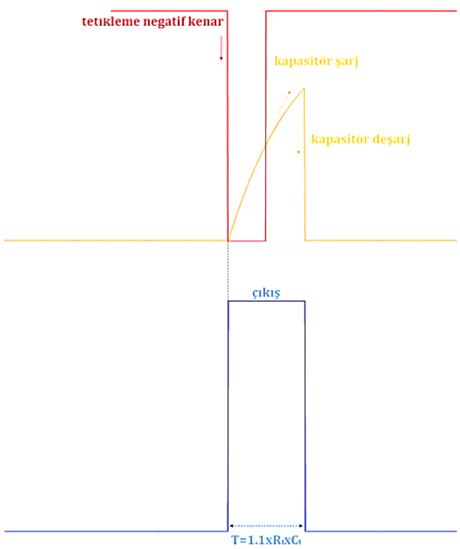
$$f=\frac{1{,}44}{(R_1+2R_2)C_1}$$



IC 555 entegresiyle yapılan monostable multivibratör devresinde tetikleme girişinden gelen negatif kenarla birlikte C_1 şarj olur, çıkıştan T süresince çıkış alınır. Bu süreçte C_1 kapasitörü deşarj olur ve çıkış gerilimi sıfıra düşer. Tekrar tetikleme alınıp C_1 kapasitörü şarj olana kadar çıkış sıfır volta sabitlenir.

$T=1{,}1xR_1xC_1$

ŞEKİL 24:IC555 İLE YAPILAN BİSTABLE MULTİVİBRATÖR



ŞEKİL 25:ŞEKİL 24:IC555 İLE YAPILAN BİSTABLE MULTİVİBRATÖR ÇIKIŞI

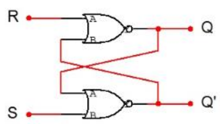
8. FLİP-FLOPLAR (FF)

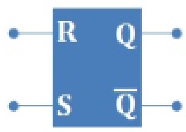
Devre çalıştığı sürece çıkış durumunu koruyan yapılara Flip-Flop (FF)denir. Flip-Floplar temel hafıza birimini oluştururlar. 1 bitlik bilgiyi saklayabilirler. Flip-Flop'larda bu bilgiyi ve bu bilginin tümleyenini gösteren (kendisi ve değili) iki çıkışı bulunur (Q ve Q'). Bir Flip-Flop'nin çıkışı dendiğinde bu çıkışın kendisidir (Q). Çıkışın değişmesi için girişin değişmesi ve tetiklemenin değişmesi gerekmektedir. Tetikleme değişmediği sürece çıkış konumunu korur.

Flip-Flop'larda tetikleme clock palse (CP) denilen kare dalga sinyal ile yapılır. Devrenin niteliğine göre kare dalganın yükselen kenarıyla ya da düşen kenarıyla tetikleme yapılabilir.nadir olarak clock palse'in pozitif düşey veya negatif düşey durumuyla da tetikleme yapılabilmektedir. Bu durum devrede özel olarak belirtilir.

8.1. SR FLİP-FLOP

R (reset-sıfırlama) ve S (set-kurma, ayarlama, başlatma) adında iki girişi olan bu devrede temel olarak set konumunda çıkış bir, reset konumunda ise çıkış sıfır olur. Her iki giriş birden "0" olursa çıkış değişmez. Her iki konumun aktif olması durumunu (hem set yap hem reset yap konumu) ise yasak konumdur. Yasak konumda çıkışın ne olacağı bilinememektedir. Bu durum doğruluk tablosundan görülebilir.

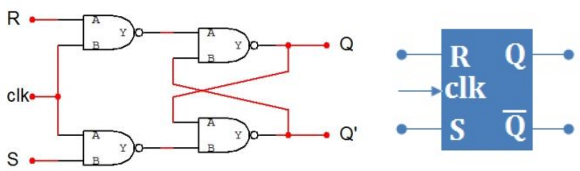




S	R	Q	Q'
0	0	Önceki konum	
0	1	0	1
1	0	1	0
1	1	Yasak konum	

ŞEKİL 26: SR FF

Tetiklemeli RS Flip-Flop'da ise RS Flip-Flop'a bir saat eklenir. Sürekli karedalga sağlayan clock palse (saat) her iki girişede VEDEĞİL kapılarıyla eklenmiştir. Tetiklemeli Flip-Flop'larda bir çıkış durumu belirlenirken kendinden önceki durum da değerlendirmeye alınır. Devreden çıkış alınabilmesi için clock palse sinyalinin sıfır olmaması gerekmektedir. Clock palse 1 iken durum değişir (Flip-Flop'lar duruma göre yükselen kenar veya düşen kenar tetiklemelidir), 0 iken durum önceki durumu korur.

