mantık devreleri

SAYISAL ELEKTRONİK

Prof. Dr. Hüseyin Ekiz



değişim

www.elektrotekno.com

BÖLÜM 1

İÇİNDEKİLER

1-sayı sistemleri
2-kodlama ve kodlar
3-boolean kuralları
4-lojik kapılar,lojik devreler
5-karnaugh haritaları
6-sayısal entereler
7-birleşik mantık devreleri
8-multi vibratörler ve flip-floplar
9-ardışıl devreler
10-sayıcılar
11-kaydediciler
12-bellek devreleri
13-programlanabilir lojik elemanlar

BÖLÜM 2

SAYI SİSTEMLERİ

(NUMBER SYSTEMS)

Giriş

Sayma ve sayı kavramının yeryüzünde ilk olarak nerede ve ne zaman doğduğu bilinmemekle beraber, bazı buluntular Sümer'lerin saymayı bildiklerini ve bugün kullandığımız onluk sayı düzeninin MS 400 dolaylarında, Hindistan'da geliştirildiğini göstermektedir. Onluk sayı düzeni daha sonra İslam bilginleri tarafından geliştirilmiş, MS 800 yıllarında onlu sayı sistemine 'Sıfır (0)' sayısı eklenmiş ve sayı düzenindeki rakam biçimleri değiştirilerek yeni bir şekil kullanılmaya başlanmıştır. Onluk sayı düzeni, Endülüs üzerinden 1200'lü yıllarda Avrupa insanına aktarılmış ve sonuçta bugün bizim ve çoğu Avrupa ülkesinin kullandığı rakam biçimleri ortaya çıkmıştır.

Onluk sayı düzeninin bulunması ve yaygın kullanılmasında büyük olasılıkla insanın iki elinde toplam on parmağın bulunmasının etkisi olmakla beraber, insanlar tarih boyunca onluk sayı düzeninin dışında başka sayma düzenlerinde kullanmışlardır. Örneğin, zaman ölçmede kullandığımız gün, saat, dakika ve saniye gibi birimler birbirinin 12 ve 5 katı biçimindedir.

Onluk sayı düzeni insan kafası için yatkın olmasına rağmen, günümüz bilgisayar teknolojisi için uygun değildir. Bu nedenle günümüz bilgisayar teknolojisinde değişik sayı düzenleri kullanılmaktadır. Bunlar; ikili (binary-dual), sekizli (octal), onaltılı (hexadecimal) sayı sistemleridir. Bu bölümde, bilgisayar teknolojisinde kullanılan sayı sistemlerini genel özellikleri ile inceledikten sonra, incelenen sayı sistemleri arasındaki ilişkileri açıklayacağız.

2.1. Sayı Sistemlerinin İncelenmesi

Sayı sistemlerini incelerken göz önünde bulundurmamız gereken ilk kavram; sayı sistemlerinde kullanılan rakam, işaret, karakter veya harfleri ve bunların temsil ettikleri anlamları açıklamaktır.

Sayı sistemlerinde kullanılan rakamın/harfin/karakterin, sayı içerisinde bulunduğu yere (basamağa) bağlı olarak temsil ettiği anlamı değişir. Anlam değişikliğini belirleyen unsur, bulunan basamağın sayı sistemine bağlı olarak taşıdığı kök/taban değeridir. Bu durumda sayı sistemine bağlı olarak değişen ikinci kavram; sayı sistemlerinde kullanılan taban değeridir.

Bir sayı sistemini ‘S’, sayı sisteminde kullanılan rakam/karakterleri ‘d’ ve kökü de ‘R’ ile gösterir ve genel olarak ‘S’ ile gösterilen sayı sistemini formülle ifade edersek;

$$S= d_nR^n +d_{n-1}R^{n-1}+.....+d_2R^2+d_1R^1+d_0R^0$$

eşitliği elde edilir. Formülde d_n-d_0; sayı değerlerini, R^n- R^0 ise; köke bağlı olarak oluşan basamak değerlerini temsil eder.

Kesirli kısmı bulunan sayıları ifade etmek için;

$$S = d_nR^n +d_{n-1}R^{n-1}+.....+d_2R^2+d_1R^1+d_0R^0 , \ d_1R^{-1}+d_2^{-2}+d_3R^{-3}+….....$$

eşitliği kullanılır.

Genel olarak ifade edilen eşitlikleri bu bölümde inceleyeceğimiz sayı sistemlerine uyarlayarak sırası ile onlu, ikili, sekizli ve onaltılı sayı sistemlerini inceleyelim.

2.1.1. Onlu (Decimal) Sayı Sistemi

Günlük hayatımızda en çok kullandığımız onluk sayı sisteminde on değişik rakam vardır ve bunlar sırasıyla; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9’dur.

Bu durumda d_n- d_0 sayı değerleri; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 sayıları ile ifade edilir ve R; taban değeri olan 10 ile gösterilir. Bu durumda daha önce ifade edilen denklem;

$$D = d_n10^n+d_{n-1}10^{n-1}+.....+d_210^2+d_110^1+d_010^0$$

şeklini alır. Kesirli kısmı bulunan onlu sayıları ifade etmek için;

$$D= d_n10^n+d_{n-1}10^{n-1}+..... +d_210^2+d_110^1+d_010^0, d_1.10^{-1}+d_2.10^{-2}+d_3.10^{-3}+....$$

eşitliği kullanılır.

Denkleme göre en sağdaki basamak en düşük ve en soldaki en yüksek anlamlı basamak olarak; 1985 sayısı,

$$1985 = 1.10^3+9.10^2+8.10^1+5.10^0$$

şeklinde yazılabilir.

2.1.2. İkili (Binary-Dual) Sayı Sistemi

‘0’ ve ‘1’ rakamları ile temsil edilen, taban değeri ‘2’ olan ve iki olasılıklı durumları ifade etmek amacıyla kullanılan sayı sistemi **‘İkili’** veya **‘Binary’** sayı sistemi olarak adlandırılır.

İkili sayı sisteminde her bir basamak **‘BİT’** olarak (**Bi**nary Digi**T**) adlandırılır. En sağdaki basamağa en Düşük Anlamlı Bit - DAB (Least Significant Bit-LSB), en soldaki basamağa en Yüksek Anlamlı Bit-YAB (Most Significant Bit-MSB) denir. Buna göre ikili sayı sistemindeki basamak değerleri;

$$B = d_n2^n +d_{n-1}2^{n-1}+.... +d_22^2 +d_12^1+d_02^0$$

eşitliği ile ifade edilebilir.

Örnek olarak ‘101101101’ ikili sayısının basamak değerlerini yazarsak;

$$B = 1.2^8 + 0.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0$$

eşitliği bulunur. Bu onluk sistemde;

$$D = 256+64+32+8+4+1 = 365$$

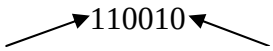
sayısına eşittir.

Aynı şekilde kesirli kısım bulunan ikili sayıların basamak değerleri:

$$B = d_n2^n + d_{n-1}2^{n-1} + ... + d_22^2 + d_12^1 + d_02^0 \ , \ d_12^{-1} + d_22^{-2} + ... + d_n2^{-n}$$

Tam sayı kısmı **Kesirli sayı kısmı**

şeklinde olur.



En anlamlı Bit (MSB) En anlamsız Bit (LSB)

İkili sayı sistemi bilgisayarlar için uygun ve bu sistemde sayıların ifade edilmesi kolay olmasına rağmen, sayıların ifade edilmesi daha çok sayıda basamak ile mümkün olmaktadır. Onlu olarak ifade edilen bir sayıyı, ikili sistemde ifade etmek için ortalama üç katı daha fazla basamağa ihtiyaç vardır. Buda ikili sayı sisteminde yapılacak işlemlerin zaman alması, zorlaşması ve hata ihtimalinin yükselmesi sonucunu doğurur. Bu sakıncaları ortadan kaldırmak için, ikili sayı sisteminin tam katları olan ve işlemlerin daha az zamanda yapılmasına imkan tanıyan (ikili sayı sistemine dönüştürülmeleri veya ters işlemi çok kolay olan) sekizli ve onaltılı sayı sistemleri kullanılır. Bununla beraber, ikili sayı sistemi bilgisayarlarda aşağıdaki amaçlar için kullanılmaktadır:

i. Gerçek sayısal değeri ifade etmek için,

ii. Veri ile ilgili bellekteki adresi belirtmek için,

iii. Komut kodu olarak,

iv. Alfabetik ve sayısal olmayan karakterleri temsil etmek için bir kod olarak,

v. Bilgisayarda dahili ve harici olarak bulunan devrelerin durumlarını belirlemesi için bir sayı grubu olarak.

2.1.3. Sekizli (Octal) Sayı Sistemi

İkili sayı sistemindeki sayıların daha kolay gösterilmesini sağlayan sayı sistemlerinden birisi, sekizli (octal) sayı sistemidir.

Sekizli sayı sisteminde taban ‘8’ ve kullanılan sayılar; 0, 1, 2, 3, 4, 5, 6, 7’dir. Genelde yetmişli yıllarda mini bilgisayarlarda çokça kullanılan sekizli sayı sistemindeki basamak değerlikleri;

$O = d_n8^n+d_{n-1}8^{n-1}+.....+d_38^3+d_28^2+d_18^1+d_08^0 , d_18^{-1} + d_28^{-2} +……$

formülü ile ifade edilir.

Onlu	İkili	Sekizli	Onaltılı
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Tablo 2.1. 0-20 arası sayıların ikili, sekizli ve onaltılı sistemlerdeki karşılıkları.

2.1.4. Onaltılık (Hexadecimal) Sayı Sistemi

İkili sayı sisteminin daha kolay gösterilmesini sağlayan ve günümüz bilgisayarlarında yaygın olarak kullanılan sayı sistemi onaltılık (hexadecimal) sayı sistemidir. Onaltılı sayı sisteminde 0 ile 9 arasındaki rakamlar ile A, B, C, D, E, F harfleri kullanılır.

Bu sayı sistemindeki sayıların genel denklemi;

$$H = d_n16^n+d_{n-1}16^{n-1}+.....+d_116^1+d_016^0 + d_116^{-1} + d_216^{-2} + d_216^{-3} +……$$

şeklinde oluşur.

Tablo 2.1'de 0-20 arasındaki onlu sayıların ikili, sekizli, onaltılı sayı sistemlerindeki karşılıkları gösterilmektedir.

Buraya kadar sayı sistemlerini açıklandı. Şimdi bu sayı sistemlerinin birbirlerine dönüşümlerini açıklayalım.

2.2. Sayı Sistemlerinin Birbirlerine Dönüştürülmeleri

Sayı sistemlerinin birlikte kullanılması, sayı sistemlerinden herhangi birisi ile ifade edilen bir büyüklüğün diğer sayı sistemlerine dönüşüm ihtiyacını ortaya çıkarır. Sayı sistemlerini tek-tek ele alarak diğer sayı sistemlerine dönüşüm prensiplerini ve yöntemlerini açıklayalım.

2.2.1. Onlu Sayıların İkili, Sekizli ve Onaltılı Sayılara Dönüşümü

Onlu bir sayı başka bir sayıya dönüştürülecekse; onlu sayı, yeni oluşacak olan sayı sisteminin taban değerine sürekli bölünür. Bölüm sonucunda elde kalanların tersten sıralanmasıyla yeni sayı sistemindeki sayı bulunur.

Onlu sayıların İkili Sayılara Dönüşümü:

Onlu bir sayı ikili bir sayıya dönüştürülecekse, onlu sayı sürekli 2'ye bölünür.

Örnek 1: $(39)_{10}$ sayısını ikili sayı sistemine çevirelim.

Bölünen	**Bölüm**		**Kalan**	
39/2	19	+	1	LSB
19/2	9	+	1	
9/2	4	+	1	
4/2	2	+	0	yazım yönü
2/2	1	+	0	MSB
			100111	

MSB: En büyük değerlikli sayı. (Most Significant Bit)

LSB: En küçük değerlikli sayı. (Least Significant Bit)

Sonuç olarak;

$$(39)_{10}=(100111)_2$$

esitliği bulunur.

Örnek 2: $(1271)_{10}$ sayısını ikili sayıya dönüştürelim.

İşlem		Bölüm	Kalan
1271 / 2	=	635	1
635 / 2	=	317	1
317 / 2	=	158	1
158 / 2	=	79	0
79 / 2	=	39	1
39 / 2	=	19	1
19 / 2	=	9	1
9 / 2	=	4	1
4 / 2	=	2	0
2 / 2	=	1	0
1	→		1

Sonuç olarak;

$$(1271)10 = (10011110111)_2$$

eşitliği bulunur.

Kesirli onlu sayılar ikili sayılara dönüştürülürken kesir kısmı 2 ile çarpılır. Çarpım sonucunda elde edilen sayının tam kısmı kaydedilerek, kesirli kısım 2 ile yeniden çarpılır. Bu işleme kesirli kısım '0' değerine (veya 0'a çok yakın bir değere) ulaşıncaya kadar devam edilir.

Örnek 3: $(0.65)_{10}$ sayısını ikili sayı sistemine çevirelim.

	Tam Kısım		
0.65 * 2 = 1.30	1	a^{-1}	
0.30 * 2 = 0.60	0	a^{-2}	Sıralama
0.60 * 2 = 1.20	1	a^{-3}	yönü
0.20			

Sonuç;

$$(0.65)_{10} \cong (0.101)_2$$

olarak bulunur. Bu örnekte görüldüğü gibi kesirli kısım 0 değerine varmayabilir. Bu gibi durumlarda işlem sonlandırılarak yuvarlatma yapılabilir.

Örnek 4: $(41.6875)_{10}$ sayısını ikili sayıya çevirelim.

Tam sayı ve kesirli kısmı bulunan bir sayıyı ikili sayıya çevirmek için, tam sayı ve kesir kısımları ayrı-ayrı dönüştürülür ve bulunan sayılar birleştirilir.

Önce tam sayı kısmını çevirelim:

İşlem	**Bölüm**	**Kalan**
41 / 2	20	1
20 / 2	10	0
10 / 2	5	1
5 / 2	2	0
2 / 2	1	0
1	1	

$(41)_{10} = (100101)_2$

Daha sonra kesirli sayı kısmının çevirimini yapalım;

	Tamsayı
0.6875 * 2 = 1.3750	1
0.3750 * 2 = 0.7500	0
0.7500 * 2 = 1.5000	1
0.5000 * 2 = 1.0000	1

$(0.6875)_{10} = (1011)_2$

Sonuçta, iki sayıyı birleştirirsek;

$$(41.6875)_{10} = (100101.1011)_2$$

eşitliği bulunur.

Onlu Sayıların Sekizli Sayılara Dönüştürülmesi :

Onlu sayı sistemindeki bir sayıyı, sekizli sisteme dönüştürmek içinde yukarıda açıklanan yöntemler kullanılır.

Örnek 5: $(153)_{10}$ sayısını sekizli sisteme çevirelim.

Verilen sayının devamlı 8 ile bölünmesi şeklinde işlem yapılır:

İşlem	**Bölüm**	**Kalan**
153 / 8	19	1
19 / 8	2	3
2		2

İşlemler sonucunda,

$$(153)_{10} = (231)_8$$

eşitliği bulunur.

Örnek 6: $(0.513)_{10}$ sayısını sekizli sayı sistemine çevirelim.

Verilen sayı devamlı 8 ile çarpılarak oluşan tam sayılar yazılır.

	Oluşan tam sayı	
0.513 x 8 = 4.104	4	↓ yazım yönü
0.104 x 8 = 0.832	0	
0.832 x 8 = 6.656	6	
0.656 x 8 = 5.248	5	
0.248 x 8 = 1.984	1	

Sonuç olarak;

$$(0.513)_{10} \cong (0.40651)_8$$

eşitliği bulunur.

Tam sayı ve kesirli kısmı bulunan onlu sayıları 8'li sayılara dönüştürme işleminde; tam sayı ve kesir kısımları ayrı ayrı dönüştürülür ve bulunan sonuçlar birlikte yazılır.

Örnek 5 ve Örnek 6'daki işlemlerden, $(153.513)_{10}$ sayısının $(231.40651)_8$ sayısına eşit olduğu söylenebilir.

Onlu Sistemdeki Sayıların Onaltılı Sayılara Dönüştürülmesi:

Onlu sistemdeki bir sayıyı onaltılık sisteme dönüştürmek için, onluk sistemin ikili ve sekizli sisteme çevrilmesindeki yöntem uygulanır. Ancak onaltılık sistemde taban '16' olduğundan, 16'ya bölme ve kalanı yazma şeklinde işlem yapılır.

Örnek 7: $(214)_{10}$ sayısını onaltılık sayı sistemine çevirelim.

Verilen sayının devamlı 16'ya bölünmesi ve kalanının yazılması şeklinde işlem yapılır:

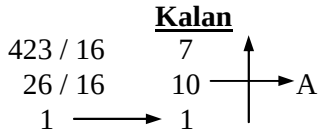
İşlem	**Bölüm**	**Kalan**	
214 / 16	13	6	→ 6 ↑
13 / 16	0	13	→ D ↑

Sonuç olarak;

$$(214)_{10} = (D6)_{16}$$

eşitliği yazılabilir.

Örnek 8: $(423)_{10} = (?)_{16}$ dönüşümünü gerçekleştirelim.



Bölme işlemi sonucunda elde edilen '10' sayısının onaltılı sistemdeki karşılığı olan 'A' değerinin yazılaması ile;

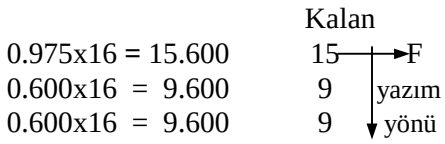
$$(423)_{10} = (1A7)_{16}$$

eşitliği elde edilir.

Kesirli ondalık sayıların onaltılı sayı sistemine dönüştürülmesi; kesirli sayının 16 ile çarpımından oluşan tam sayı kısmının alınıp, yeni sayının kesirli kısmının çarpılmaya devam etmesi şeklinde yapılır.

Örnek 9: $(0.975)_{10}$ sayısını onaltılık sisteme çevirelim.

Verilen sayı devamlı 16 ile çarpılıp, oluşan tam sayılar yazılır:



Sonuç olarak;

$$(0.975)_{10} = (0.F99)_{16}$$

eşitliği bulunur.

Örnek 10: $(214.375)_{10} = (?)_{16}$ dönüşümünü yapalım.

Tam sayı ve kesirli kısımların dönüşümü ayrı ayrı yapılacağından, tam sayı kısmını Örnek 7'den alabiliriz:

$$(214)_{10} = (D6)_{16}$$

Kesirli kısım ise;

$$(0.375)_{10} = (?)_{16}$$
$$0.375 \times 16 = 6.0$$

olarak elde edilir.

Bu durumda kesirli kısım için;

$$(0.375)_{10} = (0.6)_{16}$$

eşitliği yazılabilir.

Sonuç olarak;

$$(214.375)10 = (D6.6)_{16}$$

eşitliği bulunur.

2.2.2. İkili Sayı Sistemindeki Sayıların Onlu, Sekizli ve Onaltılı Sayı Sistemlerine Dönüştürülmesi

İkili sistemdeki bir sayı, her basamağının ağırlık katsayısı ile çarpılıp, bulunan değerlerin toplanması ile ilgili sayı sistemine dönüştürülür.

İkili Sayıların Onlu Sayılara Dönüştürülmesi:

İkili sistemdeki bir sayı, her basamağının ağırlık katsayısı ile çarpılıp, bulunan değerlerin toplanması ile Onlu sayı sistemine dönüştürülür

Örnek 11: $(11001)_2$ sayısının onluk sayı sistemindeki karşılığını bulalım.

Her bir basamakta bulunan sayı basamak değeri ile çarpılır ve bulunan sayılar toplanırsa;

$1\ 1\ 0\ 0\ 1 \longrightarrow 1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 = 16 + 8 + 0 + 0 + 1$

olur. Bu durumda;

$$(11001)_2 = (25)_{10} = 25$$

eşitliği yazılabilir.

Kesirli ikili sayının onluk sayı sistemine dönüştürülmesi; kesirli kısmın soldan sağa doğru ikinin negatif kuvvetleri şeklinde yazılıp, bu sayıların basamaklarda bulunan sayılarla çarpılması ve bulunan çarpımların toplanması şeklinde gerçekleştirilir.

Örnek 12: $(100.01)_2$ sayısını onluk sayı sistemine dönüştürelim.

Tamsayı ve kesirli kısmın basamak değerleri ile basamaklarda bulunan sayılar çarpılırsa;

$$\begin{aligned} 100.01 &= 1.2^2 + 0.2^1 + 0.2^2 \text{ , } 0.2^{-1} + 1.2^{-2} \\ &= 1.4 + 0.2 + 0.1 \text{ , } 0.1/2 + 1.1/4 \\ &= 4 + 0 + 0 \text{ , } 0 + 1/4 \\ &= (4.25)_{10} \end{aligned}$$

sayısı bulunur. Bu durumda;

$$(100.01)_2 = (4.25)_{10}$$

eşitliği elde edilir.

Örnek 13: $(1011.101)_2 = (?)_{10}$ dönüşümünü yapalım.

Sayının tam ve kesirli kısmında bulunan rakamlar ile basamak değerleri çarpılır.

$$(1011.101)_2 = 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 , 1.2^{-1} + 0.2^{-2} + 1.2^{-3}$$
$$= 8 + 0 + 2 + 1 , 0.5 + 0 + 0.125$$
$$= (11.625)_{10}$$

Dönüştürme işlemi sonucunda;

$$(1011.101)_2 = (11.625)_{10}$$

eşitliği bulunur.

İkili sayı sistemindeki sayıların sekizli ve onaltılı sayılara dönüştürülmeleri bilgisayarlarda önemli bir yere sahiptir. $2^3 = 8$ ve $2^4 = 16$ olduğundan, her bir sekizlik sayı üç bit ikili sayıya karşılık gelirken, herbir onaltılık sayı 4 bit ikili sayıya karşılık gelir.

İkili Sayıların Sekizli Sayılara Dönüştürülmesi:

İkili sistemdeki bir sayıyı sekizli sistemde ifade etmek için, ikili sistemdeki sayılar sağdan sola doğru üçerli kümeler halinde ayrılır ve en sondaki kümedeki bitlerin sayısı üçten az ise sola doğru ‘0’ eklenerek üçe tamamlanır.

Örnek 14: $(11001111011101)_2$ sayısını sekizli sayı sistemine dönüştürelim.

Üçerli kümelere ayırma ve eksik bitleri tamamlama sonucunda,

011 001 111 011 101

kümeleri elde edilir. Her kümedeki sayının onluk karşılığı yazılırsa;

$$(011\ 001\ 111\ 011\ 101)_2 = (3\ 1\ 7\ 3\ 5)_8$$

şeklinde sekizli sistemdeki sayı bulunur.

Bu durumda,

$$(11001111011101)_2 = (31735)_8$$

eşitliği yazılabilir.

Örnek 15 : $(10110001101011)_2 = (?)_8$ dönüşümünü yapalım.

Verilen sayı üçerli gruplara ayrılır ve herbir grubun temsil ettiği sekizli sayı yazılırsa;

$$010\ 110\ 001\ 101\ 011 = (26153)_8$$
$$2\quad 6\quad 1\quad 5\quad 3$$

Bu durumda

$$(10110001101011)_2 = (26153)_8$$

eşitliği elde edilir.

Kesirli ikili sayıların sekizli sayılara dönüşümü aynı yöntemle gerçekleştirilir. Yalnızca, kesirli kısımdaki gruplandırma soldan sağa doğru yapılır.

Örnek 16: $(1101101101.111100000110)_2 = (?)_8$ dönüşümünü yapalım.

Sayı, $(001\ \ 101\ \ 101\ \ 101.111\ \ 100\ \ 000\ \ 110)_2$ şeklinde gruplandırılıp, her grubun karşılığı olan ikili sayı yazılırsa;

$$1\quad 5\quad 5\quad 5\ .\ 7\quad 4\quad 0\quad 6 = (1555.7406)_8$$

sonucu elde edilir. Sonuçta;

$$(1101101101.111100000110)_2 = (1555.7406)_8$$

eşitliği bulunur.

İkili Sayıların Onaltılı Sayılara Dönüştürülmesi:

İkili sistemdeki bir sayı, her basamağının ağırlık katsayısı ile çarpılıp, bulunan değerlerin toplanması ile Onaltılı sayı sistemine dönüştürülür.

İkili sayı sisteminden onaltılık sayı sistemine dönüştürme işlemi, ikili sistemdeki sayının dörderli gruplara ayrılıp, her bir gruptaki sayıların karşılıklarının yazılması şeklinde gerçekleştirilir. Gruplama işlemine sağdan başlanır ve en sondaki grup '0' eklenerek dört bite tamamlanır. Gruplardaki sayıların karşılıkları olan sayılar yazılınca, onaltılık sistemdeki sayı elde edilir.

Örnek 17: $(10111101110000111101)_2$ sayısını onaltılık sayı sistemine dönüştürelim.

Verilen sayı dört bitlik gruplar halinde yazılırsa; 1011 1101 1100 0011 1101 şeklini alır. Bu gruplardaki sayıların onaltılık sistemdeki karşılıkları yazılırsa;

1011 1101 1100 0011 1101

B D C 3 D

sayıları elde edilir. Sonuç olarak;

$$(10111101110000111101)_2 = (BDC3D)_{16}$$

eşitliği bulunur.

Örnek 18: $(10110001101011.11110010)_2$ sayısını onaltılık sayı sistemine çevirelim.

Çevirme işlemi için önce sayının tam sayı ve kesirli kısımları 4'erli gruplara ayrılır. Herbir grubun onaltılı sistemde karşılığı olan sayı yazılır.

0010 1100 0110 1011 . 1111 0010

2 C 6 B F 2

Grupların karşılıkları olan sayılar sırası ile yazılınca; onaltılık sistemdeki sayı;

$(10110001101011.11110010)_2 = (2C6B.F2)_{16}$ olarak elde edilir.

Örnek 19: $(1100000110.1101100)_2 = (?)_{16}$ dönüşümünü yapalım.

Gruplandırma yapılıp, herbir gruptaki sayıların karşılığı yazılırsa;

0011 0000 0110 . 1101 100 = $(306.D8)_{16}$

3 0 6 D 8

sonucu elde edilir.

2.2.3. Sekizli Sistemdeki Sayıların İkili, Onlu Ve Onaltılı Sistemlere Dönüştürülmesi

Sekizli sistemdeki sayıları diğer sayı sistemlerine dönüştürmek için dönüştürülecek sayı sisteminin özelliğine uygun yöntem kullanılır.

Sekizli Sayıların İkili Sayılara Dönüştürülmesi:

Sekizli sistemdeki bir sayıyı ikili sayı sistemine dönüştürmek için, her bir basamaktaki sayının karşılığı olan ikili sayı 3 bitlik gruplar şeklinde yazılır. Gruplar halinde yazılan sayılar bir araya getirilmesi ile ikili sistemdeki sayı ortaya çıkar.

Örnek 20: $(673.124)_8$ sayısını ikili sayı sistemine çevirelim.

Önce her bir sayının karşılığı olan ikili sayı 3 bit olarak yazılır:

6=110, 7=111, 3=011, 1=001, 2=010, 4=100.

Yazılan sayılar bir araya getirilirse;

$$(673.124)_8 = (110111011.001010100)_2$$

eşitliği bulunur.

Sekizli Sayıların Onlu Sayılara Dönüştürülmesi:

Sekizli sayılar, her bir basamaktaki rakamın basamak ağırlığıyla çarpılması ve daha sonra çarpımların toplanması yoluyla onluk sayı sistemine dönüştürülür.

Örnek 21: $(372)_8$ sayısını onluk sayı sistemine çevirelim.

Herbir basamaktaki sayı basamak değerleriyle çarpılıp, bulunan sayılar toplanırsa;

$$(372)_8 = 3x8^2 + 7x8^1 + 2x8^0$$

$$= 3x64 + 7x8 + 2x1$$

$$= 250$$

sayısı bulunur.

Bu durumda;

$$(372)_8 = (250)_{10}$$

eşitliği elde edilir.

Örnek 22: $(24.6)_8 = (?)_{10}$ dönüşümünü gerçekleştirelim.

Basamaklardaki sayılar basamak değerleriyle çarpılır:

$$(24.6)_8 = 2x8^1 + 4x8^0 . 6x8^{-1}.$$

Çarpımından bulunan değerler toplanırsa;

$$= 16 + 4.75=20.75$$

sayısı bulunur. Sonuçta;

$$(24.6)_8 = (20.75)_{10}$$

eşitliği oluşur.

Sekizli sistemdeki bir sayıyı onaltılık sayı sistemine dönüştürmenin en pratik yolu, sekizlik sayıyı önce ikilik sayı sistemine dönüştürmek ve daha sonra ikili sayıyı onaltılık sayıya çevirmektir.

Örnek 23: $(5431)_8$ sayısını onaltılık sayıya dönüştürelim.

Sekizlik sayı önce ikili sayıya çevrilir.

$$(5431)_8 = (101100011001)_2$$

Daha sonra bulunan sayı dörderli gruplara ayrılıp, her bir grubun karşılığı olan onaltılı sistemdeki ifade yazılırsa;

$$1011 = B, \qquad 0001 = 1, \qquad 1001 = 9$$

eşitlikleri bulunur.

Bulunan sayılar bir araya getirilirse;

$$(B19)_{16}$$

sayısı elde edilir. Bu durumda;

$$(5431)_8 = (D19)_{16}$$

eşitliği yazılabilir.

2.2.4. Onaltılık Sistemdeki Sayıların, İkili, Sekizli ve Onlu Sayı Sistemlerine Dönüştürülmesi

Onluk sayı sistemlerinde ifade edilen bir büyüklüğü diğer sayı sistemlerine dönüştürmek için uygun yöntemler kullanılır.

Onaltılı Sayıların İkili Sayılara Dönüştürülmesi:

Onaltılı sistemdeki bir sayıyı ikili sayı sistemine dönüştürmek için; her basamaktaki sayının karşılığı olan ikili sayı 4 bit şeklinde yazılır. 4 bitlik gruplar bir araya getirilerek ikili sayı bulunur.

Örnek 24: $(5D1D69)_{16}$ sayısını ikili sisteme çevirelim.

Herbir basamaktaki onaltılık sayının karşılığı olan ikili sayı yazılırsa;

5=0101, D=1100, 1=0001, D=1100, 6=0110, 9=1001

değerleri elde edilir.

Yazılan ikili sayıların bir araya getirilmesi ile, sonuç olarak;

$$(5D1D69)_{16} = (010111010001110101101001)_2$$

eşitliği bulunur.

Örnek 25: $(E70F.CA)_{16}$ sayısını ikilik sayıya çevirelim.Herbir basamaktaki sayının karşılığı olan ikili sayı 4 basamaklı olarak yazılırsa;

1110 0111 0000 1111 **.** 1100 1010

sayıları bulunur. Bu durumda;

$$(E70F.CA)_{16} = (1110011100001111.11001010)_2$$

eşitliği elde edilir.

Onaltılı Sayıların Onlu Sayılara Dönüştürülmesi:

Onaltılı sayıyı onlu sisteme çevirmek için, her basamaktaki değer ile basamak ağırlığı çarpılır. Bulunan değerlerin toplanması ile onaltılı sistemden onlu sayı sistemine dönüşüm yapılmış olur.

Örnek 26: $(E70FCA)_{16}$ sayısını onlu sisteme dönüştürelim.

Herbir basamaktaki sayıyı basamak değerleriyle çarpıp, bulunan sayıların toplanması ile;

$$E70FCA = Ex16^5 + 7x16^4 + 0x16^3 + Fx16^2 + Cx16^1 + Ax16^0$$
$$= 1844719 + 458752 + 0 + 3840 + 192 + 10$$
$$= (2307513)_{10}$$

sayısı bulunur. Sonuçta;

$$(E70FCA)_{16} = (2307513)_{10}$$

eşitliği yazılabilir.

Örnek 27: $(5D1.D9)_{16} = (?)_{10}$ dönüşümünü yapalım.

Basamak değerlerinin basamaklardaki sayılarla çarpılıp, bulunan sayıların toplanması ile;

$$\begin{aligned} 5D1.D9 &= 5x16^2 + 13x16^1 + 1x16^0 \,.\, 13x1/16 + 9x1/256 \\ &= 1280 + 208 + 16 \,.\, 13/16 + 9/256 \\ &= (1504.8476)_{10} \end{aligned}$$

sayısı bulunur. Bu durumda,

$$(5D1.D9)_{16} = (1504.8476)_{10}$$

eşitliği yazılabilir.

Onaltılı Sayıların Sekizli Sayılara Dönüştürülmesi:

Onaltılık sayıyı sekizli sisteme çevirmek için en pratik yöntem; onaltılık sayının ikili sisteme ve daha sonra ikili sistemdeki sayının sekizli sisteme çevrilmesidir.

Örnek 28 : $(E0CA)_{16}$ sayısını sekizli sisteme çevirelim.

Önce onaltılı sayı ikili sisteme çevrilir. Onaltılı sistemdeki sayının ikili sisteme çevrilmesi için, her bir basamaktaki sayının ikili karşılığı dört bitlik olarak yazılırsa;

$$E = 1110, \quad 0 = 0000, \quad C = 1100, \quad A = 1010$$

sayıları bulunur. Bulunan sayılar birleştirilirse;

$$(E0CA)_{16} = (1110000011001010)_2$$

sayısı elde edilir. Elde edilen ikili sayı, her grubun karşılığı olan sekizli sayının üçerli gruplar halinde yazılması şeklinde sekizli sayıya dönüştürülürse;

$$(1110000011001010)_2 = (160312)_8$$

eşitliği bulunur.

Not: Bütün sayı sistemlerinde negatif sayıların dönüşümleri aynı şekilde, yalnızca sonuca (-) işareti eklenmek suretiyle yapılır.

2.3. Sayı Sistemlerinde Hesaplama

Tüm sayı sistemlerinde sayılarda işaret kullanılabilir. Yani pozitif ve negatif sayılarla hesaplama yapılabilir. Bu gerçek göz önünde bulundurularak, onluk sayılarda hesaplama yaparken aşağıdaki ilişkiler kullanılabilir. Bu ilişkiler bütün sayı sistemleri için geçerlidir.

a) +a + (+b) = a + b **b)** +a + (-b) = a - b

c) +a - (+b) = a - b **d)** +a - (-b) = a + b

İkili, sekizli ve onaltılı sistemlerdeki hesaplamalarda da 4 temel işlem (toplama, çıkarma, çarpma, bölme) kullanılır. Ancak, dijital bilgisayarlarda kullanılan temel sayı sistemi ikili sayı sistemi olduğundan, ikili sayı sistemindeki dört işlemi detaylı olarak inceleyelim.

2.3.1. İkili Sayı Sisteminde Toplama

İkili sayı sisteminde yapılan toplama işlemi, onlu sayı sisteminde olduğu gibi aynı basamaktaki sayıların toplanması şeklinde yapılır. İkili sayı sistemindeki toplama kuralları aşağıdaki şekilde sıralanabilir.

0 + 0 = 0, 1 +0 = 1, 0 + 1 = 1, 1 + 1 = 10 veya 1 + 1 = 0 Elde 1 (C=1).

‘1 + 1’ toplama işleminde sonuç olarak ‘0’ ve bir soldaki basamağa aktarılmak üzere ‘elde 1’ ortaya çıkar. Bu onluk sayılarla yapılan toplama işlemindeki 9+1 rakamlarının toplamından ‘0’ ortaya çıkması ve eldeki 1’in bir soldaki basamağa aktarılmasına benzer.

Örnek 29: İkili sayı sistemine göre aşağıdaki toplama işlemlerini gerçekleştirelim.

10	101	101
+ 01	+ 010	+ 011
11	111	1000

Not: Çok sayıda sayıların alt alta toplanmasında, iki adet 1’in ‘elde 1’ oluşturduğu bilinerek, toplanacak birlerin sayısı tesbit edilir. Her bir çift ‘1’ değeri için, ‘elde 1’ değeri bir soldaki basamağa aktarılır.

Örnek 30 : Aşağıda verilen toplama işlemlerini yapalım.

1110	1101	111011
+ 0110	1111	011011
10100	+ 1011	110101
	100111	+ 010010
		10011101

2.3.2. İkili Sayı Sisteminde Çıkarma

İkili sayılarda çıkarma işleminde özetlenen kurallar uygulanır:

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

Bu kuralların uygulandığı yöntem, **'doğrudan çıkarma yöntemi'** olarak adlandırılır. Ana sayının çıkarılan sayıdan büyük olması durumunda, yani sonucun '0' veya 0'dan büyük olması durumunda doğrudan çıkarma yöntemi kullanılabilir.

Örnek 31: Aşağıdaki çıkarma işlemlerini doğrudan çıkarma yöntemi ile yapalım.

10110	101110
- 1101	- 10011
1001	11011

Çıkarma işlemi sonucunun 0'dan küçük olması durumunda doğrudan çıkarma yöntemi kullanılamaz. Bu nedenle, sonucun 0'dan küçük çıktığı işlemleri gerçekleştirmek ve bilgisayarlarda mantıksal uyumlaştırma işlemini kolaylaştırmak amacıyla, **'tümleyen aritmetiğine göre çıkarma'** olarak adlandırılan çıkarma yöntemi kullanılır. Tümleyen aritmetiği ile çıkarma yönteminde tüm çıkarma işlemleri yapılabilmekte ve bu nedenle bilgisayarlarda bu yöntem kullanılmaktadır.

2.3.3. Tümleyen Aritmetiği

Tümleyen aritmetiği, sayısal bilgisayarlarda çıkarma işlemini gerçekleştirmek amacıyla kullanılan matematiksel bir yöntemdir.

Tümleyen aritmetiğini anlamanın en pratik yolu, taşıtlarda kullanılan kilometre sayacını göz önünde bulundurmaktır. Onlu sayı sisteminde çalışan kilometre sayaçları genelde beş basamaklıdır. 00000 başlangıç değerinden ileri doğru gidildiğinde 00001, 00002 gibi artarken, geriye doğru gidildiğinde sayacın değerleri 99999, 99998 gibi azalır. Bu sayaç örneğinde, bir adım ileri gidildiğinde 00001 ve bir adım geriye gidildiğinde 99999 değerine ulaşıldığından bu sayılara birbirinin tümleyeni denmektedir. Buna göre 00002 sayısının tümleyeni 99998 değeridir.

Araçların kilometre sayaçları üzerinde açıklanan tümleyen aritmetiğinin ikilik sayılarda uygulamasıyla iki çeşit tümleyen aritmetiği ortaya çıkar:

‘1’ tümleyeni ve ‘2’ tümleyeni.

‘1’ tümleyeni; $(2^n\text{-}N\text{-}1)$ ve ‘2’ tümleyeni; $(2^n\text{-}N)$ formülleri ile ifade edilir.

Formüldeki 'n' değeri verilen N sayısındaki basamak sayısıdır. Formüllerin incelenmesinden; ‘2’ tümleyeninin, ‘1’ tümleyenine 1 eklenmesi ile oluştuğu görülür. ‘1’ ve ‘2’ tümleyeni mantıkları, onluk sistemde ‘9’ ve ‘10’ tümleyenler şeklinde temsil edilir. Tümleyen aritmetiği çeşitleri, daha genel tabir ile, ‘r’ tabanlı bir sayı sisteminde **‘r tümleyeni’** ve **‘r-1 tümleyeni’** olarak ifade edilebilir.

Bu açıklamalar ışığında tümleyen aritmetiğini iki kısımda inceleyebiliriz: ‘r tümleyeni’ ve ‘r-1 tümleyeni’.

2.3.3.1. ‘r’ Tümleyen Aritmetiği

r tabanlı bir sayı sisteminde, n basamaklı pozitif bir tamsayı N ile temsil edilirse, N sayısının r tümleyeni $r^n\text{-}N$ $(N\neq0)$ olarak tanımlanabilir. Aşağıdaki örnekler, ‘r tümleyeni’ terimini anlamaya yardım edecektir.

Örnek 32 : $(52520)_{10}$ sayısının r tümleyenini (onlu sayı sistemi olduğundan 10 tümleyenini) bulalım.

Verilen sayıda basamak sayısı: n=5 ve taban: r=10 olduğundan; sayının r tümleyeni:

$$r^n\text{-}N = 10^5\text{-}52520 = 47480$$

olarak bulunur.

Örnek 33: $(0.3267)_{10}$ sayısının 10 tümleyenini bulalım.

Verilen sayıda tam sayı kısmı bulunmadığından basamak sayısı;

$10^n = 10^0 = 1$ olarak alınır ve sonuç olarak;
$r^0\text{-}N = 1\text{-}0.3267 = (0.6733)_{10}$

sayısı bulunur.

Örnek 34 : $(25.639)_{10}$ sayısının 10 tümleyenini bulalım.

Tam sayı kısmı 2 basamaklı olduğundan sayının 'r' tümleyeni;

r=10, n=2 ve N=25.639 değerleri ile;

$$r^n - N = 10^2\text{-}25.639 = 74.361$$

olarak bulunur.

Örnek 35 : $(101100)_2$ sayısının 2 tümleyenini bulalım.

Sayı ikili sistemde olduğundan, r=2 ve sayı 6 basamaklı olduğundan n=6 değerleri bulunur. Bu değerler formülde yerine konulursa, verilen ikili sayının 'r' tümleyeni olarak;

$$(2^6) - (101100)_2 = (1000000 - 101100)_2 = 010100$$

değeri bulunur.

Örnek 36 : $(0.0110)_2$ sayısının 2 tümleyenini bulalım.

Verilen ikili sistemdeki sayının tam sayı kısmı bulunmadığından; sayının 2 tümleyeni;

$$2^0\text{-N= 1 - 0.0110 = } (0.1010)_2$$

olarak bulunur.

Yukarıdaki açıklamalardan ve örneklerden, ikili sayı sistemindeki bir sayının 2 tümleyenini bulmanın en kolay yolunun; sayıya sağdan bakarak ilk 1'e kadar olan sayıları olduğu gibi bırakmak (1 dahil), diğer bitlerdeki değerlerin tersini almak (1 ise 0, 0 ise 1 yazmak) olduğu söylenebilir.

r tümleyeni, bütün sayı sistemleri için yukarıda verilen eşitlikten çıkartılabilir. Burada açıklanan 10 ve 2 tümleyenleri, en çok karşılaştığımız sayı sistemleri olduklarından detaylandırılmıştır.

2.3.3.2. 'r' tümleyen aritmetiği ile çıkarma

Onluk sayı sisteminde alışkın olduğumuz ve komşuya git borç al olarak isimlendirebileceğimiz doğrudan çıkarma yöntemi bilgisayarlar için çok kullanışlı değildir. Elektronik elemanlar ile çıkarma söz konusu olduğunda daha kullanışlı (etkin) olan yöntem, sayıların tümleyenini alarak toplama işlemi yapmaktır. Bu yöntemde, 'r' tabanındaki iki pozitif sayının 'M-N' işlemi aşağıdaki gibi özetlenebilir:

1. İki sayıyı çıkarma yerine M sayısının kendisi ile N sayısının 'r' tümleyeni toplanır.

2. Toplama sonucunda elde edilen değer incelenir:

   a) Eğer en soldaki basamakların toplanması sonucunda elde değeri oluşursa (işaret biti), bu değer atılır. Bulunan sonucun '(+) pozitif' olduğu kabul edilir.

   b) Eğer elde değeri oluşmazsa, toplama sonucunda elde edilen değerin 'r' tümleyeni alınır ve bulunan değerin önüne '(-) eksi' işareti konulur.

Örnek 37: 10 tümleyenini kullanarak, (72532 **–** 3250) = ? işlemini yapalım.

M=72532 72532

N=03250 ⟶ 10 tümleyeni N=96750 + 96750

elde 1 69282

işaret biti

İşaret biti 1'dir ve bu durumda sonuç; (+69282) olarak bulunur.

Örnek 38: $(03250)_{10} - (72532)_{10}$ = ? işlemini 'r' tümleyen aritmetiği yöntemi ile yapalım.

N = 03250 03250

M = 72532 ⟶ 10 tümleyeni = 27468 + 27468

elde yok ⟶ 0 30718

Bu durumda 30718 sayısının 'r' tümleyeni alınır. Sonuç olarak;

(-69282) değeri bulunur

Örnek 39: 'M – N' işlemini aşağıdaki verilen sayılarla 'r' tümleyenini kullanarak yapalım.

M = 1010100 1010100

N = 1000100 ⇒ 2 tümleyeni ⇒ + 0111100

elde 1 0010000

Sonuç olarak;

$(0010000)_2$

değeri bulunur.

Örnek 40 : M = 1000100

N = 1010100 olduğuna göre 'M – N' işlemini '2' tümleyenine göre yapalım.

N = 1010100 ise 2 tümleyeni = 0101100 bulunur.

```
      1000100
    + 0101100
    ---------
elde yok 0 | 1110000
```

Bulunan sonucun 'r' tümleyeni alınır. Sonuç ;

$(- 0010000)_2$

olarak bulunur.

Örnek 41: $(15)_{10}$ - $(20)_{10}$ işlemini ikili sayı sisteminde '2 tümleyeni' yöntemi ile yapalım.

$(15)_{10} = (01111)_2 = N$

$(20)_{10} = (10100)_2 = M \longrightarrow$ 2 tümleyeni

```
      01111
    + 01100
    -------
elde yok 0 | 11011
```

İşaret biti '0' olduğundan, bulunan sayının '2 tümleyeni' alınır. Sonuç; $-(00101)_2$ olarak bulunur. Bu sayı $(-5)_{10}$ sayısının karşılığıdır.

Örnek 42: $(219)_{10}$ - $(233)_{10}$ işlemini '2 tümleyeni' yöntemiyle yapalım.

$(219)_{10} = (11011011)_2 \longrightarrow$ 11011011

$(233)_{10} = (11101001)_2$ 2 tümleyeni $\longrightarrow$

```
    + 00010111
    ----------
elde yok 0 | 11110010
```

Bulunan sayının 2 tümleyeni alınırsa sonuç ;

$(-1110)_2$

olarak bulunur.

Örneklerden şöyle bir sonuç çıkarılabilir: r tümleyeni ile çıkarma işleminde işaret biti olarak adlandırılan bite bakılır. İşaret biti '1' ise sonucun (+), işaret biti '0' ise sonucun (-) olduğu bulunur. İşlem buna göre sonuçlandırılır.

2.3.3.3. ‘r-1’ Tümleyen Aritmetiği

r tabanına göre verilen ve yalnızca tam sayı kısmı bulunan pozitif bir n sayısının **‘r-1’ tümleyeni; ‘2^n-N-1’** formülüyle,

‘n’ basamaklı tam sayı ve ‘m’ basamaklı kesirli kısmı bulunan bir sayının **‘r-1’ tümleyeni;**

‘r^n-r^{-m}-N’

formülü ile bulunabilir.

Örnek 43: $(52520)_{10}$ sayısının ‘r-1’ tümleyenini (‘9’ tümleyenini) bulalım.

Sayının yalnızca tam sayı kısmı bulunduğundan, ‘2^n-N-1’ formülü uygulanabilir. Taban = 10 ve basamak sayısı n = 5 olduğuna göre ilgili formülden sonuç;

$$R^n\text{-}N\text{-}1 = 10^5\text{-}52520\text{-}1=47479$$

olarak bulunur.

Örnek 44: $(0.3267)_{10}$ sayısının 9 tümleyenini bulalım.

Sayının tam sayı ve kesirli kısmı bulunduğundan ilgili formül uygulanırsa;

$$r^n\text{-}r^{-m} – N = 10^0 - 10^{-4} - 0.3267 = 1\text{-}0.0001\text{-}0.3267$$
$$= 0.9999 - 0.3267 = 0.6732$$

değeri bulunur.

Örnek 45: $(101100)_2$ sayısının ‘r-1’ tümleyenini (1 tümleyeni) bulalım.

Verilen sayı ikili sistemde olduğundan r=2 ve sayıda 6 basamak bulunduğundan n=6’ dır. Bu durumda,

$$2^n\text{-}N\text{-}1=2^6\text{-}101100\text{-}1=1000000\text{-}101100\text{-}1$$

$$= (010011)_2$$

değeri bulunur.

Örnek 46: $(0.0110)_2$ sayısının 1 tümleyenini bulalım.

İkili sistemdeki sayıda tamsayı kısmı bulunmadığından n=0 ve kesirli kısım 4 basamaklı olduğundan m=4' dür. İlgili formülün uygulanması ile sonuç;

$$(2^n - 2^{-4} - 0.0110) = (1-0.0001 - 0.0110)$$

$$= (0.1111-0.0110)_2$$

$$= (0.1001)_2$$

olarak bulunur.

Örneklerden görüleceği gibi onluk sistemdeki bir sayının 'r-1' tümleyeni (9 tümleyeni); her basamağın 9'dan çıkarılması ile elde edilir. İkili sistemdeki bir sayının 'r-1' tümleyenini (1 tümleyenini) bulmak daha basittir. Verilen sayıdaki 1'ler 0, 0'lar 1 yapılınca ortaya 'r-1' tümleyeni çıkar.

İkili sayı sisteminde 1 tümleyeni kolayca bulunduğundan, 2 tümleyeninin istenildiği durumlarda; 1 tümleyenine, işleme göre '1' veya '$\mathbf{r^{-m}}$' değerinin eklenmesiyle 1 tümleyeninin üretilmesi işlemi tercih edilebilir.

Örnek 47: 1 tümleyeni $(01001011)_2$ olan sayının 2 tümleyenini bulalım.

Verilen sayıya '1' eklenmesi ile sayının 2 tümleyeni;

$$\begin{array}{r} 01001011 \\ +\quad\quad 1 \\ \hline 01001100 \end{array}$$

olarak bulunur.

Örnek 48: 1'tümleyeni $(0.1011)_2$ olan sayının 2 tümleyenini bulalım.

Verilen sayının '2' tümleyeni bulmak için önce eklenmesi gereken sayı bulunur. Eklenmesi gereken sayı;

$\mathbf{r^{-m}} = 2^{-4} = 0.0001$ olduğundan '2' tümleyeni;

$$\begin{array}{r} 0.1011 \\ +\ 0.0001 \\ \hline 0.1100 \end{array}$$

olarak bulunur.

2.3.3.4. ‘r – 1’ Tümleyen Yöntemi ile Çıkarma

‘r – 1’ tümleyeni ile çıkarma işlemi tamamen ‘r’ tümleyeni ile çıkarma işleminin aynısıdır. Yalnızca sonucun pozitif olduğu durumlarda, düzeltme biti denilen 1 sayısının eklenmesi işlemi yapılır. ‘r’ tabanında iki pozitif sayının M-N işlemi (r-1 tümleyeni yöntemi ile) aşağıdaki şekilde özetlenebilir:

1- M sayısının kendisi ile N sayısının ‘r-1’ tümleyeni toplanır.

2- Toplama sonucunda bulunan değerin taşma (işaret) biti kontrol edilir.

a- Eğer taşma biti oluşursa (işaret biti 1), bulunan değere 1 değeri eklenir.

b- Eğer taşma biti oluşmazsa (işaret biti 0), toplama sonucunda elde edilen sayının ‘r-1’ tümleyeni alınır ve önüne (-) işareti konur.

Örnek 49: M=72532,

N=03250 ise ‘M-N’ işlemini ‘r-1’ tümleyenine göre yapalım.

İşlemi yapabilmek için önce çıkarılan sayının ‘r-1’ tümleyeninin bulunması gerekir. Bulunan bu değer ile ‘M’ sayısı toplanır.

N’nin 9 tümleyeni $\Rightarrow$ 96749

(taşma /işaret biti) $\rightarrow$

```
  72532
+ 96749
-------
1 69281
```

işaret biti ‘1’ olduğundan sonuca ‘1’ eklenir. Bu durumda,

```
  69281
+     1
-------
  69282
```

değeri bulunur.

Örnek 50: M = 03250

N = 72532 ise ‘M-N’ işlemini 9 tümleyenine göre yapalım.

Çıkarılan sayının 9 tümleyeni alınıp, toplama işlemi yapılırsa;

$$\begin{array}{rcll} & & & 03250 \\ \text{N sayısının 9 tümleyeni} & \Rightarrow & 27467 & +\ 27467 \\ & & & \hline \\ & & \text{(taşma yok)} & 0 \mid 30717 \end{array}$$

İşaret biti değeri ‘0’ olduğundan, sonucun 9 tümleyenini alıp, önüne (-) işareti koymamız gerekir. Sonuç ;

$$(-\ 69282)_{10}$$

olarak bulunur.

Örnek 51: M=1010100 ve N=1000100 olduğuna göre ‘M-N’ işlemini (r-1) tümleyenine göre yapalım.

N’nin 1 tümleyeni $\Rightarrow$ 0111011 olduğundan;

$$\begin{array}{rl} & 1010100 \\ & +\ 0111011 \\ & \hline \\ \text{taşma var} \longrightarrow & 1 \mid 0001111 \end{array}$$

sayısı elde edilir. Sonuca ‘1’ eklenmesi gerekir.

$$\begin{array}{r} 0001111 \\ +\quad\quad 1 \\ \hline 00010000 \end{array}$$

Bu durumda sonuç;

$$(10000)_2$$

olarak bulunur.

Örnek 52: M = 1000100,

N = 1010100 ise M-N işlemini 1 tümleyenine göre yapalım.

1000100

N'nin 1 tümleyeni⇒ + 0101011

0 1101111 ⟶ işaret biti = 0'dır.

Bu durumda sonuç (-) dir ve cevap;

$(-0010000)_2$

olarak bulunur.

Örnek 53: $(15)_{10}$ - $(20)_{10}$ = ? işlemini 1 tümleyenine göre yapalım.

Sayılar onlu sistemde verildiğinden, sayıların ikili sisteme dönüştürülmesi gerekir. Sayılar ikili sisteme dönüştürülür ve çıkarılan sayının '1' tümleyeni alınarak toplama işlemi yapılırsa;

(15)= 01111 ⇒ 01111

(20) = 10100 ⇒ + 01011

0 11010

Bulunan sayının 1 tümleyeninin alınması ile sonuç;

$(-00101)_2$

olarak bulunur.

2.3.4. İkili Sayı Sisteminde Çarpma

İkili sayı sisteminde çarpma işleminde onluk sistemde kullanılan işlem sırası takip edilir ve '0' ve '1' değerlerinin çarpılması söz konusu olduğundan aşağıdaki kurallar geçerlidir.

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

Örnek 54: $(1011)_2 * (101)_2$ ve $(10111)_2 * (110)_2$ işlemlerini yapalım.

```
   1011                    10111
 x  101                  x  110
 ------                  ------
   1011                    00000
  0000                    10111
+1011                   +10111
 ------                  ------
 110111                  10001010
```

2.3.5. İkili Sayı Sisteminde Bölme

İkili sayılarda bölme işlemi, onluk sayı sisteminde olduğu gibi bölünenden bölenin çıkarılması işlemine sonuç sıfır kalıncaya kadar devam edilmesiyle gerçekleştirilir.

Örnek 55: $(10110)_2 \div (100)_2 = ?$ işlemini yapalım.

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

Sonuç = $(101.1)_2$ bulunur.

Örnek 56: $(1111101) \div (101) = ?$ işleminin sonucunu bulalım.

```
1111101 | 101
-101    | 11001
-------
 0101
-101
-------
 0101
-101
-------
  000
```

Sonuç = $(11001)_2$ olarak bulunur.

Tekrarlama ve Çalışma Soruları

1. Sayı sistemlerinin tarihsel gelişimini açıklayınız.
2. Bilgisayar teknolojisinde kullanılan sayı sistemlerini sıralayınız.
3. Sayı sistemlerinin taban değerine göre sahip olacakları denklemleri yazınız.
4. Onaltılı sayı sisteminde kullanılan harflerin temsil ettikleri anlamları açıklayınız.
5. Binary sayı sistemini tanımlayınız.
6. 'BIT', 'en düşük anlamlı bit''en yüksek anlamlı bit' terimlerini açıklayınız.
7. Sekizli sayı sisteminin özelliklerini özetleyiniz.
8. 0 - 20 arasındaki sayıları, ikili, sekizli ve onaltılı sistemde yazınız.
9. $(1471)_{10}=(?)_2$ işlemini yapınız.
10. $(571,571)_{10}=(?)_2$ dönüşümünü yapınız.
11. $(346,125)_{10}=(?)_8$ işlemini yapınız.
12. $(145,135)_{10}=(?)_{16}$ işlemini yapınız.
13. $(1453,1451)_{10}=(?)_{16}$ işlemini yapınız.
14. Bilgisayarlarda ikili sayı sistemi ile birlikte sekizli ve onaltılı sayı sistemlerinin tercih edilme sebepleri nelerdir?
15. $(101101011)_2 = (?)_8$ ve $(1101101.101101)_2 =(?)_8$ dönüşümlerini yapınız.
16. $(101110110110111100)_2 = (?)_{16}$ ve $(101101.10111101)_2 = (?)_{16}$ işlemlerini yapınız.
17. $(3526)_8 = (?)_2$ ve $(25.316) = (?)_2$ çevrimlerini yapınız.
18. $(2146)_8 = (?)_{16}$ ve $(142.37)_8 = (?)_{16}$ dönüşümlerini yapınız.
19. $(10110111)_2= (?)_{10}$ ve $(10110.1011)_2 = (?)_{10}$ çevrimlerini yapınız.

20. $(2164)_8 = (?)_{10}$ ve $(142.37)_8 = (?)_{10}$ dönüşümlerini yapınız.
21. $(5E3)_{16} = (?)_2$ ve $(1F2.4E9)_{16} = (?)_2$ işlemlerini yapınız.
22. $(3F1)_{16} = (?)_8$ ve $(2A.B3)_{16} = (?)_8$ çevrimlerini yapınız.
23. $(2A1)_{16} = (?)_{10}$ ve $(F2.3B)_{16} = (?)_{10}$ dönüşümlerini yapınız.
24. Aşağıdaki toplama işlemlerini yapınız.

```
 11011        11101        11110
+ 1101      + 1111         11011
------      ------       + 10111
  (?)          (?)        -------
                            (?)
```

25. Aşağıdaki çıkarma işlemlerini doğrudan çıkarma yöntemi ile yapınız.

```
  1011        11011        101110
- 111       - 10111      - 10111
-----       -------      -------
  (?)          (?)          (?)
```

26. $(55220)_{10}$ ve $(2745)_{10}$ sayılarının r tümleyenlerini bulunuz.
27. $(10111011)_2$ ve $(10110011)_2$ sayılarının r tümleyenlerini bulunuz.
28. İkili sistemdeki bir sayının r tümleyenini bulmanın pratik yöntemini açıklayınız.
29. $(55220-2745)_{10}$ işlemini 10 tümleyenine göre yapınız.
30. $(2745-55220)_{10}$ işlemini 10 tümleyenine göre yapınız.
31. $(10111011-10110011)_2$ işlemini 2 tümleyenine göre yapınız.
32. $(10110011-10111011)_2$ işlemini 2 tümleyenine göre yapınız.
33. $(49262)_{10}$ ve $(36123)_{10}$ sayılarının ‘r-1’tümleyenlerini bulunuz.
34. $(101110011)_2$ ve $(100111011)_2$ sayılarının ‘r-1’ tümleyenlerini bulunuz.

35. $(49262\text{-}36123)_{10}$ işlemini ‘9’ tümleyeni kullanılarak yapınız.
36. $(36123\text{-}49262)_{10}$ işlemini ‘9’ tümleyeni kullanılarak yapınız.
37. $(101110011 - 100111011)_2$ işlemini ‘r-1’ tümleyeni kullanılarak yapınız.
38. $(100111011 - 101110011)_2$ işlemini ‘r-1’ tümleyeni kullanılarak yapınız.
39. Aşağıdaki çarpma işlemlerini yapınız.

    $(1011011) * (101) = (?)_2$

    $(1011101) * (1100) = (?)_2$

    $(111100000) * (1000) = (?)_2$
40. Aşağıdaki bölme işlemlerini yapınız.

    $(1011011) \div (101) = (?)_2$

    $(1011101) \div (1100) = (?)_2$

    $(111100000) \div (1000) = (?)_2$

BÖLÜM 3

KODLAMA VE KODLAR

(CODING AND CODES)

Giriş

Kodlama, iki küme elemanları arasında karşılıklığı kesin olarak belirtilen kurallar bütünüdür diye tanımlanabilir. Diğer bir deyişle, görünebilen, okunabilen yazı, sayı ve işaretlerin değiştirilmesi işlemine **'kodlama'** denir. Başka bir bakış açısı ile, sonlu elemana sahip bir kümenin her bir elemanına bir kod verilmesi, kodlama olarak tanımlanır.

'Morse' alfabesi kodlamaya iyi bir örnektir.Kodlama işlemine diğer bir örnek, bilgisayarın çevresel birimleri ile merkezi işlem ünitesi arasındaki bilgi iletişimidir. Bilgisayarlarda, bir alfabetik-sayısal kaynak olan klavyeden gönderilen bilgi, 7 veya 8 bitlik ikili sayılar şeklinde kodlandıktan sonra ilgili birime gönderilir.

Kodlama işlemi yalnızca onluk sistemdeki sayıları (0, 1, 2,....,9) içerebileceği gibi, alfabetik ve alfasayısal bilgilerin kodlanmasını içerebilir. Farklı bilgileri kodlama ihtiyacı ve değişik alanlarda kodlama gereksinimi çeşitli kodlama yöntemlerini doğurmuştur.

Kodlama işlemi aşağıdaki avantajları sağlar:

1. Aritmetik işlemlerde kolaylık sağlar.
2. Hataların bulunmasını kolaylaştırır.
3. Hataların düzeltilmesi işlemlerini basitleştirir.
4. Bellek işlemlerinde verimliliği artırır.
5. Bilgilerin işlenmesi işleminin insanlarca kolayca anlaşılmasını sağlar.

Yalnızca sayısal karakterlerin kodlanmasıyla ortaya çıkan kodlara **'sayısal kodlar'** (BCD kodları) denilirken, alfabetik ve sayısal karakterlerin kodlanmasını içeren kodlama yöntemlerine **'alfasayısal kodlar'** denir. Bu durumda kodlar iki grup altında incelenebilir: sayısal ve alfa sayısal kodlar.

3.1. Sayısal Kodlar

Onlu bir sayının ikili sayı sistemindeki karşılığının yazılması ile oluşan kodlama sistemi, '**yalın ikili kodlama'** (pure binary coding) olarak isimlendirilir. Sayısal sistemlerde kullanılan kodlama sistemleri yalın ikili sayı sisteminde olmayabilir.

Yalnızca sayısal karakterlerin kullanıldığı sayısal kodlama sistemlerinin çok geniş uygulama alanı olması nedeni ile, çok farklı sayısal kodlama yöntemleri kullanılmaktadır. Sayısal kodlama yöntemlerine örnek olarak;

i- BCD kodu,

ii- Gray kodu,

iii- +3kodu,

iv- Aiken kodu,

v- 5'te 2 kodu,

vi- Bar kodu,

kodlama yöntemleri verilebilir.

Sayısal kodlama yöntemlerine örnek olarak verilen kodlama çeşitlerine genel özellikleri ile özetleyelim.

3.1.1. BCD Kodu (Binary Coded Decimal Code) - 8421 Kodu

Onluk sistemdeki bir sayının, her bir basamağının ikilik sayı sistemindeki karşılığının dört bit şeklinde yazılması ile ortaya çıkan kodlama yöntemine, **'İkili Kodlanmış Onlu Sayı Kodu - BCD kodu'** (**B**inary **C**oded **D**ecimal Code) ismi verilir. Onluk sayı sistemi 0 ile 9 arasındaki sayıları içerdiğinden, her basamaktaki sayının ikili sistemde kodlanması için 4 bite ihtiyaç vardır. Onlu bir sayıyı BCD kodlu olarak yazmak için, onlu sayının herbir basamağı 4 bitlik ikili sayı grupları şeklinde yazılır. Yazılan gruplar bir araya getirilince BCD kodlu sayı elde edilir.

Örnek 1: $(263)_{10}$ sayısını BCD kodu ile kodlayalım.

Herbir basamaktaki sayının ikili karşılığı 4 bit olarak yazılırsa;

2	6	3	
0010	0110	0011	sayıları bulunur.

Sayıların birleştirilmesiyle;

$$(263)_{10} = (001001100011)_{BCD}$$

eşitliği elde edilir.

Burada unutulmaması gereken, bulunan sayının $(263)_{10}$ sayısının ikili sayı sistemindeki karşılığı olmadığıdır.

Örnek 2: $(\underline{1001}\ 0\underline{011}\ 0\underline{110})_{BCD}$ sayısını onlu sisteme çevirelim.
Sayı dörderli gruplara ayrılarak her bir gruptaki ikili sayıların onlu karşılığı yazılırsa;

$$(1001\ 0011\ 0110)_{BCD}$$
$$\downarrow \quad \downarrow \quad \downarrow$$
$$9 \quad 3 \quad 6$$

sayıları bulunur. Bulunan sayıların bir arada yazılmasıyla sonuç olarak;

$$(100100110110)_{BCD} = (936)_{10}$$

sayısı elde edilir.

3.1.2. Gray Kodu

Gray kodlama yöntemi, basamak ağırlığı olmayan bir kodlama yöntemidir. Basamak ağırlığının olmaması, her bir basamaktaki sayıların basamak ağırlıklarına göre karşılıklarının olmamasıdır.

Sayısal elektronik ve bilgisayar giriş-çıkış işlemlerinde kullanılan Gray kodlama yöntemi, minimum değişimli kodlar sınıfı içerisinde yer alır. Bunun nedeni bir sayıdan diğerine geçerken yalnızca bir bitin konum değiştirmesidir.

Örneğin; yalın binary kodlamada $(3)_{10} = (0011)_2$ değerinden $(4)_{10} = (0100)_2$ değerine geçerken üç bitin değeri değişirken, gray kodlamada yalnızca bir bitin değeri değişir.

Gray kodlanmış sayılarda basamak değeri olmadığından, bu kodlama yönteminin aritmetik işlemlerin olduğu yerlerde kullanılması mümkün değildir. Ancak sütun esasına göre çalışan cihazlardaki hatayı azalttığından, giriş / çıkış birimlerinde ve analog - dijital çeviricilerde tercih edilirler. Yalnızca, 9'dan 0'a geçişte çok sayıda bit konum değiştirir.

Onlu sayıların karşılığı olan ikili sayıları Gray kodlanmış olarak ifade etmek için, bir sayıdan diğerine geçişte tek bir bitin değer değiştirmesi esas alınır. Tablo 3.1'de, 0-15 arasındaki onlu sayıların karşılığı olan ikili ve gray kodlanmış sayılar görülmektedir. Gray kodlu sayıların mahsuru; toplama, çıkarma ve diğer aritmetik işlemleri yapabilmek için ikili sayı sistemine dönüştürülme zorunluluğudur. Bu durumda ikili sayıları Gray koda çevirmek veya Gray kodlu bir sayının ikili karşılığını bulmak için aşağıdaki yöntemler kullanılır.

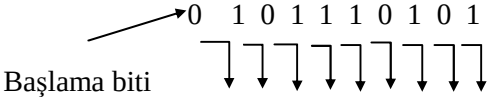
Onlu Değer	**İkili Değer** 8421	**Gray Kodu** Basamak değeri yok
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Tablo 3.1. Yalın ikili kodlu ve gray kodlu sayılar.

i- İkili Sayıların Gray Koduna Çevrilmesi:

İkili sistemdeki bir sayıyı Gray kodlu sayı şekline dönüştürmek için, en yüksek basamak değerine sahip bitin solunda '0' olduğu kabul edilip, her bit solundaki bit ile toplanarak yazılır. Bu işleme endüşük basamak değerlikli bite kadar devam edilir. Elde edilen sayı Gray kodlu sayıdır.

Örnek 3: $(101110101)_2$ ikili sistemdeki sayıyı Gray koduna çevirelim.

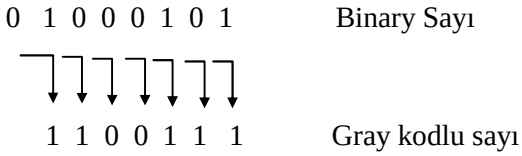


Sonuç olarak;

$$(101110101)_2 = (111001111)\ \text{Gray}$$

eşitliği yazılabilir.

Örnek 4: $(1000101)_2$ Binary sayısını Gray koduna çevirelim.



Sonuçta;

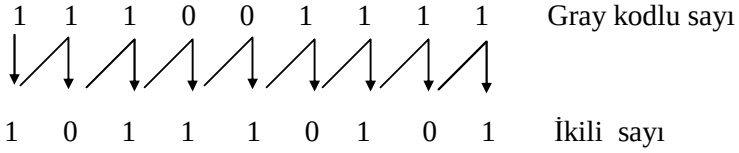
$$(1000101)_2 = (1100111)$$

eşitliği bulunur.

ii- Gray Kodlu Bir Sayının İkili Sayılara Çevrilmesi:

Gray kodlu bir sayıyı ikili sistemdeki sayı şekline dönüştürmek için, en soldaki bit olduğu gibi aşağıya indirilir ve indirilen sayıyla bir sonraki basamakta bulunan sayı toplanarak yazılır. Bulunan sayı ile bir sonraki basamaktaki sayı toplanır ve bu işleme en düşük değerlikli bite kadar devam edilir.

Örnek 5: $(111001111)_{GRAY}$ sayısını ikili sayı sistemine çevirelim.

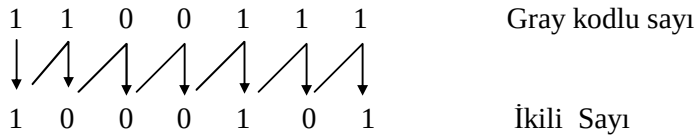


Sonuçta;

$$(111001111)_{GRAY} = (101110101)_2$$

eşitliği bulunur.

Örnek 6: $(1100111)_{GRAY}$ sayısını ikili sayı sistemine çevirelim.



Sonuç olarak;

$$(1100111)_{GRAY} = (1000101)_2$$

eşitliği bulunur.

3.1.3. Artı 3 (Excess 3) Kodu

Artı 3 kodu (+3 Code), BCD kodu ile ilgilidir ve belirli aritmetik işlemlerde işlem kolaylığı nedeniyle BCD kodu yerine kullanılır. Bir onlu sayının Artı 3 kodundaki karşılığı, onlu sayının karşılığı olan ikili sayıya 3 eklenmiş halidir. Bu nedenle bu kodlama yöntemi, '3 fazlalık kodu' olarak ta isimlendirilir. Artı 3 kodundaki sayılar, BCD kodunda olduğu gibi dört bitlik ikili sayılar şeklinde ifade edilir.

Hesaplama yapmada ve hataları düzeltmelerde sağladığı kolaylıklara rağmen, tümleyenini almadaki güçlükler nedeniyle son zamanlarda nadiren kullanılmaktadır.

Örnek 7: $(48)_{10}$ sayısını Artı 3 koduna çevirelim.

```
  4      8
+ 3    + 3      her bir basamağa 3 eklenir,
  7     11      bulunan sonuç 4 bitlik ikili sayıya çevrilir.
  ↓      ↓
0111   1011
```

Bulunan sayılar yan yana yazılarak Artı 3 kodlu sayı elde edilir. Buna göre;

$$(48)_{10}=(01111011)_{+3}$$

eşitliği yazılabilir.

Örnek 8: 3 fazlalık kodlu $(11000110)_{+3}$ sayısının onlu sistemdeki karşılığını bulalım.

Sayı dörder bitlik gruplara ayrılır ve herbir grubun karşılığı olan onlu sayı bulunur. Bu işlemlerle,

$$(11000110)_{+3} \rightarrow 1100 \quad 0110 = (12 \quad 6)_{+3}$$

sayıları elde edilir. Bulunan herbir sayıdan 3 çıkarılırsa;

```
  12      6
- 3     - 3
---    ---
  9      3
```

$(93)_{10}$ sayısı bulunur. Bu durumda,

$$(10100110)_{+3}=(93)_{10}$$

eşitliği yazılabilir.

3.1.4. 5'de 2 Kodu

5'de 2 kodunda, her onlu sayı, içinde mutlaka iki tane '1' bulunan 5 bitlik ikili sayı ile temsil edilir. Bütün sayılarda mutlaka iki tane '1' bulunduğundan hataların kolayca bulunmasını sağlar. Sayılar ikili sistemde ifade edilirken basamak değerleri '7 4 2 1 0' şeklinde sıralanır. $(0)_{10}$ sayısını 5'te 2 kodunda ifade etmek için (11000) kombinasyonu kullanılır.

Örnek 9 : $(6)_{10}$ sayısının 5'de 2 kodundaki karşılığını bulalım.

Basamak değerleri 74210 olduğundan ve mutlaka 2 tane 1 bulunması gerektiğinden;

$$(6)_{10} = (01100)_{5'de\ 2}$$

eşitliği bulunur.

Örnek 10: $(01010\ 10100)_{5'te\ 2}$ 5'te 2 kodlanmış sayının onlu sistemdeki karşılığını bulalım.

Herbir basamaktaki sayı 5 bit ile ifade edildiğinden, sayı 5 bitlik gruplara ayrılıp herbir grubun karşılığı olan onlu sayı yazılırsa;

(01010 10100)
↓ ↓
5 9

sayıları bulunur. Bu durumda,

$$(0101010100)_{5'te\ 2} = (59)_{10}$$

eşitliği yazılabilir.

0-9 arasındaki sayıların 5'te 2 kodunda ifade edilmesi ile Tablo 3.2'deki değerler elde edilir.

Desimal Sayı	**5'te 2 Kodlu Sayı**
	7 4 2 1 0
0	1 1 0 0 0
1	0 0 0 1 1
2	0 0 1 0 1
3	0 0 1 1 0
4	0 1 0 0 1
5	0 1 0 1 0
6	0 1 1 0 0
7	1 0 0 0 1
8	1 0 0 1 0
9	1 0 1 0 0

Tablo 3.2. Onlu sayıların 5'te 2 kodundaki karşılıkları

3.1.5. Eşitlik (Parity) Kodu

Binary bilginin bir yerden başka bir yere taşınması dijital sistemlerde sıkça karşılaşılan bir olaydır. Bilginin bir yerden başka bölgeye taşınması sırasında, değişik nedenlerden dolayı gürültü oluşması ve oluşan gürültünün iletilen bilgiyi bozması zaman zaman karşılaşılan hadiselerdir. Bilgi iletimi sırasında bu şekilde oluşan hataları tespit etmek ve mümkünse düzeltmek sayısal sistemlerin özelliklerindendir.

Hataları tespit etmede kullanılan en yaygın ve en kolay yöntem eşitlik biti kodlama (parity code) yöntemidir. Bu yöntemde, hataların ortaya çıkarılmasını sağlamak amacıyla BCD kodlu sayının sağındaki veya solundaki basamağa **'eşitlik biti'** (parity bit) eklenir. Eşitlik biti, kodlanan veride 1 yada 0'ların tek mi, çift mi olduğunu belirtir. İki türlü eşitlik biti yöntemi bulunmaktadır: Çift eşitlik (even parity) ve tek eşitlik (odd parity).

Çift eşitlik yönteminde; eşitlik bitinin değeri, kodlanacak bilgideki 1'lerin toplam sayısı (eşitlik biti dahil) çift olacak şekilde seçilir. Kodlanacak sayıdaki 1'lerin sayısı tek ise, eşitlik biti olarak '1' eklenir. Kodlanacak bilgideki 1'lerin sayısı çift olması durumunda ise, eşitlik biti olarak '0' eklenir.

Örnek 11: $(1000011)_2$ sayısına çift eşitlik biti yöntemine göre eşitlik biti ekleyelim.

Kodlanacak bilgide (1000011) üç adet '1' bulunduğundan, bilgideki 1'lerin sayısını çift yapmak için eşitlik biti olarak '1' eklenir ve sonuç olarak;

(11000011)

sayısı oluşur.

Örnek 12: $(1000001)_2$ sayısını çift eşitlik yöntemine göre kodlayalım.

Verilen sayıda çift sayıda '1' bulunduğundan, eşitlik biti olarak '0' eklenir ve kodlama işlemi sonucunda;

'01000001'

bilgisi oluşur.

Tek eşitlik bit yöntemi; aynı mantığa göre düzenlenir. Tek fark kodlanan bilgideki 1'lerin sayısı tek olmalıdır.

Örnek 13: $(1000001)_2$ sayısına tek eşitlik biti yöntemini uygulayalım.

'1000001' sayısında çift sayıda '1' bulunduğundan, eşitlik biti değeri '1' olur ve

kodlaşmış bilgi;

‘11000001’

değerini alır.

Örnek 14: $(1000011)_2$ sayısına tek eşitlik biti ekleyelim.

Verilen sayıda tek sayıda ‘1’ bulunduğundan, eklenecek eşitlik biti ‘0’ olur ve sonuçta;

‘01000011’

sayı dizisi elde edilir.

Eşitlik kodunda unutulmaması gereken nokta, çift veya tek eşitlik biti yönteminde eklenen bitin bilginin bir parçası olduğudur. Normalde 7 bit olarak ifade edilen bilgiler, eşitlik bitinin eklenmesiyle 8 bitlik bilgiler haline dönüşür. Eşitlik kodlama yönteminin avantajı, bilginin iletilmesi sırasında bir bitin değerinin değişmesi ihtimali olan yerlerde hatanın alıcı tarafından kolayca tespit edilebilmesidir.

3.1.6. Aiken Kodu

Aiken kodu; 4 basamaklı ve basamak değerlerinin ‘2421’ şeklinde ifade edildiği bir kodlama şeklidir. Onlu sistemde 5’e kadar olan sayıları kodlamak için sağ taraftaki basamaklar kullanılırken, 5’den büyük değerleri ifade etmek için sol taraftaki bitler tercih edilir.

Bu kodlama şekli simetrik kodlamaya bir örnektir. (0-4) arasındaki sayılar için normal ikili sayılar kullanılırken, (5-9) arasındaki sayılar için başlangıçtaki sayıların simetriği kullanılır(Tablo 5.3).

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

Tablo 3.3 Onlu sayıların Aiken kodundaki karşılıkları

Örnek 15 : $(3)_{10}$ ve $(7)_{10}$ sayılarını Aiken Koduna göre kodlayalım.

$(3)_{10}$ sayısı '2421' basamak değerleri göz önünde bulundurularak yazılırsa;

$(3) = (0011)_{Aiken}$

değeri elde edilir.

Aynı şekilde, $(7)_{10}$ sayısı basamak değerleri göz önünde bulundurularak

yazılırsa;

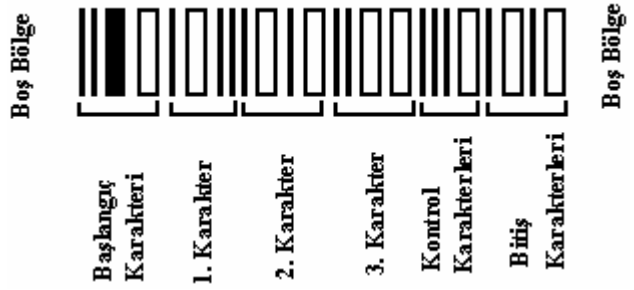
$(7)_{10} = (1101)_{Aiken}$ eşitliği bulunur.

3.1.7.Bar (Çubuk) Kodu

Onlu sayıların farklı şekilde düzenlenmiş çubuklarla ifade edildiği kodlama sistemi '**bar kodu**' olarak isimlendirilir. Diğer bir deyişle, Karakterlerin(rakam veya harf) farklı kalınlıktaki çizgiler ve boşluklar ile temsil edildiği kodlama sistemi 'barkod' olarak adlandırılır.

Klavye'ye alternatif olarak kullanılan bar kodu yöntemi, veri giriş / çıkışının kolay olması nedeniyle özellikle stoklama işlemlerinde ve marketlerde yaygın olarak kullanılmaktadır.

Karakterleri temsil etmek için kullanılan çizgilerin uygun araçlarla okunup çözümlenmesi ve bilgisayara aktarılması için çeşitli barkod yöntemleri ve barkodları okuyacak farklı teknolojiler bulunmaktadır. Şekil 3.1'de örnek bir barkod sisteminin yapısı görülmektedir.



Şekil 3.1 Barkod'un genel yapısı

Şekil 3.1 de genel yapısı verilen barkod'da bulunan bölgelerin temsil ettikleri anlamlar aşağıdaki şekilde özetlenebilir.

Başlangıç/Bitiş Karakterleri: Barkodun iki yanını tanımlayan özel karakterler, farklı barkod çeşitleri farklı başlangıç ve bitiş karakterleri kullanır.

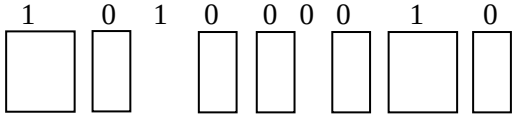
Kontrol Karakterleri**:** Bir kodda bulunan değerlerden elde edilen ve barkod içerisine yerleştirilen değerdir. Kontrol Karakteri, kodun doğru olarak çözümlenip-çözümlenmediğinin kontrolü için kullanılır. Farklı barkod çeşitleri, farklı kontrol karakterleri hesaplama yöntemleri kullanılır. Kontrol karakterinin isteğe bağlı olarak kullanıldığı barkod yöntemlerinin yanında kontrol karakterinin zorunlu olarak kullanıldığı yöntemlerde bulunmaktadır.
Boş Bölgeler: Barkod başlangıç ve bitişinde bulunması gerekli boş alanlardır.

Karakterleri ifade eden çubuk kombinasyonlarının oluşturulmasında iki farklı yöntem vardır: İki seviyeli kod ve çok seviyeli kod.

İki seviyeli bar kodlama sisteminde; geniş çubuk veya aralık (boşluk) binary '1' değerini, dar çubuk veya aralık '0' değerini ifade eder. Dar ve geniş çubukları/boşlukları ifade etmek için kullanılan yaygın standart; 0,19 mm ve 0,38 mm genişliğidir. Bu şekilde gösterimin kullanıldığı çeşitli bar kod yöntemleri bulunmaktadır. İki seviyeli kodlara örnek olarak; 39 bar kodu, 25 bar kodu ve HP41C bar kodu olarak isimlendirilen yöntemler verilebilir.

39 bar kodu, 9'da 3 kodu olarak tanımlanır ve 9 tane çubuk veya aralığı içerir. 9 çubuk veya aralıktan 3 tanesi geniştir.

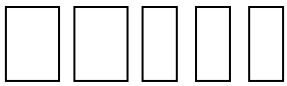
Örnek 16: Şekil 3.1'de gösterilen kombinasyon 39 bar koduna bir örnektir. Bu örnekte toplam 9 çubuk / aralık bulunmaktadır. Bunlardan 1. ve 8. sıralardaki çubuklar ile 3. sıradaki aralık lojik '1' değerini, diğer çubuk ve aralıklar '0' değerlerini temsil etmektedir.



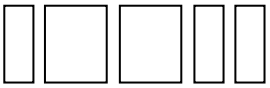
Şekil 3.1. 39 Bar koduna örnek gösterim.

25 Bar kodu olarak isimlendirilen kod, 5'te 2 kodunun çubuklarla ifade edilen şeklidir. Bu kodda bilgiler yalnızca çubuklarla ifade edilir, aralıklar bir anlam içermez. Her bilgi 5 çubuk ile oluşturulur ve bunlardan yalnızca 2 tanesi geniştir. İnce çubuklar '0', kalın çubuklar '1' anlamına gelir.

Örnek 17: Onlu sistemdeki ‘0’ ve ‘6’ değerlerini 25 bar kodu ile gösterelim.



dizisi '0' değerini temsil ederken ,

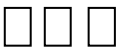


dizisi onlu '6' değerini temsil eder.

Şekil 3.3. Onlu sayı sistemindeki ‘0’ ve ‘6’ değerlerinin 25 bar kodu ile gösterilmesi.

HP41C kodu, yalnızca çubukların anlam içerdiği diğer bir çubuk kodlama yöntemidir. Bu yöntemde de dar çubuklar ‘0’ değerini temsil ederken, geniş çubuklar ‘1’ değerini temsil ederler.

Çok seviyeli kodlar grubu olarak isimlendirilen ikinci çubuk kodlama sisteminde genellikle 4 seviyeli sayısal kodlar kullanılır. Bu sistemde çubuk+aralıklar 7 karakter uzunluğundadır ve her bilginin temsilinde mutlaka 2 çubuk, 2 aralık bulunmalıdır. Kod 93 ve Kod 128 olarak isimlendirilen kodlar, 4 seviyeli bar kodlardır. Şekil 3.4’de onlu sayıların 4 seviyeli kodla ifade edilmesine örnekler görülmektedir.

Sayı	Binary Karşılığı	Örnek Kodlama
0	3 2 1 1	
1	2 2 2 1	
⋮		
9	3 1 1 2	
başla/bitir	1 1 1	

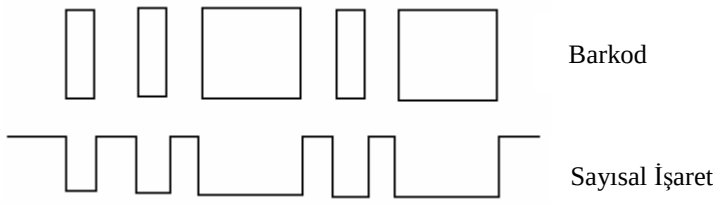
Not: Bar kodlamada verilen örnekler, anlamayı kolaylaştırmak amacıyla asitleştirilmiştir. Bu nedenle gösterilen örnekler gerçek durumu temsil etmeyebilir.

Şekil 3.4. Onlu sayıların çok seviyeli kodlarla gösterimi.

Farklı kodlama çeşitleri içerisinden ülkemizde yaygın olarak kullanılan, EAN13 kodlama standardıdır. 13 basamaklı kodlama sistemi olarak da isimlendirilen bu kodlama sistemini kullanmak için Türkiye Odalar ve Borsalar Birliğinden (TOBB) izin almak gerekir. 13 basamaklı kodlama sisteminde ilk iki veya üç basamak ülke kodu olarak kullanılır. Ülkemizde 3 basamak ülke kodunu temsil etmek için kullanılmaktadır ve ülkemizde Türkiye’nin kodunu 869’dur. Ülke

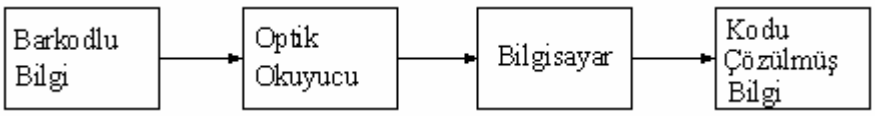
kodunu takip eden 4-6 basamak firma kodu, 3-5 basamak ise ürün kodu olarak kullanılır (firmanın özelliğine göre değişir). Bu durumda firma ürün kodu olarak 9 basamak kullanılır. 13. basamaktaki sayı kontrol için kullanılır ve 9 haneli firma-ürün kodundan üretilen sayı ile 13. basamaktaki sayı karşılaştırılarak kontrol işlemi gerçekleştirilir.

Barkodlu bilgiler barkod okuyucu ile okunur. Bir barkod tarandığında, kodda bulunan çizgiler ve boşluklar elektrik sinyallerine dönüştürülürler. Boşlukların (veya çizgiler) elektriksel olarak 0V'a, siyah çizgilerin 1V'a karşılık geldiği barkod sisteminde Şekil 3.5'deki değerler elde edilir. Elde edilen sinyalin bilgisayar tarafından anlaşılabilecek sayısal değerlere dönüştürülmesi gerekir. Bu dönüşüm işlemi kuralları önceden belirlenen bir kod çözücü ile yapılır. Barkod okuma işleminde, Şekil 3.6'da prensip şeması çizilen işlemler gerçekleştirilir.



Şekil 3.5 Barkod'un elektriksel işarete çevrilmesi

Barkodun sahip olduğu değerler, genellikle barkodun altına yazılır ve barkodun herhangi bir nedenle okunamaması durumunda barkodun değerinin el ile bilgisayara girilebilmesine imkan tanımak için kullanılır.



Şekil 3.6. Barkod okuma işleminin blok şeması

Sayısal kodlar olarak ifade edilen ve BCD kodlarına örnek olarak gösterilen kodlama yöntemlerini (bar kodu hariç) bir arada ifade edersek Tablo 3.5 elde edilir.

BCD KODLAR								
Onlu Değer / Basamak Değeri		BCD Kodu	Gray Kodu	Artı 3 Kodu	5'de 2 Kodu	Tek parity Kodu	Çift parity Kodu	Aiken kodu
		8421	Basamak değeri yok	**8421**	**74210**	Parity bit **8421**	Parity bit **8421**	**2421**
0		**0000**	**0000**	**0011**	**11000**	**1 0000**	**0 0000**	**0000**
1		**0001**	**0001**	**0100**	**00011**	**0 0001**	**1 0001**	**0001**
2		**0010**	**0011**	**0101**	**00101**	**0 0010**	**1 0010**	**0010**
3		**0011**	**0010**	**0110**	**00110**	**1 0011**	**0 0011**	**0011**
4		**0100**	**0110**	**0111**	**01001**	**0 0100**	**1 0100**	**0100**
5		**0101**	**0111**	**1000**	**01010**	**1 0101**	**0 0101**	**1011**
6		**0110**	**0101**	**1001**	**01100**	**1 0110**	**0 0110**	**1100**
7		**0111**	**0100**	**1010**	**10001**	**0 0111**	**1 0111**	**1101**
8		**1000**	**1100**	**1011**	**10010**	**0 1000**	**1 1000**	**1110**
9		**1001**	**1101**	**1100**	**10100**	**1 1001**	**0 1001**	**1111**

Tablo 3.5. Sayısal kodlama sistemlerinde onlu sayıların gösterilmesi.

3.2. Alfa Sayısal Kodlar

Bilgisayarlarda sayılarla birlikte alfabedeki harfler, noktalama işaretleri ve diğer özel karakterler kullanılmaktadır. Tüm bu bilgileri kodlamak için kullanılan yöntemler, **'Alfasayısal kodlama yöntemleri'** olarak isimlendirilir.

Alfasayısal kodlar; tüm büyük ve küçük harfleri, 7 tane noktalama işaretini, 0'dan 9'a kadar 10 sayıyı ve +, /, #, %, *, vb. karakterleri içerir. Yaygın olarak kullanılan iki türlü alfasayısal kodlama yöntemi bulunmaktadır. Bunlar, ASCII (Amerikan Standart Code For Information Interchance) ve EBCDIC (Extended BCD Interchance Code) kodlarıdır. Bu kodlardan daha yaygın olarak kullanılan ASCII kodudur.

ASCII kodu; 7 bitlik bir koddur (8. bit bazı karakterlerin kontrolü için kullanılır). ASCII kodu 2^7 = 128 bilgiyi kodlama kapasitesine sahiptir. Bu kapasite rakam ve harflerle birlikte bilgisayarda kullanılan Enter, Linefeed, vb. işlemleri ifade etmek içinde yeterlidir. Tablo 3.6'de kullanılan bazı bilgilerin ASCII kodlama yönteminde ve onaltılı sistemde kodlanmış karşılıkları görülmektedir.

İkili sayı formundaki ASCII kodlu bir sayının anlamını bulmak için, ikili sayının karşılığı olan onaltılı sayı bulunur. Onaltılı sayının ifade ettiği karakter tespit edilirse kodlanmış sayı anlamlı hale gelir.

Örnek 18 : İkili sayı sistemindeki aşağıdaki mesaj ASCII kodunda kodlanmıştır. Bu mesajın anlamı nedir?

1001000 1000101 100110 1010000

Her 7 bitlik kodun onaltılı sistemdeki karşılığı bulunup yazılırsa,

48 45 4C 50

değerleri elde edilir. Bu değerlerin temsil ettikleri bilgiler tablodan bulunup eklenirse;

48=H, 45=E, 4C=L, 50=P

eşitlikleri bulunur ve sonuç olarak, HELP bilgisine ulaşılır.

Örnek 19: 'BASIC' proglamlama dilinde program yazan bir programcı 'NEXT' boşluk 'I' yazmış olsun. Bu durumda ASCII kodunda bellekte saklanacak bilgi nedir?

Karakter	Onaltılı	İkili
N →	4E	0100 0110
E →	45	0100 0101
X →	58	0101 1000
T →	54	0101 0100
Space →	20	0010 0000
I →	49	0100 1001

Karakter	7-Bit ASCII	Sekizli	Onaltılı	Karakter	7-Bit ASCII	Sekizli	Onaltılı
A	100 0001	101	41	Y	101 1001	131	59
B	100 0010	102	42	Z	101 1010	132	5A
C	100 0011	103	43	0	011 0000	060	30
D	100 0100	104	44	1	011 0001	061	31
E	100 0101	105	45	2	011 0010	062	32
F	100 0110	106	46	3	011 0011	063	33
G	100 0111	107	47	4	011 0100	064	34
H	100 1000	110	48	5	011 0101	065	35
I	100 1001	111	49	6	011 0110	066	36
J	100 1010	112	4A	7	011 0111	067	37
K	100 1011	113	4B	8	011 1000	070	38
L	100 1100	114	4C	9	011 1001	071	39
M	100 1101	115	4D	boşluk	010 0000	040	20
N	100 1110	116	4E	.	010 1110	056	2E
O	100 1111	117	4F	(	010 1000	050	28
P	101 0000	120	50	+	010 1011	053	2B
Q	101 0001	121	51	$	010 0100	044	24
R	101 0010	122	52	*	010 1010	052	2A
S	101 0011	123	53	)	010 1001	051	29
T	101 0100	124	54	-	010 1101	055	2D
U	101 0101	125	55	/	010 1111	057	2F
V	101 0110	126	56	,	010 1100	054	2C
W	101 0111	127	57	=	011 1101	075	3D
X	101 1000	130	58	RETURN	000 1101	015	0D
				LINEFEED	000 1010	012	0A

Tablo 3.6. ASCII kodlu karakterlerin gösterilişi.

Her bir karakterin ASCII kodunda karşılığı bulunur. Onaltılı sistemde ifade edilen değerlerin ikili sistemdeki karşılığı yazılırsa, programcının girdiği karakterlerin karşılığı olan ikili sayılar bulunur.

Karakter	Onaltılı Tabanda	İkili Tabanda	Karakter	Onaltılı Tabanda	İkili Tabanda
Boşluk	40	0100 0000	H	C8	1100 1000
Ç	4A	0100 1010	I	C9	1100 1001
.	4B	0100 1011	J	D1	1101 0001
<	4C	0100 1100	K	D2	1101 0010
(	4D	0100 1101	L	D3	1101 0011
+	4E	0100 1110	M	D4	1101 0100
Ğ	5A	0101 1010	N	D5	1101 0101
İ	5B	0101 1011	O	D6	1101 0110
*	5C	0101 1100	P	D7	1101 0111
)	5D	0101 1101	Q	D8	1101 1000
;	5E	0101 1110	R	D9	1101 1001
─	60	0110 0000	S	E2	1110 0010
/	61	0110 0001	T	E3	1110 0011
,	6B	0110 1011	U	E4	1110 0100
>	6E	0110 1110	V	E5	1110 0101
?	6F	0110 1111	W	E6	1110 0110
:	7A	0111 1010	X	E7	1110 0111
Ö	7B	0111 1011	Y	E8	1110 1000
Ş	7C	0111 1100	Z	E9	1110 1001
'	7D	0111 1101	0	F0	1111 0000
=	7E	0111 1110	1	F1	1111 0001
"	7F	0111 1111	2	F2	1111 0010
A	C1	1100 0001	3	F3	1111 0011
B	C2	1100 0010	4	F4	1111 0100
C	C3	1100 0011	5	F5	1111 0101
D	C4	1100 0100	6	F6	1111 0110
E	C5	1100 0101	7	F7	1111 0111
F	C6	1100 0110	8	F8	1111 1000
G	C7	1100 0111	9	F9	1111 1001

Tablo 3.7. EBCDIC kodunda bilgilerin kodlanması.

Örnek 20: 'DIGITAL' kelimesinin elde edilebilmesi için yazılması gerekli ASCII kodlu bilgiyi yazalım.

Herbir karakterin karşılığı olan bilgilerin yazılması ile;

D = 100 0100
I = 100 1001
G = 100 0111
I = 100 1001
T = 100 0100
A = 100 0001
L = 100 1100

bilgileri elde edilir.

EBCDIC Kodu, IBM tarafından geliştirilen ancak diğer üreticiler tarafından benimsenmediği için yaygın olarak kullanılmayan, 8 bitlik alfasayısal bir kodlama sistemidir. Her bir bilginin farklı şekilde onaltılı ve ikili sistemde temsil edildiği bu kodlama sisteminde, karakterlerin eşitlikleri Tablo 3.3'de toplu olarak görülmektedir. Bu kodlama sisteminde, bilgilerin kodlanmasında kullanılan mantık ASCII kodunun aynısıdır.

Tekrarlama ve Çalışma Soruları

1. Kodlamayı tarif ediniz.
2. Kodlama çeşitleri en genel şekilde kaç gruba ayrılabilir?
3. Kodlama işleminin üstünlüklerini sıralayınız.
4. Sayısal kodları tanımlayınız.
5. Sayısal kodların yaygın olarak kullanılanlarını sıralayınız.
6. BCD kodunu tanımlayınız.
7. $(346)_{10}$ ve $(2412)_{10}$ sayılarını BCD kodlu olarak ifade ediniz.
8. $(100101010110)_{BCD}$ ve $(011010000011)_{BCD}$ sayılarının onlu sistemdeki karşılıklarını bulunuz.
9. Gray kodu tanımlayarak, Gray kodun kullanıldığı yerleri açıklayınız.
10. $(1001001100)_2$ ve $(1110000011)_2$ sayılarını Gray koduna çeviriniz.
11. $(101110)_{Gray}$ ve $(10111011)_{Gray}$ Gray kodlu sayıları ikili sisteme çeviriniz.
12. $(72)_{10}=(?)_{+3}$ ve $(00110111)_{+3}=(?)_{10}$ dönüşümlerini yapınız.
13. 5'te 2 kodunun üstünlüğü nedir?
14. $(9)_{10}$ ve $(12)_{10}$ sayılarının 5'te 2 kodundaki karşılıklarını yazınız.
15. $(10100\ 00110)_{5'te\ 2}$ ve $(10010\ 11000)_{5'te\ 2}$ sayılarının onlu sayı sistemindeki karşılıklarını bulunuz.
16. Eşitlik bitini tanımlayarak, eşitlik biti çeşitlerini yazınız.
17. Eşitlik bitinin kullanıldığı yerleri ve faydalarını sıralayınız.
18. (1011100) ve (0111000) sayılarına çift eşitlik yöntemine göre eşitlik biti ekleyiniz.

19. (1011100) ve (0111000) sayılarına tek eşitlik yöntemine göre eşitlik biti ekleyiniz.
20. Bar kodunu tarif ediniz. Bar kodların oluşturulmasında kullanılan yöntemleri özetleyiniz.
21. 39 ve 128 bar kodlarını karşılaştırınız.
22. Alfasayısal kodları özetleyiniz.
23. ASCII ve EBCDIC kodları arasındaki farkları özetleyiniz.
24. 1010011 1001111 1001110 ikili sistemdeki sayılarla ifade edilen ve ASCII kodlanmış bilgi hangi kelimeyi temsil etmektedir?
25. ‘BAŞLA’ bilgisinin bilgisayara girilmesi ile oluşan ASCII kodlu ikili sistemdeki sayı dizisini bulunuz.

BÖLÜM 4

BOOLEAN KURALLARI VE LOJİK İFADELERİN SADELEŞTİRİLMESİ

(BOOLEAN ALGEBRA AND LOGIC SIMPLIFICATION)

Giriş

Sayısal elektroniğin temeli hipoteze dayanmaktadır. 'Doğru' veya 'Yanlış' olduğu konusunda karar verilebilen fikirler **'hipotez'** olarak tanımlanır. Hipotez aynı anda hem doğru, hem yanlış olamaz, yalnızca 'doğru' veya yalnızca 'yanlış' olarak değerlendirilebilirler.

Örneğin;

'Su 0^0C'nin altında donar'

'Güneş dünya etrafında döner'

fikirlerinden birincisi 'doğru', ikincisi ise 'yanlış' olarak değerlendirilebilir ve bu nedenle bu fikirler hipotez olarak kabul edilir.

'Sağlıksız beslenen insanlar hastalanırlar' fikrinde; insanların hastalanmalarına tek etken sağlıksız beslenme olmadığından (genetik, çevre şartları, vb.) fikrin hipotez olarak değerlendirilmesi mümkün değildir. 'Doğru' veya 'yanlış' olarak tanımlanamayan fikirler hipotez olarak tanımlanmaz ve fikir olarak ifade edilirler.

Hipotez olarak ifade edilen fikirler basit veya karmaşık olabilir. Daha basit hipotezlere parçalanamayan hipotez **'basit hipotez'**, basit hipotezlerden oluşturulmuş hipotez ise **'karmaşık hipotez'** olarak isimlendirilir.

Basit veya karmaşık hipotezlerin matematiksel ifadeler şeklinde ifade edilmesi ile 'Boolean Matematiğinin' temeli oluşur. Boolean kuralları veya Boolean matematiği olarak isimlendirilen matematiğin temeli, Aristotle'nin mantığının matematiksel notasyonlara uygulanması sonucu atıldı. Matematikçi George Boole (1815-1864) tarafından 1854 yılında ortaya atılan fikirlerin Peono, Whitehead, Bertrand Russell ve diğer matematikçiler tarafından geliştirilmesi ile, sayısal elektroniğin oluşumunu sağlayan Boolean matematiği geliştirildi.

Boolean matematiğinin gelişim süreci içerisinde, kullanılan notasyon ve sembollerde değişimler oluştu ve Boolean matematiğini oluşturan kurallar (postulates), E.V. Huntington tarafından 1904 yılında basıldı. Oluşturulan kuralların Claude E. Shannon tarafından elektronik elemanlara uygulanması sonucu, Boolean kurallarının anahtarlamalı sistemlerde kullanılabileceği açıklandı (1938).

Elektronik devrelerin bir kısmını oluşturan anahtarlamalı sistemlerin temelini oluşturduğu Lojik devreler, ikili moda göre çalışır ve giriş /çıkışları '0' veya '1' değerlerinden birisini alabilir. Böyle bir devre, cebirsel veya grafiksel yöntemlerden birisi kullanılarak sadeleştirilebilir. Lojik devrelerin sadeleştirilmesinde kullanılan yöntemlerden birisi, temel prensiplere göre doğruluğu kabul edilmiş işlemler, eşitlikler ve kanunlardan oluşan Boolean kurallarıdır. Diğer bir deyişle; **'Boolean kuralları'**, dijital devrelerin sahip oldukları girişlerin etkilerini açıklamak ve verilen bir lojik eşitliği gerçekleştirilecek en iyi devreyi belirlemek amacıyla lojik ifadeleri sadeleştirmede kullanılabilir.

Lojik devrelerin işlevini açıklamak amacıyla yazılan eşitliklerde, giriş değişkenleri olarak alfabenin başındaki harfler (A, B, C, D,..) çıkış değişkenleri olarak alfabenin sonundaki harfler (I, X, Y, W, Z,..) kullanılır (Şekil 4.1).



Şekil 4.1. Lojik devrelerde giriş / çıkış değişkenlerinin belirlenmesi.

4.1. Önemli Boolean Kuralları

1850'li yıllarda George Boole tarafından geliştirilen Boolean matematiği kuralları, 'VE', 'VEYA' ve 'DEĞİL' temel mantıksal işlemlerinden oluşan sembolik bir sistemdir. George Boole, temel mantıksal işlemleri kullanılarak toplama, çıkarma, çarpma, bölme ve karşılaştırma işlemleri yapabiliyordu. Bu işlemler temelde ikili işlemlerdi ve bu nedenle birbirinin tersi olan iki durumla açıklanabiliyordu:

Doğru – Yanlış, Evet – Hayır, Açık – Kapalı, ‘1’ – ‘0’, vb.

