MANTIK DEVRELERİ

ARA SINAV RAPORU

Bilgisayar Mühendisliği

Karamanoğlu Mehmetbey Üniversitesi

By

TUĞRAN DEMİREL

MAYIS 2020

İÇİNDEKİLER

GİRİŞ

Bu rapor herhangi bir kitap amacı gütmemektedir. Mantık Devreleri dersinin belirli konularının, önemli görülen bilgilerinin belirli süzgeçlerden geçirilip bir araya getirilmesiyle oluşturulmuştur. Bu raporda “Kodlama ve Kodlar” konusu Sayı Sistemleri İncelenmesi konusu altında, “Mantık Fonksiyonlarının Çıkarılması” konusu Boolean Kuralları ve Lojik İfadelerin Sadeleştirilmesi konusu altında, “Çoklayıcı ve Azlayıcı, Kodlayıcı ve Kod Çözücü” konuları Birleşik Devre Tasarımı konusu altında yer verilmiştir.

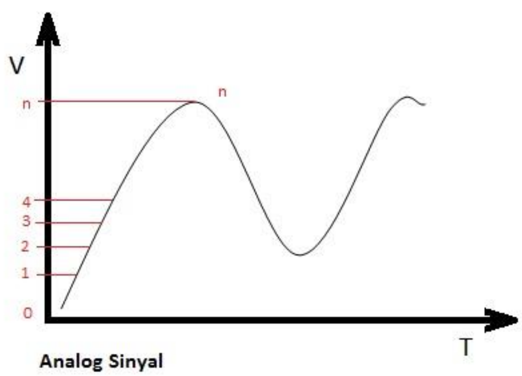
BÖLÜM 1

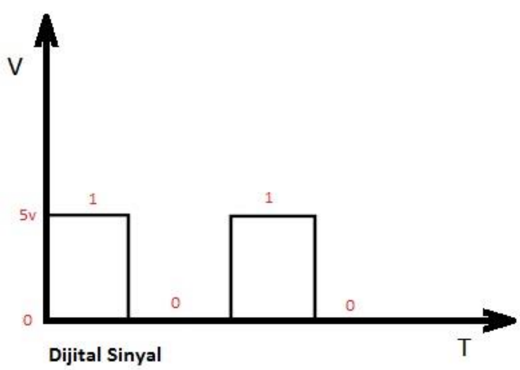
1. ANALOG VE SAYISAL KAVRAMLAR

1.1 Analog - Sayısal İşaretler

Gerçek dünyada karşılaştığımız birçok fiziksel büyüklüğün (akım, gerilim, sıcaklık vb.) değeri sürekli bir aralık içinde kesintisiz değişmektedir. Sınırlar arasındaki her türlü olası değeri alabilirler. Bu tür işaretlere “analog işaretler” denir.

Sayısal işaretler ise belirli bir aralıkta atlamalı değerler alabilen işaretlerdir. En çok bilinen sayısal işaret ikili(binary) olanıdır. İkili işarette yalnızca iki değer(1-0, var-yok gibi) söz konusudur.





1.2 Sayısal Sistemlerin Avantajları

Eskiden analog sistemlerin kullanıldığı birçok alanda(fotoğrafçılık, video-ses kayıtları vb.) günümüzde daha avantajlı olan sayısal sistemler kullanılmaktadır.

<u>*Sayısal sistem avantajlarını şu şekilde sıralayabiliriz:*</u>

1. Sayısal sistemler küçülmekte ve ucuzlamaktadır.
2. Gelişmeye ve yenilenmeye açıktır.
3. Sayısal sistemler daha hızlı çalışmaktadır.
4. Sayısal sistemlerin donanımı değiştirilmeden tekrar tekrar programlanabilir.

5. Sayısal sistemler gürültden etkilenmezler.
6. Sayısal sistemleri test etme ve hatalardan arındırmak daha kolaydır.

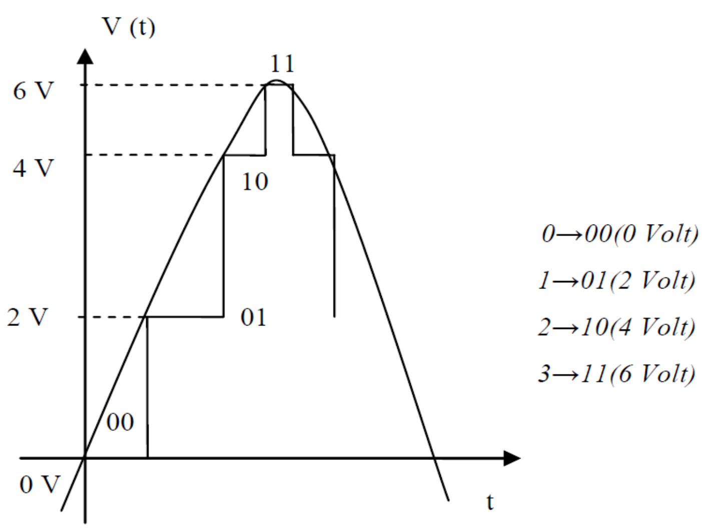
Analog sistemlerin avatajlarını şu şekilde sıralayabiliriz:

1. Sürekli ve kesintisizdir.
2. Analog sinyallerin anlaşılması daha basit olduğundan dolayı işlenmesi de basittir.
3. Analog sinyal gösterebileceği her değeri alır. Bundan dolayı da istenilen değeri tam olarak gösterebilir.

1.3 Analog – Sayısal Dönüşüm

Analog işaretin ikili bir işarete basit bir şekilde dönüştürülebilir. Bir adet kırpıcı devresi ile

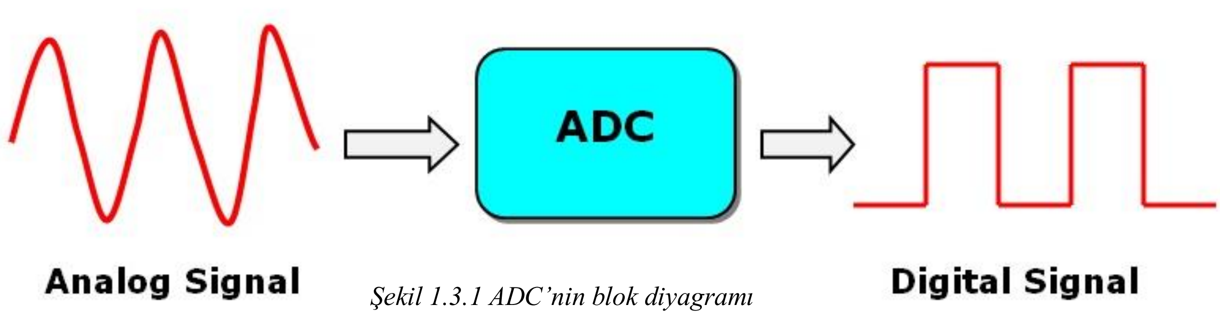
yapılabilir. Analog işareti belirli bir eşiği geçince 1, altında kalınca 0 üretilerek ikiliye dönüşüm işlemi gerçekleşir. Analog işaretten kodlanmış sayısal bir işarete yada sayısal bir işaretten analog işaret elde edilmesi için ADC(Analog to Digital Converter) yada DAC(Digital to Analog Converter) adı verilen elemanların kullanılması gerekir. Bu özel elemanların iki önemli parametresi vardır. Bunlar kodların kaç bir olacağı ve dönüştürme hızıdır. Kodların kaç bit olacağı doğrudan elemanın kaç bitlik veri olacağını yani dönüştürme duyarlılığını belirler. Bu da demektir ki dönüştürücünün bit sayısı artarsa duyarlılığıda artar.



Dönüştürme hızı ise, dönüştürmede kullanılan yönteme göre değişir. Flash dönüştürücüler en hızlı olanlarıdır. Dönüştürme hızı, bir dönüştürme işlemini en kötü durumda ne kadar zaman içersinde yapılacağını belirler.

1.3.1 ADC(Analog to Digital Converter)

Analog sistemden sayısal sisteme dönüştürme işlemi, analog işaretin taşıdığı bilgiyi sayısal bir kod ile ifade etme anlamına gelir.



Şekil 1.3.1 ADC'nin blok diyagramı

Kuantalama:

Sayısal işaret ve görüntü işleme alanlarındaki sürekli bir büyüklüğü eşit sayıda aralıklara bölme işlemine "kuantalama" denilebilir. Kuantalamada duyarlılığın artması için kuantalama kombinasyonundaki eleman sayısının artması da gerekir. Duyarlılık ise çıkış kodunda değişiklik oluşturabilecek en küçük giriş değeri olarak ifade edilebilir. Kuantalama duyarlılığı ise kullanılan ADC türüne göre değişebilmektedir.

Duyarlılığı bulmak için;

$$a = \frac{V_{max} - V_{min}}{2^n}$$

çözüm yolu kullanılabilir. Buradaki a duyarlılığı, V voltu, n ise kodlama yapılacak biti temsil etmektedir. 2^n tane kuanta düzeyi bulunurken $2^n - 1$ ise kuantalama aralığının bulunmasını sağlar.

<u>Örnek 1.1.</u> -5 Volt ile +7 Volt arasında değişen bir analog işaretin 3 Volt duyarlılıkta ifade etmek için kac bit kullanmak gerekir?

a = 3

$V = V_{max} - V_{min} \rightarrow$ 7 – (-5) = 12 V

$3 = \frac{12}{2^n}$ den n → 2 bit çıkar.

Örnek 1.2. Bir analog işaret -10 ile +20 Volt arasında sürekli değerleri alabilmektedir. Bu işaretin 200mV duyarlılıkta bir sayısal işarete dönüştürülmesi için kaç bitlik ADC gerekir?

a = 0.2 V

$V = V_{max} - V_{min} \rightarrow$ 20 – (-10) = 30 V

$0.2 = \frac{30}{2^n}$ den → n ≅ 8 bit

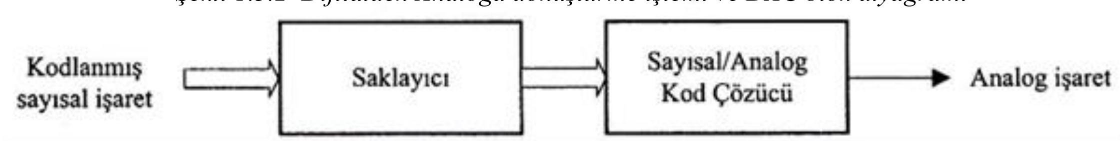
Kodlama:

Kuantalama düzeyine ikili sayı sisteminde bulunan birer kodlardan vermedir.

1.3.2 DAC(Digital To Analog Converter)

Dijitalden analoğa dönüştürücüdür. Yani sayısal veriden analog işaret üreten bir elemandır.ADC nin tam tersi işlemi yapar. Girişlerde bit cinsinden sayısal işaret uygulanır ve çıkışlarda analog işaret elde edilir.

Şekil 1.3.2 Dijitalden Analoga dönüştürme işlemi ve DAC blok diyagramı



Örnek 1.3. 4 bitlik bir DAC devresinin çıkış gerilim değeri 0-6 V arasında değişmektedir. 0000 sayısal girişi 0 V'a denk gelirken 1111 sayısal girişi ise 6 V'a denk gelmektedir. Bunun DAC dönüştürme tablosunu oluşturunuz.

$a = \frac{V_{max} - V_{min}}{2^n}$ → a = 0.375 Volt çıkar. Yada $a = \frac{V_{max} - V_{min}}{-1+2^n}$ → a = 0.4 Volt çıkar.

Sayısal Girişler	$a = \frac{V_{max} - V_{min}}{2^n}$	$a = \frac{V_{max} - V_{min}}{-1 + 2^n}$
0000	0.000	0.0
0001	0.375	0.4
0010	0.750	0.8
0011	1.125	1.2
0100	1.500	1.6
0101	1.875	2.0
0110	2.250	2.4
0111	2.625	2.8
1000	3.000	3.2
1001	3.375	3.6
1010	3.750	4.0
1011	4.125	4.4
1100	4.500	4.8
1101	4.875	5.2
1110	5.250	5.6
1111	5.652	6.0

BÖLÜM 2

2. SAYI SİSTEMLERİ

Sayma ve sayı kavramı bazı bulgular varsıtasıyla MS 400 dolaylarında Hindinstanda geliştiğini göstermektedir. Onluk sayı sistemi daha sonra İslam bilginleri tarafından geliştirildi ve MS 800 yıllarında onluk sayı sistemine 0'ı da dahil etmişlerdir ve böylelikle yeni rakam biçimleri kullanılmaya başlanmıştır. Onluk sayı sistemi ise Endülüs üzerinden Avrupa insanına aktartarıldıktan sonra günümüzde çoğu ülkenin de kullandığğı sayı biçimi ortaya çıkmıştır.

Onluk sayı sistemine insanların yatkın olmasına ragmen bilgisayar alemi için uygun değildir. Bundan dolayı çeşitli sayı düzenleri kullanılmaktadır.

Herhangi bir sayı sistemini 'S', 'd' sistemde kullanılan rakam-karakterleri ve 'R' ise sayi sisteminin kökünü temsil eder. Bunları S sistemine göre formulüze edersek;

$S = d_n R^n + d_{n-1} R^{n-1} + d_{n-2} R^{n-2} + \ldots\ldots + d_2 R^2 + d_1 R^1 + d_0 R^0$ ortaya çıkar.

2.1 Sayı Sistemlerinin İncelenmesi

2.1.1 Binary(İkili) Sayı Sistemleri

Bu sistemde iki adet sayı bulunur. Bu sayılar 0 ve 1'dir. Bundan dolayı sayı sisteminin tabanı 2'dir. Binary sayı sisteminde her basamak birer bit olarak adlandırılır ve en sağdaki en düşük bit değerini alır. Buna Least Significant Bit(LSB) denir. En soldaki ise en yüksek bit değerini alır. Buna ise Most Significant Bit(MSB) denir. Binary systemin basamak değerleri 2^n biçiminde ifade edilir. Basamak değerlerini hesaplamak için;

$S = d_n 2^n + d_{n-1} 2^{n-1} + d_{n-2} 2^{n-2} + \ldots\ldots + d_2 2^2 + d_1 2^1 + d_0 2^0$ şeklinde ifade edilebilir. Buradaki B bir sayı sistemini, d sistemde kullanılan rakam-karakterleri, 2 ise binary sistem olduğunu temsil eder.

Örnek 1.4. "1001" Binary sayı sistemini tabanlarına ayırınız.

$S = 1_3 2^3 + 0_2 2^2 + 0_1 2^1 + 1_0 2^0$ şeklinde ayrılır.

2.1.2 Octal(Sekizli) Sayı Sistemi

Sayı tabanı 8'dir. Bu sistemde kullanılan sayılar; 0, 1, 2, 3, 4, 5, 6, 7'dir. Toplam da 8 adet sayı olmasından dolayı octal(sekizli) sayı sistemi denilmiştir. Basmak değerleri;

$O = d_n 8^n + d_{n-1} 8^{n-1} + d_{n-2} 8^{n-2} + \dots\dots + d_2 8^2 + d_1 8^1 + d_0 8^0$ formülü ile ifade edilebilir.