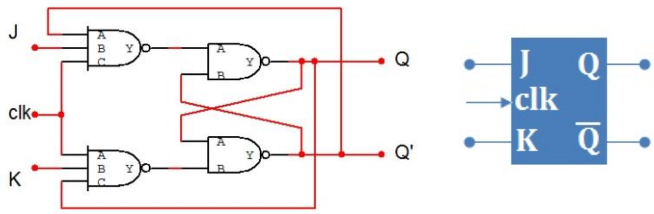


ŞEKİL 27: TETİKLEMELİ RS FF

Q	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Yasak
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Yasak

8.2. JK FLİP-FLOP

SR Flip-Flop'lardaki belirsiz durumu kaldırmak için tasarlanmıştır. J girişi SET, K girişi RESET gibi davranır. Yasak konum J=K=1 durumunda çıkış her tetiklemeyle bir önceki çıkışın tümleyeni (tersi) olur. Bu duruma toggle (değiştir) denir.



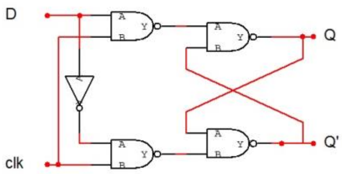
ŞEKİL 28:JK FF

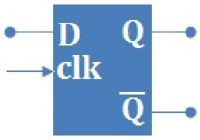
J	K	clk	Q
0	0	1	Önceki konum
0	1	1	0
1	0	1	1
1	1	1	toggle

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

8.3. D (DATA) FLİP-FLOP

SR Flip-Flop'un girişlerinin senkronize olmuş Flip-Flop'tur. Girişin biri diğerinin tümleyeni olarak tek girişe yönlendirilmiştir. Data Flip-Flop D girişindeki bilgiyi her tetiklemeyle birlikte Q çıkışına aktarır.





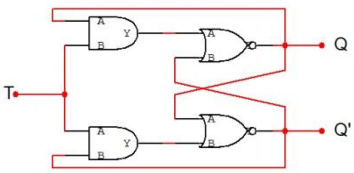
D	clk	Q
0	1	0
1	1	1

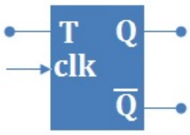
Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

ŞEKİL 29: DATA FF

8.4. T (TOGGLE) FLİP-FLOP

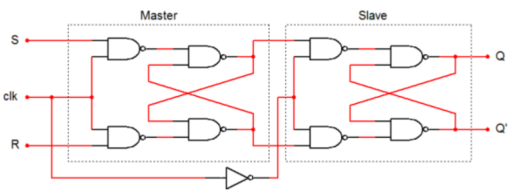
JK Flip-Flop'un girişlerinin birleştirilmesiyle Toggle Flip-Flop oluşturulur. Bu Flip-Flop girişi 1 olduğu sürece tetikleme sinyali ile birlikte çıkışı değiştirir. Giriş 0 ise, çıkış clock palse uygulansa bile aynı olarak kalır.

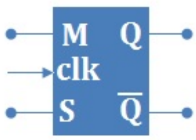




ŞEKİL 30: TOGGLE FF

8.5. MASTER-SLAVE FLİP-FLOP





9. SAYICILAR

Her clock palse ile önceden istenilen durumları sırasıyla çıkışlarına aktqran devrelere sayıcılar denir. Temel elemanı Flip-Flop'tur. Sayıcılar belirlenen Flip-Flop sayısına göre istenilen bitlerde gerçekleştirilebilir.