Başlangıçta pratik olarak görülmeyen sistem, daha sonraları yaygın olarak kullanılmaya başlandı ve ‘Boolean Matematiği / Cebiri’ veya ‘Boolean Kuralları’ olarak isimlendirildi. İkili sayı sistemi ile birleştirilen Boolean kuralları, sayısal elektronik devrelerin (buna bağlı olarak Bilgisayarların) temelini oluşturdu.

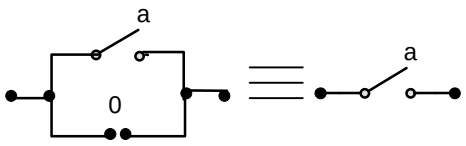
Her sistemin kendi içerisinde kuralları olması gibi, Boolean matematiğinde de kendi içerisinde kuralları vardır. Sadeleştirme işlemini gerçekleştirmede kullanılan bu kuralları genel hatları ile inceleyelim.

1. Temel Özellikler :

Boolean cebrindeki temel özellikler : etkisiz eleman, birim eleman, yutan eleman, ters eleman şeklinde sıralanabilir.

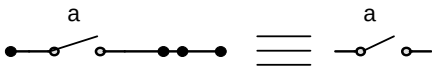
1a : Toplamada Etkisiz Eleman (0) :

A + 0 = A
0 + 0 = 0
1 + 0 = 1



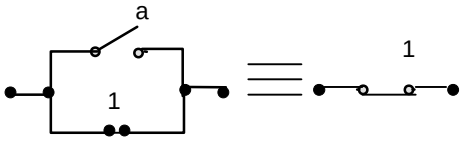
1b : Çarpmada Etkisiz Eleman (1) :

A . 1 = A
0 . 1 = 0
1 . 1 = 1



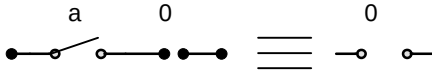
1c : Toplamada Birim Eleman :

A + 1 = 1
0 + 1 = 1
1 + 1 = 1



1d : Çarpmada Yutan eleman:

A . 0 = 0
0 . 0 = 0
1 . 0 = 0



1e : Ters eleman :

Bir değişken ‘0’ ise değili (barı, tersi vb.) ‘1’, değişken ‘1’ ise değili ‘0’ olarak alınır. Bir değişkenin değili, değişken üzerine konan çizgi veya kesme işareti ile belirtilir.

A = 0 => A' = 1 , A = 1 => A' = 0

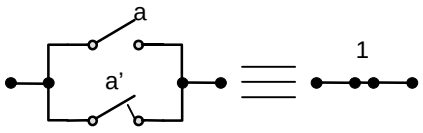
Bir değişkenin değilinin değili (tersinin tersi) kendisine eşittir :

(A" =A).

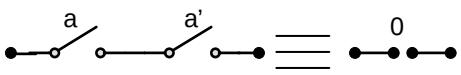
1f : Toplama ve Çarpma İşlemleri :

Boolean matematiğinde, ‘VEYA’ işlemi toplama (+) ve ‘VE’ işlemi çarpma (.) işlemlerine karşılık gelir. Boolean matematiğinde geçerli olan toplama ve çarpma işlemleri aşağıdaki şekilde özetlenebilir.

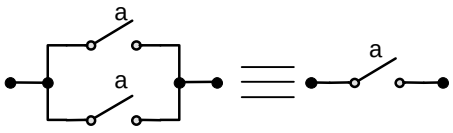
A +A' = 1
0 + 1 = 1
1 + 0 = 1



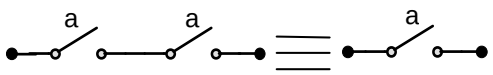
A.A' = 0
0 . 1 = 0
1 . 0 = 0



A + A = A
0 + 0 = 0
1 + 1 = 1



A . A = A
0 . 0 = 0
1 . 1 = 1



2- Sabit kuvvetlilik :

Boolean matematiğinde normal aritmetik işlemlerdeki toplama ve çarpma işlemlerinden farklı olarak kullanılan kurallardan birisi; sabit kuvvetliliktir.

a) A + A = A (A+A+A+.....+A = A) , b) A . A = A (A.A.A.A.....A = A)

3- Değişim Kanunu (Comutative Law) :

Toplama ve Çarpma işlemlerinde geçerli olan değişim kanunu aynı şekli ile Boolean matematiğinde de geçerlidir.

a) A + B = B + A b) A . B = B . A

4- Birleşme Kanunu (Assosiative Law) :

Toplama ve Çarpma işlemlerinde geçerli olan değişim kanunu aynı şekli ile Boolean matematiğinde de geçerlidir.

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

5- Dağılma Kanunu (Distributive Law) :

Gerek ‘toplamanın çarpma’ üzerindeki gereksede ‘çarpmanın toplama’ üzerindeki dağılma özellikleri olarak tanımlanan kanunlar, aynı şekli ile boolean matematiğinde kullanılmaktadır.

6- Yutma Kanunu (Absorbation Law) :

Yalnızca Boolean cebirinde geçerli olan kurallardan bir diğeri; yutma kanunudur.

a) A+ A . B = A b) A . (A+B) = A

7- Basitleştirme Kanunu (Minimisation Law) :

Toplama ve Çarpma işlemlerinde boolean matematiğinde geçerli olan bir diger kural; basitleştirme ve sadeleştirme kuralıdır.

a) $A + A' . B = A + B$ b) $A .(A'+B) = A . B$

8- De Morgan Kanunları :

'VEYADEĞİL' ve 'VEDEĞİL' işlemlerinden faydalanarak uyğulanan ve lojik işlemlerde kolaylıklar sağlayan kurallar, **'De Morgan Kanunları / Kuralları'** olarak isimlendirilir

a) $\overline{A.B} = A'+B'$ b) $\overline{A+B} = A'.B'$

4.2. Boolean Kurallarını Kullanarak Lojik Eşitliklerin Sadeleştirilmesi

Karmaşık lojik ifadeler, yukarıda özetlenen boolean matematiğindeki kurallardan faydalanarak sadeleştirilebilirler (basitleştirilebilirler). Sadeleştirilen lojik ifadelerden oluşturulacak elektronik devreler, hem daha basit hem de daha ucuz olarak gerçekleştirilebilirler.

Boolean kurallarının lojik ifadelerin basitleştirilmesinde kullanılmasına örnek olması bakımından yukarıdaki bazı eşitlikleri ispatlayalım ve fonksiyon basitleştirme işlemleri yapalım.

İşlemlerde değişkenlerin değilini ifade etmek için ' ' ' işareti kullanılırken, birleşik ifadelerin değili için '—' sembolü kullanılacaktır.

Örnek 1: 5 b'nin ispatını yapalım.

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

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

$$= .1\ A + B.C = A + B.C$$

Örnek 2: 6 a'nın ispatını yapalım.

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

Örnek 3: 6 b'nin ispatını yapalım.

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

Örnek 4: 7 a'nın ispatını yapalım.

$$A + A'.B = \overline{\overline{A + A'.B}} = \overline{A'.\overline{(A'.B)}} = \overline{A'.(\underset{A}{\underline{A}}"+B')}$$

$$= \overline{A'.(A+B')} = \overline{\underset{0}{\underline{A'.A}} + A'.B'} = \overline{\overline{A'.B'}} = \overline{\overline{A+B}} = A+B$$

Örnek 5: 7 b'nin ispatını yapalım.

$$A.(A'+B) = \underset{0}{\underline{A.A'}} + A.B = 0 + A.B = A.B$$

Örnek 6 : (A+B+C) ' = A'.B'.C' 'nin ispatını yapalım.

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

(B+C=X) olarak varsayalım.

Örnek 7 : F = A'.B + A + A.B ifadesini sadeleştirelim.

$$A'.B + A + A.B = \underset{1}{\underline{A.(1+B)}} + A'.B = \overline{\overline{A + A'.B}} = \overline{A'.\overline{(A'.B)}} = \overline{A'.(\underset{A}{\underline{A}}"+B')}$$

$$= \overline{A'.\underset{A}{\underline{A}}" + A'.B'} = \overline{\underset{0}{\underline{A'.A}} + A'.B'}$$

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

Sadeleştirme işlemi, 'B' parantezine alınarak yapılırsa;

$$A'B + A + AB = B\ .\ (A+A') + A = B\ .\ 1 + A = A+B$$

olarak bulunur. Bu sonuç, aynı sonuca farklı şekillerde ulaşılacağına iyi bir örnektir.

Örnek 8: A'.B'.C + A'.B.C + A.B' ifadesini sadeleştirelim.

$$A'.B'.C + A'.B.C + A.B' = A'.C.\underset{1}{\underline{(B+B')}} + A.B' = A'.C + A.B'$$

Örnek 9: A.B + A'.C + B.C ifadesini sadeleştirelim

$$A.B + A'.C + B.C = A.B + A'.C + B.C.\underline{(A+A')} = \underline{A.B} + \underline{A'.C} + \underline{A.B}.C + \underline{A'}.B.\underline{C}$$

(A*1=A ve A+A'=1 olduğundan sonuç değişmez)

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

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

Örnek 10: A'B'C' + A'B'C + ABC' + AB'C' ifadesini sadeleştirelim

$$A'B'C' + A'B'C + ABC' + AB'C' = A'B'.(C+C') + AC'.(B+B')$$
$$1 \qquad 1$$
$$= A'B' + AC'$$

4.3. Doğruluk Tablosu

Lojik devrelerde, giriş değişkenlerinin alabilecekleri sayısal değerleri (kombinasyonları) ve sayısal değerlere göre çıkışların durumunu gösteren tablolara, **'doğruluk tablosu'** denir. Doğruluk tabloları oluşturulurken, giriş değişken sayısına göre durum ifadesi ortaya çıkar. 'n' tane değişken için 2^n değişik durum oluşur. Örneğin; 2 değişkenli bir ifade için $2^2 = 4$ değişik durum, 3 değişkenli bir ifade için $2^3 = 8$ değişik durum elde edilir.

Örnek 11 : Giriş değişkenlerinin A ve B olduğu bir sistemde A+B işlemi gerçekleştirildiğine göre; A ve B' nin alacağı değerler ile çıkışta oluşacak değerleri tablo halinde gösterelim.

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

Örnek 12 : A ve B giriş değişkenlerine sahip lojik devrenin çıkışı f=A.B eşitliği ile gösterilmektedir. Giriş ve çıkışta oluşabilecek değerleri tablo halinde gösterelim.

A	B	A * B
0	0	0
0	1	0
1	0	0
1	1	1

Örnek 13 : Giriş değişkenleri olarak isimlendirilen A ve B değişkenlerinin alacağı sayısal değerleri ve bu değişkenlerle oluşturulabilecek bütün işlemleri doğruluk tablosu ile gösterelim.

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	1	1

Örnek 14 : $\overline{A+B} = A' \cdot B'$ De Morgan teoremini doğruluk tablosu ile ispatlayalım.

Eşitliğin iki tarafındaki işlemleri temsil eden sütunların aynı değerlere sahip olması, doğruluk tablosu yardımı ile eşitliğin doğru olduğunu ispatlar.

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 15 : $\overline{A \cdot B} = A'+B'$ eşitliğini doğruluk tablosu ile ispatlayalım.

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

Örnek 16 : F = A + A . B = A olduğunu ispatlayalım.

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

Örnek 17 : F = A . (B+C) = (A . B) + (A . C) eşitliğinin doğru olduğunu doğruluk tablosunda değişken değerlerini kullanarak ispatlayalım.

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

Örnek 18 : F = A + A . B + A'. C + C' . D = A + C + D eşitliğini doğruluk tablosu kullanarak ispatlayalım.

A	A'	B	B'	C	C'	D	D'	AB	A'C	C'D	A+AB+A'C+C'D	A+C+D
0	1	0	1	0	1	0	1	0	0	0	0	0
0	1	0	1	0	1	1	0	0	0	1	1	1
0	1	0	1	1	0	0	1	0	1	0	1	1
0	1	0	1	1	0	1	0	0	1	0	1	1
0	1	1	0	0	1	0	1	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	1	1	1
0	1	1	0	1	0	0	1	0	1	0	1	1
0	1	1	0	1	0	1	0	0	1	0	1	1
1	0	0	1	0	1	0	1	0	0	0	1	1
1	0	0	1	0	1	1	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	0	0	1	1
1	0	0	1	1	0	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1	1	0	0	1	1
1	0	1	0	0	1	1	0	1	0	1	1	1
1	0	1	0	1	0	0	1	1	0	0	1	1
1	0	1	0	1	0	1	0	1	0	0	1	1

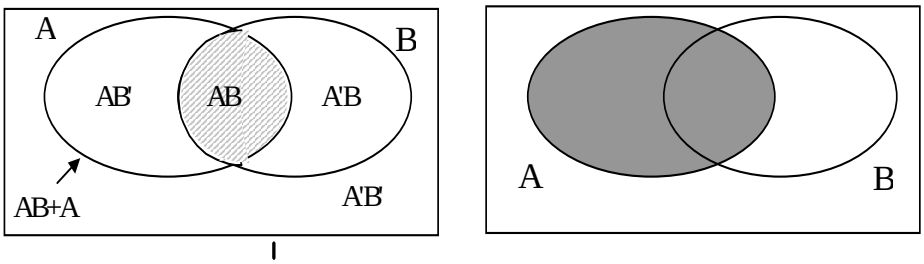
⇧ ⇧

En son iki sütundaki değerlerin eşit olması ifadelerin birbirlerine eşit olduğunu gösterir.

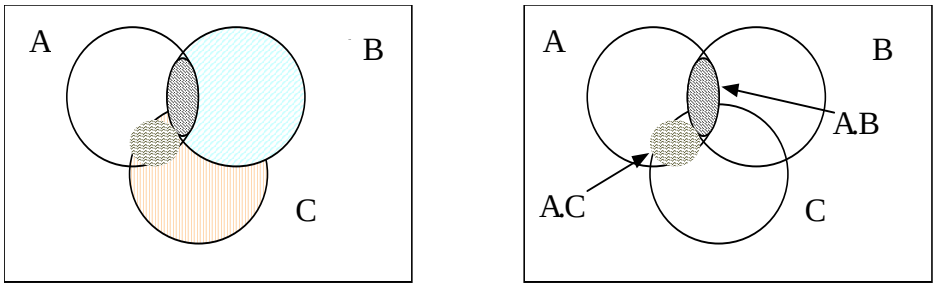
4.4. VENN Diyagramı

Venn diyagramı, Boolean değişkenleri arasındaki ilişkileri şekillerle göstermek amacıyla kullanılan yöntemdir. Diğer bir deyişle; kümeler cebrinin grafik olarak gösterimi olan Venn Diyagramları, lojik ifadeleri görsel olarak ifade etmeye yarayan bir araçtır. Bu yöntem, Şekil 4.1'deki gibi dairelerin kullanıldığı ve her dairenin bir değişkeni temsil ettiği gösterim şeklidir. Bir dairenin içerisindeki tüm noktalar değişkenin kendisini gösterirken, dairenin dışındaki tüm noktalar ise ‘değişkenin değili’ olarak ifade edilir. Örneğin A=1 ve buna bağlı olarak A'=0 olarak düşünürsek; A’yı temsil eden dairenin içindeki noktalar 1’i, dışındaki noktalar 0’ı temsil eder. Bu kabullere göre birbirini kesen iki daire ve dairelerin dışındaki noktalar Şekil 4.2’deki gibi ifade edilir.

A ve B olarak isimlendirilen iki kümenin ortak elemanları kesişim (A∩B) kümesini ve iki kümenin elemanlarının tamamı bileşim (A∪B) kümesini oluşturur. A kümesinin elemanı olmayan elemanlardan oluşan küme $A^{'}$ olarak isimlendirilen kümeyi oluşturur.



Şekil 4.2. Venn Diyagramında kümelerin oluşturulması ve AB +A = A eşitliğinin Venn diyagramı ile gösterimi.



A(B+C) AB + AC

Şekil 4.3. Dağılma kanununun Venn diyagramı ile gösterilmesi.

Venn diyagramları, Boolean cebrindeki sadeleştirmeleri veya teoremlerin geçerliliğini göstermek için kullanılabilir. Şekil 4.2, AB ve A değerlerini temsil eden, sonuçta A+AB = A eşitliğini ifade eden bölgeleri gösterir. Şeklin incelenmesinden; AB+A eşitliğini ifade eden bölge ile A' yı ifade eden bölgenin aynı olduğu görülür.

Şekil 4.3, A.(B+C) = (A.B) + (A.C) dağılma kanununun Venn diyagramı ile gösterimini göstermektedir. Bu diyagramda, birbirini kesme bölgeleri bulunan ve A, B, C ile ifade edilen üç daire bulunmaktadır. Üç değişkenli Venn diyagramı ile sekiz farklı alanı tanımlamak mümkündür. Bu örnek ile A, B, C olarak ifade edilen bölgelerin kesişme noktaları ile 'AB+AC' olarak tanımlanan bölgelerin aynı olduğu gösterilebilir.

4.5. Temel Açılımlar ve Standart İfadeler

Daha önceki konularda bahsedildiği üzere, bir binary değişkeni, ya kendi normal formu olan A olarak veya değili olan A' formu ile ifade edilebilir. Bu formlarla ifade edilebilen değişkenler fonksiyon halini aldığı zaman; '**canonical form**' (kanun-kaide) olarak adlandırılan '**minterm'** (çarpımların toplamı) veya '**maxterm'** (toplamların çarpımı) modellerinden biri ile gösterilirler.

Değişken			**Mintermler**		**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'BC'	m_2	A+B'+C	M_2
0	1	1	A'BC	m_3	A+B'+C'	M_3
1	0	0	AB'C'	m_4	A'+B+C	M_4
1	0	1	AB'C	m_5	A'+B+C'	M_5
1	1	0	ABC'	m_6	A'+B'+C	M_6
1	1	1	ABC	m_7	A'+B'+C'	M_7

Tablo 4.1. Üç değişkenli bir sistemde oluşabilecek minterm ve maxterm terimleri.

Bir boolean ifadede bulunan değişkenlerin sahip olduğu veya oluşturabileceği kombinasyonların 'VE' (çarpım) işlemi sonucunda 1 olacak şekilde uyarlanmasına (değişkenin değeri 1 ise olduğu gibi alınıp, 0 ise değili ile ifade edilerek), '**minterm'** denir. Aynı yolla, değişkenlerin kombinasyonlarının 'VEYA' (toplama) işlemi sonucunda 0 değerini almasını sağlayacak şekilde değişkenlerin şekillendirilmesine '**maxterm'** denir.

Tablo 4.1'de üç değişkenli bir sistemde değişkenlerin oluşturabileceği kombinasyonlar ve bu kombinasyonlarda elde edilecek minterm ve maxterm terimleri verilmiştir.

Bir Boolean fonksiyonu Tablo 4.1'deki doğruluk tablosundan belirli kombinasyonların seçilmesi, seçilen kombinasyonların sonuç olacak şekilde formlandırılması ve formlandırılan kombinasyonların toplanması ('VEYA' işlemine tabi tutulması) şeklinde tanımlanabilir.

Örnek 19 : Tablo 4.2'deki doğruluk tablosunda f_1 ve f_2 fonksiyonlarını minterm formu ile tanımlayalım.

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

Tablo 4.2. Fonksiyonlardaki minterm oluşturacak kombinasyonların seçilmesi.

'f_1' fonksiyonunda '1' olarak tanımlanan kombinasyonlardaki değişken değerleri; 001, 100 ve 111 olduğundan, bu kombinasyonları temsil eden değişkenler fonksiyon olarak,

$f_1 = m_1 + m_4 + m_7$ =A'B'C + AB'C' + ABC

şeklinde ifade edilir.

Aynı şekilde f_2 fonksiyonu;

$f_2 = m_3 + m_5 + m_6 + m_7$ = A'BC + AB'C + ABC' + ABC

olarak tanımlanır.

Bu örnek bir Boolean fonksiyonunun mintermlerin toplanması şeklinde tanımlanabileceği özelliğini gösterir. Bu örnekte çıkıştaki '1' değerleri referans olarak alınmıştır.

'f_1' fonksiyonunda '0' olarak tanımlanan kombinasyonlar referans olarak alınır ve kombinasyonlardaki değişkenlerin toplamı '0' olacak şekilde kombinasyonlar formlandırılırsa, Boolean cebrinin diğer bir özelliği ortaya çıkar. Bu özellik; Boolean

fonksiyonunun maxtermlerin çarpımı (AND işlemine tabi tutulması) şeklinde ifade edilebilirliği özelliğidir.

Bu özelliği gösterecek şekilde F_1 ve F_2 fonksiyonları yazılabilir. Bu durumda F_1 fonksiyonu;

$$F_1 = M_0 . M_2 . M_3 . M_5 . M_6$$

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

şeklinde, F_2 fonksiyonu ise;

$$F_2 = M_0 . M_1 . M_2 . M_4$$

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

şeklinde tanımlanır.

4.6. Mintermlerin Toplamı ve Maxtermlerin Çarpımı İfadelerinin Üretilmesi

Sadeleştirilmiş olarak verilen bir fonksiyonu, mintermlerin toplamı veya maxtermlerin çarpımı şeklinde ifade etmek belirli uygulamalar için uygun olabilir. Sadeleştirilmiş olarak verilen bir fonksiyonu mintermlerin toplamı şeklinde ifade etmek için, herbir minterm ifadesi bütün değişkenleri içerecek şekilde genişletilir. Herhangi bir kombinasyonda bulunmayan değişkenleri eklemek için kombinasyon (x + x') terimi ile 'VE' işlemine tabi tutulur (x, kombinasyonda bulunmayan değişkenlerden herbirini ifade eder).

Aşağıdaki örnekler, genişletme işlemini daha net olarak anlamaya yardım edecektir.

Örnek 20 : f=A+B'C ifadesini mintermlerin toplamı şeklinde ifade edelim.

İlk kombinasyonda (minterm ifadesinde) yalnızca A bulunduğundan B ve C değişkenleri ifadeye eklenmelidir. İlk minterme 'B' değişkeninin eklenmesiyle;

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

ifadesi elde edilir. Ancak hala 'C' değişkeni kombinasyonda yoktur ve 'C' değişkeninin eklenmesiyle;

$$A = AB(C+C') + AB'(C+C') = ABC + ABC' + AB'C + AB'C'$$

eşitliği bulunur.

İkinci kombinasyon B'C olduğundan, A değişkeninin eklenmesiyle;

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

ifadesi elde edilir. İki kombinasyonun birleştirilmesi sonucunda;

$$f = ABC+ABC'+AB'C+AB'C'+AB'C+A'B'C$$

eşitliği oluşur. AB'C kombinasyonu iki kere kullanıldığından ve x + x = x olduğundan verilen fonksiyon ;

$$f = A+B'C = A'B'C+ AB'C'+ AB'C+ ABC'+ ABC = m_1+m_4+m_5+m_6+m_7$$

şeklini alır.

Mintermlerin toplamı olarak ifade edilmek istenen bir fonksiyon '$\sum$' sembolü ile belirtilir. Bu sembolün kullanıldığı durumlarda fonksiyon;

$$F_{(A,B,C)} = \sum (1,4,5,6,7)$$

şeklinde ifade edilir.

Sadeleştirilmiş olarak verilen bir ifadeyi, tüm değişkenleri içeren maxterm kombinasyonların çarpımı şeklinde ifade etmek için yukarıdaki işlem basamaklarının aynısı kullanılır.

Örnek 21: F = (A'+B).(A+C).(B+C) fonksiyonunu makstermlerin çarpımı şeklinde ifade edelim.

Verilen eşitlikte her bir kombinasyonda bir tane değişken eksik olduğundan ;

A'+B = A' + B + CC' = (A'+B+C) . (A'+B+C')
A+C = A + C + BB' = (A+B+C) . (A+B'+C)
B+C = B + C + AA' = (A+B+C) . (A'+B+C)

eşitlikleri elde edilir.

Bu ifadeler bir araya getirilirse (1 kereden fazla yazılanları 1 kere yazarak);

$$F = (A+B+C) . (A+B'+C) . (A'+B+C) . (A'+B+C')$$
$$= M_0.M_2.M_4.M_5$$

fonksiyonu oluşur.

'$\prod$' sembolünün maxtermlerin çarpımı için kullanıldığını belirterek verilen fonksiyonu;

$$F_{(A,B,C)} = \prod (0,2,4,5)$$

şeklinde ifade edebiliriz.

4.7. Maxterm ve Minterm İfadelerin Birbirlerine Dönüştürülmesi

'Minterm' ve 'Maxterm' ifadelerin elde ediliş şekilleri göz önünde tutulursa, minterm ve maxterm ifadelerin birbirlerinin tersi (komplementi-tümleyeni) olduğu bulunabilir. Çünkü

mintermleri oluşturmak için fonksiyonlardaki '1' değerleri alınırken, maxtermleri oluşturmak için '0' değerleri alınmaktadır. Örnek olarak;

$$f_{(A,B,C)} = \sum(1,4,5,6,7)$$
$$= m_1 + m_4 + m_5 + m_6 + m_7$$

fonksiyonu incelenirse; bu fonksiyonun tümleyeni;

$$f'_{(A,B,C)} = \sum(0,2,3)$$
$$= m_0+m_2+m_3$$

olarak oluşur. Bu fonksiyondan f''in kendi karşılığı De Morgan kurallarını kullanarak elde edelirse,

$$f' = m_0+m_2+m_3 \Rightarrow f = \overline{m_0+m_2+m_3} = m_0'.m_2'.m_3'$$
$$= M_0.M_2.M_3$$
$$= \prod(0,2,3)$$

ifadesi bulunur. $m_i' = M_i$ olduğu doğruluk tablosundan görülebilir. Yine doğruluk tablosundan $M_i' = m_i$ olduğu çıkarılabilir.

Bu açıklamalardan minterm ve maxterm terimleri arasındaki dönüşüm için gerekli işlemleri; '$\sum$ sembolü $\prod$ sembolü ile değiştirilirken, ifade edilen sayılarda bulunan sayılar bulunmayan sayılarla yer değiştirir' şeklinde özetlenebilir.

Yapılan özetlemeyi,

$$F(A,B,C) = \prod(0,2,4,5)$$

fonksiyonuna uygularsak, bu maxterm ifadenin mintermlerle ifadesi;

$$f(A,B,C) = \sum(1,3,6,7)$$

şeklinde oluşur.

4.8. Lojik İşlemler

'n' ikili değişkeni ile 2 üzeri 2^n sayıda fonksiyon yazmak mümkündür. Bu durumda 2 değişkenli bir sistemde $2^4 = 16$ Boolean fonksiyonu yazılabilir. X ve Y değişkenleri ile oluşturulabilecek fonksiyonlar, F_0 'dan F_{15} 'e kadar isimlendirme ile Tablo 4.3'deki gibi sıralanabilir.

Fonksiyonlarının açıklaması verilen iki değişkenli sistemin doğruluk tablosu ve fonksiyonların işlem sonuçları, Tablo 4.4'teki gibi özetlenebilir.

Boolean Fonksiyonu	İşlem Sembolü	İsim	Açıklama
$F_0 = 0$		Null	Binary sabit ∅
$F_1 = x.y$	x.y	VE	x ve y
$F_2 = x.y'$	x/y	İnhibition	x ve y değil
$F_3 = x$		Transfer	x
$F_4 = x'.y$	y/x	İnhibition	y ve x değil
$F_5 = y$		Transfer	y
$F_6 = x.y' + x'.y$	$x \oplus y$	ÖZELVEYA	x veya y fakat ikisi
$F_7 = x + y$	x + y	VEYA	x veya y
$\overline{F_8} = x + y$	$x \downarrow y$	VEYADEĞİL	VEYA'nın değili
$F_9 = x.y + x'.y'$	$x \otimes y$	Eşitlik	x eşit y
$F_{10} = y'$	y'	Değil	y'nin değili
$F_{11} = x + y'$	$x \subset y$	Örten	Eğer y = 1 ise x
$F_{12} = x'$	x	Değil	x değil
$F_{13} = x' + y$	$x \supset y$	Örten	Eğer x = 1 ise y
$\overline{F_{14}} = x.y$	$x \uparrow y$	VEDEĞİL	VE'nin değili
$F_{15} = 1$		Tanımlama	Binary sabit 1

Tablo 4.3. İki değişkenli sistemde oluşturulabilecek 16 fonksiyonun açıklamaları.

Tablo 4.4'deki işlemlerin bir kısmı daha önce karşılaşılan fonksiyonlar (+, -, *, vb.) olmakla birlikte, bir kısmı yalnızca lojikte karşılaştığımız fonksiyonlardır (sembollerdir). Doğruluk tablosunda gösterilen 16 fonksiyon 3 grupta incelenebilir.

i- İki işlem '0' veya '1' olarak bir sabit üretir: (F_0, F_{15}).

ii- Dört işlem transfer ve tümleyen işlemleridir: (F_3, F_5, F_{10}, F_{12}).

iii- Binary değerlerin kullanıldığı 10 işlem ise sekiz farklı hesaplamayı temsil eder : VE, VEYA, VEDEĞİL, VEYADEĞİL, ÖZELVEYA, Eşitlik, İnhibition, Örten (İmplication).

x y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0 0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0 1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
işlem		.	/		/		⊕	+	↓	⊗	y'	⊂	x'	⊃	↑	

Tablo 4.4. İki binary değişkeninin oluşturabileceği 16 fonksiyon için doğruluk tablosu

Yukarıda gösterilen işlemlerden 'inhibition' and 'implication' işlemleri lojikle uğraşan kişiler tarafından kullanılmakla beraber, bilgisayarlarda nadiren kullanılır. Lojikte yaygın olarak kullanılan işlemler lojik devre elemanlarının anlatılacağı lojik kapılar konusunda detayıyla incelenecektir.

Örnek 22 : Aşağıda verilen doğruluk tablosuna göre F fonksiyonunu minterm ve maxterm yöntemlerini kullanarak sadeleştirelim.

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

1	1	0	0
1	1	1	0

i- Önce minterm ifadeler yazılır ve yazılan ifadeler sadeleştirilir;

Sadeleştirme sonucunda;

$$F(A,B,C) = \sum (0,3,4,5) = A'B'C' + A'BC + AB'C' + AB'C$$

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

$$= B'.(C' + AC) + A'BC \quad \Rightarrow \quad C' + AC = C' + A$$

$$= B'C' + B'A + A'BC$$

eşitliği elde edilir.

ii- Fonksiyon maxterm şeklinde yazılırsa;

$$F(A,B,C) = \prod (1,2,6,7) = (A+B+C').(A+B'+C).(A'+B'+C').(A'+B'+C')$$

=(AA+AB'+AC+BA+BB'+BC+C'A+ C'B' +C'C).(A'A'+A'B'+A'C'+A'B'+B'B'+B'C'+A'C+B'C+CC')

A A+AB'=+A A+AC = A(1+C)=A A+BA=A 0 A+ C'A A+BC+ C'B' 0

A' A'+A'B'=A' A'+A'C'=A' A'+A'B'=A' 0 B'C'+ B'C= B'(C'+C)=B' A'+ A'C= A' A'+ B' 0

Sadeleştirilmelerin yapılması ile ;

$$F = (A + BC + B'C').(A'+B') = AB' + A'BC + B'C'$$

ifadesi elde edilir.

Tekrarlama ve Çalışma Soruları

1. Boolean cebrindeki temel özellikler nelerdir?
2. De Morgan kurallarını özetleyiniz.
3. A'B+A+AB ifadesini sadeleştiriniz.
4. Doğruluk tablosunu tanımlayıp, 3 değişkenli doğruluk tablosunu oluşturunuz.
5. De Morgan teoremlerini doğruluk tabloları kullanarak ispatlayınız.

6. Venn şeması yöntemini özetleyiniz.
7. A+AB=A işlemini Venn şeması yöntemi ile ispatlayınız.
8. f =A'B+BC+AC sadeleştirilmiş ifadeyi, genişletilmiş minterm ifadesi olarak bulunuz.
9. f=A+B'C ifadesini, genişletilmiş maxterm ifadesi olarak bulunuz.
10. f=$\sum$(3,5,6,7) ifadesinin tümleyenini bulunuz.
11. f=$\prod$(0,4,6,7) ifadesinin tümleyenini bulunuz.
12. İki binary değişkeninin oluşturabileceği fonksiyonları doğruluk tablosu şeklinde gösteriniz.
13. $F_{(A,B,C)}$ = $\sum$(0,3,5,7) minterm ifadesini doğruluk tablosunda gösteriniz ve ifadenin fonksiyonunu yazınız.
14. $F_{(A,B,C)}$ = $\sum$(1,2,6,7) minterm ifadesinin fonksiyonunu yazınız.
15. $F_{(A,B,C)}$ =$\prod$ (1,4,5) maxterm ifadesinin doğruluk tablosunu oluşturun ve maxterm ifadesini değişkenlerle yazınız.
16. $F_{(A,B,C)}$ = $\prod$ (2,5,6,7) maxterm ifadesinin fonksiyonunu yazınız.
17. $F_{(A,B,C)}$ = $\sum$ (2,4,6,7) ifadesinin fonksiyonunu yazarak sadeleştiriniz.
18. $F_{(A,B,C)}$ = $\sum$ (3,5,7) ifadesinin fonksiyonunu yazarak sadeleştiriniz.
19. $F_{(A,B,C)}$ = $\prod$ (1,4,5,6) ifadesinin fonksiyonunu yazarak sadeleştiriniz.
20. $F_{(A,B,C)}$ = $\prod$ (0,5,6) ifadesinin fonksiyonunu yazarak sadeleştiriniz.
21. $F_{(x,y)}$ = $(x^{l}+y^{l}).(x+y)$ lojik eşitliğini sadeleştiriniz.
22. $F_{(x,y)}$ = $x.y+x^{l}y+y^{l}$ lojik eşitliğini sadeleştiriniz.
23. $F_{(A,B,C,D)}$ = B.C+B.D+A.C+ A.D lojik eşitliğini sadeleştiriniz.
24. $F_{(x,y)}$ = $x.y+\bar{x}.y+\bar{x}.\bar{y}+x.\bar{y}$ lojik eşitliğini sadeleştiriniz.
25. $F_{(A,B)}$ =$A.B+\overline{A.B}$ lojik eşitliğini sadeleştiriniz.

BÖLÜM 5

LOJİK KAPILAR VE LOJİK DEVRELER

(LOGIC GATES AND LOGIC CIRCUITS)

Giriş

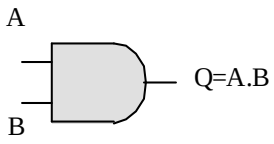
Lojik devrelerin en basit ve temel elemanı lojik kapılardır (logic gates). Lojik değişkenlerin değerlerini (gerilimleri) giriş olarak kullanan, girişten aldığı değerler üzerinde işlemler yaparak lojik eşitliğin değerine uygun değerler (gerilim) üreten elektronik devre, '**lojik kapı**' olarak isimlendirilir. Temel olarak beş farklı yapıda bulunan kapılar, basit bir sayısal elektronik devreden bilgisayara kadar cihazların temel yapı taşıdır. Flip-Flop, kaydedici, sayıcı, vb. lojik devreleri oluşturmakta kullanılan kapılar; direnç, diyot, transistör, FET, MOSFET, vb elektronik devre elemanları kullanılarak yapılırlar.

Bölüm 4'te fonksiyon şeklinde ifade edilen işlemlerden uygulama imkanı bulunanlar, lojik kapılar ile gerçekleştirilirler. Lojik kapıların sembolleri, gerçekleştirildikleri işlemler, doğruluk tabloları ve elektriki eşdeğerleri Şekil 5.1'de toplu olarak görülmektedir. Bu kapılardan yaygın olarak kullanılanlar: VE (AND), VEYA (OR), DEĞİL (NOT), VEDEĞİL (NAND), VEYADEĞİL (NOR) kapılarıdır ve bu kapılar **'temel lojik kapılar'** olarak isimlendirilir.

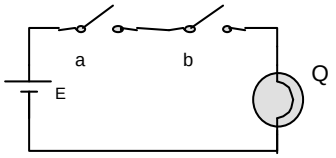
Lojik kapıların kullanılması ile oluşturulan devreler, **'lojik devreler'** olarak adlandırılır. Lojik kapılarla oluşturulan devreler, **'donanım'** olarak ta tanımlanmaktadır. Donanım terimi ile genelde elektronik, mağnetik ve mekanik devrelerin / birimlerin bir arada kullanılması durumu ifade edilmektedir.

İşlemin Adı Sembolü Yaptığı İşlem Doğruluk Tablosu Elektriksel Eşdeğeri

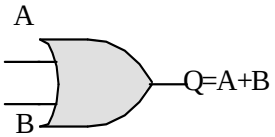
VE (AND) KAPISI



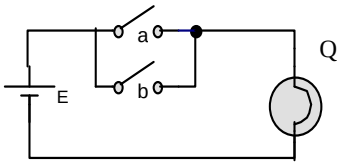
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



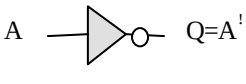
VEYA (OR) KAPISI



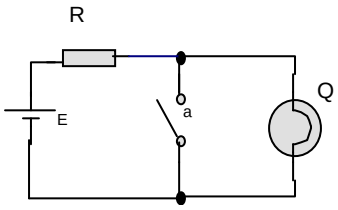
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1



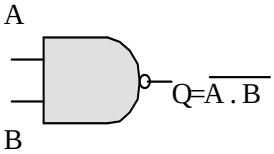
DEGİL (NOT) KAPISI



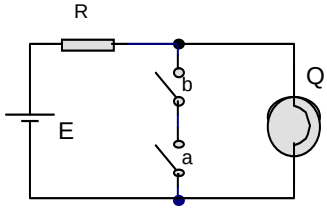
A	Q
0	1
1	0

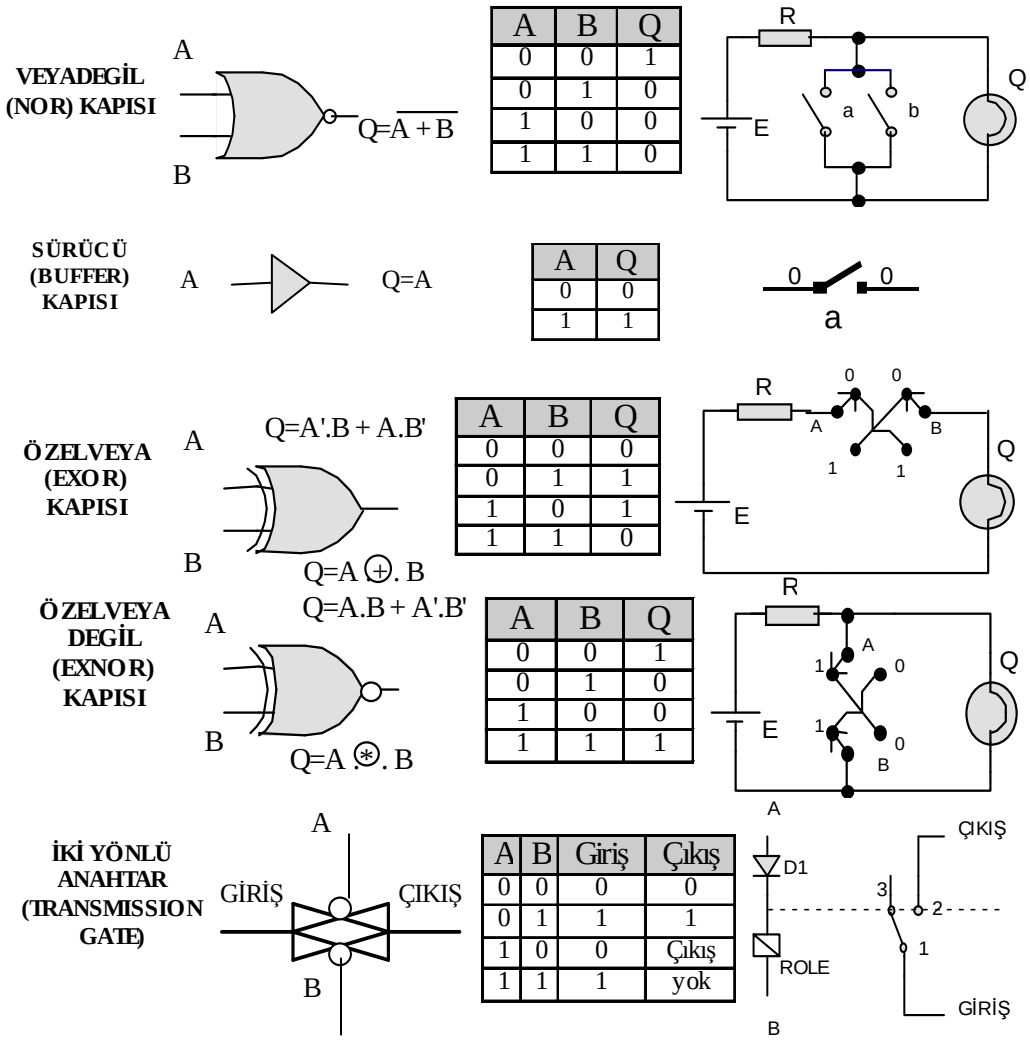


VEDEGİL (NAND) KAPISI



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





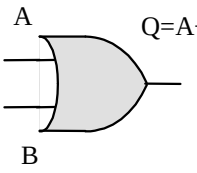
Şekil 5.1. Lojikte kullanılan kapı devreleri, yaptıkları işlemler ve elektriksel eşdeğerleri.

Şimdi, Şekil 5.1'de özetlenen ve toplu olarak verilen lojik kapıların çalışma prensiplerini inceleyelim.

5.1. 'VEYA' İşlemi ve 'VEYA' Kapısı

VEYA (OR) işlemine tabi tutulan A ve B değişkenleri, Şekil 5.2.b'de görülen doğruluk tablosu Q çıkışındaki işlemleri gerçekleştirir. 'VEYA' işleminin normal toplama işleminden farkı; iki değişkenli sistemde her iki girişin '1'

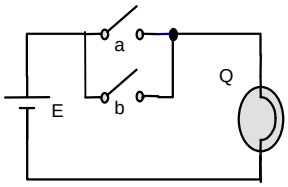
olması durumunda çıkışın Q=1+1=1 olmasıdır (Şekil 5.2.c). **'Q eşit A veya B'** olarak ifade edilen çıkış ifadesinin 1 olması için, girişlerden herhangi birinin lojik '1' olması yeterlidir.



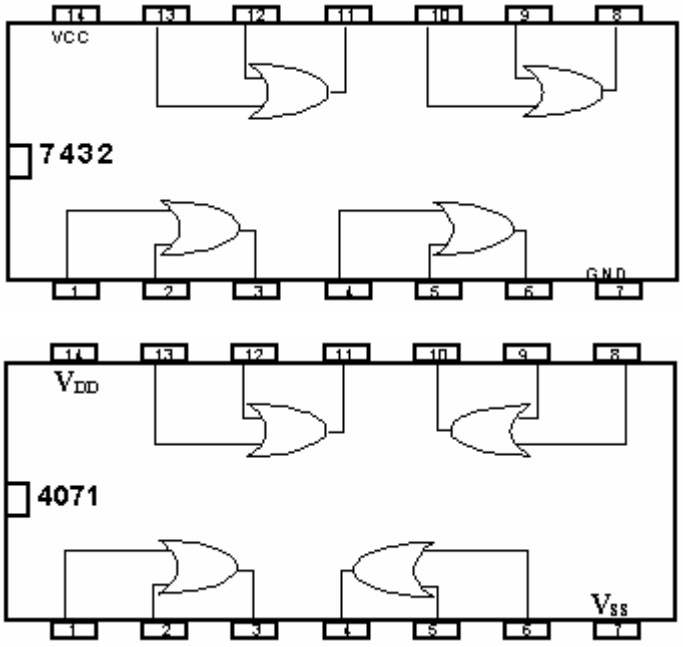
(a)

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

(b)



(c)

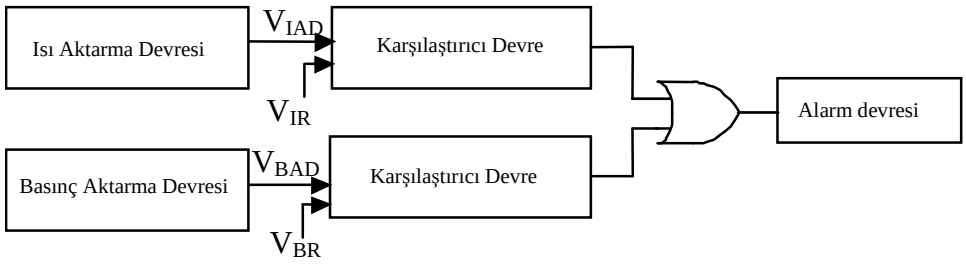


(d)

Şekil 5.2. 'VEYA' kapısı sembolü, doğruluk tablosu, elektriksel eşdeğeri ve 'VEYA' kapısının entegre içerisindeki durumu.

Şekil 5.2.d' de entegreler içerisindeki yerleşim durumu görülen 'VEYA' kapısı, birden çok girişi bulunan ve girişlerden birinin aktif hale gelmesiyle çıkışın aktif olmasının istendiği endüstriyel kontrol uygulamalarında kullanılabilir.

Örnek 1 : Bir kimyasal işlem ünitesinde ısının belirli bir seviyenin üzerine çıkması veya basıncın bir limitin altına düşmesi durumlarında bir alarm sisteminin çalışması istenmektedir. Şekil 5.3'te böyle bir devrenin blok şeması görülmektedir. Böyle bir devrenin çalışmasını genel hatları ile açıklayalım.



Şekil 5.3. 'VEYA' kapısının bir alarm devresinde kullanılması.

Şekil 5.3'de görülen ısı aktarıcı devre, ısı ile orantılı bir gerilim üretir (V_{IAD}). Üretilen gerilim ile referans olarak kullanılan V_{IR} değeri karşılaştırıcı devrede karşılaştırılır. Normalde lojik '0' seviyesinde olan karşılaştırıcı devre çıkışı, V_{IAD} gerilim değerinin V_{IR} (ısı referans) değerinden büyük olması durumunda lojik '1' değerini alır. Benzer bir işlem basınç aktarma devresi içinde uygulanırsa, basıncın belirlenen limitin altına düşmesi durumunda karşılaştırıcı devre çıkışı 0'dan 1'e değişir.

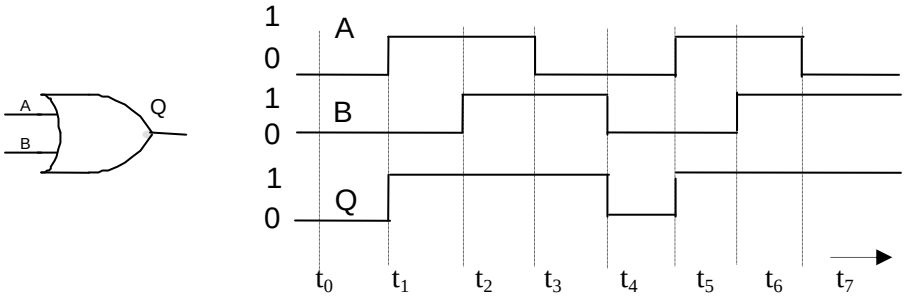
Isı veya basınç işlemlerinde kullanılan karşılaştırıcılardan birisinin çıkışının '1' olması durumunda alarm devresinin çalışmasını istiyorsak, karşılaştırıcı devrelerin çıkışlarını 'VEYA' kapısına uygulayarak istediğimiz sonucu elde edebiliriz. Alarm koşullarından birisinin gerçekleşmesi durumunda 'VEYA' kapısının çıkışı '1' olacağından, alarm devresi aktif hale gelir. Alarm koşullarının daha fazla olması durumunda, daha fazla girişli 'VEYA' kapısı kullanılmasıyla problem çözülebilir.

Girişlerin alacağı durumlara göre çıkışlarda oluşacak değerler grafiksel olarak gösterilebilir. Girişlerdeki degişimlerin belirli zaman dilimlerinde incelendiği grafiksel gösterimler, işlemlerin anlaşılmasına yardım eder.

Örnek 2: Şekil 5.4'te verilen A ve B dalga şekilleri iki girişli 'VEYA' kapısına uygulandığında, Q çıkışında oluşacak dalga şeklini çizelim.

'VEYA' kapısında girişlerden birisi '1' oluncaya kadar çıkış '0' olarak kalır (t_1). Girişlerden birisi '1' olan 'VEYA' kapısı çıkışı '1' olur. Girişlerden birinin '1' olması, çıkışın '1' olması için yeterli olduğundan, girişlerden yalnızca

birinin ‘1’ olarak kalması çıkışın ‘1’ olarak devam etmesi için yeterli olur. Bu durumda t_1 anında ‘1’ durumuna geçen çıkış dalga şekli t_4 anına kadar ‘1’ olarak devam eder. t_4-t_5 arasında her iki girişin ‘0’ olması nedeniyle çıkış ‘0’ değerini alır. t_5 anından sonra girişlerden birinin ‘1’ olması nedeniyle çıkış ‘1’ olur ve ‘1’ olarak devam eder.

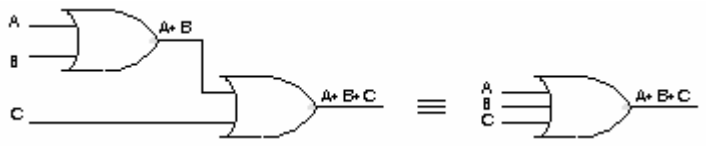


Şekil 5.4. İki girişli ‘VEYA’ kapısı örnek uygulaması.

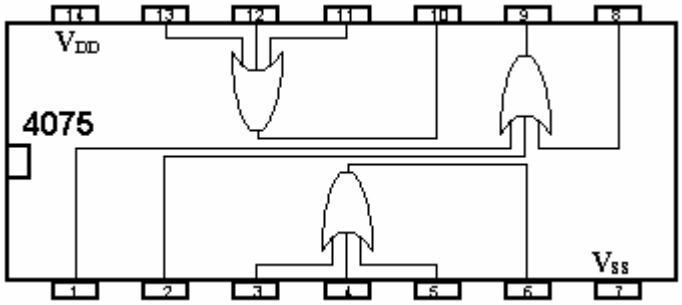
Örnek 3 : İki girişli VEYA kapıları kullanarak üç girişli VEYA kapısı oluşturmak için gerekli bağlantıyı çizerek, Şekil 5.5’te verilen A, B ve C dalga şekilleri üç girişli ‘VEYA’ kapısına uygulanması durumunda ‘VEYA’ kapısı çıkışında oluşacak dalga şeklini gösterelim.

İki girişli VEYA kapıları kullanarak üç girişli VEYA kapısı oluşturulabilir (Şekil 5.5). Üç girişli VEYA kapısının entegre içerisindeki durumu Şekil 5.6’ da görülmektedir.

Üç girişli VEYA kapısındaki A, B, C girişleri zamana göre Şekil 5.7’deki gibi değişirken, girişlerden herhangi birinin ‘1’ olması çıkışın ‘1’ olmasını sağlar. Bu diyagramda dikkat edilmesi gereken nokta, t_3 anında A girişinin 1’ den 0’a geçerken, B girişinin 0’dan 1’e değişmesidir. A ve B girişleri durumlarını yaklaşık aynı anda değiştirdiklerinden ve değişim belirli bir zaman aldığından, t_3 anında girişlerin durumları belirli değildir. Girişlerin kesin belli olmaması, çıkışta bir belirsizliğin oluşmasına ve titreşim meydana gelmesine sebep olur. Hem titreşimin meydana gelmesi, hem de meydana gelen titreşimin genlik ve genişliği girişte olan değişimin hızına bağlıdır.

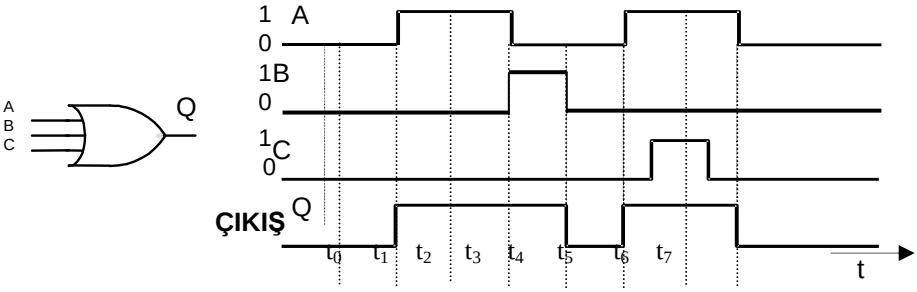


Şekil 5.5. İki girişli VEYA kapıları ile üç girişli VEYA kapısı oluşturulması.



(a)

Şekil 5.6. Üç girişli ‘VEYA’ kapısının entegre içerisindeki durumu.

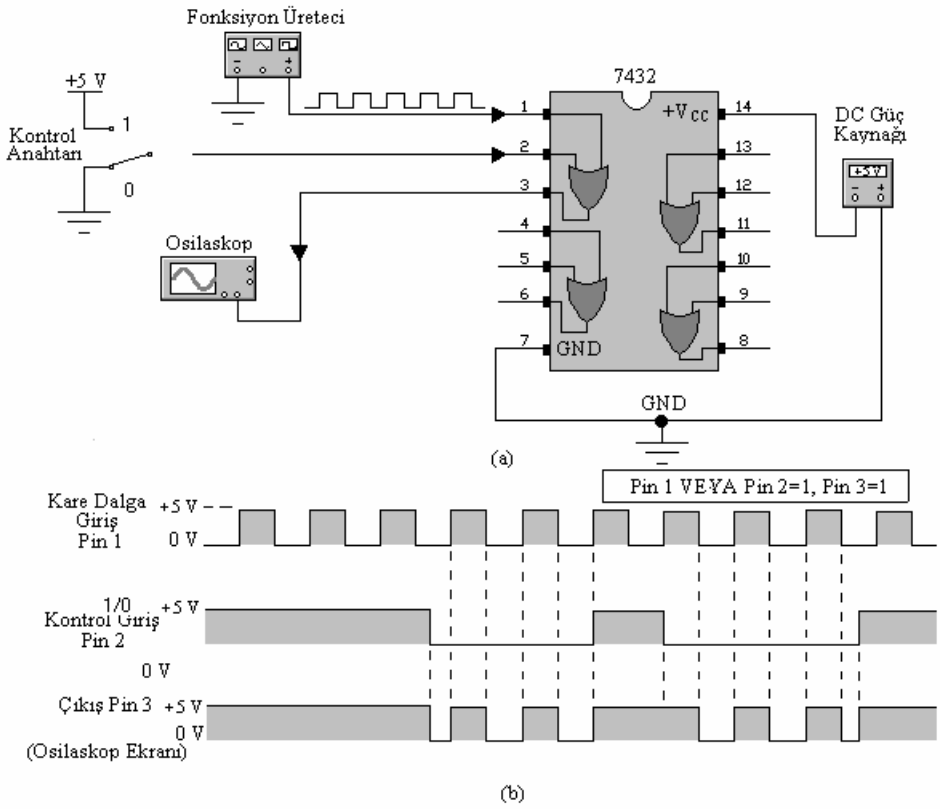


(b)

Şekil 5.7. Üç girişli ‘VEYA’ kapısı örnek uygulaması.

Örnek 4. VEYA kapısının çalışmasını (7432 Entegresi), girişlerden birisini kontrol girişi olarak kullanmak suretiyle sinyal üreteci ve osilaskop yardımıyla inceleyelim.

Kontrol girişi olarak kullanılan girişteki değere göre, ‘VEYA’ kapısının ikinci girişlerinden uygulanan sinyal çıkışta elde edilir. Çıkışta oluşan sinyal osilaskop yardımı ile takip edilebilir (Şekil 5.8).



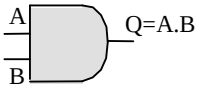
Şekil 5.8. Sinyal üreteci ve osilaskop ile ‘VEYA’ kapısı uygulaması.

5.2. ‘VE’ İşlemi ve ‘VE’ Kapısı

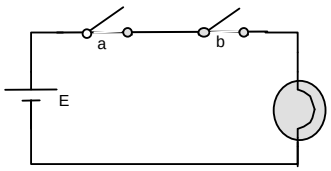
Şekil 5.9.a’ daki sembolle gösterilen ‘VE’ kapısı, Şekil 5.9.b’ de görülen doğruluk tablosundaki işlemleri gerçekleştirir. ‘VE’ kapısının gerçekleştirdiği çarpma işlemi, ‘**.**’ veya ‘*’ işareti ile gösterilir ve kapının yaptığı işlem Q=A*B şeklinde tanımlanır. Çarpma işlemini gerçekleştiren VE kapısının entegreler içerisindeki durumu Şekil 5.9.b’ de görülmektedir.

Normal çarpma işleminin gerçekleştirildiği ‘VE’ işleminde, giriş değişkenlerinin herhangi birisinin ‘0’ değerini alması ile çıkış ‘0’ değerini alırken, girişlerin hepsinin ‘1’ olması durumunda çıkış ‘1’ değerini alır. Bu

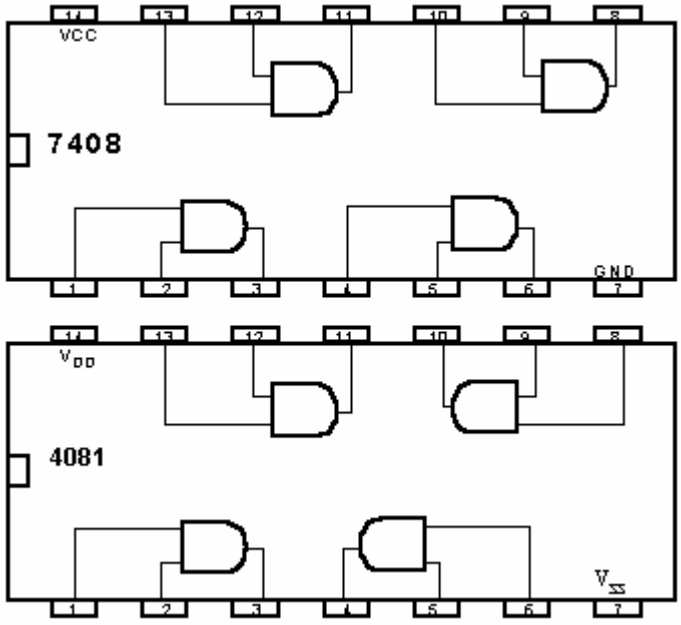
durum iki girişten fazla girişe sahip ‘VE’ kapıları için de geçerlidir. Üç girişe sahip ‘VE’ kapısının çıkışı; Q = A*B*C şeklinde, dört girişe sahip ‘VE’ kapıların çıkışı; Q = A*B*C*D şeklinde gösterilir.



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



(a) **(b)** **(c)**



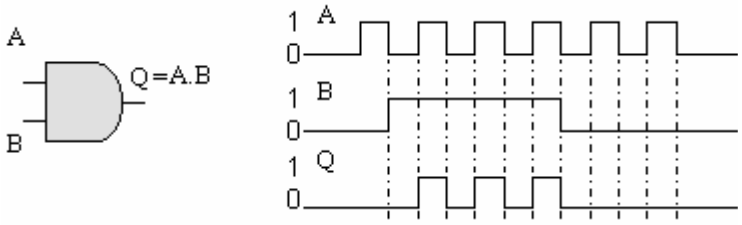
(d)

Şekil 5.9. ‘VE’ kapısı sembolü, doğruluk tablosu, elektriksel eşdeğeri ve entegre içerisindeki durumu.

İki girişli ‘VE’ kapısına uygulanan A ve B değişkenlerinin durumlarına göre oluşacak çıkış dalga şeklini çizmek, ‘VE’ kapısının işlevini anlamaya yardım edecektir.

Örnek 5 : Şekil 5.10’da verilen giriş dalga şekillerine göre ‘VE’ kapısı çıkış dalga şeklini çizelim.

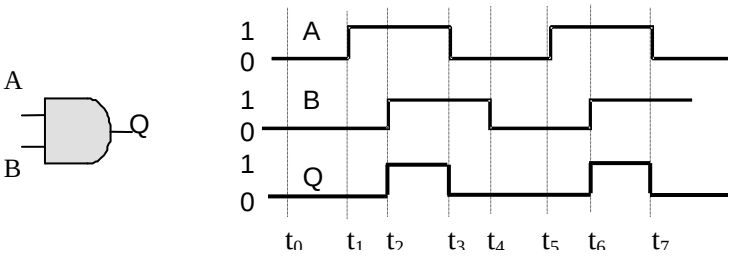
Çıkış, yalnızca iki girişin ‘1’ olması durumunda ‘1’ olur. Bu durumda çıkış her iki girişe bağlı olacağından; B girişinin ‘1’ olduğu anlarda çıkış A’ya bağlı olarak şekillenirken, B’nin ‘0’ olduğu anlarda çıkış ‘0’ değerini alır.



Şekil 5.10. ‘VE’ kapısının işlevinin örnek dalga şekilleri ile gösterilmesi.

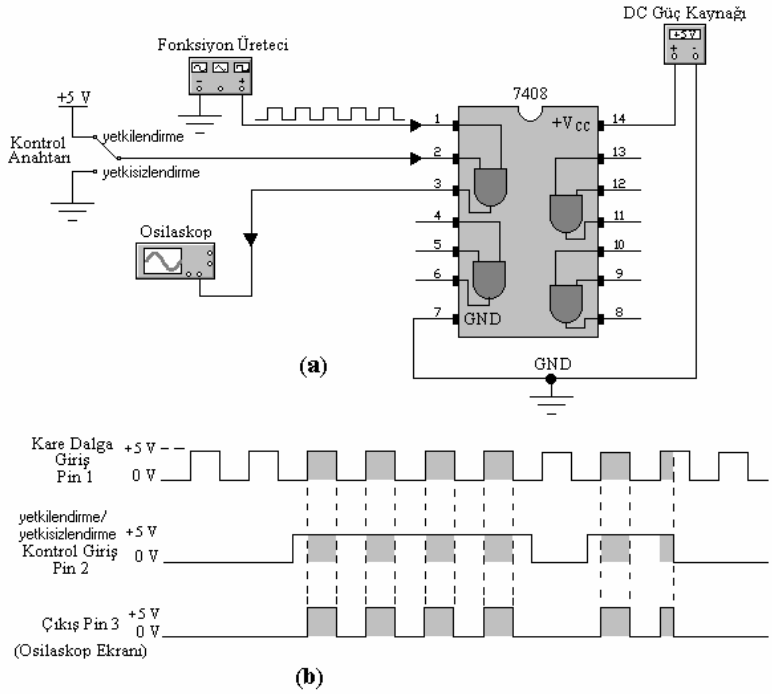
Örnek 6 : Şekil 5.11’ deki A ve B sinyallerinin iki girişli VE kapısına uyğulanması durumunda çıkışta oluşacak dalga şeklini çizelim.

Kapıya uygulanan girişlerin aldığı değerler zamana göre degişirken, her iki girişin ‘1’ olması durumunda çıkış ‘1’ olur. Şekil 5.11’de bu durum t_2-t_3 ve t_6-t_7 zaman dilimlerinde gerçekleştiğinden, çıkış yalnızca bu anlarda ‘1’ değerini alır.



Şekil 5.11. İki girişli ‘VE’ kapısı uygulaması.

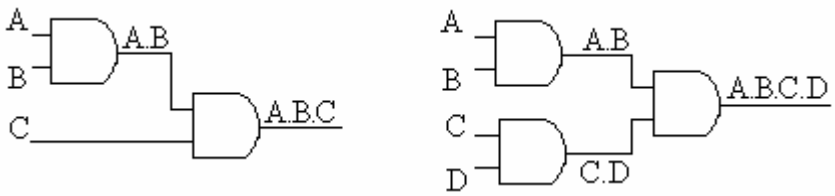
Örnek 7 : Sinyal üretecinden üretilen kare dalga sinyal ve kontrol sinyalinin VE kapısına uygulanması durumunda oluşacak çıkış sinyal şeklini osilaskop ile izleyebilmek için gerekli devreyi oluşturalım (Şekil 5.12).



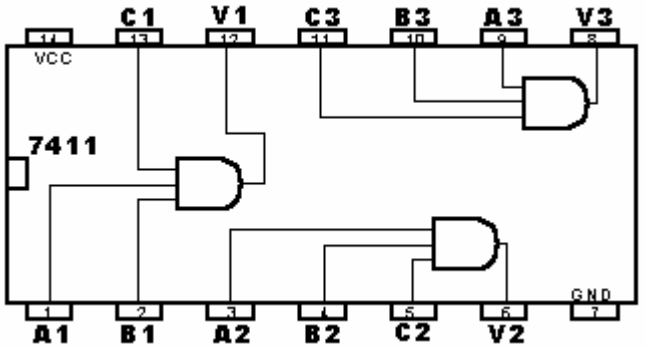
Şekil 5.12. Kare dalga sinyalin VE kapısı ve osilaskop ile izlenmesi.

Örnek 8: İki girişli 'VE' kapıları kullanarak üç ve dört girişli 'VE' kapıları oluşturalım.

İki girişli VE kapıları ile üç girişli ve dört girişli VE kapıları oluşturma şekilleri Şekil 5.13' de, üç girişli VE kapılarının entegre içerisindeki durumu ise Şekil 5.14' de görülmektedir.



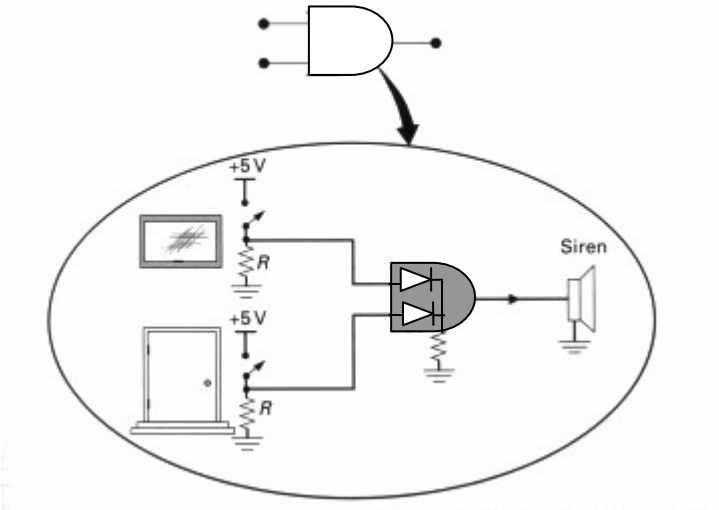
Şekil 5.13. İki girişli 'VE' kapısı kullanarak üç ve dört girişli 'VE' kapısının oluşturulması.



Şekil 5.14. 7411 entegresinin ayak bağlantısı.

Örnek 9 : Kapısı ve bir adet penceresi bulunan bir odada, kapı ve pencerenin aynı anda kapalı olması durumunda alarmın çalmasını sağlayan devreyi 'VE' kapısı kullanarak gerçekleştirelim.

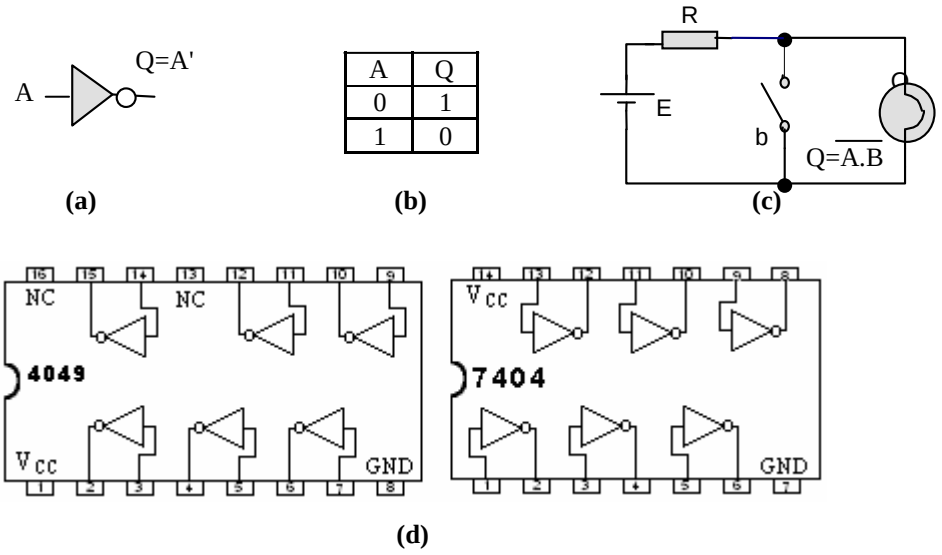
Kapı ve pencereye bağlı anahtarların oluşturduğu devrelerin giriş olarak kullanılması ve çıkışa sirenin bağlanması ile Şekil 5.15'deki devre oluşur. Oluşan devrede çıkış, yalnızca iki girişin '1' olması durumunda '1' olur. Bu durumda çıkış her iki girişe bağlı olacağından; her iki anahtarın (kapı ve pencerenin) kapalı olması durumunda çıkış '1' olur ve alarm çalışır.



Şekil 5.15. Hırsız alarm sisteminin **'**VE' kapısı ile oluşturulması.

5.3. ‘DEĞİL’ İşlemi ve ‘DEĞİL’ Kapısı

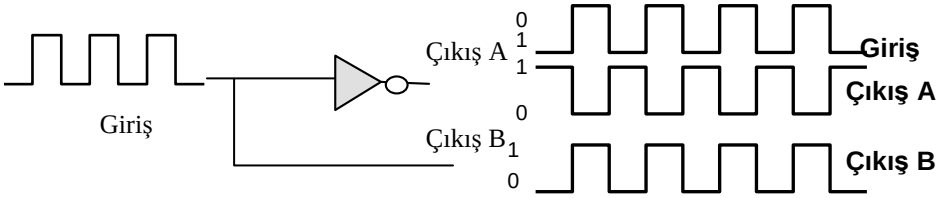
‘DEĞİL’ işlemi; ‘VE’, ‘VEYA’ işlemlerinden farklı olarak tek giriş ve tek değişken ile gerçekleştirilir. Örneğin; A değişkeni ile işlem yapılacaksa, ‘NOT’ işlemi sonucu Q=A' olarak tanımlanır ve A üzerindeki çizgi, değili (barı) olarak isimlendirilir. Şekildeki doğruluk tablosundan görüleceği üzere değişken yalnızca iki değerden birini alabilir: A=0 veya A=1. ‘DEĞİL’ işlemi, ‘tersi’ veya ‘tümleyeni’ olarak ta tanımlanır. Şekil 5.16.a’ daki sembolle gösterilen ‘DEĞİL’ kapısı her zaman tek girişe sahiptir ve çıkış devamlı girişin tersidir (Şekil 5.12.b).



Şekil 5.16. ‘DEĞİL’ kapısı sembolü, doğruluk tablosu, elektriksel eşdeğeri ve değil kapılarının entegre içerisindeki durumu.

Örnek 10 : ‘DEĞİL’ kapısı kullanarak, iki fazlı sinyal üreten bir lojik devre oluşturalım.

‘DEĞİL’ kapısı kullanılarak oluşturulan devrede, girişe uygulanan kare dalga sinyal ile aynı ve 180 derece ters fazında iki sinyal çıkış olarak oluşur (Şekil 5.17).



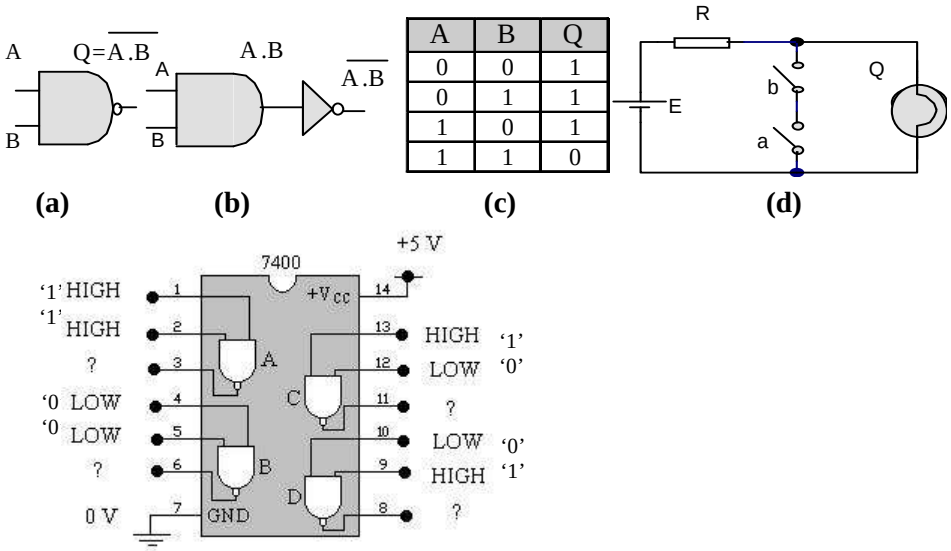
Şekil 17. İki fazlı çıkışa sahip devrenin 'DEĞİL' kapısı ile gerçekleştirilmesi.

5.4. 'VEDEĞİL' Kapısı

Lojik'de yaygın olarak kullanılan diğer bir kapı, 'VE' ile 'DEĞİL' kapılarının (işlemlerinin) birleşmesinden oluşan 'VEDEĞİL' (NAND) kapısıdır (Şekil 5.18.a). 'VEDEĞİL' kapısında, girişlerden birisinin '0' olması durumunda çıkış lojik '1' olur. Giriş değişkenlerinin tümünün '1' olması durumunda çıkış '0' değerini alır.

'VEDEĞİL' işleminin çıkış fonksiyonu, $Q = \overline{A.B}$ şeklinde yazılır ve **'Q eşittir A ve B'nin değili'** diye okunur. 'VEDEĞİL' işlemi 'VE' ile 'DEĞİL' kapılarının seri bağlanması şeklinde düşünülebilir (Şekil 5.14.b). Bu mantık ile doğruluk tablosu 'VE' kapısının tamamen tersi olarak yazılabilir (Şekil 5.18.c).

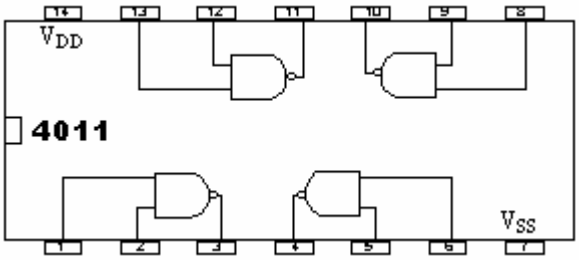
'VEDEĞİL' kapısının farklı entegreler içerisindeki durumu Şekil 5.18' de görülmektedir.



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

4 adet VEDEĞİL kapısı içeren 7400 entegresi iç yapısı

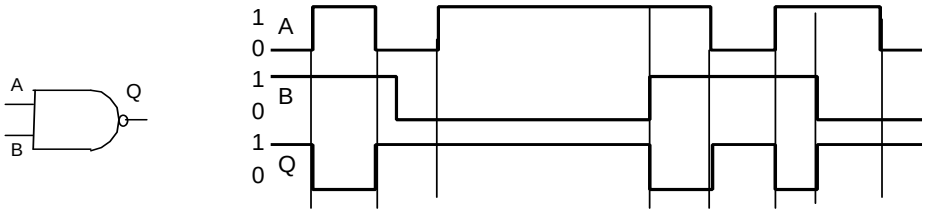
(e)



(f)

Şekil 5.18. 'VEDEĞİL' kapısı sembolü, 'Ve+Değil' eşdeğeri, doğruluk tablosu, elektriksel eşdeğeri ve entegreler içerisindeki durumu.

Örnek 11 : Şekil 5.19'da verilen A ve B dalga şekillerinin iki girişli 'VEDEĞİL' kapısına uygulanması durumunda oluşacak çıkış dalga şeklini çizelim.



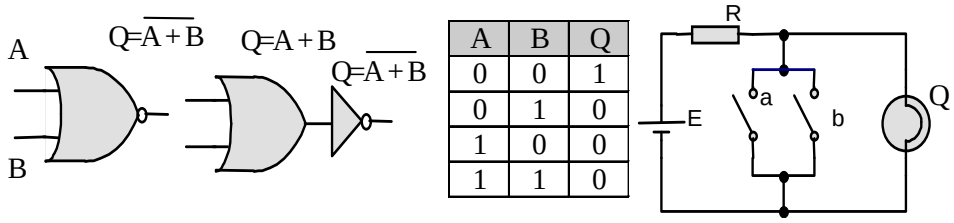
Şekil 5.19. İki girişli 'VEDEĞİL' kapısı uygulaması.

Çıkış dalga şekli birkaç yolla çizilebilir: İlk yöntem olarak; önce VE kapısının çıkışı çizilip daha sonra bunun tersi alınabilir. İkinci bir yöntem olarak; 'VEDEĞİL' kapısında, yalnızca her iki girişin '1'olması durumunda çıkış '0' olacağından, bu durumlar tespit edilip çıkış '0' olarak çizilir. Bu durum haricindeki durumlarda çıkış '1' olacağından, çıkış şekli '1' olarak tamamlanır.

5.5. 'VEYADEĞİL' (NOR) Kapısı

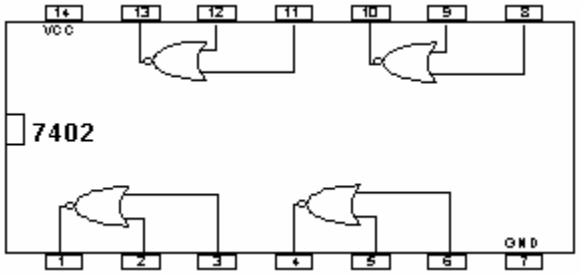
''VEYA' ve 'DEĞİL' kapılarının birleşiminden oluşan 'VEYADEĞİL' kapısı, 'VEYA' kapısının gerçekleştiği işlemin tersini yapar. Sembolü ve doğruluk tablosu Şekil 5.20' de görülen 'VEYADEĞİL' kapısında, yalnızca girişlerin

tümünün ‘0’ olması durumunda çıkış ‘1’ olur. ‘VEYADEĞİL’ kapısının işlevi Q = A+B şeklinde ifade edilir ve **‘Q eşittir A veya B’nin değili’** diye okunur.



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

(a) **(b)** **(c)** **(d)**

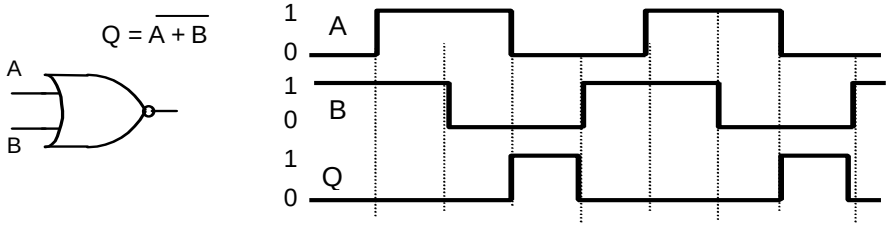


(e)

Şekil 5.20. ‘VEYADEĞİL’ kapısı sembolü, ‘VEYA+DEĞİL’ eşdeğeri, doğruluk tablosu, elektriksel eşdeğeri ve entegre içerisindeki durumu.

Örnek 12 : Şekil 5.21’de verilen A ve B dalga şekillerinin ‘VEYADEĞİL’ kapısına uygulanması durumunda çıkışta oluşacak dalga şeklini çizelim.

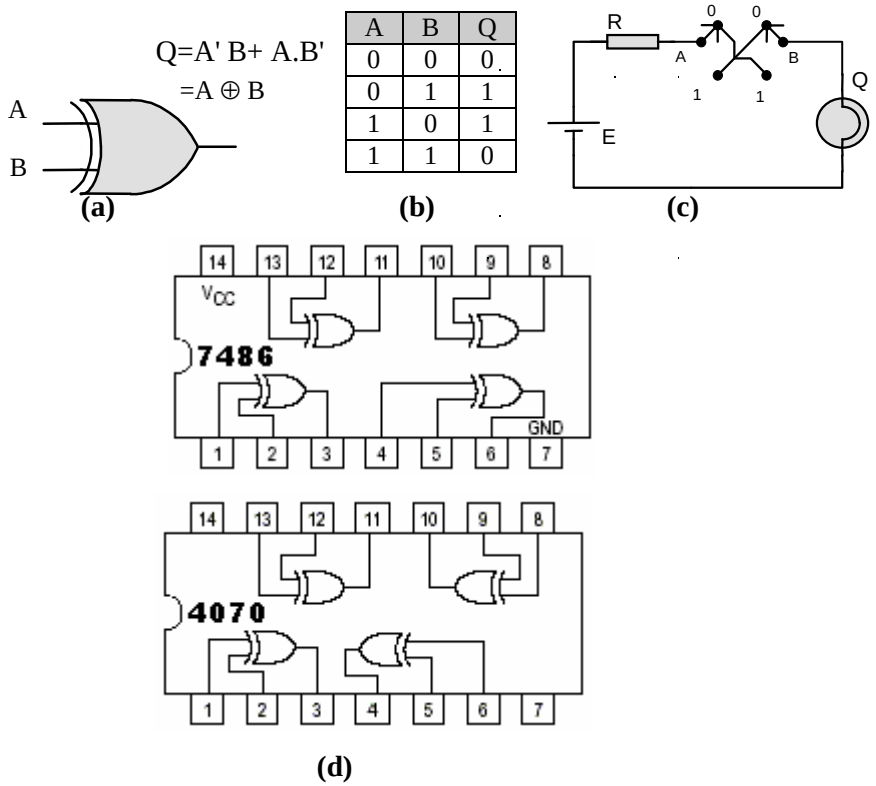
‘VEYADEĞİL’ kapısının çıkışı, ‘VEDEĞİL’ kapısında olduğu gibi iki yöntemle çizilebilir. Önce ‘VEYA’ kapısının çıkışı çizilip daha sonra tersi alınabilir. İkinci yöntemde, ‘VEYADEĞİL’ kapısında yalnızca tüm girişlerin ‘0’ olması durumunda çıkışın ‘1’ olacağı düşünülerek, önce çıkışın ‘1’ olacağı durumlar çizilir. Daha sonra girişlerin durumlarına bakılmaksızın, kalan kısımlar ‘0’ olarak tamamlanır (Şekil 5.21).



Şekil 5.21. İki girişli ‘VEYADEĞİL’ kapısı uygulaması.

5.6. ‘ÖZELVEYA’ Kapısı (EXOR GATE)

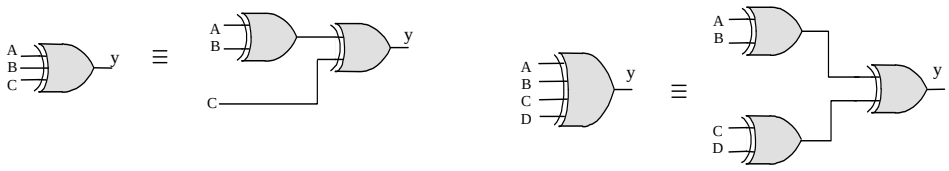
Şekil 5.22.a’ da sembolü gözüken ‘ÖZELVEYA’ kapısında, iki giriş ve bir adet çıkış bulunur. Bu kapıda, doğruluk tablosunda görüleceği üzere giriş değişkenlerinin birbirinin aynısı olduğu durumlarda çıkış ‘0’, giriş değişkenlerinin durumlarının farklı olduğu durumlarda ise çıkış ‘1’ olur (Şekil 5.22.b).



Şekil 5.22. ‘ÖZELVEYA’ kapısı sembolü, doğruluk tablosu, elektriksel eşdeğeri ve entegre içerisindeki durumu.

‘ÖZELVEYA’ kapısının çıkışı Q=A⊕B veya Q=AB'+ A'B şeklinde yazılabilir. ‘ÖZELVEYA’ kapısı, yaptığı işlem referans alınarak ‘**farklılık kapısı**’ olarak ta tanımlanır. ‘ÖZEL VEYA’ kapısına sahip entegrelerin şemaları Şekil 5.22’ de görülmektedir.

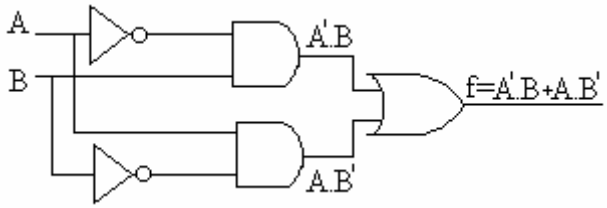
Üç veya dört girişli ‘ÖZELVEYA’ kapıları, iki girişli ‘ÖZELVEYA’ kapıları kullanılarak oluşturulabilir (Şekil 5.23). Üç veya dört girişli ‘ÖZELVEYA’ kapısında, girişlerdeki ‘1’ değerlerinin sayısı tek ise çıkış ‘1’ olurken, girişlerdeki ‘1’ değerlerinin sayısı çift ise çıkış ‘0’ olur.



Şekil 5.23. İki girişli ‘ÖZELVEYA’ kapıları kullanılarak, üç ve dört girişli ‘ÖZELVEYA’ kapılarının oluşturulması.

Örnek 13 : ‘ÖZEL VEYA’ kapısını ‘VE’, ‘VEYA’, ve ‘DEĞİL’ kapıları kullanarak oluşturalım.

‘ÖZEL VEYA’ kapısı ile $f=A.B^{'} + A^{'}.B$ işlemi gerçekleştirildiğinden, gerekli eşitliği sağlayacak devrenin çizilmesi ile Şekil 5.24’ deki lojik bağlantı oluşur.



Şekil 5.24 : ‘ÖZEL VEYA’ kapısının ‘VE-VEYA-DEĞİL’ kapıları ile oluşturulması.

5.7. ‘ÖZELVEYADEĞİL’ Kapısı (EXNOR GATE)

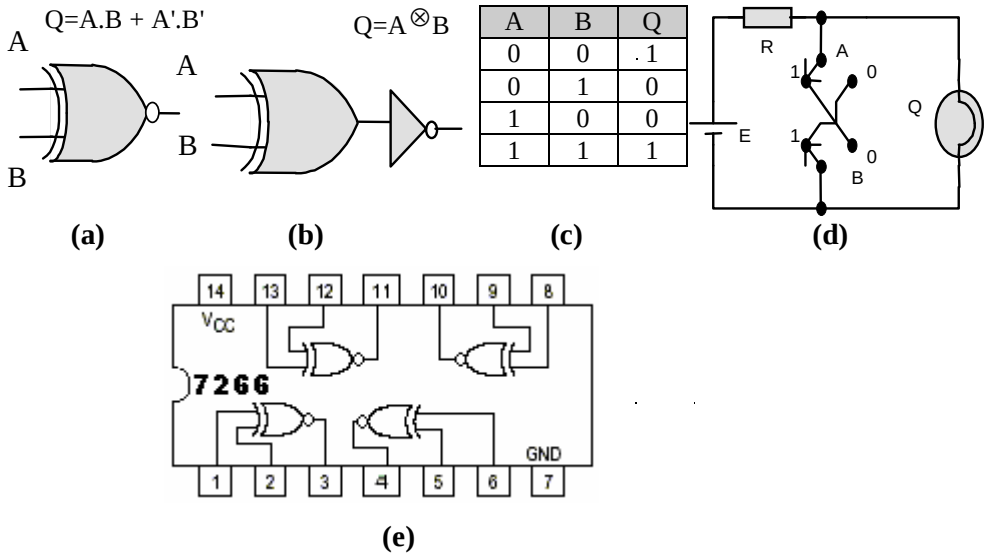
Yalnızca iki giriş ve bir çıkışa sahip olan ‘ÖZELVEYADEĞİL’ kapısında; giriş değişkenlerinin aynı değeri alması durumunda çıkış lojik ‘1’, giriş değişkenlerinin farklı değerleri alması durumunda ise çıkış lojik ‘0’ değerini alır

(Şekil 5.25.a). Bu işlem referans alınarak 'ÖZELVEYADEĞİL' kapısı, '**eşitlik kapısı**' olarak adlandırılır.

'ÖZELVEYADEĞİL' kapısının çıkış fonksiyonu;

$Q = A \otimes B$ veya $Q = AB + A'B'$

şeklinde yazılabilir (Şekil 5.25.b). Şekil 5.25.e' de 'ÖZEL VEYA' kapısının entegre devre içerisindeki durumu görülmektedir.



Şekil 5.25. 'ÖZELVEYA DEĞİL' kapısı sembolü, 'EXOR+NOT' eşdeğeri, doğruluk tablosu, elektriksel eşdeğeri ve entegre bağlantısı.

74135 entegresi, üç girişli 'ÖZELVEYA' kapısı olarak veya kontrol girişi ile 'ÖZELVEYA / ÖZELVEYADEĞİL' kapısı olarak şekillendirilebilecek yapıya sahip bir entegredir (Şekil 5.26.a).

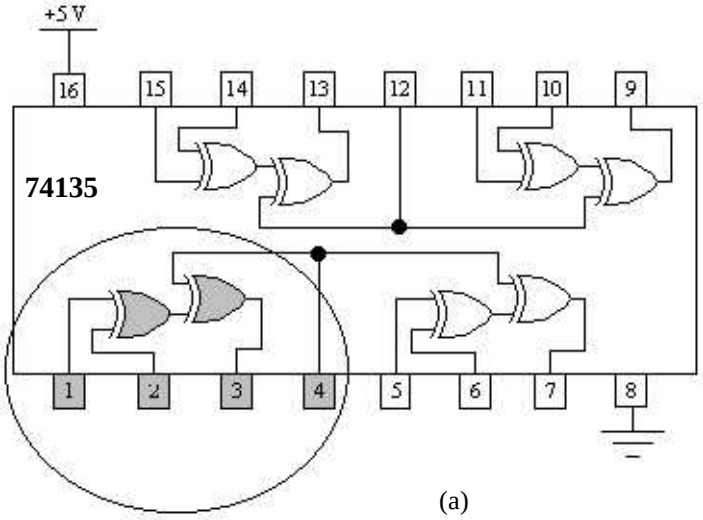
74135 entegresinde bulunan 1, 2 ve 4 nolu pinler 'ÖZELVEYA' kapısı girişleri olarak kullanılırsa; entegre üç girişli kapı olarak çalışır. Aynı şekilde, 4, 5 ve 6 nolu pinlerin giriş olarak kullanılması ile diğer üç girişli 'ÖZELVEYA' kapısı elde edilir. Bununla beraber, 74135 entegresinin genel kullanım şekli; 4 ve 12 nolu pinlerin kontrol girişleri olarak kullanılması ve bu kontrol girişleri ile entegrenin işlevinin 'ÖZELVEYA' kapısı veya 'ÖZELVEYADEĞİL' kapısı olarak şekillendirilmesidir.

4 nolu girişin kontrol girişi olarak kullanılması ve lojik '0' değerinin uygulanması durumunda; 1 ve 2 nolu girişlerin farklı olması ile çıkış '1'

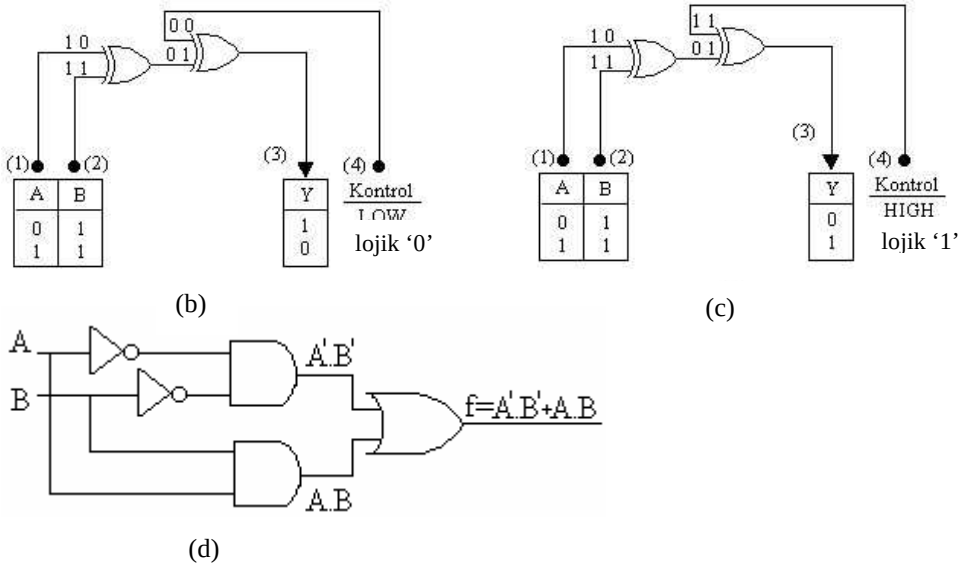
değerini alır. Bu durumda entegre, 'ÖZELVEYA' kapısı olarak işlev görür (Şekil 5.26.b). 4 nolu giriş kontrol girişi olarak kullanılır ve lojik '1' uygulanırsa; girişlerin aynı olması durumunda çıkış '1' olur ve entegre 'ÖZELVEYADEĞİL' kapısı olarak çalışır (Şekil 5.26.c).

Örnek 14 : 'ÖZEL VEYA DEĞİL' kapısını, 'VE-VEYA-DEĞİL' kapıları kullanarak oluşturalım.

'ÖZEL VEYA DEĞİL' kapısı ile $f=A.B + A^{1}.B^{1}$ işlemi gerçekleştirildiğinden, ilgili eşitliği sağlayacak devrenin oluşturulması ile Şekil 5.27' deki lojik devre oluşur.



(a)



Şekil 5.26. 74135 entegresinin iç yapısı ve 'ÖZELVEYA / ÖZELVEYADEĞİL' kapısı olarak kullanımı.

5.8. İki Yönlü Anahtar (Transmission Gate)

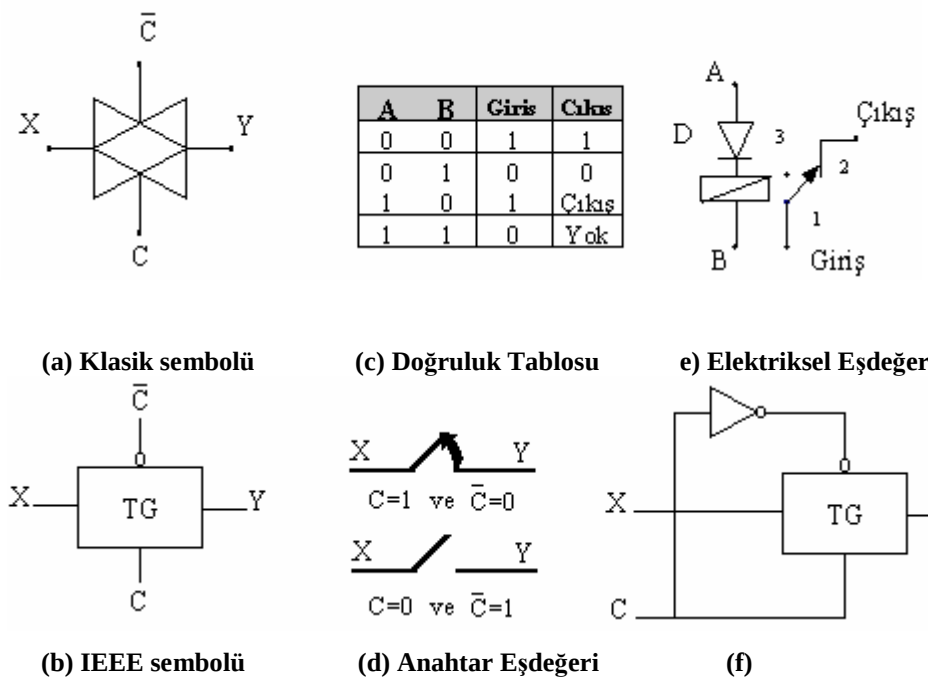
İki yönlü anahtar; lojik fonksiyonları gerçekleştirmede kullanılmasada, lojik tasarımlarda kontrol işlemlerinde kullanılan ve bu nedenle incelenmesi gerekli olan lojik elemandır.

Kontrol girişleri adı verilen girişlerin konumlarına göre giriş işaretini çıkışa aktaran veya aktarmayan iki yönlü anahtarda; iki kontrol girişi, bir sinyal giriş ve bir sinyal çıkışı olmak üzere dört bağlantı ucu bulunmaktadır (Şekil 5.27.a).

İki yönlü anahtarda, kontrol girişleri olarak adlandırılan C ve C^1 girişlerinin uygun durumda olması ile; bilgi X ile Y arasında herhangi bir yönde iletilir. C ve C^1 kontrol girişleri uygun durumda değilse, sinyal iletimi olmaz (Şekil 5.27.d). Bilgi (sinyal) iletiminin oluştuğu durum kapalı bir anahtar ile gösterilirken, bilgi iletiminin olmadığı durum açık anahtar ile gösterilir (şekil c). İki yönlü anahtarın çalışması, Şekil 5.27.f.' deki gibi lojik bağlantı ile açıklanabilir. Kontrol girişleri 'DEĞİL' kapısı ile birbirine bağlıdır ve birbirinin tersi değer olarak çalışırlar.

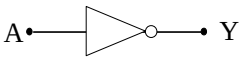
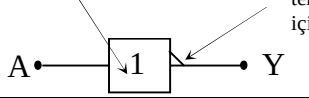
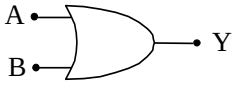
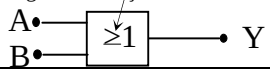
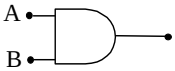
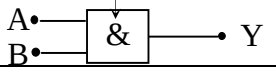
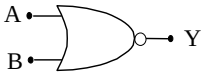
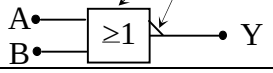
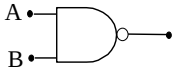
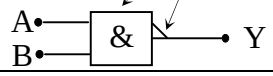
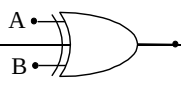
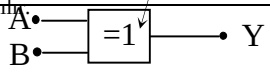
İki yönlü anahtarın çalışmasını Şekil 5.22.e' de verilen elektriksel eşdeğeri üzerinde incelersek; kontrol girişlerinden A=0 ve B=1 ise D diyotu ters polarma olacağından röle enerjilenmez ve anahtar 1→2 konumunda kalarak giriş sinyali

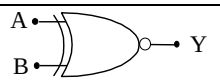
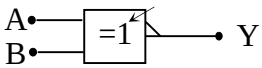
çıkışa aktarılır. A=1 ve B=0 olması durumunda D diyotu doğru yönde enerjileneceğinden röle çeker, anahtar 1→3 konumunu alır ve giriş sinyali çıkışa aktarılmaz. Bu durumda iki yönlü anahtarın çalışmasını; A=0 iken giriş sinyali çıkışa aktarılırken, A=1 olması durumunda giriş sinyali çıkışa aktarılmaz şeklinde özetleyebiliriz.



Şekil 5.27. İki yönlü anahtarın sembolü, doğruluk tablosu ve elektriksel eşdeğeri.

Buraya kadar açıklanan tüm lojik kapıların, Avrupa standartlarına göre oluşturulan sembolleri yanında Amerika IEEE tarafından geliştirilen sembolleri de kullanılmaktadır. İşlevleri tamamen aynı olan kapıların tek farkı kapıların farklı sembollerle ifade edilmeleridir (Şekil 5.28).

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. 

Şekil 5.28. Lojik kapıların Avrupa standartları ve IEEE standartları.

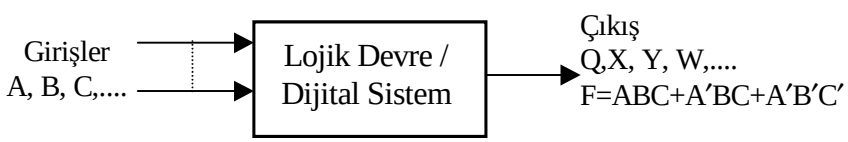
5.9. Lojik İfadelerin Lojik Elemanlarla Gerçekleştirilmesi ve Lojik Devrelerin Tasarımı

Daha önceki bölümlerde bahsedildiği üzere '0' ve '1' işareti ile çalışan devrelere '**lojik devreler**' veya '**dijital sistemler**' denir. Bilgisayarlar, en iyi bilinen dijital sistemlerdir. Dijital sistemlere örnek olarak; dijital telefon santralleri, dijital voltmetreler, frekans sayıcılar, trafik sinyalleri, hesap makineleri, video oyunları, otomobil kontrol sistemleri, sinyal üreteçleri ve dijital osiloskoplar gösterilebilir.

Lojik ifadeleri elemanlar ile gerçekleştirmek, dijital sistemleri oluşturmak ve ikili işaretleri işlemek için temel lojik devre elemanları (lojik kapılar) kullanılır. Temel lojik devre elemanları arasındaki etkileşimlere '**lojik bağlantılar'** denir. Lojik bağlantıların gerçekleştirilebilmeleri için, oluşturulmak istenen bağlantının lojik fonksiyonlar şeklinde ifade edilmesi gerekir. Lojik eşitlik (fonksiyon) olarak isimlendirilen bu ifade, yapılması gerekli işlemleri açıklar.

5.9.1. Lojik İfadelerin Lojik Elemanlar İle Gerçekleştirilmesi

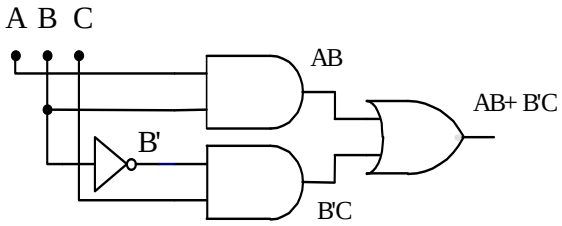
Lojik ifadelerde, lojik devrelerde ve doğruluk tablolarında giriş karakterleri veya giriş değişkenleri A,B,C.. ile gösterilirken, çıkış işaretleri veya değişkenleri Q,X,W... ile temsil edilir (Şekil 5.29). Uygulanan giriş değişkenlerine göre oluşması istenen çıkışı sağlayan lojik devre, lojikte geçerli prensipler ve lojik kapılar kullanılarak gerçekleştirilir. Gerçekleştirilmesi istenen işlem, 'F' veya 'Q' harfi ile temsil edilen eşitlik şeklinde yazılır.



Şekil 5.29. Lojik devre blok şeması.

Örnek 15 : F=AB+B'C lojik ifadesini gerçekleştirecek devreyi lojik kapılar ile oluşturalım.

Verilen eşitlikteki işlemlerin gerçekleştirilmesine, çarpma işlemi ile başlanır. Ancak çarpılan değişkenlerden 'DEĞİL' olan varsa, önce 'DEĞİL' kapısı kullanılarak bu işlem gerçekleştirilir. Çarpma işlemi 'VE' kapısı ile gerçekleştirilebileceğinden, çarpılacak ifadeler 'VE' kapısına uygulanır (Şekil 5.30).

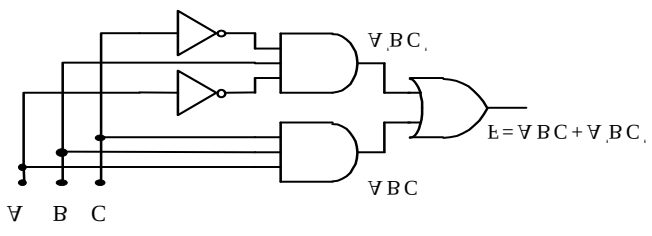


Şekil 5.30. AB+B'C ifadesinin lojik kapılarla oluşturulması.

Verilen lojik ifadedeki çarpma işlemlerinin gerçekleştirilmesi ile lojik kapı çıkışlarında elde edilen ifadeler, toplama işlemine tabi tutulur. Toplama işlemi 'VEYA' kapısı ile gerçekleştirildiği için, ifadeler 'VEYA' kapısına uygulanır. Oluşan lojik devrenin çıkışında verilen lojik ifade elde edilir.