Örnek 1.5. "1247,172" Octal tabaındali sayıyı basamaklarına ayırınız.

$O = 1.8^3 + 2.8^2 + 4.8^1 + 7.8^0 + 1.8^{-1} + 7.8^{-2} + 2.8^{-3}$ şeklinde ayrılır.

2.1.3 Decimal(Onluk) Sayı Sistemi

Günlük hayatımızda kullandığımız ve insan aklının aldığı sayı sistemidir.

Sayı tabanı 10'dur. Bu sistemde kullanılan sayılar 0, 1, 2, 3, 4, 5, 6, 7, 8, 9'dur. Toplam da 10 adet sayı olmasından dolayı decimal(onluk) sayı sistemi denilmiştir. Basamak değerleri;

$D = d_n 10^n + d_{n-1} 10^{n-1} + d_{n-2} 10^{n-2} + \dots\dots + d_2 10^2 + d_1 10^1 + d_0 10^0$ formülü ile ifade edebiliriz.

Örnek 1.6. "173,25" Decimal tabandaki sayıyı basamaklarına ayırınız.

$D = 1.10^2 + 7.10^1 + 3.10^0 + 2.10^{-1} + 5.10^{-2}$ şeklinde ayırırız.

2.1.4 Hexadecimal(Onaltılık) Sayı Sistemi

İkili sayı sisteminin daha kolay gösterilmesini sağlayan ve günümüz bilgisayarlarında yaygın olarak kullanılan sayı sistemidir. 0-9 arasındaki rakamların yanı sıra A, B, C, D, E, F harflerini de alır. Buraki harflerin belirli bir karşılığı vardır. Bunlar;

A=10, B=11, C=12, D=13, E=14, F=15'dir.

2.1.5 Kodlama ve Kodlar

Görülen, okunan yazı ve işaretlerin değiştirilmesine kodlama denilmektedir. Morse alfabesi ise kodlamaya en iyi örneklerden birisidir. Kodlama işlemi decimal sayı sistemindeki sayılar dışında alfabetik ve alfasayısal bilgilerin kodlanmasını da içerebilir.

2.1.5.1 BCD(Binary Coded Decimal Coded) Kodu

Onluk sistemde bulunan bir sayının, her basamağının ikilik sayı sistemindeki karşılığının 4'er bit olarak yazılmasına denir. Onluk sayı sistemindeki her bir sayı için ikilik sistemde 4 gerekir. Onlu sayı sistemini BCD sayı sistemine çevirmek için onlu sayı sistemindeki her bir sayı 4 bitlik ikili gurplar halinde yazılır ve yan yana getirilir. Böylece BCD kodları elde edilir.

Örnek 1.7. “$(263)_{10}$” Sayısının BCD kodunu yazınız.

1- Her basamaktaki sayının ikilik tabandaki karşılığı yazılır.
2- Elde edilen sayılar yan yana yazılarak BCD kodu elde edilir.

$\underset{0010}{2}$ $\underset{0110}{6}$ $\underset{0011}{3}$ sayıları elde edilir. Bu sayıları yan yana birleştirince de

$263_{10} = 001001100011_{BCD}$ seşitliği elde edilir.

2.1.5.2 Gray Kodu

Bu yöntem basamak ağırlığını göz önünde buludurmaz.yani her basamaktaki sayıların bir karşılığının olmamasıdır. Gray kodlama yöntemi minimum değişimli sınıflar arasında yer almaktadır. Bu yöntemde basamak ağırlığı göz önünde bulundurulmadığından aritmetik işlemlerde kullanılamaz. Bunun yerine analog – dijital çeviricilerinde ve giriş çıkış birimlerinde kullanılabilmektedir. Bunun nedeni ise sütun esasına göre çalışan cihazlarda hatayı azaltmasıdır.

- Binary Sayının Gray Koduna Çevrilmesi

Bu dönüşümü yapabilmek için en yüksek basamak değerine sahip bitin solunda '0’ var olarak kabul edilir. Bundan sonra her bit en soldan başlanarak solundaki bitle toplanarak yazılır. Bu işlem en düşük değerli basamağa kadar devam edilir. Bu işlemler sonucunda da gray kodu elde edilmiş olur.

Örnek 1.8. $(101110101)_2$ Binary sistemdeki sayıyı Gray koduna çeviriniz.

$(101110101)_2 = (1100111)$ sonucu elde edilir.

Örnek 1.9. $(1000101)_2$ Binary sistemdeki sayıyı Gray koduna çeviriniz.

$(1000101)_2 = (1100111)$ sonucu elde edilir.

- Gray Kodun Binary Koduna Çevrilmesi

Bu dönüşümü yapabilmek için en soldaki bit direk olarak aşağıya indirilir. İndirilen sayı ve bir solundaki sayı toplanarak yazma işlemi yapılır.

Örnek 2.0. $(111001111)_{GRAY}$ Gray Kodunu Binary koduna çeviriniz.

1 1 1 0 0 1 1 1 1 Gray kodlu sayı

1 0 1 1 1 0 1 0 1 İkili sayı

$(111001111)_{GRAY} = (101110101)_2$ sonucu elde edilir.

2.1.5.3 Artı 3 Kodu(Excess 3)

Bir Decimal sayının Artı 3 kodundaki karşılığı, onlu sayının karşılığı olan Binary sayıya +3 eklenmiş halidir. Yani Decimal bir sayıyın her basamağına 3 ekleyip onu da Binary koda dönüştürmemiz yetecektir.

<u>Örnek 2.1.</u> $(48)_{10}$ sayısını Artı 3 koduna dönüştürünüz.

$$\begin{array}{cc} 4 & 8 \\ \underline{+\ 3} & \underline{+\ 3} \\ 7 & 11 \\ \downarrow & \downarrow \\ 0111 & 1011 \end{array}$$

$(48)_{10} = (01111011)_{+3}$ sonucu elde edilir.

<u>Örnek 2.2.</u> $(10100110)_{+3}$ kodunu Decimal koduna dönüştürünüz.

Dönüştürülecek sayı 4'er bitlik gruplara ayrılır ve her grubun Decimal karşılığı bulunur. Bulunduktan sonra da 3 çıkarılır.

$(10100110)_{+3} \rightarrow 1010\ 0110 \rightarrow (10\ 6)_{+3}$ sayıları elde edilir. Elde edilen sayılardan 3 çıkarılırsa $(10100110)_{+3} = (73)_{10}$ Decimal kodu elde edilir.

2.2 Sayı Sistemlerinin Dönüştürülmesi

2.2.1 Binary Sayıların Decimal, Octal ve Hexadecimal Dönüşümü

- Binary Sayıların Decimal Sayılara Dönüşümü

<u>Örnek 2.3.</u> $(11001)_2$ Sayısını decimal sayı sistemine dönüştürünüz.

$S = d_n 2^n + d_{n-1}2^{n-1} + d_{n-2}2^{n-2} + \dots\dots + d_2 2^2 + d_1 2^1 + d_0 2^0$'den yararlanarak yapalım.

$S = 1.2^4 + 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0 \rightarrow (25)_{10}$'dir.

Örnek 2.4. $(100.01)_2$ Sayısını decimal sayı sistemine dönüştürünüz.

$S = d_n 2^n + d_{n-1} 2^{n-1} + d_{n-2} 2^{n-2} + \ldots\ldots + d_2 2^2 + d_1 2^1 + d_0 2^0$'den yararlanarak yapalım.

$S = 1.2^2 + 0.2^1 + 0.2^0,\ 0.2^{-1} + 0.2^{-2} \rightarrow (4.25)_{10}$'dir.

- Binary Sayıların Octal Sayılara Dönüşümü

Örnek 2.5. $(11001111011101)_2$ Sayısını octal sayı sistemine dönüştürünüz.

Octal sayı sistemine dönüştürebilmemiz için sayıları en sağdan başlayarak üçerli gruplara ayırmamız gerekir. Eksik kalanin başına da 0 eklemesi yapılır. Her üçerli gruba karşılık gelen onluk sayısı yazılır.

$(011\ \ 001\ \ 111\ \ 011\ \ 101)_2 \rightarrow (3\ 1\ 7\ \ 3\ 5)_8$ elde edilir.

- Binary Sayıların Hexadecimal Sayılara Dönüşümü

Örnek 2.6. $(10111110111000011101)_2$ Sayısını hexadecimal sayı sistemine dönüştürünüz.

Verilen ifade dörder bitlik gruplara ayrılır ve hexadecimal karşılıları yazılır. Ayırma işlemi yapılırken en sağdan başlanır.

$(1011\ \ 1101\ \ 1100\ \ 0011\ \ 1101)_2 \rightarrow (B\ D\ C\ 3\ D)_{16}$ elde edilir.

2.2.2 Octal Sayıların Binary, Decimal Dönüşümleri

- Octal Sayıların Binary Sayılara Dönüşümü

Örnek 2.7. $(673.124)_8$ Sayısının binary sayılara dönüştürünüz.

Her bir sayının kaç bite denk geldiği bulunur ve bitler yanyana yazılır.

$6 \rightarrow 0110_2$ $\quad$ $7 \rightarrow 0111_2$ $\quad$ $3 \rightarrow 0011_2$ $\quad$ $1 \rightarrow 0001_2$ $\quad$ $2 \rightarrow 0010_2$ $\quad$ $4 \rightarrow 0100_2$

$(673.124)_8 \rightarrow (011001110011.00100100100)_2$'dir.

- Octal Sayıların Decimal Sayılara Dönüşümü

Örnek 2.8. $(372)_8$ Decimal sayı sistemine dönüştürünüz.

$O = d_n 8^n + d_{n-1} 8^{n-1} + d_{n-2} 8^{n-2} + \ldots\ldots + d_2 8^2 + d_1 8^1 + d_0 8^0$ yararlanarak çözelim.

$O = 3.8^2 + 7.8^1 + 2.8^0 \rightarrow (250)_{10}$'dir.

2.2.3 Decimal Sayıların Binary, Octal ve Hexadecimal Dönüşümü

- Decimal Sayıların Binary Sayılara Dönüşümü

Decimal bir sayıyı binary sayıya dönüştürmek için decimal sayı sürekli 2'ye bölünür. En son kalan sayı da 2 ye bölündükten sonra bölümden başlanarak geriye doğru kalanları yan yana yazılır.

Örnek 2.9. $(39)_{10}$ Sayısının binary karşılığını yazınız.

İŞLEM	BÖLÜM	KALAN
$^{39}/_{2}$	19	1
$^{19}/_{2}$	9	1
$^{9}/_{2}$	4	1
$^{4}/_{2}$	2	0
$^{2}/_{2}$	1	0

Bölümün en son kalan elemanından başlanarak kalanlarla beraber yan yana yazılır.

Sonuç olarak: $(39)_{10} \rightarrow (100111)_2$ sonucu elde edilir.

- Decimal Sayıların Octal Sayılara Dönüşümü

Decimal bir sayıyı octal sayıya dönüştürmek için decimal sayı sürekli 8'e bölünür. En son kalan sayı da 8'e bölündükten sonra bölümden başlanarak geriye doğru kalanları yan yana yazılır.

Örnek 3.0. $(153)_{10}$ Sayısının octal karşılığını yazınız.

İŞLEM	BÖLÜM	KALAN
$^{153}/_{8}$	19	1
$^{19}/_{8}$	2	3
2		

Sayı 8 e bölünmeyene kadar bölünür ve bölüm – varsa işlem – kısmından itibaren yan yana yazılmaya başlanır. Sonnuç olarak: $(153)_{10} \rightarrow (231)_8$ sonucu elde edilir.

- Decimal Sayılarım Hexadecimal Sayılara Dönüşümü

Decimal bir sayıyı hexadecimal sayıya dönüştürmek için decimal sayı sürekli 16'ya bölünür. En son kalan sayı da 16'ya bölünür. 10 ve üzerindeki sayıların hexadecimal karşılığı yazılır.

Örnek 3.1. $(214)_{10}$ Sayısının hexadecimal karşılığını yazınız.

İŞLEM	**BÖLÜM**	**KALAN**
$^{214}/_{16}$	13	6
$^{13}/_{16}$	0	13 → D

Yani sonuç olarak: $(214)_{10} \rightarrow (6D)_{16}$ sonucu elde edilir.

2.2.4 Hexadecimal Sayıların Binary, Octal ve Decimal Dönüşümü

- Hexadecimal Sayıların Binary Sayılara Dönüşümü

Hexadecimal sayıların binary sayılara dönüştürmek için hexadecimal sayıların her basamağında bulunan sayıların 4'er bitlik karşılığı yazılır. Karşılıklar yazıldıktan sonra yan yana yazılarak binary karşılığı bulunmuş olur.

Örnek 3.2. $(5D1D69)_{16}$ Sayısının binary karşılığını yazınız.

BASAMAK	**4 BİT KARŞILIĞI**
5	0101
D	1101
1	0001
D	1101
6	0110
9	1001

$(5D1D69)_{16} \rightarrow (010111010001110101101001)_2$ sonucu elde edilir.

- Hexadecimal Sayıların Octal Sayılara Dönüşümü

Hexadecimal sayıları octal sayılara dönüştürmek için ilk önce binary sisteme dönüştürülür. Binary sistemde octal sisteme dönüştürülerek sonuc bulunmuş olunur.

Örnek 3.3. $(E0CA)_{16}$ Sayısının octal karşılığını yazınız.

BASAMAK	**4 BİT KARŞILIĞI**
E	1110
0	0000
C	1100

A	1010

$(E0CA)_{16} \rightarrow (1110000011001010)_2$

Elde edilen bu binary sayılar 3'erli gruplara ayrılarak octal sistemdeki karşılığı yazılır.

$(1110000011001010)_2 \rightarrow (160312)_8$

$(E0CA)_{16} \rightarrow (160312)_8$ sonucu elde edilir.

- Hexadecimal Sayıların Decimal Sayılara Dönüşümü

$H = d_n 16^n + d_{n-1}16 + d_{n-2}16^{n-2} + \ldots\ldots + d_2 16^2 + d_1 16^1 + d_0 16^0$'dan yararlanarak yapılabilir.

Örnek 3.4. $(E70DCA)_{16}$ Sayısının decimal karşılığını yazınız.

$(E70DCA)_{16} \rightarrow E.16^5 + 7.16^4 + 0.16^3 + D.16^2 + C.16^1 + A.16^0$

$\rightarrow (2307513)_{10}$ sonucu elde edilir.

2.3 Sayı Sistemlerinde Hesaplama

2.3.1 İkili Sayı Sisteminde Toplama ve Çıkarma,

- Toplama

Binary sayı sistemindeki topla aynı decimal sayı sistemindeki toplamalara benzer.

Toplama Kuralları:

0 + 0 = 0, 0 + 1 = 1, 1 + 1 = 10 veya 1 + 1 = 0 elde 1 (C=1)

1 + 1 toplamında sonuc 0 olur ve elde bir yan basamağa aktarılmak üzere 1 olur.