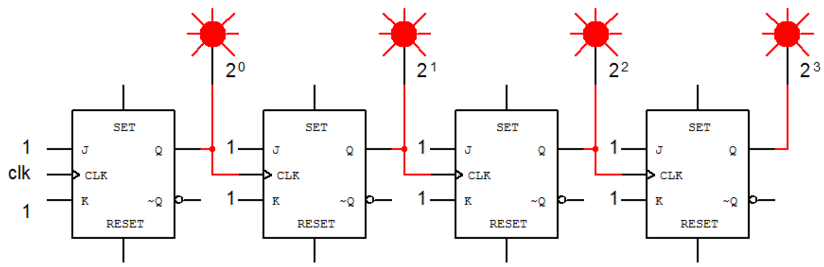
9.1. ASENKRON SAYICILAR

Asenkron sayıcılarda Flip-Floplar arası bir eş zamanlılık yoktur. Her Flip-Flop ayrıca clock palse alarak tetiklenirler. Bit sayısı artıkça devrenin çalışma süresi de artar. Örneğin bir Flip-Flop yaklaşık olarak 20 ns'de işlem yapar. 4 bitlik bir sayıcı devresinde 4 tane Flip-Flop bulunmaktadır. 1. Flip-Flop'un çıkışı 2. Flip-Flop'un clock'una, 2. Flip-Flop'un çıkışı 3. Flip-Flop'un clock'una ve 3. Flip-Flop'un çıkışı 4. Flip-Flop'un clock'una aktarılır. Bu durumda 80 ns'lik (20nsx4) bir sürede 4 bitlik bir sayı sayılır. Özetlersek 80 ns'den önce ilk Flip-Flop'a clock palse uygulamamak gerekmektedir. Bu durumda clock palse'in frekansı: $T=80ns=80.10^{-9} \rightarrow F= 1/80.10^{-9}=12,5.10^{6}$ Hz= 12,5 MHz' olmalıdır.

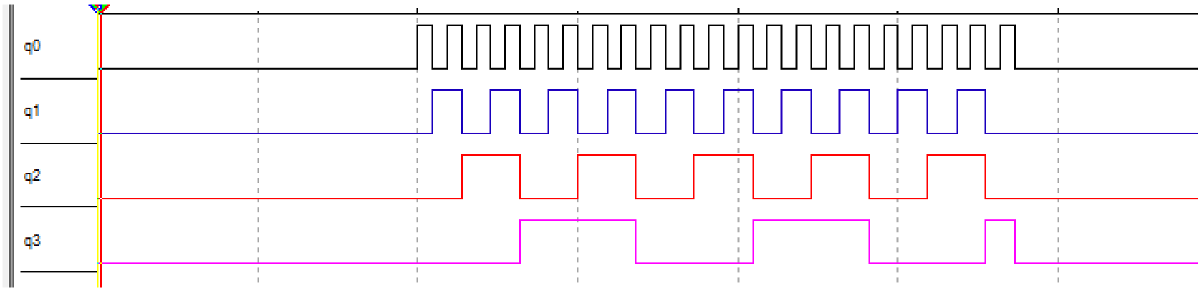
9.1.1. YUKARI ASENKRON SAYICI

Yukarı asenkron sayıcı çıkışları birer birer artırır. Çıkış en yüksek değere ulaştıktan sonra tekrar başlangıca döner. Aşağıda 0'dan 15'e kadar sayan sayıcı devresi bulunmaktadır.

Yukarı sayıcılarda JK Flip-Flop kullanılır. Flip-Flop girişleri 1-1 olduğu için, çıkış daima toggle olur (değişir). Eğer düşen kenar tetiklemeli Flip-Flop kullanılacaksa, ilk Flip-Flop hariç diğer Flip-Flop clock palse'leri bir önceki Flip-Flop'un Q çıkışından alınır. Eğer Flip-Flop yükselen kenar tetiklemeli ise tetikleme girişlerin Q' çıkışından alınır.