Örnek 16 : F = ABC+A'BC' fonksiyonunu temel lojik kapılar ile gerçekleştirelim.

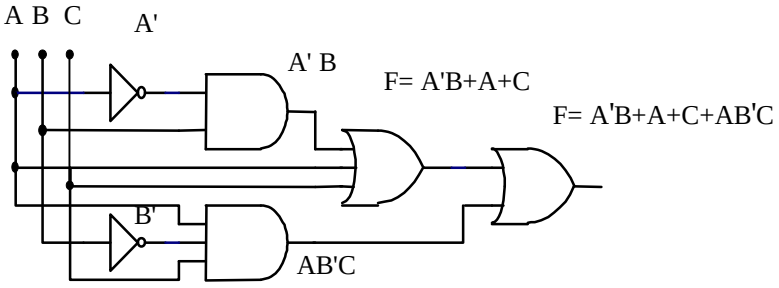
Mintermler şeklinde verilen böyle bir fonksiyonu gerçekleştirmek için, önce 'DEĞİL' ifadesinin oluşmasına yardım edecek 'NOT' işlemleri ile 'VE' kapılarıyla temsil edilen çarpma işlemleri yapılır. Bu işlemler daha sonra toplama işlemini gerçekleştirecek VEYA kapısı ile birleştirilir (Şekil 5.31).



Şekil 5.31. F=ABC+ A'BC' lojik ifadesinin temel kapı devreleri ile gerçekleştirilmesi.

Örnek 17 : F = A'B +A+C+AB'C lojik ifadesini kapı devreleri ile gerçekleştirelim.

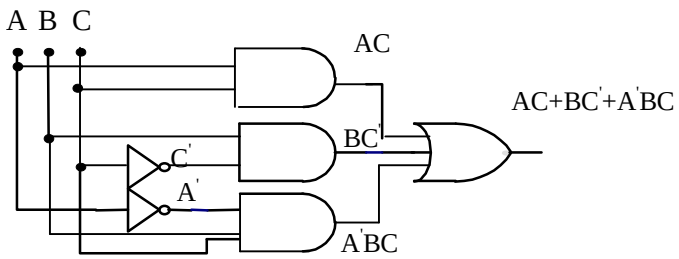
Verilen ifadede dört adet minterm ifadesinin toplanması gerektiğinden, toplama işlemleri dört girişli 'VEYA' kapısı kullanılması veya daha az sayıda girişli 'VEYA' kapılarının birlikte kullanılması ile gerçekleştirilir.



Şekil 5.32. F=A'B +A+C+AB'C ifadesinin kapı devreleriyle gerçekleştirilmesi.

'DEĞİL' kapıları ile tersi alınan değişkenler, 'VE' kapıları ile çarpma işlemine tabi tutulur. Elimizde 2 ve 3 girişli 'VEYA' kapıları bulunduğunu varsayarsak; verilen eşitliği gerçekleştirecek Şekil 5.32'deki lojik devre oluşur.

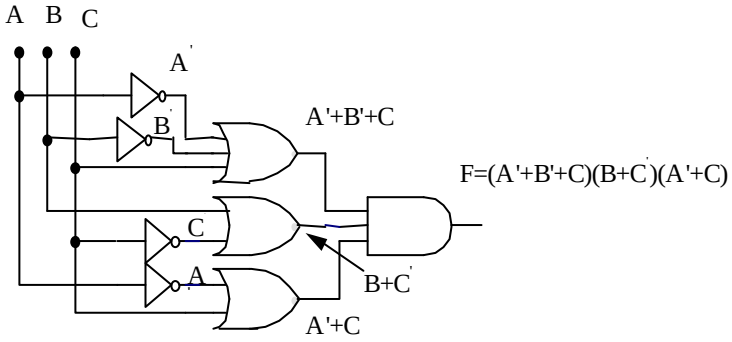
Örnek 18 : F=AC+BC'+A'BC lojik ifadesini temel lojik kapılar ile gerçekleştirelim.
Lojik ifadedeki 'DEĞİL' işlemlerinden sonra çarpma (VE) ve son olarak toplama (VEYA) işlemlerinin gerçekleştirilmesi ile Şekil 5.33' deki lojik şema oluşur.



Şekil 5.33. AC+BC'+A'BC ifadesinin gerçekleştirilmesi.

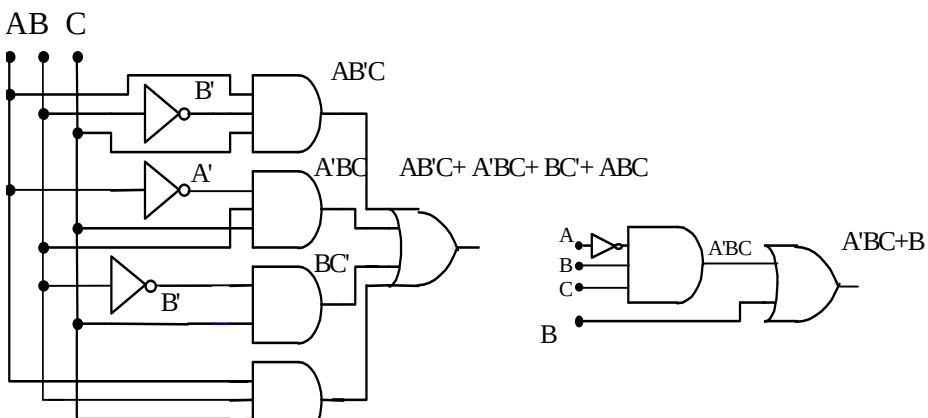
Örnek 19: F = (A'+B'+C).(B+C').(A'+C) fonksiyonunu gerçekleştirecek lojik devreyi çizelim.
Bu şekildeki bir fonksiyonda önce parantez içerisindeki ifadeler 'DEĞİL' işleminden başlanarak gerçekleştirilir. Daha sonra 'VEYA' kapıları ile birleştirilen ifadeler 'VE' kapısına uygulanarak lojik tasarım bitirilir (Şekil 5.34).

Buraya kadar ki örneklerde fonksiyonlar sadeleştirme işlemine tabi tutmadan orijinal şekli ile gerçekleştirildi. Normalde verilen fonksiyonun sadeleştirildikten sonra lojik kapılarla gerçekleştirilmesi gerekir.



Şekil 5.34. (A'+B'+C).(B+C').(A'+C) lojik ifadesinin lojik kapılarla oluşturulması.

Örnek 20 : F=AB'C+A'BC+BC'+ABC ifadesini normal şekli ve Boolean kurallarını kullanarak sadeleştirdikten sonra lojik kapılarla gerçekleştirelim.



Şekil 5.35. Bir lojik eşitliğin sadeleştirilmeden önce ve sadeleştirildikten sonra lojik kapılarla oluşturulması.

F=AB'C+A'BC+BC'+ABC=AB'C+BC(A'+A)+BC'
=AB'C+BC+BC'=AB'C+B(C'+C) 1
=AB'C+B 1

Kapı devreleri ile gerçekleştirilmiş lojik bir devrenin lojik fonksiyonunun çıkarılması ve elde edilen fonksiyonun basitleştirilmesi istenebilir. Bu durumda yapılacak işlem; lojik devreden fonksiyonu yazmak, bulunan fonksiyonu sadeleştirmek ve sadeleştirme sonucunda bulunacak yeni ifadeyi gerçekleştirmek şeklinde olur (Şekil 5.35).

Örnek 21 : Şekil 5.31.a'da verilen lojik şemanın çıkış eşitliğini yazarak, bulunan eşitliği sadeleştirip, sadeleştirilen ifadenin lojik diyagramını çizelim.

Herbir kapının çıkışındaki elde edilen eşitliği yazar ve elde edilen eşitliği sadeleştirirsek;

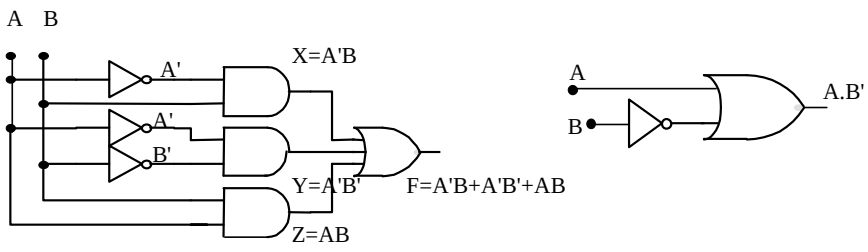
X = A'B Y = A'B' Z = AB F = A'B+A'B'+AB

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

F = A'+AB = A.(AB)' F = A.(A'+B') = AA'+AB' = AB'

F = AB'

sonucuna ulaşırız (Şekil 5.36).



Şekil 5.36. Lojik şemadan lojik eşitliğin yazılması ve sadeleştirme işlemi.

5.9.2. Lojik Devrelerin Tasarlanması ve Lojik Elemanlar Kullanılarak Gerçekleştirilmesi

Lojik devrelerin tasarımında, yapılacak işleme karar verildikten sonra, yapılacak işlemin lojik prensipler ve eşitlikler kullanılarak lojik fonksiyon şekline dönüştürülmesi ve daha sonra kapılar ile gerçekleştirilmesi sırası takip edilir.

Lojik devre tasarımında yapılacak işlemleri sıralarsak, aşağıdaki işlem sırası oluşur;

1. Yapılmak istenen işlem ayrıntıları ile açıklanır.
2. Lojik işlemin detayları belirlenir ve doğruluk tablosu haline dönüştürülür.
3. Doğruluk tablosu, lojik eşitlik (fonksiyon) şeklinde yazılır.
4. Eşitlik, mümkünse sadeleştirme işlemine tabi tutulur.
5. Sadeleştirilen lojik ifadeyi gerçekleştirecek lojik devre oluşturulur.

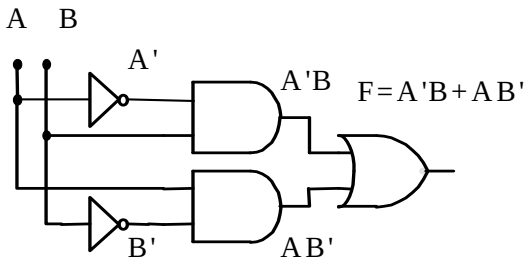
Örnek 22 : İki girişli dijital bir sistemde girişlerin farklı olduğu durumlarda çıkışın ‘1’ olmasını sağlayacak lojik devreyi tasarlayalım ve tasarlanan devreyi temel lojik elemanları ile gerçekleştirelim. Tasarımda yukarıda bahsedilen işlem sırasını takip edelim.

1. İki değişkenli doğruluk tablosu oluşturulur (Şekil 5.37.a).
2. Yapılmak istenen işin tanımından; girişlerin farklı olduğu durumlar tespit edilir.
3. Çıkışın ‘1’ olduğu kombinasyonlar yazılarak, bu kombinasyonlara ait eşitlik yazılır ve bu eşitlikler birleştirilir. Bu işlem sonucunda F=A'B+AB' eşitliği bulunur.
4. Yazılan eşitlikte sadeleştirme yapılamadığından eşitlik doğrudan lojik kapılarla gerçekleştirilir (Şekil 5.37.b).

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

F_1
F_2

$F_1 = A'B$
$F_2 = AB'$



$F=F_1+F_2=A'B+AB'$

(a) **(b)**

Şekil 5.37. Girişlerin farklı olduğu durumda çıkış veren devrenin tasarımı.

Örnek 23 : Üç girişli bir sistemde, girişlerin birden fazlasının lojik ‘1’ olduğu durumlarda çıkışın ‘1’ olmasını sağlayacak lojik devreyi, lojik tasarımda kullanılan işlem sırasına göre gerçekleştirelim.

1. Yapılması istenen iş tanımlanır. Bu işlemi gerçekleştirecek doğruluk tablosu oluşturulur.

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

2. Girişlerden birden fazlasının ‘1’ olduğu durumların tespit edilir.

3. Çıkışın ‘1’ olduğu durumları temsil eden kombinasyonlar yazılır ve yazılan kombinasyonlar birleştirilerek lojik fonksiyon elde edilir.

4. Yazılan lojik eşitlik sadeleştirilir. Sadeleştirme birkaç yolla yapılabilir.

Bulunan eşitlikte ABC grubu içerisinde bulunan değişkenlerden en az ikisinin ortak olduğu kombinasyonlarda, ortak olan değişkenler paranteze alınarak eşitlik;

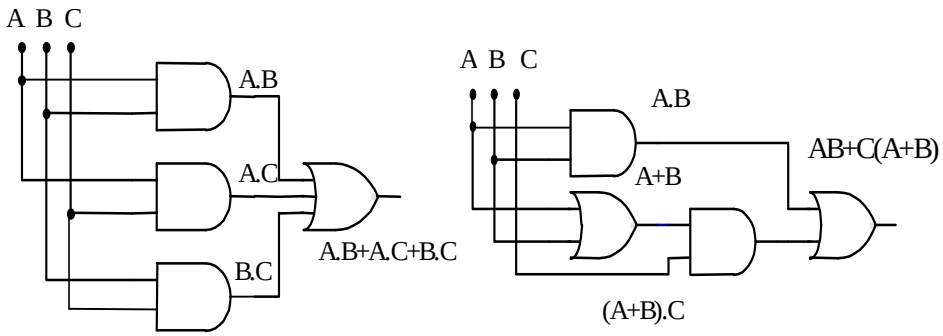
$$F=BC(\underbrace{A+A'}_{1})+AC(\underbrace{B+B'}_{1})+AB(\underbrace{C+C'}_{1})$$
$$=BC+AC+AB$$

şeklinde ifade edilebilir.

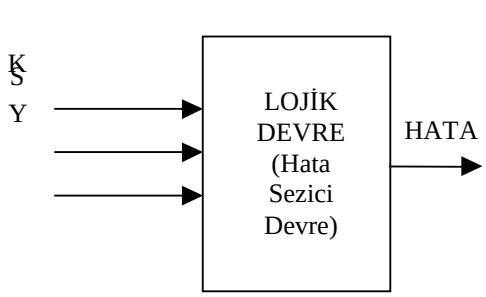
Elde edilen ifade bu şekilde gerçekleştirilebileceği gibi, ortak C parantezine alınarak;

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

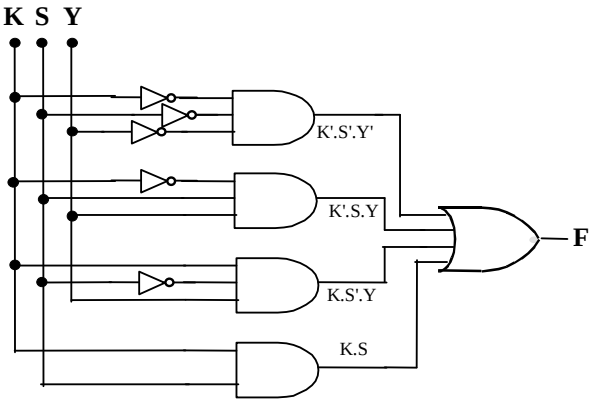
şeklinde de gerçekleştirilebilir (Şekil 5.38).



Şekil 5.38. Lojik bir devrenin tasarımlanması ve farklı şekillerde oluşturulması.



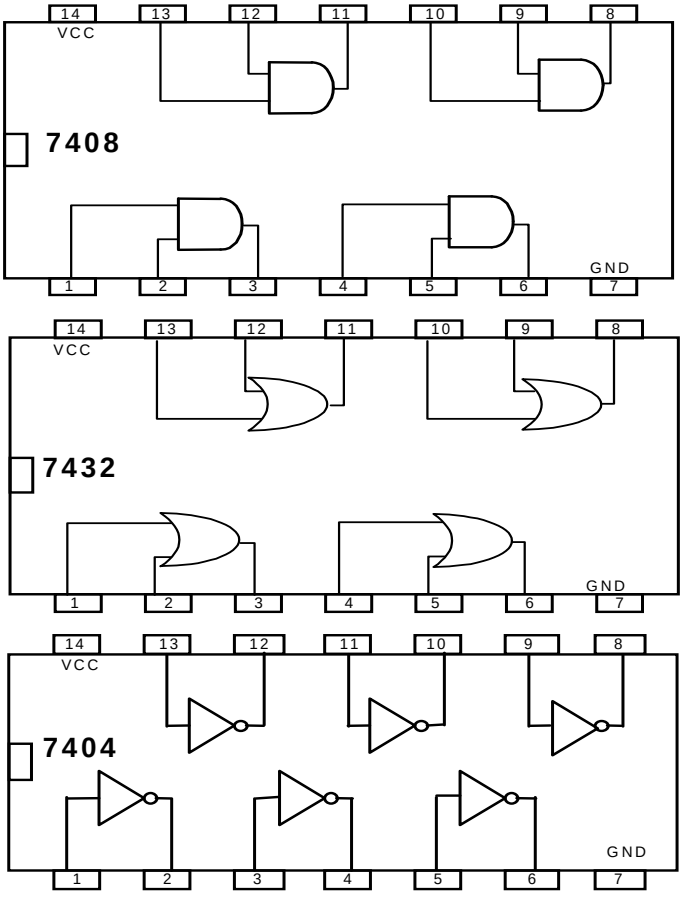
K	S	Y	HAT	
0	0	0	1	m_0= K'.S'.Y'
0	0	1	0	
0	1	0	0	
0	1	1	1	m_3= K'.S.Y
1	0	0	0	
1	0	1	1	m_5= K.S'.Y
1	1	0	1	m_6= K.S.Y'
1	1	1	1	m_7= K.S.Y



Şekil 5.39. Trafik ışık sisteminde hatalı bileşenleri sezen devrenin tasarımı ve gerçekleştirilmesi.

Örnek 24 : K (Kırmızı), Y (Yeşil), S (Sarı) bir trafik işaretindeki lambalar olsun. Bu trafik işaretinde hatalı bileşenleri sezen bir lojik devre tasarlayalım.

Oluşturulacak sistemde aynı anda yalnızca bir lambanın yanması öngörülmektedir. Lamba yanmaması durumu veya aynı anda birden fazla lambanın yanması durumu hata olarak algılanmaktadır.



Şekil 5.40. VE, VEYA ve DEĞİL kapılarının entegre içerisindeki durumları.

Hatalı bileşenleri gösteren durumlar tespit edilerek, bu durumları temsil eden bileşenler doğruluk tablosu yardımıyla yazılır. Yazılan fonksiyon kapı devreleri ile gerçekleştirilir (Şekil 5.39).

Doğruluk tablosundan elde edilen kombinasyonların fonksiyon halinde yazılması ile;

$$F= K'.S'.Y'+ K'.S.Y+ K.S'.\overbrace{Y+ K.S.Y'}+ K.S.Y$$

eşitliği oluşur. Oluşan eşitliğin sadeleştirilmesi ile;

$$F= K'.S'.Y'+ K'.S.Y+ K.S'.Y+ K.S.Y'+ K.S.Y= K'.S'.Y'+ K'.S.Y+ K.S'.Y+ K.S$$

$$KS(Y+Y')$$

$$1$$

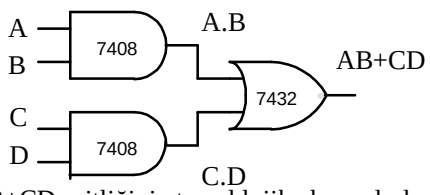
sonucu bulunur.

5.10. Lojik Kapı Entegreleri ve Temel Lojik Elemanların 'VEDEĞİL' / 'VEYADEĞİL' Kapıları İle Oluşturulması

Yapılan örneklerden görüleceği üzere, dijital sistemler 'VEYA', 'VE' ve 'DEĞİL' temel kapılarının farklı kombinasyonlarından oluşur. Lojik elemanların her biri, entegre devre içerisinde diğer elemanlarla bağlantı oluşturacak şekilde imal edilirler. Şekil 5.40'da 'VEYA', 'VE' ve 'DEĞİL' kapılarının entegre devre içerisindeki durumlarına örnekler görülmektedir.

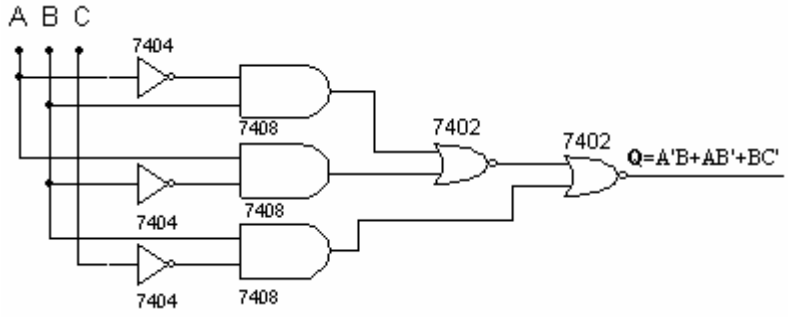
Şekil 5.40'daki entegrelerdeki elemanlar, lojik fonksiyonları gerçekleştirmede kullanılabilir.

Örnek 25 : Q = AB+CD fonksiyonunu Şekil 5.40'da entegre içerisinde yerleşimleri verilen elemanlarla oluşturalım.

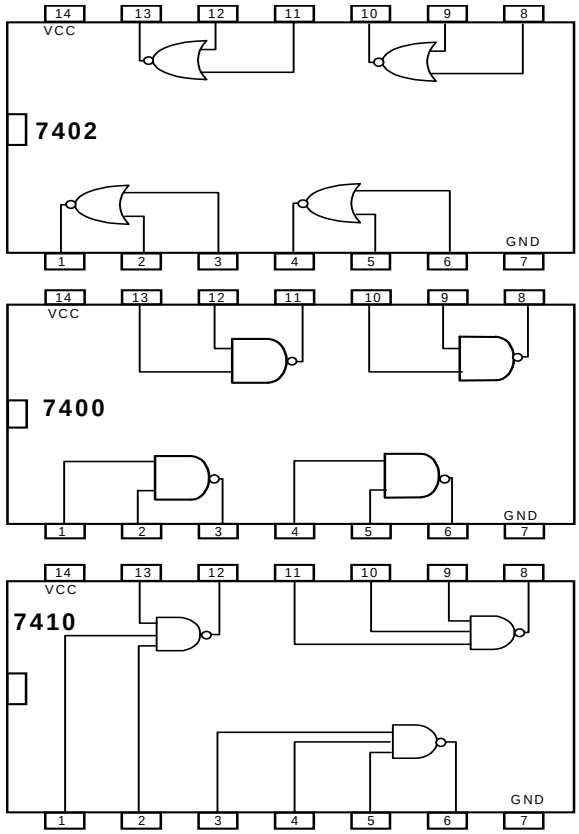


Şekil 5.41. Q = AB+CD eşitliğinin temel lojik elemanlarla oluşturulması.

Örnek 26 : Q = A'B+AB'+BC' fonksiyonunu Şekil 5.40 ve 5.43'de verilen entegre devrelerdeki elemanlarla gerçekleştirelim.

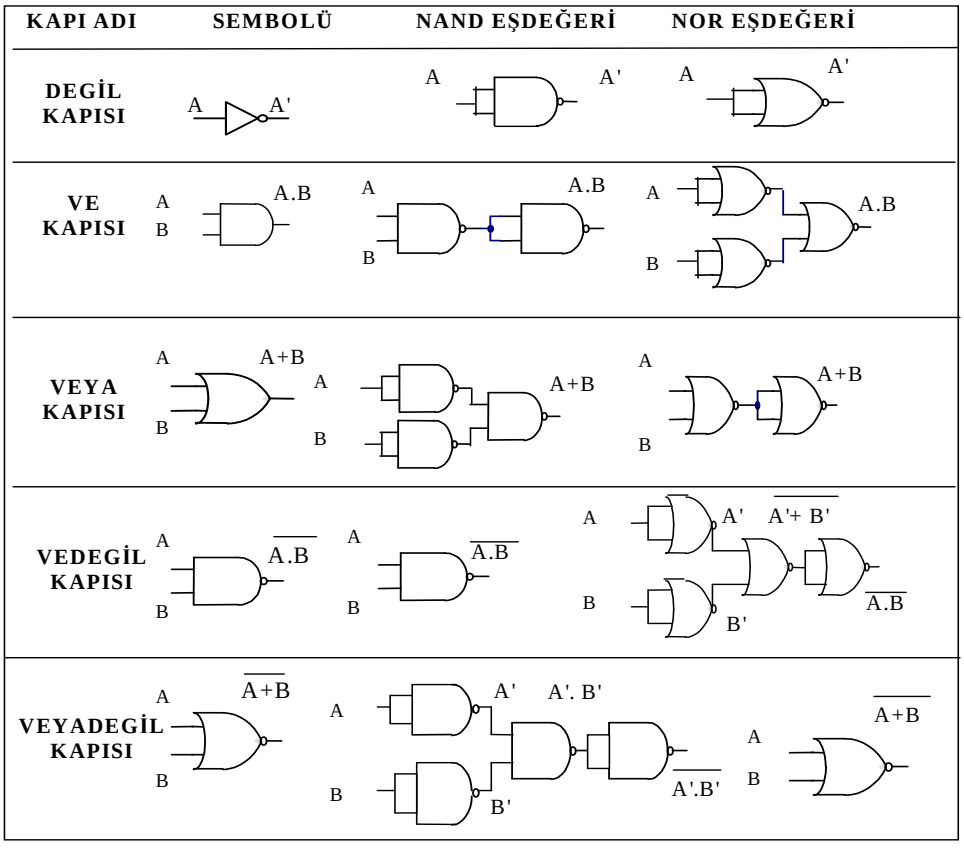


Şekil 5.42. Q =A'B+AB'+BC' fonksiyonunun temel lojik elemanlarla oluşturulması.



Şekil 5.43. ‘VEYADEĞİL’ ve ‘VEDEĞİL’ kapılarının lojik entegreler içerisindeki durumları.

Bu örneklerden görülebileceği üzere, bir lojik devreyi gerçekleştirebilmek için çok sayıda entegre devre gerekmektedir. Bu durum karmaşık devrelerde çok fazla entegre kullanımı sonucu doğurur ve büyük bir alan gereksinimi ortaya çıkar. Bu nedenden dolayı, her temel lojik eleman için bir entegre devre kullanmak ekonomik değildir. Bu düşünceden hareket edilerek, çok genel kullanımı olan iki temel lojik kapı elemanı geliştirilmiştir: ‘VEYADEĞİL’ (NOR) kapısı ile ‘VEDEĞİL’ (NAND) kapıları.



Şekil 5.44. ‘VEYADEĞİL’ ve ‘VEDEĞİL’ kapı devreleri ile temel lojik kapıların oluşturulması.

‘VEYADEĞİL’ ve ‘VEYADEĞİL’ kapılarından dört tanesi bir arada üretilmiştir. Bu kapılar, **‘VE Çeyrek / NOR Quarter’** ve **‘VEYA Çeyrek / NAND Quarter’** olarak isimlendirilir. Şekil 5.43’de ‘VEYADEĞİL’ ve ‘VEDEĞİL’ kapılarının entegre devreler içerisindeki durumları görülmektedir.

Temel lojik elemanlar olan ‘VE’, ‘VEYA’, ‘DEĞİL’, ‘VEDEĞİL’, ‘VEYADEĞİL’ kapıları, gerçekleştirilecek devrenin durumuna ve eldeki imkanlara göre ‘VEYADEĞİL’ veya ‘VEDEĞİL’ kapılarıyla oluşturulur. Şekil 5.44’de, temel lojik elemanların ‘VEYADEĞİL’ veya ‘VEDEĞİL’ kapılarıyla oluşturulması açıklanmaktadır.

Temel lojik kapıların ‘VEYADEĞİL’ ve ‘VEDEĞİL’ kapılarıyla nasıl oluşturulduğu anlamak için, temel lojik kapılardan bazılarının oluşumunu ve kapıları oluşturmadaki kabulleri inceleyelim.

i- ‘VEYADEĞİL’ veya ‘VEDEĞİL’ kapılarının girişleri birleştirilip tek bir giriş olarak kullanılırsa, bu kapılar ‘DEĞİL’ kapısına dönüşür. Çünkü; :

$(\overline{A.A}) = A'$ veya $(\overline{A+A}) = A'$ olduğundan yapılan işlem ‘DEĞİL’ kapısının özelliklerini taşır.

ii- İki ‘VEDEĞİL’ kapısıyla ‘VE’ kapısı elde edilirken, kapılardan birisi ‘DEĞİL’ kapısı olarak kullanılır. ‘DEĞİL’ kapısı; (A.B)' eşitliğinden (A.B)" eşitliğine olan değişimi gerçekleştirir. (A.B)" = A.B olduğundan, ‘VE’ kapısı 2 adet ‘VEYADEĞİL’ kullanılarak elde edilebilir.

iii- ‘VEYADEĞİL’ kapıları kullanılarak ‘VE’ kapısı elde etmek işlemi, De Morgan teoremleri kullanılarak açıklanabilir. ‘VEYADEĞİL’ kapısı (A+B)' işlemini gerçekleştirdiğinden, (A+B)' ’den (A.B) işlemi elde etmek için; (A'+B')' = (A.B) eşitliğinden faydalanılır. DEĞİL işlemini elde edebilmek için, ‘VEYADEĞİL’ kapısının iki girişi birleştirilerek tek giriş olarak kullanılır. İki ‘DEĞİL’ kapısından geçirilen değişkenler ‘VEYADEĞİL’ kapısına uygulandığında çıkışta;

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

ifadesi elde edilir.

Temel lojik kapıların ‘VEYADEĞİL’ veya ‘VEDEĞİL’ kapılarıyla oluşturulması gibi, lojik fonksiyonlar da ‘VEYADEĞİL’ ve ‘VEDEĞİL’ kapıları ile gerçekleştirilebilir.

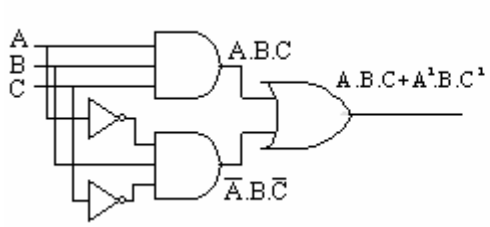
Fonksiyonların ‘VEYADEĞİL’ ve ‘VEDEĞİL’ kapıları ile gerçekleştirilmesi işleminde; ‘çizim yöntemi’ ve ‘matematiksel yöntem’ olarak isimlendirilebilecek iki farklı yöntem kullanılır.

Çizim yöntemi :

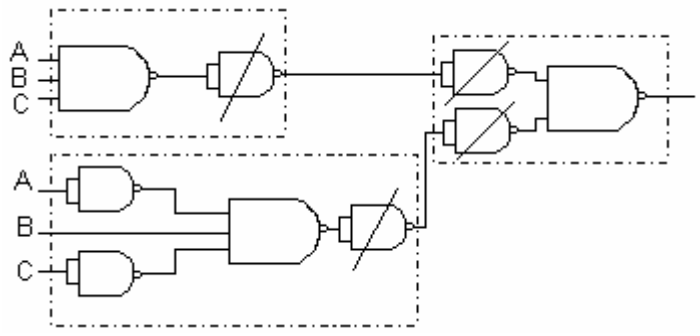
Lojik eşitlikte bulunan tüm elemanların ‘VEYADEĞİL’ veya ‘VEDEĞİL’ ile gerçekleştirilmesini sağlayacak şekilde lojik devre çizilir. Çizilen lojik devrede, iki ‘DEĞİL’ işleminin ard-arda yapılması benzeri işlemler silinerek, devre sadeleştirilir. Oluşan devre, sadece ‘VEDEĞİL’ ve ‘VEYADEĞİL’ kapıları ile gerçekleştirilmiş olur. Açıklanan bu yöntem, ‘**çizim yöntemi ile sadeleştirme**’ olarak isimlendirilir.

Örnek 27 : $F=(A.B.C)+(A^1.B.C^1)$ lojik ifadesini ‘VEDEĞİL’ kapıları ile gerçekleştirelim.

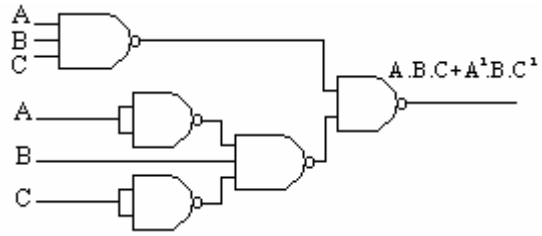
Lojik eşitliği temel lojik kapılar ile gerçekleştirip (Şekil 5.45.a), her bir kapının işlevini gerçekleştirecek ‘VEDEĞİL’ kapıları ile oluşturulan devreyi ilgili kapı yerine koyarsak Şekil 5.45’ deki lojik şema oluşur. Oluşan şemada, ard-arda bağlı ‘DEĞİL’ kapılarını kaldırarak devreyi sadeleştirirsek lojik eşitliği ‘VEDEĞİL’ kapıları ile oluşturmuş oluruz (Şekil 5.45.c).



a)



b)

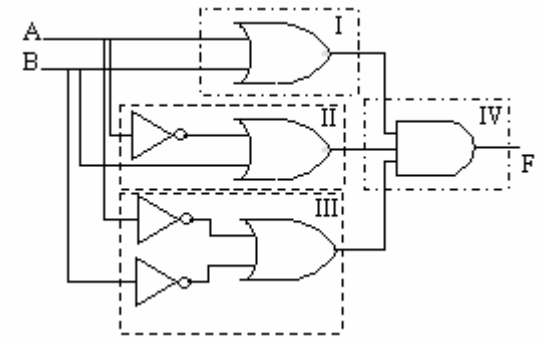


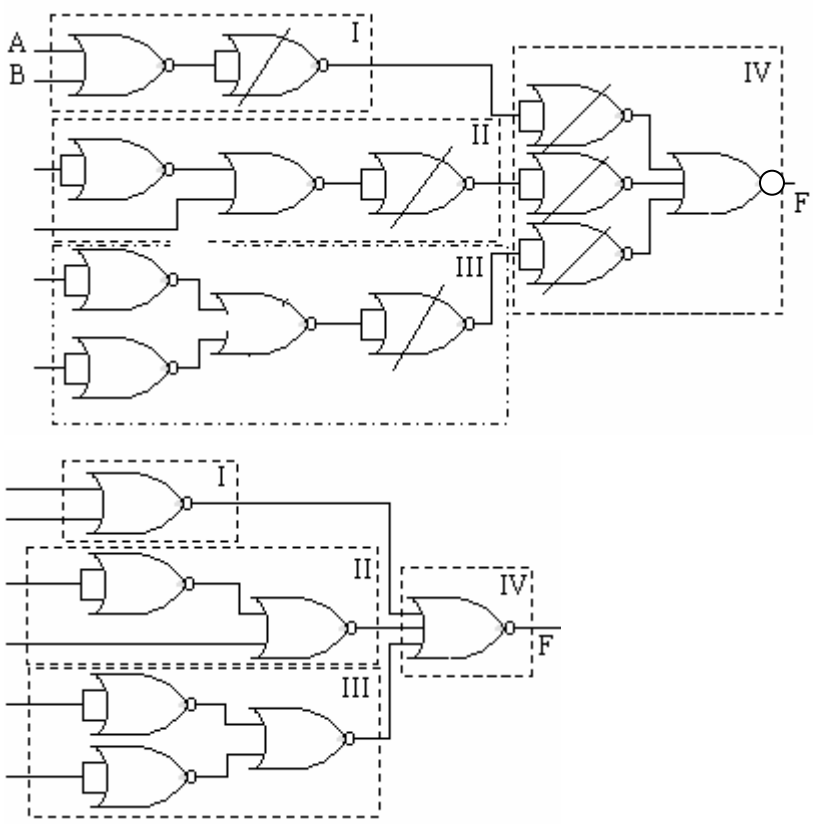
c)

Şekil 5.45 'VEDEĞİL' kapıları kullanılarak, çizim yöntemi ile sadeleştirme kullanılarak lojik eşitliklerin gerçekleştirilmesi.

Örnek 28 : F=(A+B).(A^{1}+B).(A^{1}+B^{1}) lojik eşitliğini 'VEYADEĞİL' kapıları kullanarak çizim yöntemi ile sadeleştirerek gerçekleştirelim.

Temel Lojik kapılar ile oluşturulan devredeki her bir kapının yerine, eşdeğeri olan 'VEYADEĞİL' kapıları ile oluşturulan devreyi yerleştirip, fazla olan kapıları üzerlerini çizerek kaldırırsak, eşitliği 'VEYADEĞİL' kapıları ile gerçekleyecek lojik devreyi elde ederiz (Şekil 5.46).





Şekil 5.46. Lojik eşitliklerin çizim yöntemi ile sadeleştirme kullanılarak 'VEYADEĞİL' kapıları ile gerçekleştirilmesi.

Matematiksel yöntem :

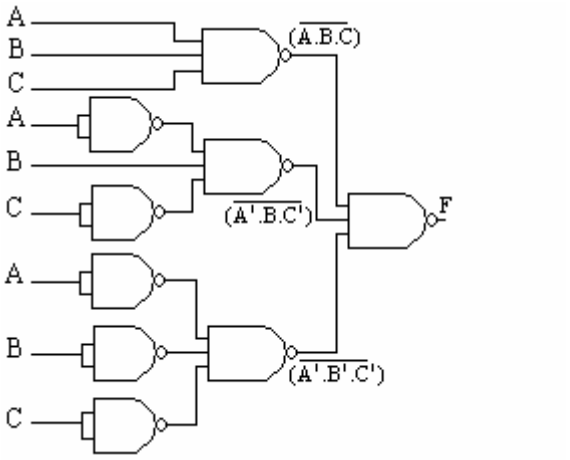
Lojik eşitliklerin, 'VEDEĞİL' kapıları ile gerçekleştirilecekse çarpım, 'VEYADEĞİL' kapıları ile gerçekleştirilecekse toplama işlemleri ile ifade edilecek şekle dönüştürüldüğü yöntem, '**matematiksel yöntem ile sadeleştirme**' olarak isimlendirilir.

Çarpımların toplamı şeklinde ifadelerin, çarpımlar şekline dönüştürülmesi için iki kere değili alınır. Demorgan kuralı uygulanarak, toplama işlemi çarpımlar haline getirilir.

Örnek 29 : $F=A.B.C+A^{1}.B.C^{1}+A^{1}.B^{1}.C^{1}$ ifadesini 'VEDEĞİL' kapıları ile gerçekleştirmek için gerekli sadeleştirmeyi yapalım.

$$F = A.B.C+A^{1}.B.C^{1}+A^{1}.B^{1}.C^{1} = \overline{\overline{A.B.C+A^{1}.B.C^{1}+A^{1}.B^{1}.C^{1}}}$$
$$= \overline{\overline{(A.B.C)}.\overline{(A^{1}.B.C^{1})}.\overline{A^{1}.B^{1}}.C^{1}}$$

Eşitlikteki tüm ifadelerin çarpımlar şeklinde ifade edilmesinden sonra devre 'VEDEĞİL' kapıları ile gerçekleştirilir.

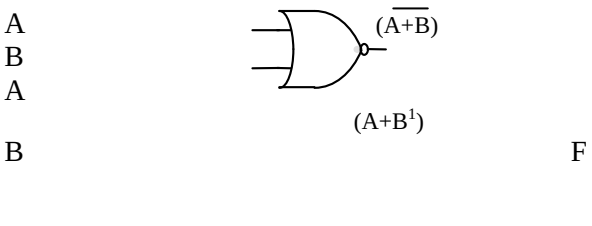


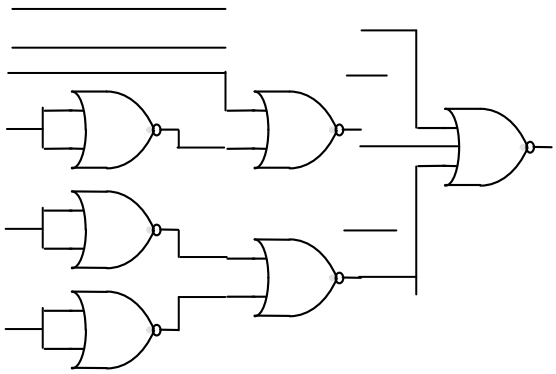
Şekil 5.47. Lojik eşitliklerin 'VEDEĞİL' kapıları ile gerçeklenmesi.

Örnek 30 : $F = (A+B).(A+B^{I})(A^{I}+B^{I})$ ifadesini 'VEYADEĞİL' kapıları ile gerçekleştirecek formata getirelim.

$$F = (A+B).(A+B^{I})(A^{I}+B^{I}) = \overline{\overline{(A+B).(A+B^{I})(A^{I}+B^{I})}}$$
$$= \overline{\overline{(A+B)}+\overline{(A+B^{I})}+\overline{(A^{I}+B^{I})}}$$

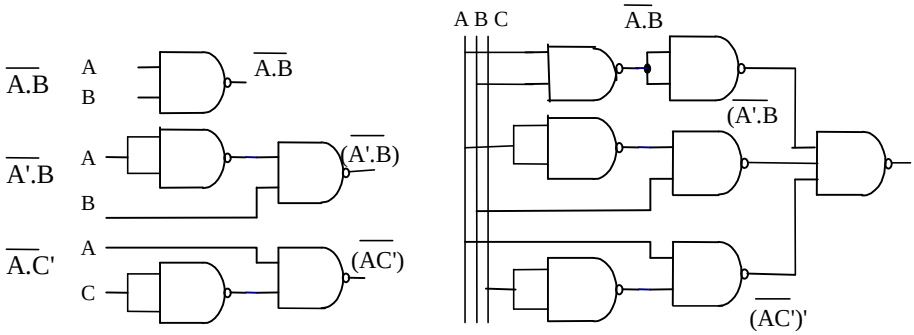
elde edilen eşitlikteki tüm işlemler 'VEYADEĞİL' kapıları ile gerçekleştirilecek formattadır (Şekil 5.48).





Şekil 5.48 Lojik Eşitliklerin ‘VEYADEĞİL’ kapıları ile gerçekleştirilmesi.

Örnek 31 : Q = (AB)'+A'B+AC' fonksiyonunu ‘VEDEĞİL’ kapılarıyla gerçekleştirelim.



$$\overline{(A.B)'' (A'.B)' (AC')'} = (A.B)'+(A'.B)''+ (AC')''$$
$$= (A.B)'+(A'.B)+ (AC')$$

Şekil 5.49. Eşitliklerin ‘VEDEĞİL’ kapılarıyla oluşturulması.

Örnek 31 : Üç bitlik oktal bir kod için çift parity çıkışı veren bir devreyi tasarlayarak VEYADEĞİL kapılarıyla gerçekleştirelim.

Tasarlanan devre VEYADEĞİL’lerle gerçekleştirileceği için, doğruluk tablosunda Maxtermleri yazmak daha pratiktir.

Doğruluk tablosu çıkış sütunundaki değerlerin fonksiyon haline getirilmesi için; maxtermler yazılır ve maxtermler ‘VEYADEĞİL’ kapıları ile gerçekleştirilir (Şekil 5.50).

Örnek 33 : Üç anahtar kullanılan bir sistemde anahtarlardan herhangi ikisinin (yalnızca iki) aynı anda kapatılması durumunda bir lambanın yanmasını sağlayacak devreyi ‘VEDEĞİL’ kapıları ile gerçekleştirelim.

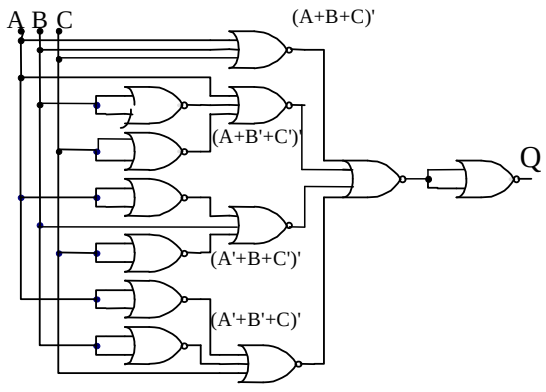
İstenilen işlemin doğruluk tablosu oluşturulur ve çıkış sütununda ‘1’ olması istenilen kombinasyonlar belirlenir (Şekil 5.51). Çıkış sütununda ‘1’ olacağı belirlenen kombinasyonlara ait fonksiyonlar yazılır. Yazılan fonksiyon ‘VEDEĞİL’ kapıları ile gerçekleştirilerek tasarım bitirilir.

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

$Q=\Pi\ (0,3,6,7)=(A+B+C).(A+B'+C').(A'+B+C').(A'+B'+C)$

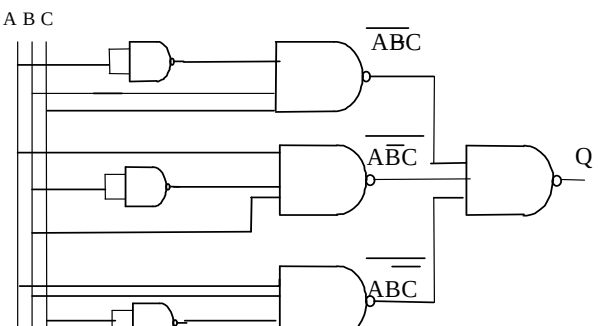
$\overline{Q'=(A+B+C).(A+B'+C').(A'+B+C').(A'+B'+C)}$

$Q'=(A+B+C)'+(A+B'+C')'+(A'+B+C')'+(A'+B'+C)\ '$



Şekil 5.50. Fonksiyonların ‘VEYADEĞİL’ kapılarıyla oluşturulması.

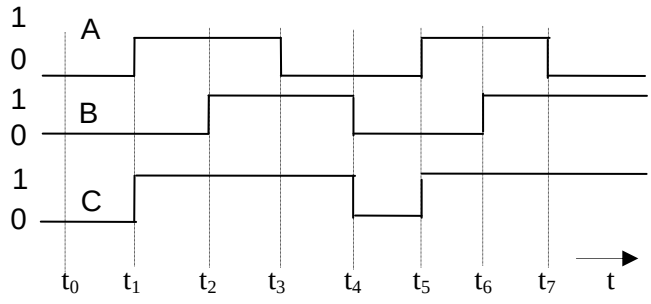
A	B	C	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



Şekil 5.51. Tasarlanan devrenin ‘VEDEĞİL’ kapılarıyla oluşturulması.

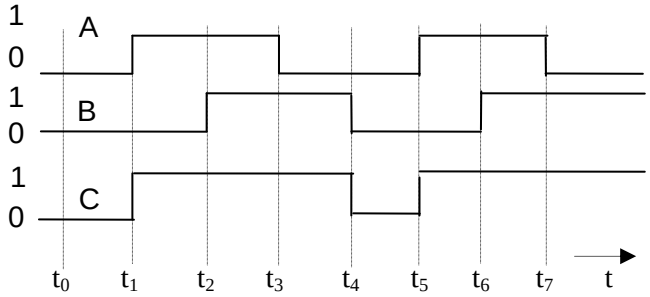
Tekrarlama ve Çalışma Soruları

1. Temel lojik kapıların isimlerini ve sembollerini sıralayınız.
2. Temel lojik kapıların doğruluk tablolarını oluşturunuz.
3. 3 girişli ‘VEYA’ kapısının sembolünü ve elektriksel eş değerini çiziniz.
4. ‘VEYA’ kapısının sanayide kullanımını bir örnekle açıklayınız.
5. Şekildeki sinyallerin üç girişli ‘VEYA’ kapısına uygulanması durumunda çıkışta oluşacak dalga şeklini çiziniz.

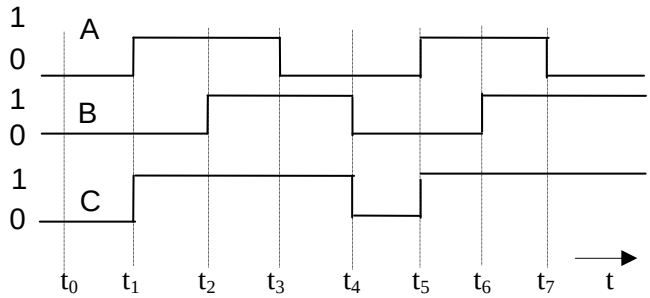


6. ‘VE’ kapısı sembolünü, doğruluk tablosunu ve elektriksel eş değerini çiziniz.
7. Şekildeki sinyallerin üç girişli ‘VE’ kapısına uygulanması durumunda çıkışta oluşacak dalga şeklini çiziniz.

8. ‘DEĞİL’ işleminin sembolünü, doğruluk tablosunu ve elektriksel eşdeğerini çiziniz.
9. ‘VEDEĞİL’ kapısının sembolünü, doğruluk tablosunu ve elektriksel eşdeğerini gösteriniz.
10. Şekildeki sinyallerin 3 girişli ‘VEDEĞİL’ kapısına uygulanması durumunda oluşacak dalga şeklini çiziniz.



11. ‘VEYADEĞİL’ kapısı sembolünü, doğruluk tablosunu ve elektriksel eşdeğerini çiziniz.
12. Şekildeki sinyallerin üç girişli ‘VEYADEĞİL’ kapısına uygulanması durumunda oluşacak dalga şeklini çizelim.



13. ‘EXOR’ kapısı sembolünü doğruluk tablosunu ve elektriksel eşdeğerini çiziniz.
14. ‘EXNOR’ kapısı sembolünü doğruluk tablosunu ve elektriksel eşdeğerini çiziniz.
15. İki yönlü anahtarın sembolünü elektriksel eş değerini ve doğruluk tablosunu çiziniz.

16. Lojik devreyi tarif ediniz.
17. Lojik bağlantıları tanımlayınız.
18. F = ABC' + A'BC + ABC fonksiyonunu kapı devreleri ile gerçekleştiriniz
19. F = ABCD' + A'BC'D + A'B'CD + A'BCD + AB'C'D' fonksiyonunu kapı devreleri ile gerçekleştiriniz.
20. F = A'B'C'D' + A'BC'D + A'B'CD + A'BC'D fonksiyonunu kapı devreleri ile gerçekleştiriniz.
21. F=(A+B'+C').(A'+B'+C).(A'+B+C) ifadesini lojik kapılar ile gerçekleştiriniz.
22. Lojik devre tasarımında takip edilmesi gerekli işlem basamaklarını sıralayınız.
23. Girişlerin aynı olması durumunda, çıkışında '1' üretecek lojik devreyi tasarlayarak, kapı devreleri ile gerçekleştiriniz.
24. Üç girişli bir sistemde, girişlerin en az ikisinin '0' olması durumunda çıkışında '1' üretecek devreyi tasarlayınız.
25. 'NOR – Quarter' ve 'NAND – Quarter' terimlerini açıklayınız.
26. 'NAND' kapıları kullanarak; 'DEĞİL', 'VEYA', 'VE', 'VEDEĞİL' kapılarını gerçekleştiriniz.
27. 'NOR' kapıları kullanarak; 'DEĞİL', 'VEYA', 'VE', 'VEDEĞİL' kapılarını gerçekleştiriniz.
28. Q = A'B + A'B' + AB ifadesini 'NAND' kapıları ile gerçekleştiriniz.
29. Q = AB'C + ABC' + ABC ifadesini 'NAND' kapıları ile gerçekleştiriniz.
30. Q=ABCD'+ A'BC'D + AB'C'D ifadesini 'NOR' kapıları ile gerçekleştiriniz.
31. Üç bitlik oktal bir kod için, tek parity çıkışı veren lojik devreyi tasarlayarak 'VEDEĞİL' kapıları ile gerçekleştiriniz.
32. Üç anahtar kullanılan bir sistemde, anahtarlardan herhangi ikisinin (yalnızca 2) kapalı olması ('0' olması) durumunda, çıkışında '1' oluşmasını sağlayacak devreyi 'VEYADEĞİL' kapı devreleri kullanarak gerçekleştiriniz.
33. $Q_{(ABCD)} = \Sigma (4,8,12,14,15)$ ifadesini 'VEDEĞİL' kapıları ile gerçekleştiriniz.
34. $Q_{(ABCD)} = \Pi(3,6,9,12,15)$ ifadesini 'VEYADEĞİL' kapıları ile gerçekleştiriniz.

35. Trafik sinyalizasyon sisteminde hatalı bileşenleri sezen lojik devreyi ‘VEYADEĞİL’ kapıları ile gerçekleştiriniz.

BÖLÜM 6

KARNAUGH (KARNO) HARİTALARI

(KARNAUGH MAPS)

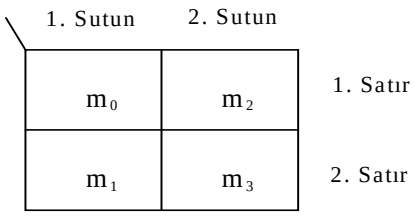
Giriş

Devre tasarımında lojik eşitlikleri oluşturmak veya oluşturulan lojik eşitlikleri grafiksel olarak sadeleştirmek için yaygın olarak kullanılan yöntemler; **'Karnaugh Haritası'** (Karnaugh Maps) ve '**Quine-McCluskey'** yöntemleridir. Bu bölümde Karnaugh haritası yöntemini açıklayacağız.

'Karnaugh haritası' (Karno çizelgesi), sadeleştirilecek eşitliğin bütün değerlerini sıralamak için kullanılan, eşitliğin alabileceği en basit (sade) şekli içeren, hücrelerin oluşturduğu bir yöntemdir. Giriş değişkenlerinin sayısı artıkça ifadelerin sadeleştirilmesinin zorlaştığı bu yöntem, giriş değişkenleri sayısının 6'ya kadar olduğu durumlarda iyi bir sonuç verir. Genelde kullanılan; 2, 3 ve 4 giriş değişkenli Karnaugh haritalarıdır (çizelgeleridir).

Hücrelerin kullanıldığı bu yöntemde, her hücre bir değer ifade eder. Bir çizelgedeki hücre sayısı 2^n ifadesiyle bulunur (n=değişken sayısı). Bu durumda iki değişkenli bir sistemde hücre sayısı $2^2=4$, üç değişkenli bir sistemde hücre sayısı $2^3=8$ olur.

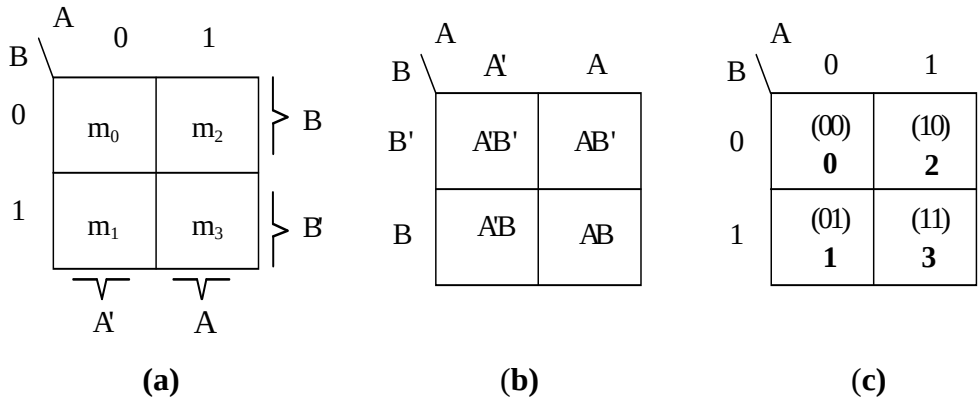
Karnaugh haritası oluşturulurken ortaya çıkan düşey doğrultuda bulunan hücrelere '**kolon veya sütun'**, yatay doğrultuda bulunanlara **'satır'** ismi verilir (Şekil 6.1).



Şekil 6.1. Karnaugh haritasında hücrelerin oluşturulması.

6.1. İki, Üç ve Dört Değişkenli Karnaugh Haritaları

Karnaugh haritasında bulunacak hücre sayısının 2^n (n = değişken sayısı) formülüyle belirlenmesi nedeni ile bulunur. Bu durumda, iki değişkenli Karnaugh haritası 2^2=4 hücre içerir. Hücrelerin her birisi, doğruluk tablosunda bulunan kombinasyonlardan (örneğin mintermlerden) birisine karşılık gelir. Hücrelerin ifade ettikleri minterm değerleri belirli bir sistematiğe göre belirlenir (Şekil 6.2):

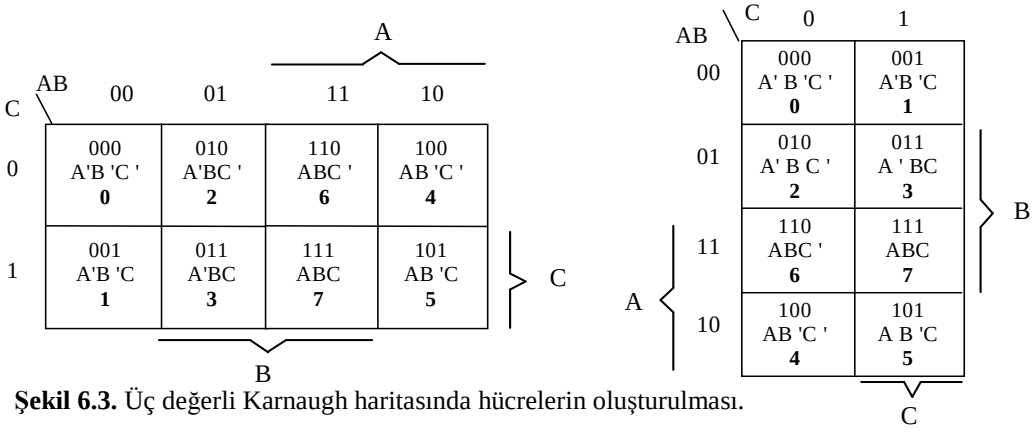


Şekil 6.2. İki değişkenli Karnaugh haritasında hücrelerin anlamları.

Karnaugh haritasının sol üs köşesi Şekil 6.2'deki gibi eğik bir şekilde çizilerek, bağımsız değişkenlerin isimleri olan A,B,C,...vb. harfler yazılır. Değişkenlerin alabileceği değerler (0 veya 1) sırasıyla yazılırsa, Şekil 6.2.a'daki durum oluşur. Değişken olarak A ve B kullanılırsa; kolonlarda A=0 veya A=1, satırlarda B=0 veya B=1 değerleri temsil edilir. Bu kabullere göre hücrelere temsil ettikleri kombinasyonlar yazılırsa, Şekil 6.2.b'deki ifadeler elde edilir.

İki değişkenli doğruluk tablosundaki hücrelerde oluşan kombinasyonların karşılıkları olan onlu ikili değerlerin hücrelerin içerisine yerleştirilmesi ile, Şekil 6.2.c'deki hücre değerleri oluşur.

Aynı prensiplere uyularak oluşturulacak üç değişkene sahip Karnaugh haritasında 2^3=8 hücre bulunur ve iki farklı yerleştirme durumu ortaya çıkar. Şekil 6.3'de gösterilen her iki yerleştirme şeklide doğrudur ve daha sonraki aşamalarda her iki tablodan elde edilecek sonuç aynı olur.



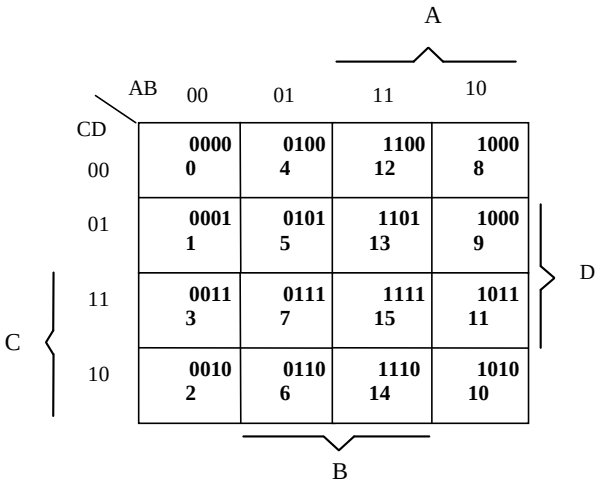
Şekil 6.3. Üç değerli Karnaugh haritasında hücrelerin oluşturulması.

Dört değişkenli Karnaugh haritasında 2^4=16 hücre bulunur. Değişkenlerin ikisi yatay, ikisi dikey eksende belirtilir. Dört değişkenli Karnaugh haritasında oluşan hücreler, hücrelerin temsil ettikleri ikili kombinasyonlar ve her hücrenin onlu karşılığı Şekil 6.4'de görülmektedir.

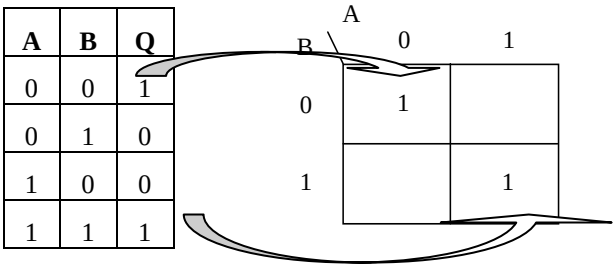
Karnaugh haritasındaki hücrelerin ifade ettikleri anlamları belirledikten sonra yapılması gerekli işlem, doğruluk tablosundaki bilgilerin Karnaugh haritasına aktarılmasıdır. Bilgilerin aktarılması işleminde, doğruluk tablosunda çıkışın '1' olduğu durumlar Karnaugh haritasındaki hücrelere taşınır. Taşınma işlemi, doğruluk tablosunda çıkışın '1' olduğu kombinasyonları temsil eden hücrelere '1' değerinin yazılması şeklinde yapılır (Şekil 6.5).

Sonuçları '0' olan ifadeler bir sonraki işlem aşaması olan gruplandırmada göz önünde bulundurulmadığı için, çizelgeye yazmaya gerek yoktur.

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	



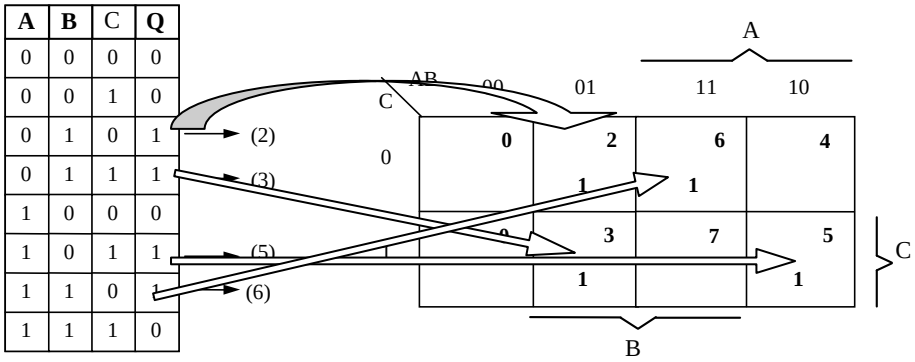
Şekil 6.4. Dört değişkenli doğruluk tablosu ve Karnaugh haritasına yerleştirilmesi.



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

Şekil 6.5. Doğruluk tablosundaki değerlerin Karnaugh haritasındaki hücrelere taşınması.

Üç değişkenli işlemlerde de doğruluk tablolarındaki ‘1’ değerleri Karnaugh haritalarında temsil edildikleri hücrelere taşınırlar (Şekil 6.6). Karnaugh haritasının dışında grup işareti ile belirtilen ve değişkenler ile gösterilen satır ve sütunlar; ilgili değişkenin ‘1’ olarak temsil edildiği satırları veya sütunları göstermektedir.



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

Şekil 6.6. Üç değişkenli doğruluk tablosundaki değerlerin Karnaugh haritasına taşınması.

6.2. 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 alınması işlemine, **‘gruplandırma’** denir. Gruplandırmada ve lojik ifadelerin oluşturulmasında takip edilecek sıra ve dikkat edilecek kurallar aşağıdaki gibi özetlenebilir:

i-Yan yana veya alt alta bulunan bir, iki veya ikinin kuvveti sayıdaki hücreler gruplandırılabilir. (2^0=1, 2^1=2, 2^2=4, 2^3=8, 2^4=16,……).

ii- Her bir gruba farklı bir isim verilir.

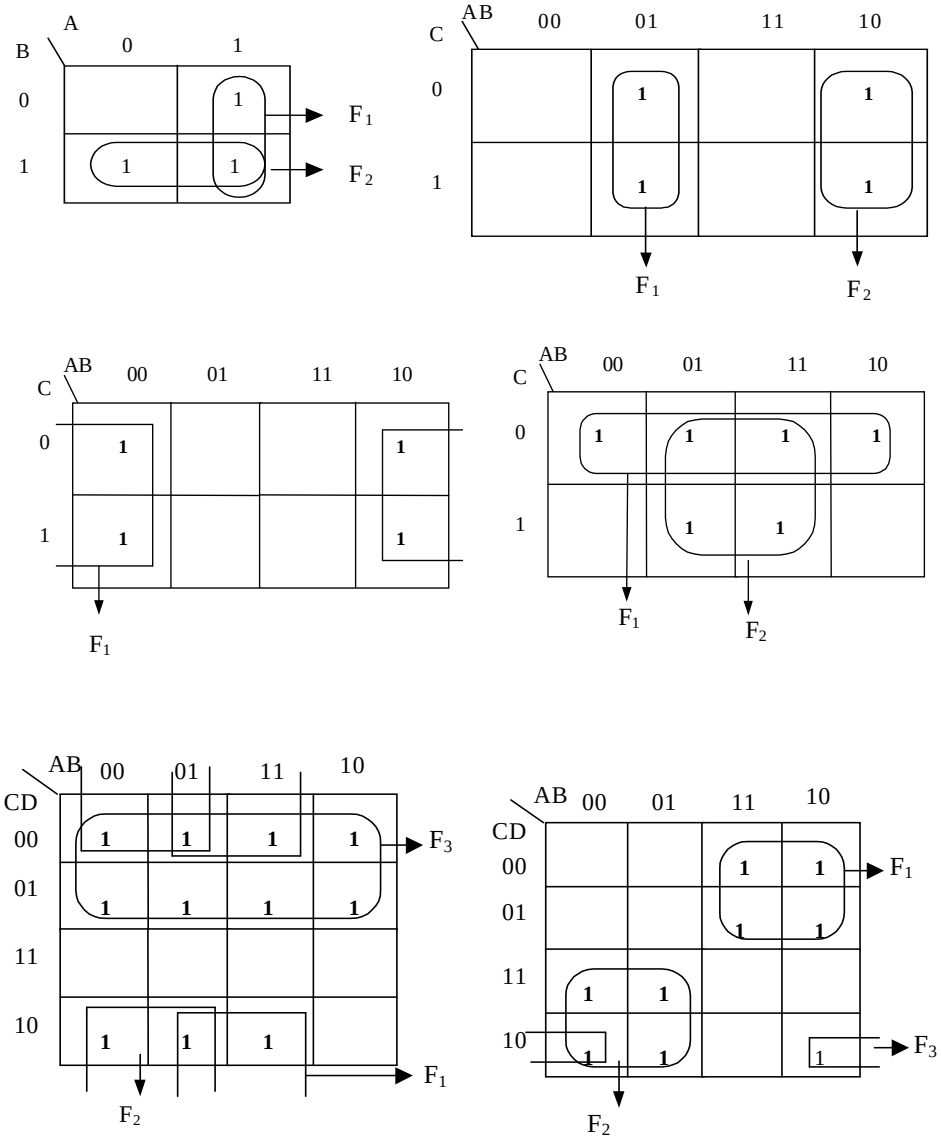
iii- Herhangi bir gruba girmiş olan ‘1’, başka bir gruba girebilir. Bu işlem sonucun daha kısalmasına yardımcı olur (Şekil 6.7).

iv-Karnaugh çizelgesini sağa sola veya yukarı aşağı bükecek olursak, çizelge silindirik bir şekle dönüşebilir. Bu durumda çizelgenin alt ve üst hücrelerinde bulunan veya başta ve sondaki hücrelerde olan 1 değerleri bitişik sayılabileceğinden gruplandırma yapılabilir.

v- İki değişkenli Karnaugh’da aynı grup içerisinde dört adet, üç değişkenli Karnaugh’da sekiz adet ‘1’ olması durumunda fonksiyon sonucu ‘1’ olur.

vi- Oluşturulan grupların ifade ettikleri kombinasyonlar, grubun bulunduğu kolon(lar) ve satır(lar)da hücreler boyunca değişim göstermeyen

değişkenler alınarak oluşturulur. Değişim gösteren değişkenler ise göz ardı edilir (Şekil 6.8) .



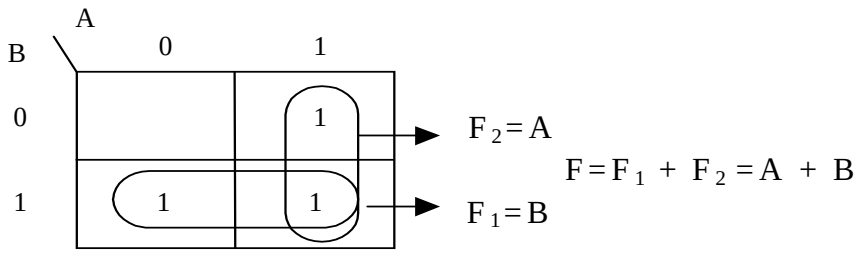
Şekil 6.7. Karnaugh haritasında gruplandırma örnekleri.

Örnek 1: Şekil 6.8'de verilen iki değişkenli Karnaugh'da bulunan değerleri gruplandırarak, gruplara ait eşitlikleri yazalım.

Yukarıda özetlenen işlemler sırası ile gerçekleştirilerek F_1 ve F_2 olarak isimlendirilen iki grup oluşturulur. Grupların oluşturup, oluşturulan grupların isimlendirilmesinden sonra yapılacak işlem, her bir grubun temsil ettiği eşitliği yazmaktır.

F_1'in ifade ettiği kombinasyon yazılırken A'nın aldığı değerlere bakılır. Grubun bulunduğu kolonlarda A; hem 0, hemde 1 değerini aldığından 'A' değişkeni yazılmaz. Grubun bulunduğu satırda B'nin aldığı değer değişmediğinden ve satır '1' değerini temsil ettiğinden; F_1=B olarak elde edilir.

Aynı şekilde F_2'nin bulunduğu kolonda 'A' yalnızca 1 değerini aldığından F_2 = A olarak yazılır. Grup her iki satırda bulunduğundan ve 'B' değişkeni hem '0', hemde '1' değerlerini aldığından B değişkeni yazılmaz. Ayrı ayrı yazılan bu grup değerleri toplanırsa, sadeleştirilmiş eşitlik elde edilir.



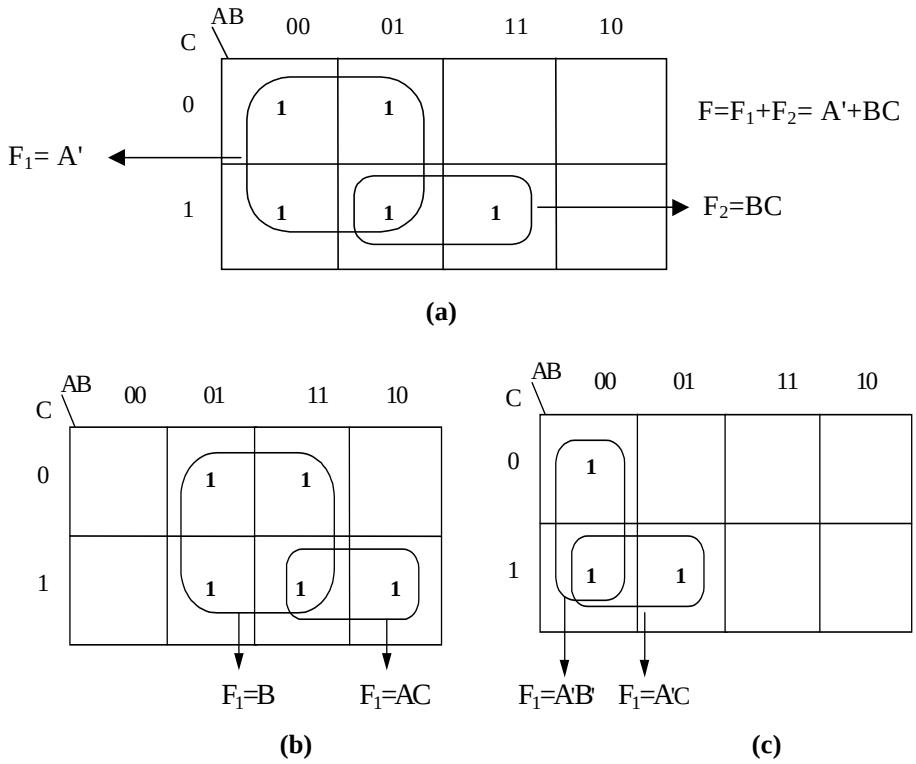
Şekil 6.8. Karnaugh haritasında gruplandırmalardan fonksiyonların yazılması.

Örnek 2: Şekil 6.9'da görülen 3 değişkenli Karnaugh haritalarında oluşturulan gruplara ait eşitlikleri bulalım ve sonuç eşitliğini yazalım.

Şekil 6.9.a'daki Karnaugh'da F_1'in ifade ettiği fonksiyon yazılırken; grubun bulunduğu sütunlarda A değeri değişim göstermediğinden ve değişim göstermeyen değer '0' olduğundan fonksiyon A' olarak oluşur. Grubun bulunduğu ve B'nin temsil edildiği sütunlarda B değeri '0' ve '1'' değerlerine sahip olduğundan B değişkeni yazılmaz. Grup her iki satırı kapladığından ve C

değişkenlerinin değerleri değişim gösterdiğinden C değişkeni de eşitlikte belirtilmez. Bu durumda, F_1=A' eşitliği oluşur.

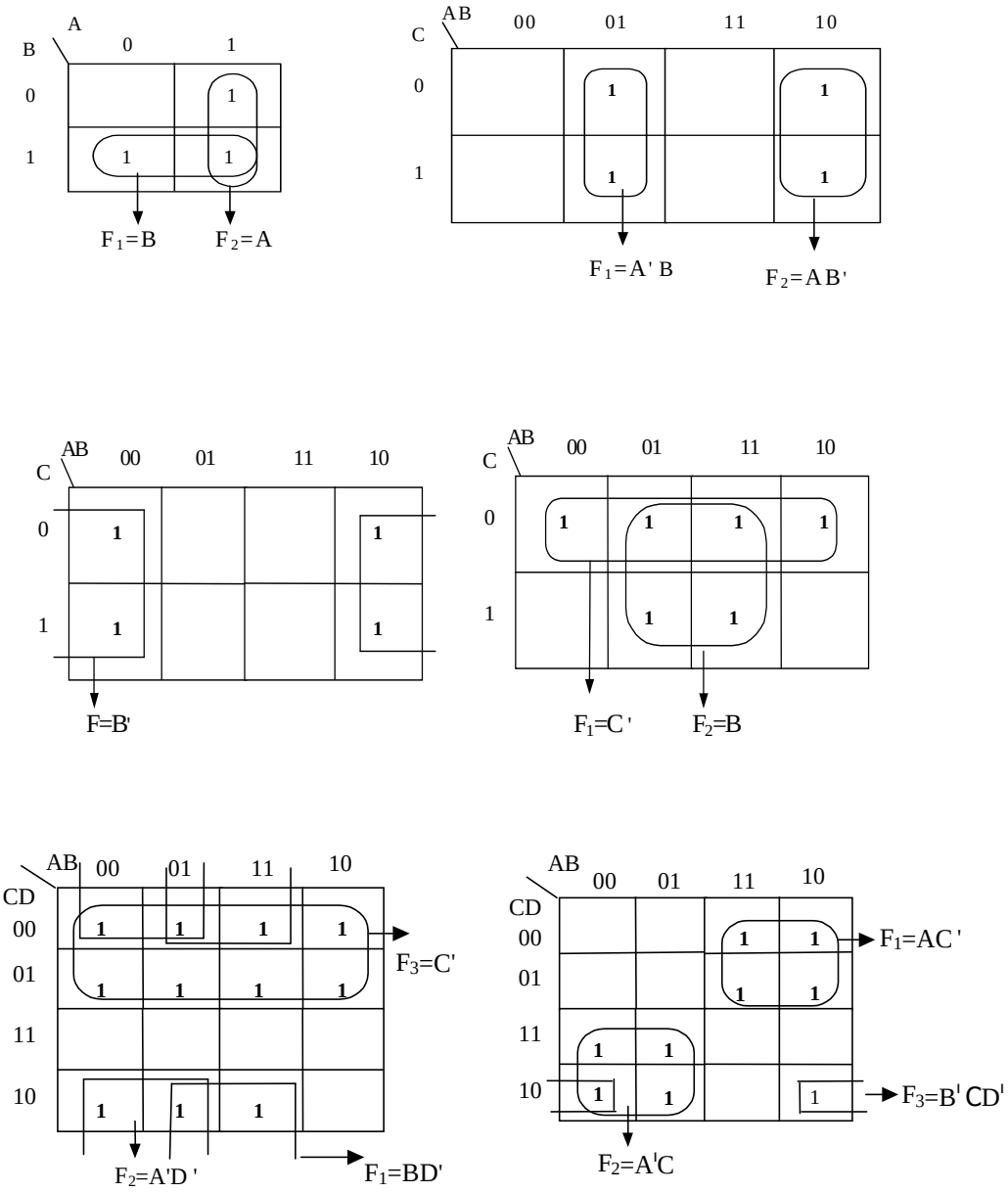
F_2 grubunun bulunduğu sütunlarda B, satırlarda C değişim göstermeyen değişken olduğundan, bu değişkenler çarpım şeklinde yazılarak eşitlik oluşturulur. F_2'nin eşitliği, F_2 = BC olarak bulunur. Bulunan eşitlikler toplanırsa, Karnaugh şemasının temsil ettiği eşitlik oluşur. Diğer Karnaugh'larda da aynı prensiplere uyularak gruplar oluşturulur.



Şekil 6.9. Üç değişkenli Karnaugh haritasında gruplandırma yapılması ve eşitlik yazılması.

Örnek 3: Şekil 6.7'de gösterilen Karnaugh'lardaki gruplara ait eşitlikleri yazalım.

Herbir Karnaugh'da yapılan gruplara ait eşitlikler yazılırsa, Şekil 6.10'daki eşitlikler elde edilir.



Şekil 6.10. Karnaugh haritasında yapılan gruplardan eşitliklerin yazılması.

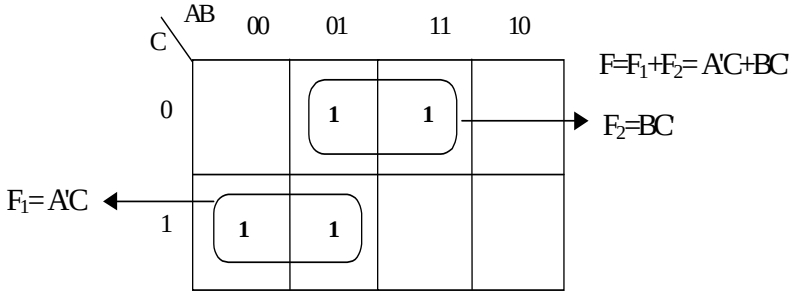
6.3. Karnaugh Haritası Kullanarak Boolean Eşitliklerin Sadeleştirilmesi

Karnaugh haritasının yaygın kullanım yerlerinden birisi, Boolean eşitliklerinin sadeleştirilmesidir. Sadeleştirme işlemi için Boolean eşitliğindeki değişken sayısına uygun Karnaugh haritası çizildikten sonra, eşitlikteki her bir mintermin temsil ettiği hücreye '1' yazılır. Örneğin; A'BC kombinasyonunun karşılığının 011, AB'C' nin karşılığının 100 olduğu gibi.

Eşitlikteki kombinasyonların temsil ettiği değerlerin karşılığı olan hücrelere '1' yazılması suretiyle taşınma işlemi bitirilir. Taşınma işlemi bitirildikten sonra gruplandırmalar yapılır. Gruplara ait eşitlikler yazılıp, bu eşitliklerin tek bir eşitlikte mintermlerin toplamı şeklinde yazılması ile sadeleştirme işlemi bitirilir.

Örnek 4: F=A'BC+A'B'C+ABC'+A'BC' lojik eşitliğini sadeleştirelim.

Her bir mintermin temsil ettiği Karnaugh haritasındaki hücreye '1' yazılması ile Şekil 6.11'deki Karnaugh elde edilir. Karnaugh haritasındaki 1'ler gruplandırılırsa; F_1 ve F_2 olarak ifade edilen iki grup ortaya çıkar.



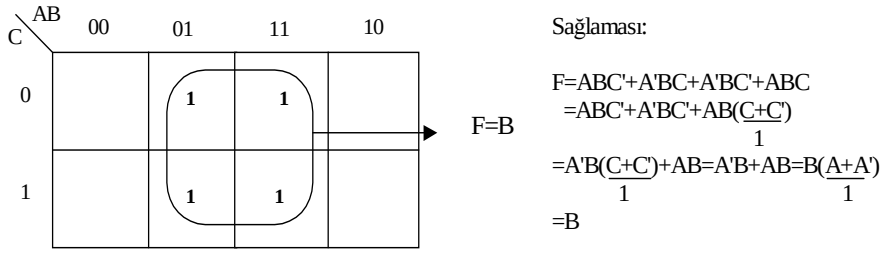
Şekil 6.11. Lojik eşitliklerin Karnaugh haritası yardımıyla sadeleştirilmesi.

Grupların ifade ettikleri kombinasyonlar yazılırsa; F_1=A'C ve F_2=BC' eşitlikleri bulunur ve sadeleştirilmiş ifade F=A'C+BC' şeklinde elde edilir. Bu eşitliğin en sadeleştirilmiş halini ifade eder.

Örnek 5: F=ABC'+A'BC+A'BC'+ABC eşitliğini Karnaugh haritası yöntemiyle sadeleştirelim. Sonucun doğru olduğunu Boolean aritmetiği kurallarıyla kontrol edelim.

Eşitlikteki mintermlerin Karnaugh haritasına taşınması ve oluşan 1'lerin gruplandırılması ile Şekil 6.12'deki Karnaugh elde edilir. Gruplandırma sonucunda F=B eşitliği elde edilir. Diğer taraftan verilen eşitliğin boolean cebiri kuralları ile sadeleştirilmesi sonucunda F=B elde edilir. Bu durum

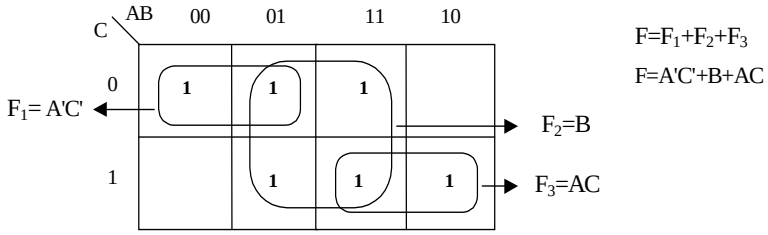
Karnaugh haritası ve Boolean eşitlikleri ile aynı sonuca ulaşılabileceğinin göstergesidir. Bu sonuç, işlemlerin birbirinin sağlamasını yapmak amacıyla kullanılabileceğini gösterir.



Şekil 6.12. Boolean eşitliklerinin Karnaugh haritası yardımıyla sadeleştirilmesi.

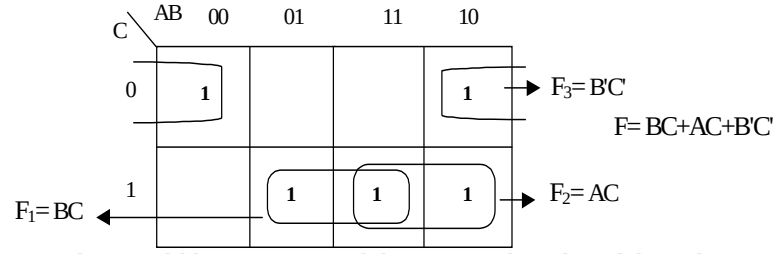
Örnek 6: F=A'B'C'+A'BC'+ABC'+A'BC+ABC+AB'C lojik eşitliğini Karnaugh haritası yardımıyla sadeleştirelim.

Fonksiyonda bulunan mintermlerin temsil edildiği hücrelere '1' değerleri yazılır ve oluşan sayılar gruplandırılır. Grupların temsil ettiği yeni minterm değerlerinin yazılması ile sadeleştirilmiş fonksiyon elde edilir.



Şekil 6.13. Boolean eşitliklerinin Karnaugh haritası yardımıyla sadeleştirilmesi.

Örnek 7: F = A'B'C'+AB'C'+A'BC+AB'C+ABC eşitliğini Karnaugh haritası yardımıyla basitleştirelim.

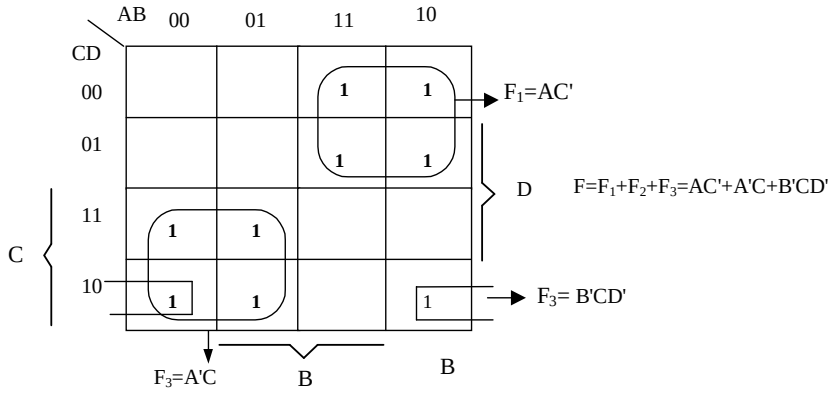


Şekil 6.14. Boolean eşitliklerinin Karnaugh haritası yardımıyla sadeleştirilmesi.

Örnek 8: Aşağıda verilen Boolean eşitliğini Karnaugh haritası kullanarak sadeleştirelim.

A

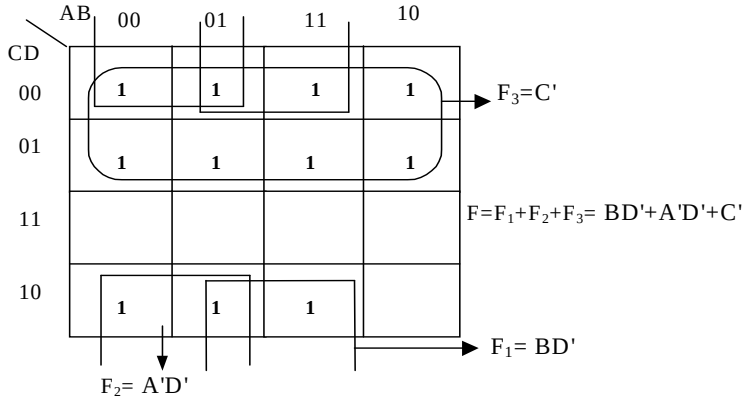
F=ABC'D'+AB'C'D'+ABC'D+AB'C'D+A'B'CD+A'BCD+A'B'CD'+A'BCD'+AB'CD'



Şekil 6.15. Dört değişkenli bir eşitlikteki mintermlerin Karnaugh haritasına taşınması ve fonksiyonun sadeleştirilmesi.

Örnek 9: F=∑(0,1,2,4,5,6,8,9,12,13,14) şeklinde verilen bir minterm ifadesini sadeleştirerek, lojik eşitlik halinde yazalım.

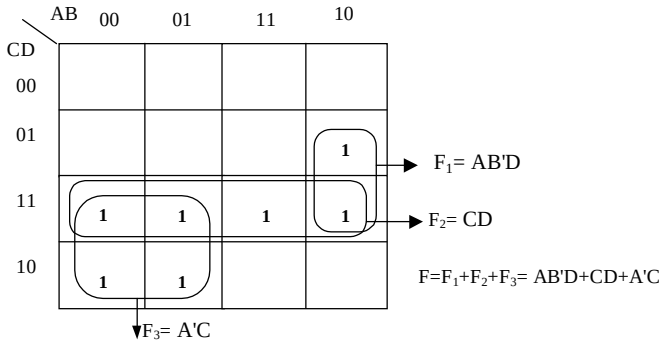
İfadedeki rakamları temsil eden hücrelere ‘1’ yazılması ile eşitlik Karnaugh haritasına taşınır. Daha sonra gruplandırma yapılması ve grupların temsil ettiği eşitliklerin yazılması ise sadeleştirme işlemi bitirilir.



Şekil 6.16. Minterm ifadelerin Karnaugh yardımıyla sadeleştirilmesi.

Örnek 10: F = A'B'CD+ABCD+A'CD'+A'CD+AB'D lojik eşitliğini Karnaugh haritası yardımı ile sadeleştirelim.

Verilen lojik eşitlikte 4 değişken bulunması nedeni ile kullanılacak Karnaugh haritasının dört değişkenli olması gerekir. Eşitlikteki mintermler, temsil edildikleri hücrelere '1' yazılması sureti ile Karnaugh haritasına taşınır. Karnaugh haritasındaki 1'ler gruplandırılıp, grupların karşılıkları olan eşitliklerin yazılması ile sadeleştirme işlemi bitirilir.



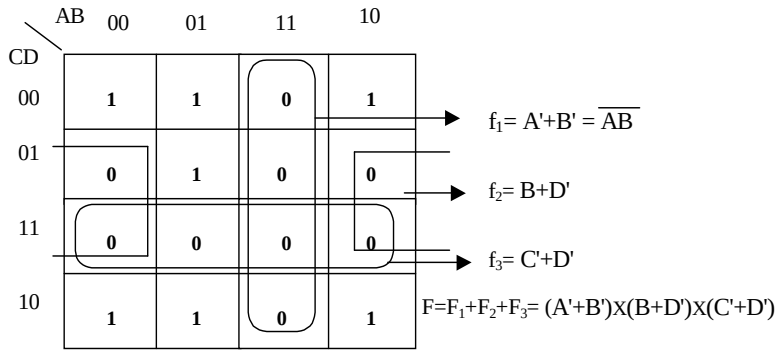
Şekil 6.17. Dört değişkenli eşitliklerin Karnaugh yardımıyla sadeleştirilmesi.

Buraya kadar olan örnekler hep mintermlerin toplamı şeklinde eşitliklerin sadeleştirilmesi idi. Makstermlerin çarpımı şeklinde olan eşitlikleri sadeleştirmede, doğruluk tablosu çıkış sütunundaki '0' olan değerler Karnaugh haritasına taşınır. Eğer eşitlik mimterm olarak verilmişse, mintermleri temsil eden hücrelere 1 yazıldıktan sonra kalan hücreler 0'larla doldurulur. Hücrelerdeki 0'lar gruplanarak makstermlerin çarpımı yazılır.

Örnek 11: $F_{(A,B,C,D)} = \Sigma(0, 2, 4, 5, 6, 8, 10)$ minterm ifadesini Karnaugh haritasındaki 0'ları kullanarak sadeleştirilmiş eşitlik olarak yazalım.

Verilen eşitlikteki rakamlar, temsil edildikleri hücrelere '1' yazılması suretiyle Karnaugh'ya taşınır. '1' yazılmayan tüm hücrelere '0' yazılarak, '0'lar gruplandırılır. Gruplardan maxterm'ler yazılır.

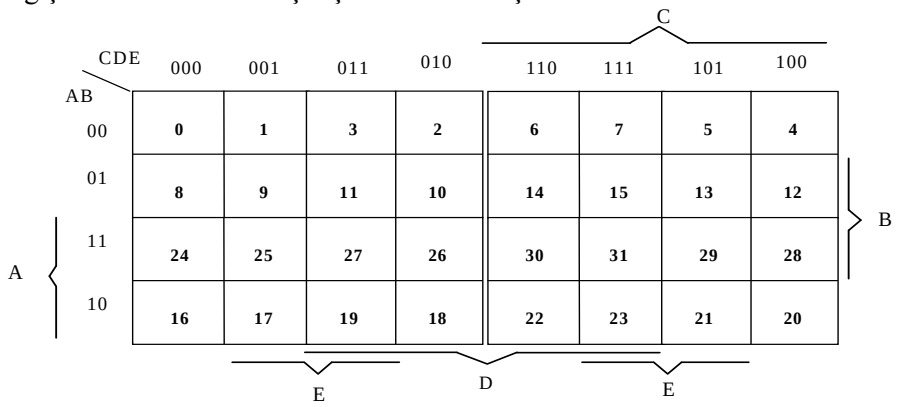
Maksterm'lerin yazılması işleminde değeri değişmeyen değişken '1' ise değişkenin değili, değişken değeri '0' ise değişkenin kendisi yazılır.



Şekil 6.18. Minterm ifadelerin 0'lar kullanılarak sadeleştirilmesi.

6.4. Beş ve Altı Değişkenli Karnaugh Haritaları

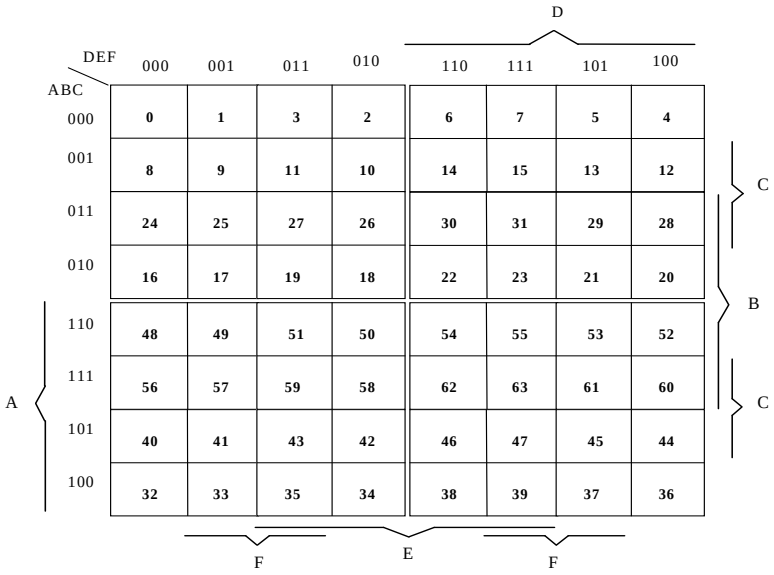
Dörtten fazla sayıdaki değişkenli Karnaugh haritaları ile çalışmak çok kolay değildir. Çok sayıdaki değişkenle işlem yaparken, hücrelerin sayısı oluşabilecek mintermlerin sayısına eşit olacağından beş değişkenli sistemde hücre sayısı 2^5=32, 6 değişkenli bir sistemde hücre sayısı 2^6=64 olur. Beş değişkenli Karnaugh Şekil 8.19'da ve 6 değişkenli Karnaugh haritası Şekil 6.20'de görülmektedir. Beş değişkenli Karnaugh haritası iki adet dört değişkenli Karnaugh haritasının bileşimi, 6 değişkenli Karnaugh haritası iki adet beş değişkenli haritanın birleşmiş hali olarak düşünülebilir.



Şekil 6.19. Beş değişkenli Karnaugh haritasında hücrelerin oluşturulması.

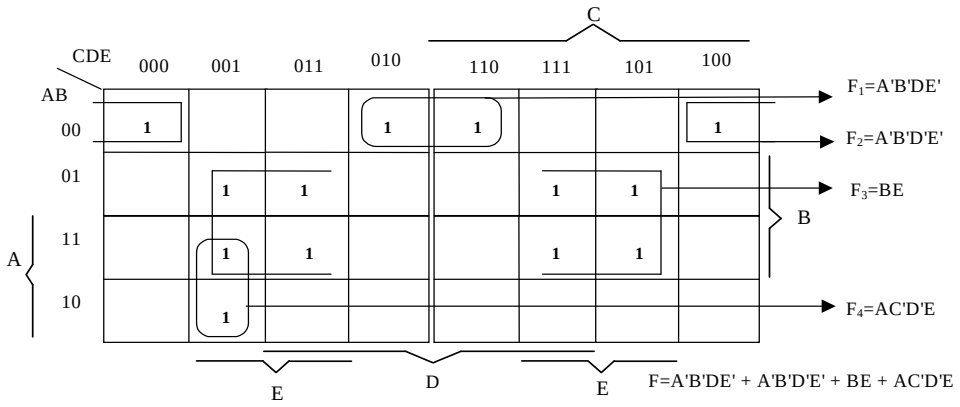
Beş ve altı değişkenli Karnaugh haritalarında oluşturulan grupları temsil eden eşitlikler yazılırken, değerleri değişim göstermeyen değişkenler alınır.

Gruplandırmalar oluşturulurken, beş değişkenli Karnaugh haritasının iki tane dört değişkenli Karnaugh haritasının bileşimi olduğu hatırlanarak, iki parça üst üste düşünülüp gruplandırma yapılabilir (Şekil 6.21).



Şekil 6.20. Altı değişkenli Karnaugh haritasında hücrelerin oluşturulması.

Örnek 12: $F_{(A,B,C,D,E)} = \Sigma(0,2,4,6,9,11,13,15,17,25,27,29,31)$ minterm ifadesini sadeleştirilmiş eşitlik olarak yazalım.

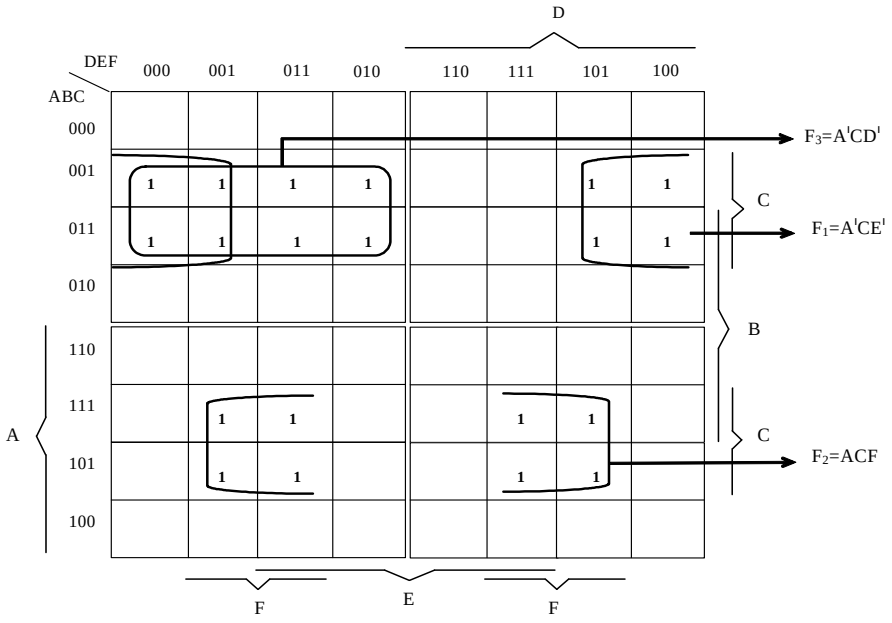


Şekil 6.21. Beş değişkenli Karnaugh haritasında eşitliklerin yazılması.

Altı değişkenli Karnaugh haritasında gruplandırmalarda, haritanın iki adet beş değişkenli Karnaugh haritasından oluşturulduğu hatırlanarak, yatay ve dikey katlama işlemi yapılabilir. Bu durumda bir araya gelen hücreler grup oluşturabilir (Şekil 6.22).

Örnek 13 : Şekil 6.22'de verilen Karnaugh haritasında bulunan 1'leri gruplayarak ilgili eşitlikleri yazalım.

Eşitlikleri yazmada, grupların bulunduğu sütun ve satırlarda değişim göstermeyen değişkenler minterm şeklinde yazılır. Ek olarak düşünülebilecek tek nokta, Karnaugh haritasının katlanması durumunda üst-üste gelecek hücrelerin grup oluşturabileceğidir.



Şekil 6.22. Altı değişkenli Karnaugh haritasında eşitliklerin yazılması.

6.5. Farketmeyen Durumlu (Don't Care'li) Lojik Eşitlikler

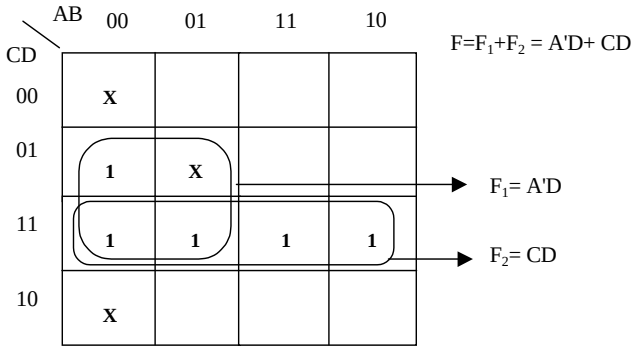
Karnaugh haritasında bulunan 1 ve 0'lar lojik fonksiyonun oluşmasında bir anlam ifade eder. Bununla beraber, giriş değişkenlerinin kesin değerler olmadığı durumlar da bulunabilir.

Örneğin; dört bitle ifade edilen onluk sistemde, 9'dan sonraki altı kombinasyon hiçbir zaman oluşmaz. Bu durumda, oluşmayan (kullanılmayan) 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) konur. Karnaugh haritasında gruplandırma yapılırken, fark etmeyenli hücreler 0 veya 1 olarak kabul edilebilir. Karar, hangi kabulün fonksiyonu daha basit hale getireceğine göre verilir.

Örnek 14: F = Σ(1,3,7,11,15) ve farketmez durumları d = Σ(0,2,5) olan bir fonksiyonu sadeleştirelim.

Sadeleştirme işlemi için, eşitlikte bulunan sayıların temsil ettiği hücrelere '1', farketmez durumlarını temsil eden hücrelere 'x' işareti konur. Oluşan '1'ler gruplandırma işlemine tabi tutulur. 'x' işaretli hücreler, 1 veya 0 olarak düşünülebilir. Örnekte fark etmeyen durumlardan birisi '1', diğeri '0' olarak kabul edilmiştir.



Şekil 6.23. Fark etmeyen durumları kullanarak boolean eşitlerin elde edilmesi.

Buraya kadar Karnaugh haritalarının sadeleştirmede kullanılmasını inceledik. Şimdi Karnaugh haritalarının lojik devrelerin tasarımında kullanılmasını inceleyelim.

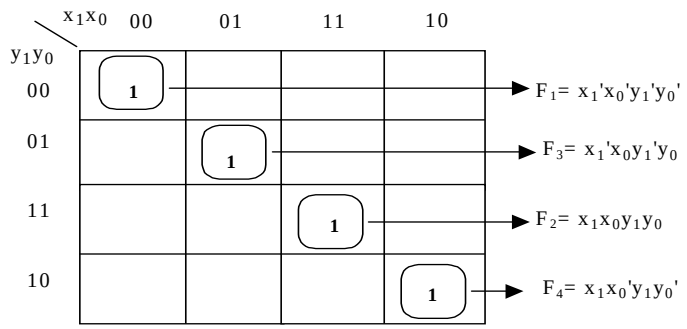
6.6. Karnaugh Haritası Yardımı ile Lojik Devrelerin Tasarımı

Karnaugh haritası kullanılarak yapılacak lojik devre tasarımında, ilk işlem olarak lojik devre tasarımındaki işlem basamakları uygulanıp doğruluk tablosu oluşturulur. Doğruluk tablosundan elde edilen değerler Karnaugh haritasına taşınarak gruplandırmalar yapılır. Grupların temsil ettiği eşitlikler yazılarak lojik fonksiyon elde edilir. Son aşamada ise eşitliği temsil eden devre çizilir.