Örnek 3.5.

$$\begin{array}{r} 10 \\ +\ 01 \\ \hline 11 \end{array} \qquad \begin{array}{r} 101 \\ +\ 010 \\ \hline 111 \end{array} \qquad \begin{array}{r} 111011 \\ 011011 \\ 110101 \\ +\ 010010 \\ \hline 10011101 \end{array}$$

- Çıkarma

Çıkarma Kuralları:

0 - 0 = 0, 0 - 1 = 1(borç 1), 1 - 1 = 0, 10 – 1 = 0, 1 - 0 = 1

Örnek 3.6.

$$\begin{array}{r} 10110 \\ -\quad 1101 \\ \hline 1001 \end{array} \qquad \begin{array}{r} 101110 \\ -\quad 10011 \\ \hline 11011 \end{array}$$

2.3.2 İkili Sayı Sisteminde Çarpma ve Bölme

- Çarpma

Çarpma Kuralları:

0 x 0 = 0, 0 x 1 = 0, 1 x 0 = 0, 1 x 1 = 1

Örnek 3.7.

$$\begin{array}{r} 1011 \\ \times\quad 101 \\ \hline 1011 \\ 0000 \\ +\quad 1011 \\ \hline 110111 \end{array} \qquad \begin{array}{r} 1011 \\ \times\quad 110 \\ \hline 00000 \\ 1011 \\ +\quad 1011 \\ \hline 10001010 \end{array}$$

- Bölme

Decimal sayı sistemindeki bölme işlemleri gibi yapılır.

Örnek 3.8. $\frac{(10110)_2}{(100)_2}$ işleminin sonucu nedir?

```
  10110 | 100
- 100     101,1
  00110
-   100
    0100
-    100
     000
```

Sonuç: $(101.1)_2$ olarak bulunur.

BÖLÜM 3

3. BOOLEAN KURALLARI VE LOJİK İFADELERİN SADELEŞTİRİLMESİ

Boolean matematiği George Bole tarafından 1854 yılında geliştirilmiş olan Boolean matematiği 1938 yılında Claude Shanon tarafından sayısal devrelerin tasarımı ve analizinde kullanılmaya başlanmıştır.

Boolean matematiği sayısal sistemlerin analizinde ve anlaşılmasında kullanılan sistemdir. Boolean matematiğinde değişkenler iki değer alabilir. 1-0 gibi. Boolean matematiğinde değişken üstünde bir çizgi ile gösterilir

Örneğin $\bar{A}$ ifadesi A'nın değili şeklinde okunur. Eğer A = 1 ise $\overline{A} = 0,$ A = 0 ise $\overline{A} = 1$ olur.

A'nın değili (A') şeklinde yazılabilir.

A ve B girişlerine uygulanan VE fonksiyonu Boolean ifadesi ''A.B'' şeklinde yazılırken, VEYA fonksiyonu ''A+B'' şeklinde yazılır.

3.1 Önemli Boolean Özellikleri

3.1.1 Değişim Kanunu

Normal toplama ve çarpma işlemlerindeki özellik ile aynıdır.

A + B = B +A　　　　A.B = B.A

3.1.2 Birleşme Kanunu

Normal toplama ve çarpma işlemlerindeki özellik ile aynıdır.

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

3.1.3 Dağılma Kanunu

Toplamanın çarpma, çarpmanın toplama üzerindeki dağılma özellikleri olan kurallar aynen Boolean matematiğinde de geçerlidir.

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

3.1.4 Yutma Kanunu

Yalnızca Boolean cebirinde geçerlidir.

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

3.1.5 Basitleştirme Kanunu

Normal toplama ve çarpma işlemkerindeki özelliklerle aynıdır.

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

3.1.6 De Morgan Kanunu

"VEYADEĞİL" ve "VEDEĞİL" işlemlerinden yararlanılarak uygulanan ve lojik işlemlerde kolaylık sağlayan kurallardır.

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

3.2 Boolean Kurallarını Kullanarak Lojik İfadelerin Sadeleştirilmesi

Karmaşık lojik ifadeler boolena matematiği kurallarından yararlanılarak sadeleştirme işlemleri yapılabilir. Böylelikle yapılacak elektronik devreler hem ucuz hemde basit bir şekilde yapılabilecektir.

Örnek 3.9. A + B + C = A'.B'.C' ispatını yapınız.

(B + C = X) olarak Kabul edelim.

$\overline{A+B+C}$ = $\overline{A+X}$ = A'.X' = A'. ($\overline{B+C}$) = A'.(B'.C') = A'.B'.C' şeklindedir.

Örnek 4.0. F = A'.B + A + A.B ifadesini sadeleştiriniz.

'B' parantezine alınırsa:

A'B + A + A.B = B.(A + A') + A = B.1 + A = A+B şeklindedir.

Bu yol ile bulunabileceği gibi farklı yollarla da cevaplar bulunabilir. Bir sorunun birden fazla çözümü olabilir.

Örnek 4.1. A’.B’.C + A’.B.C + A.B’ ifadesini sadeleştiriniz.

$\mathrm{A'.B'.C + A'.B.C + A.B' = A'.C.}\underbrace{\mathrm{(B + B')}}_{1}\mathrm{ + A.B' = A'.C + A.B'}$ şeklinde bulunur.

3.3 Doğruluk Tablosu

Lojik devrelerde, giriş değişkenlerinin alabilecekleri sayısal değerleri(kombinasyonları) ve sayısal degerlere göre çıkışların durumunu gösteren tablolara denir. Ya da başka bir deyiş ile doğruluk tablosu önerme eklemleriyle kurulmuş bileşik önermelerin tutarlılığını, eşdeğerliliğini ve çıkarımların geçerliliğini denetleyen bir yöntemdir.

Peki biz doğruluk tablosunu nerede kullanacağız:

Aslında hayatımızın hemen hemen her yerinde farkında olarak ya da olmayarak kullanıyoruz. Nasıl diyecek olursanız; Fatih Sultan Mehmet 1461 de Trabzon Rum devletini fetih etti derken bunu gercek doğru oldugunu biliyoruz. Biz bu gerçek doğruları matematiksel önerme olarak 1 diye nitelendiriyoruz. Netflixde gösterimde olan Rise of Empires: Ottoman filminde Fatih Sultan Mehmet’in babası tarafından sürgüne gönderildidği söyleniyor ve bu yanlış bir bilgidir. Bunu da matematiksel önerme olarak 0 diye nitelendiriyoruz.

Lojik devrelerde n tane giriş değişken için 2^n tane durum oluşur. Örneğin 3 değişken girerse 2^3’den 8 tane durum oluşması gerekir. Matematiksel işlemlerden + “VEYA” yı x ise “VE” yi temsil eder. İşlemler yapılırken onları göz önüne almamız gerekiyor.

Örnek 4.2. Giriş değişkeni A ve B olan bir sistemde A + B işlemi gerçekleşince çıkış giriş ve çıkış değerlerini gösteren tabloyu yapınız.

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

Örnek 4.3. Giriş değişkeni A ve B olan bir sistemde A.B işlemi gerçekleşince çıkış giriş ve çıkış değerlerini gösteren tabloyu yapınız.

A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

Örnek 4.4. Giriş değişkenleri A ve B olan bir sistemde bu giriş değişkenleri ile oluşturulabilecek bütün işlemlerin giriş ve çıkış değerlerini doğruluk tablosu ile yapını.

A	B	A'	B'	$A+B$	$A.B$	$A+A'$	$A.A'$	$B+B'$	$B.B'$	$A+B'$	$A'+B$
0	0	1	1	0	0	1	0	1	0	1	1
0	1	1	0	1	0	1	0	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	1	0	0	1	1	1	0	1	0	0	1

Örnek 4.5. $\overline{A+B} = A' + B'$ De Morgan kuralını doğruluk tablosu ile ispat ediniz.

A	A'	B	B'	$A+B$	$\overline{A+B}$	$A'.B'$
0	1	0	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0

Örnek 4.6. $F = A + A.B = A$ olduğunun isptaını doğruluk tablosu ile gösteriniz.

A	B	$A.B$	$A+A.B$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Örnek 4.7. $F = A.(B+C) = (A.B) + (A.C)$ olduğunun isptaını doğruluk tablosu ile gösteriniz.

A	B	C	$B+C$	$A.(B+C)$	$A.B$	$A.C$	$(A.B)+(A+C)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

3.4 Temel Açılımlar ve Standart İfadeler

Bir binary değişken, kendi normal formu olan A olarak veya değili olan A' formu ile ifade edilebilir. Bu formlarla ifade edilebilen değişkenler fonksiyon halini aldığı zaman; kanun kaide olarak adlandırılan 'minterm' (çarpımların toplamı) veya 'maxterm' (toplamların çarpımı) modellerinden biri ile gösterilir.

DEĞİŞKEN			MİNTERMLER		MAXTERMLER	
A	B	C	Terim	İsim	Terim	İsim
0	0	0	$A'.B'.C'$	m_0	$A+B+C$	M_0
0	0	1	$A'.B'.C$	m_1	$A+B+C'$	M_1
0	1	0	$A'.B.C'$	m_2	$A+B'+C$	M_2
0	1	1	$A'.B.C$	m_3	$A+B'+C'$	M_3
1	0	0	$A.B'.C'$	m_4	$A'+B+C$	M_4
1	0	1	$A.B'.C$	m_5	$A'+B+C'$	M_5
1	1	0	$A.B.C'$	m_6	$A'+B'+C$	M_6
1	1	1	$A.B.C$	m_7	$A+B+C$	M_7

Minterm: Bir boolean ifadede bulunan değişkenlerin sahip olduğu veya oluşturabileceği kombinasyonların "VE" işlem sonucu 1 olacak şekilde uyarlanmasına denir.

Maxterm: Bir boolean ifadede bulunan değişkenlerin sahip olduğu veya oluşturabileceği kombinasyonların "VEYA" işlem sonucu 0 olacak şekilde uyarlanmasına denir.

Bir Boolean fonksiyonu doğruluk tablosundan belirli kombinasyonların seçilmesi, seçilen kombinasyonların sonuç olacak şekilde formlandırılması ve formlandırılan kombinasyonların 'VEYA' işlemine tabi tutulması şeklinde tanımlanabilir.

Örnek 4.8. Doğruluk tablosundaki f_1 ve f_2 fonksiyonlarının maxterm ve mintermlerini bulunuz.

A	B	C	f_1	f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

$f_1$ve f_2 fonksiyonunda 1 olanlar mintermdir.

f_1 fonksiyonunda 1 olarak tanımlanan değişken değerleri: 001, 100, 111
Yani f_1 fonksiyonu:

$$f_1 = m_1 + m_4 + m_7 = A'.B'.C + A.B'.C' + A.B.C$$

f_2 fonksiyonunda 1 olarak tanımlanan değişken değerleri: 011, 101, 110, 111
Yani f_2 fonksiyonu:

$$f_2 = m_3 + m_5 + m_6 + \ m_7 = A'.B.C + A.B'.C + A.B.C' + A.B.C$$

$f_1$ve f_2 fonksiyonunda 0 olanlar maxtermdir.

f_1 fonksiyonunda 0 olarak tanımlanan değişken değerleri: 000, 010, 011, 101, 110
Yani f_1 fonksiyonu:

$$f_2 = M_0.M_2.M_3.M_5.M_6 = (A + B + C).(A + B' + C).(A + B' + C').(A' + B + C').(A' + B' + C)$$

f_1 fonksiyonunda 0 olarak tanımlanan değişken değerleri: 000, 001, 010, 100
Yani f_1 fonksiyonu:

$$f_2 = M_0.M_1.M_2.M_4 = (A + B + C).(A + B + C').(A + B' + C).(A + B' + C).(A' + B + C)$$

şeklinde tanımlanır.

3.4.1 Mintermlerin Toplamı ve Maxtermlerin Çarpımı

Sadeleştirilmiş olarak verilen bir fonksiyonu mintermlerin toplamı şeklinde ifade etmek için, herbir minterm ifadesi bütün değişkenleri kapsayacak biçimde genişletilir. Herhangi bir kombinasyonda bulunmayan değişkenleri eklemek için kombinasyon $(x + x')$ terimi ile 'VE' işlemine tabi tutulur($x, kombinasyonlarda\ bulunmayan\ bir\ değişkeni\ temsil\ etmektedir$).

Örnek 4.9. $\mathrm{f} = \mathrm{A} + \mathrm{B}'.\mathrm{C}$ ifadesini mintermlerin toplamı şeklinde ifade ediniz.

İlk kombinasyonda sadece A olduğundan oraya B ve C değişkenleri de eklenmelidir. İlk minterme B eklenmelidir. Ekleme yaparken . $B + B'$ ile çarpmalıyız. Ardından da elde ettiğimiz sonucu $C + C'$ ile çarpmalıyız.

$A = A.(B + B') = A.B + A.B'$
$A.B + A.B' = (A.B + A.B').(C + C') = A.B.C + A.B'.C + A.B.C' + A.B'.C'$ eşitliği ekde edilir.

İkinci kombinasyon $B'.C$ olduğundan A değişkeni eklenmesi gerekir.

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

$f = A.B.C + A.B'.C + A.B.C' + A.B'.C' + A.B'.C + A'.B'.C$
$f(A,B,C) = \sum(1,4,5,6,7)$

Örnek 5.0. $f = (A' + B).(A + C).(B + C)$ makstermlerini çarpım şeklinde ifade ediniz.

Verilen kombinasyonda bir değişken eksik olduğundan;

$A' + B = A + B + C.C' = (A' + B + C).(A' + B + C')$
$A + C = A + C + B.B' = (A + B + C).(A + B' + C)$
$B + C = B + C + A.A' = (A + B + C).(A' + B + C)$ eşitlikleri elde edilir.

Bu ifadeleri yan yana getirirlir ve aynı olan ifadeler sadece bir kere yazılır.

$F = (A + B + C).(A + B' + C).(A' + B + C).(A' + B + C')$
$= M_0.M_2.M_4.M_5$
$F(A,B,C) = \prod(0,2,4,5)$

3.4.2 Minterm ve Maxterm İfadelerinin Birbirine Dönüştürülmesi

Minterm ve maxtermin elde edilişine bakarak birbirlerinin tersi olduğu anlaşılır. Çünkü mintermler için fonskiyondaki 1 değerleri maxterm içinde 0 değerleri alınır.

Örnek 5.1. $f(A,B,C) = \sum(1,4,5,6,7)$ fonksiyonun tümleyenini(tersini) gösteriniz.

$$f_{(A,B,C)}{}' = m_1 + m_4 + m_5 + m_6 + m_7$$

$$f_{(A,B,C)}{}' = m_0 + m_2 + m_3$$

$$f' = m_0 + m_2 + m_3 \rightarrow f = \overline{m_0 + m_2 + m_3} = m_0'.m_2'.m_3'$$

$$= M_0 + M_2 + M_3$$

$$= \prod(0,2,3)$$

BÖLÜM 4

4. LOJİK KAPILAR VE LOJİK DEVRELER

Lojik devrelerin en basit elemanı lojik kapılardır.