ŞEKİL 31: JK FF İLE YAPILAN YUKARI ASENKRON SAYICI DEVRESİ

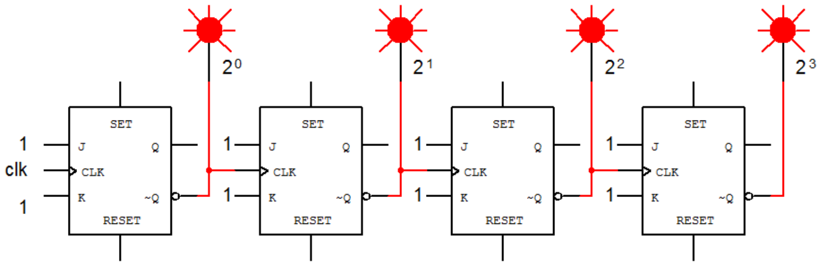


ŞEKİL 32: ASENKRON YUKARI SAYICI DALGA ŞEKİLLERİ

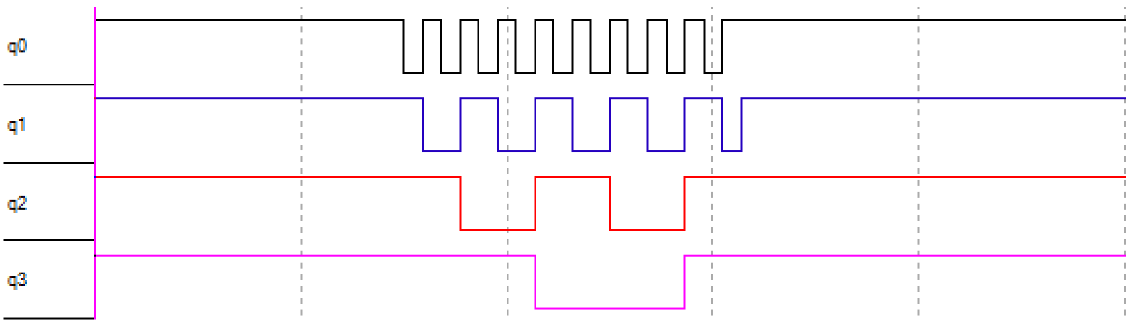
Bu devrede düşen kenar tetiklemeli Flip-Flop'lar kullanılmıştır. Dalga şekillerini incelersek 1. Flip-Flop'un çıkışı 2. Flip-Flop'un tetikleme sinytali olarak davranır. 1. Flip-Flop'un çıkışı düşen kenara ulaştığında 2. Flip-Flop tetiklenir. Bu diğer Flip-Flop'lar arasında da böyle gerçekleşir.

9.1.2. AŞAĞI ASENKRON SAYICILAR

Asenkron aşağı sayıcı ile maksimum değerden birer birer düşerek minimum değere kadar ulaşılır. Bunun için düşen kenar tetiklemeli Flip-Flop kullanılacaksa, ilk Flip-Flop hariç diğer Flip-Flop clock palse'leri bir önceki Flip-Flop'un Q' çıkışından alınır. Eğer Flip-Flop yükselen kenar tetiklemeli ise tetikleme girişlerin Q çıkışından alınır.



ŞEKİL 33: JK FF İLE YAPILAN ASENKRON AŞAĞI SAYICI DEVRESİ



ŞEKİL 34: ASENKRON AŞAĞI SAYICI DEVRESİ ÇIKIŞ DALGA FORMU

Asenkron sayıcılar aynı zamanda frekans bölücü devrelerdir. q0 tetikleme sinyalinin, q1 ise q0'ın, q2 ise q1'in, q3 ise q2'in frekanslarını ikiye böler. Örneğin clock palse 1 kHz olan bir 4 bitlik sayıcı devresinde

1. Flip-Flop çıkışı ½=500 Hz,

2. Flip-Flop çıkışı 500/2= 250 Hz

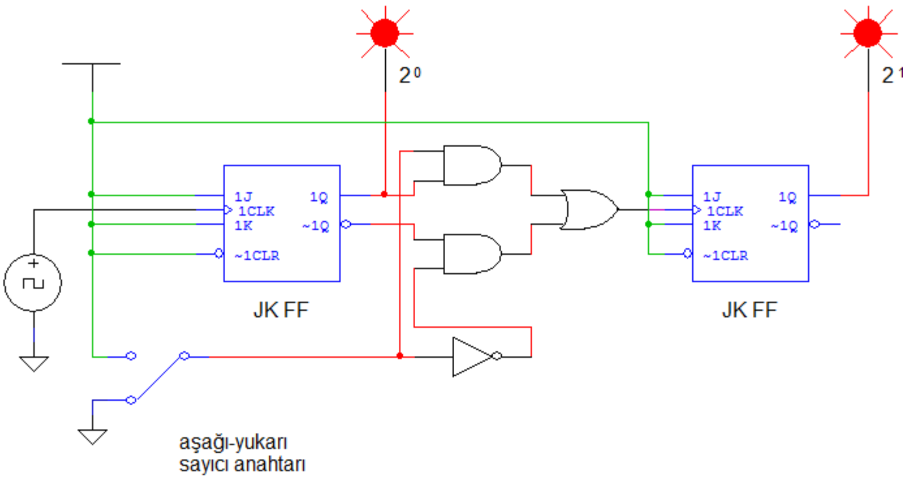
3.Flip-Flop çıkışı 250/2=125 Hz

4.Flip-Flop çıkışı 125/2=62,5 Hz 'olur.