Örnek 15: x_1x_0 iki bitlik bir sayıyı temsil etmektedir. y_1y_0 değişkenleri ise diğer iki bitlik bir sayıyı göstermektedir. x_1, x_0, y_1, y_0 değerlerini giriş olarak kullanan ve x_1x_0 ile y_1y_0 sayılarının eşit olduğu durumlarda çıkışın 1 olmasını sağlayan lojik devreyi Karnaugh haritası kullanarak tasarlayalım.

x_1	x_0	y_1	y_0	Q
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(a)

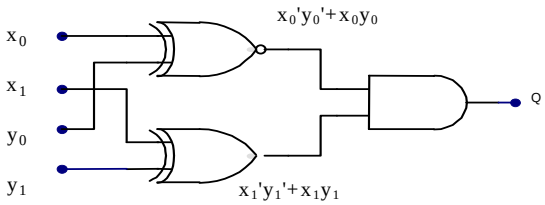


$F=F_1+F_2+F_3+F_4 = x_1'x_0'y_1'y_0' + x_1x_0y_1y_0 + x_1'x_0y_1'y_0 + x_1x_0'y_1y_0'$

$F=\sum(0,5,10,15)$

EXNOR EXOR

(b)



(c)

$Q = (x_0'y_0'+x_0y_0).(x_1'y_1'+x_1y_1) = (x_0'x_1'y_0'y_1'+x_0'x_1y_0'y_1)+(x_0x_1'y_0y_1'+x_0x_1y_0y_1)$

Şekil 6.24. Karnaugh haritası kullanarak lojik devrelerin tasarlanması.

i- İlk aşamada, tüm değişkenlerin alabileceği değerleri toplu olarak gösteren doğruluk tablosu (4 değişkenli) oluşturulur.

ii- Doğruluk tablosunda x_1,x_0 ve y_1,y_0 çiftlerinin eşit olduğu durumlar tespit edilir ve bu durumlar çıkış sütununda ilgili değerlerin karşısında'1' konularak belirtilir (Şekil 6.24.a).

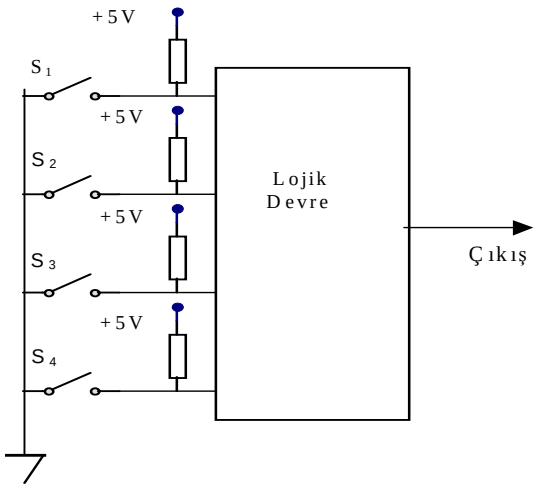
iii- Doğruluk tablosu çıkış sütunundaki '1' değerleri Karnaugh haritasındaki ilgili hücrelere taşınır (Şekil 6.24.b).

iv- Karnaugh haritasındaki '1' değerleri gruplandırılarak, gruplardan eşitlikler yazılır.

v- Eşitlikler uygun kapı devreleri ile lojik devre olarak çizilir (Şekil 6.24.c).

Örnek 16: Şekil 6.23.a'da görülen devre bir fotokopi makinesinin kontrol devresidir. Makinenin içerisinde kağıdın yolu üzerinde 4 tane anahtar bulunmaktadır. Anahtarlar normalde açıktır ve kağıt geçerken anahtarları kapatmaktadır. Herhangi iki anahtarın kapatılması durumunda çıkışta '1' oluşmasını sağlayan lojik devreyi tasarlayalım.

Yalnızca S_1 ve S_4 anahtarları aynı anda kapatılamaz.



S_1	S_2	S_3	S_4	**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	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
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

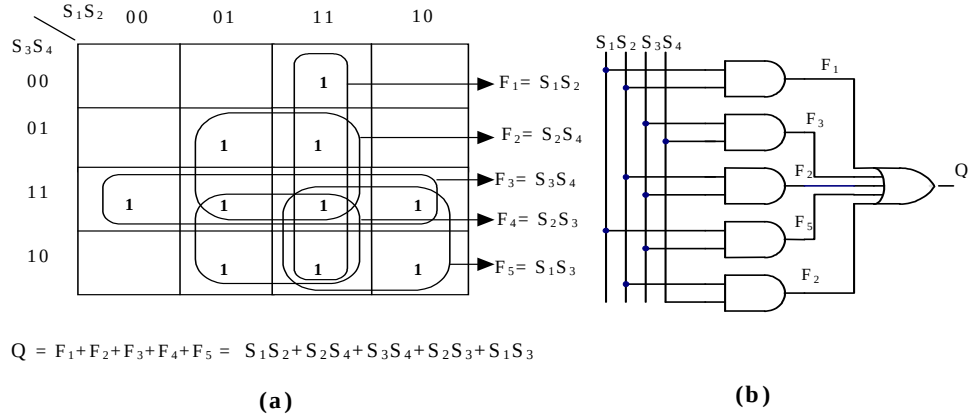
(a) **(b)**

Şekil 6.25. Lojik tasarımda doğruluk tablosunun oluşturulması.

Devre tasarımında, istenilen durumlar doğruluk tablosunun çıkış sütununda belirtilir (Şekil 6.25.b).

Doğruluk tablosundaki ‘1’ değerleri Karnaugh haritasına taşınarak, gruplandırmalar yapılır ve grupların karşılığı olan eşitlikler yazılır (Şekil 6.26.a).

Yazılan eşitlikleri gerçekleştirecek lojik devre, uygun kapılar ile oluşturulur (Şekil 6.26.b).



Şekil 6.26. Doğruluk tablosunun Karnaugh haritasına taşınması ve lojik devrenin oluşturulması.

Örnek 17: Bir sendikanın yönetim kurulunda dört temsilci bulunmakta ve her temsilci sendikaya kayıtlı 560 üyeden belirli sayıdaki üyeyi temsil etmektedir. Kararların alınması için salt çoğunluğun gerekli olduğu yönetim kurulunda, temsilcilerin birer düğmeye basması ile sonucun geçerli olup olmadığını hesaplayarak, geçerli ise bir lambanın yanmasını sağlayacak devreyi Karnaugh haritası kullanarak NAND kapılarıyla tasarlayalım.

Temsilciler: A temsilcisi: 105,
B temsilcisi: 125,
C temsilcisi: 202,
D temsilcisi: 128 üyeyi temsil etmektedir.

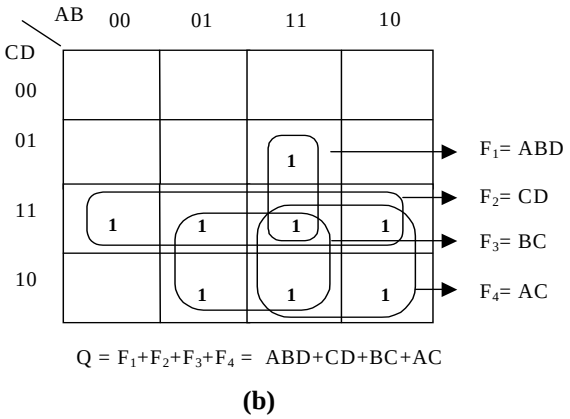
İlk işlem olarak, doğruluk tablosu yardımı ile sonucun ‘1’ olacağı kombinasyonları bulmamız gerekir. Sonucun ‘1’ olduğu kombinasyon, evetlerin salt çoğunluğun sağladığı durumlardır (Şekil 6.27.a).

Sonucun 1 olduğu kombinasyonlar Karnaugh haritasına taşınır ve taşınan değerler gruplandırılır. Gruplandırmadan elde edilen mintermler birleştirilirse, istenilen işlemi gerçekleştirilecek devrenin lojik eşitliği elde edilir (Şekil 6.27.b).

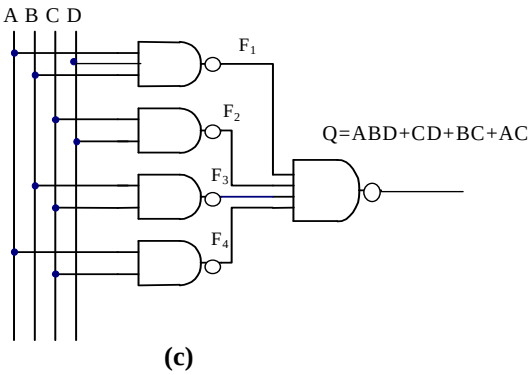
Lojik eşitliği gerçekleştirecek devre kapılar ile oluşturularak işlem bitirilir (Şekil 6.27.c).

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

(a)



(b)

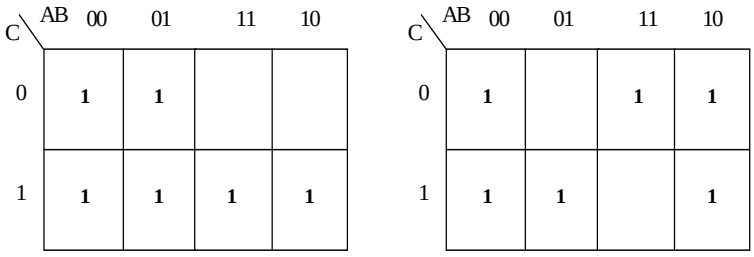


(c)

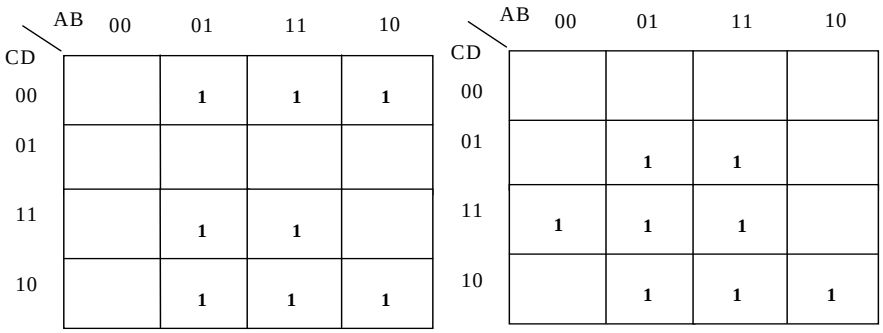
Şekil 6.27. Karnaugh haritası kullanarak lojik devrelerin tasarlanması.

Tekrarlama ve Çalışma Soruları

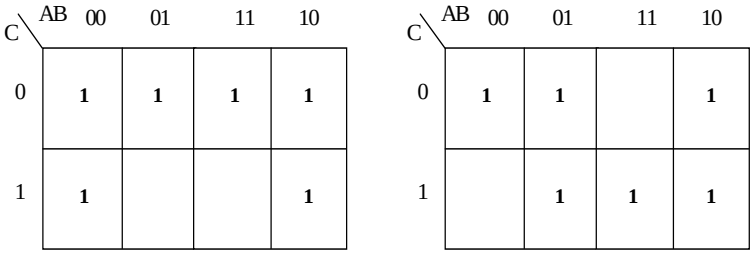
1. Boolean eşitlikleri sadeleştirmek için kullanılan yöntemler nelerdir ?
2. Üç değişkenli Karnaugh haritasını iki farklı şekilde oluşturunuz.
3. İki, üç ve dört değişkenli Karnaugh haritalarında, her bir hücrelerin temsil ettiği kombinasyonu hücrelerin içerisine yazınız.
4. Çıkış sütununda, 0,2,4,5 değerlerinde "1" değerlerine sahip, üç değişkenli doğruluk tablosundaki değerleri Karnaugh haritasına taşıyınız.
5. Dört değişkenli bir doğruluk tablosunda 1,6,7,9,14,15 değerlerinde sonuç '1'dir. Bu değerleri Karnaugh haritasına taşıyınız.
6. Karnaugh haritalarında gruplandırmayı tarif ediniz.
7. Gruplandırmada uyulacak kuralları sıralayınız.
8. Aşağıda verilen 3 değişkenli karnaugh'larda uygun gruplandırmaları yapınız.



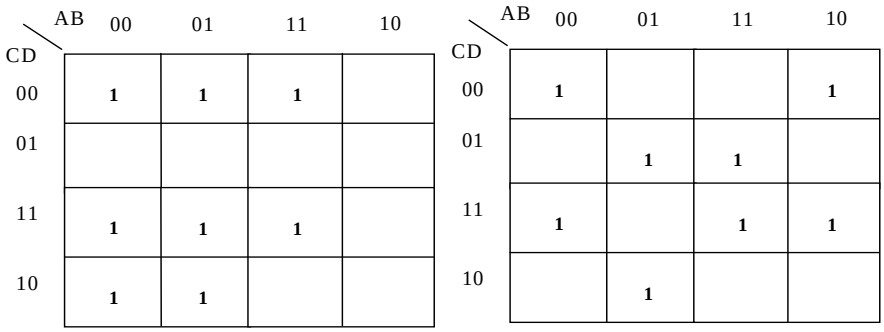
9. Verilen 4 değişkenli Karnaugh'larda uygun gruplandırmaları yapınız.



10. Verilen Karnaugh haritalarında gruplandırma yaparak, grupların temsil ettikleri eşitlikleri yazınız**.**



11. Verilen Karnaugh haritalarında gruplandırma yaparak, grupların temsil ettikleri eşitlikleri yazınız**.**



12. Lojik ifadelerin Karnaugh haritası yöntemi ile sadeleştirmesinde takip edilecek işlem sırasını özetleyiniz.

13. F = AB'C+ABC'+A'BC'+A'B'C' Lojik ifadesini Karnaugh haritası kullanarak sadeleştiriniz.

14. F = AB'C'D+AB'C'D'+A'B'CD+ABCD Lojik ifadesini Karnaugh haritası kullanılarak sadeleştiriniz.

15. Karnaugh haritalarının lojik tasarımda kullanımına kendiniz bir örnek vererek lojik devreyi çiziniz.

16. F = AB'C'D + AB'C'D + A'B'CD + A'B'CD' + A'BCD' + ABC'D' + ABCD + AB'C'D + AB'CD lojik ifadesini Karnaugh haritası kullanılarak sadeleştiriniz.

17. F = A'B'C'D' + A'B'CD + A'B'CD' + A'BC'D' + A'BCD + A'BCD' + ABC'D' + ABCD' + AB'C'D' + AB'CD Lojik ifadesini sadeleştiriniz.
18. F = A'B'C'D' + A'B'C'D + A'B'CD + A'B'CD' + A'BC'D + A'BCD + ABC'D + ABCD + AB'C'D' + AB'C'D + AB'CD + AB'CD' Lojik ifadesini Karnaugh haritası kullanılarak sadeleştiriniz.
19. F = A'B'C'D + A'BC'D' + A'BC'D + A'BCD' + ABC'D' + ABC'D + ABCD' + AB'C'D' + AB'C'D + AB'CD + AB'CD' Lojik ifadesini sadeleştiriniz.
20. F = A'B'C'D + A'B'CD + A'BC'D + A'BCD + AB'CD + ABCD Lojik ifadesini Karnaugh haritası kullanılarak sadeleştiriniz.

BÖLÜM 7

SAYISAL ENTEGRELER

(DIGITAL IC's)

Giriş

Lojik kapılar başlığı altında açıkladığımız temel lojik elemanlar; direnç, diyot, transistör, küçük değerlikli kondansatör, FET, MOSFET, vb. elektronik devre elemanlarından oluşur. Aynı çeşit veya farklı elektronik elemanların bir paket haline getirilmesi ile oluşan yeni eleman, **'entegre'** (Integrated Circuit -IC) olarak isimlendirilir. Entegreler ;

i- Yapılarında kullanılan elemanların çeşidine,

ii- Entegre içerisinde bulunan lojik kapı ve transistör sayısına,

iii- Kullanılan entegre teknolojisine,

referans alınarak sınıflandırılabilirler.

Sayısal entegreler, yapılarında kullanılan devre elemanlarına göre :

i- Bipolar entegre devreler; DTL, TTL, HTL, ECL, vb.

ii- MOS-lojik entegre devreler; NMOS, PMOS, CMOS,

olarak iki grupta toplanabilirler.

Her iki grupta bulunan elemanlar, içerdikleri lojik kapı ve transistör sayısına göre;

i- Küçük ölçekli IC'ler,
ii- Orta ölçekli IC'ler,
iii- Geniş ölçekli IC'ler,
iv- Çok geniş ölçekli IC'ler,
v-Çok-çok geniş ölçekli IC'ler,
vi-Giga ölçekli IC'ler,

olarak sınıflandırılırlar.

Bu genel sınıflandırmalarda ortaya çıkan grupların detaylarını incelemeden önce, her bir grubun sahip olduğu özellikleri açıklamamıza yardımcı olacak sayısal entegre parametrelerini (özelliklerini) ve sayısal entegre terminolojisini inceleyelim.

7.1. Sayısal Entegre Parametreleri ve Sayısal Entegre Terminolojisi

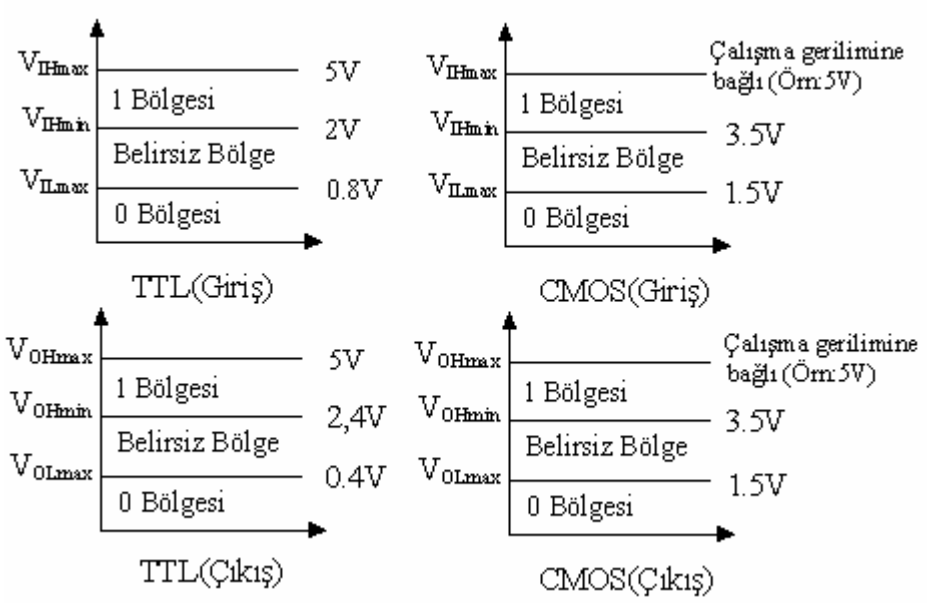
Entegre gruplarını birbirleri ile kıyaslamak için entegrelerin sahip oldukları özelliklerden faydalanılır. Entegrelerin sahip oldukları özellikler, **'entegre parametreleri'** veya **' entegre teknik özellikleri'** olarak adlandırılır. Sayısal entegre parametreleri ve entegrelerde kullanılan terimler aşağıdaki şekilde sıralanabilir:

i. Lojik gerilim seviyelerinin tanımlanması,
ii. Yükleme (çıkış) kapasitesi (Fan Out),
iii. Gürültü bağışıklığı (Noise Immunity –Noise Margin),
iv. Yayılım Gecikmesi ve Yayılım Hızı (Propagation Delay ve Propagation Speed),
v. Güç Harcaması (Power Dissipation),
vi. Hız-Güç Üretimi (Speed-Power Product),
vii. Akım kaynağı (Current Sourcing) ve Akım yutumu (Current Sinking).

Entegre parametrelerinin hangi koşullar altında geçerli olduğu 'veri kitaplarında' verilmektedir. Örneğin entegrelerin sağlıklı çalışabilmesi için garanti edilen sıcaklık endüstri tipi elemanlarda 0 ile $+70^{o}$ C ve askeri tip elemanlarda -55^{o} C ile $+125^{o}$ C arasındadır

7.1.1. Lojik Gerilim Seviyeleri

Lojik gerilim seviyeleri, lojik bir sinyalde ‘0’ ve ‘1’ seviyelerini temsil eden gerilim değerleridir. ‘1’ seviyesinin ‘H-High’ ve ‘0’ seviyesinin ‘L-Low’ ile gösterildiğini kabul edersek, gerilim seviyeleri Şekil 7.1’deki değerlere sahip olur.



Şekil 7.1. TTL ve CMOS entegrelerinin giriş ve çıkış gerilim seviyeleri.,

Şekil 7.1’de kullanılan kısaltmaların anlamlarını özetleyelim:

$V_{IL(max)}$ = V **i**nput **l**ow **max**imum - Girişin ‘0’ olması için olabilecek en yüksek gerilim seviyesi.

$V_{IH(min)}$ = **V** **i**nput **h**igh **min**imum - Girişin ‘1’ olması için gereken en düşük gerilim seviyesi.

$V_{OL(max)}$ = **V** **o**utput **l**ow **max**imum - Çıkışın lojik ‘0’ olması için olabilecek en yüksek gerilim seviyesi.

$V_{OH(min)}$ = **V** **o**utput **h**igh **min**imum - Çıkışın ‘1’ olması için gerekli en düşük gerilim seviyesi.

PARAMETRE	TANIM	7400	74L00	74LS00
$V_{OH\ min}$	'1' durumunda minimum çıkış gerilimi	2.4 V	2.4 V	2.7 V
$V_{OL\ max}$	'0' durumunda maksimum çıkış gerilimi	0.4 V	0.3 V	0.5 V
$V_{IH\ min}$	Girişin '1' olmasını garanti edecek minimum giriş seviyesi	2.0 V	2.0 V	2.0 V
$V_{IL\ max}$	Girişin '0' olmasını garanti edecek maksimum giriş seviyesi	0.8 V	0.7 V	0.8 V

Tablo 7.1. TTL 7400 entegresinin farklı serilerde sahip oldukları gerilim seviyeleri.

Giriş ve çıkış gerilim seviyelerinde bulunan sınır değerleri, akım değerleri için de geçerlidir. Ancak, akım değerinden bahsedilirken genelde bir lojik elemanın çekeceği ortalama akımdan bahsedilir.

7.1.2. Çıkış Kapasitesi (Fan Out)

Bir lojik kapının çıkışına bağlanabilecek maksimum yük sayısına **'Çıkış kapasitesi - Fan Out'** denir. 'Fan Out' değeri entegre çeşidine göre farklılık gösterir. Karmaşık devrelerde, kullanılan entegrenin çıkışına kaç tane kapı devresinin bağlanabileceğinin bilinmesi gerekir. Çıkış ucuna fazla sayıda yük bağlanırsa entegre bozulur.

Ardarda bağlanacak entegrelerin aynı cins olması uygulamada kolaylık sağlar. **'**Fan Out' değeri, kullanılan entegrelerin giriş ve çıkış akımlarına göre hesaplanır (Şekil 7.2).

Örneğin TTL entegrelerin LS tipinde bu değerler;

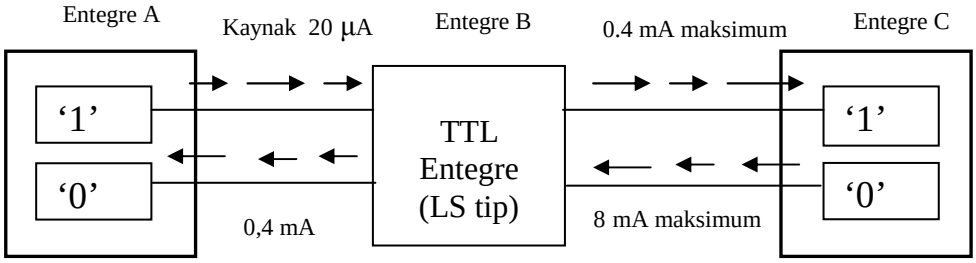
Çıkış akımı; 1 durumunda 0.4 mA (yüke doğru)

0 durumunda 8 mA (entegreye doğru)

Giriş akımı ; 1 durumunda 20 μA (entegreye doğru)

0 durumunda 0.4 mA (kaynağa doğru)

şeklindedir.



Şekil 7.2. Entegrelerin giriş ve çıkış akım değerlerine göre Fan Out’un hesaplanması.

TTL entegrelerin LS tipinde; bir entegrenin (B) entegresi giriş ucu, başka bir (A) entegresi ‘1’ seviyesindeki çıkış ucuna bağlanırsa; giriş ucundan B entegresine doğru 20 μA akım akar (Şekil 7.2). Bu anda A entegresinin çıkış akımı maksimum 0.4 mA olduğundan;

Fan Out = 0.4mA / 20 μA = 20 tane

olarak bulunur.

‘B’ Entegresinin giriş ucu, ‘A’ entegrenin ‘0’ durumundaki çıkış ucuna bağlanırsa; ‘B’ entegresinin giriş ucundan ‘A’ entegresine doğru 0.4 mA’lik akım akar. ‘A’ entegresinin üzerinden akacak akım maksimum 8 mA olduğundan, aynı anda bağlanabilecek eleman sayısı;

Fan Out = 8 mA / 0.4 mA = 20 tane

olarak bulunur.

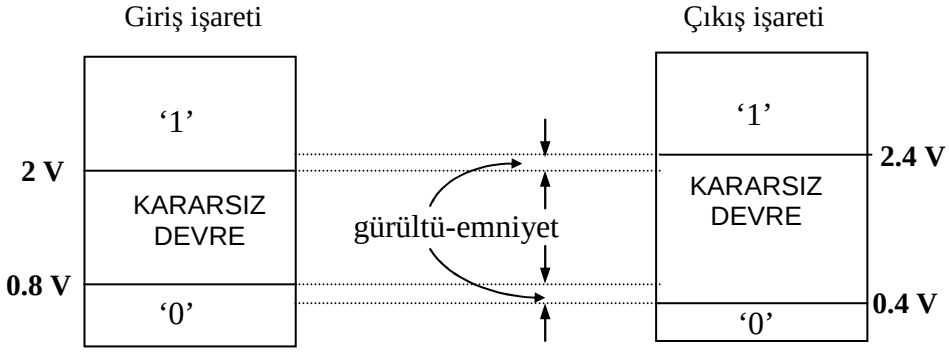
Bulunan iki değerden küçük olan alınır. Ancak, örnekte bulunan iki değer aynı olduğundan herhangi birisi alınabilir ve entegre çıkışına maksimum 20 giriş ucu bağlanabileceği söylenir.

Girişe bağlanacak kapılarında belirli sayıda olması gerekir. Girişe bağlanacak maksimum eleman sayısı kataloglarda **‘fan in’** olarak belirtilir.

7.1.3. Gürültü Bağışıklığı (Noise Immunity - Noise Margin)

Gürültü bağışıklığı, devrenin çalışmasına etki etmeyecek şekilde izin verilebilen gürültü sınırıdır ve ‘mV’ cinsinden ifade edilir. TTL entegrelerin ‘1’ konumuna gelmesi için 2.4V gereklidir derken, bunun 0.4 V’u gürültü payı olarak konmuştur. ‘0’ seviye için de durum aynıdır.

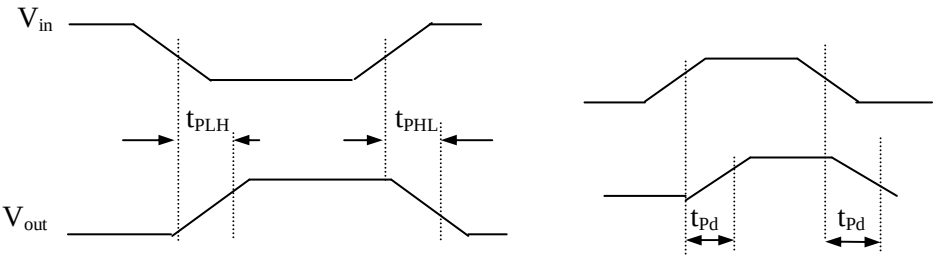
Emniyet Payı: Bir sayısal devrenin girişine gelen gürültü işaretlerini, gerçek işaretten ayırabilmesi için konulan gerilim farkına sayısal devrenin **'gürültü-emniyet payı'** denir (Şekil 7.3). Diğer bir deyişle, lojik devrelerde giriş ve çıkış işaretlerinde '1' ve '0' seviyeleri arasındaki fark, '**gürültü-emniyet payı'** olarak isimlendirilir.



Şekil 7.3. Dijital entegrelerde gürültü-emniyet payının tespiti.

7.1.4. Yayılım Gecikmesi (Propagation Delay-t_{pd}) ve Yayılım Hızı (Propagation Speed)

Dijital devrenin veya lojik kapının girişindeki değişme ile buna bağlı olarak çıkışta meydana gelecek değişme arasındaki zaman farkına **'yayılım gecikmesi'** adı verilir ve nanosaniye (nsn) cinsinden ifade edilir. Yayılım hızı ise yapılan işlemin hızını belirtir ve MHz cinsinden ifade edilir.



Şekil 7.4. Lojik entegrelerde meydana gelen yayılım gecikmeleri.

tp_{HL} ⟶ 1'den 0'a geçerken harcanan zaman,

tp_{LH} ⟶ 0'dan 1'e geçerken harcanan zaman,

tpd ⟶ iki değerin (tp_{HL} ve tp_{LH}) ortalamasıdır.

Genelde, tp_{HL} ve tp_{LH} değerleri yükleme durumlarına bağlı olarak değişebilir ve aynı değerde olmayabilir. Aynı değerde olmaması durumunda, iki değerin aritmetik ortalaması alınır.

7.1.5. Güç Harcaması (Power Dissipation)

'Güç Harcaması'; bir lojik kapının harcadığı güç miktarıdır ve 'mW' cinsiden ifade edilir. Entegrede harcanan güç genelde tek bir kaynaktan elde edilir. Entegrenin güç kaynağına bağlanacak ucu bipolar entegrelerde 'V_{CC}', CMOS entegrelerde 'V_{DD}' olarak isimlendirilir.

Bir lojik kapının bağlı olduğu güç kaynağından çekeceği akım, kapının çıkışının aldığı değere (0 veya 1) göre değişir. Bu değişiklik, entegre devrenin harcadığı gücün değişmesi sonucunu doğurur. Entegre devrenin harcadığı gücü bulmak için, çıkışın '0' ve '1' olduğu anlarda entegrede harcanan gücün ortalaması alınır.

7.1.6. Hız-Güç Üretimi (Speed - Power Product)

Sayısal entegreler başlangıçtan beri düşük yayılım gecikmesi ve küçük güç harcaması özellikleriyle tanımlanırlar. Lojik entegre aileleri çok değişik hız-güç harcaması oranına sahiptirler. Bu nedenle sayısal entegreleri kıyaslamada kullanılacak özelliklerden birisi, **'hız-güç üretimi'** olarak isimlendirilen ve kapı yayılım gecikmesi ile kapı güç harcamasının çarpımından elde edilen değerdir.

Örneğin; bir entegrede yayılım gecikmesi 10 nsn ve ortalama güç harcaması 50 mW ise;

Hız-güç üretimi = 10 nsn*50 mW = 50 piko watt-sn (50 piko joule) olarak bulunur.

Sayısal entegrelerin seçiminde düşük hız-güç üretimi tercih edilir. Bu nedenle, tasarımcılar entegrenin hızını artırarak yayılım gecikmesini düşürmeye veya güç harcamasını düşürerek hız-güç üretimini küçültmeye çalışmaktadırlar. Ancak, transistör tetiklemeli devrelerin özelliği nedeniyle her ikisini beraber yapmak zordur.

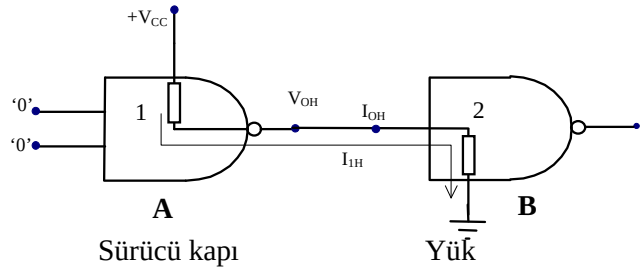
7.1.7. Akım Besleme (Current Sourcing) ve Akım Yutumu (Current Sinking)

Lojik aileleri, akımın bir lojik elemanın çıkışı ile diğer lojik elemanın girişi arasında akış şekline göre tanımlanabilir. **Akım besleme** (current sourcing) işleminde; 'A' kapısının çıkışı '1' olduğu zaman, gerçekte bir direnç gibi

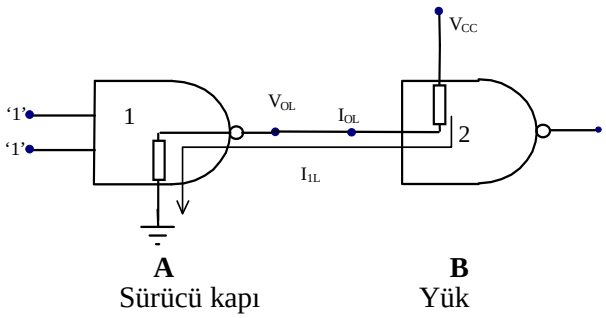
davranan ‘B’ kapısının girişine I_{IH} akımı sağlanır (Şekil 7.5.a). Bu nedenle; ‘A’ kapısının çıkışı ‘B’ kapısı girişi için bir akım kaynağı oluşturur.

‘Akım yutumu’ (Current sinking) işleminde; ‘B’ kapısının giriş devresi V_{cc} kaynağına bir direnç bağlanmış etkisi gösterir (Şekil 7.5.b). ‘A’ kapısının çıkışı ‘0’ durumuna geldiği zaman, akım ‘B’ kapısından ‘A’ kapısının çıkış direncine doğru akar. Diğer bir deyişle, ‘B’ kapısının girişini süren devrenin ‘0’ olması, girişten gelen akımı şase yapacak pozisyonda olur. Akımın şase yapılması nedeniyle bu özellik, **‘akım yutumu’** olarak adlandırılır.

Akım besleme ve akım yutumu özelliği, farklı lojik ailelerini incelerken önemli bir özellik olarak karşımıza çıkar.



a) Akım besleme işlemi.



b) Akım yutumu işlemi.

Şekil 7.5. Akım besleme ve akım yutumu devrelerinin durumu.

7.2. Entegrelerin Yapılarındaki Eleman Sayısına Göre Sınıflandırılması

Sayısal entegre teknolojisi günümüzde hızla gelişmekte ve devrelerin özelliklerine göre farklı sayıda elemanı içeren entegreler imal edilmektedir. Farklı sayıda kapı devresi içeren ve oluşumlarında farklı sayıda eleman kullanılan dijital entegreler, Tablo 7.2'deki gibi sınıflandırılabilir. Tabloda sıfır ölçekli birleştirme ile, elektronik elemanların bağımsız olarak kullanımı kastedilmektedir.

Teknolojilerin yıllara göre gelişimi ve herbir teknoloji ile üretilen entegrelerin içerdikleri eleman sayıları Tablo 7.2.b'de verilmektedir.

7.3. Lojik Entegrelerin Yapılarındaki Elemanlara Göre Sınıflandırılması

Entegre devrelerin oluşumunda, direnç, diyot, kondansatör, transistör, FET, MOSFET vb. elemanlar kullanılır. Lojik entegreler, devrelerin oluşturulmasında kullanılan devre elemanlarına (direnç, diyot, transistör vb.) göre sınıflandırılabilir. Genel olarak bipolar ve MOS lojik aileleri adı altında yapabileceğimiz sınıflandırmada, alt gruplar aşağıdaki şekilde oluşur:

A-Bipolar Lojik Ailesi

i- Direnç Diyot Lojik (Resistor Diode Logic - RDL),

ii- Direnç Transistör Lojik (Resistor Transistor Logic - RTL),

iii- Diyot Transistör Lojik (Diode Transistor Logic - DTL),

iv- Yüksek Eşikli Lojik (High Threshold Logic - HTL),

v- Transistör Transistör Lojik (Transistor Transistor Logic - TTL),

vi- Emiter Kuplajlı Lojik (Emiter Coupled Logic - ECL),

vii- Entegrelendirilmiş Enjeksiyon Lojik (Integrated Injection Logic – IIL - I^2L).

B-MOS Lojik Ailesi

i- Metal Oksit Lojik (Metal Oxide Semiconductor-MOS),

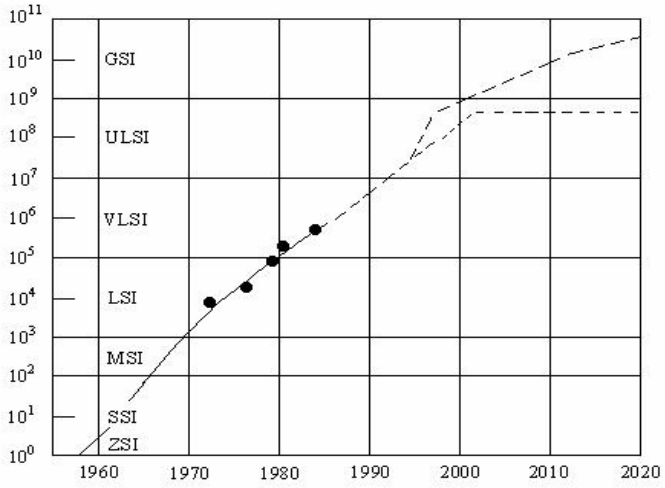
- N - MOS Lojik,
- P - MOS Lojik,

ii- CMOS Lojik (Complementary MOS-CMOS).

Entegre devrelerin yapım şekillerini ve devrelerin çalışma prensiplerini anlamak için, yukarıdaki sınıflandırmada yer alan teknikleri sırası ile inceleyelim.

ENTEGRE YOĞUNLUK DERECESİ	FONKSİYON SAYISI	İÇERDİĞİ ELEMAN SAYISI	KAPI SAYISI
Sıfır Ölçekli Birleştirme (Zero Scale İntegration - ZSI)			
Küçük Ölçekli Birleştirme (Small Scale İntegration - SSI)	2-20	100	1-12
Orta Ölçekli Birleştirme (Medium Scale İntegration-MSI)	20-100	500	13-99
Geniş Ölçekli Birleştirme (Large Scale İntegration - LSI)	100-500	100000	100-9999
Çok Geniş Ölçekli Birleştirme (Very Large Scale İnteg.-VLSI)	500-100000	250000	10000-99999
Çok Çok Geniş Ölçekli Birleş. (Ultra Large Scale İnteg.ULSI)	100000 ≤	$10^7 – 10^9$	$10^5 – 10^8$
Giga Ölçekli Birleştirme (Giga Scale İntegration - GSI)	-	$10^9 \leq$	$10^8 \leq$?

(a)



(b)

Tablo 7.2. Entegrelerin yapılarındaki eleman sayısına göre sınıflandırılması.

7.3.1. Bipolar Transistör Kullanılarak Oluşturulan Lojik Teknolojisi

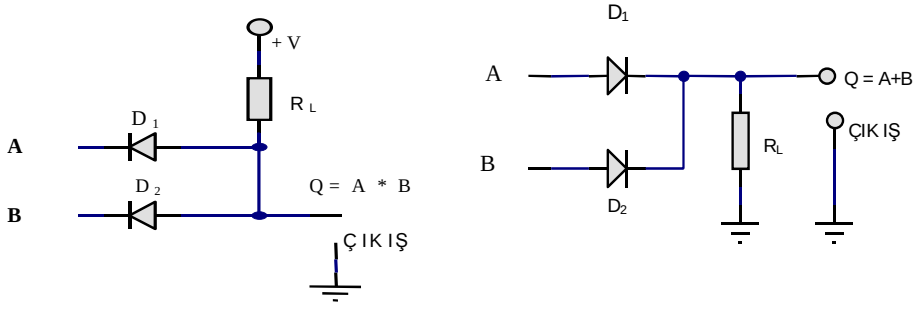
Yarıiletken ve pasif elektronik devre elemanları kullanılarak farklı şekillerde lojik kapı devreleri oluşturulur. Oluşturulan kapı devreleri, kullanılan elemana göre isimlendirilir.

7.3.1.1. Direnç Diyot Lojik (RDL)

Kapı devrelerini oluşturan temel elemanlar olarak direnç ve diyotların kullanıldığı mantık, **'Direnç Diyot Lojigi (RDL)'** olarak isimlendirilir. RDL mantığı sabit bir çıkış gerilimi veremediğinden ve tersleme (değil) işlemi yapamadığından dolayı güncelliğini yitirmiştir.

RDL yöntemi ile yapılan VE kapısı ve VEYA kapısı açık devreleri Şekil 7.6'da görülmektedir. VE kapısında, girişlerin her ikisinin lojik '1' olması durumunda diyotlar yalıtkandır ve çıkış '1' dir. Girişlerden her hangi birinin veya ikisinin '0' olması durumunda (şase yapılarak), girişi '0' olan diyot iletken olarak +V'den girişe doğru bir akım akmasına neden olur. Bu durumda çıkışta, yalnızca diyodun iletken olduğu durumda üzerinde düşen küçük gerilim (0.5V) gözükür. Bu değer, çıkışın lojik '0' olması demektir.

Lojik VEYA kapısında, A veya B girişinin lojik '1' olması durumunda ilgili diyot iletken olarak R_L direnci üzerinden bir akımın akmasına ve bir gerilim düşümüne neden olur. Çıkış uçlarında R_L üzerinde düşen gerilim gözükeceğinden, çıkışta lojik '1' oluşur. Girişlerin her ikisinin '0' olması durumunda diyotlar yalıtkandır ve R_L üzerinde gerilim düşümü olmayacağından çıkış lojik 0'dır.



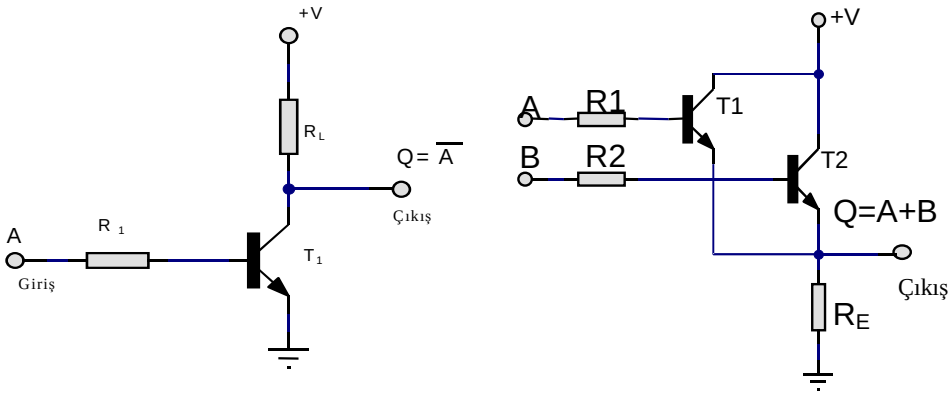
'VE' Kapısı 'VEYA' Kapısı

Şekil 7.6. RDL tekniği ile oluşturulan 'VE' ve 'VEYA' kapı devreleri açık şemaları.

7.3.1.2. Direnç Transistör Lojik (RTL)

İlk imal edilen ticari tip olan ve basit bir yapısı bulunan RTL entegreleri, diğer lojik entegre tekniklerinin geliştirilmesiyle önemini yitirmiştir.

RTL teknolojisiyle yapılan 'DEĞİL' (NOT) kapısı ortak emiter bağlantısındadır ve tersleme işlemi yapar (Şekil 7.7.a). Transistörün A girişi '1' ise transistör iletkendir. Bu anda çıkış ucu transistör üzerinden şase potansiyeline bağlanacağından '0' olur. Giriş '0' ise transistör yalıtkandır ve uygulanan gerilim transistörün E – C uçlarında gözükeceğinden çıkış '1' olur.



a) 'DEĞİL' Kapısı b) 'VEYA' Kapısı

Şekil 7.7. RTL tekniği ile yapılan 'DEĞİL' kapısı ve 'VEYA' kapısı.

Şekil 7.7.a'da görülen RTL tekniği ile yapılan 'VEYA' kapısında, girişine '1' uygulanan transistör iletken olur. +V gerilimi, transistörün iletken olmasıyla R_E üzerinde gözükeceğinden çıkışta '1' oluşur. Her iki girişin '0' olması durumunda transistörler yalıtkandır ve devreden akım akmayacağından dolayı çıkış 0'dır.

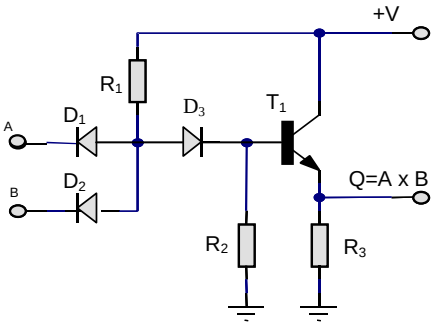
RTL tipi entegrelerin çalışma gerilimi 3V-3.6V arasındadır. RTL tipi entegreler 700 ve 900 ile başlayan sayılarla kodlanırlar.

7.3.1.3. Diyot Transistör Lojik (DTL)

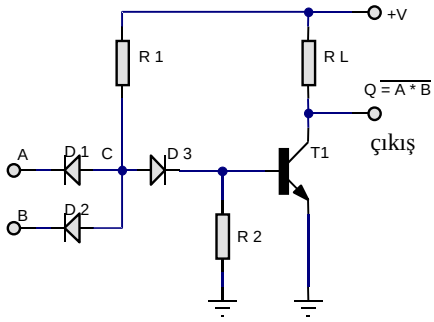
DTL tekniği; RTL tipi entegrelere göre hız, güç ve kararlılık bakımından daha iyi durumda olmalarına rağmen, uygulamadan kalkan ve yerini TTL teknolojisine bırakan entegrelerdir.

DTL teknolojisinde üretilen temel lojik eleman 'VEDEĞİL' kapısıdır (Şekil 7.8). Bu devrede A ve B girişlerinden biri veya her ikisi '0' olduğunda, girişi '0' olan diyot iletken olur ve C noktasını şase potansiyeline getirir. Bu anda D_3

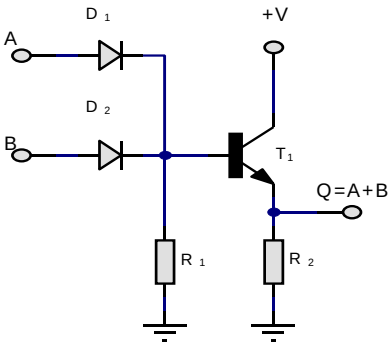
diyodu ters polarma olacağından yalıtkandır ve beyz gerilimi sağlanmayan transistör yalıtımdadır. Devreye uygulanan V gerilimi, transistör E-C arasında okunacağından çıkış ‘1’ olur.



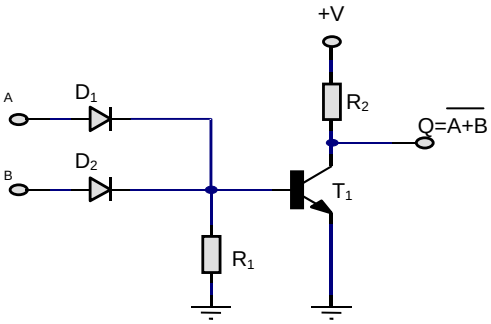
‘VE’ Kapısı



‘VEDEĞİL’ Kapısı



‘VEYA’ Kapısı



‘VEYADEĞİL’ Kapısı

Şekil 7.8. DTL teknolojisi ile temel kapı devrelerinin oluşturulması.

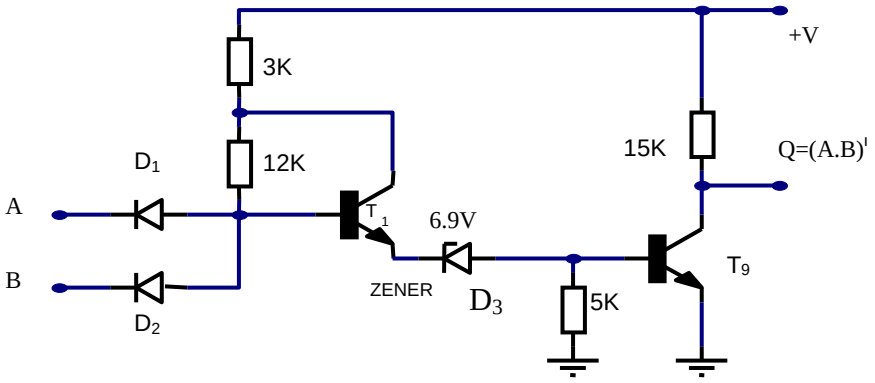
Her iki girişin ‘1’ olması durumunda; D_1 ve D_2 diyotları yalıtkan olacağından, R_1 üzerinden gelen pozitif gerilim ile D_3 iletken olur. Bu durumda, R_1 üzerinden ‘+’ gerilim alan transistör iletime geçer. V gerilimi, R_L ve transistör üzerinden akar ve çıkış ‘0’ olur.

Aynı teknoloji ile yapılan ‘VEYA’, ‘VEYADEĞİL’ ve ‘VE’ kapıları açık şemaları Şekil 7.8’de görülmektedir.

DTL devrelerin çalışma gerilimleri 5V civarındadır. Gürültü bağışıklığı düşük olan DTL entegreler, 830 ve 930 ile başlayan sayılarla kodlanırlar.

7.3.1.4. Yüksek Eşikli Lojik (HTL)

HTL tipi entegreler, DTL tipi entegrelerdeki D_3 diyotu yerine zener diyot konularak gerçekleştirilir. Böylece endüstride çeşitli elektromekanik cihazların kontrol devrelerinde kullanabilecek gürültü bağışıklığı çok iyi özel entegreler oluşur.



Şekil 7.9. HTL teknolojisi ile yapılan 'VEDEGİL' kapısı.

Şekil 7.9'da HTL tipi olarak gerçekleştirilen ve DTL tipine çok benzeyen 'VEDEĞİL' kapısı açık şeması görülmektedir. Bu devrenin DTL teknolojisi ile üretilen 'VEDEĞİL' kapısından farkı, D_3 diyodu yerine zener diyot kullanılmasıdır. Güç harcamasını azaltmak için direnç değerlerinin artırıldığı bu devrede, T_2 transistörünün iletime geçme gerilimi;

$V_{BE}+V_{D3} = 0,7+6,9 = 7,6V$ civarındadır.

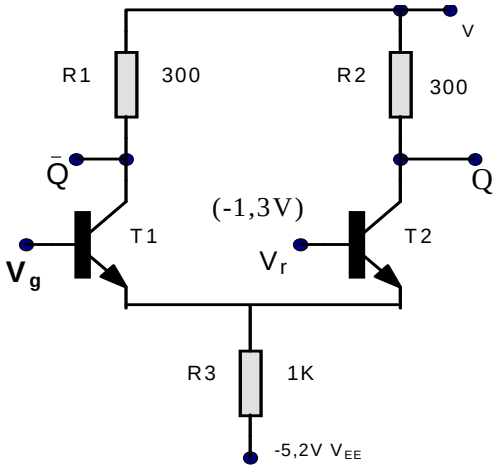
HTL tipi entegreler, çalışma gerilimleri 15V ve yayılım gecikmesi en büyük olan entegre çeşididir.

Gürültü seviyesinin yüksek olduğu bu tip entegreler, 660'lı sayılarla kodlanırlar. Örneğin; Motorola MC660, MC661 VE MC668 gibi.

7.3.1.5. Emiterden Kuplajlı Lojik (Emiter Coupled Logic - ECL)

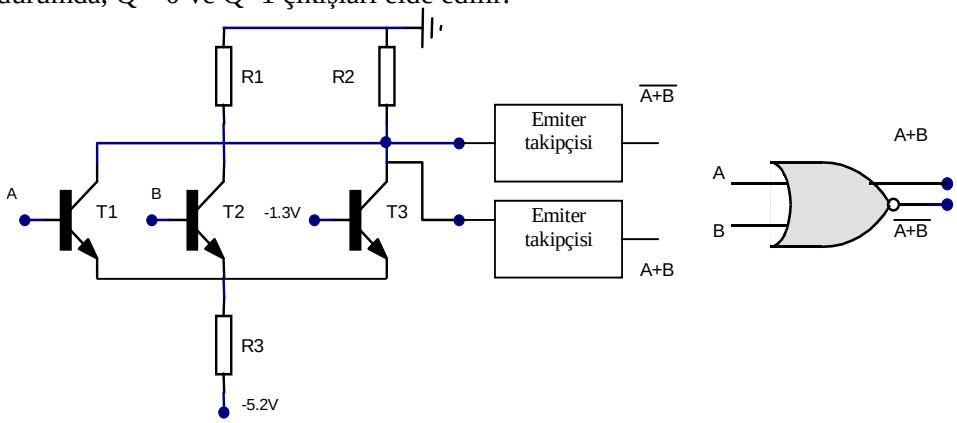
Mantık sınıfları içerisinde en hızlı yapıya sahip olan Emiterden Bağlı Lojik (ECL) devrelerde, kullanılan transistörler doyuma gitmeden çalışırlar. Transistörlerin doyuma gitmeden çalışması, transistörlerin durum değiştirmesi için gerekli akımı düşürür ve anahtarlama hızını artırır. Bu da ECL mantığı ile üretilen elemanların hızının artması sonucunu doğurur.

ECL mantığı ile yapılan devreler, '**akım modlu mantık'** (Current Mode Logic–CML) olarak ta isimlendirilirler. ECL mantık kapıları, birbirlerinin tümleyeni olan iki çıkış verirler. Temel olarak farksal yükselteç yapısındaki ECL devresi, Şekil 7.10'da görülmektedir.



Şekil 7.10. ECL mantığı ile oluşturulan 'DEĞİL' kapı devresi.

Şekil 7.10'daki devrede, V_g giriş geriliminin V_r referans gerilimine karşı aldığı duruma göre, T_1 veya T_2 transistörü iletime geçer. V_g gerilimi, V_g= -1.7 V (lojik 0) veya Vg=-0.8V (lojik 1) değerlerinden birisini alır. $V_g > V_r$ durumunda; T_1 iletime geçerek, R_1 ve transistörün üzerinden R_3 - T_1 direncine doğru bir akım akmasına neden olur. T_1 doyumda T_2 kesimde olarak özetlenebilecek bu durumda, Q' =0 ve Q=1 çıkışları elde edilir.



Şekil 7.11. ECL mantığı ile oluşturulan 'VEYA' / 'VEYADEĞİL' kapısı ve sembolü.

Vr>Vg olması durumunda; T_2 transistörü T_1'e göre daha çok iletken duruma geçer. R_2 direnci ve T_2 transistörü üzerinden R_3'e doğru bir akım akar. T_1 yalıtımda ve T_2 doyumda şeklinde özetlenebilecek bu durumda, Q'=1 ve Q=0 değerleri oluşur. T_1 veya T_2'nin doyumda olduğu durumda, R_3 direncinden ortalama 3 mA' lik akım geçer.

Lojik girişlerin sayısı, kullanılan transistörlere paralel transistörler bağlamak suretiyle artırılabilir. Ayrıca giriş-çıkış gerilim seviyeleri arasındaki farkı ortadan kaldırmak için temel ECL devresinin çıkışlarına, emiterlerinden çıkış alınan ve emiter takipçisi olarak çalışan transistörler bağlanır (Şekil 7.11).

Şekil 7.10'daki temel ECL devresi girişlerinin artırılması ile, Şekil 7.11'de görüldüğü gibi VEYA / VEYADEĞİL devresi olarak kullanılabilir.

Anahtarlama hızı çok yüksek olan ECL'de yayılım gecikmesi 1 nsn, gürültü bağışıklılığı yaklaşık 250 mV'tur. Fan-Out'u 25 civarında olan ECL mantıkta, temel lojik kapılardaki güç tüketimi 40 mW'tır.

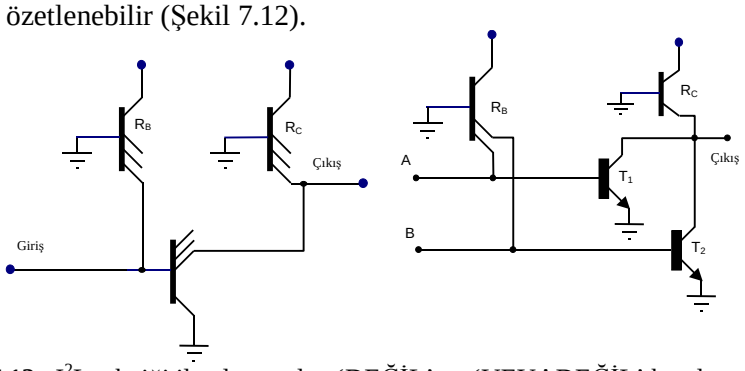
Yüksek frekanslı uygulamalarda kullanılan ECL elemanlar; düşük gürültü bağışıklılığı, yüksek güç tüketimi ve negatif gerilim kullanma gibi özellikler nedeniyle kullanılmamaktadırlar. Ayrıca ECL elemanların TTL ve MOS elemanlarla uyumlu çalışması çok zordur.

7.3.1.6. Entegreli Enjeksiyon Lojik (Integrated Injection Logic – IIL - I^2L)

Entegreli Enjeksiyon lojik kapıları; temel bazı farklılıklar dışında RTL lojik ile benzer çalışma özelliğine sahiptirler. Yapım olarak farklılıklar;

i- RTL kapıda kullanılan beyz ve kollektör dirençleri kaldırılarak bunların yerine uygun tip transistörler bağlanması,

ii- I^2L'de kullanılan transistörlerin birden fazla kollektöre sahip olması olarak özetlenebilir (Şekil 7.12).



Şekil 7.12. I^2L tekniği ile oluşturulan 'DEĞİL' ve 'VEYADEĞİL' kapıları.

I^2L entegre devre yapımında çok yer kaplayan dirençlerin kullanılmaması, bir entegre içerisine daha fazla eleman yerleştirilmesi ve maliyetin ucuzlaması sonucunu doğurur.

7.3.1.7. Tansistör Transistör Lojik (TTL)

DTL entegrelerin gelişmiş şekli olan ve giriş olarak çok elemanlı (emiterli) transistörlerin kullanıldığı TTL entegreler, en yaygın kullanılan dijital entegre grubudur. RTL ve DTL sınıflarının devamı olarak piyasaya çıkan TTL'de temel eleman 'VEDEĞİL' kapısıdır. Bununla beraber, bütün lojik kapıların TTL mantığı ile oluşturulması mümkündür. TTL entegreler 7 alt gruba ayrılabilir.

1- Standart TTL – (Standart TTL – TTL) - 74XX ile gösterilir.

2- Düşük Güçlü TTL – (Low power TTL – LTTL) – 74LXX ile gösterilir.

3- Yüksek Hızlı TTL – (High speed TTL – HTTL) – 74HXX ile gösterilir.

4- Şotki (Çok hızlı) TTL – (Schottky TTL – STTL) –74SXX ile gösterilir.

5- Düşük Güçlü Şotki TTL – (Low power schottky TTL – LSTTL) – 74LSXX ile gösterilir.

6- Geliştirilmiş Şotki ve Geliştirilmiş Düşük Güçlü Şotki TTL (Advanced Schottky and Advanced – Low Power Schottky TTL)- 74ASXX ve 74ALSXX ile gösterilir.

7- Hızlandırılmış- Geliştirilmiş Şotki TTL (Fast – Advanced Schottky TTL – FASTTL) – 74FXX ile gösterilir.

Bu alt guruplar arasında çeşitli paramatre farklılıkları mevcuttur. Ancak tüm alt guruplar genelde 5V (± 0.25 V) besleme gerilimi ile çalışırlar.

AA74YYXXX şeklinde bir isimlendirmenin kullanıldığı TTL entegrelerde, AA harfleri üretici firmanın adını gösteren harflerdir. Örneğin; DM: National Semiconductor, SN: Texas Instument ve S: Sygnetics firmasını temsil etmektedir. 74 serisi endüstri standardı olarak kullanılırken, 54 serisi askeri uygulamalarda kullanılır.

74 sayısı ile rakamlar arasındaki YY harfleri entegrenin hangi alt guruplandırma sınıfına ait olduğunu gösterir. XXX, parçanın fonksiyonunu gösteren iki veya üç basamaklı sayıdır. Örneğin; 7406 entegresi altı adet 'DEGİL' kapısı ve 74502 entegresi, dört tane iki girişli 'VEYADEĞİL' kapısını içerirken, 7430 bir tane 8 girişli 'VEDEĞİL' kapısını içerir.

TTL entegreler alt grupları arasındaki diğer bir fark, entegrelerin sahip oldukları çıkış şeklidir. Uygulamalarda, TTL entegrelerin hangi alt grupta olduğunun

bilinmesinin gerekebileceği durumlar olabileceği gibi, entegrenin hangi tip çıkışa sahip olduğunun bilinmesinin gerekebileceği durumlar da olabilir. Bu nedenle TTL alt sınıflandırmasının özelliklerini daha sonrada TTL entegrelerin sahip oldukları çıkış şekillerini inceleyelim.

Standart TTL (TTL)

Çok geniş bir kullanım alanı bulunan ve çok sayıda üretici tarafından üretilen standart TTL entegreleri, 54 ve 74 serileri ile tanımlanırlar. 74 serisi ve 54 serisi standart TTL entegreler arasındaki farklar; çalışma gerilimleri ve çalışma sıcaklığı sınırlarıdır. Örneğin; 54 serisi 4,5-5,5V arasında çalışırken, 74 serisi 4,75-5,25V arasında çalışır.

Her bir kapı için 10mW güç tüketimine sahip standart TTL entegrelerde, ortalama yayılım gecikmesi 9 nsn ve max. Hız 35MHz'dir. Standart TTL entegreler 10 adet TTL devreyi sürebilir. Standart TTL entegreler; SSI gruplandırması içerisinde kapı devreleri ve flip-flop'larda kullanılırken, MSI teknolojisinde ise kaymalı kaydedici, sayıcı, kod çözücü ve bellek devrelerinde kullanılır.

Düşük Güç Tüketimli TTL (LTTL-74L Serisi):

Standart TTL ile aynı devre yapısına sahip bu tip TTL'lerde yalnızca tüm dirençlerin değerleri artırılmıştır. Direnç değerlerinin arttırılması güç tüketimini azaltmış (1 mW) fakat yayılım gecikmesini arttırmıştır (33 nsn). 74L serisi entegreler güç tüketiminin önemli olup, hızının önemli olmadığı yerlerde (hız max. 3MHz'e düşmüştür) tercih edilirler. Ancak daha sonra anlatılacak entegre grupları daha iyi performansa sahip olduğundan, diğer gruplar 74L serisine tercih edilirler.

Yüksek Hızlı TTL (HTTL-74H Serisi):

Yüksek hızlı TTL serisi, standart TTL serisindeki direnç değerlerinin düşürülmesi ve çıkış transistörü yerine Darlington bağlantı konulması yoluyla üretilir. Bu değişiklikler daha yüksek anahtarlama hızı sonucunu (ortalama yayılım gecikmesi 6 nsn) doğurur. Ancak artan hız (50MHz) beraberinde daha çok güç tüketimini getirir (ortalama 23 mW). Schottky TTL'lerin kullanılmasıyla 74H serisinin kullanımı azalmıştır.

Şotki (Schottky) TTL (STTL-74S Serisi):

Standart, 74L, 74H serisi TTL entegrelerin hepsi kullanılan transistörlerin doyum durumlarında oluşan anahtarlama yöntemini kullanırlar. Bu durum

transistörün doyum-kesim durum değişimi sırasında bir zaman gecikmesine neden olur ve devrenin anahtarlama hızını sınırlar.

74S serisi TTL devre yapısı, transistörün tam anlamıyla yalıtıma gitmesini engelleyerek zaman gecikmesini düşürür. Bu işlem her transistörün beyzi ile kollektörü arasına Şotki engelleme diyodu (Schottky Barier Diode-SBD) bağlanarak sağlanır. SBD diyodu yalnızca 0,25V doğru yönlü polarmalandırma gerilimine sahiptir. Bu nedenle B-C bileşiminin ileri yönde polarması durumunda, diyot beyz akımının bir kısmını üzerinden akıtır.. Bu transistörün kesime gitmesi anındaki gerekli zamanı azaltır.

Anahtarlama zamanını küçültmek için küçük değerlikli dirençler kullanan 74S serisinde ortalama güç tüketimi yaklaşık 20 mW'tır. Bu değer 74H serisi ile aynıdır ve sonuçta aynı güç tüketimine karşılık iki kat hızlı eleman elde edilir.

Düşük Güçlü Şotki TTL - 74LS Serisi :

LS-TTL serisi entegreler, S-TTL serisi entegrelerin daha düşük güç ve daha düşük hızlı versiyonudur. 74S serisi entegrelerinkinden daha büyük değerli dirençler kullanılması, devrenin güç tüketimini azaltır fakat anahtarlama zamanının uzamasına neden olur.

74LS serisindeki bir 'VEDEĞİL' kapısı, ortalama 9,5 ns yayılım gecikmesine karşılık 2 mW güç tüketimine sahiptir. Standart TTL serisi ile yaklaşık aynı anahtarlama hızına sahip 74LS entegrelerdeki güç tüketimi çok daha azdır. Bu nedenle, minimum güç tüketiminin gerekli olduğu yerlerde tercih edilirler.

Geliştirilmiş Şotki TTL (74AS) ve Geliştirilmiş Düşük Güçlü Şotki TTL (74ALS) Serisi:

Entegre devre teknolojisinde yakın zamanda olan gelişmeler iki yeni tip TTL serisinin geliştirilmesini sağlamıştır: Geliştirilmiş Şotki (74AS) serisi ve Geliştirilmiş Düşük Güçlü Şotki (74ALS) serisi. 74 ALS serisi entegreler, 74AS serisine göre daha düşük güç tüketiminde daha hızlı bir özelliğe sahiptir. 74ALS serisinin fiyatları düşmeye devam ettiği sürece ve yeni fonksiyonlar gerçekleştiren entegreler geliştirildikçe, 74ALS serisi entegreler 74AS serisinin yerini alacak görünmektedir.

74ALS serisi entegreler, 74LS entegrelere göre hız ve güç tüketimi açısından üstündürler. 74ALS serisi entegreler, TTL entegreler içerisinde en düşük hız-güç üretimine sahip elemanlardır. Yaklaşık en düşük güç tüketim seviyesine (1 mW) eşit güç tüketimine sahip 74ALS serisi, yakın zamanda diğer TTL serileri yerine kullanılmaya başlanacaklardır.

7.3.1.7.1. TTL Entegrelerin Sahip Oldukları Çıkış Şekilleri

TTL entegrelerde çıkış katları, değişik uygulamalar göz önüne alınarak farklı şekillerde imal edilirler. Her çıkış şeklinin imalatçı ve uygulayıcılar açısından çeşitli üstünlükleri ve sakıncaları vardır. TTL lojik entegreler, çıkış şekillerine göre üç temel grup altında toplanabilirler;

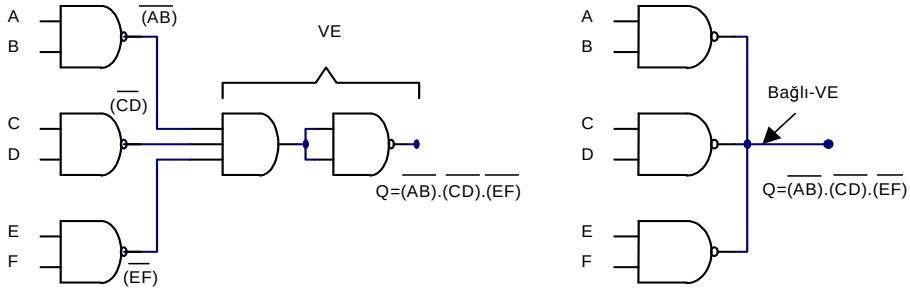
i- Yukarı çekici çıkışlı TTL,

ii- Açık kollektör çıkışlı TTL,

iii-Üç durumlu çıkışlı TTL.

i- Yukarı çekici çıkışlı TTL'ler (Totem Pole Outputs)

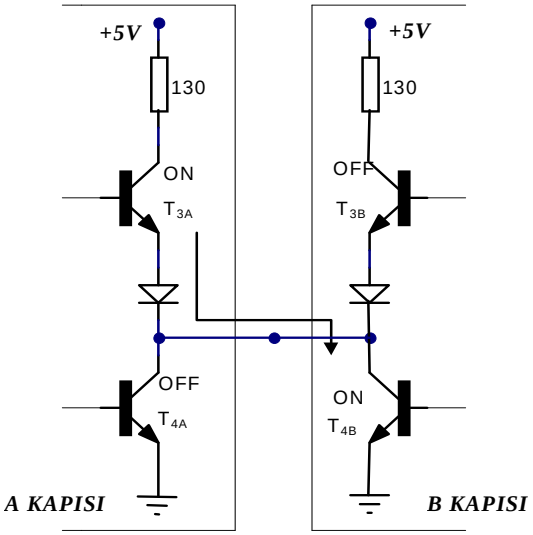
Şekil 7.13'te verilen devrenin incelenmesinden, çıkışta bulunan 'VE' kapısının birleştirme işlemi yaptığı ve çıkış fonksiyonunun $Q = (AB)'.(CD)'.(EF)'$ olduğu görülür. Aynı lojik işlem, 'VEDEĞİL' kapılarının çıkışlarının birlikte bağlanmasıyla elde edilebilir (Şekil 7.13.b). Bu bağlantı 'VE' işlemini gerçekleştirdiği için '**Bağlı-VE / Wired-AND**' olarak isimlendirilir.



(a) **(b)**

Şekil 7.13. Kapı devrelerinin çıkışlarının birleştirilmesi ve Bağlı - VE işlemi.

‘VEDEĞİL’ kapısı çıkışlarının Şekil 7.13.b’deki gibi aynı noktaya bağlanması ve çıkışlardan birisinin ‘0’ olması durumunda; ilgili kapıyı oluşturan transistör kısa devre olur ve tüm çıkışlar ‘0’ konumuna gider. Ortak çıkış noktası yalnızca tüm çıkışların ‘1’ olması durumunda ‘1’ olabilir. Bu sonuç açıkça ‘VE’ işlemini vermektedir.



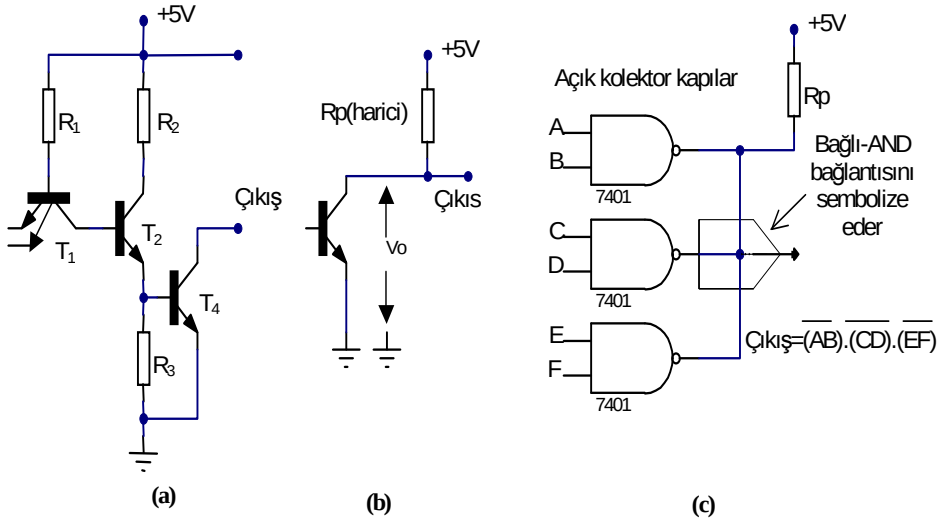
Şekil 7.14. Yukarı çekişli çıkış bağlantısı ve yukarı çekişli çıkışların bir arada bağlanması durumunda transistörler üzerinden akan akımlar.

‘Bağlı-VE’ bağlantısı daha az sayıda kapı elemanı kullanılarak ‘VE’ işleminin elde edilmesi sonucunu verir. Ancak TTL entegrelerde yaygın olarak kullanılan yukarı çekici çıkışlı elemanların bu şekilde bağlanmaları sakıncalıdır. Çünkü, farklı çıkışlara sahip kapıların birlikte bağlanması, iletimde olan transistörler üzerinden büyük bir akım akmasına neden olarak elemanları bozabilir.

Örneğin, Şekil 7.14’de görülen devrede, B kapısının çıkışının ‘0’ olması durumunda, T_{3A} ve T_{4B} üzerinden büyük bir akım akar. Bu akım, T_{3A} veya T_{4B}’yi bozabilir. Aynı devrede A kapısının çıkışının ‘1’, B kapısının çıkışının ‘0’ olması durumunda; T_{3B} ve T_{4A} üzerinden büyük bir akım akar. T_{4A} transistörü yük olarak çok küçük bir direnç göstereceğinden, devreden akan akım T_{3B} veya T_{4A}’yı bozabilir.

ii-Açık kollektör çıkışlı TTL'ler (Open Collector Outputs)

Lojik devrelerde 'Bağlı-VE' bağlantısına imkan tanımak amacıyla TTL entegreler açık kollektör (open-collector) çıkışlı olarak imal edilirler. Açık kollektör yapısı tampon (buffer), sürücü (driver) veya tampon / sürücü olarak isimlendirilir ve normal lojik elemanlara göre daha büyük akım ve/veya gerilim kapasitesi veren devrelerde kullanılır.



Şekil 7.15. Açık kollektör 'Bağlı-VE' TTL devre bağlantısı ve **'**Bağlı-VE' bağlantının sembolize edilmesi.

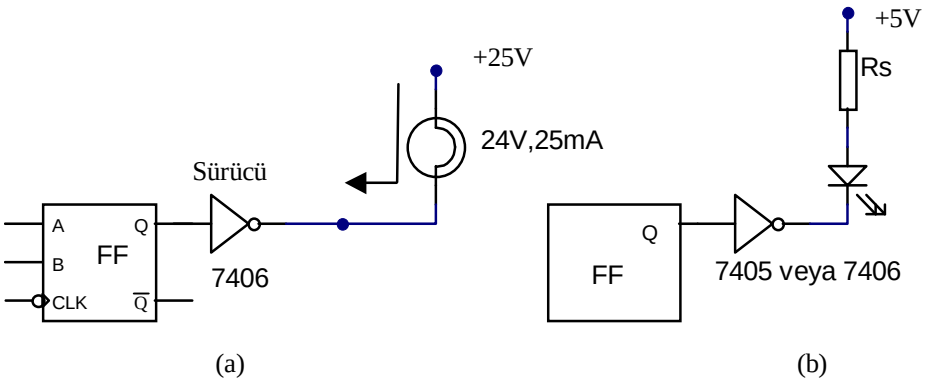
Açık kollektörlü yapıda, elemanlardan bir kısmı kullanılmayarak çıkış Şekil 7.15.a'da görüldüğü gibi T_4 transistörünün kollektöründen alınır.

T_4 transistörünün iletimde olduğu anda çıkış '0', T_4'ün kesimde olduğu anda çıkış 1'dir. Devrenin düzenli olarak çalışması için, R_p olarak isimlendirilen pull-up direncinin Şekil 7.15.b'deki gibi bağlanması gerekir. R_p direnci yardımıyla, transistörün kollektöründe '1' değerinin okunması mümkün olur. T_4'ün iletimde olması durumunda T_4'ün kollektöründe gerilim oluşmaz.

Açık kollektörlü kapılar ile 'Bağlı-VE' işlemi güvenli olarak gerçekleştirilebilir. Şekil 7.15.c'de iki girişli 'VEDEĞİL' kapılarıyla 'Bağlı-VE' işleminin oluşturulması sembolize edilmektedir. Devrenin düzgün çalışması için, R_p direncinin değerinin yüksek tutulması gereklidir. Ancak, varlığı devrenin anahtarlama hızını azaltan R_p direncinin değerinin büyük tutulması hızı daha da azaltır. Bu nedenle, açık kollektör çıkışlı devreler anahtarlama hızının önemli olduğu yerlerde tercih edilmezler.

7406 entegresi; yaygın alarak kullanılan, açık kollektör yapıya sahip, 6 adet 'DEĞİL' (NOT) içeren bir devre elemanıdır. Çıkışın '0' olması durumunda 40mA'e kadar akım çekebilen bu entegre, 30V'a kadar çıkış verebilir. 7406 entegresinde bulunan kapılar normal lojik elemanlar ile yüksek gerilim ve akımda çalışan (Örneğin; 24V, 25mA) elemanlar arasında tampon olarak kullanılabilirler (Şekil 7.16). Şekil 7.16.a'daki devrede, Q çıkışının '1' olması durumunda sürücünün çıkışı '0' olacağından lamba 24V'da 25mA'lik bir akım çekerek yanar. Çıkışın '1' olması durumunda ise akım akmaz ve lamba yanmaz.

Açık kollektörlü elemanın LED'leri sürmek için kullanılması mümkündür (Şekil 7.16.b). LED'in çekeceği akıma göre uygun sürücü elemanı seçilebilir. Devredeki R_S direnci, akım sınırlayıcı direnç olarak kullanılır.



Şekil 7.16. Yüksek gerilim / yüksek akımlı devrenin ve LED'in açık-kollektör çıkış ile sürülmesi.

iii-Üç Durum Çıkışlı TTL'ler (Tristate TTL)

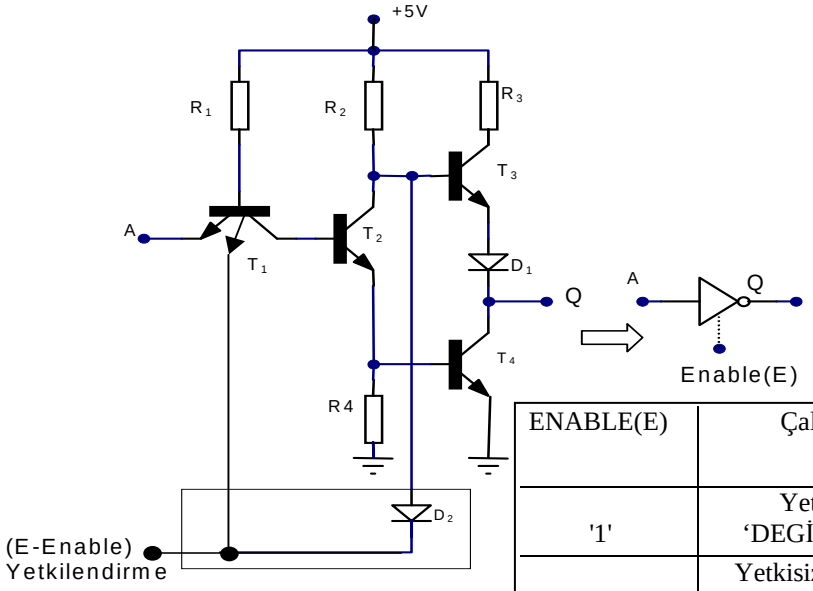
Farklı bir TTL çıkış şekli, Bağlı-VE bağlantıya imkan tanıyan ve yukarı çekişli çıkış gibi yüksek hızda çalışabilen üç konumlu TTL'dir. Bu bağlantı şeklinin üç durumlu TTL olarak isimlendirilmesinin nedeni, çıkışın üç durumdan birisinin olabilmesidir: '1', '0' ve yüksek empedans (Hi-Z). Hi-Z durumu; yukarı çekişli bağlantıdaki her iki transistörün de kesimde olduğu ve bu nedenle çıkış terminalinin 'V_{cc}' ve şaseye göre yüksek empedansa sahip bulunduğu bir durumdur. Diğer bir deyişle; çıkış ne '0', ne '1', ne de açık devredir. Çıkış, $+V_{cc}$ veya şaseye göre birkaç MΩ'luk bir dirence sahiptir.

Yüksek empedans işlemi, temel yukarı çekişli devresine Şekil 7.17'de görülen ekleme yapılarak elde edilebilir. 'DEGİL' işlemini gerçekleştiren bu devrede

iki giriş bulunur: Normal girişi temsil eden A ve Hi-Z durumunu üreten, ‘E’ ile gösterilen **‘yetkilendirme’** (enable) girişi.

Yetkilendirme girişinin durumuna göre oluşabilecek olaylar aşağıdaki gibi özetlenebilir:

Yetkilendirilmiş (Enabled) Durum: E=1 olduğu zaman devre ‘DEĞİL’ kapısı olarak çalışır. Çünkü E’deki gerilimin T_1 ve D_2 üzerinde hiçbir etkisi yoktur. Bu durumda çıkış girişin tersi olarak elde edilir.



ENABLE(E)	Çalışma Şartları
'1'	Yetkilendirilmiş ‘DEGİL’ olarak çalışır
'0'	Yetkisizlendirilmiş Çıkış Hi-Z ve A girişinin etkisi yok

(a) **(b)**

Şekil 7.17. Üç durumlu çıkışa sahip ‘DEĞİL’ lojik kapı devresi.

Yetkisizlendirilmiş (Disabled) Durum (Hi-Z): E=0 olduğu zaman, devre A lojik girişinin durumuna bakmaksızın Hi-Z konumuna gider. E girişinde 0 olması, T_1 transistörünün B-E birleşimini doğru yönde polarmalandırarak R_1’den akan akımı şase yapar ve bu durum T_2 ve T_4’ü yalıtıma götürür. E’de 0 olması D_2 diyodunu iletime götüreceğinden, T_3 transistörü yalıtıma geçer. Yukarı çekişli bağlantılı T_3 ve T_4 transistörleri yalıtımda olduğundan, çıkış terminali açık devredir (Şekil 7.17.b).

Yüksek empedanslı çıkış, sağladığı avantajlar nedeniyle Flip-Flop, kod çözücü, çoklayıcı, bellek, ADC, mikroişlemci, vb. devrelerde yaygın olarak kullanılır.

7.3.1.7.2. TTL Serisindeki Entegrelerin Karşılaştırılması

Tablo 7.2'de, TTL serilerindeki entegrelerin önemli özelliklerinden bazıları verilmektedir. Tabloda verilen tüm serilerde, performans oranları (tetikleme hızı hariç) bir 'VEDEĞİL'kapısı için verilmiştir.

PERFORMANS ORANLARI	74	74L	74H	74S	74LS	74AS	74ALS
Yayılım Gecikmesi (nsn)	9	33	6	3	9.5	1.7	4
Güç Tüketimi (mW)	90	33	138	60	19	13.6	4.8
Hız – Güç Üretimi (pj)	35	3	50	125	45	200	70
Maksimum tetikleme hızı (MHz)	10	20	10	20	20	40	20
Fan – Out							
GERİLİM ORANLARI							
V_{OH} (min)	2,4	2,4	2,4	2,7	2,7	2,5	2,5
V_{OL} (max)	0,4	0,4	0,4	0,5	0,5	0,5	0,4
V_{IH} (min)	2,0	2,0	2,0	2,0	2,0	2,0	2,0
V_{IL} (max)	0,8	0,7	0,8	0,8	0,8	0,8	0,8

Tablo 7.2. TTL alt serisindeki entegrelerin karşılaştırılması.

Tablodaki değerleri kullanarak 74LS tipi entegre için gürültü aralığını bulup standart TTL ile karşılaştıralım.

$V_{LH} = V_{OH}(\text{min}) - V_{IH}(\text{min}) = 2,7V - 2,0V = 0,7V$ (Standart TTL'nin değeri 0,4V).

$V_{HL} = V_{OL}(\text{max}) - V_{IL}(\text{max}) = 0,8V - 0,5V = 0,3V$.

7.3.1.7.3. TTL Entegrelerde Bulunan Parametrelerin İncelenmesi

TTL entegrelerin sahip oldukları özellikler, 'Veri El Kitabı - Data Hand Book' adı verilen veri kataloglarında özetlenmiştir. Verilen özellikler aşağıdaki gruplar altında incelenebilir.

i- Elektriksel verileri.

ii- Darbe davranışları.

iii- Ayak bağlantıları ve çalışma (fonksiyon) tablosu.

iv- Elemanların tavsiye edilen çalışma verileri.

i- Elektriksel Veriler

Lojik elemanların ‘1’ ve ‘0’ durumlarında çalışabilmeleri için tespit edilen sınır değerlerini açıklayan veriler, ‘**Elektriksel veriler’** olarak isimlendirilir. Lojik gerilim ve akım seviyeleri olarak da isimlendirilen bu veriler, TTL entegrelerin çeşidine göre değişiklik gösterseler de, ortalama olarak ifade edilebilirler.

V_{IH} : Sayısal entegrenin girişinde ‘1’ durumunu ifade eden gerilim değeridir (V_{IHmin}=2,0V).

V_{IL} : TTL entegrenin ‘0’ olarak algıladığı gerilimin alabileceği en yüksek değerdir (V_{ILmax} =0,8V).

V_{OH}: Sayısal entegrenin çıkışında ‘1’ değerini temsil eden en küçük gerilim değeridir ($V_{OH\ min}$=2,4V).

V_{OL}: Çıkış geriliminin ‘0’ olarak algılanabilmesi için olabilecek maksimum gerilim değeridir (V_{OLmax}=0,4V).

I_{IH}: Entegredeki giriş değeri ‘1’ iken girişin çektiği akım değeridir (TTL’de 40 μA/giriş).

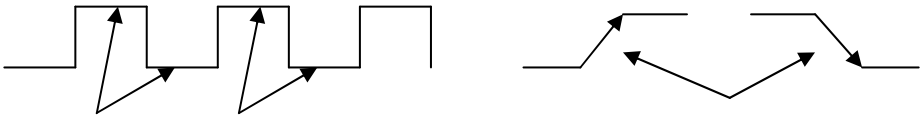
I_{IL}: Girişteki gerilim ‘0’ iken girişten akan akım değeridir. Akan akımın yönü kapıdan dışarı doğru ise (-) olarak, kapıdan içeri doğru ise (+) olarak düşünülür. Bu değer kapı başına 1,6mA’ dir.

I_{OH}: Çıkışta ‘1’ varken dışarı doğru akan akımdır (TTL’de 0,4 mA/kapıdır).

I_{OL}: Çıkış ‘0’ durumunda iken kapıdan akan akım değeridir. Kapıdan içeri doğrudur ve 16mA’dir

ii- Darbe Davranışları

Sayısal devreler, uygulanan işaretlerin seviye değişikliklerine veya işarette olan ani değişikliklere cevap verirler. Seviye değişikliklerine göre çalışan devrelere, **‘seviye tetiklemeli devreler’** denir. İşaretteki ani değişikliklere yanıt vererek çalışan devreler, **‘geçiş veya kenar tetiklemeli devreler’** olarak adlandırılır (Şekil 7.23).

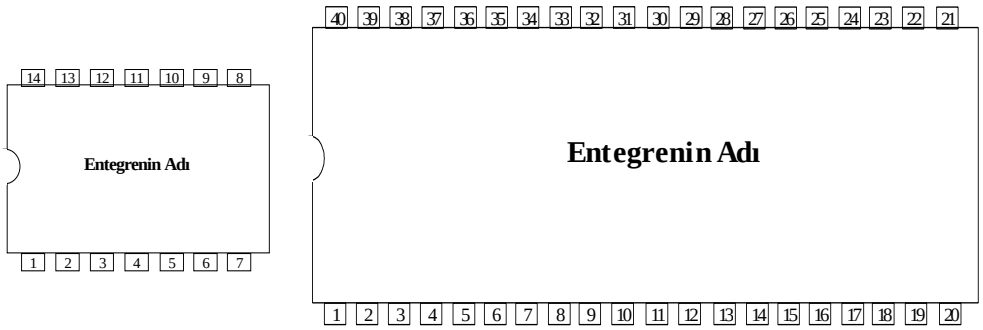


a) Seviye Tetiklemesi **b)** Kenar Tetiklemesi

Şekil 7.18. Lojik entegrelerde seviye ve kenar tetikleme durumları.

iii- Bağlantı Ayakları ve Çalışma Sayfası

Dijital entegreler genellikle iki tarafında çizgi şeklinde ayağı olan entegre yapısında **'Çift hatlı paket – Dual inline package - DIP**' imal edilirler. Şekil 7.19'da dijital entegrenin şekli ve ayak bağlantı numaraları görülmektedir. Entegrenin ayak numaraları, entegrenin üzerindeki noktadan veya girintiden başlayarak saat yönünün tersi yönünde sıralanır.



Şekil 7.19. Entegre ayak numaralarının verilmesi.

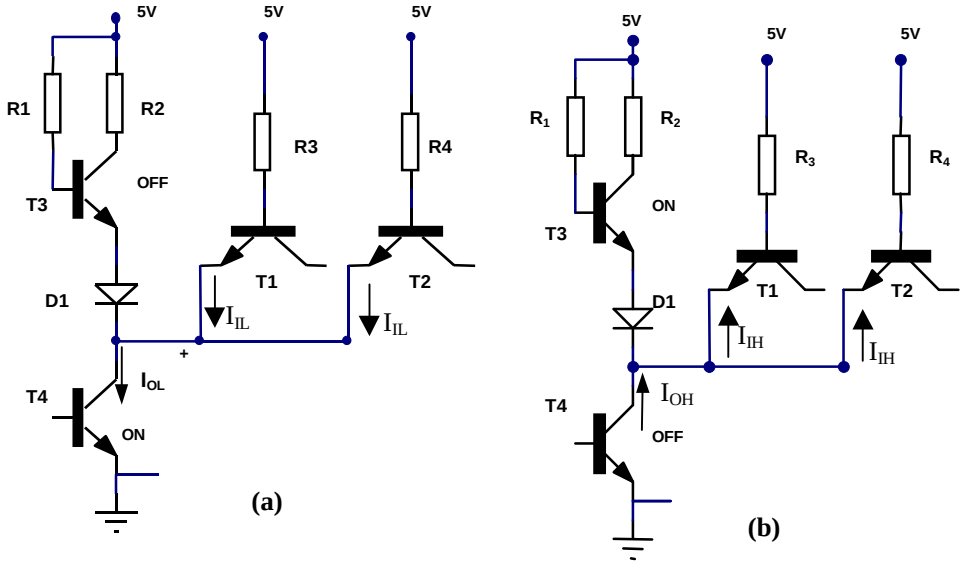
Entegrenin çalışma tablosu (data sheet), elemanların standart karakteristiklerini gösterir. Her biri farklı bir entegreye sahip özellikleri gösteren sayfaların bir araya getirilmesi ile entegre veri katalogları (IC Data Book) oluşur. TTL ve CMOS lojik ailelerde bulunan entegreleri içeren genel lojik entegre katalogları yanı sıra, yanlızca TTL entegreleri içeren TTL veri katalogları veya yanlızca CMOS entegreleri içeren CMOS veri katalogları bulunmaktadır.

iv- Çalışma Verileri

a) TTL Yük sürme (Loading Drive) ve Yüklenme Kapasitesi (Fan-Out)

Bir entegrenin çıkışının yüklenme (Fan-out) veya yük sürme (Load drive) kapasitesinin neyi ifade ettiğini anlamak önemlidir. Şekil 7.25.a'da '0' durumundaki birçok standart TTL girişini süren, standart bir TTL çıkışı görülmektedir. Çıkış '0' iken T_4 transistörü iletimdedir ve girişlerdeki I_{IL}

akımlarının toplamına eşit olan I_{OL} akımı için akım yutucu olarak görev yapar. Transistörün iletim durumunda T_4'ün kollektör-emiter direnci çok küçüktür. Bu durumda, I_{OL} akımı küçük bir V_{OL} gerilim düşümüne neden olur. Bu gerilim, entegrenin V_{OLmax} değerine erişmemelidir. Bu kısıtlama, I_{OL}'nin maksimum değerini ve sürülebilecek eleman sayısını sınırlar.



Şekil 7.20. TTL entegre çıkışının birçok girişi sürdüğü zaman oluşan akımlar.

Benzer bir durum Şekil 7.20.b'de görülen '1' çıkış durumu içinde geçerlidir. T_3 transistörü emiter takipçisi gibi çalışır ve farklı TTL girişlerindeki I_{IH} akımlarının toplamı olan I_{OH} akımını sağlar (current sourcing). Eğer çok fazla sayıda kapı devresi sürülürse; I_{OH} akımı, R_2 direnci, T_3'ün C-E bileşimi ve D_1 üzerinde düşen gerilimleri artırır. Bu gerilimlerin artması, V_{OH} geriliminin $V_{OH\ min}$ seviyesine yaklaşmasına veya $V_{OH\ min}$ seviyesinin altına düşerek belirsiz bölgeye girmesine neden olur.

Buraya kadar anlatılanların anlamı şudur: Bir TTL'nin çıkışı, '0' durumunda ne kadar akımı yutabileceği gösteren I_{OLmax} ve '1' durumunda ne kadar akım sağlayabileceğini gösteren I_{OHmax} sınırlarına sahiptir. Eğer çıkış gerilim seviyeleri katologlarda belirtilen sınırların içerisinde ise, çıkış akım sınırlarına erişilmemelidir.

Herhangi bir entegrenin çıkışının kaç tane farklı girişi sürebileceğini tespit etmek için, çıkışın sağlayabileceği akım değerlerinin (I_{OLmax} ve I_{OHmax}) ve her bir giriş için gerekli akım değerlerinin (I_{IL} ve I_{IH}) bilinmesi gerekir. Bu değerlerin

entegre kataloglarından faydalanarak bulunması ile entegrelerin yük sürme kapasitesi hesaplanabilir.

Örnek 1: Bir 7400 ‘VEDEĞİL’ kapısının çıkışı ile kaç tane 7400 ‘VEDEĞİL’ kapısı sürebileceğini hesaplayalım.

Çözüm : Önce Şekil 7.21.a’da gösterilen bağlantı şeklinde ve çıkışın ‘0’ olduğu anda oluşan şartlar tespit edilir.

Veri kataloglarından entegre için geçerli I_{IL} akım değerleri; $I_{OL(max)}$=16mA ve $I_{IL(max)}$=1,6mA olarak bulunur. Bu değerler, 7400 entegresindeki bir kapının maximum 16mA’i yutabileceğini ve yine her bir kapının kendilerini süren devreye doğru maximum 1,6mA’lik bir akım sağlayacağını gösterir. Bu durumda, çıkışın ‘0’ olduğu anda entegrenin sürebileceği kapı sayısı;

$$\text{Fan out} = I_{OL(max)} / I_{IL(max)} = 16\text{mA}/1{,}6\text{mA} = 10$$

olarak bulunur.

Bu işlemde I_{IL} akımının aslında (-) işaretli bir akım olduğu, ancak burada yaptığımız işlem için bunun gözardı edilebileceği unutulmamalıdır.

Entegrenin çıkışının ‘1’ olduğu durum aynı şekilde incelenir. Entegre kataloglarına baktığımız zaman I_{OH} ve I_{IH} değerleri (sahip oldukları işaretleri gözardı edilerek);

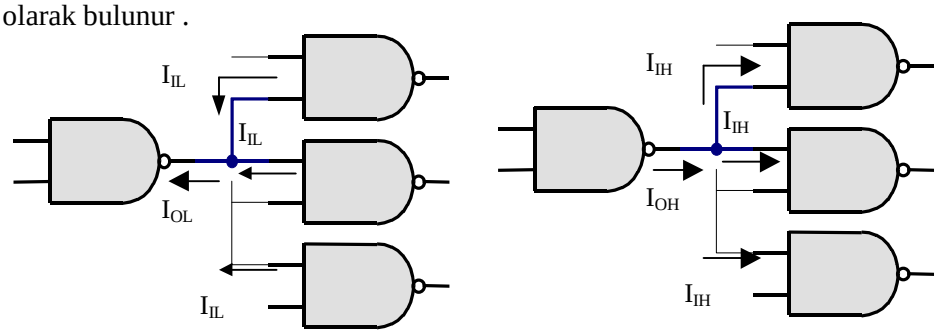
$$I_{OH(max)} = 0{,}4\ \text{mA} = 400\ \mu\text{A} \text{ ve } I_{IH(max)} = 40\ \mu\text{A}$$

olarak bulunur.

Bu değerler ile çıkış ‘1’ durumunda iken sürülebilecek kapı sayısı;

$$\text{Fan out}_{(1)} = I_{OH(max)} / I_{IH(max)} = 400\ \mu\text{A}/40\ \mu\text{A} = 10$$

olarak bulunur .



a) Sürücü çıkışı ve yük girişleri ‘0’ **b)** Sürücü çıkışı ve yük girişleri ‘1’

Şekil 7.21. Entegrelerin sürebileceği kapı sayısının bulunmasında kullanılacak akımlar.

Örnek yaptığımız hesaplamada çıkışın ‘0’ ve ‘1’ olduğu anda ‘Fan Out’ değeri 10 olarak bulundu. Bu nedenle sonuçta Fan Out = 10 denebilir. İki değerin farklı olduğu durumlarda ise küçük olan değer referans olarak alınır.

Örnek 2: 74ALS20 ‘VEDEĞİL’ kapısı ile kaç tane 74ALS20 kapısı sürebileceğini hesaplayalım.

Çözüm : Entegre kataloglarından aşağıdaki değerler okunur.

$I_{OH(max)}$ = 0,4 mA = 400 µA, $I_{OL(max)}$ = 8 mA, $I_{IH(max)}$ = 20 µA, $I_{IL(max)}$ = 0,1mA.

Çıkışın ‘1’ olduğu durumda Fan $Out_{(1)}$ = 400 µA / 20 µA = 20,

Çıkışın ‘0’ olduğu durumda Fan $Out_{(0)}$ = 8 mA / 0,1 mA = 80

olarak bulunur.

Bu durumda, ‘Fan Out’ olarak daha küçük değer olan 20 seçilir.

Örnek 3: 7404 entegresinde bulunan ‘DEĞİL’ kapısı için giriş ve çıkış yükleme değerlerini bulalım.

Çözüm: Veri kataloglarından ‘Input Loading / Fan Out’ başlığı altında ifade edilen değerler bulunur. Okunan ilk değerler 800µA /16mA’dır ve bunun anlamı aşağıdaki şekilde ifade edilir:

7404’ün girişi ‘1’ durumunda iken giriş sinyal kaynağından maximum 40 µA bir akım çekerken, girişin ‘0’ olması durumunda ise maksimum 1,6 mA’lik bir akım sağlar. Bu değerler maksimum akım değerlerini göstermektedir ve pratikte girişte I_{IH} = 10 µA ve I_{IL}=1,1mA değerleri ölçülür.

Veri kitabında görülen ikinci değer 20/10’dur ve bunun anlamı; 7404 çıkışı;

‘1’ durumunda iken; 800 µA akım sağlayabilirken,

‘0’durumunda; 16 mA akım çekebilir.

Bu değerlere göre ‘Fan-Out’ değerlerini hesaplarsak;

Fan-$Out_{(1)}$ =800µA/ 40 µA= 20

Fan-$Out_{(0)}$ =16mA/ 1,6 mA= 10

Olarak bulunur.

Bulunan değerlerden küçük olan seçileceği için Fan-Out= 10 olarak alınır.

Örnek 4: 74LS04 entegresi için yükleme değerlerini bulalım.

Çözüm: Entegrenin 'Input Loading / Fan Out' başlığı altında;

Giriş için; '1' durumunda: 20µA ve

'0' durumunda: 0.4 mA giriş yükleme akım değerleri okunmaktadır. .

Çıkış için ise;

'1' durumunda: 400 µA ve

'0' durumunda: 8 mA

akım çekme kapasitesine sahip olduğu belirtilmektedir.

Bulunan değerlerden entegrenin yüklenme değerleri;

$Fan\ Out_{(1)} = 400\mu A / 20\mu A = 20$ adet,

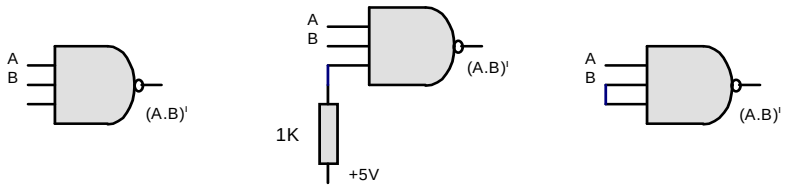
$Fan\ Out_{(0)} = 8mA / = 0{,}4mA = 20$ adet

olarak bulunur.

Yapılan hesaplamalardan, 'Fan Out' olarak 20 değeri elde edilir.

b) Kullanılmayan Girişler

Bir TTL entegrenin sahip olduğu tüm girişlerin, yapılan uygulamada kullanılmasına ihtiyaç olmayabilir. Örneğin; (AB)´ lojik işlemini gerçekleştirmek için kullanılan üç girişli 'VEDEĞİL' kapısının alabileceği üç durum Şekil 7.22'de görülmektedir.



a) Kullanılmayan girişin açıkta bırakılması

b) Kullanılmayan girişin gerilim kaynağına bağlanması

c) Kullanılmayan girişin kullanılmayan diğer bir girişe bağlanması

Şekil 7.22. Entegrelerde kullanılmayan girişlerin alabilecekleri durumlar.

Birinci yöntem olarak, kullanılmayan giriş olduğu gibi açıkta bırakılabilir (Şekil 7.22.a). Bir girişin bağlanmadan açıkta bırakıldığı durum, **'Floating'** olarak isimlendirilir. Herhangi bir yere bağlanmadan açık olarak bırakılan uçlar TTL entegrelerinde lojik '1' olarak algılanır. Bunun anlamı; TTL entegrelerde '0'

veya ‘1’ uygulanmayan ve açıkta bırakılan girişlerin, lojik ‘1’ olarak kabul edilmesi gerektiğidir. Örneğin; Şekil 7.22.a’daki durumda girişte lojik ‘1’ algılanacağından, ‘VEDEĞİL’ kapısında (A.B.1)' = (A.B)' fonksiyonu oluşur ve istenen sonuç elde edilir.

Kullanılmayan girişin açıkta bırakılması istenen sonucu vermesine rağmen, kullanılmayan girişin bir anten gibi davranarak hatalı çıkışlar oluşturması mümkündür. Bu nedenle, ikinci yöntem olarak adlandırılacak kullanılmayan girişin +V’ye bağlanması yöntemi kullanılır. Kullanılmayan girişin bir direnç üzerinden +5 V’luk kaynağa bağlanması, girişin kesin ‘1’ olması sonucunu verir (Şekil 7.22.b). ‘VE’ kapıları için de kullanılabilecek bu yöntemde, akım sınırlayıcı olarak kullanılan dirence çok sayıda kullanılmayan giriş bağlanabilir. ‘VEYA’ ve ‘VEYADEĞİL’ kapıları için, kullanılmayan girişlerin açıkta bırakılması veya +5 V’a bağlanması uygun değildir. Bu tip kapılarda kullanılmayan girişler ya şaseye bağlanarak ‘0’ yapılmalı, ya da kullanılan girişlerden birine bağlanmalıdır.

Kullanılmayan girişler için uygulanan diğer bir yöntem; kullanılmayan girişin kullanılan girişlerden birisine bağlanmasıdır (Şekil 7.22.c). Bu yöntem tüm kapı çeşitlerine uygulanabilir. İki veya daha fazla girişin birbirine bağlanması ve tek bir giriş olarak kullanılması durumunda, oluşan girişin yükleme faktörü tüm girişlerin yükleme faktörleri toplamına eşit olur. Yalnızca ‘VE’ ve ‘VEDEĞİL’ kapılarının ‘0’ durumunda, kaç tane giriş bir araya bağlanırsa bağlansın tek bir giriş yükleme faktörü gibi davranır.

7.3.2. Metal Oksit Yarıiletken Elemanlarından Oluşturulan Lojik Teknolojisi

İsmini yarıiletkene kaplanan oksit yalıtkan üzerindeki elektrodun yapısından alır. MOS teknolojisi (Metal-Oxide-Semiconductor - MOS) ile üretilen transistörler, alan etkili transistörlerdir. Bu transistörler FET veya MOSFET olarak adlandırılır.

MOS dijital entegrelerin temel yapı elemanı olan MOSFET’lerin üstünlüğü, üretiminin basit ve ucuz olmasının yanında, küçük yer kaplaması ve güç harcamasının çok düşük olmasıdır. Entegre içerisinde bipolar transistörlerle 50 mm^2 yer kaplayan bir eleman, MOSFET ile 1 mm^2 yer kaplar. Yapımında çok yer kaplayan dirençlerin kullanılmadığı MOSFET entegrelerde, bir entegrenin içerisine bipolar transistörlerle yapılan entegrelere göre çok daha fazla sayıda devre elemanı yerleştirilebilir. Bu durum LSI ve VLSI tipi entegre sınıflarında MOSFET teknolojisi kullanılması sonucunu doğurur.