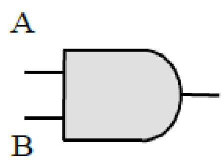
Lojik Kapılar: Lojik değişkenlerin değerlerini giriş olarak kullanan ve aldığı çıkş değerleri üzerinde işlemler yaparak lojik eşitliğin değerine uygun değerler üreten elektronik devredir.

Lojik Devre: Lojik kapıların kullanılması ile oluşturulan devrelerdir. Bu devreler “donanım” olarak da tanımlanmaktadır.

Aşağıdaki şekilde lojik kapıların sembolleri, gerçekleştirildikleri işlemler, doğruluk tabloları ve elektriki eşdeğerleri görülmekteir. Aşağıda yaygın olarak kullanılan kapılara da yer verilmiştir.

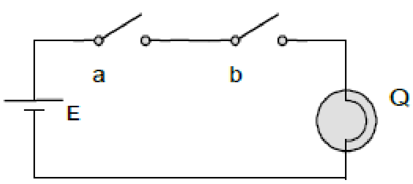
İşlemin Adı Sembolü Yaptığı İşlem Doğruluk Tablosu Elektriksel Eşdeğeri

VE (AND) KAPISI

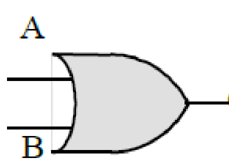


Q=A.B

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

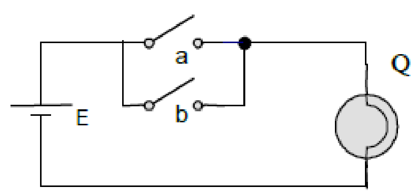


VEYA (OR) KAPISI

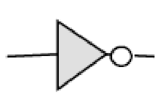


Q=A+B

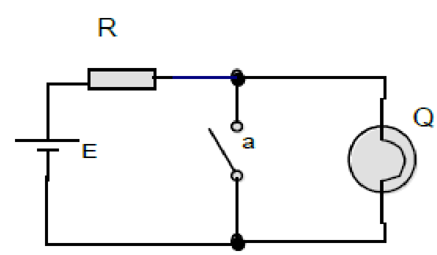
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1



DEGİL (NOT) KAPISI

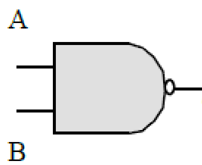
A  $Q=A^{!}$

A	Q
0	1
1	0



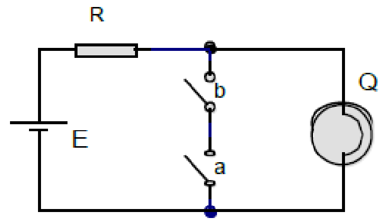
İşlemin Adı — Sembolü — Yaptığı İşlem — Doğruluk Tablosu — Elektriksel Eşdeğeri

VEDEGİL (NAND) KAPISI

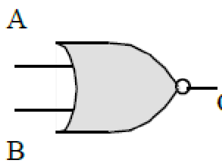


$Q=\overline{A \,.\, B}$

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

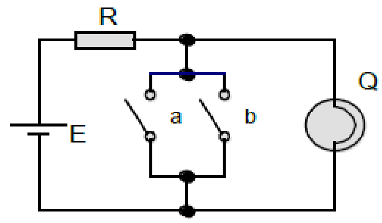


VEYADEGİL (NOR) KAPISI

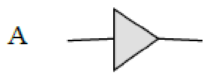


$Q=\overline{A + B}$

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

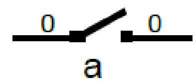


SÜRÜCÜ (BUFFER) KAPISI

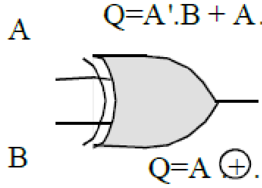


Q=A

A	Q
0	0
1	1



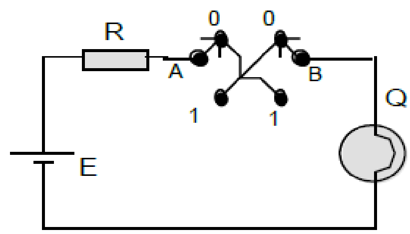
ÖZELVEYA (EXOR) KAPISI



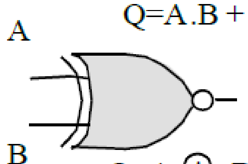
$Q=A'.B + A.B'$

$Q=A \oplus B$

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0



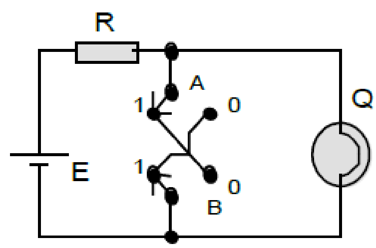
ÖZELVEYA DEGİL (EXNOR) KAPISI



$Q=A.B + A'.B'$

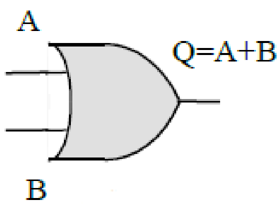
$Q=A \circledast B$

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

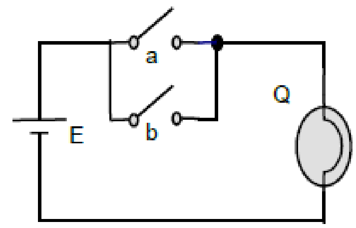


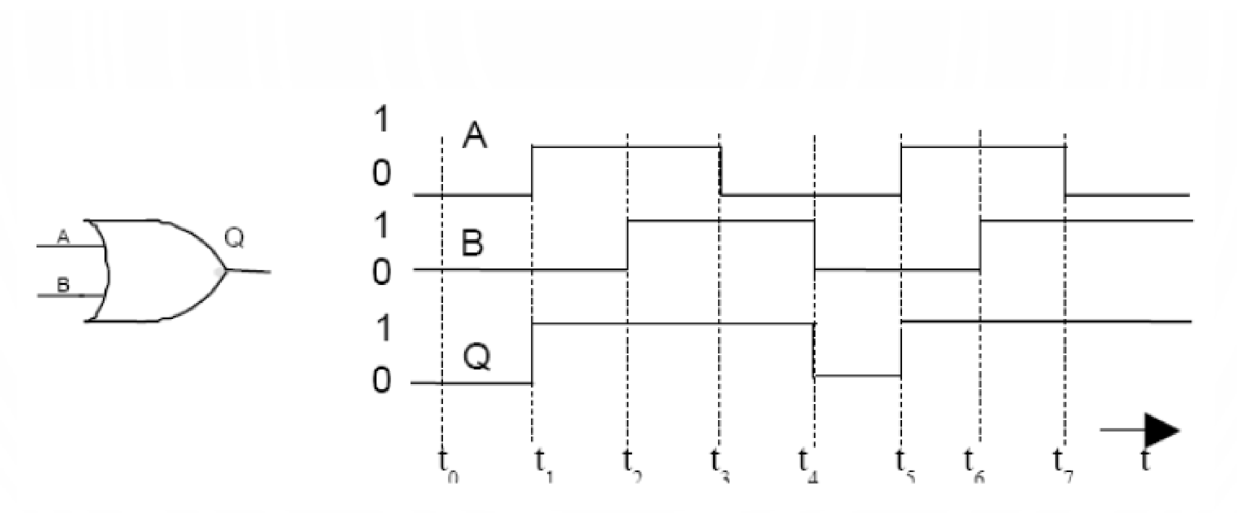
4.1 “VEYA” İşlemi ve “VEYA” Kapısı

VEYA ‘nın diğer adı ingilizcedeki OR’dur. VEYA-OR işlemine tabi tutulan A ve B adında iki adet giriş değişkeni ile doğruluk tablosunda bir Q çıkış değerleri elde edilir.



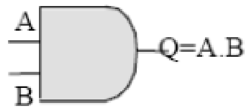
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1





4.2 “VE” İşlemi ve “VE” Kapısı

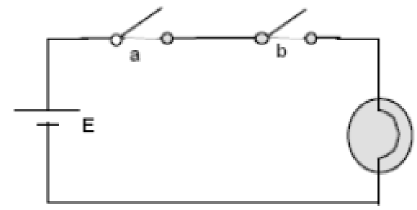
VE ‘nin diğer adı ingilizcedeki AND’dir. VE-AND işlemine tabi tutulan A ve B adında iki adet giriş değişkeni ile doğruluk tablosunda bir Q çıkış değerleri elde edilir



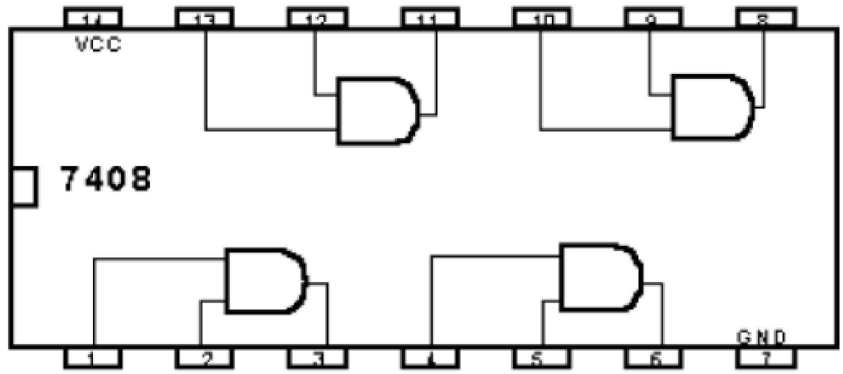
(a)

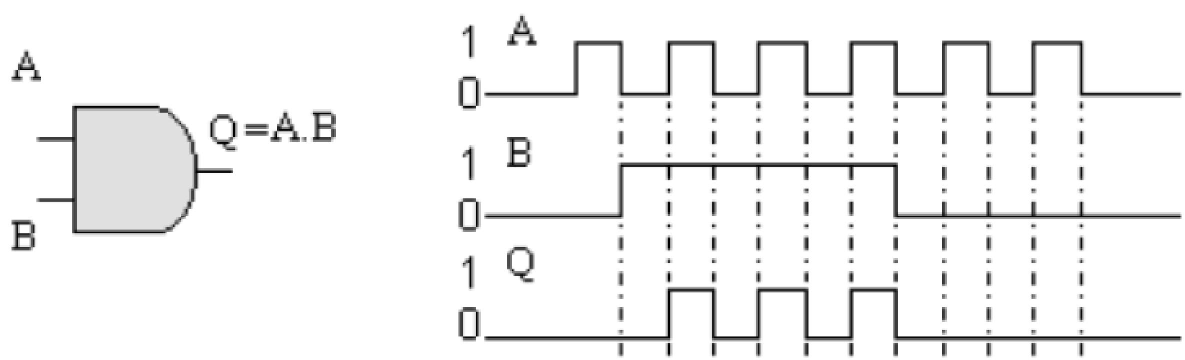
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

(b)



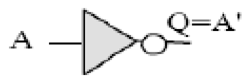
(c)



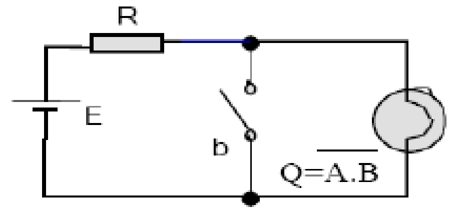


4.3 “DEĞİL” İşlemi ve “DEĞİL” Kapısı

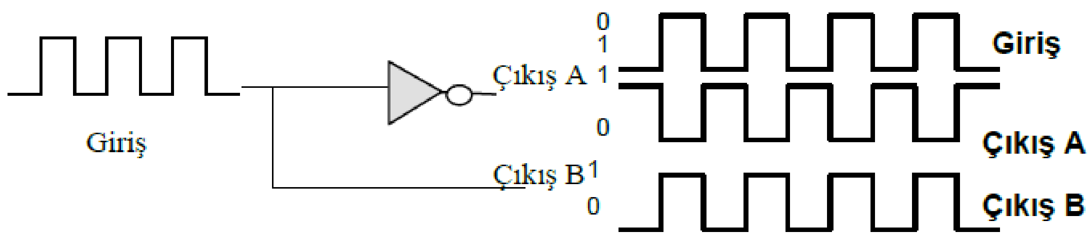
“DEĞİL” in diğer adı ingilizcedeki “NOT”dır. “DEĞİL” işlemi “VE”, “VEYA” dan farklı olarak bir adet giriş ve bir adet çıkış değişkenleri ile gerçekleştirilir. Örneğin A değişkeni ile işlem yapılacaksa “DEĞİL” işlemi sonucu $Q = A'$olarak tanımlanır. A değişkeni iki değer alabilir. 1 veya 0’ı alır. A’nın değeri 1 ise değili yani A' 0 olur. A’nın değeri 0 ise değili yani A' 1 olur. “DEĞİL” kapısı her zaman tek girişlidir ve çıkış sürekli girişn tersi biçimdedir.



A	Q
0	1
1	0



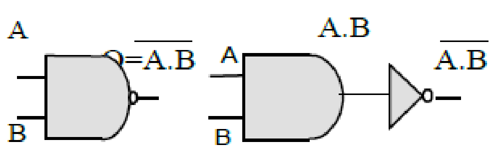
<u>Örnek 5.2.</u> “DEĞİL” kapısı kullanarak birbirinin tersi iki sinyal üreten lojik devreyi yapınız.



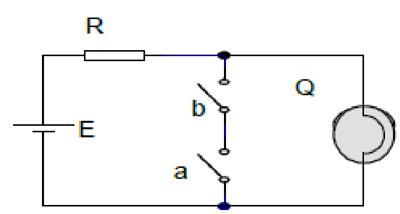
‘DEĞİL’ kapısı kullanılarak oluşturulan devrede, girişe uygulanan kare dalga sinyal ile aynı ve 180 derece ters fazında iki sinyal çıkış olarak oluşur.

4.4 “VEDEĞİL” Kapısı

“VE” ile “DEĞİL” kapısının birleşmesi sonucunda ortaya çıkmıştır. İngilizce de ismi “NAND” olarak geçmektedir. “NAND” kapısında giriş değerlerinden birisi 1 ise çıkış 0 olur. Giriş değişkeninin 0 olması durumunda da çıkış değeri 1 olur. “VEDEĞİL” işleminin çıkış fonksiyonu $Q = \overline{A.B}$ şeklindedir. “VEDEĞİL” kapısı “VE” ile “DEĞİL” kapısının seri bağlanması ile oluşturulmuştur.