Kısaca n bitlik bir sayıcı için çıkış frekansı : $F_{out}=\frac{F_{clk}}{2^n}$

9.1.3. AŞAĞI VE YUKARI ASENKRON SAYICILAR

Aşağı ve yukarı asenkron sayıcılarda 2. Flip-Flop'un tetikleyici girişi olarak 1. Flip-Flop'un Q veya Q' çıkışı seçilir. Devreden de görülebileceği gibi anahtar 1 konumundayken 1. Flip-Flop'un Q çıkışı, anahtar 0 konumundayken Q' çıkışı aktif olarak diğer Flip-Flop'u tetikler.

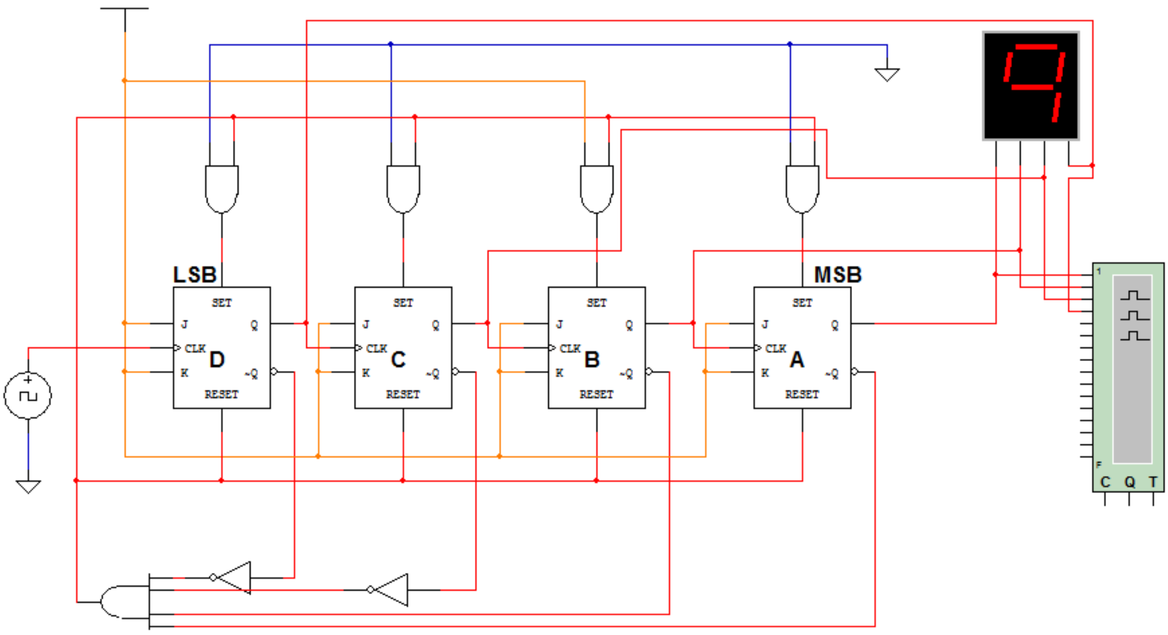


ŞEKİL 35: ASENKRON AŞAĞI-YUKARI SAYICI DEVRESİ

9.1.4. RESETLEMELİ-PRESETLEMELİ SAYICILAR

Flip-Flop'larda reset girişi aktif yapıldığında Flip-Flop'un çıkışı 0 olur. Yani Flip-Flop resetlernir. Set girişi aktif yapıldığında ise Flip-Flop'un çıkışı 1 olur. Bu duruma preset (pre-set) ön kurulma/ön hazırlık denir.

Şimdiye kadar gördüğümün sayıcılarda başlama ve bitiş bilgileri o devrenin alabileceği minimum ve maksimum değerlerdi. Devrenin sayma sayıları reset ve set girişleri ile tasarlanabilir. 4 bitlik bir sayıcıda 3-12 arası ya da 5-10 arası ileri sayma veya 11-4 geri sayma yaptırılabilir.



ŞEKİL 36: PRESETLEMELİ VE RESETLEMELİ ASENKRON SAYICI DEVRESİ

Yukarıdaki presetlemeli resetlemeli sayıcı örneğinde 12'den 4'de geri sayıcı tasarımı yapılmıştır. ABCD Flip-Flop'ların girişleri 0100 olarak set edilmiştir. Çıkış 1100 olduğunda reset girişleri aktif yapılmıştır. Reset girişiyle beraber set girişi de aktif yapılmıştır böylece 12-4 sayıcı elde edilmiştir.