MOS entegrelerin dezavantajı düşük işlem hızıdır. Bu özelliğin gözardı edildiği bir çok uygulamada MOS lojik, bipolar lojige üstün bir alternatif oluşturur. MOS lojigi detaylandırmadan önce MOSFET'leri özetlemek konunun anlaşılmasına yardımcı olacaktır .

7.3.2.1. P-MOS ve N-MOS Teknolojileri

MOSFET'ler, çalışma şekli olarak azaltan (depletion) ve çoğaltan (enhancement) olmak üzere iki tipte imal edilirler. MOS entegrelerde genelde çoğaltan tip MOSFET'ler ON-OFF anahtar olarak kullanılır.

Diğer taraftan MOSFET'ler uygulanan polarma şekline göre, N-kanal ve P-kanal olmak üzere iki farklı yapıda üretilir. Yapılarında N-kanal ve P-kanal MOSFET kullanan MOS teknolojisi üç grup altında toplanabilir:

i- Yalnızca P-Kanal çoğaltan MOSFET'lerin kullanıldığı P-MOS lojik.

ii- Yalnızca N-Kanal çoğaltan MOSFET'ler kullanılan N-MOS lojik.

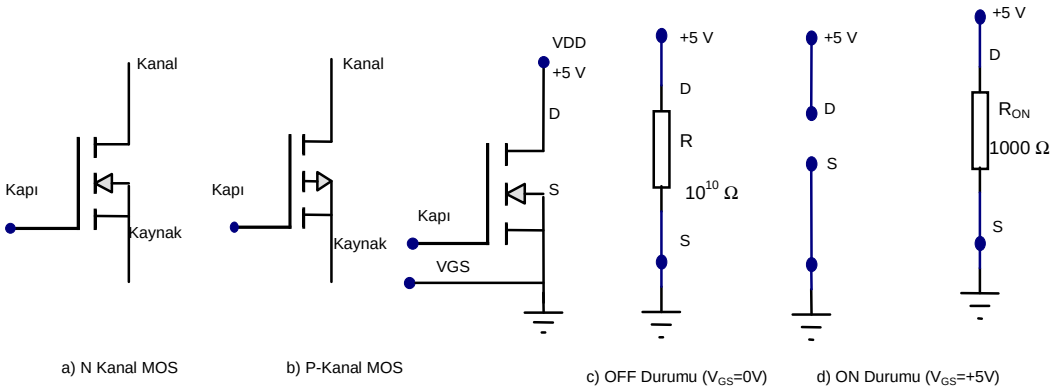
iii- P ve N kanal elemanların kullanıldığı CMOS lojik.

Şekil 7.23'de N-kanal ve P-kanal çoğaltan tip MOSFET'lerin sembolleri görülmektedir. Okların yönünün MOSFET'in tipini gösterdiği sembolde, kaynak (source) ve kanal (drain) arasındaki kesik çizgiler bu elemanlar arasında normalde bir bağlantı olmadığını gösterir. MOSFET'in üçüncü elemanı olan kapı (gate) ile diğer elemanlar arasında büyük bir direnç ($>10^{10}$ MΩ) mevcuttur.

N-kanal MOSFET'te kanal ucu daima (+) ile beslenir. Kapı ile kaynak arasına uygulanan gerilim (V_{GS}), giriş gerilimi olarak adlandırılır ve kanal ile kaynak arasındaki direnci kontrol etmek için kullanılır. Kontrol edilen direnç, MOSFET'in iletim - yalıtım durumunu belirler (Şekil 7.24).

V_{GS} = 0 V olduğu durumda eleman kesimdedir ve kaynak ile kanal arasında bir iletim yoktur. MOSFET'in kesimdeki direnci 10^{10} Ω civarındadır ve bu açık devre anlamına gelir.

V_{GS} gerilimi pozitif olduğu ve V_T ($V_{threshold}$) eşik gerilimine (V_T = +1.5V) ulaştığı zaman MOSFET iletime başlar. V_{GS} = +5V olduğu durumda ise, kaynak-kanal direnci R_{ON} = 1KΩ değerine düşer. Bu durumda N-MOSFET'in çok büyük bir dirençten, küçük bir direnç değerine düştüğü ve bir anahtar gibi çalıştığı görülür.



Şekil 7.23. Çoğaltılmış tip MOSFET'lerin sembolleri ve N Kanal MOSFET'in anahtarlama durumları.

N-MOS ve P-MOS tipi elemanlar polarma yönleri hariç aynı prensibe göre çalışlar. Bu tiplerden N-MOS elemanlar P-MOS elemanlara göre iki kat daha hızlıdır ve entegre içerisinde yaklaşık iki kat eleman yoğunluğuna sahiptir. Bu iki tip MOS elemanlar, LSI ve VLSI tipi uygulamalarda (mikroişlemci, hafızalar, ROM) tercih edilirler.

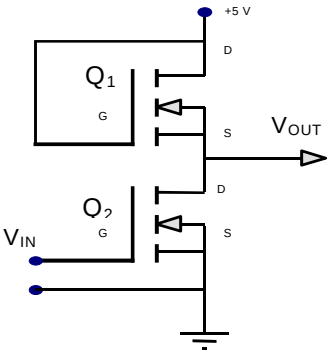
N-MOS 'DEĞİL' Kapısı

Şekil 7.24.a'da N-kanal MOSFET'lerle oluşturulan 'DEGİL' kapısı görülmektedir. Q_1'in yük MOSFET'i, Q_2'nin anahtarlama MOSFET'i olarak adlandırıldığı bu devrede, sabit olarak +5 V'a bağlı Q_1 daima 'ON' durumundadır ve yük direnci (R_{ON}) olarak çalışır. 'ON' ve 'OFF' durumları arasında durum değiştirerek çalışan Q_2 elemanının özellikleri Q_1'den farklıdır. Q_1 ve Q_2'nin 'ON' ve 'OFF' durumlarındaki özellikleri Şekil 7.24.b'de özetlenmiştir.

Şekil 7.24.a'daki devreyi analiz etmenin en iyi yolu; her bir MOS elemanını bir direnç olarak düşünüp, ortaya çıkan gerilim bölücünün durumuna göre oluşan çıkış gerilimlerini incelemektir.

V_{IN} = 0 durumunda; Q_2 kesimde olduğundan $R_{OFF} = 10^{10}\Omega$ ve Q_1 elemanı iletimde olduğundan R_{ON} = 100 KΩ değerlerine sahip olur. Bu durumda oluşan gerilim bölücüden yaklaşık +5V çıkış elde edilir.

V_{IN} = +5 durumunda; Q_2 iletime geçer ve R_{ON} = 1 KΩ değerine sahip olduğundan, çıkış V_{OUT} = 1*(+5) / 101 = 0.05 V değerini alır. DEĞİL kapısı olarak işlev gören bu devre, 'VEYADEĞİL' veya 'VEDEĞİL' kapısı olarak şekillendirilebilir.



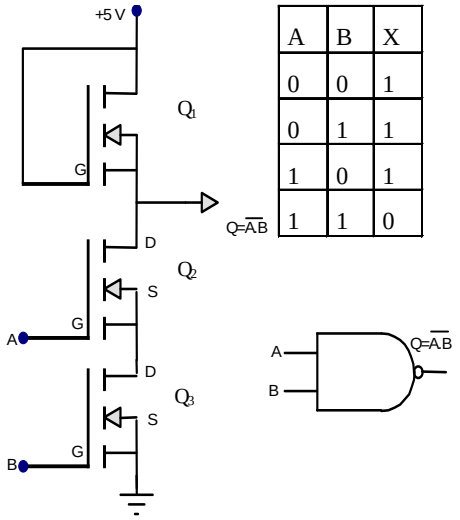
V_{IN}	Q_1	Q_2	V_{OUT}
0V Lojik "0"	R_{ON}=100 KΩ	$R_{OFF}=10^{10}$ Ω	+5 V Lojik "1"
+5 V Lojik "1"	R_{ON}=100 KΩ	R_{ON}=1 KΩ	+0.05 V Lojik "0"

(a) **(b)**

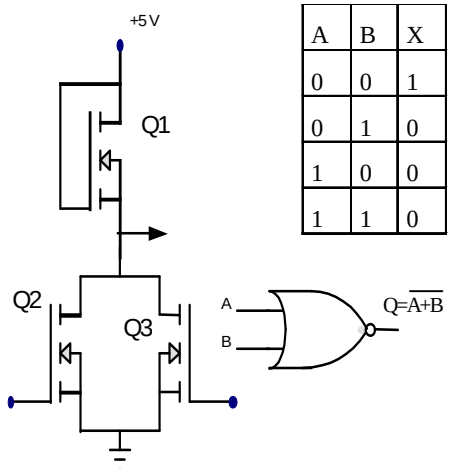
Şekil 7.24. N-MOS ‘DEĞİL’ kapısı ve özellikleri.

N-MOS ‘VEDEĞİL’ ve ‘VEYADEĞİL’ Kapıları

‘VEDEĞİL’ işleminde; Q_1 yine yük direnci olarak çalışırken, Q_2 ve Q_3 elemanları A ve B girişleri ile kontrol edilirler (Şekil 7.25.a). A veya B girişlerinden herhangi birisinin ‘0’ olması durumunda ilgili MOSFET yalıtımda olacağından, Q = 1 (+5V) olur. Her iki girişin ‘1’ olması durumunda ise, her iki MOSFET’de iletime geçer ve Q = 0 değerini alır. Anlatılanlardan devrenin ‘VEDEĞİL’ işlemi yaptığı görülmektedir.



A	B	X
0	0	1
0	1	1
1	0	1
1	1	0



A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(a) **(b)**

Şekil 7.25. N-MOS tekniği ile oluşturulan ‘VEDEĞİL’ ve ‘VEYADEĞİL’ kapıları.

N-MOS elemanları ile yapılan 'VEYADEĞİL' kapısında; Q_1 yine yük direnci olarak kullanılırken, Q_2 ve Q_3 elemanları paralel olarak bağlanır (Şekil 7.30.b). A ve B girişlerinden herhangi birisinin '1' olması durumunda ilgili MOSFET iletime geçer ve çıkış '0' olur. Her iki girişin '0' olması durumunda Q_2 ve Q_3 yalıtımda olduğundan, çıkış '1' olur. Bu durumda devrenin yaptığı iş, 'VEYADEĞİL' işlemine eşit olur.

P-MOS ve N-MOS Lojik Elemanlarının Özellikleri:

Bipolar lojik ailelerle karşılaştırıldığı zaman, daha düşük işlem hızı, daha az güç tüketimi, daha iyi gürültü bağışıklığı, daha geniş besleme gerilim sınırları, daha büyük 'Fan Out' gibi üstünlüklere sahip MOS lojik ailesinin özellikleri aşağıdaki şekilde özetlenebilir.

i- İşlem hızı : N-MOS tipi bir 'VEDEĞİL' kapısındaki yayılım gecikmesi 50 ns'dir. Bunun iki nedeni; çıkışın '1' olduğu durumlarda yüksek çıkış empedansı (100 KΩ) ve sürülen lojik kapı devresinden dolayı oluşan kapasitif yükleme etkisidir. MOS lojik elemanları, girişleri yüksek giriş direncine ($>10^{12}$ Ω) ve kabul edilebilir bir giriş kapasitansına (MOS kapasitör) sahiptirler. Büyük değerli R_{OUT} ve büyük değerli $C_{yük}$'un birleşmesi artan anahtarlama zamanı sonucunu doğurur.

ii- Gürültü sınırı : N-MOS gürültü sınırı, V_{DD}= 5 V durumunda 1.5 V civarındadır ve daha yüksek V_{DD} değerleri için orantılı olarak büyür.

iii- Fan-Out : Her bir MOSFET girişindeki çok yüksek giriş direnci nedeniyle, MOS lojikteki elemanların sınırsız 'Fan Out' kapasitesine sahip olduğu düşünülebilir. Bu düşünce DC ve düşük frekanstaki uygulamalar için doğrudur. Bununla beraber, 100 KHz'ten daha büyük frekanslarda oluşan kapı giriş kapasiteleri nedeniyle 50 civarındaki bir Fan Out'ta randımanlı çalışır.

iv- Güç harcaması : Kullanılan büyük dirençler nedeniyle MOS lojik devreler düşük güç harcamasına sahiptir. Örneğin bir 'DEĞİL' kapısı için harcanan güç 0.1 mW civarındadır.

v- İşlem yoğunluğu : MOS lojik yapısında tek çeşit eleman kullanıldığından üretimi en kolay lojik ailedir. MOS lojikte direnç veya diyot gibi elemanlar kullanılmaz. Düşük güç tüketimi ile bu özellik birleşince, MOS elemanları LSI devreler için (geniş hafızalar, hesap makinesi, mikroişlemciler) ideal duruma gelir.

vi- Statik hassasiyet ve kullanılmayan girişler : MOS elemanlar insanların üzerinde bulunan statik elektriğe karşı çok hassastırlar. Çünkü çok yüksek giriş direnci nedeni ile, insanlar üzerinden gelen deşarj çok büyük bir

gerilim düşümüne neden olur ve bu gerilim devreyi tahrip edebilir. Bu nedenle MOS entegreler ile çalışılırken;

1. Çıplak el ile entegreye dokunulmamalı,
2. Lehimlemede topraklı havya kullanılmalı,
3. Kullanılmayan entegre ayakları boşluğa bırakılmamalıdır (+ V'ye veya şaseye bağlanmalıdır).

7.3.2.2. Tamamlayıcı MOS Lojik (CMOS Teknolojisi)

Tamamlayıcı MOS Complementary MOS Logic (CMOS) lojik ailesinde P ve N kanal MOSFET'ler birlikte kullanılır. CMOS ailesi, P-MOS ve N-MOS kanalın sahip olduğu üstünlükleri aynı devrede toplar. CMOS entegreler, P ve N tipi MOS'lara göre çok daha karmaşık bir yapıya ve daha düşük eleman yoğunluğuna sahiptirler.

CMOS entegreler, PMOS ve NMOS teknolojilerine göre daha yüksek hız ve daha düşük güç tüketimi gibi avantajlara sahiptirler. LSI alanı içerisinde pek fazla kullanılmayan CMOS ailesi, MSI alanında TTL'nin rakibi olarak kullanılmaktadır. TTL'ye göre daha basit üretim işlemine ve daha yüksek paketleme yoğunluğuna sahip CMOS ailesi, bu özellikleri nedeni ile bir entegre içerisinde daha fazla sayıda eleman oluşturmasına imkan tanır. Genelde TTL'lere göre daha yavaş olan CMOS entegrelerin yeni serilerinde yüksek hızlı elemanlar üretilmektedir.

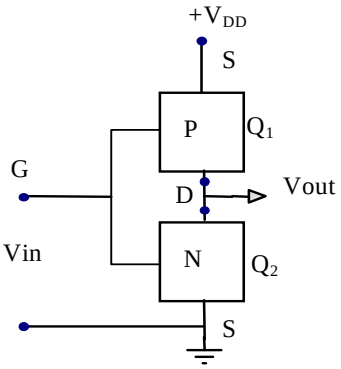
CMOS 'DEĞİL' Kapısı

CMOS DEĞİL kapısı, N-MOSFET ve P-MOSFET kullanılarak Şekil 7.26'deki gibi gerçekleştirilir. İki MOSFET'e sahip olan CMOS DEĞİL kapısında, P-MOS'un kaynak (S) ucu $+V_{DD}$'ye, N-MOS'un kaynak (S) ucu ise şaseye bağlanmıştır. Her iki MOS'un kapı (G) girişleri birlikte bağlanarak tek bir giriş olarak kullanılır. Aynı şekilde, her iki MOS'un kanalları (D) ortak bağlanarak çıkış olarak kullanılmaktadır.

CMOS'da $+V_{DD}$ değeri lojik '1' değerini temsil ederken, 0 V değeri lojik '0' değerini temsil eder. $V_{IN} = +V_{DD}$ olduğu durumda Q_1 kesimde olacağından, $R_{OFF} = 10^{10}$ Ω'luk bir dirence sahiptir. Bu anda Q_2 'nin kapı girişinde $+V_{DD}$ olduğundan, Q_2 iletimde ve $R_{ON} = 1$ KΩ değerine sahip olur. Q_1'in R_{OFF} değeri ile Q_2'nin R_{ON} değeri gerilim bölücü olarak düşünülürse; $V_{OUT} = 0$ değeri elde edilir.

$V_{IN} = 0$ durumda ise; G terminalinde S'ye göre daha negatif gerilim bulunan Q_1 iletimde olurken, Q_2 ters polarmalanarak kesimde kalır. Bu durumda çıkış,

yaklaşık $V_{OUT} = +V_{DD}$ değerini alır. Özetlenen bu işlem lojik ‘DEĞİL’ işlemine karşılık gelir.

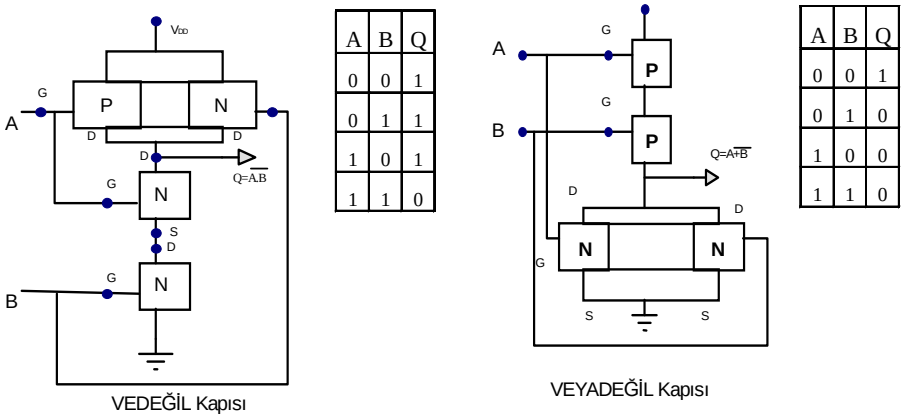


V_{IN}	Q_1	Q_2	V_{OUT}
"1" $+V_{DD}$	OFF $R_{OFF}=10^{10}\Omega$	ON R_{ON}= 1K	$\cong$ 0 V
"0" 0 V	ON R_{ON}= 1K	OFF $R_{OFF}=10^{10}\Omega$	$\cong +V_{DD}$

Şekil 7.26. CMOS ‘DEĞİL’ kapısının N-tipi ve P-tipi MOSFET kullanılarak gerçekleştirilmesi.

CMOS ‘VEDEĞİL’ ve ‘VEYADEĞİL’ Kapıları

Temel ‘DEĞİL’ kapısını şekillendirerek oluşturulacak diğer bir kapı devresi, açık şeması Şekil 7.27.a’da gösterilen ‘VEDEĞİL’ kapısıdır. Bu kapı devresinde yalnızca her iki girişin ‘1’ olması durumunda çıkış ‘0’ olur. A ve B girişlerinin her ikisinin veya herhangi birisinin ‘0’ olması durumunda ilgili N-MOS kesimde olacağından, çıkışta ‘1’ gözükür. Sonuçta ‘VEDEĞİL’ işlemi gerçekleşir.



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

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

(a) **(b)**

Şekil 7.27. CMOS tekniği ile oluşturulan ‘VEDEĞİL’ ve ‘VEYADEĞİL’ kapıları.

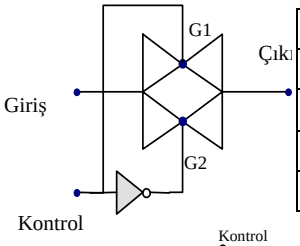
CMOS ‘VEYADEĞİL’ kapısı, seri bağlı P-MOS elemanlara paralel bağlı N-MOS elemanlar bağlanmak suretiyle elde edilir (Şekil 7.27.b). Girişlerin her ikisinin ‘0’ olması durumunda, paralel bağlı N-MOS elemanlar yalıtımda olacağından çıkış ‘1’ olur. Girişlerden herhangi birisinin ‘1’ olması durumunda ise, ilgili N-MOS iletimde olacağından çıkış şase potansiyeline yaklaşır ve ‘0’ olur.

CMOS İki Yönlü Anahtar (Bileteral Switch) / İletişim Kapısı (Transmision Gate):

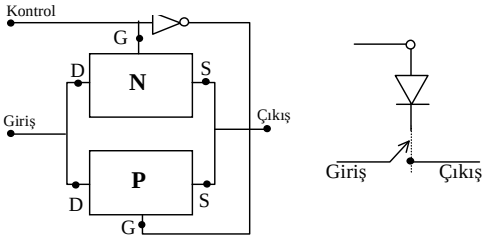
İletişim kapısı; TTL ve diğer bipolar lojik ailelerde olmayan, giriş lojik seviyesi ile kontrol edilen, işareti her iki yönde de iletilebilen bir kapıdır. Bu kapı hem analog hem de dijital devrelerde kullanılır.

Şekil 7.28’de sembolü ve açık şeması görülen iletişim kapısında, N ve P MOSFET elemanları parelel olarak bağlanmıştır. Bu özellik nedeniyle, iletişim kapısı girişine uygulanan pozitif ve negatif gerilimleri iletebilir.

Kapıdaki kontrol girişi, anahtarın kapalı veya açık olmasını sağlar. Kontrol girişinin ‘1’ olması durumunda, her iki MOSFET’te iletime geçeceğinde anahtar kapalı (ON) duruma gelir. Kontrol girişinin ‘0’ olması durumunda, her iki MOSFET’te kesimde olacağından anahtar açıktır (OFF). İdeal olarak elektromagnetik röle gibi çalışan iletişim kapısı pratikte kapalı olduğu zaman R_{ON} = 100Ω-1000Ω, açık olduğu zaman $R_{OFF} = 10^{12}$ Ω değerini alır.



Giriş	Kontrol	G_1	G_2	Çıkış	
1	1	1	0	1 Giriş	Düşük Empedans
0	1	1	0	0 Çıkış	
1	0	0	1	Çıkış yok	Yüksek Empedans
0	0	0	1	Çıkış yok	

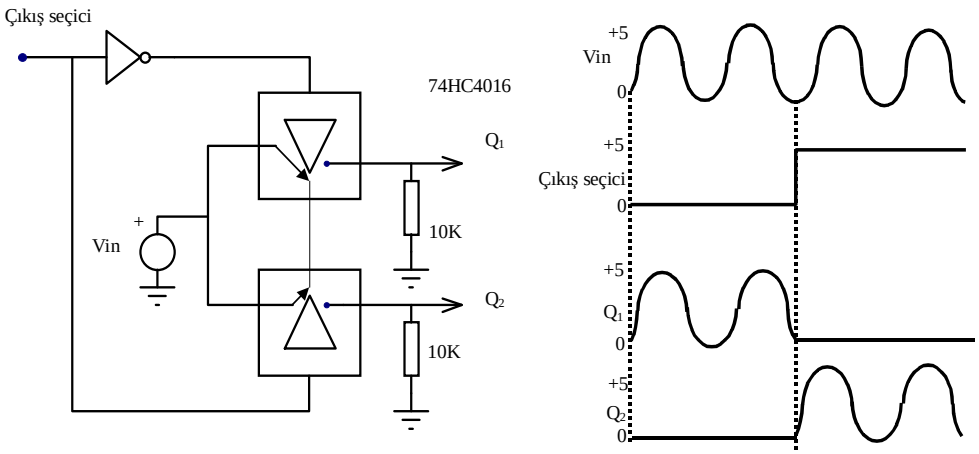


Şekil 7.28. İki yönlü kapı sembolü, fonksiyon tablosu ve açık şeması.

Elemanın iki yönlü anahtar olarak isimlendirilmesinin nedeni, giriş ve çıkış terminallerinin yer değiştirebilmesidir. Girişe uygulanan sinyal, $+V_{DD}$ ve 0 V sınırları içerisinde kalması şartıyla analog veya dijital bir sinyal olabilir. 4016 ve 4066 CMOS entegreleri (veya 74HC4016), yukarıda açıklandığı şekilde dört adet iletişim kapısı içerir. Bu kapılar birbirinden bağımsız kullanılabilirler.

İletişim kapısının kullanımına örnek olması ve bu elemanın analog sinyalleri anahtarlamasını açıklaması açısından aşağıdaki örnek işlemi gerçekleştirelim.

Örnek 9 : Bir analog sinyalin uygulanacağı yeri, bir lojik sinyalin seviyesi ile kontrol etmek istiyoruz. Lojik sinyalin '0' olması durumunda analog sinyal 1 nolu devreye, lojik sinyalin '1' olması durumunda analog sinyal 2 nolu devreye uygulanacaktır. Bu işlemi iletişim kapıları kullanarak gerçekleştirelim.



(a) **(b)**

Şekil 7.29. Analog sinyalin iki farklı kopyasının çıkarılması ve çıkış dalga şekilleri.

Çözüm: Bu durumda iki farklı çıkış veren parelel iki iletişim kapısı kullanmamız gerekir. Anolog sinyal bu kapıların girişlerine aynı anda uygulanır. Paralel bağlı kapılardan hangisinin iletimde olacağına karar veren bir kontrol devresi gerekmektedir. Lojik kontrol sinyalini bir DEĞİL kapısından geçirerek bu işlemi gerçekleştirebiliriz. Çıkışlara yük dirençleri bağlanması ile Şekil 7.29.a'daki devre elde edilir.

Elde edilen devrede giriş sinyali 0-(+5) V sınırları içerisinde olursa, Şekil 7.29.b'deki sinyaller elde edilir. Analog sinyalin (+) ve (-) kısımlarının olmasını istiyorsak, bunun için uygun olan devre elemanlarını kullanmamız gerekir.

4316 ve 74HC4316 entegreleri iki yönlü analog sinyalleri anahtarlayabilirler. Ancak ikinci bir gerilim kaynağına ihtiyaç duyarlar. Çıkış ise (–5V) ile (+5V) oranında değişebilir.

İletişim kapıları çok değişik alanlarda kullanılmaktadır. Bunlardan birkaçı; analog sinyal anahtarlama / seçme, kırpıcı, modülatör / demodülatör, dijital sinyal anahtarlama / seçme, analog - dijital ve dijital - analog çeviriciler şeklinde sıralanabilir.

CMOS Entegrelerin Özellikleri

i- CMOS entegrelerde kullanılan seriler :

CMOS ailesindeki dijital entegreler çeşitli serilerde üretilmektedir. İlk üretilen CMOS'lar 4000A serisinde iken, geliştirilmiş versiyonu olan CMOS'lar daha yüksek çıkış akımı ile 4000B serisi olarak üretilmektedirler. 74C ve 54C serisi olarak üretilen CMOS serisi entegreler, TTL entegresi ile fonksiyon ve pin olarak tam uyumludur. TTL eşdeğerleri yerine kullanılabilen 74C serisinin performansı, 4000 serisi ile aynıdır. Motorola firması 4000 serisini 14000 serisi olarak üretmektedir. Yani, 4004 ile 14004 fonksiyon ve ayak uyumludur. Sygnetics firması, aynı seriyi HCT4001 şeklinde tanımlamaktadır.

74C serisinin geliştirilmiş şekli, daha hızlı olarak imal edilen 74LS serisi TTL'lerin karşılığı olan 74HC serisi (yüksek hızlı CMOS) entegrelerdir. Bu tipin daha geliştirilmiş serisi ise, doğrudan TTL çıkışlar tarafından sürelebilen 74HCT serisidir. Farklı lojik ailelere mensup entegrelerin birlikte kullanılma durumu daha sonraki konularda incelenecektir.

ii- Besleme Gerilimi ve Gerilim Seviyeleri :

4000 ve 74C serisi entegreler; 3-15 Volt arasındaki gerilimler ile, 74HC ve 74HCT serileri ise 2-6 Volt arasındaki gerilimler ile çalışırlar. CMOS ve TTL entegreler birlikte kullanıldığı zaman, CMOS'un V_{DD} ve TTL'in V_{CC} gerilimlerini aynı kaynaktan sağlamak için her iki tip eleman +5 V ile çalıştırılabilir.

CMOS elemanların yalnızca CMOS elemanları sürme durumunda çıkış gerilim seviyesi; '0' konumunda yaklaşık 0 V, '1' konumunda ise $+V_{DD}$ değerinde olur. CMOS entegreler V_{DD} = +5V ile çalıştırıldığı zaman, $V_{IL(MAX)}$ = 1.5 V ve $V_{IH(MİN)}$ = 3.5 V olarak kabul edilir.

iii- Gürültü Sınırları :

CMOS entegrelerin DC gürültü bağışıklıkları; $V_{NH} = V_{NL} = \%30\ V_{DD}$ olarak ifade edilir. V_{DD} = +5 V ise, her iki lojik durumundaki gürültü sınırı 1.5 V olarak bulunur. Bu değer TTL ve ECL'den daha iyidir ve bu durum yüksek gürültü ortamındaki uygulamalarda CMOS'u üstün duruma geçirir.

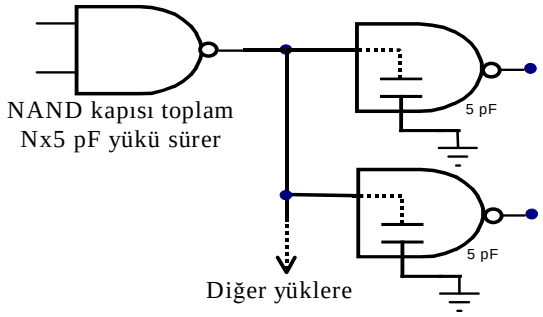
iv- Güç Harcaması :

CMOS lojik devreler statik durumda iken (durum değiştirmediği zaman) güç harcaması çok düşüktür. Bunun nedeni; Şekil 7.27'deki 'VEYADEĞİL' ve 'VEDEĞİL' kapılarının açık şemalarından görüleceği üzere, + V_{DD} ile şase arasındaki akım yolu üzerinde kesim durumunda bir MOSFET in olması ve buna bağlı olarak yüksek direncin bulunmasıdır. Bu durum, V_{DD} = +5 V olduğu zaman her bir kapı için 2.5 nW'lık bir güç harcamasını ortaya çıkarır. V_{DD}= +10 V durumunda dahi 10 nW'lık bir güç harcaması oluşur.

DC çalışmada düşük seviyede kalan CMOS entegrelerin güç tüketimi, anahtarlama durumu söz konusu olduğu durumlarda yükselir. Örneğin; DC çalışmada 10 mW olan güç tüketimi; 100 KHz'de P_D = 0.1 mW ve 1 MHz'te P_D= 1mW değerini alır. Bunun nedeni, çıkışa bağlı yük nedeni ile oluşan kapasitenin üzerinden akan şarj akımıdır.

v- Fan - Out :

N-MOS ve P-MOS'larda olduğu gibi, CMOS girişleri de çok büyük bir dirence sahiptirler ($10^{10}\Omega$) ve sinyal kaynağından bir akım çekmezler. Ancak her bir CMOS giriş ortalama 5 pF'lık yük etkisi oluşturur (Şekil 7.30). Bu giriş kapasitesi, bir CMOS elemanın sürebileceği giriş sayısını sınırlar. Bir CMOS elemanın çıkışı, bağlı bulunan tüm girişlerdeki kapasitelerin paralel kombinasyonunu şarj / deşarj etmek zorunda olduğundan, sürülen elemanların sayısının artması çıkış anahtarlama zamanını artırır.



Şekil 7.30. CMOS'a bağlı yüklerin kapasite etkisi oluşturması.

CMOS elemanlarda eklenen her bir yük ile, sürücü devrenin yayılım gecikmesi ortalama 3 nsn artar. Yüksüz durumda t_{PHL}=30 nsn yayılım gecikmesine sahip bir 'VEDEĞİL' kapısının, 20 yüke sahip olması durumundaki yayılım gecikmesi:

Toplam yayılım gecikmesi = $t_{PHL(yüksüz)}$ + (yük sayısı) * 3

= 30 nsn + 20 (3 nsn) = 90 nsn

olur.

Bu durum Fan-Out'un maksimum yayılım gecikmesine bağlı olduğu sonucunu doğurur. Genel olarak 1MHz'in altındaki uygulamalarda 'Fan Out' değeri 50 sayısı ile sınırlandırılır.

vi- Anahtarlama Hızı :

Her ne kadar CMOS büyük kapasitif yükleri sürmek zorunda olsada, düşük çıkış direnci (P-MOSFET'in '1' durumundaki R_{ON} direnci $\leq$1 KΩ dur) yük kapasitansının hızlı şarj olmasını sağlar. Bu durum anahtarlama hızını artırır.

4000 serisi 'VEDEĞİL' kapısı; V_{DD}= +5 V'ta ortalama t_{Pd} = 50 nsn,V_{DD} = +10 V'ta t_{Pd} = 25 nsn yayılım gecikmesine sahiptir. V_{DD}'nin artmasıyla t_{Pd} değerinin artmasının nedeni; yüksek besleme geriliminde MOSFET'in R_{ON} değerinin küçülmesidir. Bu durum, yüksek hız gerektiren yerlerde V_{DD} geriliminin mümkün olduğu kadar yüksek tutulması gerektiğini ortaya çıkarsa da, V_{DD}'nin artması ile güç tüketiminin artacağı ortadadır.

74HC veya 74 HCL serisindeki CMOS 'VEDEĞİL' kapısı; V_{DD} = 5 V iken, ortalama t_{PD} = 8 nsn yayılım gecikmesine sahiptir ve bu hız 74 LS serisine karşılık gelir.

vii- Kullanılmayan Girişler ve Statik Şarj Hassasiyeti:

CMOS girişleri kesinlikle açık bırakılmamalı, sabit kaynağın terminallerinden birisine (şase veya +V_{DD}) veya diğer bir girişe bağlanmalıdır. Bağlanmayan bir giriş, gürültü ve statik şarjlara karşı duyarlıdır. Statik şarj nedeniyle yüksek giriş dirençlerinde düşen gerilim, MOSFET'lerin kanal (S) ile kapısı (G) arasındaki yalıtımın bozulmasına neden olabilir. Ayrıca, P ve N kanal MOSFET'lerin her ikisini iletimde tutarak güç tüketiminin artmasına ve aşırı ısınmaya neden olur.

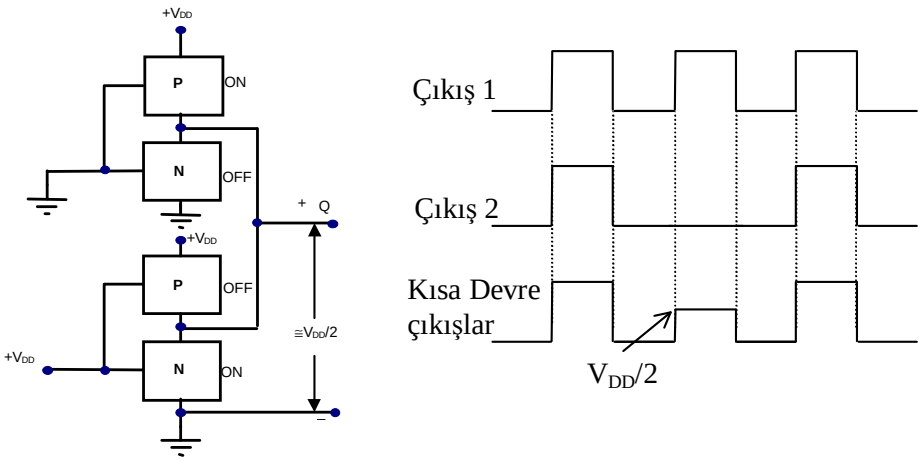
Yeni üretilen CMOS elemanların çoğu, herbir girişe zener koruma diyodu eklenmek suretiyle statik şarjın neden olduğu bozulmalara karşı korunmaktadır. Bu diyotlar tehlikeli gerilim sınırında iletime geçerek girişin yalıtım bölgesinin bozulmasına engel olurlar. Ancak zenerin bazı durumlarda uygun değerde ve zamanda iletime geçmeyeceği düşünülerek, yeni CMOS'lar içinde daha önce

açıklanan koruma tedbirleri uygulanmaktadır. Ayrıca, entegre doğrudan devreye lehimlenme yerine soketlerle devreye bağlanmalıdır.

viii- CMOS Açık Kanal (Open Drain) ve Üçdurumlu (Tristate) Çıkışlar :

Normal CMOS çıkışları, kesinlikle birlikte bağlanmamalıdır. Şekil 7.31'da iki 'DEĞİL' kapısının çıkışının birlikte bağlantısı görülmektedir. Çıkışları birlikte bağlanan kapılardan üstteki 'DEĞİL' kapısında P-MOS iletime geçer ve R_{ON} = 1 KΩ değerini alır. Alttaki 'DEĞİL' kapısında N kanal MOSFET iletime geçer ve R_{ON} = 1 KΩ değerine sahip olur. Ortaya çıkan 1 KΩ' luk dirençler gerilim bölücü olarak çalışacaklarından, Q çıkışı yaklaşık V_{DD} / 2 değerini alır.

Q çıkışında oluşan gerilim, tanımlanmayan bir değere sahip olduğundan ($V_{IL(max)}$ = %30V_{DD} ve $V_{IH(mın)}$ = %70V_{DD}), diğer elemanları sürmek için kullanılamaz. Ayrıca iletim durumundaki iki MOSFET'ten akan akım, normal şartlarda akacak akımdan çok daha büyük değerde olacağından entegre bozulabilir. Birlikte bağlanan çıkışların farklı konumlara sahip olması durumunda, çıkışta oluşacak sinyal Şekli 7.31.b'de görülmektedir.

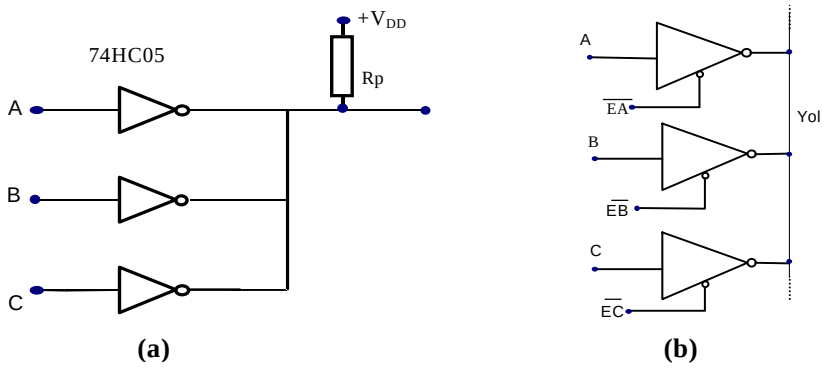


Şekil 7.31. Birlikte bağlanan CMOS çıkışlar ve çıkışların farklı değerleri alması durumunda oluşacak dalga şekli.

İki kapının çıkışının birlikte bağlanması durumunda ortaya çıkan sakıncayı ortadan kaldırmak amacıyla, bazı CMOS elemanlar TTL açık-kollektör çıkışa sahip elemanların eşdeğeri olacak şekilde açık-kanal (open-drain) yapısında üretilirler. Bu elemanlarda P-kanal MOSFET kaldırıldığından, çıkış başka bir

noktaya bağlanmayan N-kanal MOSFET'ten alınır. Böyle bir bağlantıda '1'seviyeli çıkış sinyali üretmek için, harici Pull-up direnci kullanılır. Açık-kanal yapıdaki elemanlar 'Bağlantılı-VE' bağlanabilirler (Şekil 7.32.a).

Bazı CMOS entegrelerin çıkışları ise, TTL üçdurum yapısına benzer bir yapıya sahiptirler. Böyle bir bağlantıda TTL'ler için söylenen herşey CMOS'lar içinde geçerlidir. CMOS üçdurumlu çıkışlara sahip CMOS elemanların çıkışları, Şekil 7.32.b'deki gibi bir yol (bus) üzerinde birlikte bağlanabilir.



Şekil 7.32. CMOS açık kanal 'bağlantılı-VE' bağlantı ve CMOS üç durumlu çıkışların yol (bus) bağlantısı.

7.4. Lojik Elemanlarda Ara Uyum

Devre tasarımlarında, farklı özelliklere sahip elemanların sistemin farklı yerlerinde kullanılması gerekebilir. Böyle durumlarda arabirim elemanlarına ihtiyaç duyulur. '**Arabirim (interface) devresi** '; sürücü devrenin çıkışı ile yük olarak bağlanan devrenin girişini uygunlaştırmak amacıyla kullanılan bir devredir. Arabirim devreleri çoğu zaman farklı özelliklere sahip dijital sistemleri bir arada kullanmayı sağlayan devrelerdir.

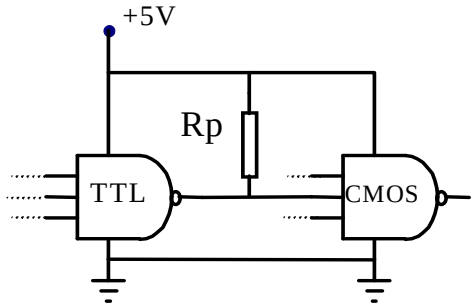
Aynı seri içerisinde bulunan entegreler, 'Fan Out' sınırlaması göz önünde bulundurularak, başka bir özelliğe bakılmaksızın bir arada kullanılırlar. Farklı lojik ailelerden veya aynı lojik ailenin farklı serilerinden bir arada kullanılan entegrelerde ise, birlikte kullanılan elemanların gerilim ve akım değerleriyle birlikte diğer parametrelerin göz önünde bulundurulması gerekir.

CMOS ve TTL entegreler birlikte kullanıldıklarında, bu entegrelere ait özellikler göz önünde bulundurulmalıdır. Tablo 7.1, CMOS ve TTL entegrelerin giriş ve çıkış gerilim / akım değerlerini özetlemektedir. Bu değerlerle birlikte

çalışma gerilimleri ve giriş/çıkış empedansları farklılık gösterir. Farklı parametrelere sahip elemanları bir arada kullanırken, sürücü olarak kullanılan elamanın çıkışının, yük olarak kullanılan elemanın akım ve gerilim gereksinimlerini karşılayabilmesi gerekir. Bu durumda, TTL ve CMOS elemanların birbirlerini sürmesi durumlarını detaylandıralım.

7.4.1. TTL Devrelerin CMOS Devreleri Sürmesi

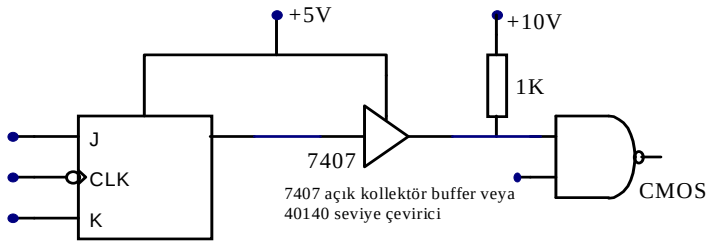
Yük olarak kullanılan CMOS entegrelerin giriş akım değerlerinin, sürücü olarak kullanılan TTL ailesinin çıkış akımlarına göre çok düşük olması nedeniyle herhangi bir problem oluşmaz. Bununla beraber, TTL çıkış gerilimleri ile CMOS giriş gerilim seviyelerini karşılaştırdığımızda, TTL serilerindeki $V_{OH(min)}$ değerinin CMOS ailesinden 4000B ve 74HC serilerinin $V_{IH(min)}$ seviyelerine göre çok düşük olduğu görülür. Bu problemi çözmek için, TTL çıkış gerilim seviyesinin CMOS için kabul edilebilir bir sınıra yükseltilmesi gerekir. Bu durumda kullanılan en yaygın çözüm; TTL çıkışı ile +5V kaynak arasına bir pull-up direnci bağlamaktır. Pull-up direncinin bağlanması ile, TTL çıkışı '1' konumunda +5V'a yükselir. Pull-up direnci olarak Rp = 1KΩ~10KΩ değerinde bir direnç kullanılır (Şekil 7.33).



Şekil 7.33. TTL'nin CMOS'u sürmesi için harici pull-up direnci bağlantısı.

CMOS elemanların V_{DD} = +5V'tan büyük gerilim ile çalıştırılması durumunda, daha büyük arauyum problemi ile karşılaşılır. Örneğin; V_{DD} = 10V olması durumunda CMOS eleman girişinin $V_{IH(min)}$ = 7V olması gerekir. TTL elemanlar +5V'tan büyük gerilim ile çalışmayacağından, pull-up direnci kullanılamaz. V_{DD}'ye pull-up direnci bağlanamayınca diğer alternatifler ortaya çıkar. Bu alternatiflerden birisi; TTL yukarı çekişli (totem-pole) çıkışı ile +5V'tan büyük gerilimle ile çalışan CMOS arasında 7407 açık-kollektör tampon (Şekil 7.34) veya 40104 gibi seviye çevirici (level-translator) devresi kullanmaktır. Bu

CMOS entegresi, düşük gerilimi (TTL'den aldığı) yüksek gerilim seviyesine dönüştürür.

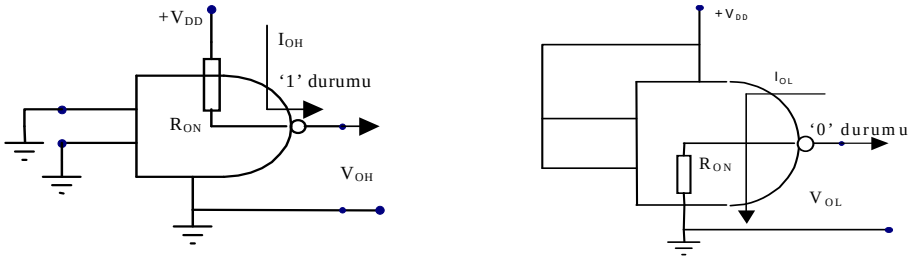


Şekil 7.34. TTL yüksek gerilimli CMOS bağlantısı için tampon kullanılması.

7.4.2. CMOS Entegrelerin TTL Entegreleri Sürmesi

CMOS entegrelerin TTL elemanları sürme durumunda oluşacak olayları incelemeden önce, CMOS elemanların çıkışlarının '1' veya '0' olması durumunda sahip oldukları karakteristikleri hatırlayalım.

Çıkışın '1' olması durumunda; bir ucu + kaynağa bağlı R_{ON} direnci, çıkışı $+V_{DD}$ kaynağına bağlar (Şekil 7.35.a). Bu durumda çıkışta R_{ON} direnci üzerinden $+V_{DD}$ gerilimi görülür. R_{ON} direncinin değeri 100Ω-1000Ω arasında değişir.



(a) **(b)**

Şekil 7.35. CMOS elemanların farklı çıkış değerlerinde alacağı durumlar.

Çıkışın '0' olması halinde, bir ucu şaseye bağlı R_{ON} direnci, çıkışı şase potansiyeline getirir (Şekil 7.35.b). Bu durumunda devre akım yutucu olarak çalışır.

Hatırlatmalar ışığı altında, CMOS elemanın çıkışın '1' olması durumunda (V_{OH}), TTL elemanların girişleri için gerekli gerilim seviyesini sağlayacağı

görülür. Diğer taraftan CMOS çıkış akımı (I_{OH}), TTL elemanların giriş akımlarını (I_{IH}) sağlar.

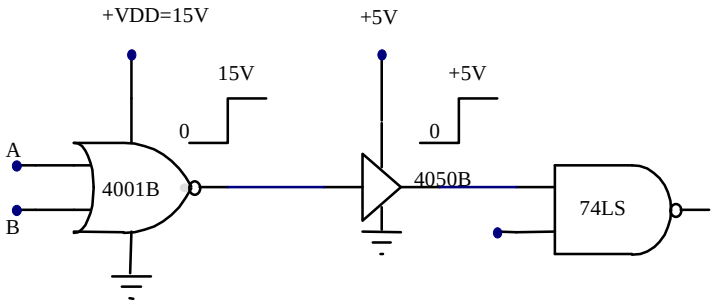
Çıkışın '0' olması durumunda, TTL elemanlar 100µA ile 2mA arasında bir akım çekerler. 74HC ve 74HCT serileri 4mA'lik akım yutma kapasitesine sahip olduklarından, bu seriler tek TTL yükleri sürebilirler. 4000B serisi ise, 74 veya 74AS serisinden bir elemanı dahi sürmek için yeterli değildir.

CMOS elemanların sürebileceği eleman sayısı;

$$\text{Fan-Out} = I_{OL(max)} / I_{IL(max)}$$

formülü ile hesaplanabilir. Hesaplama sonucunda CMOS elemanın çıkışına bağlı elemanları süremeyeceği ortaya çıkarsa, arabirim elemanları kullanılır. Örneğin; 4050B ve 74LS125 elemanları bu amaç için kullanılan arabirim elemanlarıdır.

CMOS'ların TTL'leri sürmesi durumunda karşılaşılan diğer bir problem, elemanların farklı gerilimlerle çalışmasıdır. Örneğin; +15V ile çalışan bir CMOS elemanı, +5V V_{CC} gerilimi ile çalışan TTL elemanını süremez. Bu durumda iki eleman arasında gerilim dönüşümü yapan bir arabirim elemanına ihtiyaç vardır. 4050B tampon elemanı bu işlemi gerçekleştirebilir (Şekil 7.36). Bu eleman CMOS'tan gelen sinyalin seviyesi ne olursa olsun, çıkışında TTL için uygun olan +5V seviyesinde bir gerilim üretir.

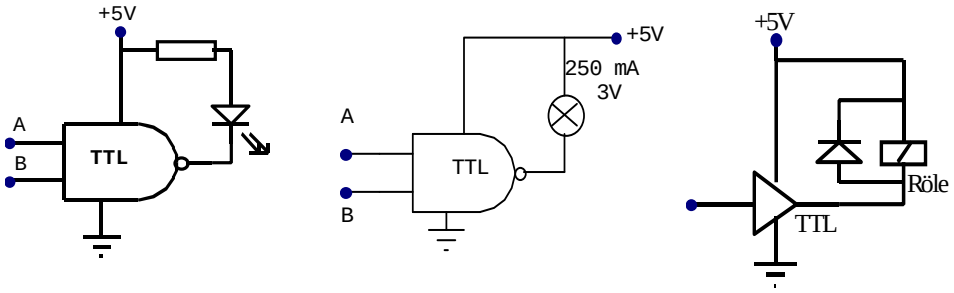


Şekil 7.36. 4050B entegresinin gerilim seviye çevirici olarak kullanılması.

7.4.3. TTL ve CMOS Entegrelere Yük Bağlantıları

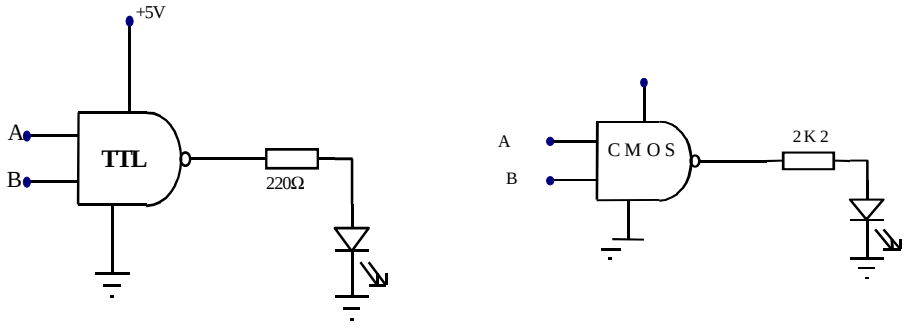
TTL veya CMOS elemanların çıkışlarına, entegre devre özelliklerine uymak şartıyla çok basit ve değişik yöntemlerle yük bağlantısı yapılabilir. Diğer bir deyişle, yük bağlantı şekli bağlanacak yükün cinsine ve entegrenin çıkış bağlantı şekline göre değişiklik gösterir.

'Açık-kollektör' çıkışa sahip entegrelerin çıkışlarına Pull-up direnci bağlanması gerektiğinden, pull-up direnci olarak çalışabilecek Led, Lamba, Röle gibi elemanlar doğrudan çıkışa bağlanabilir (Şekil 7.37). Entegrenin çıkışının sürülecek yük için yeterli olmaması durumunda, pull-up direnci ve bir güç transistörü ile yük sürülebilir.



(a) LED'in sürülmesi. (b) Lambanın sürülmesi. (c) Rölenin sürülmesi.

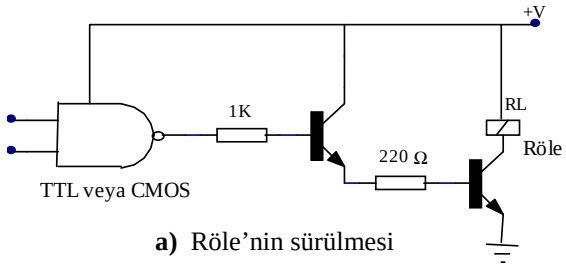
Şekil 7.37. Açık-Kollektör TTL'lere yük bağlantı şekilleri.



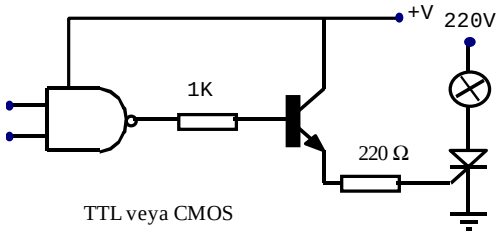
Şekil 7.38. Totem-pole elemanlara yük bağlantı şekilleri.

Totem-pole çıkışlı TTL ve CMOS elemanlar, düşük akımlı yükleri doğrudan sürebilirler. Her iki lojik ailede bağlantı şekli aynıdır, yalnızca akım sınırlayıcı direncin değeri değişir. Şekil 7.38de totem-pole çıkışlı entegrelere yüklerin doğrudan bağlantısı görülmektedir.

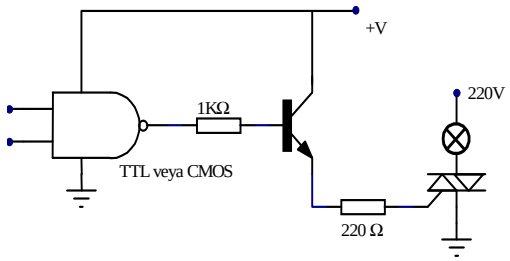
TTL veya CMOS elemanların tristör; triyak, röle, vb., yüksek güçlü sistemleri kontrol eden elemanları sürmesi mümkündür. Bu durumda Şekil 7.39'deki bağlantı prensiplerine dikkat edilmelidir.



a) Röle'nin sürülmesi



b)Tristörün sürülmesi



c) Triyak'ın sürülmesi

Şekil 7.39. Lojik entegreler ile güç kontrol elemanlarının sürülmesi.

7.5. TTL ve CMOS Elemanlarla Kullanılan Göstergeler

TTL ve CMOS elemanlar kullanılarak yapılan lojik ve aritmetik işlemlerden elde edilen bilgileri görüntülemek için dijital göstergelerden faydalanılır. Lojik elemanların farklı özelliklerde olması ve farklı yapıdaki yerlerde kullanılması, farklı karakteristiklere sahip gösterge ihtiyacını ortaya çıkarmıştır. Özellikle onlu sayıları görüntülemek amacıyla yedi parçalı gösterge olarak isimlendirilen elemanlardan faydalanılır. Yedi parçalı göstergeler farklı yapım şekillerine sahiptir. Yedi parçalı göstergeleri farklı yapı şekilleri ile birlikte inceleyelim

7.5.1. Yedi Parçalı Göstergeler (7 Segment Displays)

Sayısal göstergelerin bir çoğu, 0-9 arasındaki rakamları ve bazen onaltılık sistemdeki A-F harflerini göstermek için yedi parçalı gösterge elemanlarını kullanırlar. Yedi parçalı göstergeler, parçalardan her birisi içinden akım geçtiği zaman ışık yayacak şekilde özelliğe sahip malzemelerden yapılırlar. İçinden akım geçen parçalar ışık yayar ve oluşturulmak istenen şekil ortaya çıkar. Parçalar için gerekli sinyaller, uygun kod çözücü / sürücüler üzerinden elde edilir. Örneğin; BCD'den yedi parçalı sisteme dönüştürme işi, 7446 veya 7447 entegreleriyle gerçekleştirilebilir (Şekil 7.40).

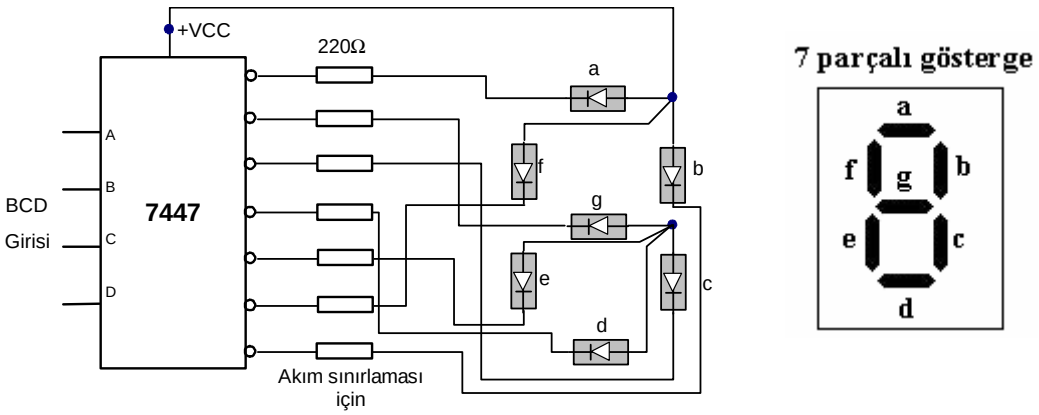
Yedi parçalı göstergeleri oluşturmak için kullanılan malzemeler; LED'ler (Light Emitting Diodes) ve parıldayan flamanlardır (Incondescent flamentler). Aynı çalışma prensibine sahip bu elemanların farkı; şekilleri oluşturmak amacıyla kullanılan parçaların farklı malzemelerden yapılmasıdır.

7.5.1.1. LED Kullanılarak Oluşturulan Yedi Parçalı Göstergeler

Işık yaymak amacıyla LED'lerin kullanıldığı göstergelerde, herbir parça için bir adet LED kullanılır. Parçalardaki LED'ler doğru yönde gerilim uygulanması ile, ilgili LED iletime geçerek ışık yayar.

Yedi parçalı göstergeler, ortak anotlu (Common anode) veya ortak katotlu (Common cathode) olarak imal edilirler. Uygulama yapılan devrenin özelliğine göre bunlardan biri tercih edilir.

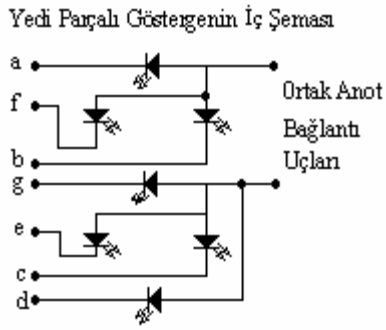
Ortak anotlu göstergede tüm anotlar birleştirilerek doğrudan $+V_{cc}$'ye bağlanır. Parçalarda bulunan LED'lerin katotları ise akım sınırlayıcı dirençler üzerinden sürücü entegresinin uygun çıkışları ile irtibatlanır (Şekil 7.40).

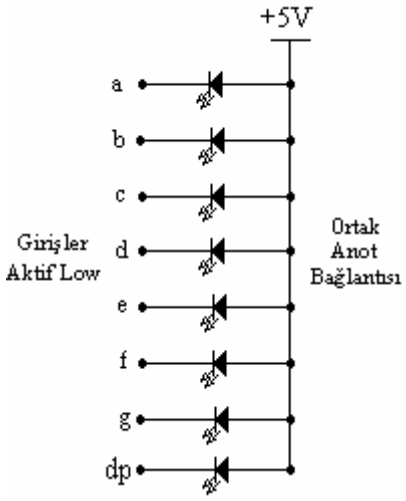


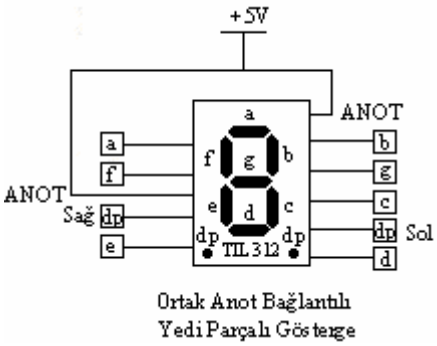
Şekil 7.40. Yedi parçalı göstergeye kod çözücü / sürücü entegresinin bağlanması.

Şekil 7.40'daki bağlantıdan görüleceği üzere, ışık vermesi istenilen LED'in bağlı olduğu entegre çıkışı '0' yapılarak LED'in iletime geçmesi ve ışık yayması sağlanır. Bu durumda, entegre çıkışında uygun kombinasyonlar oluşturularak bütün rakamları görüntülemek mümkün olur. Örneğin; '5' rakamını oluşturmak için; a, f, g, c, d LED'lerinin iletimde (ışık vermesi), b ve e LED'lerinin yalıtımda (ışık vermemesi) olması gerekir. Ortak anodlu göstergeleri sürmek için kullanılan 7447 entegresinin fonksiyon tablosu ve ortak anodlu göstergenin özellikleri Şekil 7.41'de görülmektedir.

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



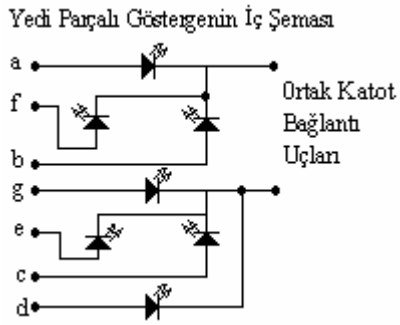


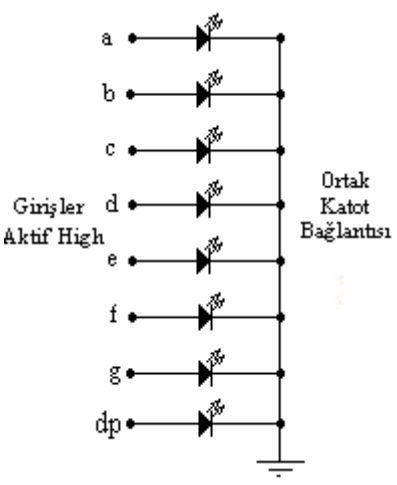


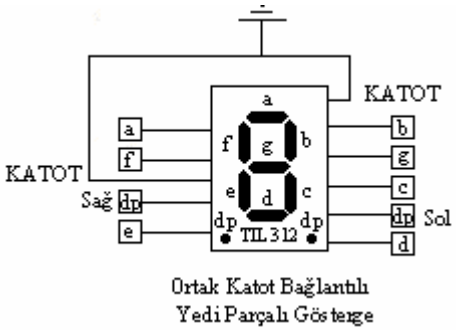
Şekil 7.41. Ortak anotlu yedi parçalı göstergenin kullanımı.

Ortak katotlu göstergenin ortak anotlu göstergeden tek farkı, katotların birleştirilmiş olmasıdır (Şekil 7.42). Ortak katotlu göstergeyi sürmek için; çıkışı aktif ‘1’ olan kod çözücü / sürücü (decoder / driver) devrelere ihtiyaç vardır. 7448 entegresi bu amaç için tasarlanmıştır .

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







Şekil 7.42. Şekil Ortak katotlu yedi parçalı göstergenin kullanımı.

Örnek 6: Şekil 7.41'deki devrede kullanılan LED'ler 10 mA ile 40 mA arasında akım çekerler. Her bir LED'in 2.7 voltta 10 mA akım çektiğini kabul ederek, her bir parçadan yaklaşık 10 mA akım akmasını sağlayacak akım sınırlayıcı direncin değerini hesaplayalım.

Çözüm : Şekil 7.41'de görülen dirençler, $+V_{cc}$ (5V) ile segment gerilimi (2,7 V) arasındaki farka eşit bir gerilim düşümüne neden olurlar. Dirençlerde düşen gerilim;

5V-2.7V = 2.3V olur.

Bu durumda parçalara seri bağlı dirençlerin değeri;

Rs = 2.3V / 10mA = 230 ohm

olarak bulunur. Standart direnç değeri olarak 220 ohm'luk direnç kullanılması iyi bir seçimdir.

7.5.1.2. Parıldayan Flamanlı Yedi Parçalı Göstergeler

Parıldayan flamanlı (Incasdescent flament) göstergeler, parça olarak ışık yayan flamanların kullanıldığı göstergelerdir. Bu flamanlardan akım geçirildiği zaman ısınırlar ve beyaz parlak bir ışık yayarlar. Göstergelerde renkli bir görünüm elde etmek için göstergeler boyanırlar.

Bu göstergelerde güç harcaması çok olduğundan, pil ile çalışan hesap makinası ve ölçü aleti gibi cihazlarda kullanışlı değillerdir. Bununla beraber yazar kasalarda ve güç harcamasının çok önemli olmadığı, elektrik ile çalışan sistemlerde kullanılırlar.

7.5.2. Sıvı Kristal Göstergeler (Liquid Crystal Displays - LCD)

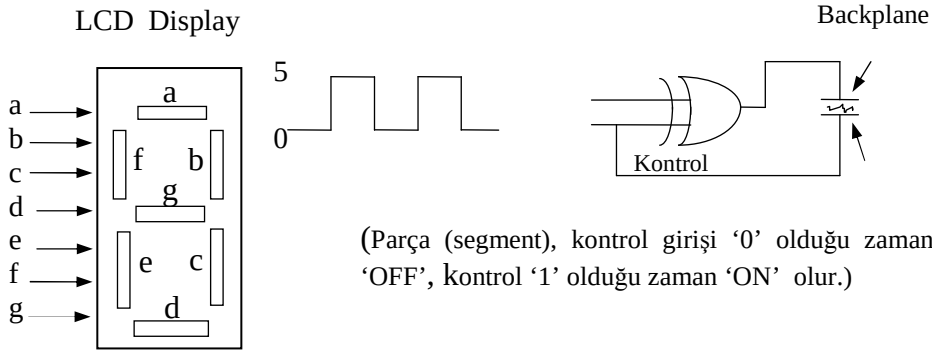
Sıvı kristal göstergeler (LCD'ler) genelde yedi parçalı LED'li göstergeler gibi dizayn edilirler (Şekil 7.43.a). LCD'ler düşük gerilim (3-15 V_{eff}) ve düşük frekanslı (25-60 Hz) sinyallerde çalışır ve çok az akım çekerler. AC gerilim, tüm parçalar için ortak olan tabanalan (backblane) ile parçalar arasına uygulanır ve bir parçayı çalıştırmak (turn on) için kullanılır. Parça ile tabanalan bir kondansatör özelliği oluşturur ve bu kondansatör AC akımın frekansı düşük kaldığı sürece çok küçük bir akım çeker. Ancak frekansın 25 Hz'in altına düşmemesi gerekir.

LCD'ler, parıldayan ve LED göstergelerden çok daha az akım çektiklerinden, hesap makinesi ve saat gibi pil ile çalışan cihazlarda yaygın olarak kullanılırlar. Bununla beraber, LCD göstergeler LED göstergeler gibi ışık yayınlamazlar ve bu nedenle harici bir ışık kaynağına ihtiyaç duyarlar.

LCD'nin Sürülmesi : Bir LCD parçası ile tabanalanı arasına AC gerilim uygulandığı zaman LCD parçası 'ON' olurken, DC gerilim uygulandığı zaman parça 'OFF' olur. AC gerilim yerine, AC gerilim etkisi oluşturan kare dalga bir sinyal, parça ile tabanalan arasına uygulanır.

Şekil 7.43.b'de bir parçaya kare dalga geriliminin uygulanışı görülmektedir. 40 Hz'lik kare dalga, tabanalanı ile 'ÖZELVEYA' kapısı girişine aynı anda uygulanır. 'ÖZELVEYA' kapısının diğer girişi, parçanın 'ON' veya 'OFF' olduğunu kontrol etmek için kullanılan kontrol girişidir.

Kontrol girişi '0' olduğu zaman, ÖZELVEYA'nın çıkışı 40 Hz'lik kare dalganın tamamen aynısı olur. Bu durum, parça ile tabanalanı arasına uygulanan gerilimin aynı olması sonucunu doğurur ve parça 'OFF' olur. Kontrol girişi '1' olduğu zaman, ÖZELVEYA çıkışı 40 Hz'lik kare dalga geriliminin tersi olur ve bu durum, parça ile tabanalanına uygulanan gerilimin birbirinin tersi olması sonucunu doğurur. Sonuç olarak; parça gerilimi +5V olurken, tabanalanı -5V olur. Bu ters gerilim, parçayı 'ON' konumuna getirir. Anahtarlama işlemi yedi parça için uygulanınca, yedi parçalı LCD gösterge ortaya çıkar.



Şekil 7.43. Sıvı kristal gösterge görünüşü ve kontrol devresi.

LCD'ler genelde CMOS elemanlarla birlikte kullanılır. Bunun nedeni; CMOS elemanların daha az enerji kullanmaları ve TTL elemanların, '0' seviyesinde 0.4V değerinde bir gerilim üretmeleridir. Bu gerilim LCD'nin ömrünü kısaltır. CMOS 4511 entegresi, BCD'den yedi parçalı LCD'ye kod çözücü / sürücü elemanıdır.

Tekrarlama ve Çalışma Soruları

1. Lojik elemanları, hangi elektronik devre elemanlarından oluşur?

2. Sayısal entegreleri kullanılan devre elemanına göre sınıflandırınız.

3. Sayısal entegreleri, yapılarında bulunan lojik kapı ve transistör sayısına göre sınıflandırınız.

4. 'Entegre parametreleri' terimini tanımlayınız.

5. Sayısal entegrelerde kullanılan parametreleri sıralayınız.

6. Lojik gerilim seviyesini şekille açıklayınız.

7. V_{IL} , V_{IH} , V_{OL} , V_{OH} terimlerinin açık anlamlarını yazınız.

8. 'Fan Out' terimini açıklayınız.

9. TTL entegrelerin LS tipi için 'Fan Out' değerini hesaplayınız.

10. 'Gürültülü bağışıklığı' ve 'Emniyet payı' terimlerini açıklayınız.

11. 'Yayılım gecikmesi' ve 'yayılım hızı' terimlerini şekiller yardımıyla açıklayınız.

12. Entegrelerde harcanan güç ne olarak tanımlanır ve birimi nedir?

13. 'Hız-Güç üretimi' nedir?

14. 'Akım besleme' ve 'akım yutumu' terimlerini açıklayınız.

15. Entegreleri yapılarındaki eleman sayısına göre sınıflandırınız.

16. Bipolar lojik ailesi içerisinde bulunan alt grupları sıralayınız.

17. MOS lojik ailesi hangi alt grup lojik aileleri içerir?

18. Direnç Transistör Lojik ile 'VEYA' kapısı oluşturarak, çalışmasını anlatınız.

19. Diyot Transistör Lojik ile 'VEDEĞİL' kapısı oluşturarak, çalışmasını anlatınız.

20. 'Yüksek Eşitlik Lojik' ile imal edilen elemanların çalışma gerilimleri ve serileri nedir?

21. 'Akım modlu mantık' nedir?

22. ECL mantığı ile 'VEYA' kapısı oluşturarak çalışmasını anlatınız.

23. ECL entegre devrelerin genel özelliklerini sıralayınız.

24. I^2L ile RTL teknolojileri arasındaki farkı 'DEĞİL' kapısı üzerinde açıklayınız.

25. TTL ailesi içerisindeki alt grupları sıralayınız.

26. TTL entegrelerin isimlendirilmesinde kullanılan 'AA74YYXXX' formatındaki harflerin anlamlarını açıklayınız.

27. TTL entegrelerin sahip oldukları çıkış şekillerini şekillerle açıklayınız.

28. 'Bağlantılı-VE' bağlantısı nedir?

29. Üç durumlu çıkışa sahip devrede 'yetkisizlendirme' durumu ne anlama gelmektedir?

30. 'Pull-up' transistörü nedir?

31. Standart TTL'lere ait çalışma gerilimi, güç tüketimi, 'Fan Out' değerleri nedir?

32. Standart ve Düşük güç tüketimli TTL arasındaki temel fark nedir?

33. Yüksek hızlı TTL serisinde kullanılan düşük direnç değerleri ve Darlington bağlantı ne sonuçlar doğurur?

34. Şotki TTL teknolojisinde kullanılan ve diğer TTL ailelerinden farklı olan mantık nedir?

35. Şotki diyotunun işlevi nedir?

36. Geliştirilmiş düşük güçlü Şotki serisinin genel özelliklerini sıralayınız.

37. TTL entegrelere ait 'Data Hand Book'larda verilen temel özellikler nelerdir?

38. 'Seviye Tetiklemeli Devre' ne demektir?

39. Entegre veri sayfaları ve veri katalogları hangi tür bilgileri içerir?

40. Yük sürme ve yükleme kapasitesini açıklayınız.

41. 7400 'VEDEĞİL' kapısının kaç tane 74ALS20 'VEDEĞİL' kapısını sürebileceğini hesaplayınız.

42. 'Fan in' terimini açıklayınız.

43. Fan Out'u '1' ve '0' durumlarında aynı olan entegrenin genel karakteristiği nedir?

44. 'Input Loading / Fan Out' başlığı altında, giriş için 400µ / 20µ ve çıkış için 100µ / 100µ değerlerine sahip ise Fan Out değerini bulunuz.

45. TTL entegrelerde, kullanılmayan girişin alabileceği durumları gösteriniz.

46. ‘Floating’ nedir?

47. ‘VE’ ve ‘VEDEĞİL’ kapılarında kullanılmayan girişler için uygun olan bağlantı çeşitleri nelerdir.

48. ‘VEYA’ ve ‘VEYADEĞİL’ kapılarında kullanılmayan girişler için uygun bağlantı çeşitlerini açıklayınız.

49. MOS teknolojisi ile üretilen dijital entegrelerin temel üstünlüğü nedir?

50. N-MOS ve P-MOS teknolojisi ile üretilen elemanları karşılaştırınız.

51. N-MOS ve P-MOS elemanların sembollerini çizerek, P kanal MOSFET’in anahtarlama durumunu şekille açıklayınız.

52. N-MOS ile üretilen ‘VEDEĞİL’ ve ‘VEYADEĞİL’ kapılarının şekillerini çizerek, çalışma prensiplerini özetleyiniz.

53. CMOS entegrelerde bulunan parametreleri özetleyiniz.

54. Arabirim elemanı nedir?

55. TTL entegredeki elemanların CMOS elemanları sürmesi durumunda hangi problemler ile karşılaşılır?

56. CMOS elemanların TTL elemanları sürmesi durumunda ortaya çıkacak sorunlar nelerdir?

57. TTL ve CMOS elemanlara yük bağlantı şekillerini özetleyiniz.

58. 7 Parçalı göstergeyi açıklayınız.

59. LED ile yapılan ortak anot ve ortak katotlu bağlantıları açıklayınız.

BÖLÜM 8

BİLEŞİK MANTIK DEVRELERİ

(COMBINATIONAL LOGIC)

Giriş

Dijital sistemlerde kullanılan mantık devreleri, bileşik (combinational) devreler ve ardışıl sıralı (sequential) devreler şeklinde gruplandırılabilir.

Temel lojik kapılardan oluşan ve devrelerin çıkışları doğrudan girişlerin o anki durumlarına göre belirlenen devrelere, **'bileşik mantık devreleri'** denir. Bileşik devreler bazen, **'birleşik mantık devreleri'** olarak ta isimlendirilir. Bir bileşik devre, giriş değişkenleri, lojik kapılar ve çıkış değişkenlerinden oluşur (Şekil 8.1). Lojik kapı, giriş değişkenlerini alır, bunları işler ve çıkış için bilgi (değişkenler) üretir. Yapılan işlem, ikili giriş verilerin işlenmesi ve uygun çıkış verileri şekline dönüştürülmesidir.



Şekil 8.1. Bileşik mantık devresi blok şeması.

Giriş verileri; bir harici kaynaktan gelen ‘n’ sayıda ikili giriş değişkenlerini, çıkış verileri; bir harici devreye doğru yönelmiş ‘m’ sayıda çıkış değişkenlerini içerir (Şekil 8.1). Giriş değişkenlerinin değeri, 2^n sayıda farklı ikili giriş kombinasyonundan birisi olabilir ve herbir giriş kombinasyonu için yalnızca bir çıkış kombinasyonu mevcuttur.

Çok farklı uygulama alanları bulunan bileşik mantık devreleri, dört farklı grup altında incelenebilir:

i- Kodlama ile İlgili Lojik Devreler: Kodlayıcı (Encoder), Kod çözücü (decoder), Kod değiştirici / çevirici (Code converter).

ii- Çoklayıcı Devreler, Veri seçiciler (Multiplexer-Data selector).

iii- Azlayıcı Devreler, Veri dağıtıcılar (Demultiplexer - Data distributor).

iv- Kıyaslama ve Aritmetik İşlemler ile İlgili Devreler: Karşılaştırıcı (comparator), Toplayıcı (adder), Çıkarıcı (substractor), Çarpıcı (multiplier).

Farklı alanlarda farklı amaçlar için kullanılan bileşik devrelerin tasarımında aynı işlem sırası takip edilir. Bu nedenle bileşik devreleri detaylandırmadan önce, bileşik devrelerin tasarımında kullanılan prensipleri inceleyelim.

8.1. Bileşik Devre Tasarım Esasları

Bir bileşik devrenin tasarımı, problemlerin ifade edilmesiyle başlayıp, lojik devrenin çizilmesi ile biter. Lojik tasarımın içerdiği işlem basamakları aşağıdaki şekilde özetlenebilir:

i- Problem belirlenir.

ii- Giriş değişkenlerinin sayısı ve gerekli çıkış değişkenleri tespit edilir.

iii- Giriş ve çıkış olarak kullanılacak değişkenlere isim verilir.

iv- Giriş ve çıkış değişkenleri arasındaki gerekli ilişkiyi belirleyen doğruluk tablosu yapılır.

v- Her bir çıkış için uygun Boolean fonksiyonu yazılır.

vi- Elde edilen Boolean fonksiyonları sadeleştirilir.

vii- Lojik devre çizilir.

İşlem basamakları içerisinde en çok dikkat edilmesi gereken adım, sözlerle çerçevesini çizdiğimiz işlemlerin doğruluk tablosu haline getirilmesidir. Çünkü bu aşamada yapılacak bir hata, daha sonraki tüm adımlarda hatalı işlem yapılması sonucunu doğurur.

Doğruluk tablosundan elde edilen çıkış eşitliklerini sadeleştirmek için, sadeleştirme yöntemlerinden uygun olan birisi kullanılabilir. Sadeleştirilen eşitliklerin aşağıdaki özellikleri taşıması istenir:

i- En az sayıda lojik kapı içermesi.

ii- Herbir kapının en az sayıda girişe sahip olması.

iii- Devrenin minimum yayılım zamanına sahip olması.

iv- Devrenin minimum sayıda bağlantı içermesi.

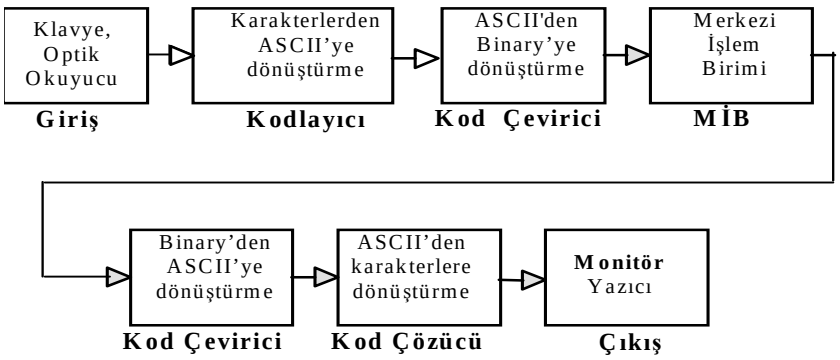
v- Herbir kapının, sürme kapasitesi sınırının altında eleman sürmesi.

Buradaki tüm kriterlerin aynı anda sağlanması mümkün olmayabilir. Bu durumda, kriterler arasından tasarlanan devrenin aranan özelliklere göre seçim yapılır.

8.2. Kodlama İle İlgili Lojik Devreler

Daha önceki konularda işlenen kodlama işlemi, dijital bilgisayarlarda yoğun olarak gerçekleştirilen işlemlerden birisidir. Bilgisayarda yapılan işlemleri blok şema olarak göstermek, bilgisayarda yapılan kodlama ile ilgili işlemler hakkında fikir verebilir.

Bilgisayarda gerçekleştirilen işlemleri özetleyen Şekil 8.2'deki şemadan görüleceği üzere; veri, bilgisayarın merkezi işlem birimi ile çevre birimleri arasında kodlanmış olarak (ASCII kodu) gönderilir. Kodlanmış bilgiler klavye, optik okuyucu gibi karakter kaynaklarından elde edilir. Bu kaynaklardan elde edilen bilgiler ASCII kodundadır. Bilgisayarın merkezi işlem biriminin (MİB) ikili sayılarla çalışması nedeniyle, ASCII kodlanmış karakterler MİB girişinde ikili sayılara dönüştürülür. Veriler üzerinde yapılan tüm işlemler, MİB'de ikili sayı formunda gerçekleştirilir.



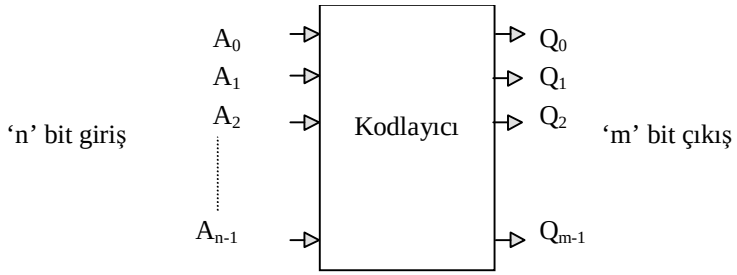
Şekil 8.2. Bilgisayarda girişten çıkışa bilgi çevriminin blok şeması.

MİB'de işlenen bilgilerin çevre birimlere ulaşması için, veri üzerinde daha önce yapılmış olan işlemlerin tersi işlemlerin yapılması gerekir. Bu nedenle, MİB'ne kadar yapılan işlemlerin karşıtları, ters sırasıyla yeniden yapılır. Şimdi Şekil 8.2'de blok şema ile özetlenen işlemlerin kodlama ile ilgili olanlarını sırası ile inceleyelim.

8.2.1. Kodlayıcı Devreler (Encoders)

'n' bit girişli bir sistemde, girişindeki bilgiyi ikili sayı sisteminde kodlanmış olarak çıkışında veren bileşik devreye, **'kodlayıcı devre'** (encoder) denir. Farklı bir bakış açısı ile, insanlar tarafından kolayca anlaşılabilen rakam ve krakterlerin farklı bilgiler şekline dönüştürülmesini sağlayan devreler, 'kodlayıcı devreler' olarak isimlendirilir.

Kodlayıcı devrelerde, herhangi bir anda girişlerden sadece bir tanesi aktif olabilir ve aktif olan girişe göre 'm' bitli çıkış kodu üretilir. Şekil 8.3'de, 'n' bit girişli 'm' bit çıkışlı bir kodlayıcı devrenin blok şeması görülmektedir.

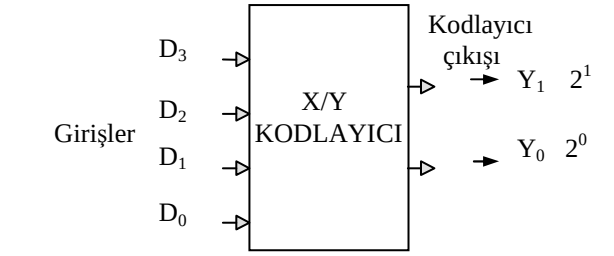


Şekil 8.3. Kodlayıcı devresi genel blok şeması.

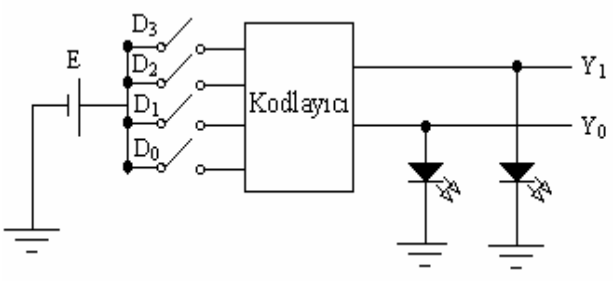
Dört girişe sahip bir kodlayıcının çıkışı, ikili sisteme Şekil 8.4'deki gibi dönüştürülebilir. Dört giriş, çıkışta iki bitlik ikili sayı ile temsil edilir. Girişlerden herhangi birinin aktif olması ile çıkışlar uygun kombinasyonu alır. Bu devreye, '**4 girişten 2 çıkışa kodlayıcı devresi**' denebilir. Aynı şekilde, 8 giriş ve üç bit çıkış koduna sahip bir kodlayıcı devresi, sekizli sistemden ikili sisteme kodlama işlemi yapar. Şekil 8.5'de sekizli sistemden ikili sisteme kodlayıcı devrenin doğruluk tablosu ve lojik devresi görülmektedir.

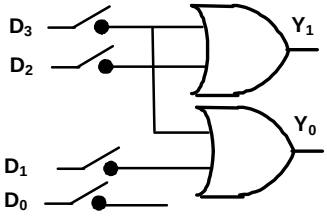
Şekil 8.5'deki devrede girişlerden herhangi biri aktif (1) olunca, çıkışlar uygun kombinasyonda '1' değerlerine sahip olur. Örneğin; A_3'ün aktif olması ile çıkışlar Q_2=0, Q_1=1, Q_0=1 değerlerini alır.

A_0 girişi hiçbir yere bağlanmamıştır. Çünkü, 'Q_3, Q_2, Q_1' çıkışları '000' değerlerine sahiptir.

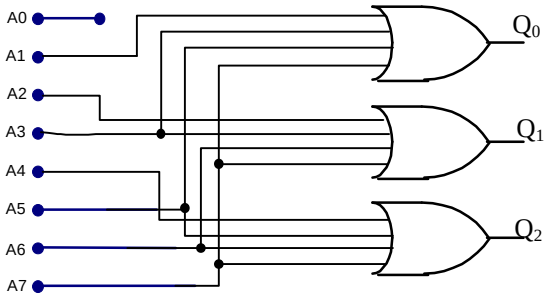


D_3	D_2	D_1	D_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1





Şekil 8.4. İkili sistemde çıkış veren kodlayıcı sembolü, doğruluk tablosu ve uygulaması.



A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	X	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	0	X	0	1	0
0	0	0	0	1	0	0	X	0	1	1
0	0	0	1	0	0	0	X	1	0	0
0	0	1	0	0	0	0	X	1	0	1
0	1	0	0	0	0	0	X	1	1	0
1	0	0	0	0	0	0	X	1	1	1

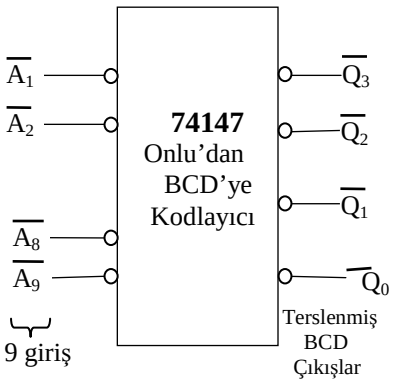
Şekil 8.5. Sekizli sistemden ikili sisteme kodlayıcı devresi ve doğruluk tablosu.

Aynı anda yalnız bir girişin aktif olduğu durumlarda kodlama işlemi gerçekleştirilir. Birden fazla girişin aynı anda aktif olması durumunda problem oluşur. Bu durumda çıkışta kodlanan bilgi, girişlerden farklı bir değeri ifade eder. Bu problemi ortadan kaldırmak yani iki girişin aynı anda aktif olmasını önlemek için, öncelikli kodlayıcı olarak isimlendirilen sistem uygulanır.

Öncelikli kodlayıcı sistemde, birden fazla girişin aynı anda '1' olması durumunda girişlerden yalnızca biri seçilir (en yüksek değere sahip olan) ve seçilen girişe göre çıkış verilir. 74147, 74LS148, 4532 ve 74HC147 entegreleri, öncelikli kodlayıcılı entegrelerdir.

Onluk sistemden ikili sisteme kodlayan bir kodlayıcı devresi (BCD), 74147 entegresi kullanılarak gerçekleştirilebilir. 74147 entegresi 9 adet girişe (aktif '0') sahiptir ve bu girişler 1'den 9'a kadarki sayıları temsil ederler. Çıkışlarda elde edilen kodlanmış değerler, en yüksek değerlikli girişin terslenmiş BCD kodlu eşdeğeridir (Şekil 8.6).

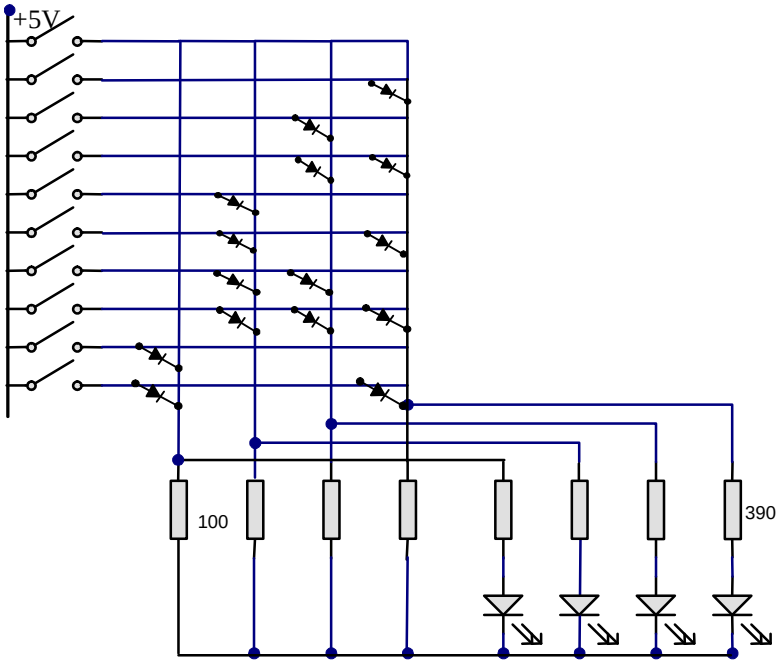
$\overline{A_1}$	$\overline{A_2}$	$\overline{A_3}$	$\overline{A_4}$	$\overline{A_5}$	$\overline{A_6}$	$\overline{A_7}$	$\overline{A_8}$	$\overline{A_9}$	$\overline{Q_3}$	$\overline{Q_2}$	$\overline{Q_1}$	$\overline{Q_0}$
1	1	1	1	1	1	1	1	1	1	1	1	1
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0



Şekil 8.6. Onlu sistemden BCD'ye kodlayıcı devre doğruluk tablosu ve blok şeması.

74147 entegresinin doğruluk tablosu incelenirse; ilk satırın çıkışındaki değerin 0000'ın tümleyeni (tersi) yani 1111 olduğu ve bunun BCD 15'e karşılık geldiği görülür. İkinci satırdaki çıkış değerlerinin A_9'nun karşılığı olarak 1001 değerinin tümleyeni yani 0110 olduğu bulunur. Takip eden satırlarda, aynı şekilde aktif olan girişin karşılığı olan ikili sayının tümleyeninin bulunduğu görülür. 74147 entegresi çıkışı, girişlerden hiçbirinin aktif olmadığı durumda 1111 olur ve bu değer 0'ı temsil eder.

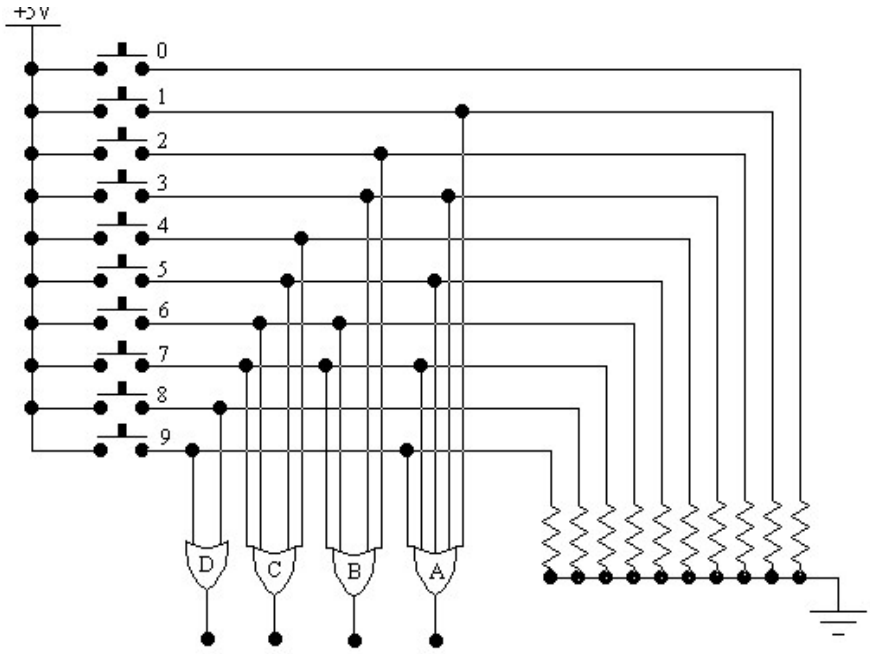
Onlu sistemden BCD'ye kodlayıcı devrelerin en basit şekli, diyot matrisle yapılan ve anahtar kodlayıcı (switch encoder) diye isimlendirilen kodlama devresidir (Şekil 8.7). Bu şekilde yapılan bir kodlama işleminde; anahtarlardan birisi kapatıldığı zaman, devre kapatılan anahtara bağlı olarak çıkışında uygun ikili sayı üretir. Örneğin; 3 nolu anahtara basıldığı zaman çıkışta '0011' değeri okunurken, 7 nolu anahtara basıldığı anda '0111' çıkışı elde edilir.



Şekil 8.7. Diyot matris, onlu sayı sisteminden ikili'ye kodlayıcı devresi.

Onlu sistemden BCD'ye dönüştürücü devreyi 'VEYA' kapıları ile oluşturursak, Şekil 8.8'deki devre oluşur. Desimal sayılara karşılık gelen anahtarlardan birisine basılması ile 'VEYA' kapılarının çıkışlarında (A,B,C ve D) '0' veya '1' değerleri elde edilir. Elde edilen değerler, kapatılan anahtarların karşılığı olarak üretilen ikili kodlu sayılardır. Örneğin; 5 nolu anahtarın kapatılması ile elde edilen '0101' ikili değeri, onlu tabandaki '5' sayısının BCD kodlu değeridir. Aynı şekilde, '9' sayısının karşılığı olarak çıkışta BCD kodlu '1001' değeri oluşur.

Bu uygulamadakine benzer şekilde, matris anahtarlar ve 74C922 entegresi kullanarak onaltılı sistemdeki sayıların ikili sayılara dönüştürülmesi mümkündür.



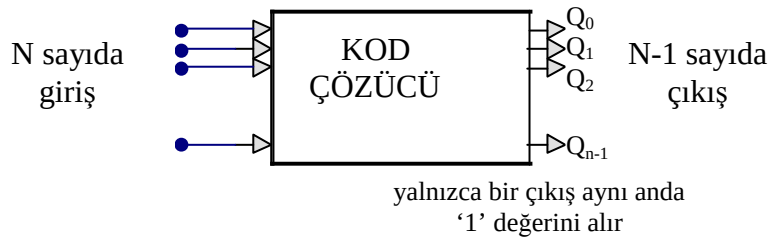
	D	C	B	A
A	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

İkili Kelimeler

Şekil 8.8. 'VEYA' kapıları ile oluşturulan onludan ikili sayı sistemine kodlayıcı devresi.

8.2.2. Kod Çözücüler (Decoders)

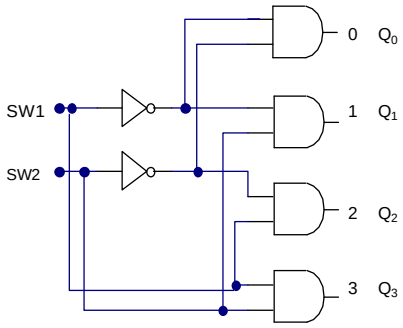
Dijital sistemlerde bilgiler ikili sayılar olarak temsil edilir ve yapılan işlemler ikili sayılarla gerçekleştirilir. '**Kod çözücü**' (decoder) devresi; kodlayıcı devresinin tersini yaparak, 'n' sayıdaki giriş hattından gelen ikili bilgileri maksimum 2^n sayıda çıkış hattına dönüştüren bileşik bir devredir. Diğer bir deyişle; değişik formlarda ifade edilen bilgilerin insanların kolayca anlayabileceği şekle dönüştürülmesini sağlayan devreler, **'kod çözücü devreler'** olarak isimlendirilir. Kodu çözülen 'n' bitli bilginin kullanılmayan girişleri varsa kod çözücü çıkışındaki çıkış sayısı 2^n'den az olur. Şekil 8.9'da kod çözücü blok şeması görülmektedir.



Şekil 8.9. Kod çözücü blok şeması.

Şekil 8.10'da iki giriş ve dört çıkışlı çözücü devresi lojik şeması görülmektedir. Bu devrede iki bitlik girişin kodu çözülerek, dört farklı çıkış üretilir. Her bir giriş kombinasyonunda yalnızca bir çıkış '1' durumundadır. Örneğin; 00 giriş durumunda Q_0 çıkışı '1' durumunda iken, 10 giriş kombinasyonunda Q_2 çıkışı '1' değerini alır.

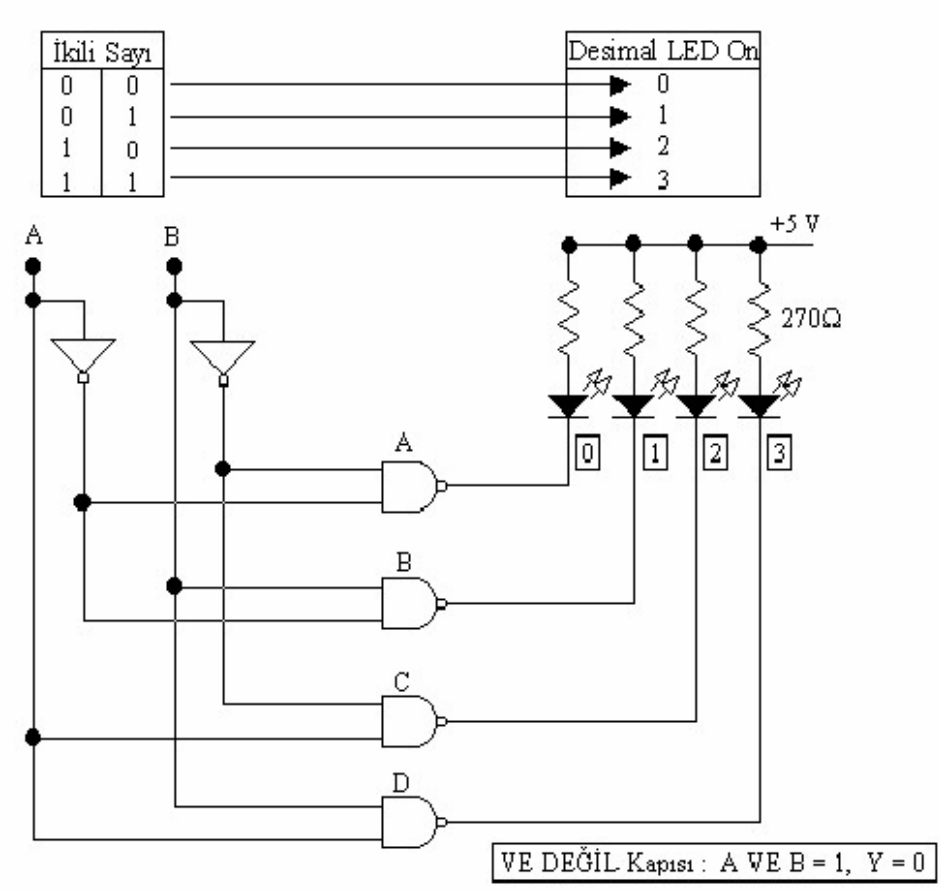
Girişler		**Çıkışlar**			
sw_1	**sw_2**	**Q_0**	**Q_1**	**Q_2**	**Q_3**
B	A	0	1	2	3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Şekil 8.10. İki giriş dört çıkış kod çözücü doğruluk tablosu ve açık şeması.

İkili sistemden onlu sisteme kod çözücü devresi, girişinden uygulanan ikili bilgilere göre çıkışlarından birisi '0' değerini alır (LED yanar). Çıkışlardaki LED'lerden her birisi onlu sistemdeki sayılardan birisini temsil eder.

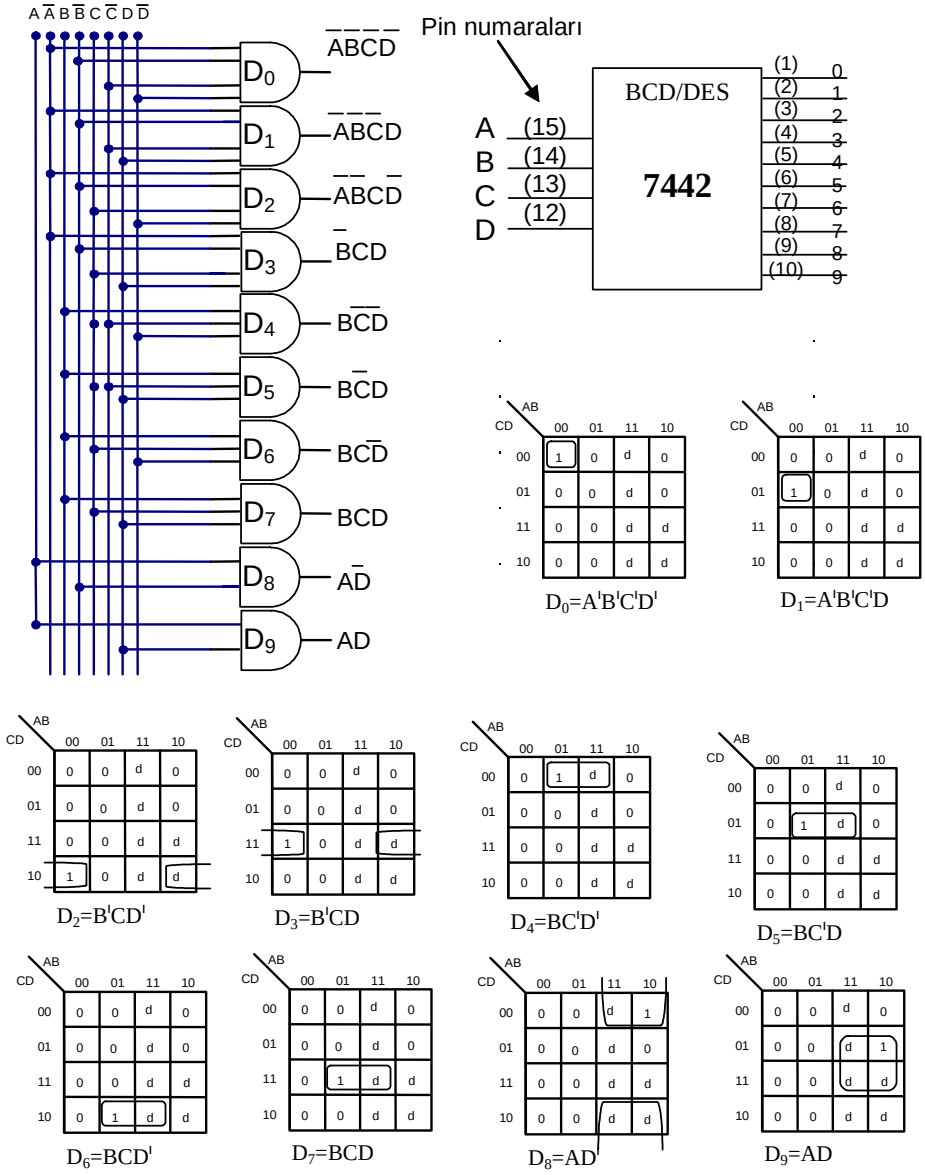
Şekil 8.11'deki devrede her iki girişin '0' olması durumunda; girişlerin terslenerek uygulandığı 'A' kapısının çıkışı '0' değerini alır ve katodundaki gerilim anoduna göre (-) olan LED yanar. Diğer kapı devreleri giriş değerlerine göre çıkışlarında uygun olan LED'in yanmasını sağlar. Her bir LED bir onlu sayıyı temsil ettiğinden onlu değer çıkıştan bulunmuş olur.