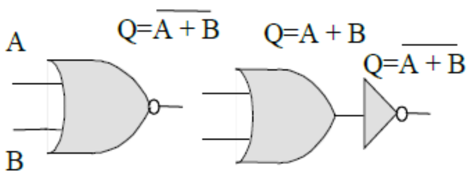


A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

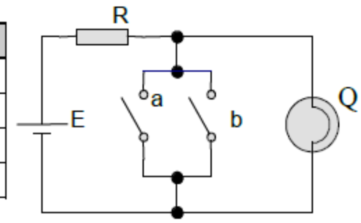


4.5 “VEYADEĞİL” Kapısı

“VEYA” ile “DEĞİL” kapısının birleşmesi sonucunda ortaya çıkmıştır. İngilizce de ismi “NOR” olarak geçmektedir. “NOR” kapısında giriş değerlerinin tümü 0 ise çıkış 1 olur. “VEYADEĞİL” işleminin çıkış fonksiyonu $Q = \overline{A + B}$ şeklindedir.

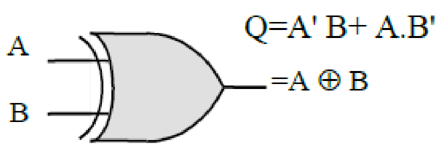


A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

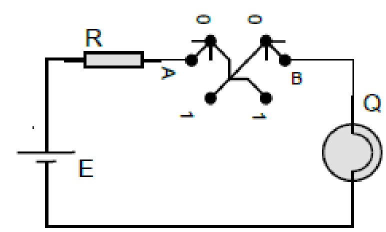


4.6 “ÖZEL VEYA” Kapısı.

Bu kapının iki adet giriş ve bir adet çıkışı bulunur. Giriş değişkenler aynı ise çıkış değişkeni 0 olur. Giriş değişkenleri farklı ise çıkış değişkeni 1 olur.

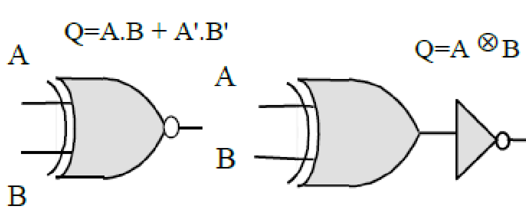


A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

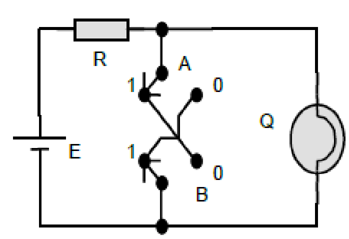


4.7 “ÖZEL VEYADEĞİL” Kapısı

Bu kapının iki adet giriş ve bir adet çıkışı bulunur. Giriş değişkenler aynı ise çıkış değişkeni 1 olur. Giriş değişkenleri farklı ise çıkış değişkeni 0 olur.



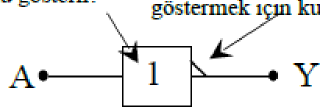
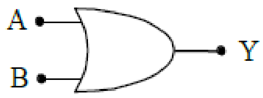
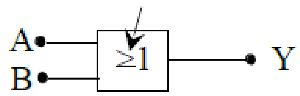
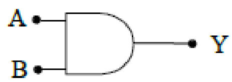
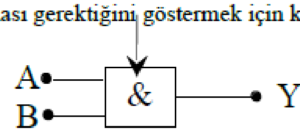
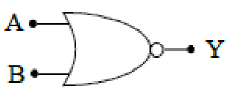
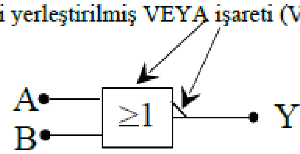
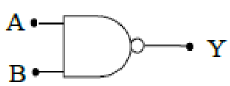
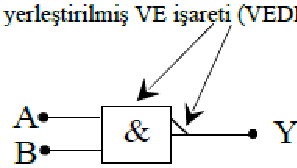
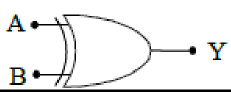
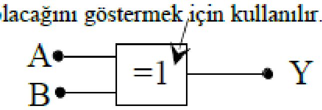
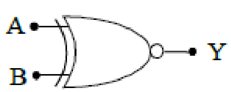
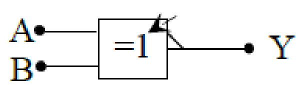
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1



4.8 Lojik İfadelerin Lojik Elemanlarla Gerçekleştirilmesi ve Lojik Devrelerin Tasarımı

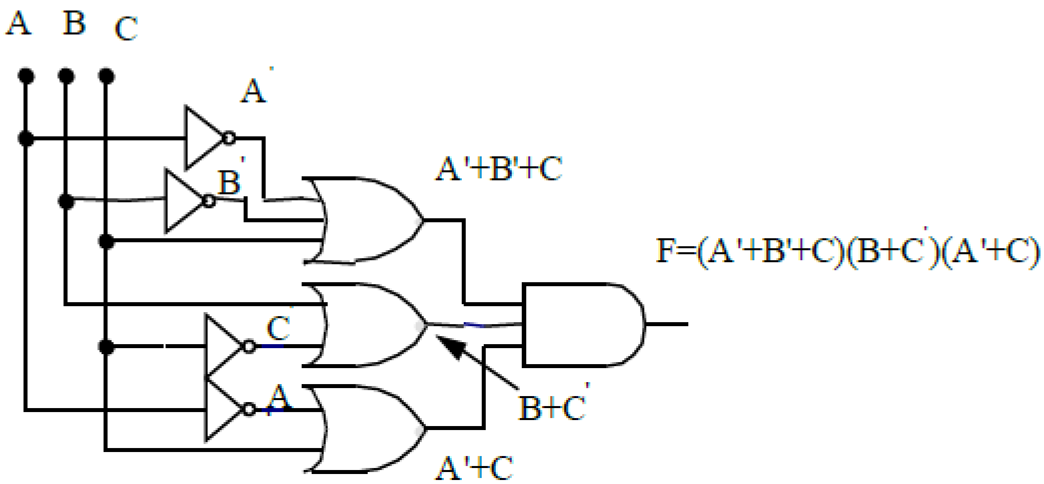
Lojik ifadeleri elemanlarla gerçekleştirmek, ikili işaretleri işlemek için lojik kapılar kullanılır.

Lojik kapılar arasındaki etkileşime lojik bağlantı denilmektedir.

Klasik Lojik Kapı Sembolleri		IEEE/ANSI Lojik Kapı Sembolleri
DEĞİL KAPISI		"1" işareti kapının sadece bir girişinin olduğunu gösterir. Üçgen, lojik olarak terslemeyi göstermek için kullanılır. 
VEYA KAPISI		"≥1"(1'e eşit veya daha büyük) işareti, girişlerden biri veya daha fazlası lojik 1 olduğu zaman çıkışın '1' olacağını göstermek için kullanılır. 
VE KAPISI		"&" (VE) işareti, çıkışın '1' olması için A VE B nin (bütün girişlerin) '1' olması gerektiğini göstermek için kullanılır. 
VEYADEĞİL KAPISI		Çıkışına tersleyici yerleştirilmiş VEYA işareti (VEYADEĞİL) 
VEDEĞİL KAPISI		Çıkışına tersleyici yerleştirilmiş VE işareti (VEDEĞİL) 
ÖZELVEYA KAPISI		"=1" işareti, girişlerden yalnızca birinin lojik 1 olması durumunda çıkışın 1'e eşit olacağını göstermek için kullanılır. 
ÖZELVEYA DEĞİL KAPISI		Çıkışına tersleyici yerleştirilmiş ÖZELVEYADEĞİL Kapısı sembolü. Girişlerden ikisinin aynı olduğu durumda çıkışın Lojik '1' olacağını gösterir. 

<u>Örnek 5.3.</u> $F = (A' + B' + C).(B + C').(A' + C)$ fonksiyonunu gerçekleştirecek lojik devreyi çiziniz.

Bu şekildeki bir fonksiyonda önce parantez icerisindeki ifadeler "DEĞİL'" işleminden Başlanarak gerçekleştirilir. Daha sonra "VEYA" kapıları ile birleştirilen ifadeler "VE" kapısına uygulanarak lojik tasarım bitirilir.

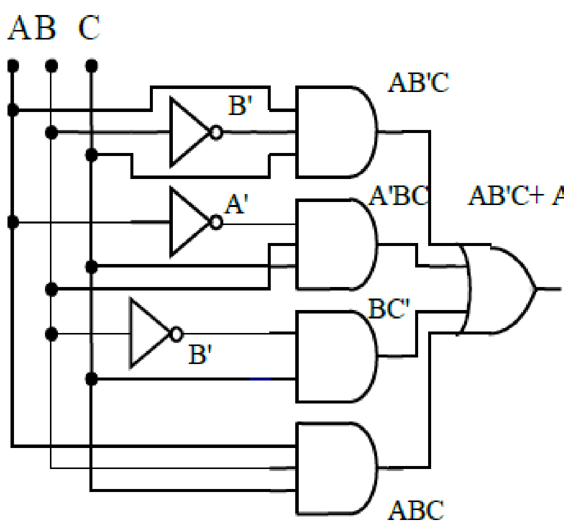


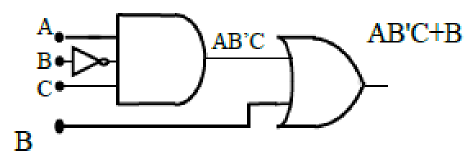
Örnek 5.4. $F = AB'C + A'BC + BC' + ABC$ ifadesini normal ve Boolean kuralları ile sadeleştirildikten sonraki şeklini çiziniz.

$F = A.B.C' + A'.B.C + B.C' + A.B.C = A.B'.C + B.C.\underbrace{(A + A')}_{1} + B.C'$

$= A.B'.C + B.C + B.C' = A.B'.C + B.\underbrace{(C + C')}_{1}{}'$

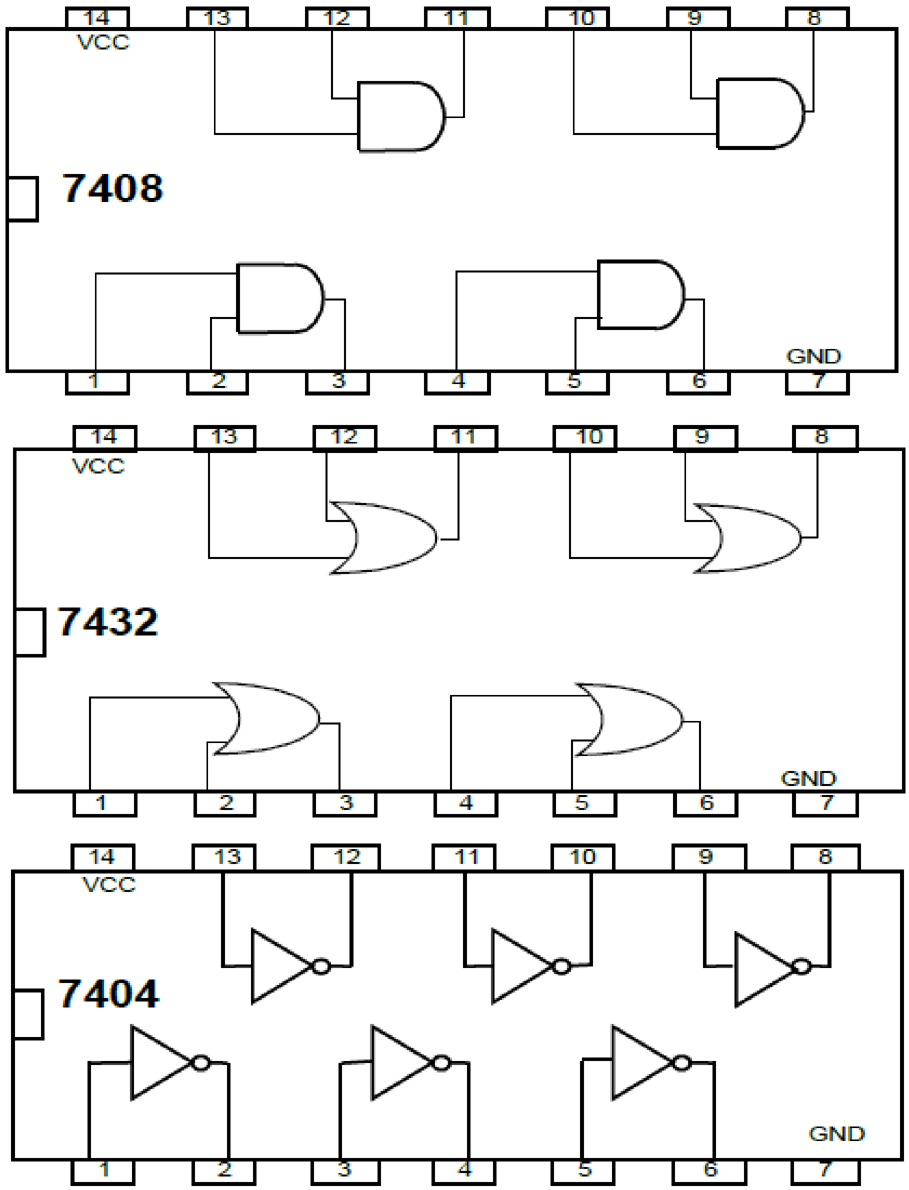
$= A.B'.C + B$



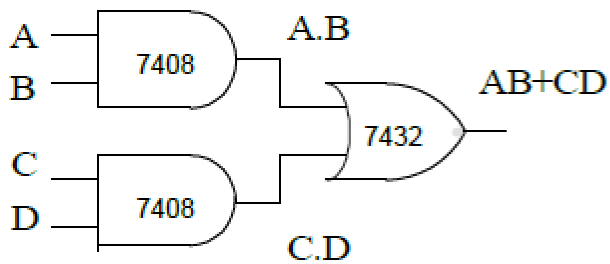


4.9 Lojik Kapı Entegreleri ve Temel Lojik Elemanlarının “VEDEĞİL/VEYADEĞİL” Kapılarıyla Oluşturulması

Dijital sistemler ‘VEYA’, ‘VE’ ve ‘DEĞİL’ lojik kapıların farklı kombinasyonlarından oluşur. Lojik elemanlar devre içerisinde bağlantı kurması ile imal edilir. Aşağıdaki şekilde “VEYA”, “VE” ve “DEĞİL” kapılarının entegre içerisindeki durumları bulunmaktadır.



<u>Örnek 5.5.</u> $Q = A.B + C.D$ fonkisyonunu entegre içerisindeki elemanlarla oluşturunuz.

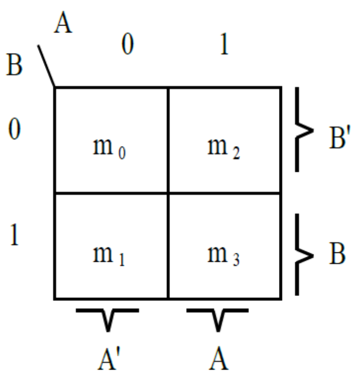

A
B
7408
A.B
7432
AB+CD
C
D
7408
C.D

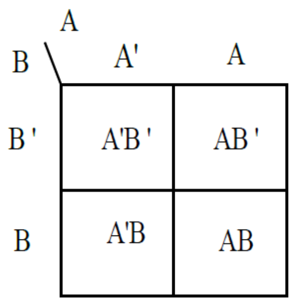
BÖLÜM 5

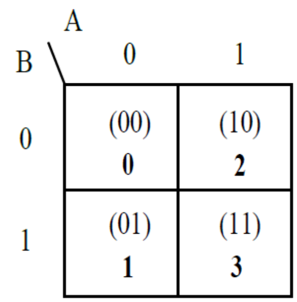
5. KARNAUGH HARİTALARI VE SADELEŞTİRİLMESİ

5.1 İki, Üç, Dört Değişkenli Karnaugh Haritaları

Karnaugh haritasındaki hücre sayısının 2^n formülüyle belirlenir. Buradaki n değişken sayısını ifade eder. İki değişkenli Karnaugh haritası $2^2 = 4$ hücre içerir. İki giriş değişkeni için dört minterm yazılabilir. Dolayısıyla haritada her minterime karşılık gelen bir kare olmak üzere dört kare vardır.