9.2. SENKRON SAYICILAR

Senkron yani eş zamanlı sayıcılarda tetikleme sinyali tüm Flip-Flop'lara aynı clock palse'den sağlanır. Böylelikle Flip-Flop'lar arası eş zamanlılık ve zamandan tasarruf elde edilir. ancak senkron sayıcı tasarlamak için asenkrondan daha fazla kapı kullanmak gerekebilir. Her sayım için senkron sayıcıyı ayrıca tasarlamak gerekir.

Örneğin JK Flip-Flop kullanılarak tasarlanacak olan 0-3-5-7 ve tekrar sayan bir sayıcı için (ring-halka sayıcı) tasarım yapmak için öncelikle durum geçiş tabloları oluşturulmalıdır.

Durum geçişleri Karnough haritaları kullanarak sadeleştirilmelidir. 3 bitlik bir sayıcı için 3 adet Flip-Flop kullanılmalıdır. Her Flip-Flop için bir J girişi ve bir K girişi vardır.

ABC (şimdiki durum)	**ABC** (sonraki durum)
000	011
011	101
101	111
111	000

Önceki durum	Sonraki durum	**JK**
0	0	0X
0	1	1X
1	0	X1
1	1	X0

A bitinin J girişi için 000 durumundan 011 durumuma geçilmiş. Öncelikle 000'ın yeri karnough haritasında işaretlenir. Bu durumda A biti 0 durumundan 0 durumuna geçmiştir. JK Flip-Flop'un girişleri 0-0 durumu için 0X'dir. Yani J girişi 0 olmalıdır. Haritaya 000 yerine 0 yazılmalıdır. Sonra 011'den 101'e geçişte A biti 0'dan 1'e geçmiştir. JK Flip-Flop 0-1 geçişte girişleri X1'dir. Yani J girişi X olmalıdır. Böylelikle tüm tablo J ve K girişleri için doldurulur. Geri kalan noktalara önemsiz giriş X denir. Sadeleştirme yapılır.

JA	00	01	11	10
0	0			
1		X		

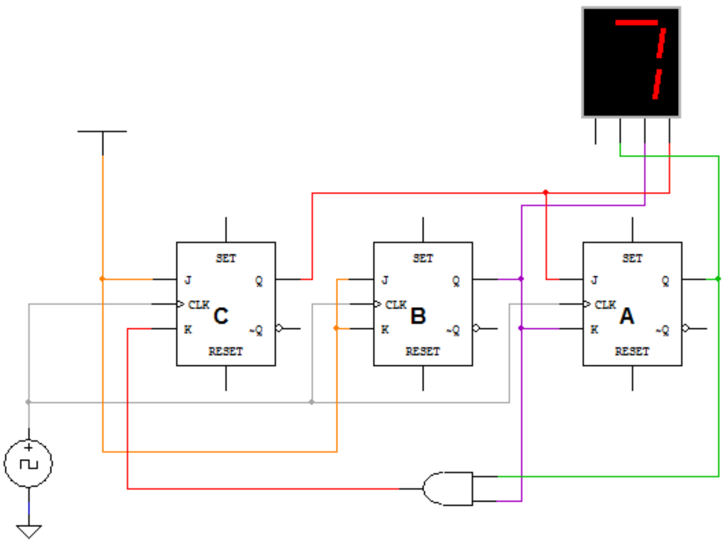
0-3-5-7 sayıcı için gerekli sadeleştirmeler yapılırsa:

JA=B
KA=B

JB=1
KB=1

JC=1
KC=AB olur.

Final tasarım şu şekilde gerçekleştirilir:



ŞEKİL 37: SENKRON (RİNG) SAYICI DEVRESİ

Senkron sayıcılar diğer Flip-Flop'lar kullanılarak da gerçekleştirebilir. Durum geçiş tabloları o Flip-Flop'un geçişlerine göre düzenlenir.

10. KAYDEDİCİLER

Flip-Flop'lar temel bilgi saklama elemanlarıdır. Her bir Flip-Flop bir bitlik bilgiyi saklayabilirler. 2 bit ve daha fazla bilgi saklayan elemanlara kaydediciler denir. Her Flip-Flop bir bit saklayabilir dolayısıyla bir kaydedici barındırdığı Flip-Flop sayısı kadar bit kaydedebilir.