Şekil 8.11. NAND kapıları ile oluşturulan ikili sistemden onlu sisteme kod çözücü devresi.

Örnek 1: BCD'den onlu sisteme çevirimde kullanılan ve 0-9 arasındaki sayılara kod çözme işlemini gerçekleştiren devreyi tasarlayalım.

Bu şekilde çalışan kod çözücüler entegre devre olarak imal edilmiş olsalar da kod çözücü tasarlama mantığı açısından iyi bir örnek olacaktır. BCD'den onlu sisteme kod çözücü devresinin dört bitlik girişine karşılık, her biri bir ondalık sayıyı temsil eden 10 çıkış bulunur. Bu şekildeki bir devre, **'4 giriş / 10 çıkışlı BCD kod çözücü'** olarak isimlendirilir.



Şekil 8.12. 4 Giriş / 10 Çıkışlı BCD kod çözücü devresi karnaugh haritası, açık şeması ve entegre devresi.

Tasarım sırasında devrenin 10 çıkışının bulunması nedeniyle, 10 tane Karnaugh haritası çizip herbirine ait ilgili fonksiyonu yazmak gerekir. Aynı anda çıkışlardan yalnızca biri '1' olacağından, çıkışları Q_0 - Q_9 şeklinde isimlendirip ilgili hücrelere yerleştirmek, fonksiyonları yazmamıza imkan tanır. Eşitlikleri yazmada, BCD sistemde kullanılmayan sayıları temsil eden kombinasyonlar fark etmeyen olarak ifade edilir (Şekil 8.12).

Fark etmeyenleri de kullanarak (karnaugh haritasında 'd' ile gösterilen), her bir çıkış için ilgili fonksiyonları yazarsak;

$D_o=A'B'C'D'$, $D_1=A'B'C'D$, $D_2=A'B'CD'$, $D_3=B'CD$, $D_4=BC'D'$,

$D_5=BC'D$, $D_6=BCD'$, $D_7=BCD$, $D_8=AB'$, $D_9=AD$

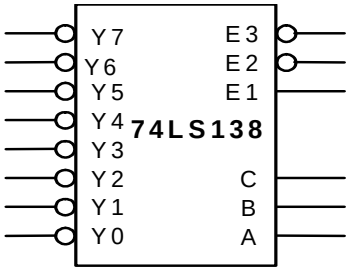
eşitlikleri elde edilir. Elde edilen eşitlikleri gerçekleştirecek lojik kapıların çizilmesi ile, Şekil 8.12'daki lojik devre oluşur.

Kod çözücüler, bilgisayarlarda hafıza devrelerinde depolanan bir bilginin adresini bulmak amacı ile yaygın olarak kullanılırlar. 7442 entegresi BCD'den onlu sisteme kod çözme işlemini yapar (Şekil 8.12). 7445 entegresi ise BCD'den onlu sisteme kod çözücü / sürücü devresidir. Sürücü terimi, entegrenin açık-kollektör olması nedeniyle verilmiştir.

Diğer bir kod çözücü uygulaması, ikili bilgilerin onlu olarak görüntülenmesini sağlayan, ikili'den yedi parçalı göstergeye çevirme yapan kod çözücü devresidir. Bu devre kod çeviriciler başlığı altında incelenecektir. Ancak bu arada, kod çözücülerde kullanılan yetkilendirme girişini açıklayalım.

Yetkilendirme (Enable) Girişi:

Kod çözücü entegrelerin büyük bir çoğunluğu, entegrenin çalışmasını kontrol eden bir veya birden çok yetkilendirme girişi içerir. Yetkilendirme girişine uygun sinyal verilmediği sürece kod çözücü devre çalışmaz.



Şekil 8.13. 8'de 1 kod çözücü entegresi ayak bağlantısı.

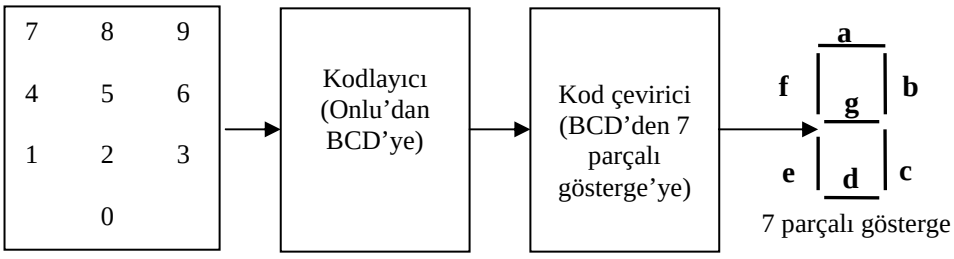
Yetkilendirme girişine uygun sinyal verildikten sonra, uygulanan girişe karşılık gelen çıkış elde edilir. 74LS138 entegresi, 3 yetkilendirme girişine sahip ikili sistemden 8'li çıkışa kod çözücü elemandır. Şekil 8.13'de blok şeması görülen bu entegrenin E_1, E_2 ve E_3 yetkilendirme girişleri, entegrenin içerisinde bir 'VE' kapısı ile birleştirilmiştir. Entegrenin kod çözme işlevini yapabilmesi için 'VE' kapısının çıkışının '1' ve yetkilendirme girişlerinin $E_1=E_2=0$ ve $E_3=1$ olması gerekir. 8 çıkışa sahip kod çözücü devreleri bir arada kullanıldığında, yetkilendirme girişlerinin yardımıyla 16 çıkışa veya 32 çıkışa sahip kod çözücü devreler tasarlanabilir.

8.2.3. Kod Çeviriciler (Code Converters)

'**Kod çevirici**', bir kodlama yönteminde ifade edilen bilgiyi, başka bir kodlama yöntemine çeviren lojik bir devredir. Kod çevirici devrelere örnek olarak, BCD'den yedi parçalı göstergeye, ikili'den BCD'ye, ikili'den gray koda, gray kod'dan ikili'ye, BCD'den ASCII ve EBCDIC'ye veya tersine kod çevirmeleri verilebilir.

Hesap makinelerinde veya bilgisayarlarda kullanılan tuş takımı / gösterge sistemi, kod çevirme işlemlerinin birkaçının bir arada yapıldığı bir düzenektir. Tuş takımı / gösterge sisteminde, tuş takımıyla gösterge arasında kodlama ve kod çevirme işlemleri yapılır (Şekil 8.14). Tuş takımındaki tuşlara basılmak suretiyle elde edilen değerler, onlu sistemden BCD'ye dönüştürülür (kodlayıcı). BCD olarak elde edilen bilgiler, BCD'den 7 parçalı göstergeye kod çevirme işleminden geçirilir ve göstergede onlu olarak okunur.

Tuş takımı / gösterge sisteminde bulunan devreler bir arada düşünülerek kod çevirici olarak isimlendirilebilir. Bu durumda, Şekil 8.14'de görülen kod çevirici işleminin genel anlamlı olduğu ve kodlayıcı/kod çözücü işlemlerini içerdiği söylenebilir. Bu özellik, kod çevirici devrelerin çok değişik yerlerde kullanılması sonucunu doğurur. Kod çevirici kullanılan devrenin (yerin) özelliklerine göre tasarlanır.



Şekil 8.14. Tuş takımı / gösterge sistemi blok şeması.

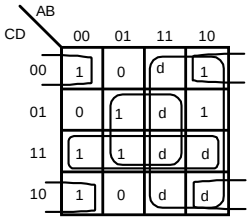
Örnek 2: Kod çevirici devrelerin tasarım prensibini açıklamak için, BCD'den yedi parçalı gösterge sistemine kod çevirici devrelerin tasarım adımlarını açıklayalım (Şekil 8.14). Yedi parçalı göstergelerin 7.Bölümde açıklanması nedeniyle doğrudan tasarım işlemini anlatalım.

BCD kodlu bilgiyi 7 parçalı göstergeye çeviren kod çevirici devrenin tasarımı aşağıdaki işlem basamakları ile gerçekleştirilir.

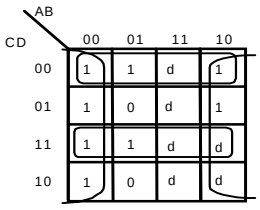
i- Çevrimi yapılacak kodların doğruluk tablosu çıkarılır. Giriş bilgisi BCD kodlu sayılar olduğundan giriş değerleri olarak BCD kodlu sayılar kullanılır. Çıkış ise yedi parçalı göstergedir. Yedi parçalı gösterge için 7 farklı bilgi gerektiğinden, doğruluk tablosunda bütün bilgilerin temsil edilebilmesi için 7 çıkış sütununa ihtiyaç vardır (Şekil 8.15).

ii- Doğruluk tablosundaki her bir çıkışa göre Karnaugh haritası çizilir. Doğruluk tablosunda çıkışı ifade eden her bir sütun bir Karnaugh haritası ile temsil edilir (Şekil 8.15).

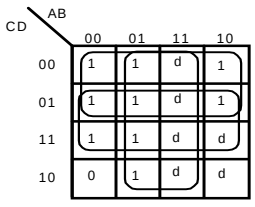
Onlu	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1



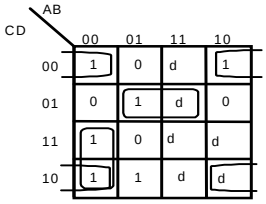
a=A+CD+BD+B'D'



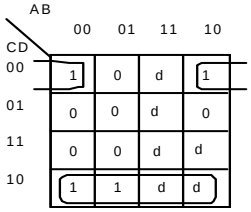
b=B'+CD+C'D'



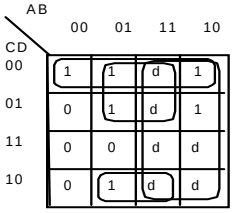
c=A'C'+B'C'+A'D+A'B



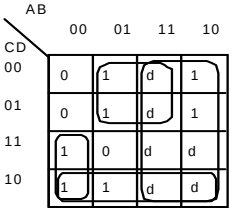
d=C'D'+BC'D+A'B'C+AB'



e=CD'+B'C'D'



f=BCD'+A +BC'+C'D'



g=CD'+BC'+A 'B'C+A

Şekil 8.15. BCD'den yedi parçalı göstergeye kod çevirici devresi tasarımı.

iii- Karnaugh haritalarından faydalanılarak lojik eşitlikler yazılır.

iv- Elde edilen eşitlikleri temsil eden lojik devreler çizilir. Her bir lojik devrenin çıkışının yedi parçalı göstergedeki uygun parçaya bağlanması ile kod çevirici tasarımı tamamlanır.

Örnek 3: BCD kodlu bir sayıyı 3 Fazlalık (Excess-three) koduna çevirecek devreyi tasarlayalım.

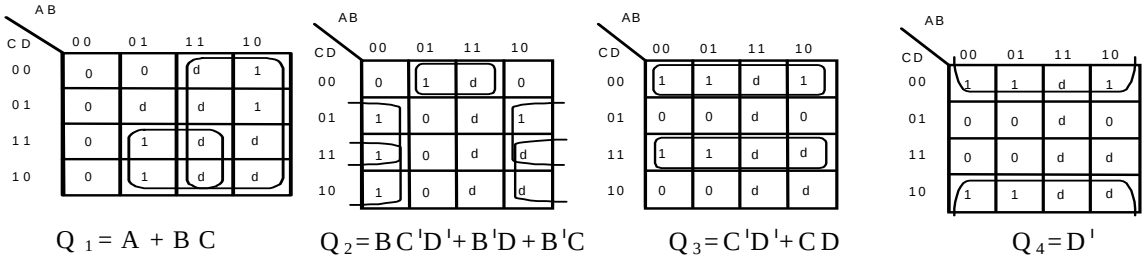
i- BCD ve 3-Fazlalık kodlu sayıları gösteren doğruluk tablosu oluşturulur (Şekil 8.16). BCD kodlu sayıların dört bit ile ifade edilmesi nedeniyle, dört farklı çıkış bulunması gerekir.

ii- Doğruluk tablosu çıkış sütunlarındaki değerler Karnaugh haritalarına taşınır.

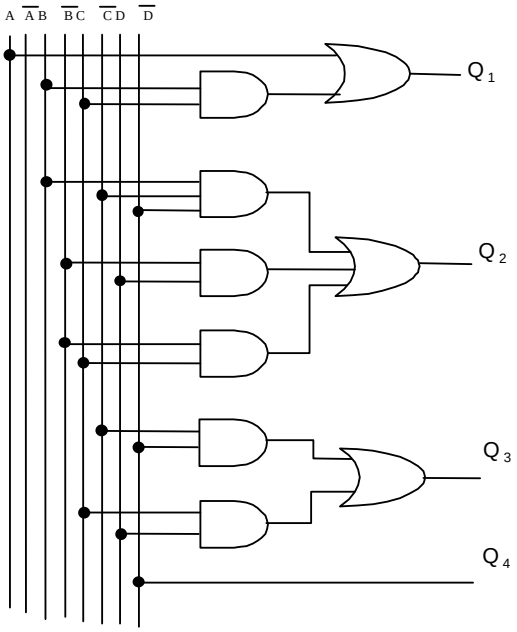
iii- Karnaugh haritalarına taşınan bilgilerin gruplandırılması ile herbir çıkış için uygun Boolean eşitliği yazılır (Şekil 8.16).

iv- Yazılan eşitliklerin kapı devreleri ile oluşturulması ile, BCD'den Artı 3 koduna çeviren kod çevirici devresinin tasarımı tamamlanır (Şekil 8.17).

Onlu	A	B	C	D	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0



Şekil 8.16. BCD'den Artı 3 koduna kod çevirici devresi tasarımı.



Şekil 8.17. BCD'den Artı 3 koduna çevirici devresi lojik şeması.

8.2.4. Kodlama ile İlgili Devrelere ait Uygulamalar

Kodlama ile ilgili devreler olarak tanımlanan kodlayıcı, kod çözücü ve kod çevirici devrelerin çok farklı alanlarda çeşitli amaçlara yönelik uygulama devreleri bulunmaktadır. Bu devrelere birkaç örnek verelim.

8.2.4.1. Yedi Parçalı LED Gösterge Kod Çevirici / Kod Çözücü Devreleri

7447 TTL lojik entegresi, LED'lerden oluşan yedi parçalı göstergeleri çalıştırmak için kullanılan kod çözücü / sürücü entegresidir (Şekil 8.18).

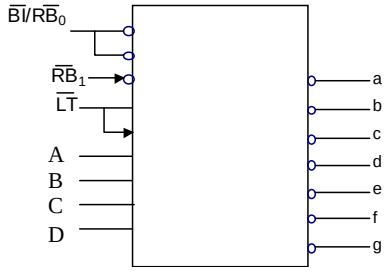
Lojik sembolü Şekil 8.18.a'da verilen 7447 entegresi, Şekil 8.18.b'deki ayak bağlantısına sahiptir. Entegrede dört giriş ve yedi çıkış bulunmaktadır. Şekil 8.18.c'deki tabloda farklı giriş kombinasyonlarında çıkışların (segmentlerin) aldığı değerler görülmektedir. Giriş değerlerine göre çıkışlarda oluşan değerlere bağlı olarak, Şekil 8.18.b'deki yedi parçalı göstergede sayılar oluşur.

7447 lojik sembolünde değişik amaç için kullanılan üç adet pin bulunmaktadır: LED test girişi (LT), söndürme (blanking) girişi (BI) ve dalgalı söndürme (ripple blanking) girişi (RBI). Her üç girişte aktif '0' girişlerdir.

7447 BCD'den yedi parçalı göstergeye kod çevirici devresi aktif '0' çıkışa sahiptir ve ortak

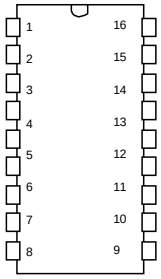
anodlu göstergeleri sürmek için kullanılır. 7448 entegresi ise, aktif ‘1’ çıkışa sahiptir ve bu nedenle ortak katotlu göstergeleri sürmek için kullanılırlar.

LED test girişi olan LT’nin ‘0’ olması ile tüm çıkışlar ‘0’ olur ve bu çıkışlar göstergedeki tüm parçaların yanarak test edilmesini sağlar. BI girişi ile, entegrenin giriş değerlerine bakılmaksızın çıkışlarının sürdüğü tüm gösterge parçaları söndürülür. Bu özellik, çok sayıda göstergenin bir arada kullanılması durumunda, gözükmesi istenmeyen göstergelerin tamamen kullanım dışı bırakılması amacıyla kullanılır. Üçüncü giriş olan RBI, entegreye uyguladığımız gerilimi değiştirmeden göstergenin parlaklığını değiştirmemize olanak sağlar. RBI’ye uygulanan sinyalin ‘ON’ ve ‘OFF’ zamanları değiştirilerek gösterge parçalarının parlaklığının değiştirilmesi sağlanır.

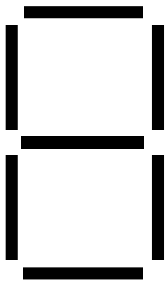


7446A, 7447A, 74LS47
Lojik Sembolü

(a)



7447 ayak bağlantısı



(b)

ONLUK SAYI veya FONKSİYON	GİRİŞLER						$\overline{BI}$ / $\overline{RBO}$	ÇIKIŞLAR							NOT
	(LT)	(RBI)	D	C	B	A		a	b	c	d	e	f	g	
0	1	1	0	0	0	0	1	ON	ON	ON	ON	ON	ON	OFF	1
1	1	X	0	0	0	1	1	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	1	X	0	0	1	0	1	ON	ON	OFF	ON	ON	OFF	ON	
3	1	X	0	0	1	1	1	ON	ON	ON	ON	OFF	OFF	ON	
4	1	X	0	1	0	0	1	OFF	ON	ON	OFF	OFF	ON	ON	
5	1	X	0	1	0	1	1	ON	OFF	ON	ON	OFF	ON	ON	
6	1	X	0	1	1	0	1	OFF	OFF	ON	ON	ON	ON	ON	
7	1	X	0	1	1	1	1	ON	ON	ON	OFF	OFF	OFF	OFF	
8	1	X	1	0	0	0	1	ON	ON	ON	ON	ON	ON	ON	
9	1	X	1	0	0	1	1	ON	ON	ON	OFF	OFF	ON	ON	
10	1	X	1	0	1	0	1	OFF	OFF	OFF	ON	ON	OFF	ON	
10	1	X	1	0	1	1	1	OFF	OFF	ON	ON	OFF	OFF	ON	
12	1	X	1	1	0	0	1	OFF	ON	OFF	OFF	OFF	ON	ON	
13	1	X	1	1	0	1	1	ON	OFF	OFF	ON	OFF	ON	ON	
14	1	X	1	1	1	0	1	OFF	OFF	OFF	ON	ON	ON	ON	
15	1	X	1	1	1	1	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	1	0	0	0	0	0	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	0	X	X	X	X	X	1	ON	ON	ON	ON	ON	ON	ON	4

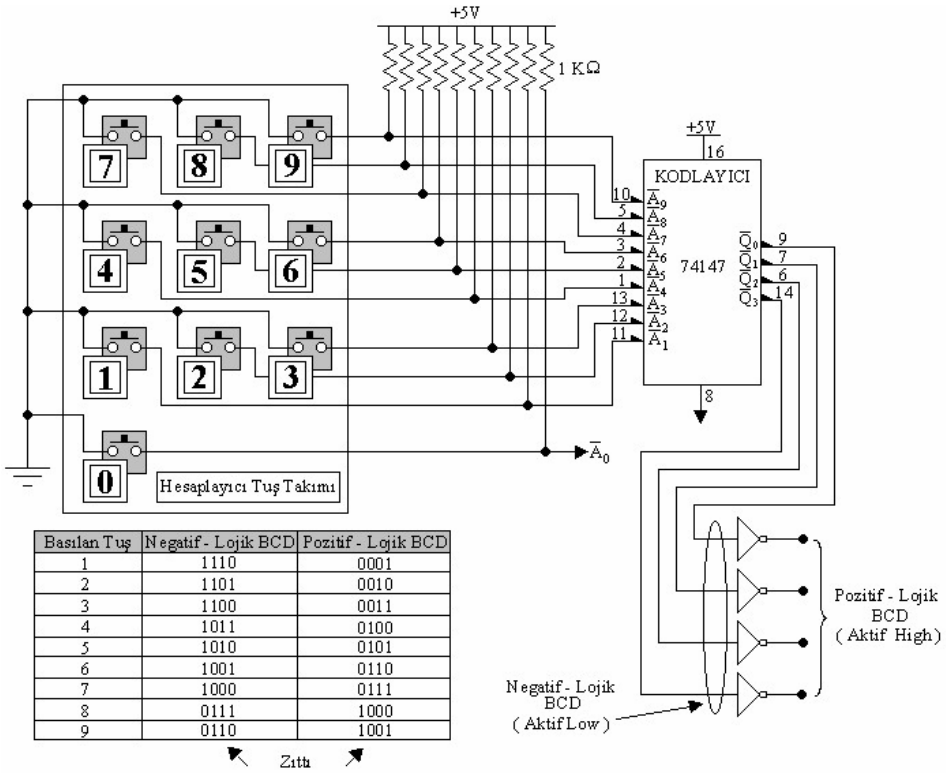
(c)

Şekil 8.18. Yedi parçalı gösterge kod çözücü devresi

8.2.4.2. 0-9 Tuş Takımı Kodlayıcı Devresi

Kodlayıcı devreler, hesap makinası tuş takımı ile diğer birimler arasında kullanılabilir (Şekil 8.19). Tuş takımı kodlayıcı devresinin girişleri, 0-9 arasındaki 10 adet tuşa bağlı anahtarların durumlarına göre değişir. Şekil 8.19'daki devrede anahtarların çıkışlarına bağlı olan pull-up dirençleri, 74147 kodlayıcı entegresi girişlerinin normalde '1' durumunda olmasını sağlar.

Tuş takımındaki herhangi bir tuşa basılması ile, tuşa bağlı olan anahtar kapanır ve ilgili giriş '0' seviyesine gelir. Girişi '0' olan entegre girişi aktif olur. Şekil 8.19'daki tabloda, farklı giriş durumlarında oluşan çıkış değerleri verilmektedir. Örneğin; 2 nolu anahtara basılması durumunda, entegrenin A2 girişi aktif olur ve çıkışta '1101' değerleri elde edilir. Negatif lojik mantığına göre oluşan bu değerlerin değil kapılarına uygulanması ile '0010' değerleri bulunur.



Basılan Tuş	Negatif - Lojik BCD	Pozitif - Lojik BCD
1	1110	0001
2	1101	0010
3	1100	0011
4	1011	0100
5	1010	0101
6	1001	0110
7	1000	0111
8	0111	1000
9	0110	1001

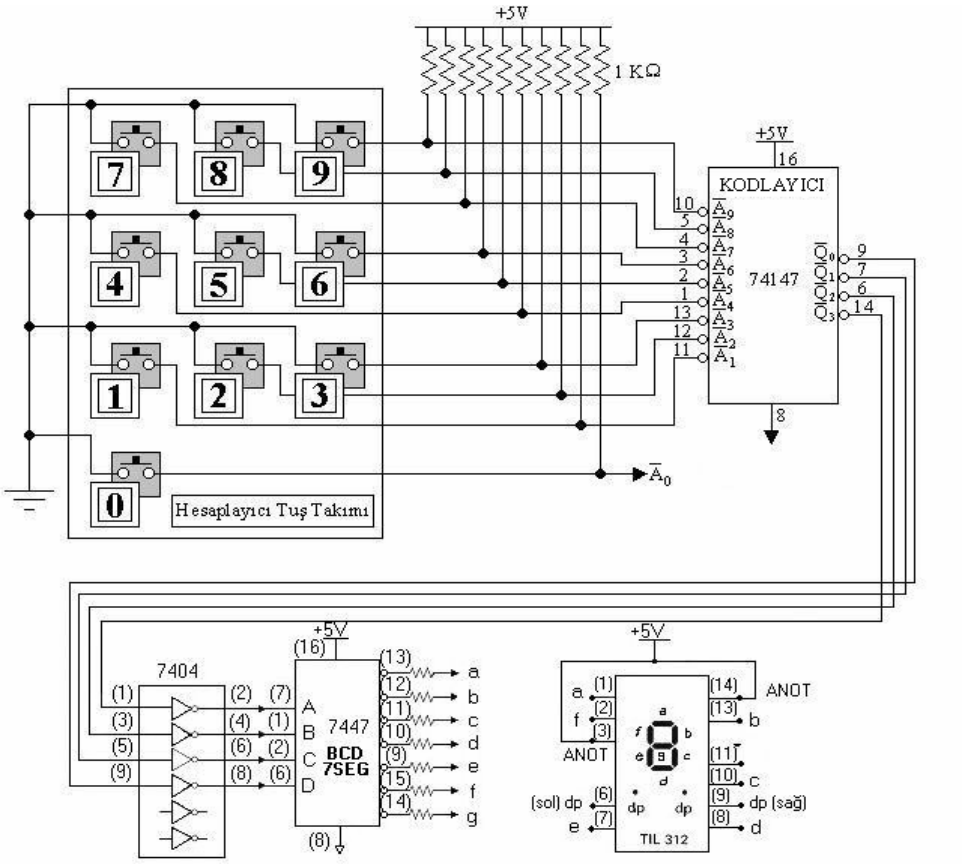
Şekil 8.19. Tuş takımı kodlayıcı devresi lojik şeması.

Tuş takımındaki anahtarların hiçbirisine basılmaması durumunda tüm girişler '1' olur ve girişlerden hiçbirisi aktif olmayan entegrenin çıkışlarında '1111' değerleri oluşur. Negatif lojiğe göre oluşan bu değerlerin tersinin alınması ile '0000' çıkış değerleri bulunur.

8.2.4.3. Kodlayıcı ve Kod Çevirici Devresi

Kodlama ile ilgili devrelere örnek olarak verilen devrelerin birleştirilmesi ile aynı devrede kodlayıcı ve kod çevirici devresi birleştirilmiş olur (Şekil 8.20).

Kodlayıcı devresi tuş takımındaki desimal değerler, 74147 entegresi ile BCD'ye dönüştürülür. 7404 entegresinde bulunan 'DEĞİL' kapıları ile tersleri alınan BCD kodlu değer, 7447 entegresine (BCD'den yedi parçalı entegreye kod çevirici) uygulanır. Kod çevirici devresi, BCD girişlerdeki değerlere bağlı olarak yedi parçalı göstergede sayılar oluşturur. Kod çevirici çıkışındaki değerler aktif '0' olduğundan, ortak anotlu yedi parçalı göstergeleri sürmek için kullanılabilir (Şekil 8.20). Şekildeki devrede bir tuşa basılması durumunda, basılan tuşun temsil ettiği desimal değer yedi parçalı göstergede okunur.

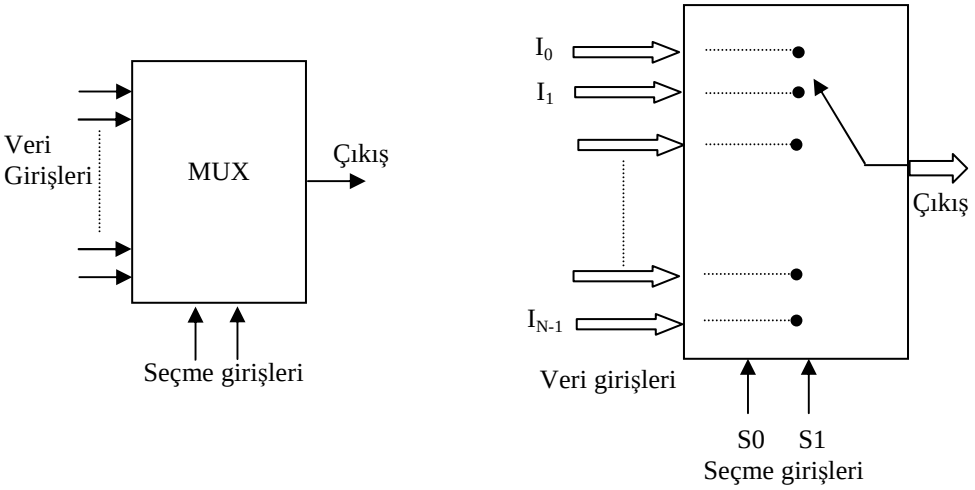


Şekil 8.20. Kodlayıcı ve kod çevirici devresi lojik şeması.

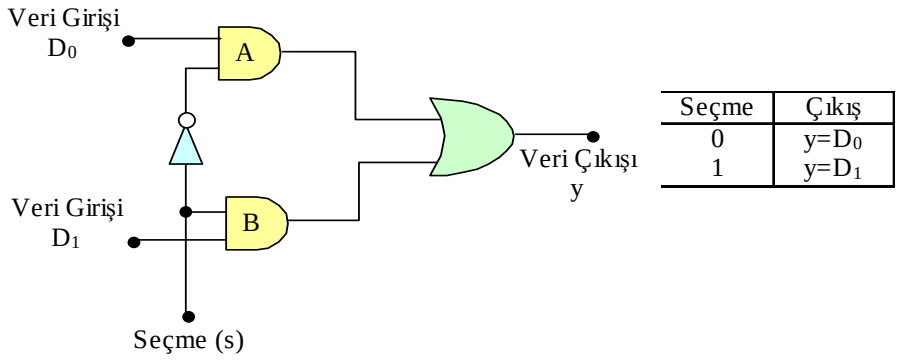
8.3. Çoklayıcılar - Veri Seçiciler (Multiplexers - Data Selectors)

Çok sayıdaki giriş bilgisinin zaman paylaşımlı olarak sırayla çıkışa aktarılması olayı, **'multiplexing - veri seçme / çoklama'** olarak tanımlanır.

Bir çok giriş hattından gelen bilgilerden birisini seçerek uygun çıkış hattına yönlendirilmesini sağlayan bileşik devrelere **'çoklayıcı / veri seçici devreler'** (multiplexer) denir ve ÇOĞ (MUX) sembolü ile gösterilir. Birçok veri transferi, zaman paylaşım tekniği kullanılarak multiplekser devreleri yardımıyla gerçekleştirilir. Çoklayıcılar, orijinal isminden hareketle çoğu kere 'multiplexer-multipleksır' olarak adlandırılır. Kitapta üç isim beraber kullanılacaktır.



(a) **b)**



Seçme	Çıkış
0	$y=D_0$
1	$y=D_1$

(c)

Şekil 8.21. Çoklayıcı sembolü, fonksiyon şeması ve 2x1 MUX lojik şeması.

Şekil 8.21.a'da sembolü ve Şekil 8.21.b'de fonksiyon şeması görünen veri seçici devresinde girişteki bilgilerden uygun olanının seçilmesi işlemi seçme girişleri (select inputs) ile yapılır. Veri seçicilerde, 2^n sayıdaki giriş hattından uygun olanı seçmek için 'n' sayıda seçme hattına ihtiyaç vardır. Dijital olarak kontrol edilebilen çok pozisyonlu anahtar gibi işlem yapan veri seçiciler, seçme hattının girişlerindeki değere göre çıkışa aktarılacak giriş hattına karar verir (Şekil 8.21.b).

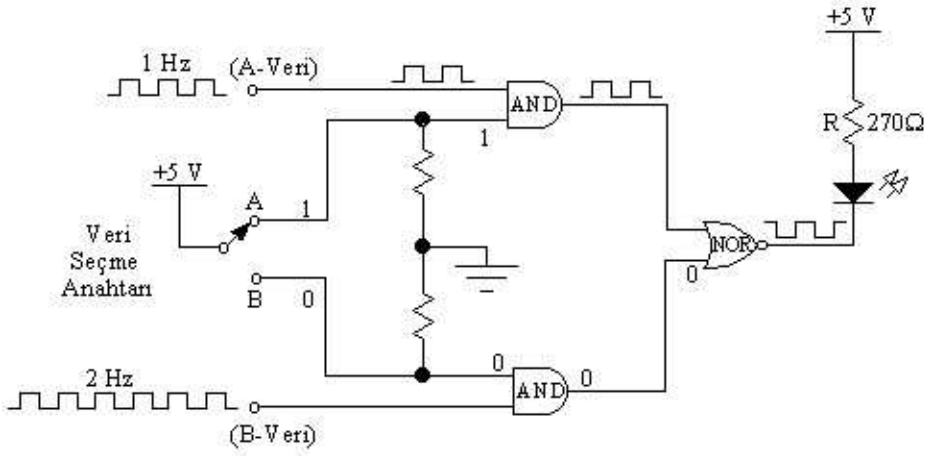
Şekil 8.21.c'deki devrede, girişlerden birisi seçme girişi yardımıyla seçilerek çıkışa aktarılır. İki girişten birisini seçerek çıkışa aktaran bu devre 2x1 MUX olarak isimlendirilir.

Veri seçme işlemini açıklamak için, iki farklı kaynaktan gelen bilgiye göre çıkışındaki bir LED'in farklı hızlarda yanıp sönmesini sağlayacak Şekil 8.22'deki lojik devreyi inceleyelim.

Şekil 8.22'deki devrede bulunan iki girişten çıkışa aktarılmak istenilen bir anahtar yardımı ile seçilir ve seçilen 'VE' kapısının ikinci girişine uygulanan bilgi ilgili 'VE' kapısı çıkışında görülür.

'VE' kapılarının çıkışlarının bağlı olduğu 'VEDEĞİL' kapısının girişlerinden birisinin '1' olması ile çıkışı '0' olur ve Anot'u +5V'a bağlı LED yanar. Girişlerden her ikisinin '0' olması durumunda ise 'VEDEĞİL' kapısının çıkışı '1' olur ve katoduna '1' gelen LED yanmaz.

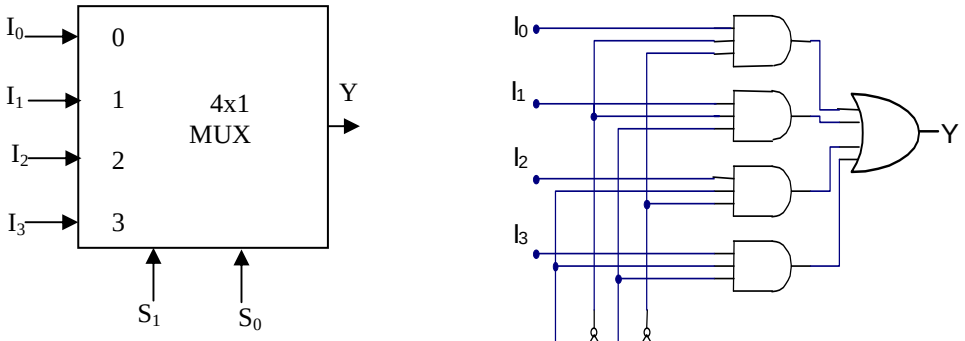
LED'in yanıp/sönme hızı, girişlerden uygulanan sinyalin frekansı ile doğrudan ilişkili olarak değişir. Seçme anahtarı ile yapılan işlem, çok veriden/bilgiden istenileni seçmedir.



Şekil 8.22. Kapı devreleri ile gerçekleştirilen veri seçme işlemi.

Şekil 8.23'de blok şeması ve lojik devresi verilen dört girişli multiplexer devresinde; S_0, S_1 girişlerinin kombinasyonuna göre girişlerden birisi çıkışa aktarılır. Diğer bir değişle, her farklı seçme kombinasyonunda bir giriş çıkışta gözükür. Örneğin, I_0 girişi S_0=0, S_1=0 kombinasyonu sonucu çıkışa aktarılırken, I_2 girişi S_0=0, S_1=1 kombinasyonu sonucu çıkışta görülür. 4x1 MUX olarak isimlendirilen bu devreye benzer şekilde iki, sekiz ve onaltı girişli multiplexer devreleri TTL ve CMOS entegre olarak piyasada bulunmaktadır.

Birden fazla çoklayıcı içeren entegrelerde, elemanın çalışmasını kontrol eden bir yetkilendirme girişi (E) ile çıkışta normal ve terslenmiş çıkışların oluşmasını sağlayan kontrol girişleri bulunabilir. Şekil 8.24'de görülen 8*1 MUX devresinde, E=0 olduğu zaman S_2, S_1, S_0 seçici girişleri veri girişlerinden birisini (I_0-I_7), 'Y' çıkışına göndermek üzere seçer. E=1 olduğunda ise, seçici girişlerin durumlarına bakılmaksızın Y=0 değerini alır.

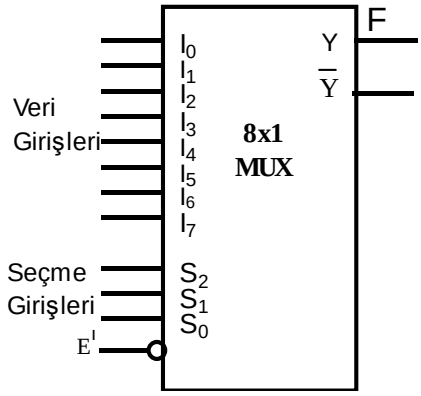


S_1	S_0	Y
0	0	0
0	1	1
1	0	2
1	1	3

Şekil 8.23. 4x1 MUX sembolü, fonksiyon tablosu ve lojik devresi.

Multipleksırları paralel bağlayarak giriş sayısını artırmak mümkündür. Şekil 8.25'de 8 girişli iki multipleksır ile 16 girişli multipleksır oluşturulması görülmektedir. Bu bağlantıda, iki multipleksırın çıkışı birleştirilerek tek çıkış haline getirilmiştir. S_3, S_2, S_1, S_0 seçici girişleri ile, 16 veri girişinden birisinin çıkışa aktarılması mümkündür.

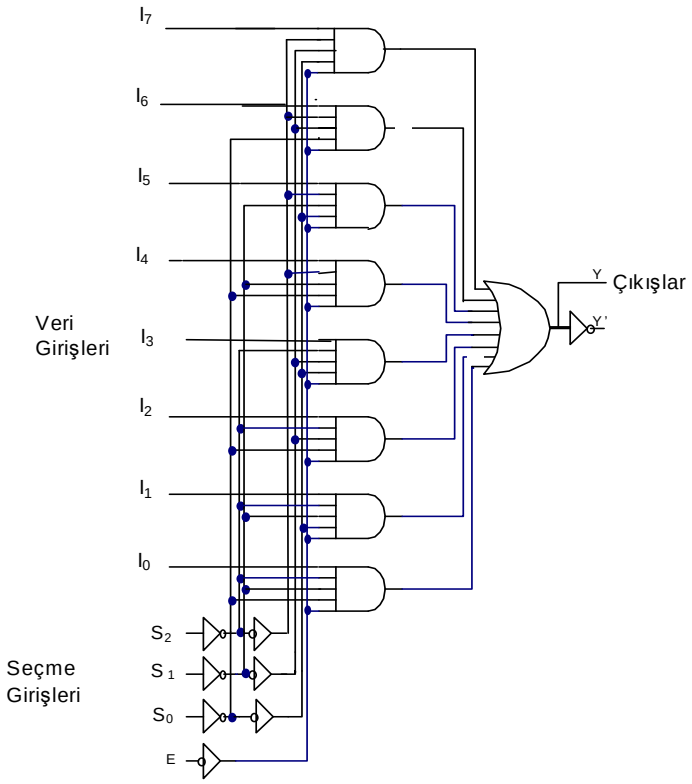
S_3 seçici girişi multipleksırlerden hangisinin aktif olacağına karar verir. S_3 = 0 olduğu zaman, üstteki multipleksır yetkilendirilir. S_2, S_1, S_0 seçici girişleri, üstteki multipleksır girişlerinden hangisinin Q çıkışında görüleceğini belirler. S = 1 durumunda ise, alttaki multipleksır yetkilendirilir ve S_2, S_1, S_0 seçici girişleri ile alttaki multipleksırın veri girişlerinden birisinin Q çıkışında görülmesi sağlanır.



a) 8x1 MUX Lojik sembolü

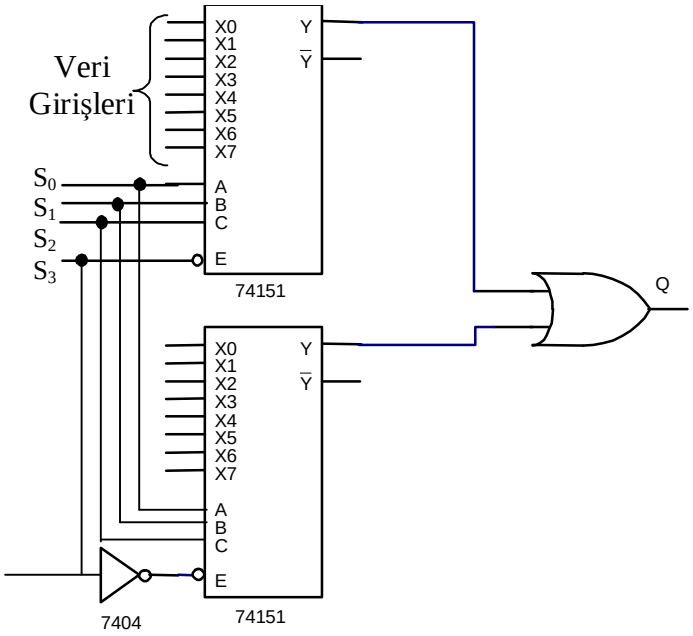
GİRİŞLER				ÇIKIŞLAR	
E'	S_2	S_1	S_0	Y'	Y
1	X	X	X	1	0
0	0	0	0	I_0'	I_0
0	0	0	1	I_1'	I_1
0	0	1	0	I_2'	I_2
0	0	1	1	I_3'	I_3
0	1	0	0	I_4'	I_4
0	1	0	1	I_5'	I_5
0	1	1	0	I_6'	I_6
0	1	1	1	I_7'	I_7

b) 8.1 MUX Doğruluk tablosu



c) 8x1 MUX Lojik Devresi

Şekil 8.24. 8x1 MUX lojik sembolü, doğruluk tablosu ve lojik devresi.

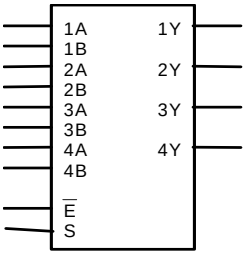


Şekil 8.25. İki adet 8 girişli çoklayıcı ile 16x1 çoklayıcı yapımı.

Aynı entegrenin içerisinde iki veya daha fazla sayıda çoklayıcı bulunduğunda, çoklayıcılara ait seçme ve yetkilendirme girişleri çoklayıcının tamamı için ortak kullanılabilir.

74157 entegresinde dört adet iki girişli çoklayıcı bulunur. Bu çoklayıcılar iki adet 4 girişli çoklayıcı olarak kullanılabilirler. Entegrenin 4 girişli çoklayıcı olarak kullanılması durumunda, çoklayıcının tamamındaki iki hattan birisini seçmek için bir adet ‘S’ seçme hattı yeterli olur (Şekil 8.26). E' yetkilendirme girişi çoklayıcıyı E=0 olduğu zaman yetkilendirir. Çoklayıcıların yetkilendirilmesi ve S=0 olması ile A girişlerinden birisi çıkışa bağlanır. S=1 olduğunda ise, B girişlerinden birisi çıkışa ulaşır. Bu durumda devrenin lojik sembolü ve doğruluk tablosu Şekil 8.26’daki gibi çizilebilir.

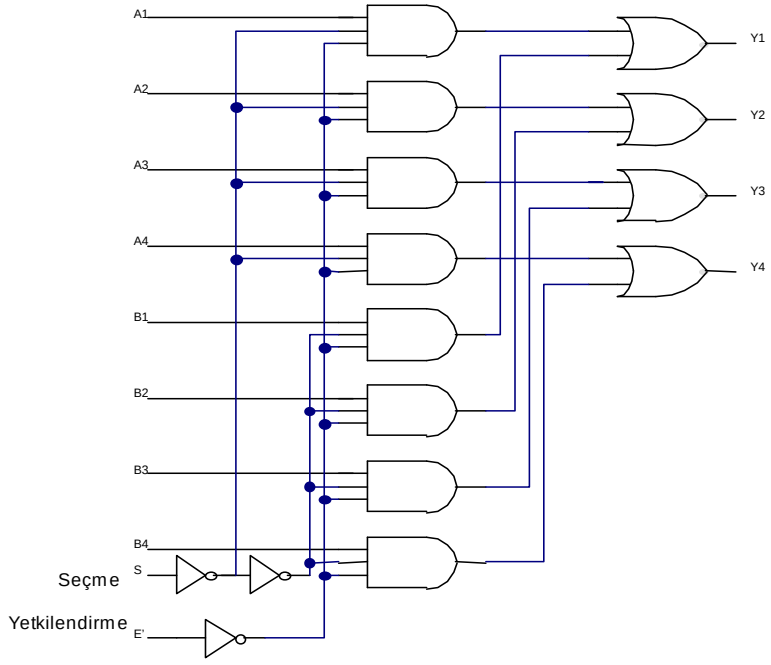
Daha önceki kısımlarda bahsedildiği üzere, çoklayıcı devrelerinin çalışma şekli kod çözücü devrelere çok benzer. Bu nedenle, kod çözücü devreler çoklayıcı olarak kullanılabilir. Kod çözücü devrenin çıkışı, çoklayıcı giriş hatlarıyla kontrol edilebilir. Gerçekleştirilen fonksiyona dahil edilecek kombinasyonlar ilgili giriş hatları 1’e eşitlenerek seçilirken, fonksiyona dahil edilmeyen kombinasyonlar giriş hatları 0’a eşitlenerek yetkisizlendirilir.



74157 MUX

E'	**S**	**Y_1**	**Y_2**	**Y_3**	**Y_4**
1	X	0	0	0	0
0	0	A_1	A_2	A_3	A_4
0	1	B_1	B_2	B_3	B_4

Doğruluk Tablosu



Şekil 8.26. 74157 entegresi ile yapılan 4 girişli multiplexerin sembolü, doğruluk tablosu ve lojik şeması.

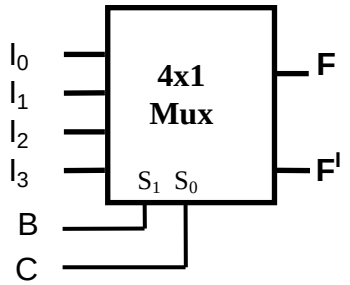
8.3.1. Çoklayıcı Uygulamaları

Çoklayıcı devreleri, çeşitli dijital sistemlerde farklı uygulama alanlarında kullanılırlar. Örnek uygulama alanları; veri yönlendirme, işlem sıralama, paralel seri veri dönüşümü, lojik bir fonksiyon üretimi, vb. şeklinde sıralanabilir. Bu uygulama alanlarından bir kısmına örnekler verelim.

8.3.1.1. Boolean Fonksiyonlarının ve Bileşik Devrelerin Gerçekleştirilmesi

'n' değişkenli herhangi bir fonksiyonu, 2^{n-1} giriş / 1 çıkışlı çoklayıcı ile gerçekleştirmek mümkündür. Boolean fonksiyonun çoklayıcı ile gerçekleştirilmesinde, n+1 değişkenli bir fonksiyonun değişkenlerinden 'n' tanesi bir çoklayıcının seçme hatlarına bağlanırken, kalan tek değişken çoklayıcı girişleri için kullanılır. Örneğin, ABC üç değişkenli bir Boolean fonksiyonunda A değişkenini tek kalan değişken kabul edilip veri girişleri için kullanılırsa, çoklayıcı girişleri A, A', 1, 0 değerlerinden birini alır. Bu dört değerin çoklayıcı girişlerine, diğer değişkenlerin (B,C) seçici girişlere uygulanması ile, Boolean fonksiyonun çoklayıcı ile gerçekleştirilmesi mümkün olur (Şekil 8.27).

Verilen bir Boolean eşitliğini çoklayıcı yardımıyla gerçekleştirmek için, çoklayıcı uygulama tablolarının oluşturulması gerekir. Fonksiyonun uygulama tablolarının oluşturulması ile, çoklayıcının girişlerine uygulanacak bilgiler bulunur.



Şekil 8.27. Boolean fonksiyonun çoklayıcı ile gerçekleştirilmesi.

Uygulama tablosu oluşturma işleminde takip edilecek sırayı, örnek bir uygulama ile birlikte açıklayalım. Açıklama sırasında, üç değişkenli bir fonksiyonda en yüksek basamak değerini ifade eden A değişkeninin giriş olarak kullanıldığını kabul edeceğiz.

$F_{(A,B,C)}=\Sigma(1,3,5,6)$ fonksiyonunun 4x1 çoklayıcı ile gerçekleştirilmesine yardım edecek uygulama tablosunu çıkaralım:

i- Boolean fonksiyonunun ifade ettiği değerler doğruluk tablosunda gösterilir. Doğruluk tablosunda kombinasyonlar sıralanarak, çıkışta '1' oluşması istenen kombinasyonlar belirlenir (Şekil 8.28.a).

ii- Çoklayıcının girişlerinin isimleri yatay olarak yazılarak, altlarına iki sıra halinde

bütün mintermler sıralanır. Sıralamada, ilk satırda A=0 olan mintermler, ikinci satırda A=1 olan mintermler yer alır. İlk satır A', ikinci satır A olarak isimlendirilir (Şekil 8.28.b).

iii- Doğruluk tablosunda çıkışın '1' olduğu minterm değerleri daire içine alınır (Şekil 8.28.c).

iv- Her bir sütun ayrı ayrı incelenir. Bir sütundaki iki minterm daire içerisine alınmışsa ilgili çoklayıcı girişine '1', iki mintermde daire içerisine alınmamışsa ilgili çoklayıcı girişine '0' uygulanacağını gösteren işaretleme yapılır (Şekil 8.28.d).

iv- Değişkenin kendisinin temsil edildiği alt minterm (örneğimizde A) daire içerisine alınmışsa ilgili çoklayıcı girişine A, değişkenin değilinin (A') temsil edildiği minterm daire içerisine alınmışsa ilgili çoklayıcı girişine A' uygulanacağını gösteren değerler verilir (Şekil 8.28.d). Böylece uygulama tablosu tamamlanır.

Multiplexer	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

a) Doğruluk Tablosu

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

(b)

	I_0	I_1	I_2	I_3
A'	0	(1)	2	(3)
A	4	(5)	(6)	7

(c)

	I_0	I_1	I_2	I_3
A'	0	(1)	2	(3)
A	4	(5)	(6)	7
	0	1	A	A'

(d)

Şekil 8.28. Çoklayıcılarda doğruluk ve uygulama tablolarının oluşturulması.

Elde edilen uygulama tablosunun 4x1 MUX devresine uygulanması ile, Şekil 8.29'deki devre oluşur. Oluşan devrede, B ve C değişkenleri seçme girişlerine uygulanırken, çoklayıcı girişleri I_0=0, I_1=1, I_2=A, I_3=A' şeklinde değer alır.

Çoklayıcı girişlerine uygulanması gerekli değeri bulduğumuza göre seçme girişlerinin farklı durumlarında oluşacak çıkış değerlerinin inceleyelim.

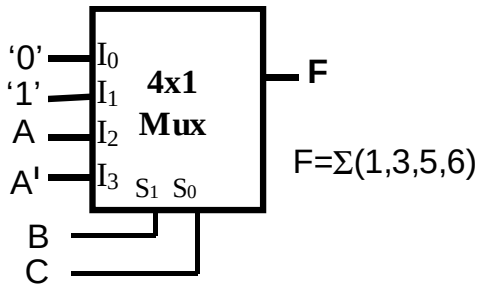
Seçici girişleri BC=00 durumunda iken, I_0 girişi seçilir ve I_0=0 olduğundan F=0'dır. Dolayısıyla iki kombinasyonda, yani m_0=A'B'C' ve m_4=AB'C' değerlerinde F=0 olur. Çünkü BC=00 iken, A değeri ne olursa olsun çıkış 0'dır.

BC=01 iken I_1 girişi seçilir ve I_1=1 olduğundan F=1 olur. Bunun anlamı, m_1=A'B'C ve m_5=AB'C kombinasyonlarında F=1 olmasıdır. Çünkü BC=01 iken, A değeri ne olursa olsun çıkış 1'dir.

BC=10 iken I_2 girişi seçilir ve bu girişe A bağlı olduğu için m_6= ABC' kombinasyonunda F=1 olur. Ancak m_2=A'BC' kombinasyonu için A=0 olacağından, F=0 değerini alır.

Son olarak, BC=11 olduğunda, I_3 girişi seçilir ve bu girişe A' bağlı olduğu için m_3=A'BC kombinasyonunda F=1 olur. Ancak m_7=ABC kombinasyonu için F=0 değerini alır. Bu bilgiler, Şekil 8.29'deki doğruluk tablosunda özetlenen bilgilerin analitik açıklamasından başka bir şey değildir.

Çoklayıcı girişlerindeki değişkenlerden en soldakini değil de başka bir değişkeni çoklayıcı girişi olarak kullanmak istiyorsak, uygulama tablosunda gerekli değişikliği yapmak şartıyla istediğimiz değişkeni çoklayıcı girişi için seçebiliriz.



Şekil 8.29. F = Σ(1,3,5,6) fonksiyonunun 4x1 MUX ile gerçekleştirilmesi.

Örnek 4: $F_{(ABC)}$ = Σ(1,3,5,6) fonksiyonunu 4x1 MUX ile gerçekleştireceğimizi ve S_1ve S_0 seçme girişleri için A ve B değişkenlerini, çoklayıcı girişleri için C değişkenini kullanacağımızı varsayalım.

C değişkeni çift sayılarda tümlenip tek sayılarda tümlenmeyeceğinden, uygulama tablosundaki iki minterm satırının düzenlenmesi Şekil 8.30.a'daki gibi oluşur.

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

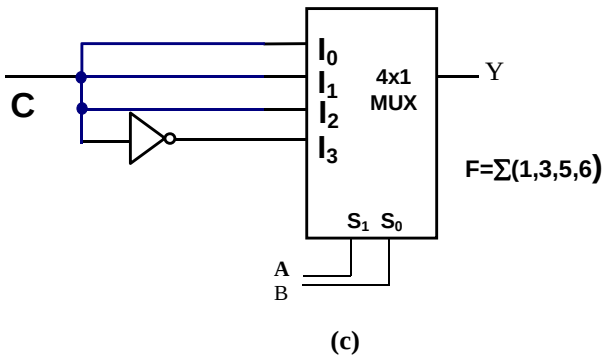
Uygulama tablosu

(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)

Şekil 8.30. 4x1 Çoklayıcı devre uygulaması.

F= ∑(1,3,5,6) fonksiyonunu, oluşan tabloya yukarıdaki kuralları göz önüne alarak uygularsak, Şekil 8.30.b'deki uygulama tablosu oluşur. Tablodan; I_0=C, I_1=C, I_2=C ve I_3=C' bağlantılarının yapılması gerektiği bulunur.

Uygulama tablosundan elde edilen çıkış bağlantılarının yapılması ile Şekil 8.30.c'deki çoklayıcı devre bağlantısı ortaya çıkar.

Örnek 5: $F_{(A,B,C,D)} = \sum(0,1,3,4,8,9,15)$ fonksiyonunu, 8x1 çoklayıcı ile gerçekleştirelim.

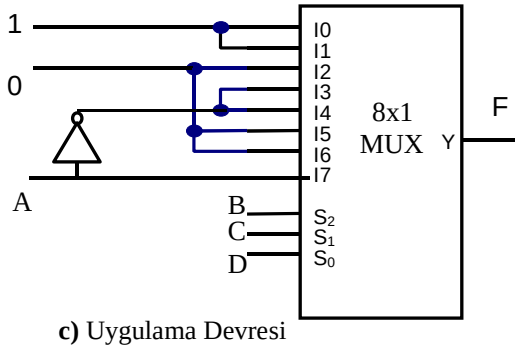
Dört değişkenli bir fonksiyon olduğundan, üç seçme hattı ve sekiz girişli bir çoklayıcıya ihtiyaç vardır. A değişkeni veri girişine, BCD değişkenleri seçme girişlerine uygulanırsa, Şekil 8.31.a'daki uygulama tablosu oluşur.

	I_0	**I_1**	**I_2**	**I_3**	**I_4**	**I_5**	**I_6**	**I_7**
A'	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15

a) Uygulama tablosu

	I_0	**I_1**	**I_2**	**I_3**	**I_4**	**I_5**	**I_6**	**I_7**
A'	(0)	(1)	2	(3)	(4)	5	6	7
A	(8)	(9)	10	11	12	13	14	(15)
	1	1	0	A'	A'	0	0	A

b) Uygulama tablosuna değerlerin yerleştirilmesi



c) Uygulama Devresi

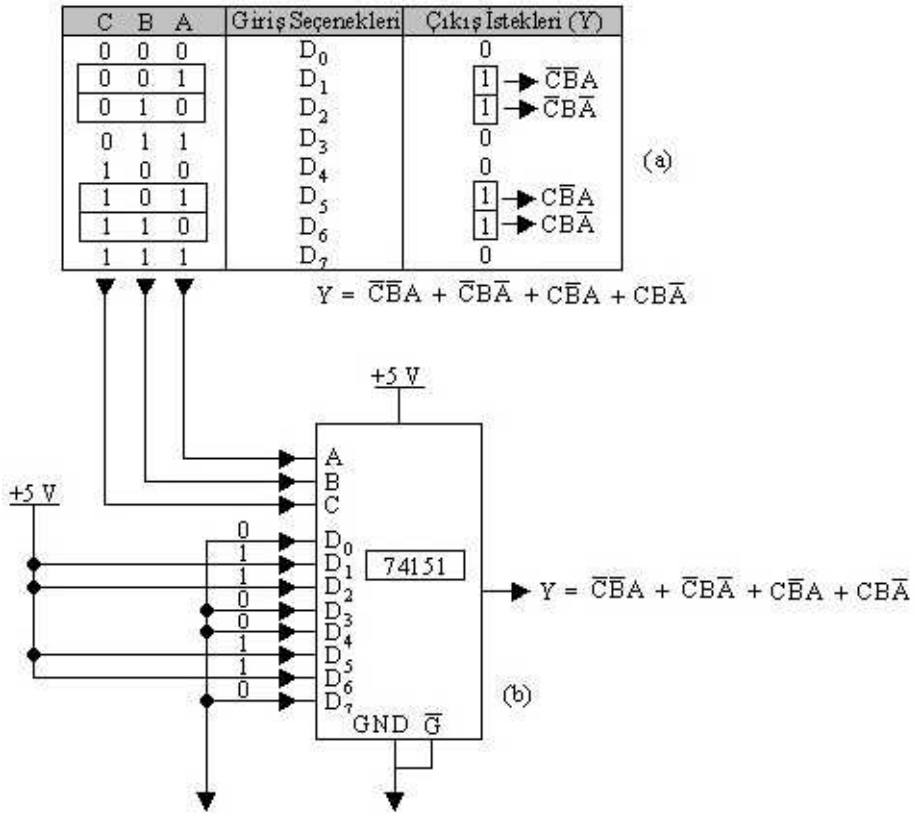
Şekil 8.31. Dört değişkenli boolean fonksiyonunun çoklayıcı ile gerçekleştirilmesi.

Oluşan uygulama tablosunun ilk satırının A', ikinci satırının A olduğunu göz önünde bulundurulur ve fonksiyondaki minterm değerleri uygulama tablosuna taşınırsa, Şekil 8.31.b'deki uygulama tablosu elde edilir.

Uygulama tablosundan elde edilen bilgilerin; 8x1 MUX'a uyarlanması ile Şekil 8.31.c'deki lojik devre oluşur.

Giriş değişken sayısınca seçme girişi bulunan çoklayıcı kullanılması durumunda tablo yapmaya gerek yoktur. Girişler doğrudan seçme girişlerine uygulanırken, çıkışın '1' olmasının istendiği kombinasyonlar +Vcc'ye, çıkışın '0' olmasının istendiği kombinasyonlar ise şaseye bağlanır (Şekil 8.32).

Şekil 8.32.'deki tabloda çıkışın '1' olmasının istendiği durumlar belirlenir. Girişler doğrudan seçme girişlerine uygulanırken, çıkışın '1' olmasının istendiği durumlar (D_1, D_2, D_5, D_6) +Vcc'ye, çıkışın '0' olmasının istendiği durumlar ise (D_0, D_3, D_4, D_7) şaseye bağlanır.



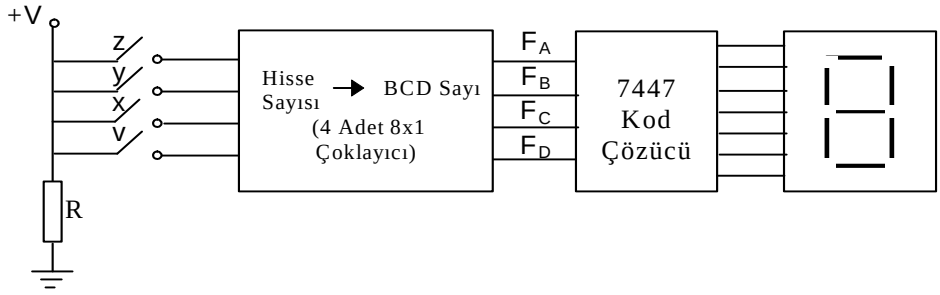
Şekil 8.32. Lojik eşitliğin çoklayıcı ile gerçekleştirilmesi.

Tasarım sırasında genel olarak lojik devre tasarımında kullanılan işlem sırası takip edilir. Tasarım aşamalarını örnek uygulama ile inceleyelim.

Örnek 6: Aşağıda tasarım özellikleri verilen lojik devreyi 74151 multipleksır entegresi kullanarak gerçekleştirelim.

Tasarım özellikleri : Küçük bir şirketin 10 hissesi bulunmakta ve her hisse, hissedarlar toplantısında hisse sahibine bir oy hakkı vermektedir. 10 hissenin 4 hissedar arasındaki paylaşımı; Hissedar A=1 hisse , Hissedar B=2 hisse, Hissedar C=3 hisse, Hissedar D=4 hisse şeklindedir. Hissedarların her birinin önünde, hisse sayısı oranını temsil eden ve 'evet' için kapanan, 'hayır' için açılan bir anahtar bulunmaktadır.

Yapılan oylamalarda, evet oyu veren toplam hisse sayısını gösteren bir devre tasarlanması istenmektedir. Bütün hissedarların karar için, evet oyu kullanması halinde ekranda '0' görülmesi yeterlidir. Gerekli sayıyı göstermek için, 7 parçalı gösterge ve kod çözücü kullanılacaktır. Hissedarların anahtarlarından gelen verileri 7447 kod çözücü entegresinde kullanılacak şekilde BCD'ye dönüştüren bileşik devreyi, dört adet 8x1 Multipleksır entegresi (74151) kullanarak gerçekleştirelim (Şekil 8.33).



Şekil 8.33. Multipleksır yardımı ile tasarlanması istenen bileşik devre blok şeması.

i- Her bir hissedarın temsil ettiği oy oranı farklı olduğundan, hissedarların evet dedikleri zamanki değerleri BCD'ye çeviren doğruluk tablosu oluşturulur (Şekil 8.34). Doğruluk tablosunda dört farklı devreyi ifade eden dört farklı çıkış sütunu bulunur.Doğruluk tablosunda giriş değişkenleri olarak; z, y, x, w harfleri, çıkış değişkenleri olarak; F_A, F_B, F_C, F_D sembolleri kullanılacaktır.

ii- Doğruluk tablosu çıkış sütunlarından her birisi ayrı bir uygulama tablosuna taşınır (Şekil 8.35).

iii- Uygulama tablosunda bulunan ve multipleksır girişlerinin temsil eden değerler ilgili multipleksır girişlerine taşınır.

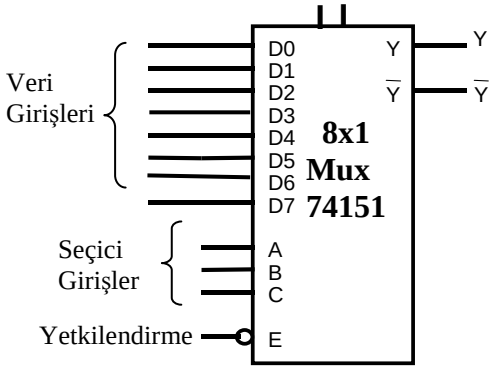
iv- Herbiri 7447'yi sürecek ikili bilgilerden birisini temsil eden multipleksır çıkışları elde edilir.

v- Uygulama tablosundaki verilerin multipleksır bağlantısında kullanılması ile oluşturulan multipleksır çıkışları 7447 kod çevirici entegreyi sürmek için kullanılır.

Dört adet Multipleksırın çıkışlarından alınan bilgiler (F_A, F_B, F_C, F_D) 7447 entegresinin girişlerine verilirse, oylama sonucu elde edilen ve BCD koduna dönüştürülen sayı yedi parçalı göstergede gözükür. Tüm hissedarların 'Evet' kullandığı durumda elde edilen onlu sayı sistemindeki '10' değerini göstermek için iki gösterge kullanılabilir.

	(4) Z	(3) Y	(2) X	(1) W	F_A	F_B	F_C	F_D
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	0	1	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	1	0
8	1	0	0	0	0	1	0	0
9	1	0	0	1	0	1	0	1
10	1	0	1	0	0	1	1	0
11	1	0	1	1	0	1	1	1
12	1	1	0	0	0	1	1	1
13	1	1	0	1	1	0	0	0
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	1

(a)



(b)

Şekil 8.34. Kod çevirici işlemin doğruluk tablosu ve 74151 entegresi lojik şeması.

8.3.1.2. Paralel- Seri Veri Dönüşümü

Sayısal sistemlerde bulunan birimler arasında veri iletimi genelde paralel olarak yapılır. Verilerin uzak mesafelerde iletiminde ise, paralel iletimin pahalı olması nedeni ile seri veri iletimi kullanılmaktadır. Bu durum, paralelden seriye veri dönüşümü ihtiyacını doğurmaktadır.

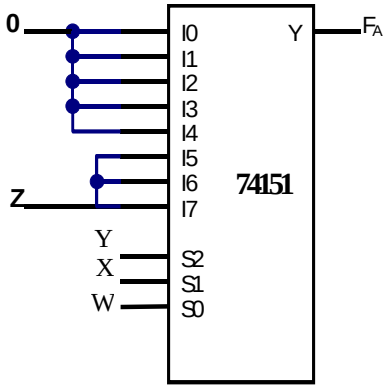
Paralelden seriye veri dönüşümünü gerçekleştirecek basit yöntemlerden birisi, multiplexer ve sayıcı devreleri kullanılarak gerçekleştirilen bileşik lojik devresidir (Şekil 8.36). Sekiz bitlik paralel-seri veri dönüşümüm için, paralel bilgiler bulunduğu birimden multiplexer'in girişlerine uygulanır. 0-7 arasında sayan ve ikili değerleri sırayla çıkış olarak veren sayıcı devresi çıkışı, multiplexerin seçme girişlerine uygulanır. Seçme girişlerindeki değerlere bağlı olarak, girişlerden birisindeki bilgi multiplexerin çıkışında gözükür. Girişler sırasıyla çıkışta gözükeceğinden, paralel bilgi seri bilgi şekline dönüştürülmüş olur (Şekil 8.36.b).

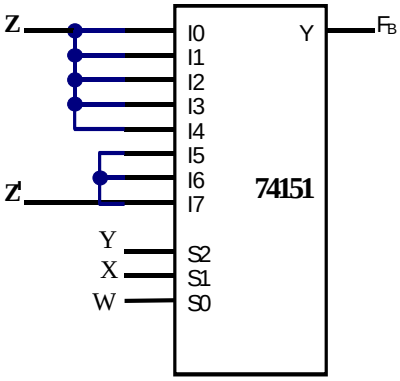
$F_{A(Z,Y,X,W)} = \sum(13,14,15)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Z'	0	1	2	3	4	5	6	7
Z	8	9	10	11	12	(13)	(14)	(15)
	0	0	0	0	0	Z	Z	Z

$F_{B(Z,Y,X,W)} = \sum(5,6,7,8,9,10,11,12)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Z'	0	1	2	3	4	(5)	(6)	(7)
Z	(8)	(9)	(10)	(11)	(12)	13	14	15
	Z	Z	Z	Z	Z	Z'	Z'	Z'



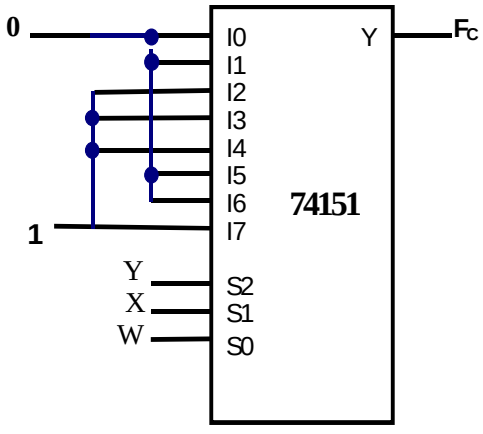


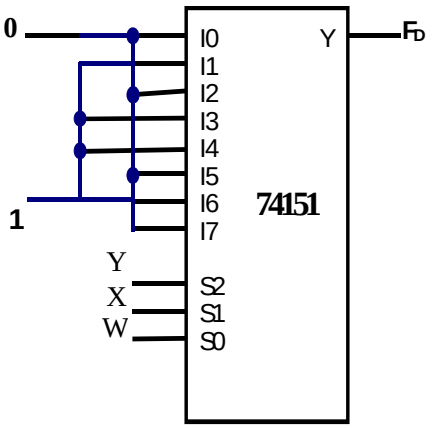
$F_{C(Z,Y,X,W)} = \sum(2,3,4,7,10,11,12,15)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Z'	0	1	(2)	(3)	(4)	5	6	(7)
Z	8	9	(10)	(11)	(12)	13	14	(15)
	0	0	1	1	1	0	0	1

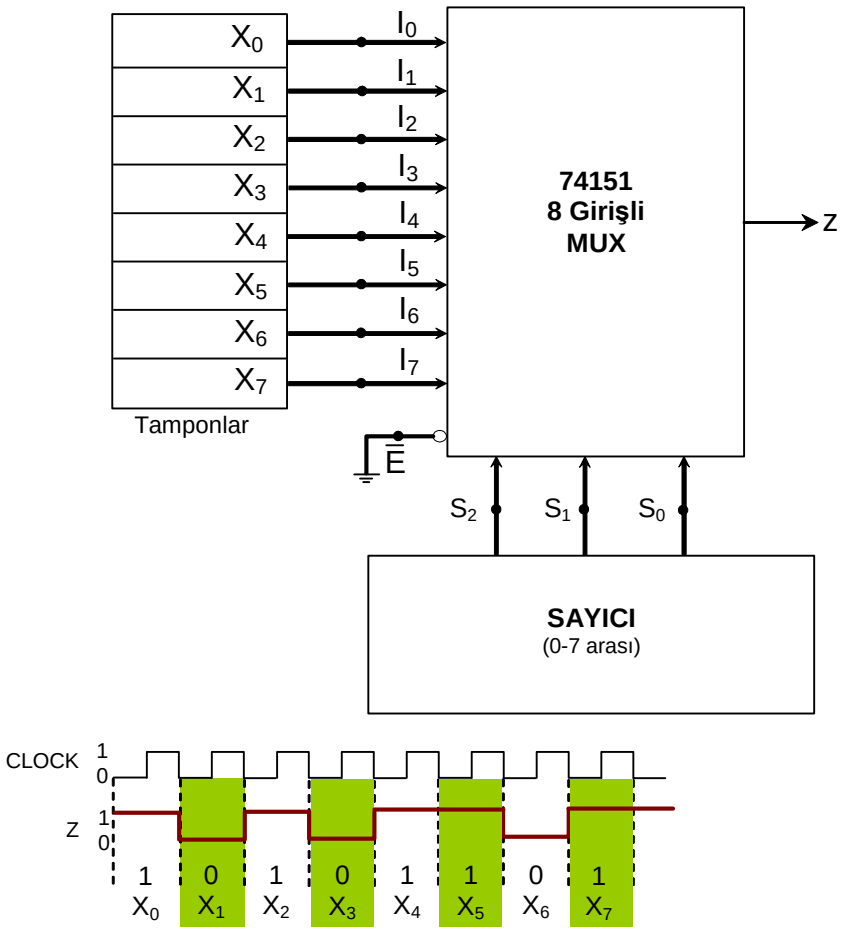
$F_{D(Z,Y,X,W)} = \sum(1,3,4,6,9,11,12,14)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
Z'	0	(1)	2	(3)	(4)	5	(6)	7
Z	8	(9)	10	(11)	(12)	13	(14)	15
	0	1	0	1	1	0	1	0





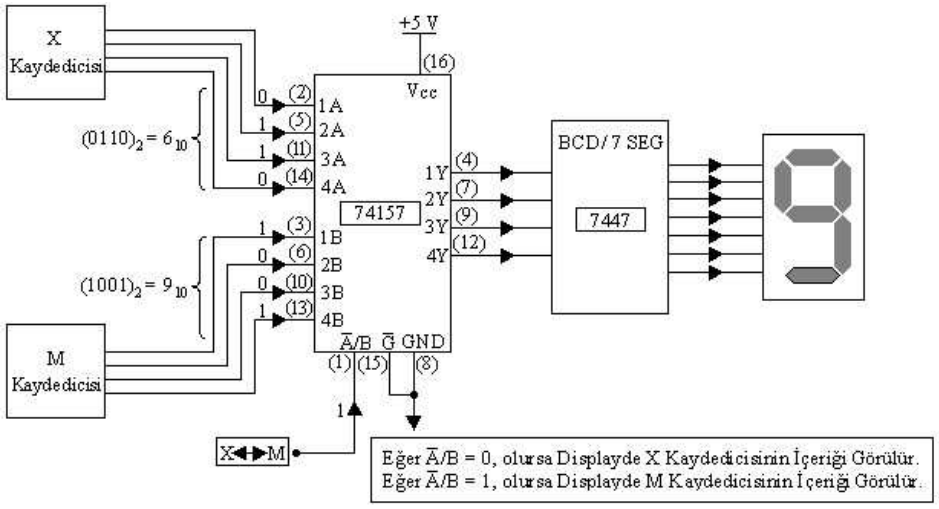
Şekil 8.35. Bileşik devrelerin multipleksır kullanılarak oluşturulması.



Şekil 8.36. Seri-paralel veri dönüşümü devresi ve “10110101” girişi için çıkış dalga şekilleri.

8.3.1.3.Veri Yönlendirme İşleminin Çoklayıcı ile Gerçekleştirilmesi

Multipleksırlar, birçok kaynaktan gelen veriyi tek bir hedefe doğru yönlendirebilirler. Şekil 8.37’de iki farklı kaynakta kaydedilen tek basamaklı onlu sayının tek bir göstergede görüntülenmesi için gerekli lojik bağlantı görülmektedir. Seçme girişinin durumuna göre A veya B grubundaki girişlerdeki bilgiler çıkışa aktarılır. Seçme girişindeki (A/B) değer ‘0’ ise X kaydedicisindeki değerler çıkışa aktarılırken, seçme girişinin ‘1’ yapılması durumunda B kaydedicisindeki değerler çıkışta gözükür. Multipleksır çıkışındaki değerler kod çevirici entegrede yedi parçalı göstergede gözükecek şekle dönüştürülür.



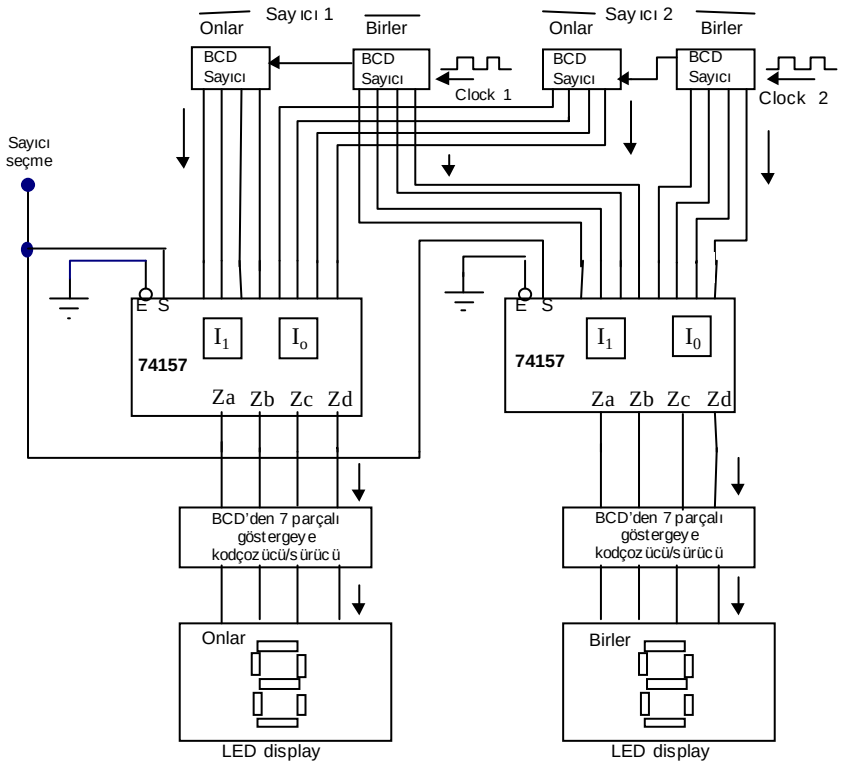
Şekil 8.37. Farklı sayıların aynı göstergede görüntülenmesi.

74157 Multipleksır entegresinin kullanıldığı yerlerden birisi, iki BCD sayıcının herhangi birisinin içeriğini, tekbir kod çözücü / sürücü ve LED gösterge seti kullanarak görüntülemektir. Multipleksırın veri yönlendirme amacıyla kullanıldığı iki basamaklı iki sayıcı devresinin yapısı, Şekil 8.38'de görülmektedir.

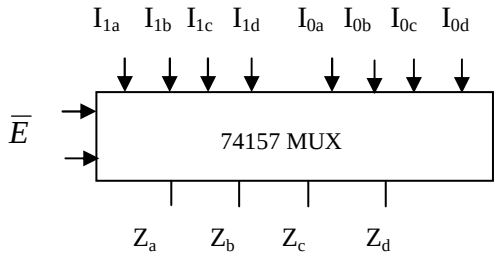
Şekil 8.38'deki devrede herbir sayıcı iki tane kaskat BCD kademesi içerir ve her biri kendi 'clock' sinyali ile sürülür. Sayıcı Seçme (Counter Select) hattı '1' olduğu zaman, 1 nolu sayıcının çıkışları multipleksırlar üzerinden geçerek kod çözücü / sürücü devrelere ulaşır ve göstergelerde gözükür. Bunun nedeni, kullanılan multipleksırın birbirinden bağımsız iki adet 4x1 MUX'a sahip olması ve seçme girişinin değerine göre bunlardan birinin aktif hale gelmesidir (Şekil 8.39). Sayıcı seçme girişi '0' olduğu zaman, 2 nolu sayıcı çıkışları multipleksırlar üzerinden geçer ve göstergelerde belirir.

Anlatılanlardan, sayıcılardan birisinin içeriğinin sayıcı seçme girişinin kontrolü altında göstergelerde okunacağı bulunur. Oluşan işlem, **'veri yönlerdirme (data routing)'** olarak isimlendirilir. Veri yönlendirme işleminin yaygın kullanılma yerlerinden birisi dijital saatlerdir.

Multipleksırın dijital saatlerde kullanılmasıyla çok sayıda farklı bilginin (saatler, aylar, günler v.b.) aynı göstergede gözükmesi mümkün olur. Bu kullanım; zaman paylaşımı şeklinde kullanışın yanı sıra, devrenin karmaşıklığı, hatların sayısı ve güç kullanımında avantaj sağlar.



Şekil 8.38. Multiplexer'in veri yönlerdirme amacıyla kullanılması.



E	**S**	$\mathbf{Z_a}$	$\mathbf{Z_b}$	$\mathbf{Z_c}$	$\mathbf{Z_d}$
1	X	0	0	0	0
0	0	I_{oa}	I_{ob}	I_{oc}	I_{od}
0	1	I_{1a}	I_{1b}	I_{1c}	I_{1d}

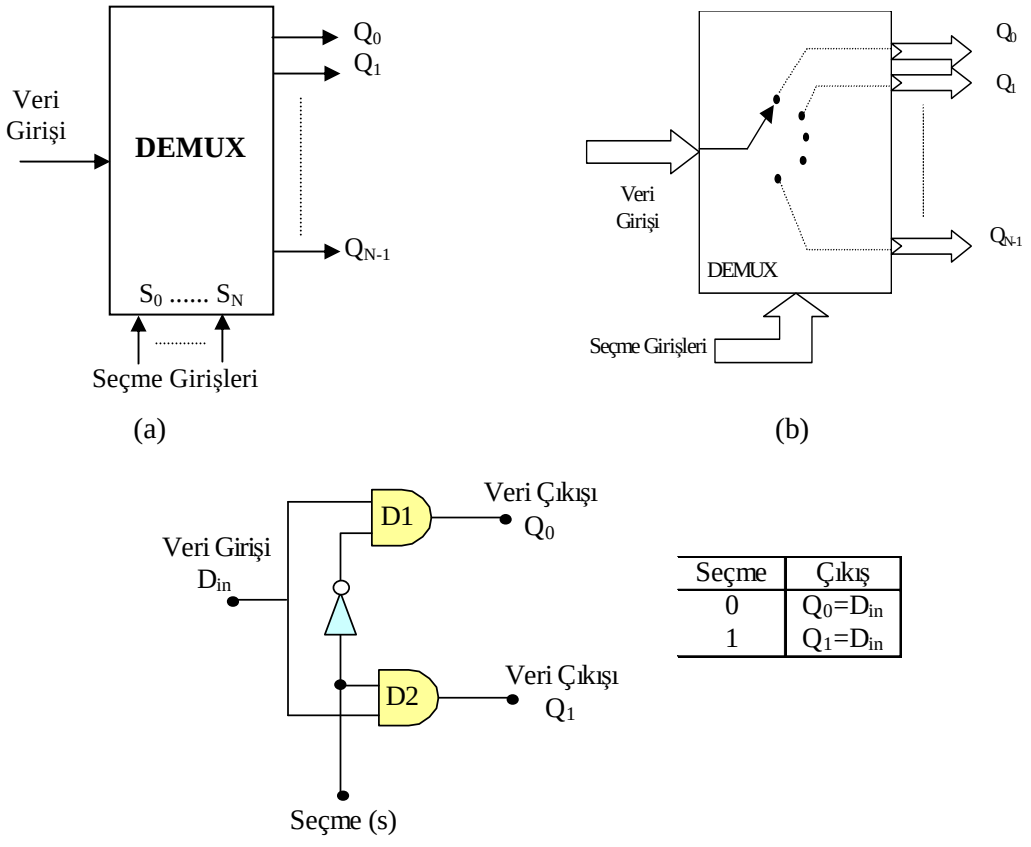
Şekil 8.39. Yetkilendirme ve seçme girişlerine sahip 74157 multipleksır entegresi blok şeması.

Paralel bilgilerin seriye çevrilmesi (Paralel-to-Serial Conversion) ve yapılacak işlemlerin sıralanması (Operation Sequencing), multipleksır devrelerin diğer kullanım alanlarıdır. Ancak bu konular Flip-Flop ve dijital-analog çevirici konularını içerdiğinden, ilgili konuların açıklamasından sonra anlatılabilir.

8.4. Azlayıcılar - Veri Dağıtıcılar (Demultiplexers - Data Distributors)

Tek bir girişten aldığı bilgileri, her bir çeşit giriş bilgisi farklı çıkışta olacak şekilde dağıtım yapan devrelere, **'Azlayıcı / Veri dağıtıcı devreler'** (Demultiplexer / Data Distributor) ismi verilir (Şekil 8.40.a).

Multiplexer'ın yaptığı işlemin tersini yapan bu devrede seçici girişlerin değeri, giriş verilerinin hangi çıkışa gönderileceğini belirler. Özet olarak; 'demultiplexer devresi, tek bir kaynaktan gelen bilgileri seçme girişleri yardımıyla ayırarak, N çıkış hattından birisine gönderen çok konumlu bir anahtardır' denebilir (Şekil 8.40.b).



Seçme	Çıkış
0	$Q_0=D_{in}$
1	$Q_1=D_{in}$

Şekil 8.40. Demultiplexerin sembolü ve fonksiyon şeması.

Şekil 8.40.c'deki devrede, tek bir girişten uygulanan bilgiyi seçme girişi yardımıyla iki çıkıştan birisine yönlendiren demultiplexer lojik şeması görülmektedir. Seçme girişinin '0' yapılması durumunda D_1 kapısı aktif olup veri girişindeki bilgiyi Q_0 çıkışına aktarırken, seçme girtişinin '1' olması durumunda veri girişindeki bilgi D_1 kapısı üzerinden Q_1 çıkışına aktarılır.

1x8 demultiplexer devresinde tek bir hattan gelen giriş verisi, seçme girişlerinin değerlerine göre 8 adet çıkıştan birisine yönlendirilir. Girişin hangi çıkışa doğru yönlendirileceğine, seçme girişlerinin kombinasyonuna göre karar verilir (Şekil 8.41).

Seçme kodu			**Çıkışlar**							
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	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

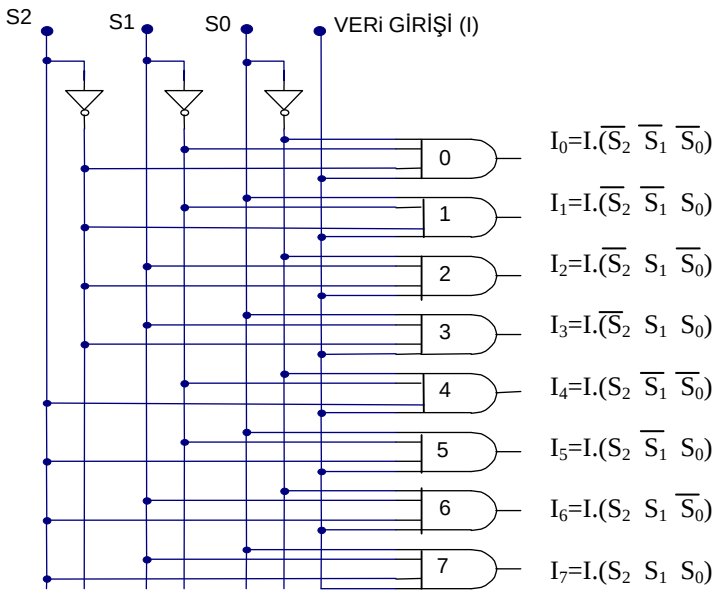
I: Giriş verisi

Şekil 8.41. 1'den 8 hatta demultiplexer devresi doğruluk tablosu.

Şekil 8.42'de, **'1 hattan 8 hatta demultipleksır'** olarak isimlendirilen ve 1 hattan gelen giriş verilerini, 8 farklı çıkıştan birisine yerleştiren demultiplexer devresi lojik şeması görülmektedir. Seçme girişlerinin her bir kombinasyonunda, 'VE' kapılarından birisi yetkilendirilir ve girişteki verinin çıkışa aktarılması sağlanır. Örneğin, $S_2S_1S_0$ = 000 durumunda yalnızca 0 nolu 'VE' kapısı yetkilendirilirken, $S_2S_1S_0$ =011'de 3 nolu 'VE' kapısı yetkilendirilir.

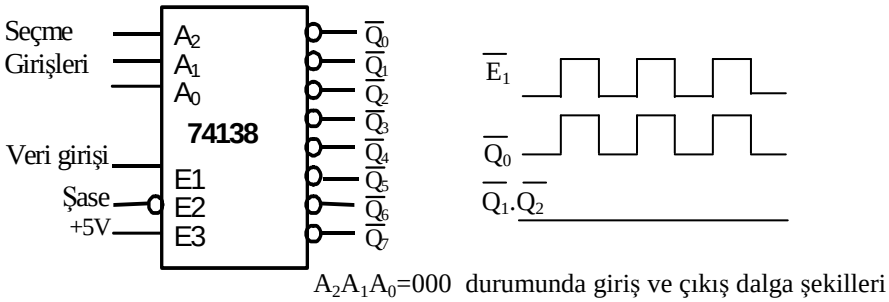
Şekil 8.42'de görülen demultiplexer devresi, 3 hattan 8 hatta kod çözücü devresine çok benzer. Yalnızca, kapılara 'I' giriş olarak dördüncü bir giriş eklenmiştir. Bununla beraber, birçok kod çözücü entegresi, kod çözücü devrelerine fazladan bir giriş olarak eklenen yetkilendirme girişine sahiptir. Bu durumda; kod çözücü devredeki yetkilendirme girişi demultiplexer'deki 'I' veri girişi ve kod çözücünün A, B, C girişleri de demultiplexer'deki seçici girişler olarak kullanılırsa, kod çözücü entegrelerin demultipleksır olarak kullanılabileceği sonucu ortaya çıkar.

Entegre devre üreticileri ürettikleri bu tip malzemeleri, **'kod çözücü / demultiplexer'** olarak adlandırmakta ve eleman kullanıcı tarafından isteğe uygun olarak kullanılmaktadır. Örneğin, 74LS138 entegresi hem 8'de 1 kod çözücü olarak, hem de 1x8 demultiplexer olarak kullanılabilir (Şekil 8.42). Entegredeki E_1 yetkilendirme girişi I veri girişi olarak kullanılırken, diğer iki yetkilendirme girişi aktif konum alacak şekilde bağlanır.



Şekil 8.42. 1'den 8 hatta demultipleksır lojik devresi.

Entegre demultiplexer olarak kullanılırken, $A_2A_1A_0$ girişleri seçme girişleri olarak kullanılır. Örneğin, seçme girişlerinin 000 olduğu bir durumda yalnızca Q_0 çıkışı aktif olurken, diğerleri '1' değerini alır. Q_0 çıkışı E_1'in '0' değerini almasıyla '0', E_1'in '1' olması durumunda '1' olur. Diğer bir deyişle, Q_0 çıkışı E_1'in aldığı değeri takip ederken (I girişinde olduğu gibi), diğer çıkışlar '1' konumunda bulunur. Aynı şekilde, $A_2A_1A_0$ seçici girişlerine farklı bir kombinasyon uygulandığında ilgili çıkış I veri girişinin aldığı değeri takip eder. Şekil 8.43'de, seçici girişlerin $A_2A_1A_0$=000 değerine sahip olması durumunda Q_0 çıkışının $I_{(E)}$ girişi ile ilişkisi görülmektedir.



$A_2A_1A_0$=000 durumunda giriş ve çıkış dalga şekilleri

Şekil 8.43. 8'de 1 kod çözücü devresinin demultiplexer olarak kullanılması ve örnek sinyal şekli.

8.4.1. Demultipleksır Uygulamaları

Demultiplexer devreleri, tek bir verinin farklı yerlerde kullanılmasını sağlayacak uygulamalar yanında, multiplexer ile birlikte sistemleri basitleştirmek amacıyla kullanılır. Bu kullanım alanlarını birer örnek ile detaylandıralım.

8.4.1.1. Tetikleme (Clock) Demultipleksır

Demultiplexer devresinin uygulama alanından birisi, tetikleme demultiplexer'dir. 74LS138 demultiplexer entegresiyle yapılabilen bu uygulamada, tekbir kaynaktan gelen tetikleme sinyali uygun olan çıkışa yönlendirilir. Örneğin, $A_2A_1A_0$=100 durumunda tetikleme sinyali Q_4'e doğru yönlendirilirken, başka bir seçme kombinasyonunda farklı bir çıkışa bağlı olan düzeneğe tetikleme sinyali sağlanır. Bu durumda, demultiplexer yardımıyla tek bir tetikleme sinyali çok sayıda düzenekte kullanılabilir.

8.4.1.2. Güvenlik Görüntüleme Sistemi

Bir fabrikada kullanılan güvenlik görüntüleme sisteminde, çok sayıda kapıya bağlı olarak çalışan açık / kapalı anahtarlar bulunmaktadır. Her bir kapı ile bir anahtarın durumu kontrol edilmekte ve anahtarın durumu LED'ler ile görüntülenmektedir. LED'ler güvenlik biriminin bulunduğu uzak bir noktada görüntüleme paneline yerleştirilmiştir. Bu devreyi lojik elemanlar yardımıyla gerçekleştirelim.

Şekil 8.44'de yapılması istenen, 8 kapının kullanıldığı güvenlik görüntüleme sistemi prensip şeması görülmektedir. Böyle bir uygulamada yapılacak işlem; her bir kapıdan gelen anahtardan, gösterge panelindeki ilgili LED'e bir hat çekmektir. Bu sistem çok sayıda kablo döşenmesini gerektirir. Uygulanması istenilen devrede bulunan iki birim arasında 8 adet kablo çekilmesi gerekmektedir.

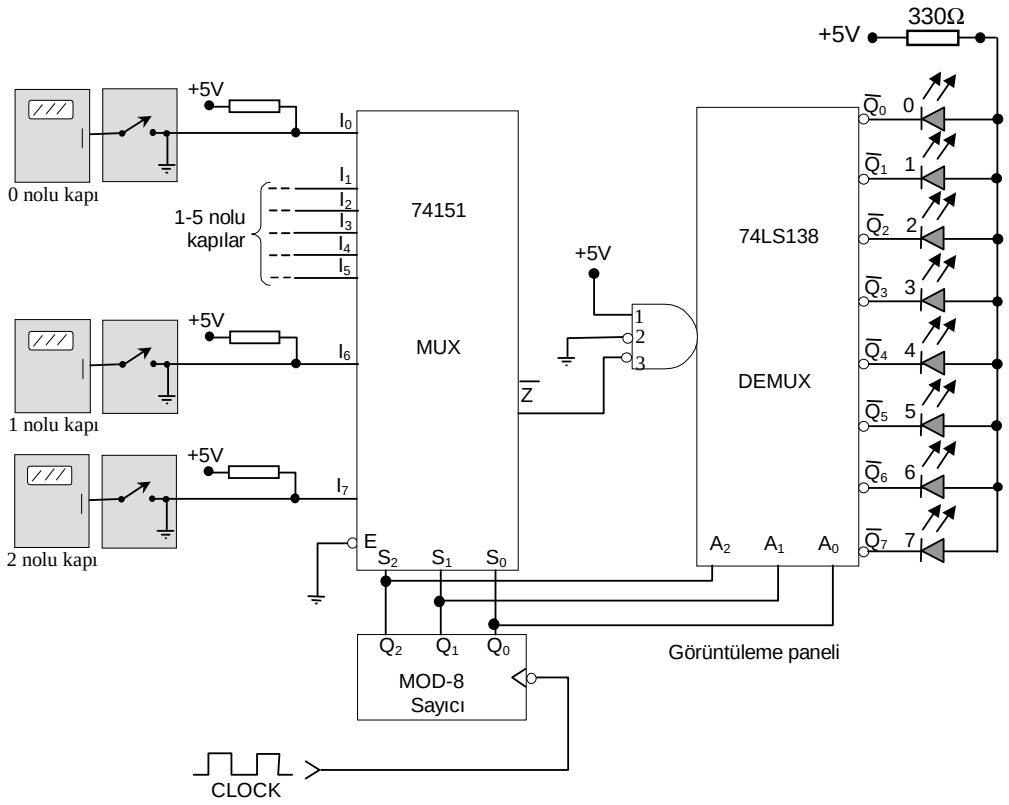
Bu sistemi gerçekleştirmenin diğer bir yöntemi, multiplexer / demultiplexer kombinasyonu kullanmaktır.

Prensip şeması çizilen devrede, sekiz kapıda bulunan anahtarlar multiplexer için veri girişleridir. Kapılar açık olduğu zaman '1' sinyali üretilirken, kapalı olduğu zaman '0' üretilir. Kapıların durumuna göre bu bilgiler multiplexer girişinde gözükür. Mod 8 sayıcı, hem multiplexer hem de demultiplexer için seçici giriş sinyalleri üretir. Multiplexer devresi, girişlere bağlı kapılardan gelen sinyallerden birisini çıkışına aktarır. Bu çıkıştan alınan sinyal demultiplexer'in 'I' girişine uygulanır.

MOD 8 sayıcı çıkışından demultiplexer seçici girişlerine uygulanan sinyaller, I girişinden gelen bilgiyi çıkışlara bağlı olan LED'lere gönderir. Diğer bir deyişle, seri şekilde I girişine gelen bilgiler seçici girişler yardımıyla uygun olan çıkışa aktarılır. Kapılardan birisinin açık olması durumunda, ilgili LED flaş yaparak yanıp-söner. Flaş süresi tetikleme sinyalinin frekansını değiştirmek suretiyle ayarlanabilir.

Yapılan uygulamada, iki uzak sistem arasında 8 hat çekilmesi yerine, multiplexer ve

demultiplexer kullanarak 4 hat ile sistem gerçekleştirilir. Demultiplexer devresi, verilerin eş zamanlı olarak transferini sağlayan ‘Eşzamanlı Veri İletme Sistemi’ (Synchronous Data Transmission System) olarak çalışabilir. Bu konu sayıcı ve kaymalı kaydedici konularını içerdiğinden burada açıklanmayacaktır.



Şekil 8.44. Multiplexer ve demultiplexer’in kullanıldığı güvenlik görüntüleme sistemi prensip şeması.

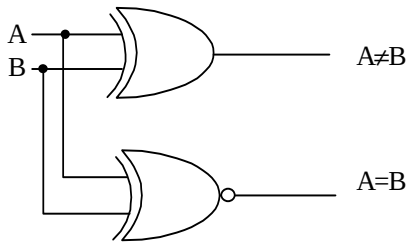
8.5. Karşılaştırıcı ve Aritmetik İşlem Devreleri (Arithmetic Logic Unit)

‘Karşılaştırıcı devreleri’, farklı kaynaklardan gelen bilgileri karşılaştırmak amacıyla düzenlenen devreler olarak düşünebilir. Bileşik lojikte en çok kullanılan devrelerden olan toplayıcı ve çıkarıcı devreler ise, ‘Aritmetik İşlem Devreleri’ olarak isimlendirilir. Karşılaştırıcı ve aritmetik işlem devreleri ‘Kıyaslama Devreleri’ veya ‘Aritmetik Mantık Birimi’ olarak tanımlanır. Diğer bir değişle, kıyaslama devreleri karşılaştırıcılar ve aritmetik lojik devrelerine genelde verilen bir isimdir.

8.5.1. Karşılaştırıcılar (Comparators)

İki sayıyı karşılaştıran ve büyüklüklerini belirleyen bileşik devreler, '**büyüklük karşılaştırıcı**' (magnitude comparator) olarak isimlendirilir. Karşılaştırma sonucu; A>B, A=B veya A<B'yi belirleyen üç konum ile belirlenir. En yaygın kullanım yerleri Aritmetik Lojik devrelerdir. Karşılaştırıcı devreleri, girişleri aynı veya farklı iken çıkış veren kontrol devrelerinde ve ikili karşılaştırmanın kullanıldığı adres bulma devrelerinde kullanılır.

En basit karşılaştırıcı devresi, tek bitlik A ve B sayılarının eşitlik durumunu karşılaştıran karşılaştırıcı devresidir. Bu devrede A=B durumunda çıkışlardan birisi '1' olurken, A≠B durumunda diğeri '1' olur (Şekil 8.45).



Girişler		Çıkışlar	
A	**B**	**A≠B**	**A=B**
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Şekil 8.45. Bir bitlik iki sayının karşılaştırması.

İki bitlik bilgiyi karşılaştıran ve A=B, A>B ve A<B çıkışlarını üreten devreyi tasarlayalım.

Devrenin doğruluk tablosu oluşturulur ve çıkışı temsil eden fonksiyonlar yazılırsa, Şekil 8.46.a'daki eşitlikler elde edilir. Elde edilen eşitlikleri temsil eden devrenin çizilmesi ile Şekil 8.46.b'deki lojik devre oluşur.

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

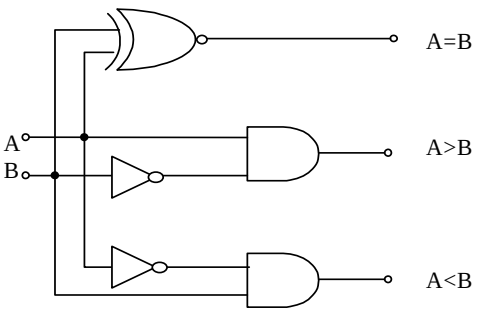
$A>B=AB^{I}$

$A=B=A^{I}B^{I}+AB$

$=A \odot B$

$A<B=A^{I}B$

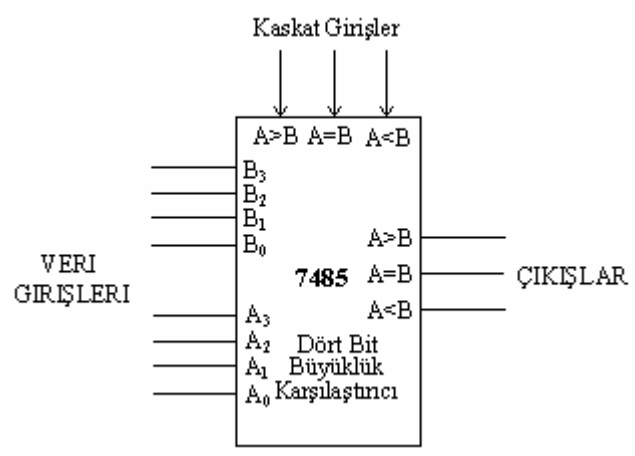
(a)



(b)

Şekil 8.46. Bir bitlik iki sayıyı karşılaştıran lojik devre tasarımı.

7485 entegresi, dört bitlik iki sayıyı karşılaştıran ve karşılaştırılan bitlerin durumuna göre çıkış oluşturan lojik elemandır (Şekil 8.47). Entegre, karşılaştırılacak sayı girişleri ile birlikte çok sayıda entegrenin bir arada kullanılmasına imkan tanıyan karşılaştırma girişlerine sahiptir. Entegrenin dört bitlik karşılaştırma işlemine ait doğruluk tablosu Şekil 8.48'da görülmektedir.



Şekil 8.47. 7485 karşılaştırıcı entegresi lojik sembolü.

Dört bitlik büyüklük karşılaştırıcı devresi, A girişlerine uygulanan $A_3A_2A_1A_0$ sayıları ile B girişlerine uygulanan $B_3B_2B_1B_0$ sayılarını karşılaştırır. Bütün bitlerde A_i ile B_i değerleri eşitse, iki sayı birbirine eşittir.

Sayıları karşılaştırma işleminde, önce A_3 ve B_3 bitlerini karşılaştırır. Eğer $A_3>B_3$ ise, diğer bitlerin karşılaştırmasına gerek yoktur. Eğer $A_3=B_3$ ise, daha düşük basamak değerine sahip iki bitin karşılaştırılmasına geçilir. Karşılaştırma işlemine, eşit olmayan bir basamak çiftine ulaşıncaya kadar devam edilir. A'nın ilgili hanesi '1've B'ninki '0' ise A>B sonucuna, A'nın ilgili hanesi '0' ve B'ninki '1' ise, A<B sonucuna varılır. Eğer bütün basamaklardaki değerler birbirine eşitse, A=B sonucuna ulaşılır. Karşılaştırma sonucunda varılan karara göre ilgili çıkış '1' yapılır (Şekil 8.48).