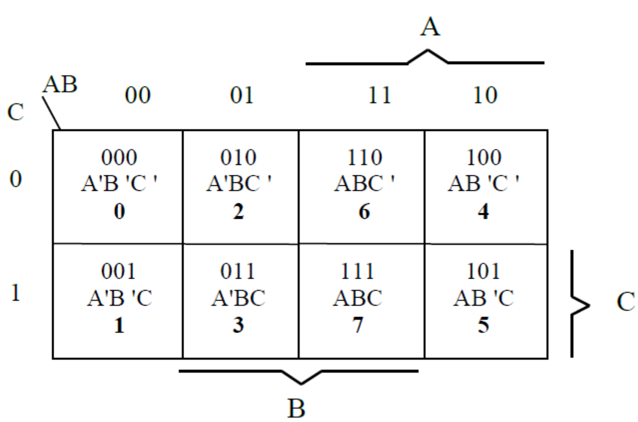


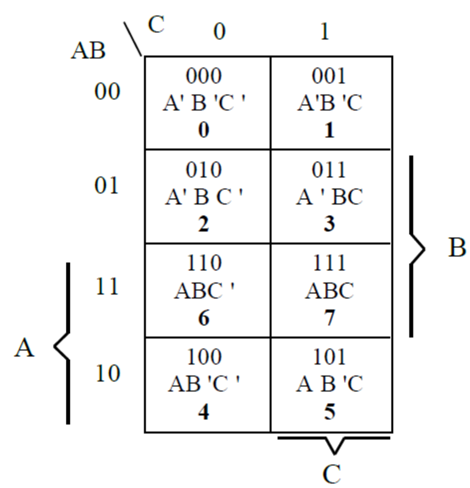




Karnaugh haritasının sol üs köşesine değişkenlerin isimleri A,B,C,...vb. harfler yazılır. Değişkenler en fazla iki değer alabilir. 0 ve 1.

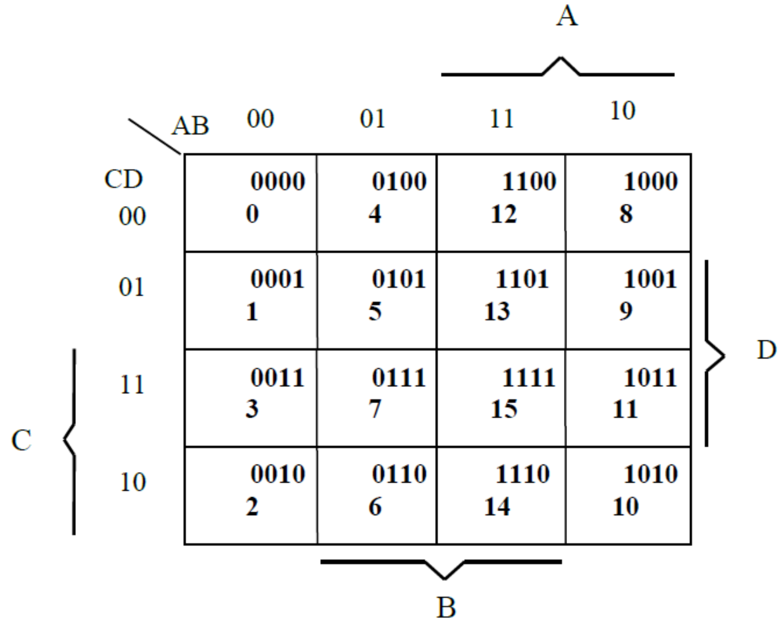
Üç değişkene sahip Karnaugh haritasında $2^3 = 8$ hücre bulunur ve iki farklı yerleştirme durumu ortaya çıkar.





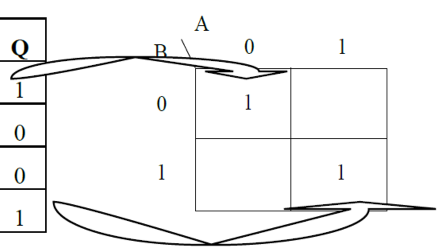
Üç değişkene sahip Karnaugh haritasında $2^4 = 16$ hücre bulunur.

A	B	C	D	Q
0	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	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	



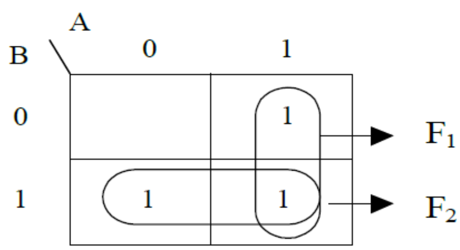
Doğruluk tablosunda mintermleri 1 olan değerler karşılık gelen yerlerine Karnaugh haritasında yerleştirilmelidir.

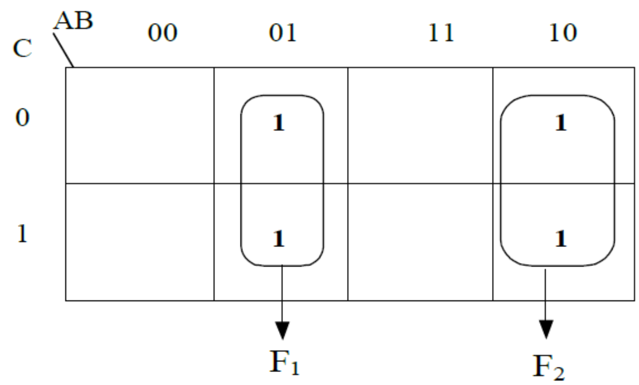
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

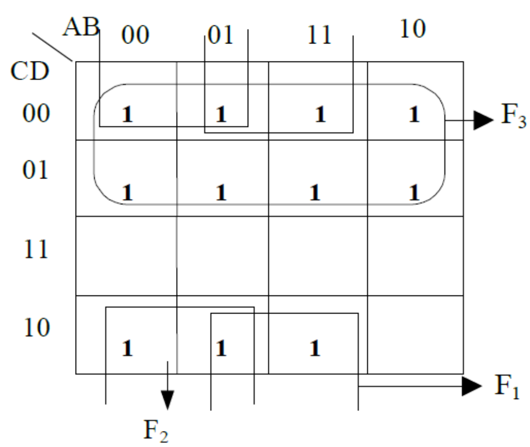


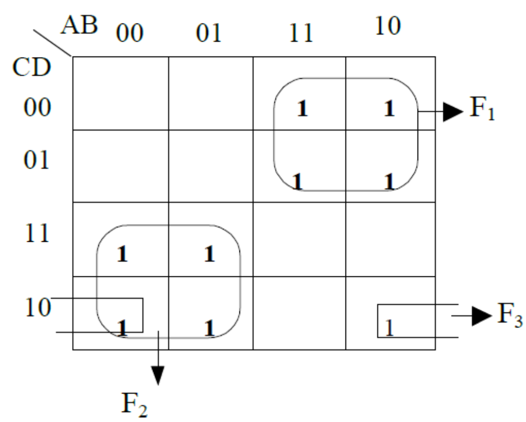
5.2 Karnaugh Haritalarındaki Hücrelerin Gruplandırılması ve Gruplardan Eşitliklerin Yazılması

Doğruluk tablosundaki değerlerin Karnaugh haritalarındaki hücrelere taşınmasından sonra gruplandırma yapılır. Yan yana veya alt alta bulunan hücrelerdeki ‘1’ sayılarının halka içerisine alınmasına denir.









Örnek 5.6. $F = A.B.C + A'.B'.C + B.C$ ifadesini Karnaught haritasına yerleştiriniz.

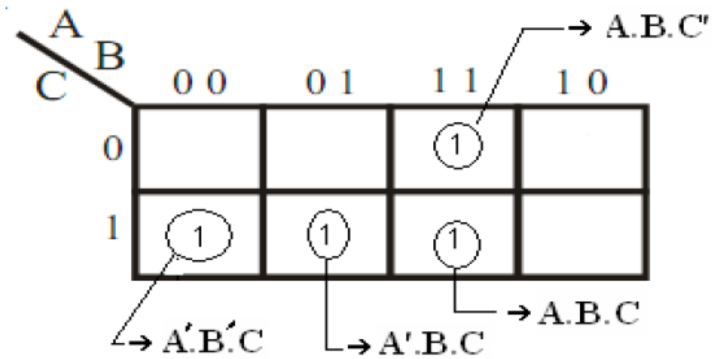
1. Adım: Giriş değişken sayısını bul ve çıkış değişken sayısını hesapla.
2. Adım: Doğruluk tablosunu hazırla.
3. Adım: Mintermleri 1 olanları karşılık gelen Karnaught tablsouna yerleştir.

$2^3 = 8$ adet çıkış değişkeni olacak.

A	B	C	F
0	0	0	0
0	0	1	$1 \rightarrow (A'.B'.C)$
0	1	0	0
0	1	1	$1 \rightarrow (A'.B.C)$
1	0	0	0
1	0	1	0
1	1	0	$1 \rightarrow (A.B.C')$
1	1	1	$1 \rightarrow (A.B.C)$

F fonksiyonunda 4 adet çıkış değeri elde edilmiştir. Bunun sebebi $(A'.B.C)$ ile $(A.B.C)$ çıkışları

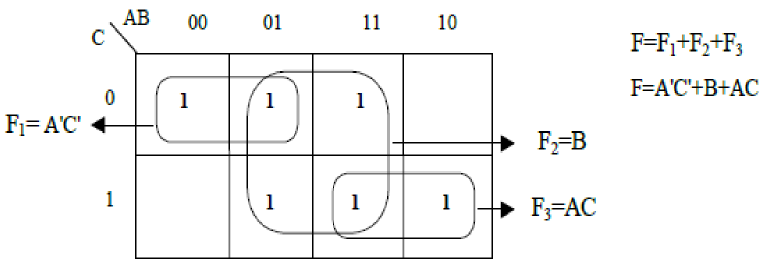
$(B.C)$ ifadesine karşılık gelmesidir. Bundan dolayı haritaya 4 adet bir eklenecektir.



5.3 Karnaugh Haritası Kullanarak Boolean Eşitliklerin Sadeleştirilmesi

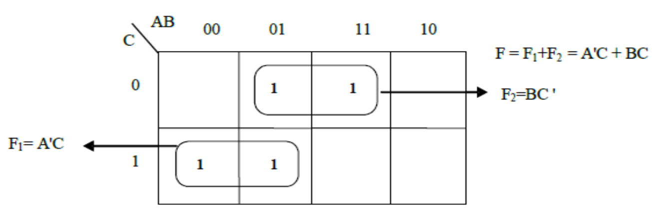
Örnek 5.7. $F = A'.B'.C' + A'.B.C' + A.B.C' + A'.B.C + A.B.C + A.B'.C$ ifadesini Karnaugh haritası ile sadeleştiriniz.

Fonksiyonun mintermlerine 1 yazılır.

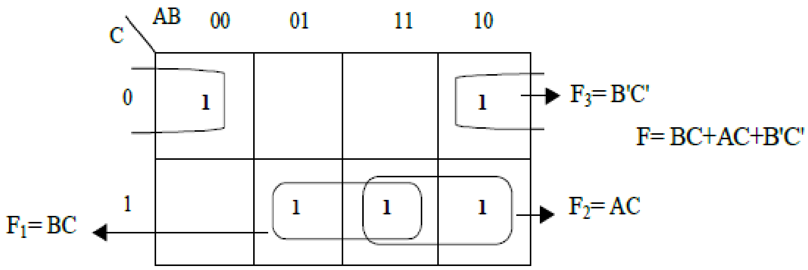


Örnek 5.8. $F = A'.B.C + A'.B'.C + A.B.C' + A'.B.C'$ ifadesini Karnaugh haritası ile sadeleştiriniz.

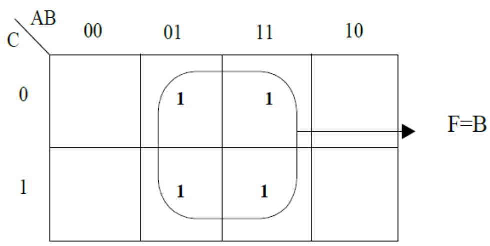
Fonksiyonun mintermlerine 1 yazılır.



Örnek 5.9. $F = A'.B'.C' + A.B'.C' + A'.B.C + A.B'.C + A.B.C$ ifadesini Karnaugh haritası ile birleştirelim.



Örnek 6.0. $F = A.B.C' + A'B.C + A'B.C' + A.B.C$ eşitliğini Karnaugh haritası ile sadeleştiriniz. Sonucunu Boolean aritmetiği ile kontrol ediniz.