Kaymalı kaydediciler (shift register) şu gruplara ayrılır:

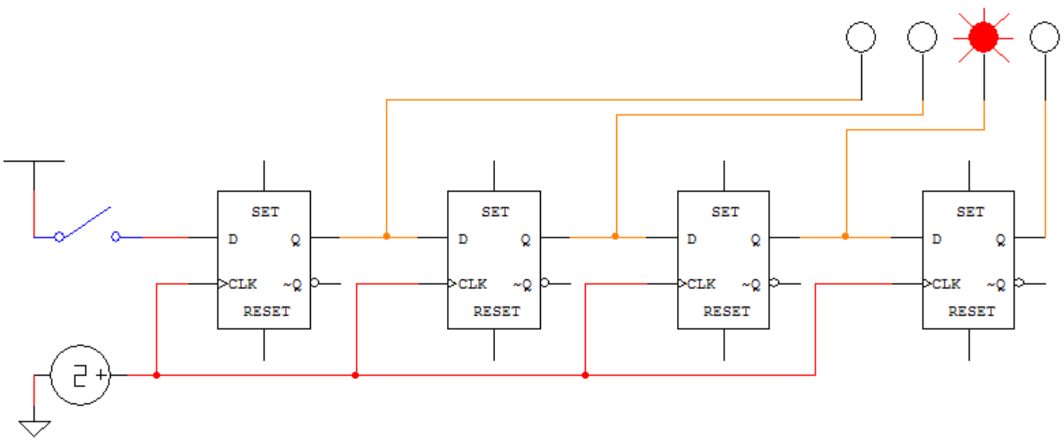
1. Kayma işlemlerine göre kaydediciler:
    a. Sağa kaymalı
    b. Sola kaymalı
    c. Sağa ve sola kaymalı
2. Bilgi giriş çıkışlarına göre kaydediciler:
    a. Seri giriş-çıkış
    b. Seri giriş- paralel çıkış
    c. Paralel giriş-çıkış
    d. Paralel giriş- seri çıkış

10.1 KAYDIRMALI KAYDEDİCİLER

A. Sağa kaydırmalı kaydediciler

Genellikle kaydırmalı kaydediciler için D tipi Flip-Flop kullanılır. Eğer JK veya RS Flip-Flop kullanılacaksa girişler arasına bir NOT kapısı eklenmeli ve aynı giriş hem J-K hem de R-S için uygulanmalıdır.

Örneğin 4 bitlik D Flip-Flop kullanılarak yapılan seri giriş-paralel çıkış kaydedici devresi, seri girişten gelen biti (0100 bilgisi) paralel çıkışlarına aktarır. Her clock palse ile (kaydırma palsi-shift pulse) bilgi bir sağa kayar.



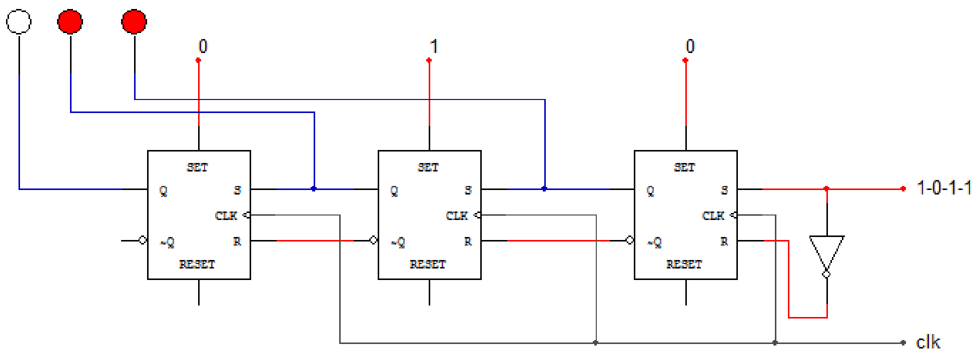
ŞEKİL 38: SAĞA KAYMALI KAYDEDİCİ

0100 bilgi girişi ile 3 kaydırma palsi için doğruluk tablosu:

Shift pulse	Seri Data	Flip-Flop3	Flip-Flop2	Flip-Flop1	
0	-	0	1	0	← Paralel data
1	1	1	0	1	
2	0	0	1	0	
3	1	1	0	1	
4	1	0	1	1	

B. Sola Kaydırmalı Kaydediciler

Bu tip kaydediciler bilgiyi sola kaydırarak muhafaza ederler. Örneğin RS Flip-Flop ile yapılan, paralel giriş 010 bilgisi, seri girişten ise 1011 bilgilerini girip, 3 bitlik kaydedici için 4. Clock palse'de ki durumu inceleyelim.