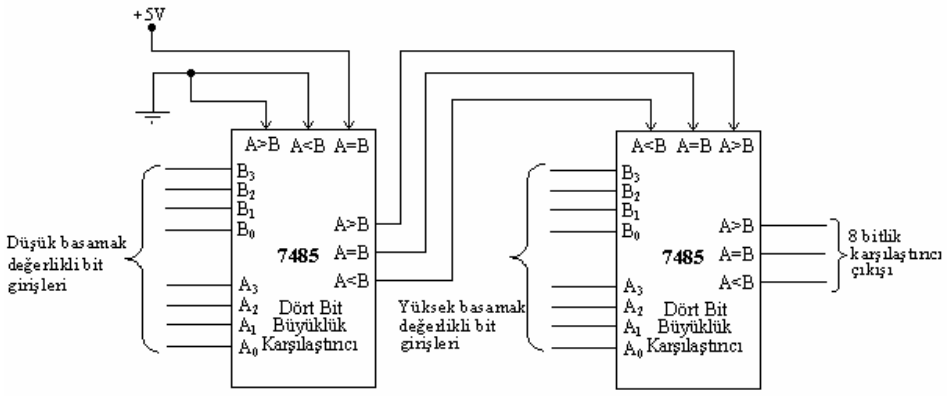
Dört bitten daha büyük sayıların karşılaştırılması için, iki veya daha fazla sayıda dört bitlik karşılaştırıcı kaskat şeklinde bağlanabilir. İki karıştırıcının kaskat şeklinde bağlanması durumunda, düşük basamaklı bitleri karşılaştıran devre çıkışları, yüksek basamakları karşılaştıran devrenin ilgili girişlerine Şekil 8.49'daki gibi bağlanır. Bu bağlantı ile, 8 bitlik iki sayının karşılaştırılması işlemi gerçekleştirilebilir.

Sekiz bitin karşılaştırılması işleminde, kaskat girişlerin değerlerine bakılmaksızın yüksek değerli dört bit karşılaştırılır. Yüksek değerlikli bitlerin eşit olması durumunda, düşük değerli dört biti karşılaştıran entegrenin çıkışının uygulandığı kaskat girişleri değerlendirilerek, 8 bitin karşılaştırılması sonucunu veren çıkışlarda karşılaştırma sonucu okunur.

Büyüklük karşılaştırıcılar, bilgisayarlarda (mikroişlemcili sistemlerde) adres kod çözücü devrelerin bir parçası olarak kullanılır. Bu kullanımda, bir karşılaştırıcı bilgisayarın merkezi işlem biriminin ürettiği adres kodu ile bellekte bulunan adresin kodunu karşılaştırır. Karşılaştırılan kodlar eşitse, istenilen işlemi yapmak üzere ilgili devreyi aktif hale getirir.

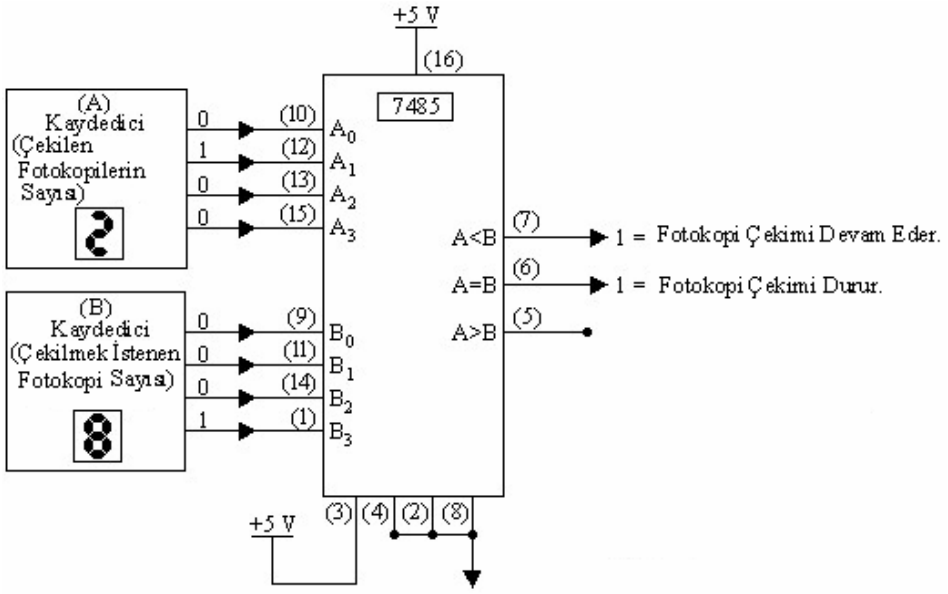
A_3,B_3	A_2,B_2	A_1,B_1	A_0,B_0	$I_{A>B}$	$I_{A<B}$	$I_{A=B}$	$Q_{A>B}$	$Q_{A<B}$	$Q_{A=B}$
$A_3>B_3$	X	X	X	X	X	X	1	0	0
$A_3<B_3$	X	X	X	X	X	X	0	1	0
$A_3=B_3$	$A_2>B_2$	X	X	X	X	X	1	0	0
$A_3=B_3$	$A_2<B_2$	X	X	X	X	X	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1>B_1$	X	X	X	X	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1<B_1$	X	X	X	X	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0>B_0$	X	X	X	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0<B_0$	X	X	X	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	0	0	1	0	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	1	0	0	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	X	X	1	0	0	1
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	0	0	0	1	1	0
$A_3=B_3$	$A_2=B_2$	$A_1=B_1$	$A_0=B_0$	1	1	0	0	0	0

Şekil 8.48. Dört bitlik karşılaştırma işlemi doğruluk tablosu.



Şekil 8.49. Karşılaştırıcıların kaskat bağlanması.

Karşılaştırıcı devresine örnek olarak; Şekil 8.50'deki fotokopi makinası kontrol devresini verebiliriz. 7485 entegresi kullanılarak oluşturulan devrede; çekilmek istenen fotokopi sayısını temsil eden değer 'B' girişine, çekilen fotokopileri sayan devrenin çıkışı ise 'A' girişine uygulanır. İki sayı birbirine eşit oluncaya kadar A<B çıkışı '1' olur ve fotokopi çekimi devam eder. A ve B girişlerindeki değerlerin aynı olması durumunda; 'A=B' çıkışı '1' olur ve fotokopi çekim işlemi durur.



Şekil 8.50. Fotokopi makinası kontrol devresi.

8.5.2. Aritmetik İşlem Devreleri

Toplama, çıkarma, çarpma, bölme işlemlerini yapan devrelere, '**Aritmetik İşlem Devreleri'** denir. Bilgisayarlarda ve hesap makinalarında, temel işlemler toplama ve çıkartma işlemleridir. Çarpma işlemi; toplama işleminin tekrarlanması, bölme işlemi ise; çıkartma işleminin tekrarlanması ile yapılır. Bu nedenle toplayıcı ve çıkarıcı devrelerini detaylı olarak inceleyeceğiz.

8.5.2.1. Toplayıcı Devreleri (Adders)

Bilgisayarlar ve hesap makinaları, her biri çok sayıda bite sahip iki adet ikili sayıyı toplama işlemini gerçekleştirirler. En basit toplama işlemi dört olası temel işlemi içerir.

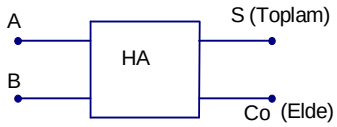
0+0=0,
0+1=1,
1+0=1,
1+1=10, (Elde 1, Toplam = 0)

İlk üç işlemde tek basamaklı bir sayı elde edilirken, son işlemde ikinci basamak ortaya çıkar ve ikinci basamak **'elde biti'** (carry bit) olarak isimlendirilir. İki biti toplayan devreler

‘**yarım toplayıcı**’ olarak, üç bitin toplamını yapan devreler ise ‘**tam toplayıcı’** olarak isimlendirilir. Yarım toplayıcı terimi, tam toplayıcıyı oluşturmak için iki tane yarım toplayıcı kullanılmasından ileri gelmektedir. Yarım ve tam toplayıcı tasarım prensiplerini açıkladıktan sonra toplayıcı uygulamalarını inceleyelim.

i- Yarım Toplayıcı (Half Adder – HA)

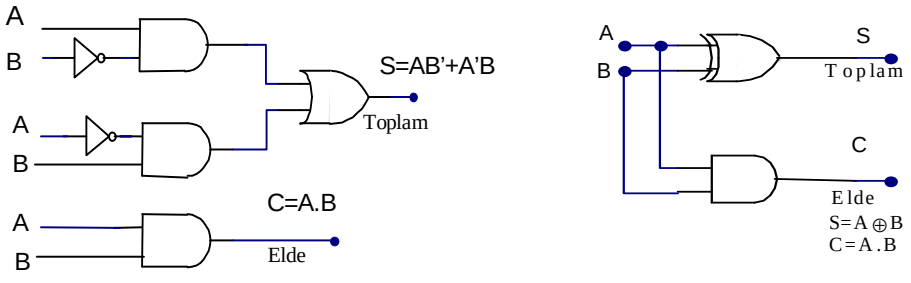
Girişine uygulanan iki biti toplayıp, sonucu toplam (sum) ve elde (carry) şeklinde veren toplayıcı devresi, ‘**yarım toplayıcı**’ olarak isimlendirilir (Şekil 51). Yarım toplayıcı devresi, doğruluk tablosundan elde edilen fonksiyonların lojik devresinin çizilmesi ile oluşturulur. Oluşan devrede, ‘Toplam’ ve ‘Elde’ değerlerini temsil eden iki çıkış bulunur (Şekil 8.52).



A	B	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Şekil 8.51. Yarım toplayıcı sembolü ve doğruluk tablosu.

Yarım toplayıcı çıkışlarındaki sadeleştirilmiş fonksiyonlar, S = A'B+AB' ve C = AB şeklinde elde edilir (Şekil 8.52.a). Girişlerin A ve B, çıkışların S ve C değişkenleri ile ifade edildiği yarım toplayıcı devresi, bir ‘Özel-VEYA’ (EXOR) ve bir ‘VE’ kapısıyla oluşturulabilir (Şekil 8.52.b).

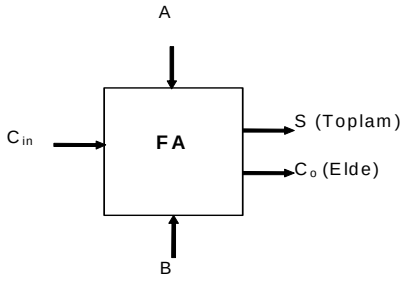


(a) **(b)**

Şekil 8.52. Yarım toplayıcı devresi lojik şemaları.

ii - Tam Toplayıcı (Full Adder - FA)

Üç adet Bir bitlik sayının toplamını gerçekleştiren ve sonucu S ve C olarak isimlendirilen iki çıkış hattında gösteren düzenek, '**Tam Toplayıcı**' olarak isimlendirilir (Şekil 8.53.a.). Girişlerden ikisi toplanacak bitleri gösterirken, üçüncü giriş bir önceki düşük değerlikli basamaktan gelen eldeyi (carry) ifade etmek için kullanılır. Tam toplayıcı devresi tasarlamak için Şekil 8.53.b'deki doğruluk tablosundan faydalanılabilir.



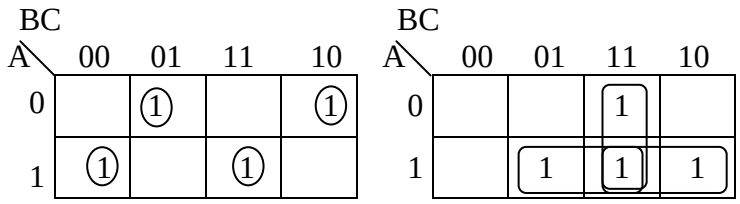
(a)

A	B	C	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b)

Şekil 8.53. Tam toplayıcı sembolü ve doğruluk tablosu.

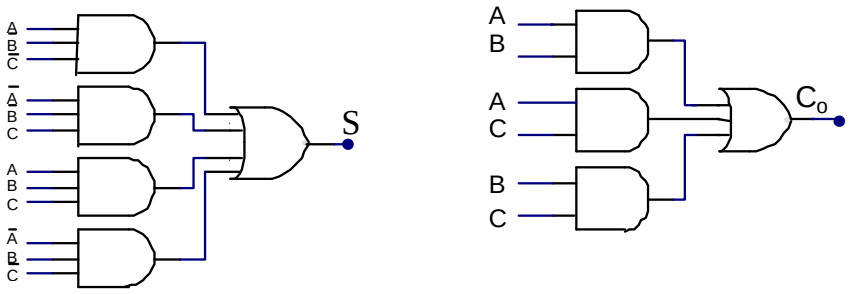
Toplayıcı tasarımında, doğruluk tablosunda giriş değişkenlerinin alabileceği değerler sıralandıktan sonra, kombinasyonlarda bulunan '1' değerleri Boolean kurallarına göre toplanıp, sonuçlar S ve C_o kolonlarına yazılır. Devrede iki çıkış bulunduğundan, her bir çıkış için uygun olan değerleri içeren sütunlar oluşturulur ve sütunlardaki değerler Karnaugh haritalarına taşınır. Karnaugh haritalarından lojik eşitlikler elde edilir ve elde edilen eşitliklerin lojik şemaları çizilir.



S=$A\bar{B}\bar{C}+\bar{A}\bar{B}C+ABC+\bar{A}B\bar{C}$ **Co**=AC+BC+AB

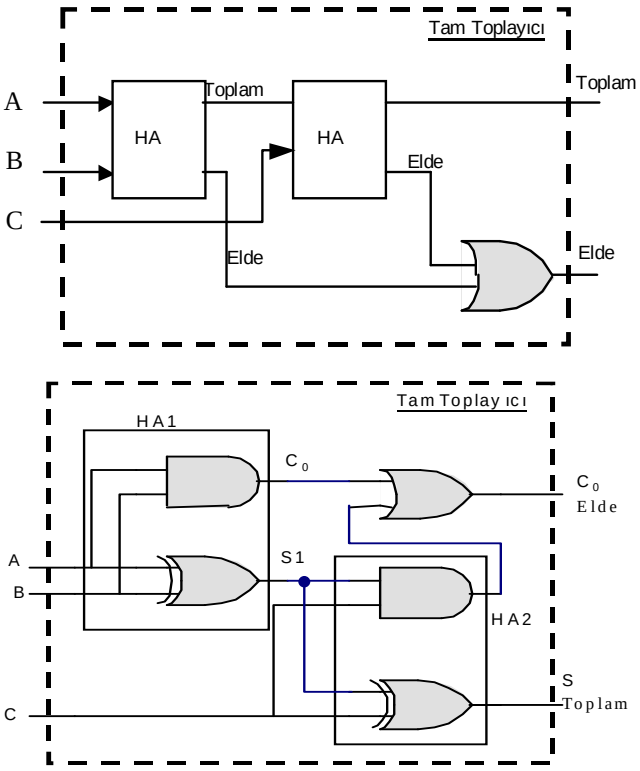
Şekil 8.54. Tam toplayıcı tasarımı için Karnaugh haritalarının oluşturulması.

Şekil 8.53'deki doğruluk tablosundaki değerlerin Karnaugh haritalarına taşınması ile Şekil 8.54'deki haritalar elde edilir. Haritalardan elde edilen S ve Co eşitliklerine ait devrelerin çizilmesi ile, Şekil 8.55'teki lojik devreler oluşur.



Şekil 8.55. Tam toplayıcı lojik devresi.

Tasarım sonucunda çizilen lojik devrelerle yapılabilecek tam toplama işlemi, iki adet yarım toplayıcı ve bir 'VEYA' kapısı kullanılarak gerçekleştirilebilir (Şekil 8.56). Bu şekilde gerçekleştirilen devrede; ikinci yarım toplayıcının S çıkışı, ilk yarım toplayıcının S çıkışı ile C'nin Özel-VEYA'ya uygulanmasının sonucudur.



Şekil 8.56. İki yarım toplayıcı ve 'VEYA' kapısı ile tam toplayıcı elde edilmesi.

Bu durumda toplam çıkışı;

$$S=C \oplus (A \oplus B)$$

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

$$=C'A'B+C'AB'+C[(A'B)'.(AB)']$$

$$=C'A'B+C'AB'+C[(A+B').(A'+B)]$$

$$=C'A'B+C'AB'+C[AA'+AB+A'B'+BB']$$

($AA'=0$, $BB'=0$)

$$=C'A'B+C'AB'+ABC+A'B'C$$

sonucunu verir. Elde çıkışı ise;

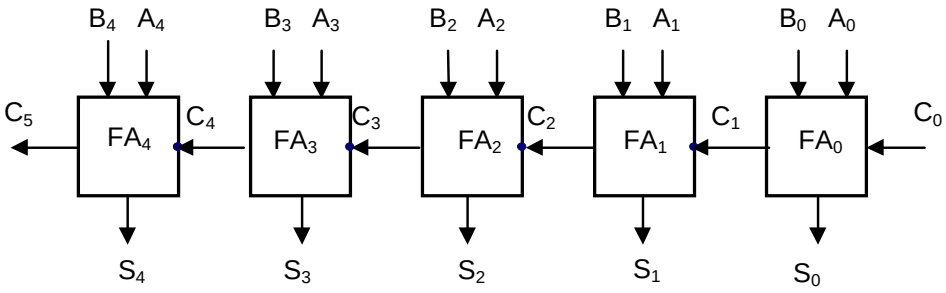
$$Co=C'(AB'+AB)+AB' = AB'C+ABC+AB$$

şeklinde yazılabilir.

iii- Paralel Toplayıcı :

Yarım ve tam toplayıcı işlemlerinde, tek bitlik sayıların toplamı işlemi açıklandı. Bununla beraber, her biri çok sayıda ikili basamak içeren iki sayının toplanması işlemini aynı anda yapan devrelere ihtiyaç vardır. Bilgisayarlarda ve hesap makinalarında çok sayıda bite sahip iki sayıyı aynı anda toplayan devreler '**paralel toplayıcı**' olarak isimlendirilir. Şekil 8.57'de, her biri beş bitlik iki sayıyı toplayan paralel toplayıcının blok şeması görülmektedir. Bu devrede toplama işlemi, en düşük basamaklı bilgilerin toplanması ile başlar.

En düşük değerli basamakta C_o biti '0' olduğundan; A_o ve B_o değerleri toplanarak S_0 ve C_0 çıkışlarına gönderilir. Bunun dışındaki basamakları toplamak için, A_i, B_i, C_i bitler toplanarak ilgili S_i ve C_i çıkışlarında gösterilir. C_i çıkışındaki bilgi, bir sonraki yüksek basamak değerlikli bitlerin toplandığı FA_i'nın C_i girişine uygulanır.

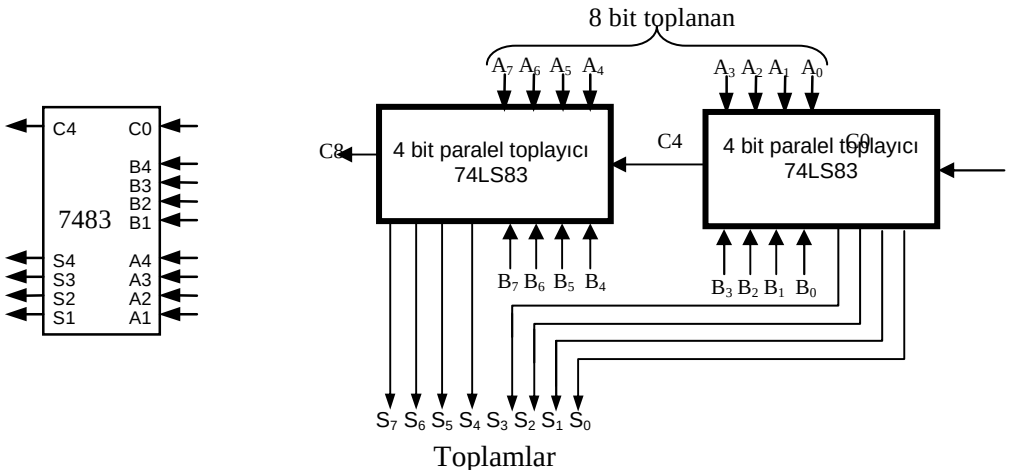


Şekil 8.57. Beş bitlik iki sayının paralel toplayıcı ile toplanması.

Sonuç olarak; her bir FA, girişlere uygulanan üç bitin (A, B ve C) toplamını yaparak, toplam sonucunu S ve C çıkışlarında gösterir. Örneğin, FA_3 tam toplayıcı devresi A_3, B_3 ve C_3 değerlerini toplayarak sonucu C_4 ve S_3 çıkışlarında gösterir.

Yapılan örnekte, beş bitlik iki sayının toplanması açıklandı. Günümüz bilgisayar sistemlerinde aynı anda toplanacak sayılar 8 ile 64 bit arasında değişmektedir.

Pratikte tüm FA'lardaki toplama işlemi aynı anda yapıldığından, paralel toplayıcılar çok hızlı işlem yaparlar. Piyasada 7483, 74283, 74LS83A ve 74HC283 (CMOS) gibi farklı yapıda dört bitlik paralel toplayıcılar bulunmaktadır. Şekil 8.58.a'da 74LS283 tipi dört bitlik paralel toplayıcının fonksiyon şeması görülmektedir. Dört bitlik paralel toplayıcı iki adet dört bitlik girişe (A_3,A_2,A_1,A_0 ve B_3,B_2,B_1,B_0) ve en düşük basamaklı bit (LSB) için kullanılan C_o girişine sahiptir. Çıkış olarak; dört adet toplam çıkışı (S_3, S_2, S_1, S_0) ile birlikte en yüksek basamaklı bitin elde çıkışı olan C_4 bulunur.



a) 74LS83 sembolü b) Paralel toplayıcıların kaskat bağlantısı.

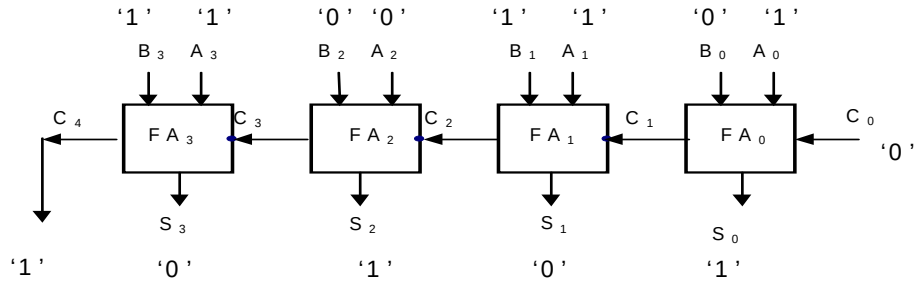
Şekil 8.58. Paralel toplayıcı sembolü ve paralel toplayıcıların kaskat bağlanması.

Paralel toplayıcıları kaskat bağlayarak, daha fazla sayıdaki bitleri paralel olarak toplamak mümkündür. Şekil 8.58.b'de iki adet dört bitlik paralel toplayıcının blok şeması görülmektedir. Kaskat bağlanan paralel toplayıcılarda, sağdaki toplayıcı düşük basamak değerlikli dört biti toplayarak, bitlerin toplamını ve toplam sonucunda oluşan elde bitini çıkış olarak verir.

Yüksek değerlikli bitleri toplayan ikinci paralel toplayıcı, elde girişi olarak sağdaki toplayıcıdan elde edilen C_4 çıkışını kullanır ve bitlerin toplamı ile elde bilgisini çıkış olarak verir. C_8 çıkışı, ikinci paralel toplayıcının işlemler sonucunda elde ettiği elde değerini gösterir. Bu çıkış elde bilgisi olarak daha sonraki toplayıcılar için kullanılabilir.

Örnek 7 : Dört bitlik paralel toplayıcı devresi ile, 1011 ve 1010 ikili sayılarını toplama işlemini yapalım.

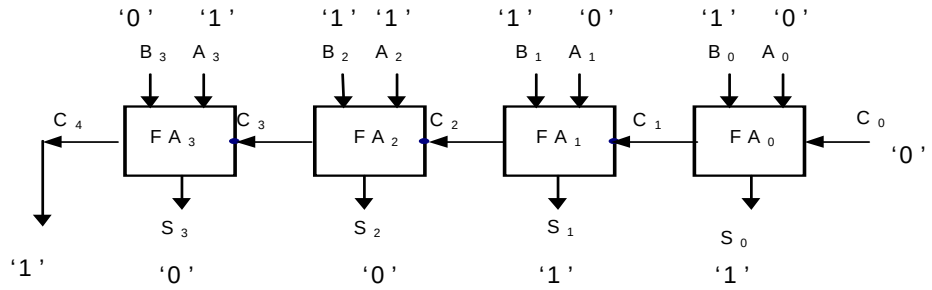
Herbir tam toplayıcıya uygulanan sayılar ile, elde girişlerinden gelen değer toplanır. Toplama sonucu ilgili toplam (S) ve elde (C) çıkışlarına gönderilir. Tüm tam toplayıcılarda oluşan değerlerin yazılması ile Şekil 8.59'daki değerler oluşur.



Şekil 8.59. Dört bitlik paralel toplayıcı uygulaması.

Örnek 8 : 0111 + 1100 işlemini dört bitlik paralel toplayıcı ile yapmak için gerekli devreyi çizerek, işlem sonuçlarını gösterelim.

Toplanacak sayılar, tam toplayıcıların girişlerine uygulanarak çıkışları yazılırsa Şekil 8.60'daki değerler bulunur.



Şekil 8.60. Paralel toplayıcı ile örnek toplama işlemi.

8.5.2.2. Çıkarıcı Devreleri (Subtractor Circuits)

Lojik devrelerde yapılan ikinci temel işlem çıkarmadır. İki bitin çıkarmasını yapan devreye '**yarım çıkarıcı'**, üç bitin çıkarmasını yapan devreye ise '**tam çıkarıcı'** devresi denir.



$D = A^{I}B + AB^{I} = A + B$

$B = A^{I}B$

(a)

Girişler		Çıkışlar	
A	B	Fark A-B	Borç (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

(b)

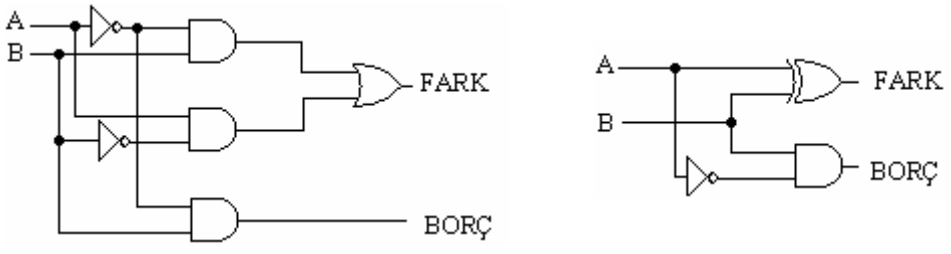
Şekil 8.61. Yarım çıkarıcı sembolü ve doğruluk tablosu.

i- Yarım Çıkarıcı Devresi (Half Subtractor)

İki bitin çıkarması işlemini yapan çıkarıcı devresinde, iki giriş ve iki çıkış bulunur (Şekil 8.61.a). Çıkışlardan birisi sayının farkını (difference-D), diğeri borç bitini (borrow-B) gösterir. İki bitin çıkarılması işleminde dört farklı durum oluşur:

$0 - 0 = 0$
$1 - 0 = 1$
$1 - 1 = 0$
$0 - 1 = 1$ (Borç 1)

A-B işleminde A<B olduğunu zaman '0–1' işlemi oluşur ve bu durumda bir yüksek değerli basamaktan '1' borç alınır. Borç çıkışı, doğruluk tablosunda ayrı bir sutün olarak gösterilir. Yarım toplayıcı devresinde oluşan işlemlerin doğruluk tablosu ve doğruluk tablosuna göre oluşan fonksiyonlar Şekil 9.61'de gösterilmektedir.



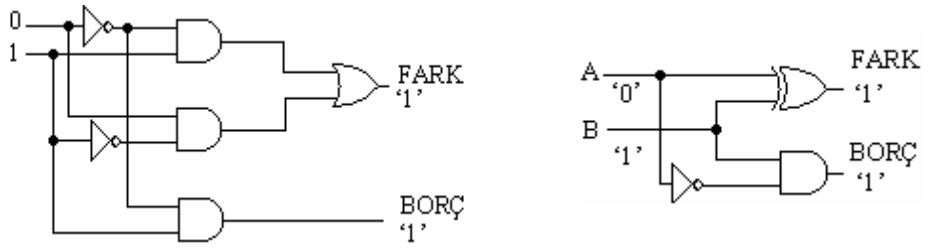
(a) **(b)**

Şekil 8.62. Yarım çıkarıcı lojik devreleri.

Doğruluk tablosundan elde edilen fonksiyonları gerçekleştirecek devrenin çizilmesi ile, Şekil 8.62.a'daki devre elde edilir. Devreden elde edilen 'Fark' çıkışındaki eşitliği 'Özel-VEYA' kapısı ile gerçekleştirerek devre çizilirse, Şekil 8.62.b'deki lojik devre oluşur.

Örnek 9 : Yarım çıkarıcı devreleri ile 0-1 işlemini yapalım.

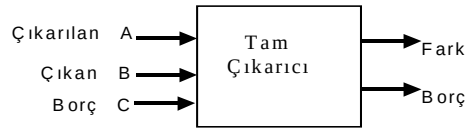
Yarım toplayıcı devre girişlerine A=0 ve B=1 değerleri uygulanırsa; Şekil 9.63'deki çıkış değerleri elde edilir.



Şekil 8.63. Yarım çıkarıcı devreleri ile örnek çıkarma işlemi.

ii- Tam Çıkarıcı Devresi (Full Subtractor)

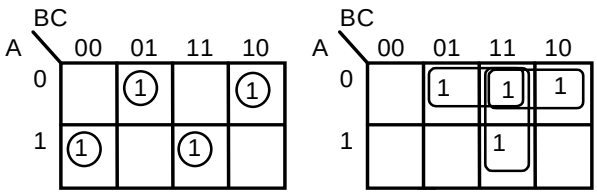
Daha düşük değerli basamak tarafından '1' borç alınmış olabileceğini dikkate alarak iki biti birbirinden çıkaran bileşik devre, '**tam çıkarıcı**' olarak isimlendirilir. Üç giriş ve iki çıkışa sahip tam çıkarıcı devresinde girişler; çıkarılan, çıkan ve borcu gösterirken, çıkışlardan biri farkı diğeri borcu gösterir (Şekil 8.64.a).



(a)

Girişler			Çıkışlar	
A	B	C	Fark	Borç
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

(b)

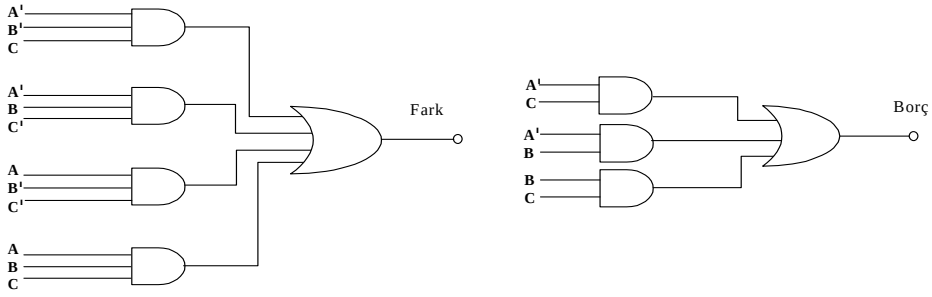


FARK=A'B'C+A'BC'+AB'C'+ABC BORÇ =A'C+A'B+BC

(c)

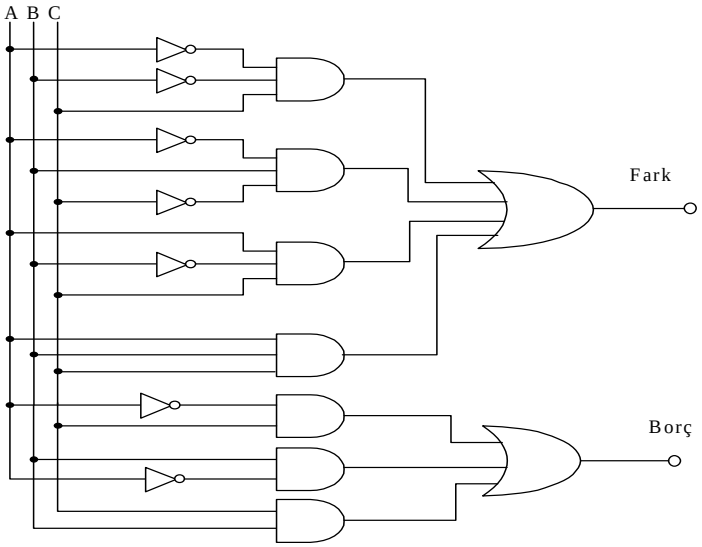
Şekil 8.64. Tam çıkarıcı devresi sembolü, doğruluk tablosu ve Karnaugh şemaları.

Girişin üç adet olması nedeniyle, doğruluk tablosunda 8 farklı durum bulunur (Şekil 8.64.b) Doğruluk tablosundaki A=B=0, C=1 durumu, C bitinin daha önce düşük basamak değerli kademeden '1' borç alındığını gösterir. Doğruluk tablosu çıkış sütunlarındaki değerlerin Karnaugh haritasına taşınması ile Şekil 8.64.c'deki eşitlikler elde edilir. Elde edilen eşitliklerden, 'Fark' sadeleştirilemese de, 'Borç' ifadesi sadeleştirilebilir.



Şekil 8.65. Tam çıkarıcı devresinde 'Fark' ve'Borç' çıkışları.

Karnaugh haritalarından yazılan eşitliklerin lojik şemalarının çizilmesi ile, Şekil 8.65'de görülen tam çıkarıcı devresi elde edilir. Bu devrede 'fark' ve 'borç' ifadelerini temsil eden devreler ayrı ayrı çizilmiş ve değişkenlerin hem kendilerinin hem de tümleyenlerinin girişlere uygulandığı kabul edilmiştir. İki ifadenin beraber çizilmesi ve değişkenlerin yanlızca kendilerinin girişlere uygulanması durumunda, Şekil 8.66'daki lojik devreler oluşur.

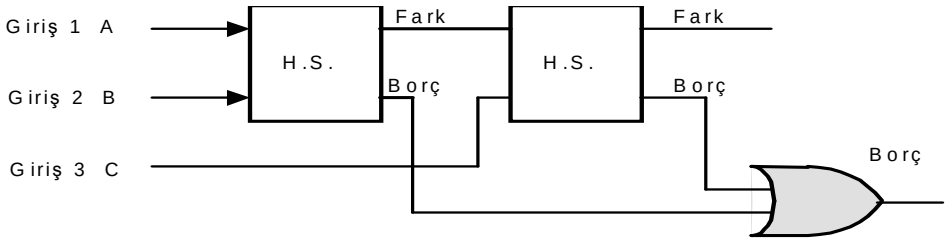


Şekil 8.66. Tam çıkarıcı devresi lojik şeması.

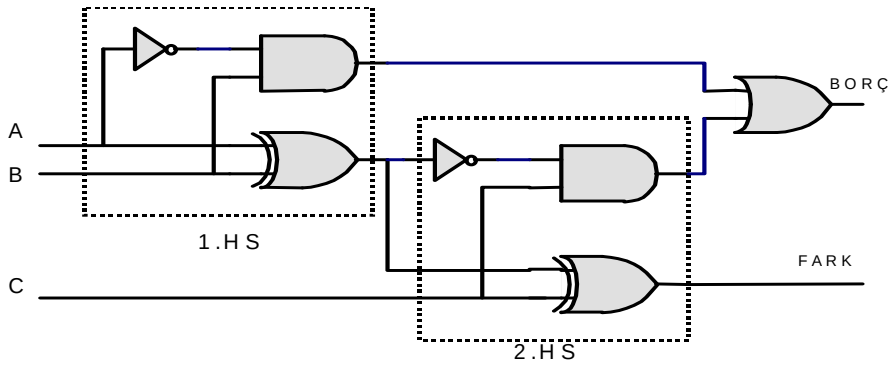
iii- İki Yarım Çıkarıcı Kullanılarak Tam Çıkarıcı Elde Edilmesi :

İki yarım toplayıcı kullanılarak tam toplayıcı yapıldığı gibi, iki yarım çıkarıcı (H.S.) kullanılarak tam çıkarıcı oluşturulabilir (Şekil 8.67). İki yarım çıkarıcı ile oluşturulan tam çıkarıcı devresinde, I. Yarım çıkarıcının 'Fark' çıkışı II. yarım çıkarıcının girişlerinden birisi olarak uyğulanırken, C girişi ikinci yarım çıkarıcının diğer girişine uygulanır. İki yarım çıkarıcının 'Borç' çıkışları bir 'VEYA' kapısı ile birleştirilerek, tam çıkarıcı devresi 'Borç' çıkışı elde edilir.

Şekil 8.67'de blok şeması görülen tam çıkarıcı devresinde blok olarak gösterilen yarım çıkarıcıların yerine açık devrelerinin yerleştirilmesi ile tam toplayıcı lojik devresi elde edilir (Şekil 8.68). Bu devrenin çıkışının tasarım yoluyla elde edilen ve Şekil 8.66'da şeması verilen devre ile aynı işlevi yaptığının kontrolünü yapalım:



Şekil 8.67. İki yarım çıkarıcı ile tam toplayıcı oluşturulması blok şeması.



Şekil 8.68. İki yarım çıkarıcı ile tam çıkarıcı devre oluşturulması lojik şeması.

Kontrol için, çizilen lojik devrenin 'Borç' ve 'Fark' çıkışlarını bulup, bulunan değerler ile tasarımda bulunan eşitlikleri karşılaştıralım.

Çizilen devrede çıkışlar;

Fark = C ⊕ (A⊕B) ve Borç = $A'B + (\overline{A \oplus B}) . C$

şeklinde oluşur. Oluşan eşitlikler açılırsa;

Fark = C ⊕ (A ⊕ B)

A ⊕ B = X

kabul edersek;

Fark = C ⊕ X

= $C'X + CX'$ ($\overline{A \oplus B} = \overline{A'B + AB'}$)

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

= $A'BC' + AB'C' + A'BC + AB'C$

Borç = $A'B + (\overline{A \oplus B}) .C$

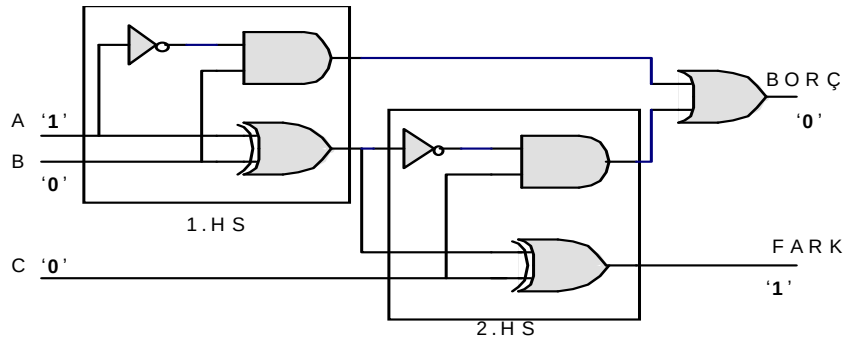
= $A'B + (A'B' + AB) .C$

= $A'B + A'B'C + ABC$

şeklindeki genişletilmiş ifadeler bulunur. Bulunan 'Fark' ve 'Borç' ifadeleri, tasarım yolu ile bulunan çıkış ifadeleri ile aynıdır. Bu ifadelerin aynı olması, çizilen tam çıkarıcı devresinin doğru olduğu sonucunu verir. Devrenin doğru olduğunu bulduğumuza göre, bu devre ile çıkarma örnekleri yapalım.

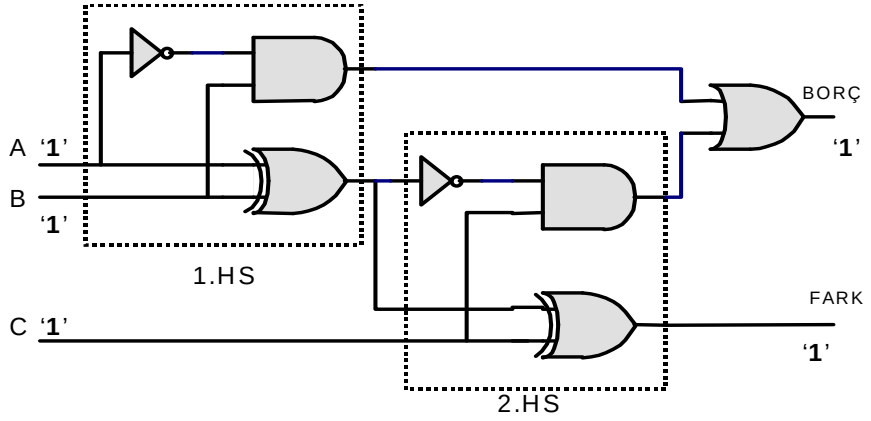
Örnek 10 : Şekil 8.69'daki tam çıkarıcı devresi ile, A=1 ve B=C=0 olduğu durumda oluşan çıkışları bulalım.

Tam çıkarıcıya A=1 ve B=C=0 girişleri uygulaması ile, Fark=1 ve Borç=0 çıkışları oluşur.



Şekil 8.69. A=1 ve B=C=0 girişlerinde tam çıkarıcıda oluşan çıkışlar.

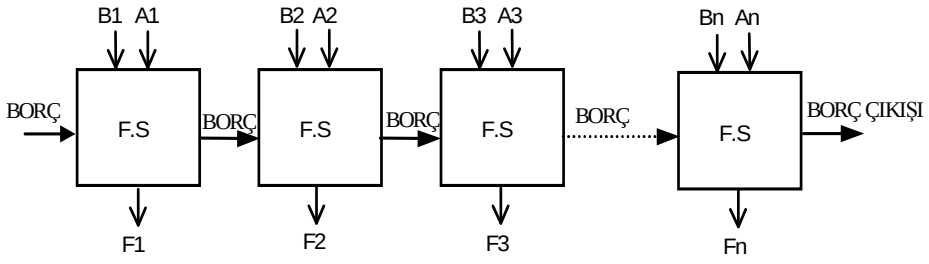
Örnek 11 : Tam çıkarıcı devrede, A=B=C=1 durumunda ‘Fark’ ve ‘Borç’ çıkışlarında oluşan değerleri gösterelim.



Şekil 8.70. A=B=C=1 girişlerinde, tam çıkarıcı devresinde oluşan çıkış değerleri.

iv – Paralel Çıkarıcı devresi:

‘n’ bitlik iki adet ikili sayıyı çıkaran paralel çıkarıcı devresinde, paralel toplayıcılarda olduğu gibi ‘n’ sayıda tam çıkarıcı (F.S.) devresi kullanılır (Şekil 8.71). Blok şema olarak gösterilen paralel çıkarıcılarda en sondaki borç çıkışı ‘1’ ise; çıkarmanın sunucunun pozitif, ‘0’ ise sonucun negatif olduğunu gösterir.



Şekil 8.71. Paralel çıkarıcı devresi blok şeması.

Dört bitlik toplama işlemi yapabilen 7483 entegresi ile dört bitlik çıkarma işlemi gerçekleştirilebilir. Bu entegre, C_0=0 olduğu zaman toplama, C_0=1 durumunda ise çıkarma işlemi yapar. Çıkarma işlemi yapılırken, çıkarılan sayıyı temsil eden ‘A’ bitleri entegreye aynen uygulanırken, çıkan sayıyı temsil eden ‘B’ bitlerinin tümleyenleri alınarak entegreye uygulanır.

v- İki tümleyeni ile toplama ve çıkarma işlemi:

Birçok bilgisayar sistemi, negatif sayıları ifade etmek veya çıkarma işlemini gerçekleştirmek için '2 tümleyeni' aritmetiğini kullanır. Negatif sayıları ifade etmek için 2 tümleyeni aritmetiği kullanılıyorsa, işaretli (-veya +) sayıların toplanması ve çıkarması işlemleri yalnızca toplama yolu ile gerçekleştirilir.

Toplama İşlemi :

Negatif sayıların 2 tümleyeni formunda ifade edilmesi durumunda pozitif ve negatif sayıların toplanması temel paralel toplama devresi ile gerçekleştirilebilir. Şekil 8.72'de (-3) ve (+6) sayılarının paralel toplayıcı ile toplanması işlemi görülmektedir.

Yapılan işlem 'Toplama' olmasına rağmen, sayıların işaretleri farklı olduğundan toplanan sayıların farkı alınır. Fark alma işleminde;

'+' işaretli sayıya, '-' işaretli sayının iki tümleyeni eklenir.

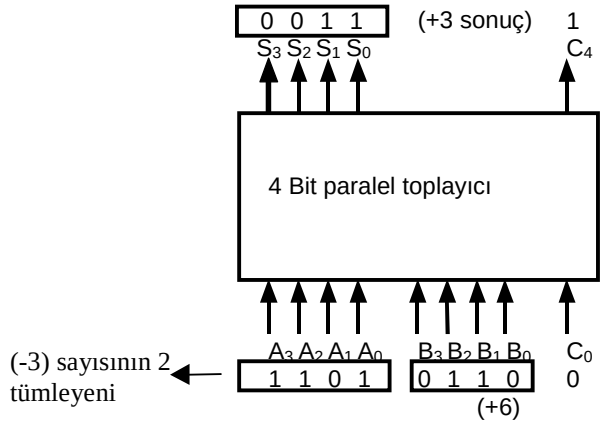
Bulunan sonuçta elde olup olmadığına bakılır:

- Elde varsa atılır ve bulunan sonuç pozitiftir.

- Elde yoksa, elde edilen sayının '2 tümleyeni' alınır ve sayının önüne '-' işareti konur.

Örnek 12: (-3) ve (+6) sayılarını, ikili paralel toplayıcı ile toplayalım:

(-3) sayısının tümleyeni A girişlerine, pozitif (+6) sayısı ise B girişlerine uygulanırsa, fark çıkışlarından sonuç elde edilir. Borç çıkışını temsil eden C_4 çıkışı '1' olduğundan, sonuç pozitiftir.



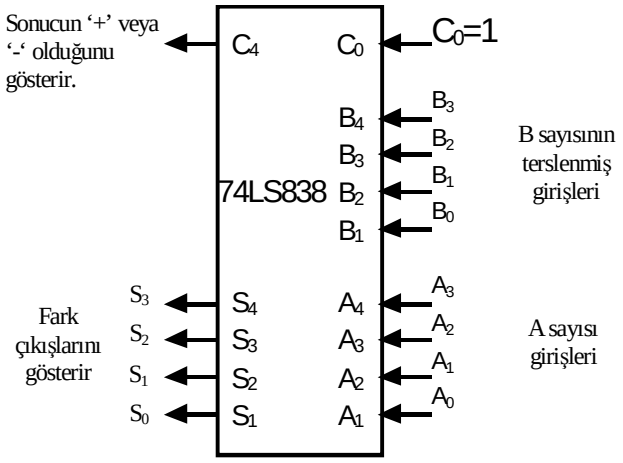
Şekil 8.72. Negatif ve pozitif sayıların paralel toplayıcı ile toplanması.

Çıkarma İşlemi :

Çıkarma işlemi için 2 tümleyen aritmetiği yöntemi kullanılması durumlarında, çıkan sayının 2 tümleyeni alınarak toplama işlemi yapılır. Örneğin, A-B işlemi yapılıyorsa, A sayısı olduğu gibi bırakılıp, B sayısının 2 tümleyeni alınır. Daha sonra, A sayısı ile tümleyeni alınan B sayısı toplanır ve iki sayı arasındaki fark toplayıcı çıkışından okunur.

Dört bitlik paralel toplayıcı devresi, yukarıda açıklanan yöntem ile çıkarma işlemi yapacak şekilde düzenlenebilir. Daha önceki bölümlerden hatırlanacağı üzere, ikili sayının 2 tümleyeni, her bir bitin tersi yazılarak ve en düşük değerlikli bite '1' ekleyerek elde edilir. Bu işlem, B sayısının terslenmiş bitlerinin B_3, B_2, B_1 ve B_0 girişlerine, eklenen '1' sayısının ise C_0 girişine uygulanması ile gerçekleştirilir. Böylece, B sayısındaki rakamların tersi alınıp, en düşük değerlikli basamağa '1' eklenmiş olur. A sayısının doğrudan A_3, A_2, A_1 ve A_0 girişlerine uygulanması ile gerekli bilgiler paralel toplayıcı girişlerine uygulanır (Şekil 8.73).

Dört bitlik paralel toplayıcıda S_0 - S_3 çıkışları çıkarma işleminin sonucunu gösterirken, C_4 çıkışı sonucun '-' veya '+' olduğunu belirtir. C_4 =1 ise sonuç pozitif, C_4=0 ise sonuç negatiftir.



Şekil 8.73. Çıkarma işleminin 2 tümleyeni kullanarak paralel toplayıcılar ile gerçekleştirilmesi.

Örnek 13 : Çıkarma işleminin nasıl yapıldığını açıklamak için; (+4) - (+6) işlemini yapalım.

i- A (+4=0100) ve B (+6=0110) sayıları toplayıcı girişlerine uygulanır. Ancak, B sayısının 2 tümleyeni alınması gerektiğinden, B sayının 2 tümleyeni alınarak '1010' şeklinde B girişine uygulanmalıdır.

ii- Bu durumda, 0100 sayısı ile 1001 sayısı, C_0=1 eklenerek toplama işlemine tabi tutulur.

iii- Sonuç olarak 1110 sayısı elde edilir. Bu sayının işaret biti olarak '0' değerine sahip olması, sonucun negatif ve 2 tümleyeni formunda olduğunu gösterir.

iv- Bulunan sayının 2 tümleyeni alınarak önüne '-' işareti konulmasıyla, doğru sonuç (-0010) bulunur.

Aynı entegreyi toplama ve çıkarma devresi olarak kullanmak mümkündür. Bu şekilde tasarlanan devreler Flip-Flop ve kaydedici içerdiğinden daha sonraki konularda incelenecektir.

BCD Toplayıcı :

Çoğu bilgisayarlar ve hesap makineleri, onlu sayılarla işlem yaparken BCD kodunu kullanır. Bu kodlama sisteminde her bir onlu sayı, 0000 ile 1001 arasındaki dört bitlik sayı ile ifade edilir. BCD kodlu sayılarla yapılan toplama işleminde oluşan olayları inceledikten sonra, BCD toplayıcı lojik devresini çizelim.

i- Onlu sayıların toplanmasında toplama sonucunda elde edilen sayılar 9 veya 9'dan küçük ise, sonuç toplanan sayıların toplamına eşittir ve herhangi bir işleme gerek yoktur.

Örnek 14 : $(45)_{10}$ ve $(33)_{10}$ sayılarını toplayalım.

Toplanan sayılarda bulunan herbir basamaktaki sayı BCD kodlu sayı olarak yazıldıktan sonra, aynı basamak değerine sahip sayılar toplanır.

45	0100	0101
+ 33	+ 0011	0011
78	0111	1000

Sonuç olarak bulunan $(78)_{10}$ sayısında, her iki basamakta bulunan sayılar 9'u geçmediği için elde yoktur. Bu durumda BCD toplama işleminde bulunan sayı doğrudan sonucu gösterir.

ii- İki sayının toplamı sonucunda elde edilen BCD kodlu sayı 9'dan (1001) büyük ise, elde edilen sonucun düzeltme işlemine tabi tutulması gerekir. Düzeltme işlemi, toplama sonucuna $(6)_{10} = (0110)_2$ sayısının eklenmesidir.

Örnek 15 : $(6)_{10}$ ve $(7)_{10}$ sayılarını toplayalım.

Her iki sayı BCD kodlu olarak yazıldıktan sonra toplama işlemi yapılırsa;

6	0110	
+ 7	+ 0111	
13	1101	sonucu bulunur.

Sonuç $(1101)_2$ olduğundan ve BCD formuyla 9'a kadar olan sayılar ifade edilebildiğinden, sonuca $(0110)_2$ sayısının eklenmesi gerekir. Sonuca bu sayının eklenmesi ile;

```
      1101
+     0110
  0001 0011   BCD kodlu (13)10 sayısı bulunur.
    1    3
```

Örnek 16 : $(47)_{10}$ ve $(35)_{10}$ sayılarını BCD formunda toplayalım.

Verilen sayılar basamak bazında BCD kodlu sayı olarak ifade edildikten sonra, herbir basamaktaki sayılar toplanır.

```
  47    0100  0111
+ 35  + 0011  0101
  82    0111  1100    9'dan büyük olduğundan;
      +   1   0110    6 sayısı eklenmelidir. Bu sayının eklenmesi ile
    1000    0010    BCD kodlu sayıların toplamının doğru sonucu bulunur.
      8       2
```

Örnek 17 : $(275+641)_{10}$ işlemini BCD kodlu sayıların toplanması formunda yapalım.

Toplanacak sayılar BCD formuna dönüştürülür ve sayılar toplanırsa;

```
  275      0010  0111  0101
+ 641      0110  0100  0001
  916      1000  1011  0110   sonucu bulunur. 1001'den büyük olanlara '0110'
             +  1  0110       sayısının eklenmesiyle
           1001  0001  0110   BCD kodlu → (916)10 sayısı bulunur.
```

İki sayının BCD toplayıcı formunda toplanması ile ilgili olarak yapılan açıklamalar ve örnekler ışığı altında, iki BCD sayıyı toplayan 4 bitlik paralel toplayıcı (74LS83 entegresi) devresini inceleyelim. Eğer iki BCD sayı A_3,A_2,A_1,A_0 ve B_3,B_2,B_1,B_0 sembolleriyle gösterilir ve toplama işlemine tabi tutulurlarsa;

$$\begin{array}{r} A_3A_2A_1A_0 \\ +\ B_3B_2B_1B_0 \\ \hline S_4S_3S_2S_1S_0 \end{array}$$

ikili sayısı elde edilir. S_4 basamağı C_4 çıkışında bulunan değeri ifade eder.

Lojik BCD toplayıcı, yukarıdaki toplama işlemini gerçekleştirmesinin yanında toplama işlemini sonucunun 01001'den büyük olduğu durumu tespit eden lojik devreye sahip olmalıdır. Doğrultma işleminin gerekli olduğu durumlar, Şekil 8.74'deki tabloda listelenen kombinasyonlardır.

Listelenen kombinasyonları incelersek aşağıdaki genellemeleri yapabiliriz.

i- S_4=1 olduğu bütün durumlar,

ii- S_3=1 iken S_1 veya S_2'nin 1 olduğu durumlar, doğrultma işleminin gerekli olduğu durumlardır.

Bu durumlar $f=S_4+S_3\,(S_2+S_1)$ fonksiyonu ile ifade edilebilir. f=1 olduğu durumlarda, sonuca $(0110)_2$ eklenerek düzeltme işleminin yapılması ve elde biti üretilmesi gereklidir. Şekil 8.75'te düzeltme devresini içeren BCD toplayıcı devrenin şeması görülmektedir.

S_4	S_3	S_2	S_1	S_0	
0	1	0	1	0	(10)
0	1	0	1	1	(11)
0	1	1	0	0	(12)
0	1	1	0	1	(13
0	1	1	1	0	(14)
0	1	1	1	1	(15)
1	0	0	0	0	(16)
1	0	0	0	1	(17)
1	0	0	1	0	(18)

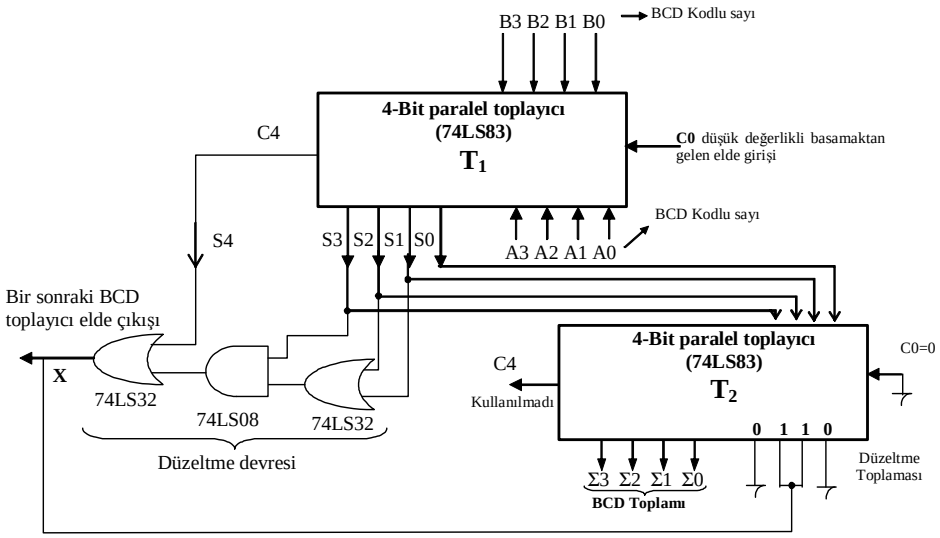
Şekil 8.74. BCD toplama işleminde düzeltme işlemi gerektiren kombinasyonlar.

Toplayıcı devresinde, BCD kodlu '$A_3A_2A_1A_0$' ve '$B_3B_2B_1B_0$' sayıları, '$S_3S_2S_1S_0$' toplama sonucunu elde etmek için 4 bitlik T_1 toplayıcısında toplanır. T_2 BCD toplayıcı yalnızca f=1 olduğu durumlarda sonuca $(0110)_2$ sayısını eklemek ve '$\Sigma_3\ \Sigma_2\ \Sigma_1\ \Sigma_0$' ile sembolize edilen doğru sonucu bulmak için kullanılır.

f=0 olduğu ve düzeltmenin gerekmediği durumlarda,

$$\Sigma_3\ \Sigma_2\ \Sigma_1\ \Sigma_0 = S_3S_2S_1S_0$$

olur. Birden çok basamaklı onlu sayıların toplanması durumunda, her bir basamağı toplayan ayrı bir BCD toplayıcıya ihtiyaç vardır. Bu durumdaki onlu sayıları toplama işlemi, her bir basamak için bir toplayıcı kullanmak ve toplayıcıları birbirleriyle ilişkilendirmek suretiyle gerçekleştirilir.



Şekil 8.75. Dört bitlik BCD toplayıcı ve düzeltme devresi.

vi - Çarpma Devresi

İkili sayılarda çarpma işlemi, onlu sayılardaki çarpma işlemi gibi yapılır. Bunun yanında, çarpma işleminin tekrarlanan toplama işlemleri ile yapılması mümkündür. Çarpma işleminin tekrarlanan toplama işlemleri yöntemi ile yapılması 7485 ve benzeri entegrelerle gerçekleştirilebilir.

Örnek 18 : Çarpma işlemi yapan lojik devre tasarımına örnek olması amacıyla ikili bir sayıyı kendisi ile çarpan yani karesini alan devreyi tasarlayalım.

Tasarlanacak devre, iki bitlik sayıların çarpımını yapacağından iki girişe sahip olur. Çıkışında ise $2^2 = 4$ değişik durum bulunur. Çıkış bit sayısını bulmanın en kolay yolu, giriş değişkenlerinden en büyük değerli olanını alıp, bu sayının karesini bulmaktır. Bulunan sayının en az kaç bit ile temsil edilebileceği tespit edilir. Böylece çıkış bit sayısı bulunur.

Girişler		Çıkışlar			
A	B	F_3	F_2	F_1	F_0
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

$F_0 = A^1B + AB = B\,(A^1 + A)$

$F_1 = 0$

$F_2 = AB^1$

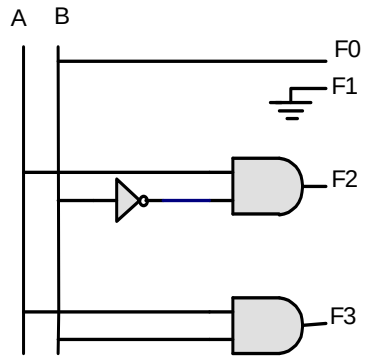
$F_3 = AB$

Şekil 8.76. İki bitlik sayının karesini alan lojik devrenin doğruluk tablosu ve çıkış eşitlikleri.

Yapılan örnekte en büyük sayı, $(11)_2$ ikili sayısı ile temsil edilen $(3)_{10}$ sayısıdır. Bu sayının karesi olan $(9)_{10}$ sayısı en az dört bit ile yazılacağından, devrenin çıkışı 4 adet olarak bulunur. Her bir çıkış, ikili olarak farklı basamak değerini temsil eder. Girişlere bağımlı olarak '1' değerini alacak çıkışlara '1' yazılırsa, Şekil 8.76'daki doğruluk tablosu ve çıkış ifadeleri oluşur.

Doğruluk tablosundan elde edilen fonksiyonlar sadeleştirilerek, Şekil 8.77'deki lojik devre elde edilir.

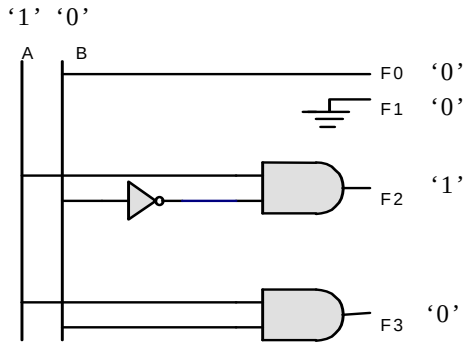
GİRİŞLER		ÇIKIŞLAR			
A	B	F3	F2	F1	F0
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1



Şekil 8.77. İki bitlik ikili sayının karesini alan lojik devre şeması.

Örnek 19: İki bitlik ikili sayının karesini alan devrede $(10)_2$ sayısının karesini alalım.

Giriş sayısı A ve B girişlerine uygulanırsa, girişlere göre oluşacak çıkışlardaki değerler, giriş sayısının karesidir.

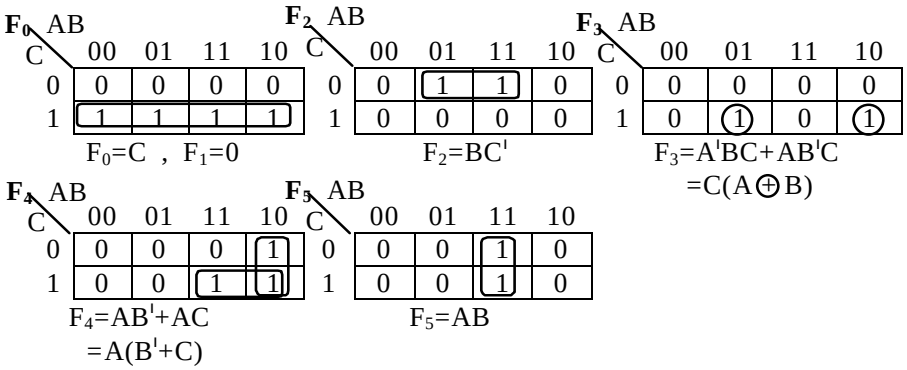


Şekil 8.78. $(10)_2$ sayısının karesi olan $(01000)_2$ sayısının elde edilmesi.

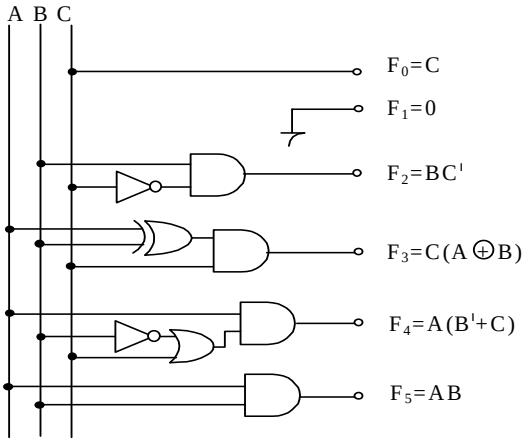
Örnek 20 : Üç bitlik ikili sayının karesini alan devreyi tasarlayalım.

Üç bitlik ikili sayı ile temsil edilen en büyük onlu sayı $(7)_{10}$ olduğundan sonuç olarak en yüksek $(49)_{10}$ sayısı bulunur. Bu sayı ikili sistemde altı basamak ile temsil edilebilir.

Girişler			Çıkışlar					
A	**B**	**C**	**F_5**	**F_4**	**F_3**	**F_2**	**F_1**	**F_0**
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1



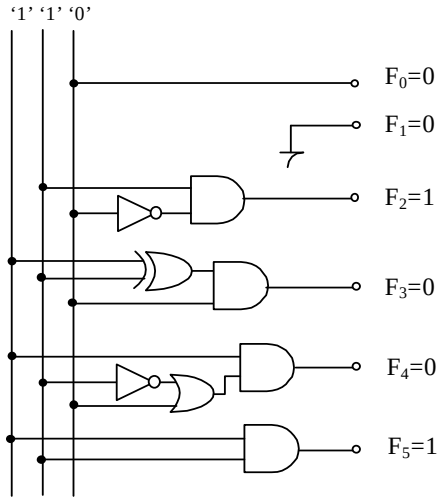
Şekil 8.79. Üç bitlik sayının karesini alan devrenin doğruluk tablosu ve Karnaugh haritaları.



Şekil 8.80. Üç bitlik ikili sayının karesini alan devrenin lojik şeması.

Bu durumunda, tasarımda oluşturulacak devrenin 6 çıkışı bulunmalıdır. Giriş değerlerine göre oluşacak sayıların doğruluk tablosunda gösterilmesi ile, Şekil 8.79'daki tablo oluşur. Doğruluk tablosu çıkış sütunlarındaki değerler Karnaugh haritaları kullanılarak sadeleştirilir ve elde edilen fonksiyonları temsil eden lojik devreler çizilirse, Şekil 8.80'deki lojik devre elde edilir.

Örnek 21 : Üç bitlik ikili sayının karesini alan devreye $(110)_2$ sayısı uygulanması durumunda oluşacak çıkış değerlerini bulalım.
Üç bitlik sayının karesini alan Şekil 8.81'deki devreye giriş sayısı uygulanması durumunda, çıkışlarda oluşacak değerler girişteki sayının karesidir.



Şekil 8.81. $(110)_2$ sayısının karesinin alınması.

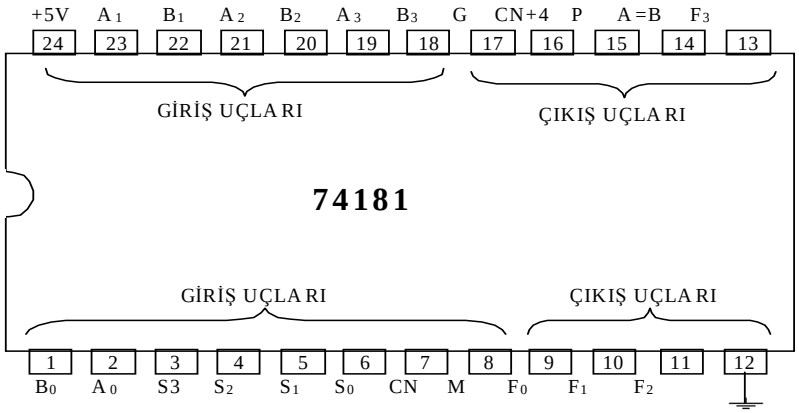
Çarpma işlemi, işlemin hızlı olarak yapılması amacıyla paralel ikili çarpma devreleri ile gerçekleştirilir. 74284 Entegresi, paralel ikili çarpma işlemi için kullanılan entegrelere örnek olarak verilebilir.

Temel aritmetik işlemlerin (toplama, çıkarma, çarpma) ve lojik işlemlerin (VE, VEYA, VEDEĞİL, v.b) yapılmasında kullanılan çok fonksiyonlu Aritmetik Mantık Devresi entegreleri bulunmaktadır. Bu entegrelere örnek olarak 74181 Entegresi verilebilir (Şekil 8.82). 74181 entegresi, aritmetik veya mantık işlemlerinden birisini yapacak şekilde programlanabilir. Bu işlem entegrenin 8 nolu M giriş ucu ile yapılır. Entegre, M=0 iken aritmetik işlemleri, M=1 iken ise lojik işlemleri yapmak üzere programlanır.

74181 entegresindeki uçlardan 4 tanesi A sayısı girişi (A_3, A_2, A_1, A_0), 4 tanesi B sayısı girişi (B_3, B_2, B_1, B_0), 4 tanesi de A ve B sayıları arasında yapılan aritmetik ve lojik işlemler sonucu ortaya çıkan değerlerin (F ile gösterilen) çıkışı (S_0, S_1, S_2, S_3) için kullanılır. S_0, S_1, S_2, S_3 dizisi ile yapılacak işlem (toplama, çıkarma, AND, OR, v.b.) tanımlanır.

Örneğin; M=1 ve $S_3S_2S_1S_0$=1110 ise F=A+B işlemi gerçekleştirilir. Diğer bir değişle, A girişleri ile B girişleri 'VEYA' işlemine tabi tutulur.

Bu girişlerin farklı kombinasyona sahip olması ile yapılabilecek işlemler, fonksiyon tablosu olarak entegre veri kataloglarında verilmektedir. 74181 Fonksiyon tablosuna göre, entegreyi beş çeşit aritmetik işleme, 16 tip lojik işleme ve 8 tip hem lojik hem de aritmetik işleme programlamak mümkündür. Bunların anlamı; 74181 entegresi ile 32 çeşit işlem yapmanın mümkün olduğudur.



Şekil 8.82. 74181 Aritmetik – Lojik işlem entegresi ayak bağlantısı.

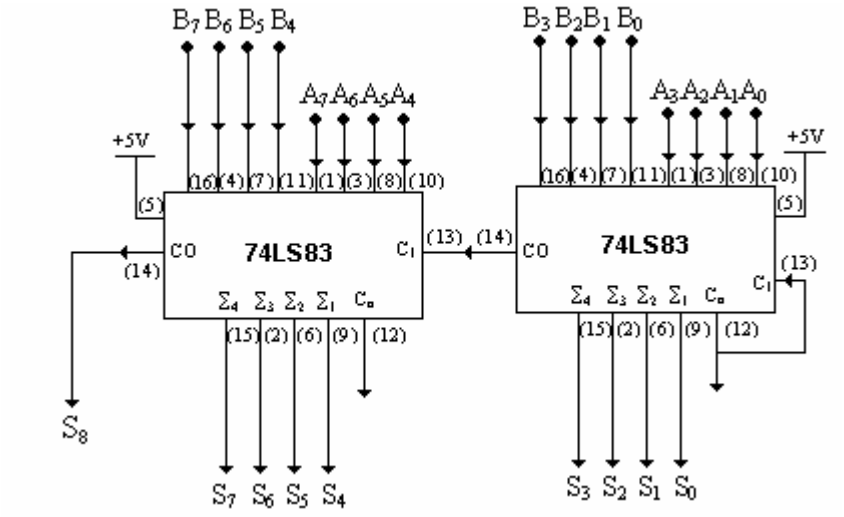
8.5.3. Aritmetik İşlemler ile İlgili Uygulamalar

Aritmetik işlemleri gerçekleştiren ve mikroişlemcilerin Aritmetik-Mantık Birimlerini (ALU) oluşturan devreler, farklı elemanlar kullanılarak değişik şekillerde gerçekleştirilebilir. Aritmetik işlemleri gerçekleştiren devrelere örnek vererek, bunların çalışma prensiplerini inceleyelim.

8.5.3.1. Paralel Toplayıcı (8 bit) Devresi

Paralel toplama işlemlerinde yaygın olarak kullanılan elemanlardan birisi, 7483 entegresidir. İki adet 7483 entegresi kullanılarak 8 bitlik paralel toplayıcı gerçekleştirilebilir (Şekil 8.83). Oluşturulan devrede, düşük değerli bitleri toplayan entegrenin elde (Ci) girişi '0' yapılmalıdır. İkinci entegrenin elde girişi ise, düşük değerli bitleri toplayan entegrenin elde

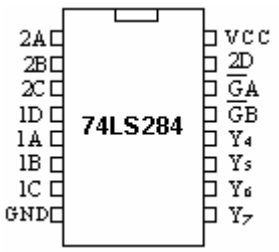
çıkışına (Co) bağlanır. Bu bağlantı ile her biri 8 bitlik A ve B sayıları toplanır, sonuçta 8 bitlik toplam çıkışları ile elde çıkışı üretilir.



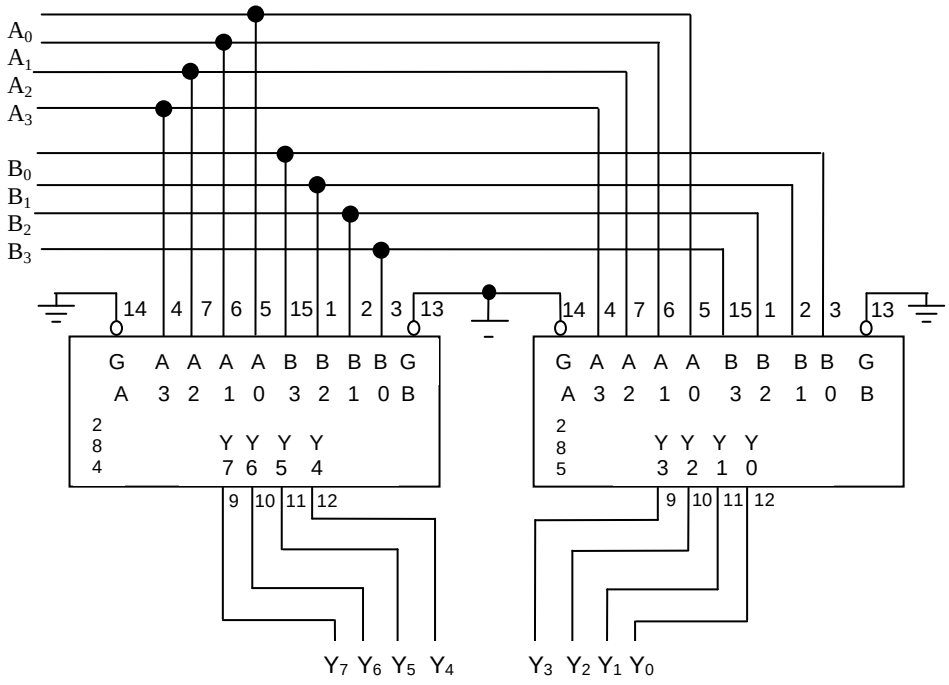
Şekil 8.83. Sekiz bit paralel toplayıcı devresi oluşturulması.

8.5.3.2. Dört Bitlik Sayıların Çarpılması

Piyasada iki bitlik çarpma işlemleri gerçekleştiren 74284 ve 74285 entegreleri bulunmaktadır (Şekil 8.84). Aynı işlemi gerçekleştiren ve birbirinin benzeri ayak bağlantılarına sahip olan 74284 ve 74285 entegreleri birlikte kullanılarak dört bitlik iki sayının çarpımı işlemini gerçekleştirmek mümkündür. Dört bitlik iki sayının çarpımı için iki entegre girişleri birbirine paralel bağlanır. Şekil 8.85'deki prensip şeması çizilen devrede, çarpma işlemi sonucunda elde edilen sayının ilk dört biti 74284 entegresi çıkışlarından, son dört biti ise 74285 entegresi çıkışlarından alınır.



Şekil 8.84. 74LS284 entegresi ayak bağlantıları



Şekil 8.85. 4x4 bitlik çarpma devresi

8.5.3.3. Aritmetik Mantık Birimi Devreleri

Aritmetik Mantık Birimini oluşturmanın en basit yöntemlerinden birisi 74181 entegresi kullanmaktır (Şekil 8.82). 74181 entegresi, iki adet dört bitlik giriş üzerinde 16 aritmetik işlem veya 16 mantık işlemi gerçekleştirilebilir.

Tekrarlama Ve Çalışma Soruları

1. Bileşik devreyi tanımlayınız.

2. Bileşik devreler kaç gruba ayrılır?

3. Bileşik devre tasarımında takip edilecek işlem sırasını sıralayınız.

4. Bileşik bir devrenim sadeleştirilmesin sonra taşıması gerekli özellikleri nelerdir?

5. Bilgisayarlarda yapılan kodlama ile ilgili işlemleri blok şema olarak gösteriniz.

6. ‘Kodlayıcı devre’ ne demektir ?

7. 4 girişten 2 çıkışa kodlayıcı devrenin açık şemasını çizerek, çalışmasını açıklayınız.

8. Kodlayıcı devrelerde çıkış sayının tespit edilmesi işlemini özetleyiniz.

9. Üç girişli bir kodlayıcı devresinin çalışmasını lojik devre üzerinde açıklayınız.

10. Öncelikli kodlayıcı devrenin özelliğini özetleyiniz.

11. 74147 entegresi kullanarak BCD kodlayıcı tasarımını yapınız.

12. Onlu'dan ikili'ye en basit kodlayıcı devresi nasıl gerçekleştirilir.

13. Diyot matris'le yapılan onlu'dan BCD'ye kodlayıcı devresinin şeklini çizerek, çalışmasını açıklayınız.

14. Kod çözücü devreyi tanımlayınız

15. 2 girişli kod çözücü devrenin doğruluk tablosunu ve lojik devresini çiziniz.

16. 4 girişli / 10 çıkışlı kod çözücü devrenin lojik devresini çiziniz.

17. Bilgisayarlarda kullanılan kodlayıcı ve kod çözücü devrelere örnekler veriniz.

18. Entegrelerde bulunan yetkilendirme girişinin işlevini açıklayınız.

19. Kod çevirici devreyi tanımlayınız.

20. 5'te 2 kodundan, BCD'ye kod çevirici devrenin doğruluk tablosunu oluşturunuz.

21. BCD'den 7 parçalı göstergeye kod çevirici devrenin tasarım aşamalarını özetleyiniz.

22. 'Çoklayıcı' devreyi tanımlayınız.

23. 'Multiplexing' olayını açıklayınız.

24. 4x1 MUX devresi doğruluk tablosunu oluşturarak, lojik şemasını çiziniz.

25. 8x1 MUX devresi lojik şemasını çiziniz.

26. Multiplexer'lerin giriş sayısı nasıl arttırılabilir?

27. İki adet 4x1 MUX kullanarak 8x1 MUX oluşturunuz.

28. 4 Adet 2x1 MUX kullanarak 2 Adet 4x1 MUX tasarımı yapınız.

29. Multiplexer uygulama alanlarına örnekler veriniz.

30. $F_{(A,B,C)} = \Sigma\ (0,2,4,6)$ fonksiyonunun 4x1 MUX ile gerçekleştiriniz.

31. $F_{(A,B,C)} = \Sigma\ (2,5,6,7)$ fonksiyonunun 4x1 MUX ile gerçekleştiriniz.

32. $F_{(A,B,C)} = \Sigma\ (1,7,8,9,13,15)$ fonksiyonunun 8x1 MUX ile gerçekleştiriniz.

33. Multiplexer'in veri yönlendirme amacıyla kullanımına örnek veriniz.

34. Multiplexer'in bileşik devre tasarımında kullanımını bir örnekle açıklayınız.

35. ‘Azlayıcı’ devresini tanımlayınız.

36. Demultiplexer’ın yaptığı işi, fonksiyon şeması yardımı ile açıklayınız.

37. 1x8 Demultiplexer’a ait doğruluk tablosu oluşturunuz.

38. 1x8 Demultiplexer lojik şemasını çiziniz.

39. Bir entegrenin hem kod çözücü hem de demultiplexer olarak kullanılmasını örnek ile açıklayınız.

40. Demultiplexer uygulama alanlarına örnekler veriniz.

41. Kıyaslama devrelerini tanımlayarak, gruplandırınız.

42. Büyüklük karşılaştırıcıyı tanımlayınız

43. Tek bitlik iki sayının eşitlik durumunu belirleyen bileşik devrenin lojik şemasını ve doğruluk tablosunu çiziniz.

44. İki biti karşılaştırarak; $A=B_1$ $A>B$ ve $A<B$ çıkışlarını üreten bileşik devreyi tasarlayınız.

45. Dört bitlik iki sayının karşılaştırılmasından oluşan doğruluk tablosunu oluşturunuz.

46. Karşılaştırıcıların kaskat bağlantısını örnekle açıklayınız.

47. Aritmetik işlem devrelerini tanımlayınız.

48. Yarım toplayıcı ve tam toplayıcı devrelerinin tanımlayınız.

49. Yarım toplayıcı doğruluk tablosunu çizerek, çıkış sütunlarındaki değerlere göre toplama işlemini gerçekleştirecek lojik devreyi çiziniz.

50. Tam toplayıcı devresi doğruluk tablosunu oluşturarak, doğruluk tablosundaki çıkış değerlerini sadeleştirip, toplam ve elde eşitliklerini yazınız.

51. Tam toplayıcı devreyi, iki adet yarım toplayıcı kullanarak oluşturmayı açıklayınız.

52. Paralel toplama işlemini 4 bitlik iki sayının toplanması işlemini kullanarak açıklayınız.

53. 10110+11011 işlemini paralel toplayıcı kullanarak yapınız.

54. Yarım çıkarıcı ve Tam çıkarıcı terimlerini açıklayınız.

55. Yarım çıkarıcı lojik devresini tasarlayınız.

56. Tasarladığınız devreye 1-0 uygulanması durumunda oluşacak çıkış değerlerini gösteriniz.

57. Tam çıkarıcı devresi sembolü ve doğruluk tablosunu çiziniz.

58. Tam çıkarıcı devresi doğruluk tablosundaki değerleri sadeleştirerek sadeleştirilmiş

eşitliği lojik kapılar ile gerçekleştiriniz.

59. İki yarım çıkarıcı kullanarak, tam çıkarıcı devresi oluşturunuz.

60. A=B=1 ve C=0 giriş değerlerine sahip tam toplayıcı devrenin çıkış değerlerini, lojik şema üzerinde gösteriniz.

61. Dört bitlik iki sayıyı paralel olarak devrenin blok şemasını çiziniz.

62. (11001-10110) işlemini paralel çıkarıcı ile yaparak, çıkış değerlerini lojik şema üzerinde gösteriniz.

63. 'İki tümleyeni ile çıkarma' işlemini özetleyiniz.

64. (-5) ve (+9) sayılarını, iki tümleyeni yöntemi kullanarak paralel toplayıcı kullanarak yapınız.

65. (+8) – (+5) işlemini iki tümleyeni yöntemi ile paralel toplayıcı ile yapınız.

66. 'BCD toplayıcı' nedir?

67. (317+623) işlemini BCD toplayıcı ile yapınız.

68. İkili sayılarla çarpma işlemini gerçekleştirmek için kullanılan yöntemler nelerdir?

69. İki bitlik ikili sayının karesini alan lojik devreyi tasarlayınız.

70. 74181 entegresi özelliklerini özetleyiniz.

BÖLÜM 9

MULTİVİBRATÖR VE FLİP – FLOPLAR

(FLIP-FLOPS)

Giriş

Kare veya dikdörtgen sinyal üreten elektronik devreler '**Multivibratör**' olarak adlandırılır. Multivibratörlerin ürettiği sinyaller, dijital devrelerde tetikleme sinyali olarak kullanılır. Multivibratörler üç gruba ayrılır:

i- Serbest çalışan (Astable) multivibratörler,

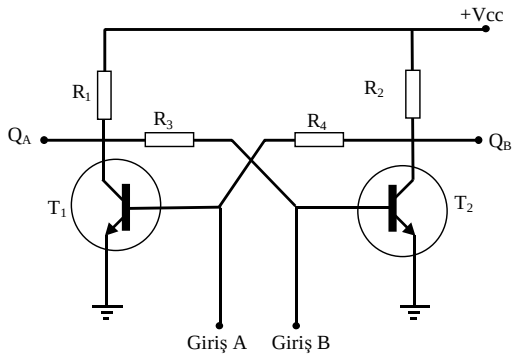
ii- Tek kararlı (Monostable) multivibratörler,

iii- Çift kararlı (Bistable) multivibratörler.

Dışarıdan bir müdahale gereksinim duymadan, belirli aralıklarla durum değiştirerek sinyal üreten multivibratör, '**serbest çalışan multivibratör**' olarak isimlendirilir.

Dışarıdan uygulanan uygun bir sinyal ile durumunu geçici bir süre değiştirip, bir süre sonra tekrar eski durumuna dönen multivibratör devresi, '**tek kararlı multivibratör**' olarak adlandırılır. Bu tip multivibratörde, girişe uygulanan sinyal ile devrenin eski durumuna dönme süresi ayarlanarak, istenilen şekilde kare veya dikdörtgen dalga üretilebilir.

Dışarıdan bir müdahale yapılmadığı sürece bulunduğu durumu sonsuza dek koruyan multivibratör devresi, '**çift kararlı multivibratör**' olarak adlandırılır. Bu bölümde inceleyeceğimiz flip-flop devreleri çift kararlı multivibratör uygulamaları olduğundan, öncelikle bu tip multivibratörü oluşturan elektronik devreyi inceleyelim.



Şekil 9.1. Çift kararlı multivibratör devresi.

Şekil 9.1'de elektronik devresi görülen çift kararlı multivibratör devresine dışarıdan bir etki yapılmadığı sürece transistörler durumlarını ve çıkışlarda sahip oldukları değerlerini korurlar. Devrede iki transistör aynı çalışma durumunda bulunamaz. Devrenin çalışmasını açıklamaya yardımcı olması amacıyla T_1'in iletimde, T_2'nin yalıtımda olduğunu kabul edelim.

T_1 iletimde iken, T_2'nin beyzi R_3 üzerinden şase potansiyelindedir (yaklaşık 0 V) ve bu nedenle yalıtkandır. Dışarıdan uygulanacak bir etki (sinyal) ile T_1 transistörü yalıtıma götürülürse, T_1'in kolektöründe R_1 üzerinden +V gerilimi gözükür. Bu durumda lojik '0' değerine sahip olan Q_A çıkışı durum değiştirir ve Q_A=1 değerini alır.

T_1'in kollektöründeki gerilim R_3 üzerinden T_2 transistörünün beyzine uygulandığından, T_2 iletime geçer. T_2'nin kolektörü şase potansiyeline gelir ve bu noktaya bağlı olan T_1 transistörünü yalıtımda tutar. Bu durum, T_2'nin girişine bir etki uygulanıp yalıtıma götürülünceye kadar devam eder.

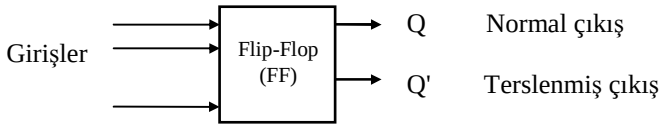
Çalışması anlatılan devre, dijital elektronikte yaygın olarak kullanılan ve bu nedenle geniş olarak inceleyeceğimiz Flip-Flop (FF) devresinin temelini oluşturur.

9.1. Flip-Flop'lar ve Flip-Flop Çeşitleri

Devreye çalışma gerilimi uygulandığı sürece durumunu devamlı olarak koruyabilen multivibratör çeşidi, '**Flip-Flop**' olarak adlandırılır ve 'FF' harfleri ile sembolize edilir.

Lojik kapılarla oluşturulan flip-flop'lar, lojik devrelerde kullanılan en önemli bellek elemanlarıdır. Flip-flop'ları oluşturan lojik kapılar normalde kendi başlarına bilgi saklama kapasitesine sahip değillerdir. Ancak, birkaç tane kapı devresi bilgi saklama işlemi oluşturacak şekilde bağlanarak bilgi saklama kapasitesi oluşturulabilir.

Bir bitlik bilgi saklama yeteneğine sahip flip-flop devrelerinde, biri saklanan bilginin (bitin) normal değerine, diğeri tümleyen değerine sahip iki çıkış bulunur. Şekil 9.2'de sembolü görülen FF, Q ve Q' olarak isimlendirilen ve birbirinin tersi olan iki çıkışa sahiptir. **'Q çıkışı normal çıkış, Q' ise terslenmiş çıkış'** olarak kabul edilir. Çıkış dendiği zaman, Q çıkışı referans alınır. Örneğin; FF'nin çıkışı '1' dendiği zaman, Q=1 kastediliyordur ve bu durumda Q'=0 değerine sahiptir. İki farklı konumdan (0 veya 1) birisini alabilen FF'de, çıkışların alacağı değerleri FF'nin konumunu değiştirme kapasitesine sahip girişler belirler. FF'nin çıkış durumunu değiştirmek için, girişin tetiklenmesi gerekir. FF'nin çıkışını değiştiren tetikleme darbesinin sona ermesinden sonra, FF'nin çıkışı konumunu korur. Bu durum FF'nin bellek özelliği göstermesini sağlar. Diğer bir deyişle, tetikleme sinyali ile FF'nin durumu değiştirilmediği sürece FF durumunu sonsuza kadar koruyabilir.



Şekil 9.2. Flip - Flop devresi genel sembolü.

FF'ler **'Latch'** veya **'çift kararlı multivibratör'** gibi diğer isimlerle de adlandırılır. 'Latch' terimi belirli tip FF'ler için kullanılırken, 'çift kararlı multivibratör' (bistable multivibrator) terimi FF'ler için kullanılan daha teknik bir terimdir.

İkili bilgilerin flip-flop'a farklı şekillerde uygulanması sonucu, flip-flop çeşitleri oluşur. Diğer bir deyişle; temel FF devresinin bağlantısının değiştirilmesi ve yeni özellikler eklenmesi ile oluşturulan çeşitli FF türleri bulunmaktadır. Bunlardan en çok kullanılanlar;

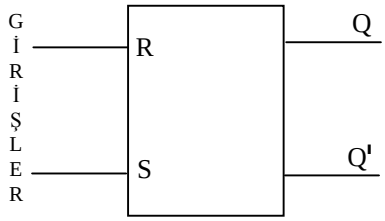
1-) R - S (Reset - Set) tipi FF.
2-) Tetiklemeli (clocked) R- S FF.
3-) J - K Tipi FF.
4-) Ana - Uydu (Master - Slave) Tipi FF.
5-) D (Data) Tipi FF.
6-) T (Toggle) Tipi FF.

olarak sıralanabilir. En çok kullanılan FF çeşitlerini ayrıntılı olarak inceleyelim.

9.1.1. RS Flip-Flop Devresi

İki çıkışa sahip FF'de, iki farklı çıkış durumu ortaya çıkar: Q=0, Q'=1 ve Q=1, Q' =0 durumları. Q=0 ve Q'=1 olduğu durum, çıkış '0' veya **'reset-sıfırla'** durumu olarak tanımlanır. Oluşabilecek ikinci durumunu ifade eden Q=1 ve Q'=0 durumu, çıkış '1' veya **'set-kur'** durumu olarak adlandırılır.

Bu durumda, FF'nin Q=0 olmasını sağlayan girişi 'Reset-R', Q=1 durumunu oluşturan girişi ise 'Set-S' olarak düşünülebilir. Çıkışların birbirinin tersi olduğu bu durumlar, normal çalışma durumları olarak kabul edilir. FF girişlerinin 'set-kur (S)' ve 'reset-sil (R)' olarak isimlendirilmesi ile, R-S FF olarak adlandırılan FF türü ortaya çıkar (Şekil 9.3). RS tipi FF; Q=1 ve Q'=0 iken '**set'** (kurma), Q=0 ve Q'=1 olduğunda ise '**reset'** (silme) durumundadır.

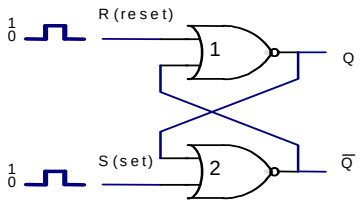


Şekil 9.3. RS tipi FF sembolü.

Temel FF devresi, iki 'VEDEĞİL' veya iki 'VEYADEĞİL' kapısı ile gerçekleştirilebilir. 'VEDEĞİL' kapılarıyla yapılan devre 'VEDEĞİL' **latch'** veya kısaca **'lanch'** olarak, 'VEYADEĞİL' kapıları kullanılarak oluşturulan devre ise '**VEYADEĞİL lanch'** diye isimlendirilir. Her iki tip 'lanch' devresinde, kapılardan birinin çıkışı diğerinin girişine çapraz şekilde bağlanmıştır. Q ve Q' olarak isimlendirilen kapı çıkışları, 'lanch' çıkışlarıdır (Şekil 9.4).

'VEYADEĞİL' kapısının girişlerinden birisinin '1' olması, çıkışının '0' olması için yeterlidir. Her iki girişin '0' olması durumunda, çıkış '1' olur. 'VEYADEĞİL' kapısının özelliğini hatırlattıktan sonra, 'VEYADEĞİL' kapıları ile oluşturulan RS FF devresinin çalışmasını inceleyelim (Şekil 9.4).

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	S=1, R=0 dan sonra
0	1	0	1	
0	0	0	1	S=0, R=1 den sonra
1	1	0	0	(yasak)



Şekil 9.4. 'VEYADEĞİL' kapıları ile yapılan RS-FF devresi ve doğruluk tablosu.

'VEYADEĞİL' kapılarıyla oluşturulan FF devresinde, FF'nin durumu değiştirilmediği sürece her iki giriş '0' değerine sahiptir (Şekil 9.4). Set girişine '1' uygulanması, 2 nolu 'VEYADEĞİL' kapısının çıkışının ve 1 nolu 'VEYADEĞİL' kapısının girişlerinin birisinin '0' olmasını sağlar. Bu anda 1 nolu 'VEYADEĞİL' kapısının her iki girişi '0' değerini alır. Her iki girişi '0' olan 1 nolu 'VEYADEĞİL' kapısı çıkışı '1' olur. Bu durum, FF'nin 'set' durumuna (Q=1) geçmesine neden olur.

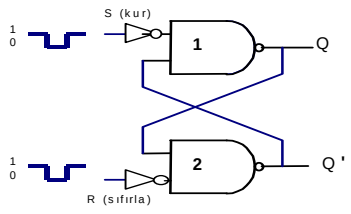
Reset girişine '1' uygulanması; 1 nolu 'VEYADEĞİL' kapısının çıkışının ve 2 nolu 'VEYADEĞİL' kapısının girişinden birisinin '0' olmasına neden olur. Aynı anda diğer girişi de '0' olan 2 nolu 'VEYADEĞİL' kapısının çıkışı '1' değerini alır. Bu durum, FF'nin Q=0 (reset) durumuna geçmesini sağlar.

Her iki girişe '0' uygulanması, FF set durumunda iken set girişine '1' uygulanması veya FF reset durumunda iken reset girişine '1' uygulanması durumları, FF' un konumunu (çıkışları) değiştirmez. Her iki girişin '0' olduğu durum, **'belirsizlik'** olarak tanımlanır ve bu durumda FF en son bulunduğu konumu korur. Her iki girişe '1' uygulanması durumunda ise iki çıkışta 0'a gider. **'Tanımsız'** olarak isimlendirilen bu durumdan kaçınılması gerekir.

RS FF, 'VEYADEĞİL' kapıları ile oluşturulmasına benzer şekilde, 'VEDEĞİL' kapıları ile de oluşturulabilir. Şekil 9.5'de 'VEDEĞİL' kapılarıyla oluşturulan RS FF devresi ve doğruluk tablosu görülmektedir. 'VEDEĞİL' kapısının girişlerinden biri '0' olduğu durumda çıkış '1' değerini alır. Devrenin çalışmasını anlamak için, farklı giriş durumlarında oluşacak olayları inceleyelim:

Girişlere S=0, R=1 uygulanması durumunda, **2** nolu 'VEDEĞİL' kapısının girişlerinden birisi '0' olur. Girişlerden birisinin '0' olması, ilgili 'VEDEĞİL' kapısının çıkışının '1' olması için yeterlidir. Bu durumda Q'=1 değerini alır. Bu değer ile 1 nolu 'VEDEĞİL' kapısının her iki girişi '1' olacağından, Q=0 çıkışı oluşur. Sonuç olarak; S=0, R=1 giriş değerlerinde çıkışlarda Q=0, Q'=1 değerleri okunur. Q=0 ve Q'=1 oluşmasını sağlayan işlem, '**sıfırlama-(resetting)**' olarak, Q=0 durumu ise '**sıfır (reset) durumu**' olarak isimlendirilir.

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	S=1, R=0 dan sonra
0	1	0	1	
0	0	0	1	S=0, R=1 den sonra
1	1	1	1	(yasak)



Şekil 9.5. 'VEDEĞİL' kapıları ile oluşturulan RS-FF devresi ve doğruluk tablosu.

'VEDEĞİL' kapısı ile oluşturulan FF devresi girişlerine S=1 ve R=0 değerleri uygulandığında; 1 nolu 'VEDEĞİL' kapısının 'DEĞİL' kapısına bağlı olan girişi '0' olur ve girişlerinden birisi '0' olan bu kapının çıkışı '1' değerlerini alır. Bu durumda, 2 nolu 'VEDEĞİL' kapısının her iki girişi '1' olduğundan Q'=0 değeri oluşur. Q=1 ve Q'=0 konumunda iken, R=1 olsa bile NAND–1'in çıkışı '1', NAND-2'nin çıkışı '0' değerlerini saklar. FF devresinde S=1 olması ile oluşan Q=1 durumu; **'kurma (set) durumu',** Q=1 olmasını sağlayan işlem ise; **' kurma (setting)'** olarak isimlendirilir.

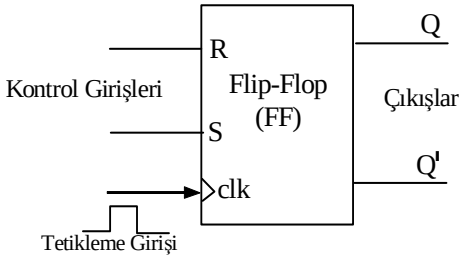
Q=0 ve Q'=1 iken; S=0 ve R=0 girişleri uygulanırsa, Q=0 değeri (çıkışı) 'VEDEĞİL'-2'nin çıkışının '1' konumunda kalmasını sağlar. Bu durum; Q=0 ve Q'=1 iken, S=R=0 yapılması devrede herhangi bir değişiklik yapmaz diye özetlenebilir.

Her iki girişin aynı anda '1' olması durumda Q=Q'=1 olur ve istenilmeyen bir durum ortaya çıkar. **'Yasak durum'** olarak adlandırılan bu durumdaki çalışmadan kaçınmak gerekir. Bu durum, S=R=1 durumu 'VEDEĞİL' ile yapılan FF'lerde kullanılamaz diye özetlenebilir.

Burada açıklanan ve RS (reset-set) Flip-Flop olarak isimlendirilen FF bağlantısının değiştirilmesi ve yeni özellikler eklenmesi ile diğer FF çeşitleri oluşturulur.

9.1.2. Tetiklemeli - Saatli (Clocked - Triggered) R- S Tipi Flip- Flop

FF'nin konumunun değiştirilmesi işlemi, **'tetikleme'** olarak adlandırılır. Tetikleme girişinin bulunmadığı, FF durumlarının kontrol girişlerindeki bilginin değişmesi anında değiştiği FF'ler 'Asenkron FF'ler' olarak isimlendirilir. Senkron sistemlerde herhangi bir çıkışın değerinin değişebileceği zamanlar, **'tetikleme sinyali'** (clock - saat) adı verilen, kare veya dikdörtgen şeklindeki sinyal tarafından belirlenir. Tetikleme sinyali olarak kullanılan sinyalin aktif durum (kenar) değişimi 0'dan 1'e ise buna **'pozitif kenar değişimi'** (positive going transition - PGT), aktif durum değişimi 1'den 0'a doğru ise **'negatif kenar değişimi'** (NGT) denir.

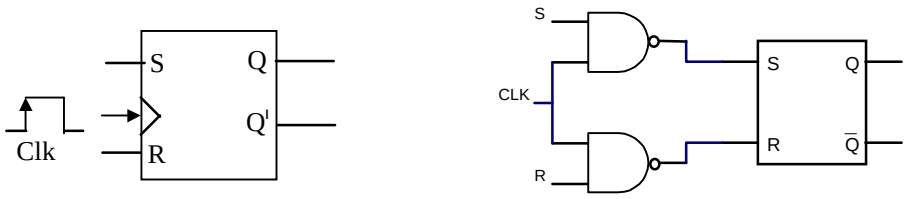


Şekil 9.6. Tetiklemeli RS FF sembolü.

'Tetikleme sinyali' olarak isimlendirilen sinyal, tetiklemeli FF'lerde bulunan üçüncü girişe uygulanır. Tetikleme işleminin oluşması için gerekli tetikleme sinyalini FF'ye uygulamak

amacıyla FF'de oluşturulan üçüncü giriş, **'tetikleme girişi'** olarak adlandırılır (Şekil 9.6). Tetikleme girişi, 'clk' harfleri ile gösterilir ve **'clk girişi'** olarak da isimlendirilir. Tetikleme girişinin eklenmesiyle senkron ardışıl devre durumuna gelen FF devresinde, R-S girişlerinin değişmesi çıkışları hemen değiştiremez. Çıkışların değişmesi, tetikleme girişine uygulanan işarete bağımlıdır.

Tetiklemeli RS FF'ler, RS FF devresinin girişlerine kapı devreleri eklemek suretiyle elde edilir. Tetikleme girişi eklenmiş R-S FF devresinin sembolü ve 'VEDEĞİL' kapılarıyla tetikleme girişi oluşturulması işlemi Şekil 9.7'de görülmektedir. Yapılan işlem, RS FF'nin girişlerine 3. giriş oluşturacak şekilde 'VEDEĞİL' kapıları bağlanmasıdır.



Şekil 9.7. Tetiklemeli R-S FF'nin sembolü ve 'VEDEĞİL' girişli tetiklemeli RS FF oluşturulması.

Şekil 9.8.a ve 9.8.b'de 'VEDEĞİL' kapılarıyla oluşturulan tetikleme girişinin, 'VEYADEĞİL' ve 'VEDEĞİL' kapılarıyla oluşturulan RS FF devresine eklenmesi ile oluşan tetiklemeli RS FF devreleri görülmektedir. Şekil 9.8.c'deki doğruluk tablosunda, farklı S-R giriş durumlarında 'Clk' sinyalinin etkisi açıklanmaktadır. Şekil 9.8.d'de S ve R girişlerinin aldıkları farklı değerler ve 'Clk' sinyali ile Q çıkışının aldığı durumlar dalga şekilleri olarak gösterilmektedir.

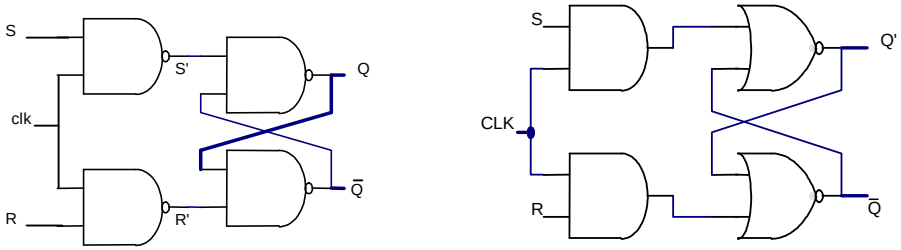
'VE' ve 'VEYADEĞİL' kapıları ile oluşturulan tetiklemeli RS FF'de oluşan olaylar aşağıdaki şekilde özetlenebilir:

Başlangıçta R=S=0 ve Q=0 iken; Clk=1 değerini alsa bile, FF'nin durumu değişmez ve Q=0 durumu devam eder. Bu durum, R=S=0 girişlerinde FF mevcut durumunu korur şeklinde özetlenebilir.

S=1, R=0 girişlerinde, 'Clk' sinyalinin yükselen kenarı ile FF etkilenir ve Q=1 değerini alır. S, R girişleri durumlarını korurken, 'Clk' sinyali durum değiştirse dahi Q=1 durumu devam eder. S=1, R=0 durumunda iken, 'Clk' sinyalinin yükselen kenarının oluşturduğu Q=1 çıkışı, **'set'** durumu olarak isimlendirilir.