Sağlaması:

$$F = A.B.C' + A'.B.C + A'.B.C' + A.B.C$$

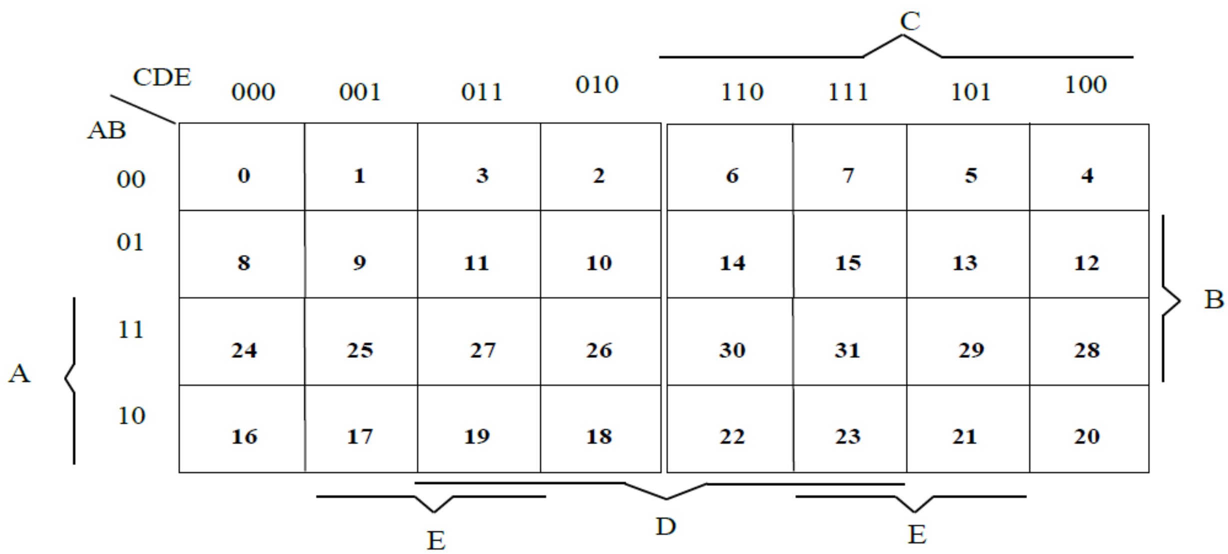
$$= A.B.C' + A'.B.C' + A.B\,\underbrace{(C + C')}_{1}$$

$$= A'.B.\underbrace{(C + C')}_{1} + A.B = A'.B + A.B$$

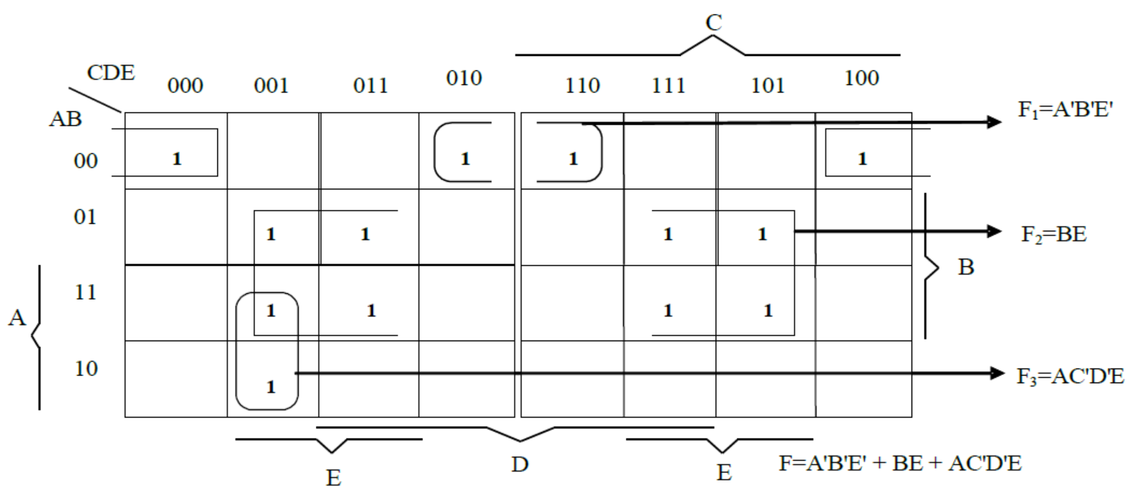
$$= B.\underbrace{(A + A')}_{1} = B \text{ ‘dir.}$$

5.4 Beş ve Altı Değişkenli Karnaugh Haritaları

Karnaugh haritasındaki hücre sayısının 2^n formülüyle belirlenir. Hücrelerin sayısı oluşabilecek mintermlerin sayısına eşit olacaktır. $2^5 = 32, 2^6 = 64$ hücre oluşacaktır.



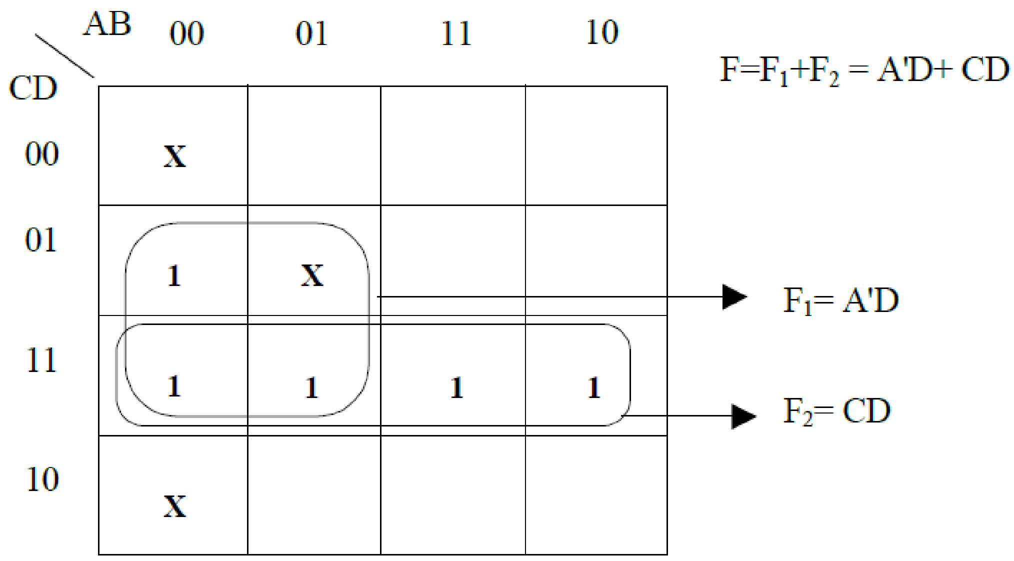
Örnek 6.1. $F_{(A,B,C,D,E)} = \sum(0,2,4,6,9,11,13,15,17,25,27,29,31)$ minterm ifadesini sadeleştirilmiş olarak yazınız.



5.5 Farketmeyen Durumlu Lojik Eşitlikler

Dört bitle ifade edilen onluk sistemde, 9'dan sonraki altı kombinasyon hiçbir zaman oluşmaz. Bu durumda, oluşmayan kombinasyonların aldığı değerler göz ardı edilebilir. 'Fark etmeyen durumlar' olarak isimlendirilen bu durumlar, eşitlikleri basitleştirmeye yardım eder. Fark etmeyen kombinasyonların temsil ettikleri hücrelere 1 veya 0 değerlerini koymak mümkün değildir. Bu nedenle, oluşmayan kombinasyonları temsil eden hücrelere 'X' veya 'd' işareti yerleştirilir.

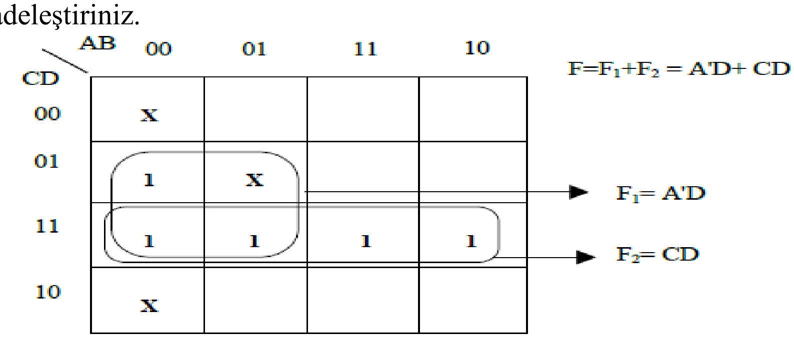
Örnek 6.2. $F = \sum(1,3,7,11,15)$ ve farketmez durumları $d = \sum(0,2,5)$ olan bir fonksiyonu sadeleştiriniz.



5.6 Farketmeyen Durumlu Lojik Eşitlikler

Örneğin; dört bitle ifade edilen onluk sistemde, 9'dan sonraki altı kombinasyon hiçbir zaman oluşmaz. Bu durumda, oluşmayan kombinasyonların aldığı değerler göz ardı edilebilir. 'Fark etmeyen durumlar' olarak isimlendirilen bu durumlar, eşitlikleri basitleştirmeye yardım eder. Fark etmeyen kombinasyonların temsil ettikleri hücrelere '1' veya '0' değerlerini koymak mümkün değildir. Bu nedenle, oluşmayan kombinasyonları temsil eden hücrelere 'X' veya 'd' işareti (1 ve 0 ifadelerinden ayırmak için) yerleştirilir.

Örnek 6.3. $F_{(A,B,C,D,)} = \sum(1,3,7,11,15)$ ve farketmez durumu $d = \sum(0,2,5)$ olan bir fonksiyonu sadeleştiriniz.

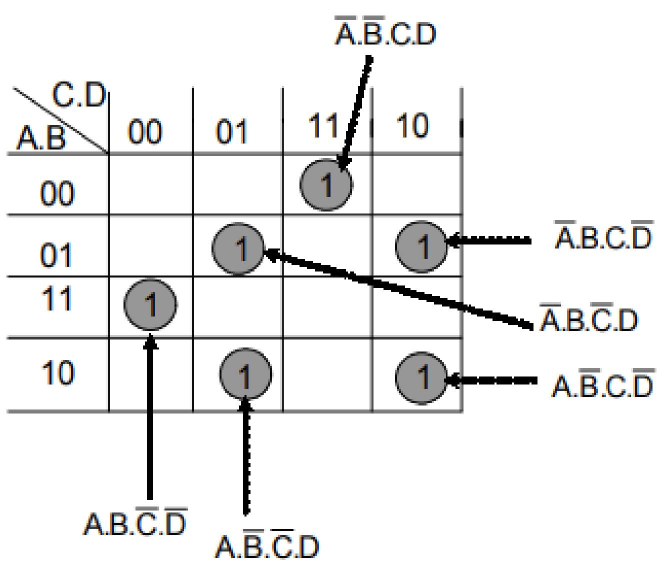


5.7 Karnaugh Haritası Yardımı ile Lojik Devrelerin Tasarımı

Sayısal devre tasarımında önemli nokta tasarımı istenen devrenin çalışmasının anlaşılmasıdır. Devrenin çalışması, yani girişlerin durumuna bağlı olarak çıkışın ne olması gerektiğinin bilinmesidir. Bu durumlara bağlı olarak doğruluk tablosu hazırlanır. Doğruluk tablosundan elde edilen bu değerler Karnaugh haritaları yardımı ile sadeleştirildikten sonra devre çizilerek tasarım tamamlanır.

Meseala; Bir sayısal devrenin çalışması dört anahtarla kontrol edilecektir. Eğer anahtarlardan herhangi ikisi kapalı ise devre çıkışının '1',diğer bütün durumlarda devre çıkışının '0' olması istenmektedir.

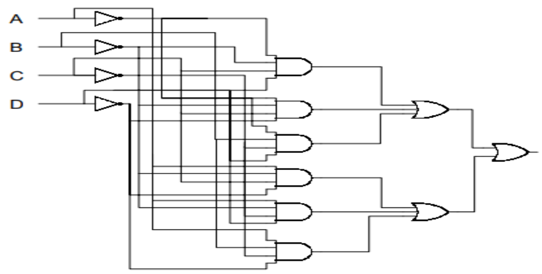
A	B	C	D	Q
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0



Lojik ifade:

$$Q = \bar{A}.\bar{B}.C.D + \bar{A}.B.\bar{C}.D + A.\bar{B}.\bar{C}.D + \bar{A}.B.C.\bar{D} + A+.\bar{B}.C.\bar{D} + A.B.\bar{C}.\bar{D}$$

$$Q = m_3 + m_5 + m_6 + m_9 + \ m_{11} + m_{12}$$


A
B
C
D

BÖLÜM 6

6. TABLO YÖNTEMİ İLE SADELEŞTİRME

7 ve daha fazla değişkene sahip lojik ifadelerin sadeleştirilmesinde kullanılmaktadır.

Örnek 6.4. $F_{(A,B,C)} = A'.B'.C + A'.B.C + A.B'$ ifadesini Boolean Aritmetik işlemleri kullanarak, Karnaugh Harita Yöntemi ve Tablo Yöntemi kullanarak çözünüz.

Boolean Aritmetiği Kullanarak:

$$F_{(A,B,C)} = A'.B'.C + A'.B.C + A.B' = A'.C.\underbrace{(B + B')}_{1} + A.B' = A'.C + A.B'$$

Karnaugh Harita Yöntemi Kullanarak:

Minterm

$A\backslash BC$	00	01	11	10
0	0	1	1	0
1	1	1	0	0

Maxterm

$A\backslash BC$	00	01	11	10
0	0	1	1	0
1	1	1	0	0

$$F_{(A,B,C)} = A'.B'.C + A'.B.C + A.B' = A'.B'.C + A'.B.C + A.B'.C + A.B'.C'$$
$$= m_1 + m_3 + m_5 + m_4) = m_1 + m_3 + m_4 + m_5 = \sum(1,3,4,5) = \prod(0,2,6,7)$$

Tablo Yöntemi Kullanarak:
Bu yöntemde verilen ifadenin minterm ya da maxterm hali elde edilir.

$F_{(A,B,C)} = \sum(1,3,4,5)$
$F_{(A,B,C)} = A'.C + A.B'$

	A	B	C
m_1	0	0	1
m_3	0	1	1
m_4	1	0	0
m_5	1	0	1

$F_{(A,B,C)} = \prod(0,2,6,7)$
$F_{(A,B,C)} = (A' + B').(A + C)$

	A	B	C
m_0	0	0	0
m_2	0	1	0
m_6	1	1	0
m_7	1	1	1

Minterm ve maxtermlerden elde edilen ifadelerin karşılaştırılması:

Minterm

A	B	C	A'	B'	$A.C'$	$A.B'$	$A.C' + A.B'$
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	1

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

Maxterm

A	B	C	A'	B'	$A' + B'$	$A + C$	$(A' + B').(A + C)$
0	0	0	1	1	1	0	0
0	0	1	1	1	1	1	1
0	1	0	1	0	1	0	0
0	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	0	0	1	0
1	1	1	0	0	0	1	0

BÖLÜM 7

7. BİLEŞİK DEVRE TASARIMI

Temel lojik kapılardan oluşan ve devrelerin çıkışları doğrudan girişlerin o anki durumlarına göre belirlenen devreler, bileşik mantık devreleri olarak adlandırılır.

7.1 Birleşik Devre Tasarımı Esasları

Bir devre tasarımı problemlerin ifade edilmesinden başlanıp lojik devrenin çizilebilmesiyle biter. Lojik devre işlem basamakları:

- Problem belirlenir.
- Giriş değişkenlerinin sayısı ve gerekli çıkış değişkenleri tespit edilir.
- Giriş ve çıkış olarak kullanılacak değişkenlere isim verilir.
- Giriş ve çıkış değişkenleri arasındaki ilişkiyi belirleyen doğruluk tablosu oluşturulur.
- Her bir çıkış için uygun Boolean fonksiyonu yazılır.
- Elde edilen Boolean fonksiyonları sadeleştirilir ve lojik devre çizilir.

şeklindedir. Bu işlem basamakları sırasında en çok dikkat edilmesi gereken husus doğruluk tablosuna getirilme aşamasıdır. Burada yapılacak olan en ufak bir hata sonraki adımları da etkileyecektir.