ŞEKİL 39:SOLA KAYMALI KAYDEDİCİ

Shift pulse	Data	Flip-Flop1	Flip-Flop2	Flip-Flop3	Flip-Flop4
-	-				
0	0	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0

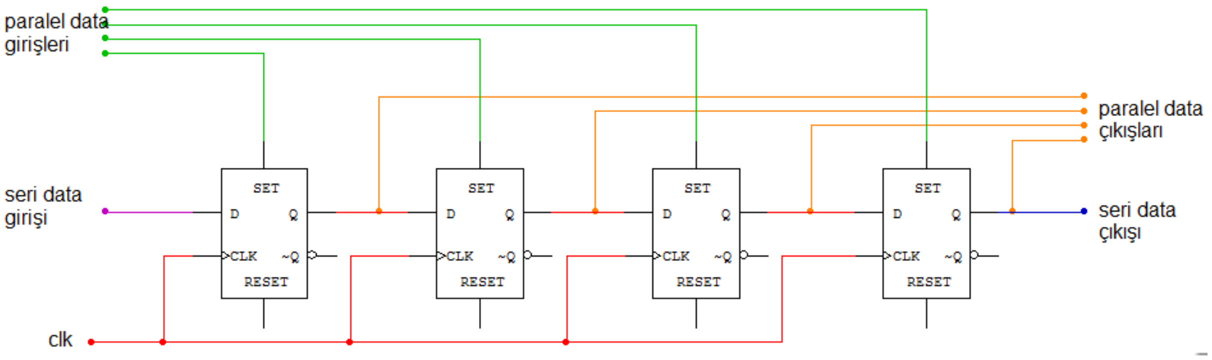
C. Sağa-Sola Kaymalı Kaydediciler

İki yönde bilgi kaydedebilen bu devreler temel aritmetik işlemi gerçekleştirirler. Sola doğru kaymada çarpma, sağa doğru kaymada bölme işlemi yaparlar. Sağa-sola doğru kaymayı belirleyen anahtarı ile kontrol edilirler. Bu işlemi yapan belli entegreler mevcuttur. Ör/74179

10.2 KAYDEDİCİLERİN TRANSFER YÖNTEMLERİ

Daha önce bahsedildiği gibi bir kaydedici 4 şekilde bilgi transferi yapabilir.

a. Seri giriş-çıkış
b. Seri giriş- paralel çıkış
c. Paralel giriş-çıkış
d. Paralel giriş- seri çıkış



ŞEKİL 40: KAYMALI KAYDEDİCİLERDE GİRİŞ-ÇIKIŞ KOMBİNASYONLARI

Shift pulse	**Seri Data Girişi**	Flip-Flop1	Flip-Flop2	Flip-Flop3	Flip-Flop4	
0	0	0	1	0	0	**← paralel data girişi**
1	1	1	0	1	0	
2	0	0	1	0	1	
3	1	1	0	1	0	
4	0	0	1	0	1	**←paralel data çıkışı**

← Seri data çıkışı

Bir kaydedici devresi sadece seri giriş veya sadece paralel giriş olabileceği gibi hem seri hem paralel girişe sahip olabilir. Kaydedici çıkışları seri olabilir veya paralel olabilir. Paralel çıkışta son bit aynı zamanda seri çıkışı verir. Yani paralel çıkışa sahip bir kaydedici aynı zamanda seri çıkışa da sahiptir.

KAYNAKÇA

ADC-DAC DEVRELERİ. 07 04, 2014 tarihinde http://megep.meb.gov.tr/mte_program_modul/moduller_pdf/Adc-dac%20Devreleri.pdf adresinden alındı

Başkent Üniversitesi. 07 04, 2014 tarihinde http://www.baskent.edu.tr/~aerdamar/BME%20423_chp01.pdf adresinden alındı

Binary Converter. 07 01, 2014 tarihinde http://www.exploringbinary.com/binary-converter/ adresinden alındı

ÇÖLKESEN, T. A. (2001). Lojik Devre Tasarım. Papatya Yayıncılık.

DoCircuits. 06 30, 2014 tarihinde DoCircuits: http://www.docircuits.com/circuit-editor adresinden alındı

EKİZ, H. (2010). *Mantık Devreleri.* İstanbul: Değişim Yayınları.

R.L., T. (1994). Schaum's Outline of Theory and Problems of Digital Principles. USA: McGraw-Hill.

Sayıcılar. Mersin Üniversitesi Uzaktan Eğitim Merkezi.

YAĞIMLI, M., & AKAR, F. (2009). *Dijital Elektronik.* İstanbul: Beta Basım Dağıtım.