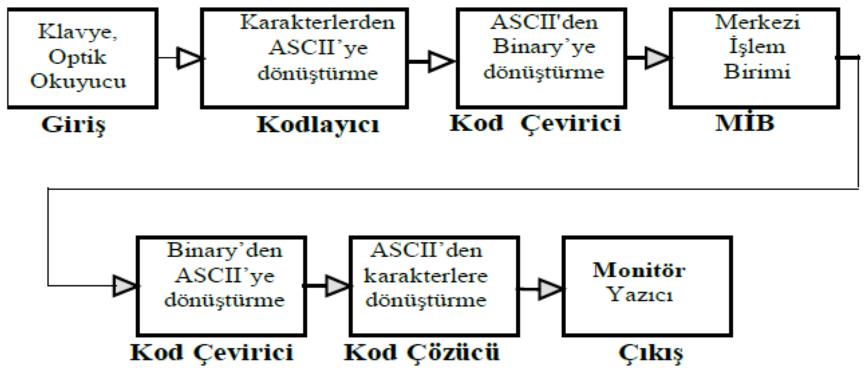
Sadeleştirme işlemi bittikten sonra sadeleştirilen eşitliklerin taşıması gereken özellikler:

- En az sayıda lojik kapı içermesi.
- Herbir kapının en az sayıda girişe sahip olması.
- Devrenin minimum yayılım zamanına sahip olması.
- Devrenin minimum sayıda bağlantı içermesi.
- Herbir kapının, sürme kapasitesi sınırının altında eleman sürmesi.

şeklindedir. Buradaki koşulların hepsi aynı anda sağlanmayabilir. Bu zaman da tasarlanan devreye göre seçim yapılır.

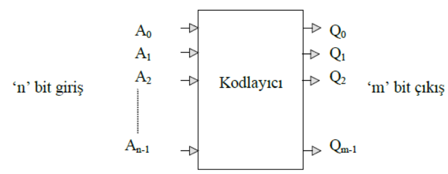
7.2 Kodlama ile İlgili Lojik Devreler

Daha önceki konularda bahsettiğim kodlama işlemi, dijital bilgisayarlarda yoğun olarak kullanılan bir işlemdir.



7.2.1 Kodlayıcı Devreler(Endecoders)

Girişindeki bilgiyi binary sayı sistemine kodlanmış çıkaran devreye denilebilir. Ya da insanlar tarafından anlaşılan rakam-karakterleri farklı bilgilere dönüştüren devreler de denilebilir.



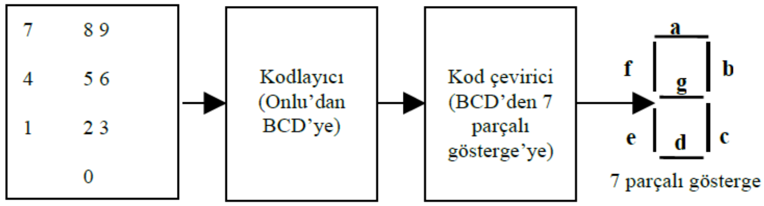
Kodlayıcı devrelerde, herhangi bir anda girişlerden sadece bir tanesi aktif olabilir. Aktif olan girişe göre 'm' bitli cıkış kodu üretilir. 2^ngirişli bir devrede n adet çıkış oluşmaktadır.

7.2.2 Kod Çözücüler(Decoders)

Daha önceki konularda bahsettiğim kodlama işlemi, dijital bilgisayarlarda yoğun olarak kullanılan bir işlemdir. Dijital sistemlerde bilgiler binary sayılar olarak temsil edilir. Bundan dolayı yapılan işlemler de binary üzerinden yapılır. Kod çözücüler . n çıkışlı bir devrede maksimum 2^nadet giriş oluşturmaktadır. Farklı bir deyişle; farklı bilgilere dönüşmüş ifadeleri insanların anlayabileceği rakam-karakterlere dönüştüren devrelere denir.

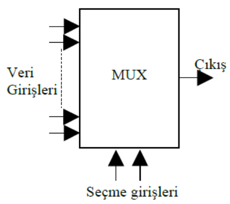
7.2.3 Kod Çeviriciler

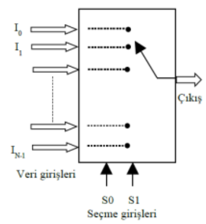
Bir kodlamadaki ifadeyi farklı bir kodlama şekline dönüştüren lojik devrelere denir. Bu devrelere örnek olarak BCD koddan yedi parçalı göstergeye, Binary'den BCD koda vs dönüştürücüler örnek olarak verilebilir.



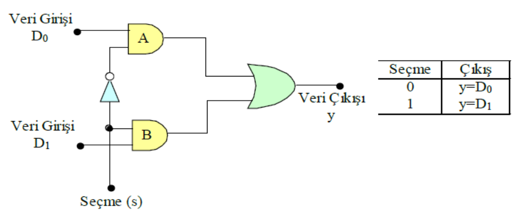
7.3 Çoklayıcılar – Veri Seçiciler(Multiplexer – Data Selector)

Çok sayıdaki giriş bilgisinin sırasıyla çıkışa aktarılması olayına denir. Bir çok girişden gelen bilgiyi uygun çıkışa yönlendiren devrelere denir. Bir çok veri transferi bu devreler ile yapılmaktadır. MUX ya da ÇOĞ olarak da adlanırılabilir.





Veri seçicilerde 2^nsayıdaki giriş hattından uygun olanı seçmek için n tane sayıda seçme hattına ihtiyaç vardır. Çok pozisyonlu anahtar gibi işlem yapan yapan veri seçiciler, girişdeki degree göre çıkış değeri üretecektir.



7.3.1 Çoklayıcı Uygulamaları

Boolean Fonksiyonlarının ve Bileşik Devrelerin Gerçekleştirilmesi:

n değişkenli bir fonksiyonda 2^{n-1}giriş $n-1$ çıkışlı çoklayıcı ile gerçekleştirilebilir.

<u>**Örnek 6.5.**</u> $F_{(A,B,C)} = \sum(1,3,5,6)$ fonksiyonunu 4x1MUX ile gerçeklereceğimizi ve S_0ve S_1 seçme girişleri için A ve B değişkenlerini, çoklayıcı girişleri için C değişkenini kullanacağımızı düşünelim.

C değişkeni çift sayılarda tümlenip tek sayılarda tümlenmez.

$F_{(A,B,C)} = \sum(1,3,5,6)$ fonkisyonunu yukardaki kurallara göre uyarlarsak oluşan tabloda $I_0 = C$, $I_1 = C$, $I_2 = C$, $I_3 = C'$ bağlantıları yapılacaktır.

	I_0	I_1	I_2	I_3
C'	0	2	4	6
C	1	3	5	7

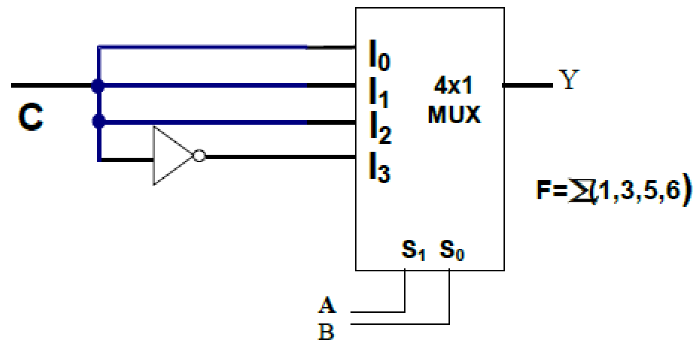
Uygulama tablosu

(a)

	I_0	I_1	I_2	I_3
C'	0	2	4	6
C	1	3	5	7
	C	C	C	C'

Uygulama tablosu

(b)



(c)

Gerekli bağlantılar yapıldıktan sonra sonuç (c)'deki gibi olacaktır.

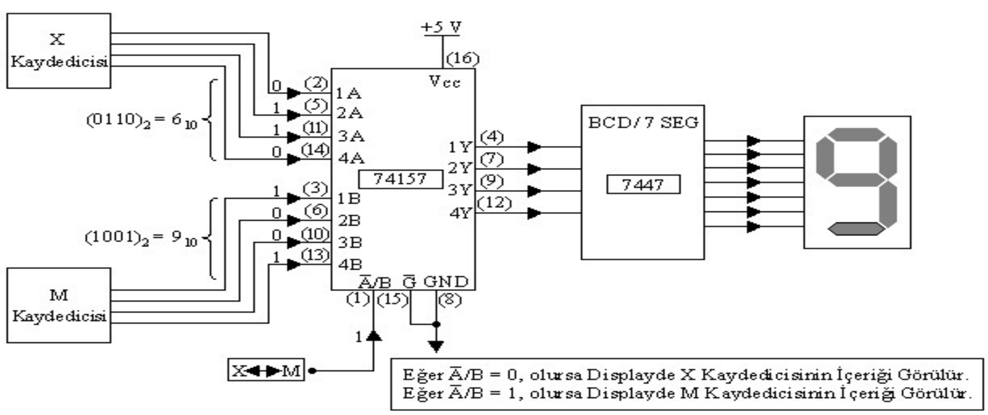
Paralel- Seri Veri Dönüşümü:

Sayısal sistemlerde genelde very iletimi paralel olarak yapılır. Paralel iletim maaliyetlidir ve uzak mesafelerdeki very iletimlerinde seri very iletimi kullanılır. Bundan dolayı paralelden seriye dönüşüme ihtiyaç doğmuştur.

Multiplexer ve sayıcı devreleri kullanılırak bu dönüşüm yapılabilmektedir.

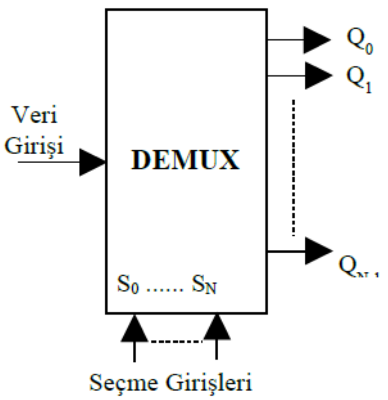
Veri Yönlendirme İşleminin Çoklayıcı ile Gerçekleştirilmesi:

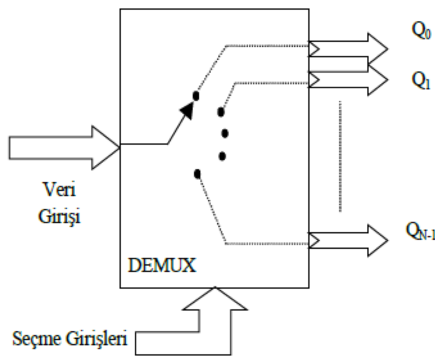
Mutliplexerler birdan fazla bilgiyi tek bir hedefe doğru yönlendirirler. Multiplexer ç ıkışındaki değerler kod çevirici entegrede yedi parçalı göstergede gözükecek şekile dönüştürülür.



7.4 Azlayıcılar – Veri Dağıtıcılar(Demultiplexer)

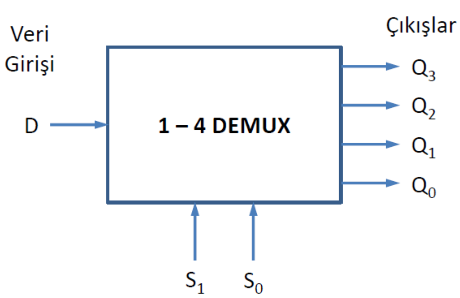
Bir adet girişten aldığı bilgileri farklı çıkışlara dağıtan devrelere denir. Multiplexer'in tersi işlemini yapan bu devreler seçici girişlerin değeri, giriş verilerinin hangi çıkışa yönlendirileceğine karar verir. Yani; Azlayıcılar - Veri Dağıtıcılar tek bir kaynaktan gelen bilgileri seçme girişleri yardımıyla ayırarak, n tane seçme girişi ile 2^n çıkış hattından birisine gönderen çok konumlu bir anahtar gibi düşünülebilir.

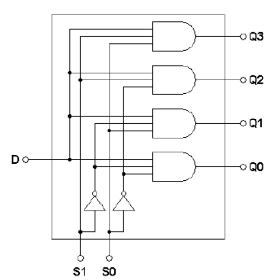




1-4 DEMULTIPLEXER:

- 2 adet seçme girişi bulunur ve bundan dolayı 4 adet çıkış seçilebilir.
- Seçilen çıkış hattına ait "VE" kapısı aktif durumdadır. Veri girişinden gelen lojik bilgi seçilen çıkışa aktarılır.
- Bu durumda diğer "VE" kapıları çıkışı 0'dır.

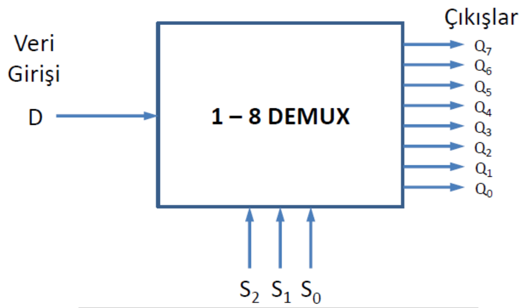


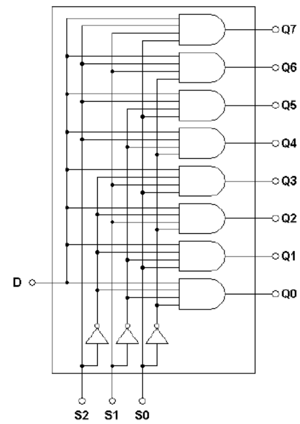


SEÇME GİRİŞLERİ		ÇIKIŞLAR			
S_1	S_0	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

1 – 8 DEMULTIPLEXER:

- 3 adet seçme girişi bulunur ve bundan dolayı 8 adet çıkış seçilebilir.
- Seçilen çıkış hattına ait "VE" kapısı aktif durumdadır. Veri girişinden gelen lojik bilgi seçilen çıkışa aktarılır.
- Bu durumda diğer "VE" kapıları çıkışı 0'dır.





SEÇME GİRİŞLERİ			ÇIKIŞLAR							
S_2	S_1	S_0	Q_7	Q_6	Q_5	Q_4	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	0	0	0	D
0	0	1	0	0	0	0	0	0	D	0
0	1	0	0	0	0	0	0	D	0	0
0	1	1	0	0	0	0	D	0	0	0
1	0	0	0	0	0	D	0	0	0	0
1	0	1	0	0	D	0	0	0	0	0
1	1	0	0	D	0	0	0	0	0	0
1	1	1	D	0	0	0	0	0	0	0

7.4.1 Demultiplexer Uygulamaları

Tek bir verinin farklı yerlerde kullanılmasını sağlayacak uygulamalar yanında, multiplexer ile birlikte sistemleri basitleştirmek amacıyla kullanılır.

Tetikleme (Clock) Demultiplexer:

Demultiplexer devresinin kullanıldığı alanlardan birisi tetikleme demultiplexer’dir. 74LS138 demultiplexer entegresiyle yapılabilen bu uygulamada, tek bir kaynaktan gelen tetikleme sinyali uygun olan çıkışa yönlendirilir.

KAYNAKÇA

Amasya Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Müh. Ders Notları

Prof. Dr. Hüseyin Ekiz, Sayısal Elektronik

MEB 522EE0245, Temel Mantık Devreleri

Mohammed H Ibrahim, Lojik Devreler ve Tasarım,

MEB 522EE0253, ADC ve DAC Devreleri, Elektrik Elektronik Teknolojisi