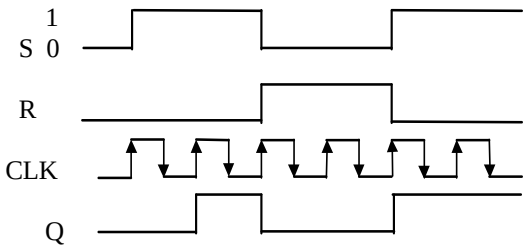
S=0, R=1 değerlerine sahip iken 'Clk' sinyalinin ilk yükselen kenarı ile Q=0, Q'=1 değerleri oluşur. Bu durum S=0, R=1 olduğu sürece devam eder ve **'sıfır (reset)'** konumu olarak adlandırılır.

S=1, R=1 durumu çıkışta belirsizlik oluşturacağından bu durumdan kaçınılmalıdır. Belirsiz çıkış, FF'nin kullanıldığı devrede düzensiz çalışmaya neden olur.



a)'VEDEĞİL' kapılı tetiklemeli R-S FF. **b)**'VE/VEYADEĞİL' kapılı tetiklemeli FF.

R	S	CLK	Çıkış Q
0	0	1	Q_0 (Değişiklik yok)
0	1	1	1
1	0	1	0
1	1	1	Belirsiz



c) Doğruluk tablosu. **d)** RS FF örnek durum değişimleri.

Şekil 9.8. Tetiklemeli R-S Flip-Flop lojik devreleri, karakteristik tablosu ve örnek durum değişimi.

Sonuç olarak; senkronize kontrol girişleri olarak isimlendirilen R ve S girişlerinin, 'Clk' sinyali yardımıyla FF'nin çıkış durumunu belirlediği söylenebilir. Negatif tetiklemeli FF'lerde oluşan doğruluk tablosu, pozitif kenar tetiklemeli R-S FF'lerin aynısıdır. Tek fark; durum değişmesini sağlayan işaretin 1'den 0'a giderken FF'nin konumunu değiştirmesidir. Şekil 9.8'de gösterilen her iki tip tetiklemeli R-S FF, çeşitli dijital devrelerde kullanılmaktadır.

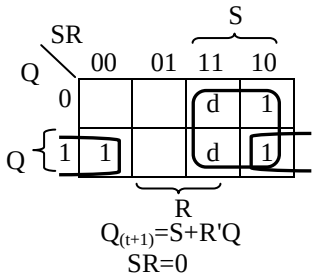
Tetiklemeli FF olarak D ve JK Flip-Flop'lar tercih edilmektedir. RS FF'lerin, tetikleme işleminden beklenen işlemleri gerçekleştirmemesi nedeni ile RS FF'ler tetiklemesiz asenkron ardışıl devrelerde kullanılır. Tetikleme işleminin bulunduğu devrelerde ise JK veya D tipi FF'ler kullanılır ve özellikle daha basit yapıda olması nedeni ile D FF'ler tercih edilir.

FF doğruluk tablosunda Q çıkışı olarak belirtilen değerler, FF'nin bulunulan andaki durumunu gösterir. S ve R girişlerinin mümkün olan kombinasyonlarına tetikleme sinyali uygulanması durumunda oluşan yeni çıkış durumları, **'Q(t+1) durumu'** olarak isimlendirilir.

Q değeri göz önünde tutularak (üçüncü giriş olarak kabul edilip) R-S değerlerine göre Q(t+1) çıkışı yazılırsa; Şekil 9.9.a'daki doğruluk tablosu oluşur. Oluşan doğruluk tablosu, '**FF karakteristik tablosu**' olarak isimlendirilir.

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

(a)



(b)

Şekil 9.9. Tetiklemeli R-S FF'nin karakteristik tablosu ve karakteristik denklemi.

Oluşan doğruluk tablosunun Karnaugh haritasına taşınması ile, Şekil 9.9.b'deki Karnaugh haritası ortaya çıkar. Doğruluk tablosunda bulunan belirsiz durumlar, Karnaugh haritasında farketmez-(d) olarak gösterilir. Karnaugh haritasında elde edilen eşitlik, önceki durum 'Q' ve SR girişlerinin fonksiyonu olarak sonraki durumun değerini vermektedir. Elde edilen eşitliğe SR=0 tanımlaması, S ve R girişlerinin aynı anda 1 değerini alamayacağını belirtmek için eklenmelidir. Elde edilen eşitlik, ilgili FF'ye ait '**karakteristik denklem**' olarak isimlendirilir.

9.1.3. J-K Flip-Flop (JK FF)

J-K Flip-Flop, RS FF'lerin belirsizlik durumunu ortadan kaldırmak amacıyla geliştirilmiş FF çeşididir. Tanımsız durumların tanımlı hale geldiği J-K FF'lerde, J ve K girişleri FF'i kurmak ve silmek için S ve R girişleri gibi davranır (Şekil 9.10.a). J girişinin S'ye, K girişinin R'ye denk geldiği J-K FF'lerde; J=K=1 durumunda iken, tetikleme sinyalinin her pozitif kenarı ile çıkış bir önceki durumun tersi değerini alır. Yani Q=1 ise 0'a, Q=0 ise 1'e geçiş olur. Bu işlem, **'toggle'** (ters çevirme) olarak isimlendirilir. Bu işlemin devamlı gerçekleştirilmesi için, J=1 ve K=1 durumunda bırakılır.

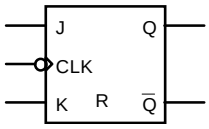
Şekil 9.10.b'deki doğruluk tablosu, J-K FF'nin çalışmasını özetlemektedir. JK FF doğruluk tablosu, her iki girişin '1' olduğu durum haricinde tetiklemeli RS FF doğruluk tablosunun aynısıdır.

Şekil 9.11.a'da lojik devre şeması görülen JK Flip-Flop devresinde, Q çıkışı K ve 'Clk' girişleriyle, Q' çıkışı J ve 'Clk' girişleriyle beraber 'VE' işlemine tabi tutulmuştur.

Q=0 iken, J=K=0 girişlerinin uygulanması ile çıkış bir önceki durumun aynısı olur. Bu durum Q(t+1) = Q(t) şeklinde özetlenebilir.

J=0, K=1 ve J=1, K=0 giriş durumlarında çıkış J değerlerini izler. Yani J=0, K=1 durumunda Q=0 ve J=1, K=0 durumunda Q=1 değerini alır.

J=0 ve K=1 iken; tetikleme sinyali ('clk' palsı) ile FF'nin Q=0 değerini alması durumu, **'Reset'** olarak tanımlanır. J=1, K=0 durumunda iken 'Clk' palsı ile Q=1 değerini alması durumu ise, **'Set'** olarak belirtilir. Farklı JK kombinasyonlarında, 'Clk' sinyali ile Q çıkışında oluşacak durum değişimleri Şekil 9.11.b'de görülmektedir.

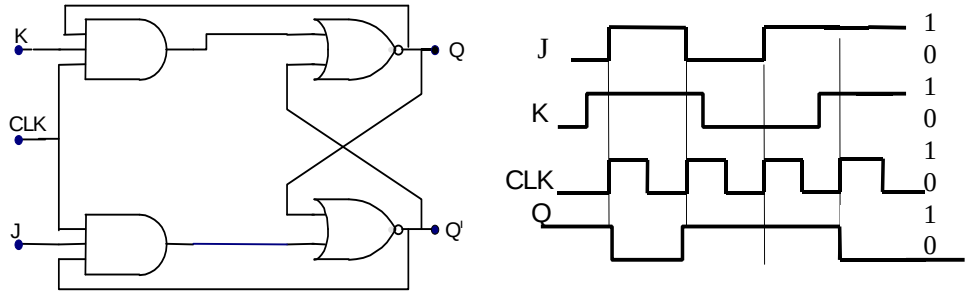


(a)

J	K	CLK	Q
0	0	1	Q_0
1	0	1	1
0	1	1	0
1	1	1	Toggle

(b)

Şekil 9.10. J-K FF sembolü ve doğruluk tablosu.



(a) **(b)**

Şekil 9.11. JK FF lojik devre şeması ve örnek durum değişimleri.

JK FF'nin çalışması Q çıkışının önceki durumu göz önünde tutularak doğruluk tablosunda gösterilirse, Şekil 9.12.a'daki karakteristik tablo ortaya çıkar. Bu tablonun Karnaugh haritasına taşınması ile, **Q(t+1)** durumu fonksiyon olarak tanımlanabilir (Şekil 9.12.b). JK FF'de yapılan işlem;

i- J=0, K=1 ve J=1,K=0 durumlarında çıkış J değerini izler.

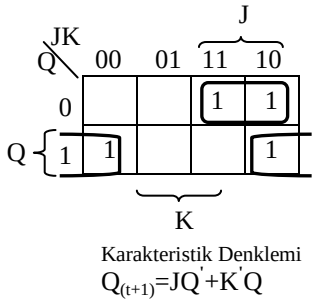
ii- J=K=0 durumunda önceki Q çıkışını koruyan devre, J=K=1 durumunda önceki çıkışın tersine sahip olur şeklinde özetlenebilir.

Bu açıklamalardan; 'RS FF'de bulunan yasak durum JK FF'de ortadan kalkmıştır' genellemesi yapılabilir.

Negatif tetiklemeli JK FF'nin çalışması ve çıkış durumu, FF'nin durum değiştirme anının tetikleme sinyalinin negatife giden kenarında olması haricinde pozitif tetiklemeli devrenin aynısıdır.

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

(a)

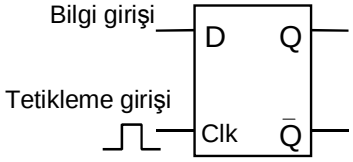


(b)

Şekil 9.12. J-K FF karakteristik tablosu ve karakteristik denklemi.

9.1.4 D Tipi Flip-Flop (D FF)

Temel RS FF'den üretilen diğer bir FF çeşidi, tek bir senkron kontrol girişine sahip olan D (Data) tipi FF devresidir (Şekil 9.13.a). D tipi FF'de oluşan işlemler basittir; Q çıkışı tetikleme sinyalinin gelmesi ile D kontrol girişinin sahip olduğu değeri alır (Şekil 9.13.b). Bunun anlamı; D=0 iken, tetikleme sinyalinin gelmesi ile Q çıkışı '0' değerine sahip olur demektir. D=0 durumu devam ettiği sürece, 'Clk' sinyalinin durumu değişse bile Q=0 değerini korur. D=1 değerini alması durumunda, ilk gelen tetikleme sinyalinin pozitif kenarında Q=1 değerini alır. Burada, Q çıkışının yalnızca tetikleme sinyalinin pozitif kenarlarında durum değiştirdiği unutulmamalıdır (Şekil 9.14.b).



D	Clk	Q
0	1	0
1	1	1

(a) **(b)**

Şekil 9.13. D tipi FF'nin sembolü ve doğruluk tablosu.

D tipi FF devresi, RS FF'nin girişine 'DEĞİL' kapısı bağlanarak elde edilebilir (Şekil 9.14.a). Eklenen 'DEĞİL' kapısı, hem RS FF'lerde belirsiz durum oluşturan R=S=1 durumunu ortadan kaldırır, hem de çıkışın D girişini takip etmesini sağlar.



(a) **(b)**

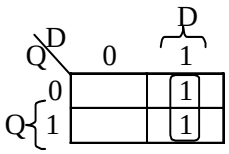
Şekil 9.14. D tipi FF'nin RS ile elde edilmesi ve örnek durum değişimleri.

D tipi FF'nin Q çıkışı dikkate alınarak doğruluk tablosu oluşturulur (Şekil 9.15.a) ve oluşan doğruluk tablosu Karnaugh haritalarına taşınırsa, Şekil 9.15.b'deki karakteristik denklem elde edilir. Elde edilen denklemden görüleceği üzere, FF'nin alacağı yeni durum D'ye bağımlıdır.

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

D FF karakteristik tablosu

(a)



Karakteristik Denklemi

$Q_{(t+1)}=D$

(b)

Şekil 9.15. D tipi FF karakteristik tablosu ve karakteristik denklemi.

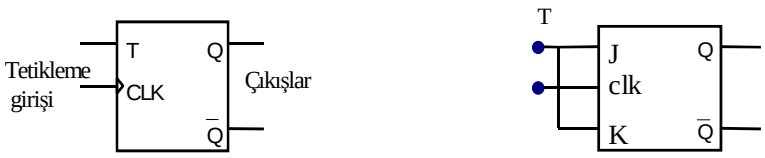
D tipi FF özelliğinden dolayı, paralel çıkışa sahip devrelerde çıkışlar arasında senkronizeyi sağlamak amacıyla kullanılır. Birleşik devrenin çıkışları, tampon olarak kullanılan D tipi FF'nin girişlerine uygulanarak tetikleme sinyali ile senkronizasyon sağlanır.

9.1.5. T Tipi Flip Flop (T FF)

J-K FF'nin girişlerinin birleştirilip tek giriş olarak kullanılmasıyla oluşan devre, '**Toggle FF' (T tipi FF)** olarak isimlendirilir (Şekil 9.16). 'Toggle', durumdan duruma geçme demektir. T tipi FF'de; T=1 iken, tetikleme sinyalinin uygulanmasıyla sahip olunan çıkışın terslenmiş hali elde edilir.

T tipi FF'de Q=0 ve T=0 iken, 'Clk' sinyalinin gelmesi durumunda çıkışta Q=0 değeri korunur. Q=0 ve T=1 değerlerinde, ilk gelen 'Clk' sinyali ile çıkış durum değiştirerek bir önceki durumun tersi olur ve Q=1 değerini alır.

Q=1 ve T=0 iken, 'Clk' sinyali uygulansa bile devre sahip olduğu çıkışı korur ve Q=1 değerini alır. Q=1, T=1 iken 'Clk' sinyali ile çıkış terslenir ve Q=0 olur.



Şekil 9.16. T Tipi FF'nin sembolü ve JK FF'den T FF elde edilişi.

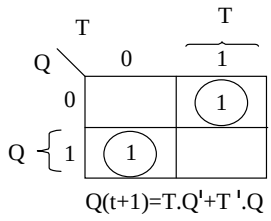
Anlatılan işlemlerin doğruluk tablosuna taşınması ile Şekil 9.17.a'daki karakteristik tablo oluşur. Tablodaki değerlerin Karnaugh haritasına taşınması ile karateristik denklem elde edilir (Şekil 9.17.b).

T tipi FF'de oluşan işlemler; 'T=0 iken FF önceki konumunu korurken, T=1 durumunda çıkış mevcut durumunun tersi olur' şeklinde özetlenebilir.

Q	T	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Karakteristik Tablo

(a)



Karakteristik Denklem

(b)

Şekil 9.17. T FF karakteristik tablosu ve karakteristik denklemi.

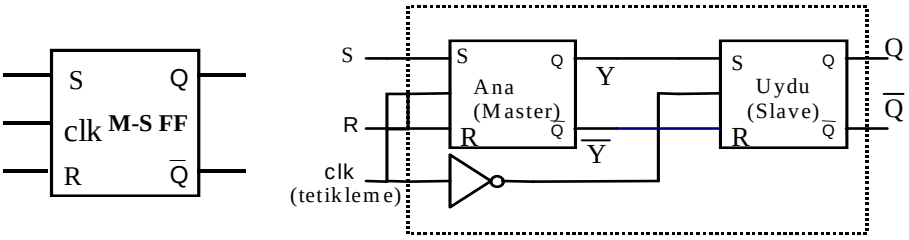
9.1.6. Ana Uydu (Master-Slave) Flip-Flop

Ana-Uydu (Master-Slave) tipi FF devresi, Şekil 9.18'de görüldüğü gibi iki RS Flip-Flop ve bir 'DEĞİL' kapısı ile oluşturulur. Devrede kullanılan FF'lerden birisi ana devreyi, diğeri ise ana devreye bağlı olarak çalışan uydu FF'yi oluşturur.

Ana-Uydu FF devresi, 'Clk' girişi '1' olduğunda 'Ana FF' devresi, '0' olduğunda 'Uydu FF' devresi çalışacak şekilde düzenlenmiştir.

Tetikleme girişine '0' uygulandığı zaman, 'DEĞİL' kapısının çıkışı uydu FF'nin 'Clk' girişini '1' yapar. Bu durumda Q çıkışı Y'ye, Q' çıkışı da Y çıkışına eşittir. Bu anda ana

FF'nin tetikleme girişi '0' olduğundan bu FF çalışmaz ve bir önceki konumu korur. Bunun anlamı; uydu FF'nin Q ve Q' çıkış değerlerinin, ana FF'nin bir önceki değerlerine sahip olmasıdır.



Şekil 9.18. Ana-Uydu FF sembolü ve lojik şeması.

Tetikleme girişinin '1' yapılması durumunda, ana FF etkin duruma gelirken, uydu FF etkisizdir. Ana FF'nin çıkışları, girişlerin durumuna göre değerler alır. Devrenin çıkışı, uydu FF'nin durumuna eşittir.

Ana-Uydu Flip-Floplar genelde negatif kenar tetiklemeli olarak çalışırlar. RS Tipi FF'lerle oluşturulan Ana-Uydu FF'nin diğer FF çeşitleri ile oluşturulması mümkündür.

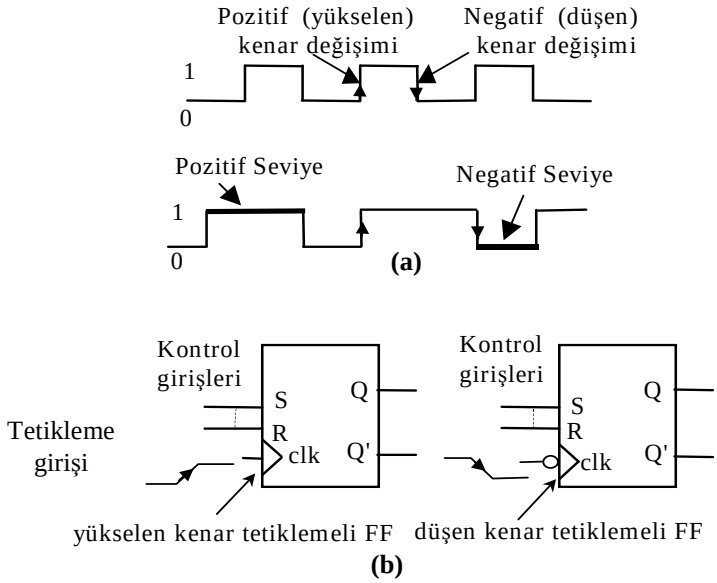
9.2. Tetikleme Sinyali ve FF'lerde Tetikleme

FF'lerin tetikleme girişine uygulanan kare veya dikdörtgen şeklindeki sinyaller, '**tetikleme sinyali / palsı**' olarak adlandırılır. Diğer bir değişle; bir ardışıl devrenin durumunu değiştirip yeni bir duruma geçmesi için (bir sonraki durum) dışarıdan uygulanan uyarma işaretine **'tetikleme sinyali'** denir. FF'lerdeki tetikleme işlemi, tetikleme sinyalinin durum değişimi veya 0/1 seviyeleri sırasında gerçekleşir (Şekil 9.19.a).

FF devresini etkileyen sinyalin değişim yönü, FF sembolü üzerinde belirtilir. FF sembolündeki '>' işareti, tetikleme sinyalinin 0'dan 1'e veya 1'den 0'a geçişi sırasında FF girişlerinin etkili olacağını gösterir. '>' işaretinin önündeki daire (o) bulunması, FF'nin düşen kenar ile tetikleneceğini gösterir (Şekil 9.19.b).

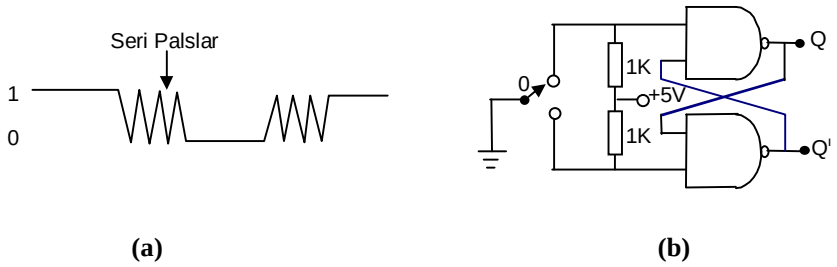
FF devresi, tetikleme sinyalinin pozitif kenarında tetikleniyorsa **'pozitif kenar tetiklemeli',** sinyalin negatif kenarında tetikleniyorsa **'negatif kenar tetiklemeli'** devre olarak tanımlanır. Tetikleme işlemi 1 veya 0 değeri sırasında gerçekleşiyorsa, düşey tetiklemeli devre olarak adlandırılır.

FF devrelerine tetikleme sinyali anahtar yardımıyla uygulanırsa, anahtarın kapatılması ve açılması sırasında, anahtarın mekaniksel yapısı nedeniyle sıçramalar oluşur (Şekil 9.20.a).



Şekil 9.19. Tetikleme sinyalleri ve tetikleme sinyallerinin FF'lere uygulanması.

Tetikleme sinyalinde meydana gelen sıçramalar, senkronizasyonlu devrelerde karışıklıklara neden olabilir. Çünkü kısa süreli seri palslar, sistemin çalışmasına etki eden parazitler oluşturulabilir. Parazitlerin oluşması olayı, **'Sıçrama Olayı'** (Bouncing) olarak adlandırılır. Parazitlerin oluşmasını engellemek için yapılan işlem ise, **'anahtarlı sıçrama önleme'** (Switch Debouncing) diye tanımlanır (Şekil 9.20.b).



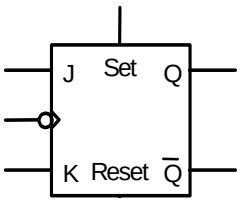
Şekil 9.20. Anahtar kontaklarında açılması ve kapanması sırasında oluşan sıçrama etkisi ve etkiyi kaldırmak için kullanılan anahtarlı sıçrama önleme devresi.

9.3. FF'lerde Asenkron Girişler

Buraya kadar incelediğimiz RS, JK, D, T, Ana-Uydu tipi Flip-Flop'ların sahip oldukları girişler, **'kontrol girişleri'** veya **'senkronize girişler'** olarak isimlendirilir. Bu girişler, tetikleme girişi (Clk) ile ilişkili olarak çalışırlar. Bununla beraber tetiklemeli FF'lerin çoğu, senkronize girişler ve tetikleme girişi ile bağımsız olarak çalışan bir veya daha fazla asenkron girişe sahiptir.

'Asenkron girişler', FF'nin çıkış durumunu '1' konumuna getiren **'set işlemi'** veya çıkışın durumunu '0' konumuna getiren **'reset işlemi'** için kullanılır. Set ve reset işlemleri yapan asenkron girişler, diğer girişlere göre daha üstündürler ve diğer girişlerin durumuna bakmaksızın FF'nin çıkış durumunu belirlerler.

Şekil 9.21'de, set ve reset asenkron girişlerine sahip tetiklemeli J-K FF'nin sembolü ve doğruluk tablosu görülmektedir. Negatif kenar tetiklemeli bu devrede, her iki asenkron girişin '0' olması durumunda tetiklemeli J-K FF işlemi gerçekleştirilir. Set=0, Reset=1 olması durumunda, diğer girişlerin durumuna bakılmaksızın FF çıkışı Q= 0 konumuna getirilir. Reset=0, Set=1 durumunda ise, başka bir etkene bakılmaksızın FF'nin çıkışı Q=1 değerini alır. Set=1, Reset=1 durumu ise belirsiz bir çıkışa neden olacağından kullanılmaz.

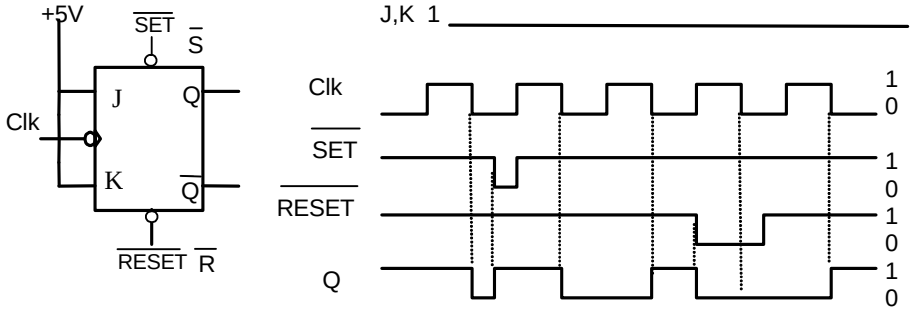


Set	Reset	FF'nin tepkisi
1	1	Normal işlem
0	1	Q=1
1	0	Q=0
0	0	Kullanılmaz

Şekil 9.21. Asenkron girişli J-K FF sembolü ve doğruluk tablosu.

Asenkron girişler, herhangi bir anda FF'nin durumunu belirlemek veya bir devrede birlikte kullanılan FF'lerin hepsini birlikte set / reset konumuna getirmek için kullanılabilirler. Pratikte kullanılan FF entegreleri, set / reset asenkron girişlerine veya yalnızca reset girişine sahiptir. Bu elamanlardan bir kısmı negatif kenar tetiklemeli iken, bir kısmı pozitif kenar tetiklemelidir. Elemanlar negatif kenar tetiklemeli ise, bu özellik $\overline{\text{set}}$ ve $\overline{\text{reset}}$ sembolleri ile ifade edilir.

Anlatılan özelliklere sahip bir FF'nin sembolü ve girişlerine uygulanan değişik dalga şekillerine elemanın verdiği tepki Şekil 9.22'de görülmektedir.



Şekil 9.22. Asenkron girişlere sahip J-K FF sembolü ve değişik durumlarda oluşan çıkış dalga şekilleri.

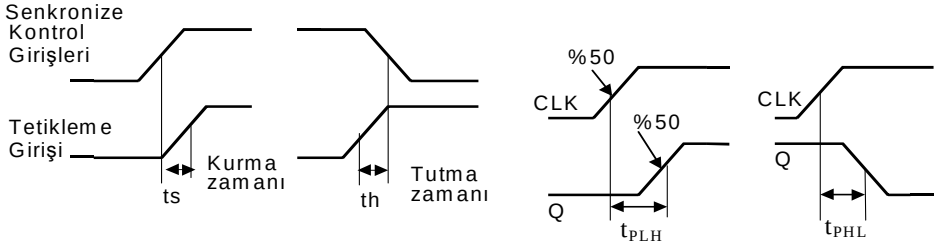
9.4. Flip-Flop Parametreleri

Flip-Flop entegre devre üreticileri, FF'lerin sahip oldukları özellikleri açıklayan zamanlama parametreleri ve karakteristikler belirlemişlerdir. Bunlardan önemli olanları, TTL ve CMOS ailelerindeki elamanlar üzerinde inceleyelim.

i- Kurma ve Tutma Zamanları (Setup and Holding Times):

Kurma zamanı (setup time-ts), tetikleme sinyalinin devrenin durumunu değiştirmesi için geçen zamandır. Entegre devre üreticileri genelde izin verilebilen minimum kurma zamanını belirtirler. Eğer bu zaman gereksinimi sağlanmazsa, devre tetikleme sinyallerinin durum değişikliklerine güvenli tepki vermeyebilir.

'Tutma zamanı' (holding time-th), tetikleme sinyalinin senkronize girişlerdeki durum değişikliğini hissedebilmesi için gerekli zaman aralığıdır. Üreticiler tarafından minimum sınırı belirlenen bu zamanın sağlanamaması durumunda FF güvenli olarak tetiklenemez.



Şekil 9.23. FF kontrol girişleri için kurma - tutma zamanları ve FF yayılım gecikmeleri.

Tetiklemeli FF'nin tetikleme sinyaline düzgün tepki verebilmesi için, senkronize girişlerin belirli bir süre değişmeden durması gerekir (kurma ve tutma zamanlarından dolayı). Bu süre, tetikleme sinyalinin durum değiştirmesinden önce 'ts' kadar, durum değişikliğinden sonra 'th' kadar olmalıdır (Şekil 9.23.a). FF'lerde genelde kurma zamanı 5-50 nsn, tutma zamanı 1-10 nsn arasında değişir. Senkron sistemlerde çok önemli olan bu zamanların ölçümünde, durum değişikliklerinin %50 seviyesi referans olarak alınır.

ii- Yayılım Gecikmesi (Propagation Delay):

FF'lerda tetikleme sinyalinin uygulandığı an ile, çıkışın konum değiştirdiği an arasındaki zaman farkına **'yayılım gecikmesi'** denir. FF'nin durumunun 0'dan 1'e geçerken ve 1'den 0'a geçerken oluşan yayılım gecikmeleri, üretici kataloglarında t_{PHL} ve t_{PLH} şeklinde maksimum değerler olarak verilir (Şekil 9.23.b).

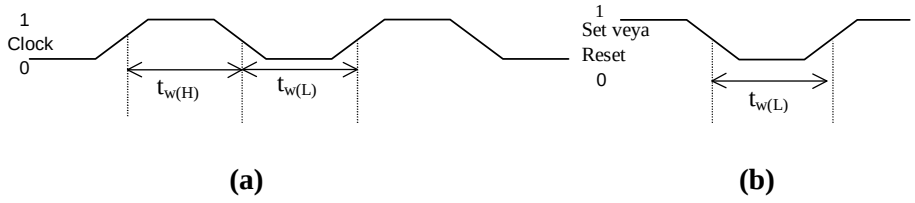
Yeni tasarlanan FF entegrelerinde yağılım gecikmesi birkaç nane saniye ile 1μsn arasında değişir. Genelde aynı değere sahip olmayan t_{PHL} ve t_{PLH} zamanları, Q çıkışından sürülen yüklerin artması ile orantılı olarak yükselir.

iii- Maksimum Tetikleme Frekansı (Maximum Clocking Frecuency – fmax):

'Tetikleme frekansı', FF'nin tetikleme girişine güvenli bir tetikleme için uygulanabilecek en yüksek frekansı ifade eder. F_{max}'ın temsil ettiği değer, aynı seri numaralı FF'lerde dahi farklılık gösterebilir. Verilen sınırın altındaki frekanslarda devre randımanlı çalışırken, bunun üzerindeki değerlerde olumsuz durumlarla karşılaşılabilir.

iv- Tetikleme sinyali '1' - '0' zamanları ve Asenkron Giriş Sinyal Genişliği:

Üretici firmalar, tetikleme sinyalinin '1' ve '0' durumlarında kalması gerekli minimum zamanı (Clock pulse High and Low Times) belirlerler. '0' durumunda kalması gerekli minimum zaman '$t_{w(L)}$' olarak, '1' durumunda kalması gerekli minimum zaman '$t_{w(H)}$' olarak isimlendirilir. Durum değiştirmelerin %50 seviyeleri arasındaki zaman olarak tanımlanan bu sürelerin sağlanamaması durumunda güvenli bir tetikleme işlemi oluşmayabilir (Şekil 9.24.a).



Şekil 9.24. Tetikleme sinyali durumları ve asenkron giriş sinyal genişliği.

Üreticilerin belirlediği diğer bir parametre, 'set' ve 'reset' girişlerinin (asenkron girişlerin) aktif tutulma zamanlarıdır. Minimum değer olarak verilen bu zamanların sağlanamaması durumunda 'set' ve 'reset' işlemleri gerçekleşmeyebilir (Şekil 9.24.b).

v- Tetikleme sinyali durum değiştirme zamanı :

Durum değiştirme zamanı (Clock Transition Times), tetikleme sinyalinin 1'den 0'a veya 0'dan 1 durumuna değişirken geçen zamandır. Güvenli bir tetikleme işlemi için, tetikleme sinyali durum değişim zamanlarının çok küçük tutulması gereklidir. Durum değiştirme zamanının uzun olması durumunda tetikleme işlemi yanlış yorumlanabilir veya tetikleme işlemi oluşmayabilir.

Üreticiler her bir entegre için gerekli maksimum durum değiştirme zamanını vermeseler de, lojik aileler için ortalama değerler vermişlerdir. Örneğin; TTL ailesi için durum değiştirme zamanı ≤50 nsn ve CMOS ailesi için ≤200 nsn olmalıdır.

9.5. Flip-Flop Durum Geçiş Tabloları

'Flip-Flop doğruluk tablosu', FF'nin özelliğini ve çalışmasını tanımlar. Flip-flop'ların özelliklerini tanımlayan RS, JK D ve T FF'lere ait doğruluk tabloları Şekil 9.25'de verilmiştir. Flip-Flopların çalışma durumunu analiz etmede oldukça kullanışlı olan doğruluk tabloları yardımıyla, giriş ve mevcut durum değerlerinden faydalanılarak sonraki durum kolayca bulunabilir. Tablodaki **Q(t)** mevcut durumu, **Q(t+1)** ise tetikleme sinyali uygulanınca meydana gelen yeni durumu açıklar. Şimdi Flip-Flop'lara ait doğruluk tabloları inceleyelim:

Şekil 9.25.a'da görülen RS Flip-Flop'un doğruluk tablosundan; R=S=0 durumunda önceki ile sonraki lojik seviyelerin aynı olduğunu görürüz. S=1 yapıldığında FF çıkışı Q=1 pozisyonunu alırken, R=1 olduğunda FF'nin çıkışı Q=0 durumunu alır. S=R=1 durumunda ise çıkış durumu belirsizdir.

J-K FF'e ait doğruluk tablosu, tanımsız durum haricinde R-S FF'ninki ile aynıdır. Kontrol girişlerinin J=K=1 olması durumunda, yeni durum mevcut durumun lojik olarak tersidir.

D tipi FF'lerde Q(t+1) durumu tamamen D girişine bağlıdır. Mevcut durumun çıkışa bir etkisi yoktur.

T tipi FF'lerin yeni durumu T=0 olduğunda mevcut durum ile aynı iken, T=1 olduğunda mevcut durumun tersidir.

Lojik devre tasarımı sırasında, FF durumlarında gerekli değişmeyi sağlayacak giriş değişkenlerini belirlemek önemlidir. Giriş değişken değerlerinin belirlenmesinde, FF'lerda durum değişikliğini sağlayacak giriş seviyelerini gösteren (her bir tip FF için farklı olan)

tablolardan faydalanılır. Doğruluk tablolarından faydalanılarak oluşturulan bu tablolara, **'durum geçiş tabloları'** adı verilir.

S	R	$Q_{(t+1)}$
0	0	$Q_{(t)}$
0	1	0
1	0	1
1	1	B ?

a) R-S FF

J	K	$Q_{(t+1)}$
0	0	$Q_{(t)}$
0	1	0
1	0	1
$\overline{1}$	1	$Q_{(t)}$

b) J-K FF

D	$Q_{(t+1)}$
0	0
1	1

c) D FF

T	$Q_{(t+1)}$
0	$Q_{(t)}$
1	$\overline{Q}_{(t)}$

d) T FF

Şekil 9.25. Flip Flop'ların doğruluk tabloları.

Şekil 9.26'da 4 çeşit FF'e ait durum geçiş tabloları görülmektedir. Tablolarda Q(t) ve Q(t+1) olarak gösterilen kolonlar, sırasıyla mevcut durumu ve istenilen durum (sonraki durum) değişikliğini gösterir. Mevcut durumdan sonraki duruma geçiş sırasında, dört farklı geçişten biri oluşabilir. Dört farklı geçiş işlemi için gerekli giriş değerlerini gösteren bilgiler durum geçiş tablosundan alınabilir.

Durum geçiş sırasında, giriş değerlerinin önemli olmadığı durumlar **'fark etmezlik'** olarak ifade edilir ve 'X' veya 'd' ile gösterilir. 'X' ile ifade edilen bilginin '1' veya '0' olmasında bir farklılık yoktur.

Şekil 9.25'de verilen doğruluk tablolarından faydalanarak, RS, JK, D ve T tipi FF'lere ait durum geçiş tablolarının oluşturulmasını açıklayalım:

RS FF'lerde mevcut ve yeni durumun aynı olması isteniyorsa, RS FF doğruluk tablosundan R=S=0 giriş değerlerine sahip olması gerektiği bulunur (Şekil 9.26.a). Bununla beraber, S=0 olması çıkışın '0' olması için yeterli olduğundan, 0'dan 0'a durum değişikliğinde R'nin alacağı değerin önemi yoktur. Bu nedenle, geçiş tablosuna S=0, R=X değerleri yazılabilir. Durum geçiş tablosundan, FF'in '0' dan '1' durumuna geçmesi isteniyorsa; S=1 ve R=0 değerlerinin, aynı şekilde 1'den 0' a geçiş isteniyorsa; S=0 ve R=1 değerlerinin uygulanması gerektiği bulunabilir. '1' konumundaki çıkışın durumunu '1' olarak koruması için R=0 değeri uygulanırken, S'ye uygulanacak bilginin önemi yoktur.

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

a) RS FF

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

b) JK FF

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

c) D FF

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

d) T FF

Şekil 9.26. Flip- Flop'ların durum geçiş tabloları.

JK FF'lerde mevcut ve yeni durumun '0' olması durumunda; J=0, K=X uygulanmalıdır (Şekil 9.26.b). Aynı şekilde, çıkış '1' durumunda iken konumunun devam etmesi isteniyorsa girişlerde K=0, J=X değerleri bulunmalıdır. FF çıkışının 0'dan 1'e değişmesi isteniyorsa; J=1 değerinin uygulanması yeterlidir ve K'ya uygulanacak değerin önemi yoktur. Çıkış konumunun 1'den 0'a geçmesi isteniyorsa; K=1 uygulanması çıkışın '0' değerini alması için yeterli olduğundan, J girişinin değeri önemsizdir.

D tipi FF geçiş tablosundan (Şekil 9.26.c), FF'nin sonraki durumunun her zaman D girişine eşit olduğu söyleyebiliriz. Yani mevcut durumun bir etkisi yoktur. $Q_{(t+1)}$ değerinin ne olması isteniyorsa, D'ye aynı bilginin uygulanması yeterlidir.

T tipi FF'ye ait durum geçiş tablosundan (Şekil 9.26.d); T=1 durumunda FF'nin durum değiştirdiğini, T=0 olduğunda ise FF'nin çıkışının bir önceki durumunu koruduğunu bulabiliriz. Bu nedenle; T tipi FF'nin çıkışının bir önceki durumunu koruması isteniyorsa T=0, durumunu değiştirmesi isteniyorsa T=1 yapılmalıdır.

Durum geçiş tabloları sayıcı tasarımında önemli bir yere sahip olduğundan, sayıcılar konusunda sıkça kullanılacaktır.

9.6. Flip-Flop Giriş Fonksiyonları

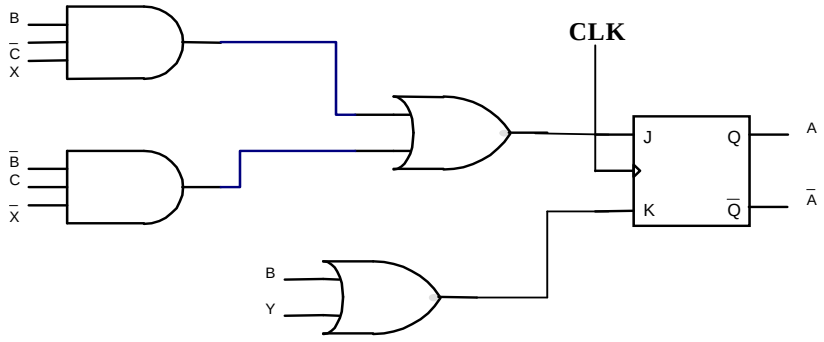
Bir FF'nin giriş fonksiyonu dendiğinde, FF devresinin giriş katında bulunan birleşik devrelerce üretilip FF girişlerine uygulanan ve FF'nin istenen yeni durumlara ulaşmasını sağlayan lojik kapı devreleri anlaşılır. FF'lerin giriş fonksiyonlarını ifade eden denklemler, hem harici girişlerin hem de şimdiki (mevcut) durumların fonksiyonu olarak yazılır. FF devresinin giriş değişkenleri iki harfli bir simge ile gösterilir ve birinci harf girişin adını, ikinci harf ise FF'nin adını simgeler. Örneğin; bir devrede bulunan J-K FF'lere ait girişler J_A, J_B, J_C ve K_A, K_B, K_C simgeleri ile belirtilir. Aynı durum diğer FF çeşitleri için de geçerlidir.

Örnek olarak aşağıdaki FF giriş fonksiyonlarını ele alalım:

$$J_A=BC'x+B'Cx' \qquad \text{ve} \qquad K_A=B+y$$

denklemlerinde, J_A ve K_A iki Boolean eşitliğini gösterirler.

Bu eşitliklerdeki J ve K harfleri FF'nin girişlerini, 'A' ise FF'nin adını gösterir. Denklemlerdeki ifadeler Boolean fonksiyonları olduğundan, birleşik devre olarak kapı devreleri ile gerçekleştirilebilir. Gerçekleştirilen devre 'A' çıkışına sahip bir FF'nin J-K girişlerine uygulanırsa, Şekil 9.27'deki devre elde edilir. Fonksiyonlardan oluşturulan birleşik devreler, FF'nin ilgili girişlerine uygulanır.



Şekil 9.27. J_A ve K_A giriş fonksiyonlarının uygulanması

FF giriş fonksiyonları, sıralı (ardışıl) bir devrenin lojik devresinin tanımlanabilmesi için uygun bir cebirsel eşitlik sağlar. Bu eşitlik, FF'yi çalıştıran devreyi tam olarak tanımlar. Bazı durumlarda, ardışıl bir devreyi mantık devresi çizerek belirtmek yerine, FF'lerin giriş fonksiyonları ile belirtmek daha kolaydır. Bu durum ardışıl devreler konusunda kullanılacaktır.

Tekrarlama ve Çalışma Soruları

1. Multivibratörü tanımlayınız.

2. Multivibratör çeşitlerini yazarak, herbir çeşit multivibratörü tanımlayınız.

3. Çift kararlı multivibratör açık şemasını çizerek, devrenin çalışmasını özetleyiniz.

4. Flip-Flop'u tanımlayınız.

5. FF çeşitlerini yazınız.

6. 'Latch' ne demektir?

7. RS FF ile tetiklemeli RS FF arasındaki farkı tanımlayınız.

8. 'VEDEĞİL' ve 'VE / VEYADEĞİL' kapıları ile oluşturulan tetiklemeli RS FF şekillerini çiziniz.

9. Tetiklemeli RS FF'nin karakteristik denklemini doğruluk tablosu ve Karnaugh yardımıyla yazınız.

10. JK FF'nin lojik şemasını çizerek, devrenin çalışmasını anlatınız.

11. JK FF'nin karakteristik denklemini yazınız

12. D ve T tipi FF'lere ait doğruluk tablolarını çizerek, genel özelliklerini yorumlayınız.

13. D ve T tipi FF'lere ait karakteristik denklemleri oluşturunuz.

14. Ana-Uydu FF'nin çalışmasını şekille özetleyiniz.

15. Ana uydu FF karakteristik denklemini yazınız.

16. Tetikleme sinyalini tanımlayınız.

17. Negatif ve pozitif kenar tetiklemesi terimlerini açıklayınız.

18. 'Bouncing' olayını açıklayınız.

19. Bouncing olayını önleyecek devreyi çizerek, işlevini özetleyiniz.

20. FF'lerde kullanılan asenkron girişlerin isimlerini ve işlevlerini yazınız.

21. Asenkron girişlere sahip JK FF'de, asenkron girişlerin etkisini özetleyiniz.

22. FF'lerde üreticilerin belirlediği parametreleri sıralayınız.

23. FF'lerde yayılım gecikmesi ve maksimum tetikleme frekansı terimlerini açıklayınız.

24. FF'lerde asenkron giriş sinyal genişliği parametresi ne anlama gelmektedir?

25. FF karakteristik tablolarını özetleyiniz.

26. FF geçiş tablolarını çizerek, herbir FF'ye ait geçiş tablosunu yorumlayınız.

27. FF karakteristik tablosu ile geçiş tabloları arasındaki fark nedir?

28. FF geçiş tablolarının en yaygın kullanım yeri neresidir?

29. FF'lerde 'farketmezlik' durumu hangi şartlarda kullanılır?

30. JK FF'ye ait karakteristik tablodan faydalanarak, geçiş tablosu oluşturulması işlemini özetleyiniz.

31. T tipi FF'ye ait geçiş tablosunu oluşturunuz.

32. 'FF giriş fonksiyonu' terimini açıklayınız.

33. FF giriş fonksiyonu denklemleri nasıl oluşturulur?

34. FF girişlerine ait giriş fonksiyonu nasıl gösterilir?

35. J_A = XA'B + X'AB + XAB ve K_A = X'A'B + X'AB' + XA'B' giriş fonksiyonlarına sahip ardışıl devrenin şemasını çiziniz.

BÖLÜM 10

SENKRON SIRALI / ARDIŞIL MANTIK DEVRELERİ

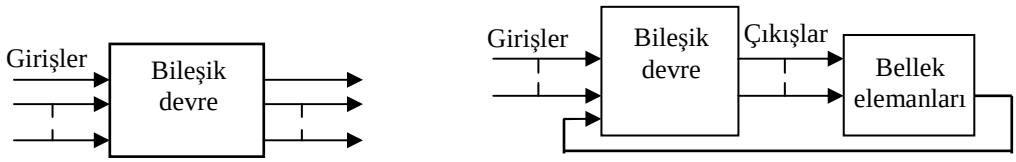
(SYNCHRONOUS SEQUENTIAL LOGIC)

Giriş

Lojik kapı elemanlarından oluşan ve Bölüm 8'de detaylandırılan devreler, **'bileşik devreler'** olarak isimlendirilir. Geri besleme ve zamanlama gibi olayların olmadığı bu devrelerde, belli bir anda girişe gelen bilgiler belli bir amaç doğrultusunda işlemden geçtikten sonra çıkışa aktarılır (Şekil 10.1.a). Devrede bellek olmaması nedeniyle, daha önceki giriş / devre şartlarından etkilenme diye bir şey söz konusu değildir.

Bileşik devrelere geri besleme ve zamanlama gibi olguların eklenmesiyle oluşan devrelere, **'sıralı veya ardışıl lojik devreler'** denir (Şekil 10.1.a). Ardışıl devrelerde sıkça kullanılan bellek elemanları, bileşik devrelere bir geri besleme (feedback) yolu sağlayacak şekilde bağlanır (Şekil 10.1.b).

Bilgi saklama yeteneğine sahip elemanlar olan belleklerde saklanan ikili bilgiler, sıralı devrenin durumunu tanımlar. Harici girişlerden ikili bilgiler alan ardışıl devre, hariçten aldığı ikili bilgi ile, bellek elemanlarının mevcut durumlarını birleştirerek çıkışta oluşacak ikili değeri belirler.

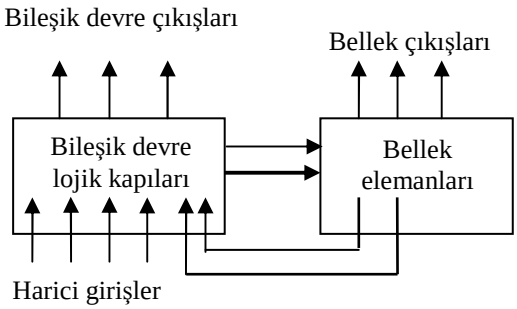


a) Bileşik devre blok şeması. **b)** Ardışıl devre blok şeması.

Şekil 10.1. Bileşik ve Ardışıl devre blok şemaları.

Bileşik devrenin gerçekleştirdiği işlem sonucunda, bellek elemanlarının durumunda değişiklik yapma koşulu da belirlenir. Şekil 10.2'deki blok şemadan görüldüğü gibi, harici girişler ve bellek elemanındaki bilgilerin değerlendirilmesi sonucu değişik çıkışlar elde edilir. Elde edilen çıkışlardan bir kısmı bellek elemanlarında saklanacak yeni ikili değeri temsil ederler. Bu işlemlerden çıkarılan sonuç; ardışıl devrenin harici çıkışlarının harici girişler ile bellek elemanlarında saklanan bilginin bir fonksiyonu olduğudur.

Ardışıl (sıralı) devreler, sinyallerin zamanlamasına bağlı olarak iki grupta toplanabilir: Senkron (eş zamanlı) ve asenkron (eş zamanlı olmayan) ardışıl devreler.



Şekil 10.2. Ardışıl devrelerde önceki durumların değerlendirilmesi.

Senkron ardışıl devrelerde, bellek elemanlarını sadece belirli zaman dilimlerinde etkileyen sinyaller kullanılır. Sinyaller oluşturulurken sınırlı süreli darbeler kullanılır. Bu şekilde darbelerin kullanıldığı bir sistemde zorluk, farklı kaynaklardan aynı kapının girişlerine ulaşan iki bilginin (sinyalin) önceden tahmin edilmeyen gecikmeler göstermesi ve bilgilerin birbirinden bir miktar da olsa ayrılmasıdır.

Senkron sıralı devrelerde, senkron çalışma periyodik tetikleme sinyalleri (clock pulses) dizisi üreten ana sinyal üreteci (master - clock generator) ile sağlanır.

Bellek elemanlarının girişinde tetikleme sinyalleri kullanan senkron sıralı devreler, **'tetiklemeli (saatli) sıralı devreler'** (clocked sequential logic) olarak adlandırılır.

Kararsızlık probleminin oluşmadığı bu devrelerde, bellek elemanı olarak Flip-Flop devreleri kullanılır.

Asenkron (eş zamanlı olmayan) sıralı devrelerde, çıkışın değişmesi giriş sinyallerinin bir veya daha fazlasının değişmesine bağımlıdır. Giriş ve buna bağlı olarak çıkış durumu herhangi bir anda etkilenebilir (değişebilir).

Asenkron sıralı devrelerde yaygın olarak kullanılan bellek elemanları zaman gecikmeli elemanlardır. Zaman gecikmeli bir elemanın bellek yeteneği, sinyalin eleman içinde yayılmasının belli bir zaman alması olgusundan kaynaklanmaktadır. Pratikte mantık kapılarında meydana gelen gecikme, ihtiyaç duyulan gecikmeyi sağlayarak fiziksel gecikme ünitelerine olan ihtiyacı ortadan kaldırır.

Kapılarla oluşturulan asenkron devreler geri beslemeli bileşik devre olarak değerlendirilebilir ve bu şekildeki devrelerde geri besleme nedeni ile devrenin kararlı çalışma durumu bozulabilir.

10.1 Ardışıl Devrelerin Analizi

Ardışıl devre, bileşiminde en az bir FF bulunduran, bunun ötesinde giriş-çıkış durumunu belirleyen bileşik devreler (lojik kapılar) içeren düzeneklerdir. Ardışıl devrelerin davranışı; girişlerine, çıkışlarına ve mevcut durum ile alabilecekleri gelecek durumlara göre belirlenir. Ardışıl bir devrenin analizi; girişlerin, çıkışların ve iş sırası tablosu veya şemasının elde edilmesi şeklinde gerçekleştirilir.

Ardışıl devrelerin davranışını tanımlayan boolean ifadeleri yazılabilir. Ancak, bu ifadelerin doğrudan veya dolaylı olarak gerekli zaman sırasını içermesi gerekir. Ardışıl devreyi analiz etmek için, tetiklemeli ardışıl devreye bir örnek vererek ardışıl devrelerin davranışlarını açıklayan çeşitli yöntemleri inceleyelim.

Şekil 10.3'te genel bir ardışıl devre örneği görülmektedir. Devre, bir 'x' giriş değişkeni ile A ve B olarak isimlendirilen iki adet tetiklemeli R-S FF'ye sahiptir. A FF'sinin S_A ve R_A girişleri, C_1 ve C_2 ile simgelenen bileşik devreler tarafından sağlanır. B FF'sinin S_B ve R_B girişleri ise, C_3 ve C_4 bileşik devreleri tarafından üretilir. Devrenin incelenmesini kolaylaştırmak amacıyla, FF'lerin çıkışlarından kapı girişlerine olan bağlantılar gösterilmemiş, bunun yerine bağlantılar girişlerine konan harf sembolleriyle belirtilmiştir. Devrede kullanılan FF'ler tetiklemeli olduklarından, girişlerin etkisi tetikleme palsleri sırasında algılanır ve FF'lerin geçiş özelliklerine göre A(t+1), A'(t +1), B(t+1), B'(t+1) gelecek durumları üretilir.

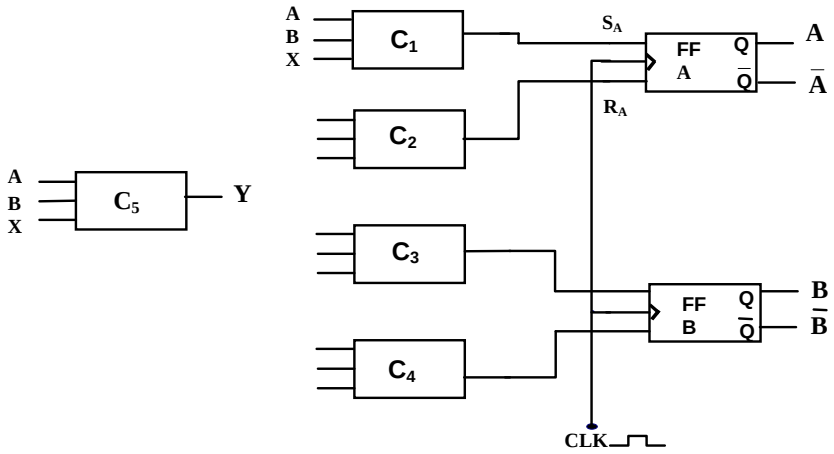
Bu şekilde genel özellikleri tanımlanan bir ardışıl devrenin analizi üç değişik yöntemle yapılabilir:

í- Durum Geçiş Şeması Yöntemi,

íí- Durum Geçiş Tablosu Yöntemi,

ííí- Durum Denklemleri Yöntemi,

Bu yöntemleri sırası ile açıklayalım.



Şekil 10.3. Ardışıl devre örneği.

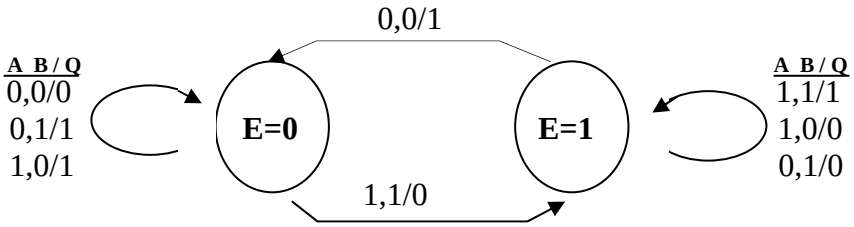
10.1.1. Durum Geçiş Şeması Yöntemi

Durum geçiş şeması yöntemi; ardışıl devrenin sözel anlatımını en iyi yansıtan, mevcut durum-gelecek durum ilişkisini görsel biçimde sergileyen bir yöntemdir. Bu yöntemde, devrenin her durumu bir daireyle, durumlar arasındaki geçişler ise daireleri birbirine bağlayan yönlü oklar ile gösterilir. Her dairenin içindeki ikili sayı, dairenin temsil ettiği durumu tanımlar. Geçişleri gösteren oklar üzerine ise söz konusu durum geçişinin hangi girdilerle gerçekleştiği ve çıktının ne olduğu yazılır. Bu gösterim için kesme işareti (/) kullanılır ve bu kesme işaretinin soluna girdiler, sağına ise çıktılar yazılır. Bir durum değişikliği söz konusu değil ise, durum geçiş oku başladığı noktaya döner (Şekil 10.4).

Örnek 1: Tam toplayıcı işlemini ardışıl devre ile gerçekleştirmede oluşan olayları inceleyelim.

Bir toplayıcı devresinde elde terimi ‘E’, giriş değişkenleri ise ‘A’ ve ‘B’ sembolleriyle gösterilebilir. Sözgelimi E=0 iken, giriş değerleri A=1, B=1 olduğunda E=1 durumuna geçer ve toplayıcının çıkışı ‘0’ olur. E=1 durumunda iken; A=0, B=0 dışındaki giriş değerlerinde devrenin durumunda bir değişiklik olmaz (Şekil 10.4).

Kesme ‘/’ işaretinin solundaki değerler ‘A’ ve ‘B’ girişlerini, sağındaki terim ise ‘Q’ çıkışının durumunu ifade eder. Daire içerisindeki ‘E’ değeri ile giriş değerlerinin toplamını, devrenin alacağı konumu (Q) ve E’nin yeni değerini belirler. Örneğin E=0 iken; girişlerin ‘0,1’ olması durumunda E=0 konumunu korurken, Q=1 değerini alır. Bu durumda geçiş oku başladığı daireye geri döner. E=1 iken; girişlere ‘0, 0’ uygulanması durumunda Q=1 değerini alırken, ‘E’ durum değiştirerek ‘0’ değerini alır. Geçiş oku E=o olan diğer daireye doğru değişimi gösterir.



a) Ardışıl toplayıcı blok şeması.



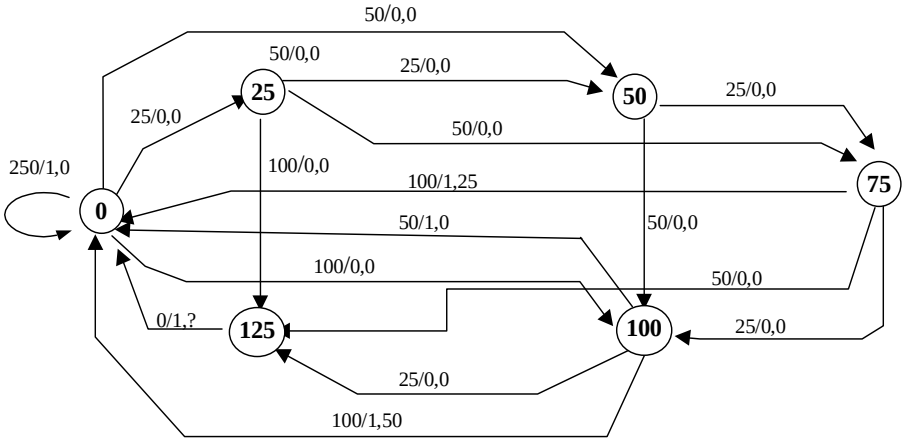
b) Ardışıl toplayıcı durum geçiş şeması.

Şekil 10.4. Ardışıl toplayıcı blok şeması ve durum geçiş şeması.

Örnek 2: Paralı satıcı kutu kola makinesi:

Değişik yerlerde karşılaştığımız paralı kola satıcı makinenin durum geçiş şemasını çizelim. Makinenin girdilerinin 25, 50, 100 ve 250 birimlik madeni paralar olduğunu varsayalım. Çıkışında ise kutu kola düşsün ve gereğinde paranın üzerini versin. Kolanın 150 birim değerinde olduğunu varsayarak, makinenin alacağı durumları; boş, kasada 25 var, kasada 50 var, kasada 75 var, kasada 100 var ve kasada 125 var diye sıralayabiliriz. Bu değerleri kasanın alacağı durumlar kabul ederek daireler içerisinde gösterebiliriz.

Makine ‘kasada 125 birim var’ durumuna geldikten sonra hangi para atılırsa atılsın bir kola kutusu düşmekte, paranın üstü geri verilip makine boş durumuna dönmektedir. Kasada olabilecek paraları durumlar olarak tespit ettikten sonra atılabilecek para birimlerini ‘/’ işaretinin solunda gösterip, ‘/’ işaretinin sağında ise kolanın düşüp (1), düşmediği (0) ve ayrıca kalan para miktarı gösterilirse, Şekil 10.5’deki durum şeması elde edilir.



Şekil 10.5. Kutu kola makinası için durum geçiş şeması.

Bu örnekte sözel anlatımda geçiş şemasını çıkardık. Lojik şeması çizilen ardışıl devrede geçiş şeması oluşturulabilir. Bunu da bir örnekle açıklayalım.

Örnek 3: Şekil 10.6'da verilen mantık devresi, ardışıl bir fonksiyonu gerçekleştirmektedir. Devrenin girişi 'x', çıkışı 'y', durumlar ise; 'A' ve 'B' çiftidir. Bu devrenin durum şemasını oluşturalım.

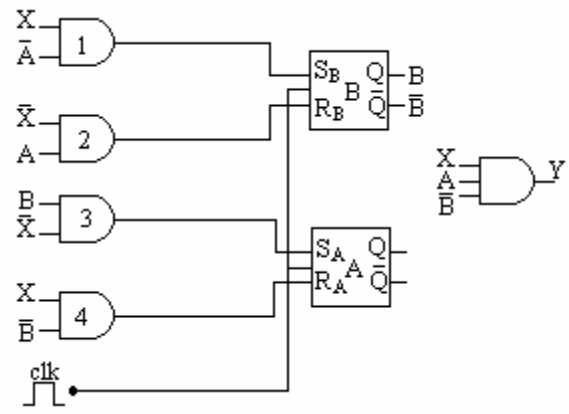
Devrenin incelenmesine bir başlangıç durumundan, örneğin AB=00 durumundan başlanır. X=1 olduğunda, FF girişleri;

$$S_A = X'.B=0, \quad R_A = X.B' = 1, \quad S_B = X.A' = 1 \quad \text{ve} \quad R_B = X'.A = 0$$

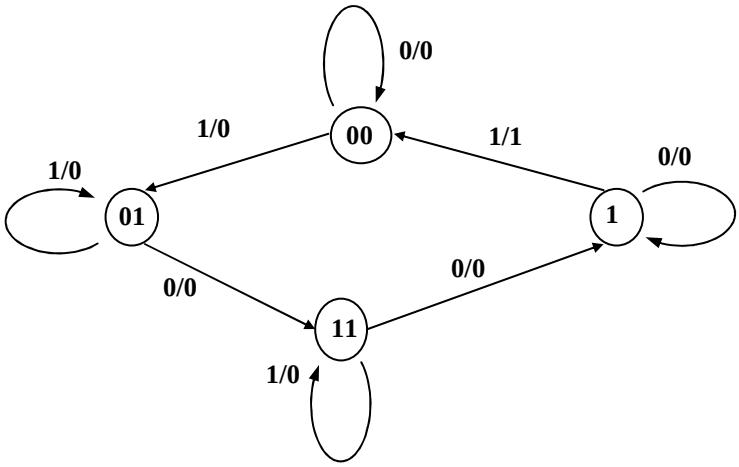
değerini alır. Bu giriş değerlerine sahip FF'lerin bir sonraki durumu AB=01 olarak belirlenir. Bu anda devrenin çıkışı; **y=X.A.B'=**0 olur.

Devrenin incelenmesinden; devrenin alabileceği durumları temsil eden AB değişkenlerinin, 00, 01, 10 ve 11 kombinasyonlarını alabileceği bulunur. Bu kombinasyonlar devrenin durumları olarak alınıp devredeki durum değişkenleri incelenirse, Şekil 10.7'deki durum geçiş şeması elde edilir.

Devrenin çalışmasını yorumlamaya yardımcı olan durum şemasını bu şekilde elde etmek karmaşık bir yöntemdir. Bu nedenle, bir devrenin durum şeması daha sonra anlatılacak durum tablosundan doğrudan çizilebilir.



Şekil 10.6. Ardışıl devre lojik şeması.



Şekil 10.7. Ardışıl devrenin durum şemasının çizilmesi.

10.1.2. Durum Geçiş Tabloları

Durum geçiş tablosu, ardışıl bir devrede mevcut durum ve giriş-çıkış değişkenleri arasındaki ilişkileri sergileyen bir tablodur. Bu tablo hazırlanırken, devrenin durum değişkenleri sayısı ile girdilerin sayısı göz önüne alınır. Bu durumda tabloda, flip-flopların tetikleme palsından önceki durumunu belirten ‘mevcut durum’, tetikleme palsı uygulandıktan sonraki durumu ifade eden ‘sonraki durum’ ve ‘çıkış’ olmak üzere üç kısım bulunur. Hem sonraki durum, hem de çıkış bölümleri, X=0 ve X=1 için olmak üzere 2 sütundan oluşur.

Tablo oluştururken, durum değişkenleri sayısı ile girdi sayısı göz önüne alınır. Buna göre devrede ‘m’ tane FF varsa, ‘m’ tane durum değişkeni tanımlanır ve devrede 2^m tane değişik

durum söz konusu olur. Ayrıca devrenin ‘n’ tane girişi varsa, 2^n tane farklı giriş değerleri birleşimi söz konusu olur.

Tablo oluşturulurken, önce değişken bileşimlerini içeren bir sütun hazırlanır. Daha sonra, durum tablosunun oluşturulmasına varsayılan bir ilk değerle başlanır. Analiz her ne kadar keyfi bir durumla başlayabilirse de, Şekil 10.6’daki devreyi incelemeye ‘00’ durumu ile başlayacağız.

Mevcut durum ‘00’ iken; A=0 ve B=0 olur. Bu anda her iki FF reset konumunda ve X=0’dır. Bu değerlerin uygulandığı Şekil 10.6’daki, ‘VE’ kapılarından hiç birisi mantık '1' sinyali üretmez ve dolayısıyla sonraki durum değişmeden kalır. Bu durumda, AB=00 ve X=0 iken, tetikleme sinyalinden sonraki durum ‘00’ olarak belirlenir.

AB=00 iken X=1 olunca; 1. kapı B FF’sinin S girişine ‘1’ sinyali üretirken, 4. Kapı A FF’sinin R girişine ‘1’ sinyali üretir. Tetikleme sinyali FF’leri tetikleyince, A FF’si çıkışı ‘0’ ve B FF’si çıkışı ‘1’ olur. Bu bilgiler tabloya taşınırsa, sonraki durum AB=01 olarak oluşur.

Aynı şekilde, diğer üç olası mevcut durumdan başlayarak sonraki durum elde edilebilir. Elde edilen değerler; girişlerin, mevcut durumun ve kullanılan FF türünün bir fonksiyonudur (Şekil 10.8).

MEVCUT DURUM	SONRAKİ DURUM		ÇIKIŞ	
	X=0	X=1	X=0	X=1
AB	AB	AB	Y	Y
00	00	01	0	0
01	11	01	0	0
10	10	00	0	1
11	10	11	0	0

Şekil 10.8. Şekil 10.6’daki devrenin durum geçiş tablosu.

Lojik devreden çıkış bölümünün değerlerini bulmak daha kolaydır. Örnek devrede, yalnızca X=1, A=1 ve B=0 iken ‘y’ çıkışı 1’e eşittir. Bu nedenle; mevcut durum=10 ve X=1 durumunda y=1, bu durum dışındaki bütün durumlarda çıkış sütunları '0' olarak yazılır.

Ardışık bir devrenin harici çıkışları, mantık elemanlarından ve FF çıkışlarından alınabilir. Mantık kapılarından çıkış alınması halinde durum tablosunda çıkış bölümü gereklidir. Çıkış olarak yalnızca FF çıkışları söz konusu ise, FF’lerin çıkışları mevcut durum sütununda zaten verildiğinden durum tablosunun çıkış bölümü iptal edilebilir.

10.1.3. Durum Denklemleri

Durum denklemleri, flip-flop'ların durum geçiş şartlarını tanımlayan matematiksel bir ifadedir. Denklemin sol tarafı FF'nin sonraki durumunu, bir boolean fonksiyonu olan sağ taraf ise sonraki durumu 1'e eşitleyen mevcut durum şartlarını tanımlar. Sonraki durum şartlarının harici giriş değişkenleri ve FF değerleriyle tanımlaması dışında, durum denklemi şeklen FF karakteristik denklemine benzer.

Durum denklemi, doğrudan doğruya durum tablosundan elde edilir. Örneğin; A FF'sine ait durum denklemi Şekil 10.8'den elde edilebilir.

Sonraki durum sütunlarından A FF'sinin '1' durumuna dört kez geçtiğini görürüz. Bu durumlar; X=0 iken AB= 01, 10, 11 durumları ile, X=1 iken AB=11 olduğu bileşenlerdir. Anlatılanlar bir durum denkleminde matematiksel olarak;

$$A(t+1)=(A'B+AB'+AB)X' + ABX$$

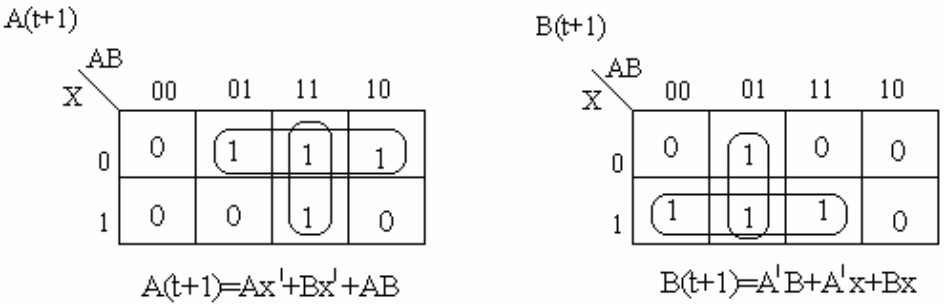
formülü ile ifade edilebilir. Denklemin sol tarafı; FF'nin harf sembolünü ve FF'nin bir sonraki alacağı değeri ifade eden zaman fonksiyon sembolünü içerir [A(t+1)].

Aynı şekilde, B FF'sinin dört kere '1' değerine sahip olduğu durum tablosunda görülmektedir. Bu durumda, tablodan faydalanarak B FF'si için;

$$B(t+1)=A'BX'+(A'B'+A'B+AB)X$$

denklemi yazılabilir.

Durum denklemlerinin sağ tarafı mevcut durum için bir boolean fonksiyonudur. Bu fonksiyon 1'e eşit olunca, bir sonraki tetikleme palsı ile A FF'sinin sonraki durumu '1' olur. Fonksiyonun '0' olması ise, sonraki durumu '0'a eşitler.



Şekil 10.9. FF durum denklemlerinin Karnaugh haritası yardımıyla sadeleştirilmesi.

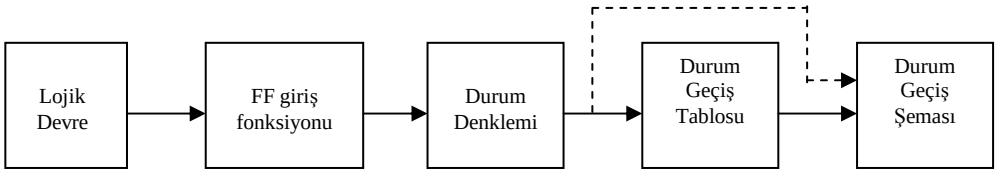
A ve B FF' lerinin durum denklemleri, Şekil 10.9'daki Karnaugh haritaları ile sadeleştirilebilir. Yukarıda bulunan durum denklemleri Karnaugh haritalarına taşınır ve Karnaugh yardımıyla sadeleştirme yapılırsa, sadeleştirme sonucunda;

$$A(t+1)=Ax'+Bx'+AB \qquad \text{ve} \qquad B(t+1)=A'B+A'x+Bx$$

denklemleri bulunur.

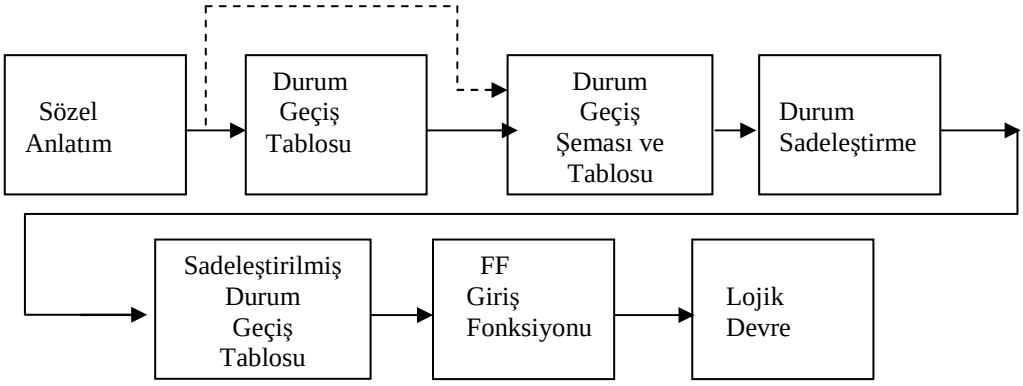
10.2. Ardışıl Devre Tasarımı

Ardışıl bir devrenin çözümlenmesi, lojik devreden yola çıkarak durum geçiş şeması ve durum geçiş tablosunun oluşturulması ve böylece devrenin giriş – çıkış ilişkisinin saptanması ile sonuçlanan bir işlemdir. Bu işlem, Şekil 10.11'deki blok şemada özetlenmektedir.



Şekil 10.11. Ardışıl devre çözümleme işleminin aşamaları.

Ardışıl devrenin tasarımı ise; oluşturulması istenen devrenin sözcük olarak ifade edilmesinden başlayıp, devre çözümleme işleminin aşamalarının ters sırada uygulanarak lojik devrenin çizilmesi ile sonuçlanan bir işlemler dizisidir (Şekil 10.12).



Şekil 10.12. Ardışıl devre tasarım aşamaları.

Her tasarım işleminde olduğu gibi, tasarlanan devrenin en düşük maliyet ile gerçekleştirilmesini sağlamak amacıyla devrenin en az sayıda eleman ile gerçekleştirilmesi gerekir. Devre çizimi aşamasından önce, ardışıl bir devrenin maliyetini lojik kapıların ve FF'lerin belirlediği göz önünde bulundurularak, devre en basit şekline getirilir. Devrenin basitleştirilmesine yönelik bu işleme, **'durum sadeleştirme'** denir. Ardışıl devre tasarımında takip edilecek aşamalarda bulunan ve daha önce açıklanmayan konulardan birisi olan devre sadeleştirmesi (durum indirgenmesi) işlemini inceledikten sonra, tasarım aşamalarını örneklerle açıklayalım.

10.2.1. Durum Sadeleştirme (Durum İndirgeme)

Her tasarım sürecinde lojik devrenin maliyetinin en aza indirilmesi hedeflenir. Maliyeti düşürmenin en kısa yolu; kullanılan elemanların (FF ve kapı devreleri) sayısını azaltacak işlemlerin yapılmasıdır. Ardışıl bir devrede kullanılan eleman sayısını azaltmak amacıyla yapılan işlem, '**durum indirgeme'** veya '**durum sadeleştirme**' olarak adlandırılır. Durum indirgemesi, giriş–çıkış ilişkisini yerine getirecek devreler arasından en az donanıma gereksinim duyanı bulmaktır. Diğer bir deyişle, harici giriş-çıkış gereksinimlerini aynen koruyarak durum sayısını azaltma işlemidir. Durum indirgemesinin, kullanılan FF sayısını azaltırken gerekli lojik kapı sayısının artmasına neden olabileceğini unutmamak gerekir.

İndirgeme işlemini açıklamadan önce belirtilmesi gereken önemli bir nokta; indirgeme işleminin yalnızca giriş ve çıkış denklemleri veya değerleri ile ilgilenilmesi durumunda kullanılabileceğidir. Mevcut durumların doğrudan çıktı olarak kullanıldığı devrelerde (örneğin sayıcılar) yada oluşturulan ara durumların işlevinin olduğu sistemlerde durum indirgemesi yapılamaz.

Durum indirgemesi işlemi, 'yuvaya dönen–reset' ve 'yuvaya dönmeyen–nonreset' devreler için farklı şekillerde gerçekleştirilir.

Ardışıl bir devre, belirli işlemler sonucunda belirli bir duruma yani bir yuva durumuna dönüyorsa, '**yuvaya dönen devre'** olarak nitelendirilir. Yuvaya dönen devrelerde durum indirgemesi, eşdeğer durumları kaynaştırılması ile gerçekleştirilir. İki mevcut durumun eşdeğer sayılabilmesi için, gelecek durumlarının ve ürettikleri çıktıların aynı olması gerekir. Kaynaştırılacak durumlar, durum tablosundan yararlanılarak belirlenir.

Örnek 4: Şekil 10.13.a'da durum geçiş tablosu verilen ardışıl devrenin durum sayısını indirgeyelim.

Bu örnekte 'yuva', 'E' durumudur. Tablonun incelenmesinden; 'A' ve 'D' durumlarının aynı gelecek ve aynı çıktı değerlerine sahip yani eşdeğer oldukları görülür. Dolayısıyla bu iki durum A=D özdeşliği ile birleştirilirse; 'D' durumu silinir ve 'D' görülen yere 'A' yazılırsa, Şekil 10.13.b'deki yeni tablo elde edilir. Sonuç olarak; beş olan durum sayısı dörde ve devre için gerekli FF sayısı üçten ikiye indirgenmiş olur.

Durum şeması verilen bir devrede durum indirgeme işlemi yapılması gerektiğinde, durum şemasından durum tablosunun oluşturulması gerekir. Şekil 10.14'de yalnızca giriş - çıkış sırası önemli olan, iç durumların yalnızca gerekli sırayı sağlamak için kullanıldığı bir devrenin durum şeması görülmektedir. Bu nedenle, dairelerin içerisindeki işaretli durumlar ikili değerleri yerine harf sembolleri ile gösterilmiştir.

Şimdiki durum	Gelecek durum (Sonraki)		Çıktı	
	X=0	X=1	X=0	=1
A	A	C	0	1
B	C	D	1	0
C	D	E	1	1
D	A	C	0	1
E	E	E	0	1

(a)

Şimdiki durum	Gelecek durum		Çıktı	
	X=0	X=1	X=0	X=1
A	A	C	0	1
B	C	A	1	0
C	A	E	1	1
E	E	E	0	1

(b)

Şekil 10.13. Durum tablosu verilen bir devrenin durum sayısının azaltılması.

Durum şemasında daireler içerisindeki harfleri mevcut durum kabul ederek, gelecek durum ve çıkışları elde edebiliriz.

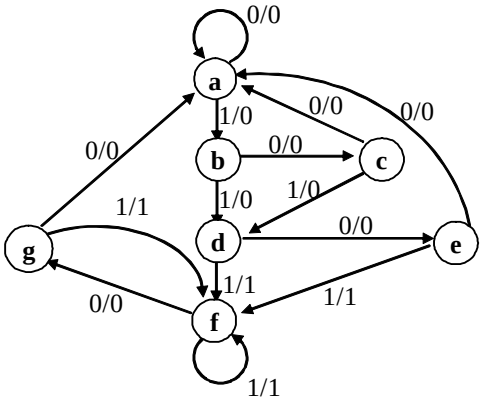
Devre ilk başlangıç 'a' durumunda iken, '0' girişi '0' çıkış üretir ve devre 'a' konumunda kalır. Mevcut durum 'a' iken girişin '1' olması; '0' çıkış ve sonraki durum olarak 'b' konumunu verir.

Mevcut durum 'b' iken giriş '0' olunca, çıkış '0' ve sonraki durum 'c' olur. Durumları inceleme işlemine devam edilmesiyle, Şekil 10.15.a'daki tablo oluşur.

Elde edilen durum geçiş tablosunda, 'D' ile 'F' ve 'E' ile 'G' durumlarının eşdeğer olduğu görülür. Eşdeğer olan çiftlerden 'F' ve 'G' durumları silinir ve gelecek durum sütununda 'F' görünen yere 'D', 'G' görülen yere 'E' yazılırsa, Şekil 10.15.b' deki tablo elde edilir.

İncelenen örnekte ardışıl devrenin durum sayısı 7'den 5'e indirilmiş, ancak gerekli FF sayısında bir azalma olmamıştır.

Çünkü, 'm' sayıda FF ile 3^m sayıda farklı durum temsil edilebilir. Bu nedenle; 3 FF ile 2^3=8 farklı durum temsil edilebilse de, 5 farklı durumda ancak 3 adet FF ile oluşturulabilir. Bununla beraber, Şekil 10.15.a'daki 7 durumlu tablo kullanılması ile kullanılmayan bir durum kalırken, 5 durumlu tablo kullanılması halinde kullanılmayan üç durum kalır. Kullanılmayan durumlar devre tasarımı sırasında farketmez durumları (don't care) ifade eder. Farketmez durumları genellikle daha basit bir Boolean fonksiyonu oluşturduğu için, 5 konumlu devre 7 konumlu devreye göre daha az lojik kapı gerektirebilir.



Şekil 10.14. Örnek durum şeması.

Şimdiki Durum	Gelecek Durum X=0	Gelecek Durum X=1	Çıktı X=0	Çıktı X=1
A	A	B	0	0
B	C	D	0	0
C	A	D	0	0
D	E	F	0	1
E	A	F	0	1
F	E	F	0	1
G	A	F	0	1

(a)

Şimdiki Durum	Gelecek Durum X=0	Gelecek Durum X=1	Çıktı X=0	Çıktı X=1
A	A	B	0	0
B	C	D	0	0
C	A	D	0	0
D	E	D	0	1
E	A	D	0	1

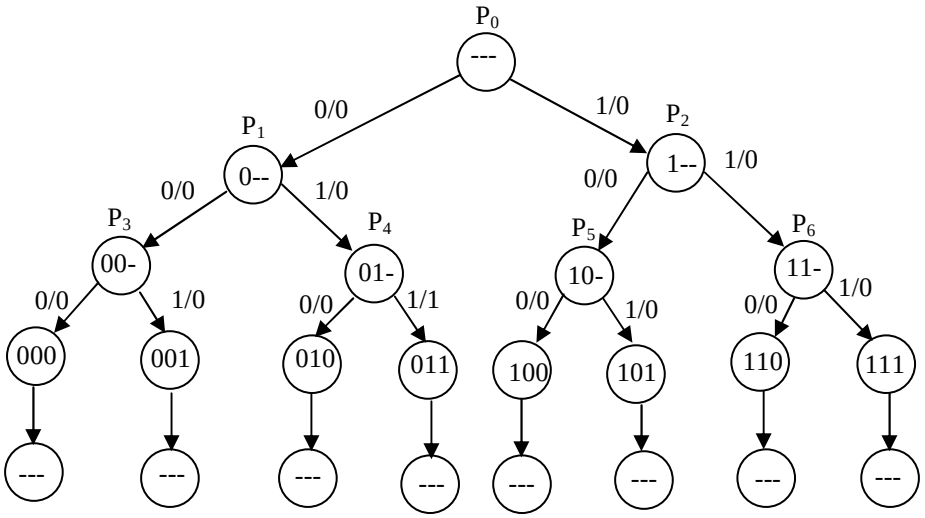
(b)

Şekil 10.15. Durum şemasından durum tablosunun elde edilmesi ve tablonun indirgenmesi.

Örnek 5: Parola Seçici:

Bir haberleşme sisteminde bilgiler üçerli bit öbekleri biçiminde iletilmektedir. Bu öbeklerden ‘011’ kombinasyonu anahtar bileşke (parola) olarak kabul edilmekte ve bu değerlere sahip öbeğin varlığı seçildiğinde çıktı ‘1’ olup, bir alarm devresini çalıştırmakta, diğer bütün kombinasyonlar için çıktı ‘0’ olmaktadır. Çıkışın ‘0’ olduğu durumlarda başlangıç durumuna dönülmektedir. Bu işlemleri gerçekleştirecek ardışık devreyi tasarlayalım.

Yapılan açıklamaları referans alan, sistemin durum geçiş şeması ve durum geçiş tablosu Şekil 10.16'daki gibi çizilebilir. Ayrıca durum geçiş tablosunda her bir üçerli grup oluşurken, makinenin belleğinde birikmiş bulunan bilgiler de yazılabilir.



(a)

Şimdiki durum	Gelecek durum X=0	Gelecek durum X=1	Çıktı X=0	Çıktı X=1
P_0	P_1	P_2	0	0
P_1	P_3	P_4	0	0
P_2	P_5	P_6	0	0
P_3	P_0	P_0	0	0
P_4	P_0	P_0	0	1
P_5	P_0	P_0	0	0
P_6	P_0	P_0	0	0

(b)

Şekil 10.16. Parola seçici durum geçiş şeması ve durum geçiş tablosu.

Ardışıl durum tablosunda, 'P_0' durumunun yuva durumu olduğu kolayca anlaşılabilir. Durum geçiş tablosundan, P_3, P_5, ve P_6 durumlarının hem X=0 hem de X=1 girdileri için aynı çıktıya ve gelecek duruma sahip oldukları görülmektedir. Bu durumlar P_3 durumu altında toplanırsa, Şekil 10.17'deki indirgenmiş durum tablosu elde edilir. İki tablonun karşılaştırılmasından; durum sayısının azaltılmasına karşılık, gerekli FF sayısında bir azalma olmadığı görülür.

Şimdiki durum	Gelecek durum X=0	X=1	Çıktı X=0	X=1
P_0	P_1	P_2	0	0
P_1	P_3	P_4	0	0
P_2	P_3	P_3	0	0
P_3	P_0	P_0	0	0
P_4	P_0	P_0	0	1

Şekil 10.17. Parola seçici indirgenmiş durum tablosu.

10.2.2. Durum Tahsisi (Durum Atanması)

Ardışıl bir devrenin bileşik devre kısmının maliyeti, bileşik devrelerin sadeleştirilmesi için kullanılan sadeleştirme yöntemleri ile azaltabilir. Ancak bileşik devrenin sadeleştirilmesi sırasında; başlangıçtaki harf simgeleriyle gösterilmiş olan durumlara birer ikili değer atama işlemi, yani durum ataması denen başka bir faktör ortaya çıkar. Bu faktör özellikle ardışıl devre dış giriş-çıkış uçları açısından değerlendirildiği zaman faydalıdır. Oysa, sayıcı ve zamanlayıcı gibi devrelerde erişilen durum aynı anda devrenin çıktısını oluşturduğundan, durum ataması konusunda hiçbir esneklik olamaz ve durumun ikili değeri çıktının değerine eşittir.

Mevcut birçok durum tahsisi yöntemi bulunmaktadır (Armstrons-Hunaphrey vb.). Ancak bu yöntemler içerisinden minimum maliyetli bir bileşik devre garanti edecek bir durum tahsisi (ataması) yöntemi belirlenememiştir. Bu yöntemler anahtarlama ve durum makinaları teorisi ile ilgili kaynaklarda bulunabilir. Biz burada ardışıl devrenin tasarım aşamalarını örneklerle inceleyelim.

10.2.3. Ardışıl Devre Tasarım Aşamaları ve Tasarım Örnekleri

Ardışıl devrenin tasarımı, devrenin özelliklerinin tanımlamasıyla başlar ve bir mantık şemasıyla veya mantık şeması elde etmek için kullanılabilecek Boolean fonksiyonları ile devam eder. Doğruluk tablosuyla tam olarak tanımlanabilen bileşik devrelerin tersine, sıralı devreler özellik tanımı için bir durum tablosu gerektirir. Bu nedenle ardışıl devrelerin tasarımında ilk adım; durum tablosu, durum şeması veya durum denklemlerinden birisinin oluşturulmasıdır (Şekil 10.12).

Diğer yandan, senkron ardışıl bir devre FF ve lojik kapılardan oluşur. Bu nedenle, tasarım sırasında kapıların oluşturduğu bileşik devre ve FF'lerin oluşturduğu ardışıl devrenin birlikte tasarlanması gerekir. Tasarım sırasında takip edilecek adımlar aşağıdaki gibi özetlenebilir:

i- Devre sözel olarak açıklanır. Bu arada devre şeması, zamanlama şeması veya diğer uygun bilgiler açıklanabilir. Bunlardan devrenin durum geçiş şeması çizilebilir.

ii- Devre konusunda verilen tüm bilgiler ve durum geçiş şeması değerlendirilerek durum tablosu oluşturulur.

iii- Durumlar simgelerle (örneğin harf) ifade edilerek, durum indirgeme yöntemleri yardımıyla durum sayısı azaltılmaya çalışılır.

iv- Elde edilen durum tablosunun harf içermesi halinde, her bir duruma karşılık binary değer atanır.

v- Eldeki durum sayısına bağlı olarak gereken FF sayısı belirlenir ve her bir FF'ye büyük harf sembol verilir.

vi- Kullanılacak FF türü (SR, JK, T, D türlerinden biri) seçilir.

vii- Durum geçiş şemasından yola çıkarak ve seçilen FF türünün davranışı göz önüne alınarak FF durum denklemleri oluşturulur.

viii- Karnaugh haritası veya farklı bir sadeleştirme yöntemi kullanılarak sadeleştirme yapılır ve FF giriş fonksiyonları elde edilir.

ix- Lojik devre çizilir ve FF bağlantıları gösterilir.

Ardışıl devre tasarımında tecrübe kazandıktan sonra, tasarım için sıralama aşamaları azaltmak ve tasarım işlemini hızlandırmak mümkündür. Ayrıca durum sayısının ve durumlara atanan ikili değerlerin bilinmesi durumunda 3 ve 4 nolu basamaklar atlanabilir.

Tasarımda kullanılacak FF türü tasarım sırasında belirlenebileceği gibi, tasarımcının piyasadan temin edebildiği türlere de bağlı olabilir. Dijital sistemlerin birçoğu, mevcutlar arasında en kullanışlı FF türü olması nedeniyle JK FF'ler ile oluşturulur. FF türlerinin tümünün bulunduğu durumlarda; veri transferi gerektiren (kaymalı kaydedici gibi) uygulamalarda RS veya D tipi, sayıcı gibi uygulamalarda JK veya T tipi, genel uygulamalarda ise JK tipi FF'ler tercih edilir.

Örnek 6: Şekil 10.18'de blok şeması ve çalışma programı verilen lojik devreyi RS, JK, D ve T tipi FF'ler kullanarak ayrı-ayrı tasarlayalım.

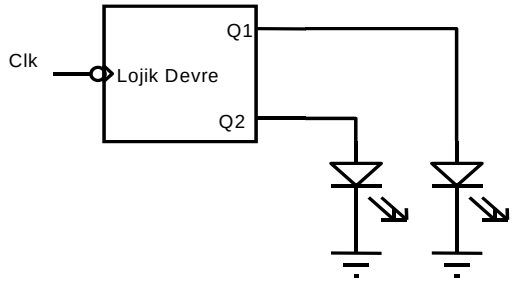
Çalışma programında görüldüğü gibi, ilk tetikleme işareti ile Q_1=0, Q_2=1 olacak yani D_1 sönük D_2 yanık olacaktır. Daha sonraki tetikleme işaretlerinde sırasıyla;

Q_1=0, Q_2=0	yani	D_1 ve D_2 sönük	durumları oluşacaktır.
Q_1=1, Q_2=0		D_1 yanık, D_2 sönük	
Q_1=1, Q_2=1		D_1 ve D_2 yanık	

Özetlenen çıkış dizisini gerçekleştiren devre ilk durumuna döner ve işlem tekrar başlar. Açıklamalardan anlaşılacağı üzere, devre çıkışlarının değişme süresini tetikleme işaretinin

frekansı tayin eder. Tetikleme işaretinin frekansının artması ile LED'in yanıp sönme süreleri azalır.

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



Şekil 10.18. Örnek lojik devre blok şeması ve çalışma programı.

i- Tasarım sırasında daha önce verilen işlem sırasından bazıları ihmal edilebilir. Bu durumda; durum sayısı 4 olduğundan kullanılacak FF sayısı 2 olarak tespit edilir.

ii- Seçilen FF tipi için, devre çıkış tablosu ve FF geçiş tablosundan faydalanılarak devre durumu geçiş tablosu hazırlanır (Şekil 10.19).

iii- Her bir geçiş tablosu Karnaugh haritasına uygulanarak, sadeleştirilmiş lojik ifade elde edilir.

iv- Bulunan eşitliklere göre, FF'lere kapılar eklenerek lojik devre çizilir.

Çözüm 1: İlk olarak RS FF ile tasarım yapılacağına ve kullanılacak FF sayısı 2 olacağına göre; devrenin çıkış durumlarını gösteren devre çıkış tablosu hazırlanır (Şekil 10.19.a.). Daha sonra Şekil 10.19.b'de verilen RS FF geçiş tablosundan faydalanılarak, Q_1 için S_1, R_1 ve Q_2 için S_2, R_2 geçişleri oluşturulur (Şekil 10.19.c).

Şekil 10.19.c'deki durum geçiş tablosunun oluşumunda kullanılacak FF'ye ait geçiş tablosundan faydalanılır. Q_1 ve Q_2' nin çıkışı sırasıyla; Q_1 çıkışı için 0-0-1-1-0 ve Q_2 çıkışı için 1-0-0-1-1 şeklindedir. Bu değişimlere göre S_1-R_1 ve S_2-R_2 değerleri yazılır.

Bu değerler;

Q_1; 0'dan 0'a geçerken S_1=0, R_1=d

0'dan 1'e geçerken S_1=1, R_1=0

1'den 1'e geçerken S_1=d, R_1=0

1'den 0'a geçerken S_1=0, R_1=1

şeklindedir.

Q_1 için yazılan bu değerler gibi, Q_2 için R_2 ve S_2 değerlerinin bulunarak tabloya yazılması ile Şekil 10.19.c'deki durum geçiş tablosu elde edilir.

	Q1	Q2	S1	R1	S2	R2
	0	1				
	0	0				
	1	0				
	1	1				
İlk durum→	0	1				

a) Devre çıkış tablosu.

Qt	Qt+1	S	R
0	0	0	d
0	1	1	0
1	0	0	1
1	1	d	0

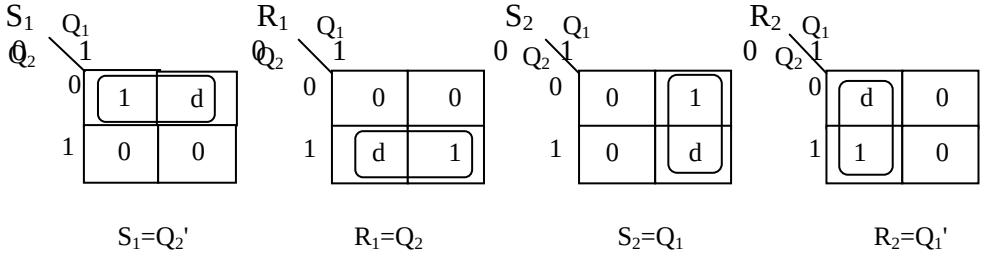
b) RS FF geçiş tablosu.

Q_1	Q_2	S_1	R_1	S_2	R_2
0↓	1↓	0	d	0	1
0↓	0↓	1	0	0	d
1↓	0↓	d	0	1	0
1↓	1↓	0	1	d	0
0↓	1↓				

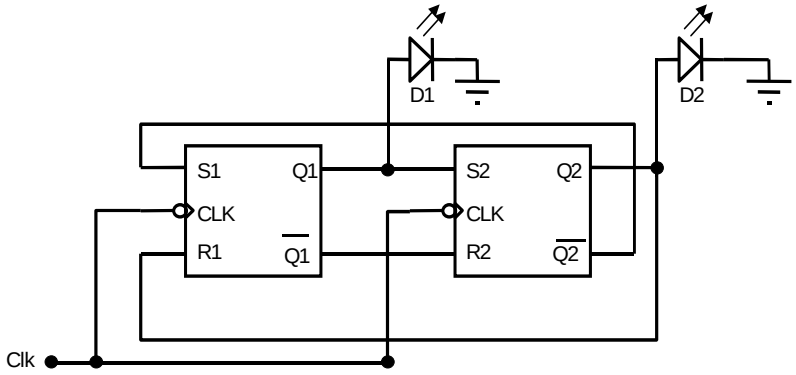
c) Devre durum geçiş tablosu.

Şekil 10.19. Ardışıl devre tasarımında devre çıkış tablosu ve RS FF geçiş tablosundan faydalanılarak devre durum geçiş tablosunun oluşturulması.

Bulunan değerler S_1, R_1, S_2 ve R_2 için oluşturulan Karnaugh haritalarına taşınıp, bu değişkenler için basitleştirilmiş eşitlikler elde edilir (Şekil 10.20). Elde edilen eşitlikleri temsil eden lojik devrenin çizilmesi ile, Şekil 10.21'deki devre oluşur.



Şekil 10.20. Geçiş tablolarının Karnaugh haritalarına taşınması ve eşitliklerin yazılması.



Şekil 10.21. Şekil 10.18'deki işlemi, RS FF'lerle gerçekleştirecek ardışıl devrenin lojik şeması.

Çözüm 2: Aynı problemi JK FF'ler kullanarak gerçekleştirelim. Aynı işlem sırası ve 2 adet JK FF kullanılarak, aynı işlemi gerçekleştirecek devreyi çizelim.

Önce Şekil 10.22.a'da verilen JK geçiş tablosunu kullanarak, Şekil 10.22.b'de verilen J_1, K_1, J_2, K_2 girişleri için FF geçiş tablosu oluşturulur. FF'lerin geçiş tablosundaki değerlerin Karnaugh haritalarına taşınması ve gruplandırma yapılması sonucunda Şekil 10.22.c'deki eşitlikler elde edilir.

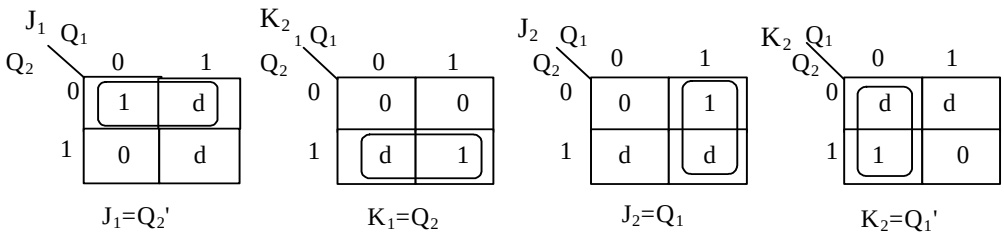
Qt	Qt+1	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

a) JK FF geçiş tablosu.

Başlangıç durumu→

Q_1	Q_2	J_1	K_1	J_2	K_2
0 ↓	1 ↓	0	d	d	1
0 ↓	0 ↓	1	d	0	d
1 ↓	0 ↓	d	0	1	d
1 ↓	1 ↓	d	1	d	0
0 ↓	1 ↓				

b) Devre durum geçiş tablosu.

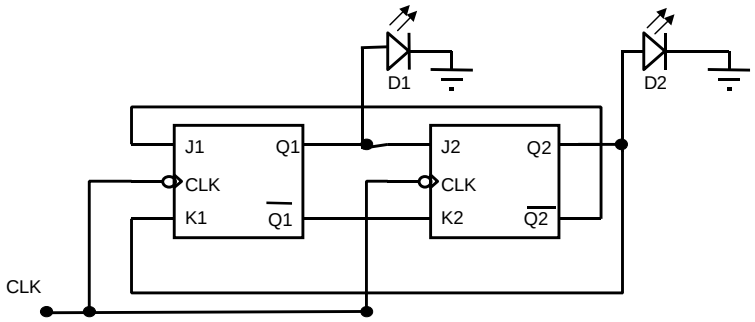


$J_1=Q_2'$ $K_1=Q_2$ $J_2=Q_1$ $K_2=Q_1'$

c) Karnaugh haritaları ile lojik fonksiyonların elde edilmesi.

Şekil 10.22. Şekil 10.18'deki devrenin JK FF'lerle tasarlanması.

Karnaugh şemalarından yazılan eşitlikleri temsil eden lojik devrenin çizilmesi ile, Şekil 10.23'deki ardışıl devre oluşur.



Şekil 10.23. Şekil 10.18'deki ardışıl devrenin JK FF'lerle gerçekleştirilmesi.

Çözüm 3: İstenilen devreyi D tipi FF'lerle gerçekleştirmek için; Şekil 10.24.a'daki D tipi FF geçiş tablosundan faydalanılarak Şekil 10.24.b'deki FF'lere ait geçiş tablosu oluşturulur.

Geçiş tablolarının Karnaugh haritalarına taşınması ile, gerçekleştirilmesi gerekli fonksiyonlar yazılır (Şekil 10.24.c). Yazılan fonksiyonları temsil eden lojik devrenin çizilmesi ile Şekil 10.24.d'deki ardışıl devre oluşur.

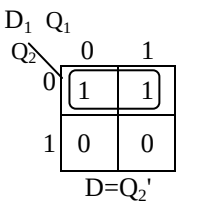
Qn	Q(n+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

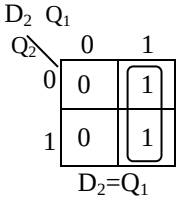
a) D tipi FF geçiş tablosu.

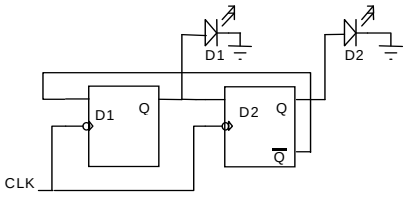
Q_1	Q_2	D_1	D_2
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1
0	1		

Başlangıç durumu →

b) FF'ler için geçiş tablosu.







c) Geçiş tablolarının Karnaugh haritalarına taşınması. **d)** Tasarlanan devrenin D tipi FF' lerle gerçekleştirilmesi.

Şekil 10.24. Şekil 10.18'deki devrenin D FF'lerle gerçekleştirilmesi.

Çözüm 4: İstenilen devreyi T tipi FF'lerle oluşturmak için; Şekil 10.25.a'daki T tipi FF geçiş tablosundan faydalanılarak Şekil 10.25.b'deki T_1 ve T_2 FF'lerine ait geçiş tabloları oluşturulur.

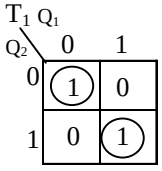
Oluşturulan T_1 ve T_2 değerlerinin Karnaugh haritalarına taşınması ile, gerçekleştirilecek devreyi temsil eden fonksiyonlar yazılır (Şekil 10.25.c). Yazılan fonksiyonları gerçekleştirecek devrenin çizilmesiyle, Şekil 10.25.d'deki devre oluşur.

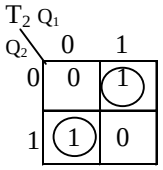
Qn	Q(n+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

a) T tipi FF geçiş tablosu.

Q_1	Q_2	T_1	T_2
0	1	0	1
0	0	1	0
1	0	0	1
1	1	1	0
0	1		

b) Devre durum geçiş tablosu.

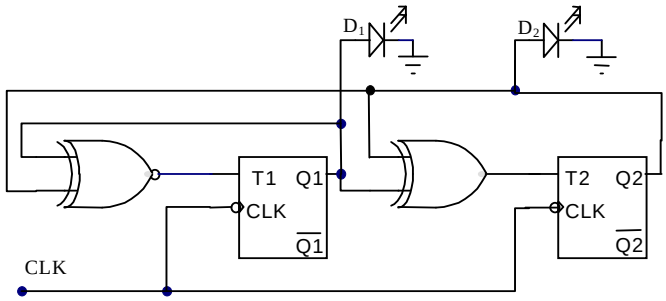




$T_1 = Q_1'Q_2' + Q_1Q_2 = \overline{Q_1 \odot Q_2}$

$T_2 = Q_1'Q_2 + Q_1Q_2' = \overline{Q_1 \oplus Q_2}$

c) Durum geçiş tablosunun Karnaugh haritalarına taşınması.



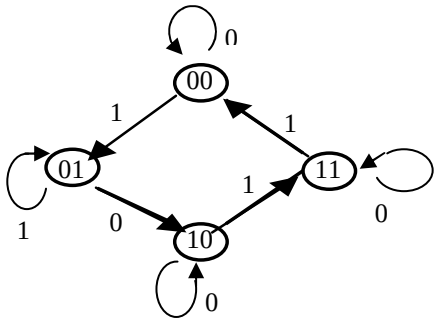
d) Tasarlanan devrenin T tipi FF'ler ile oluşturulması.

Şekil 10.25. Şekil 10.18'deki devrenin T tipi FF'lerle gerçekleştirilmesi.

Örnek 8: Şekil 10.26.a'da durum şeması verilen tetiklemeli sıralı devreyi, JK FF'lerle oluşturmak için gerekli tasarımı yapalım.

Durum geçiş şeması, ikili değerleri belirlenmiş olan dört durumdan oluşur. Oklar kesme (/) işareti olmaksızın tek bir ikili rakamla işaretlendiği için, bir giriş değişkenine karşılık çıkış değişkeni olmadığı ve FF'lerin durumlarının devrenin çıkışları olarak değerlendirilebileceği kabul edilir. Dört durumu göstermek için ihtiyaç duyulan iki FF; A ve B harfleriyle, giriş değişkeni ise; 'X' ile gösterilirse; Şekil 10.26.b' deki durum tablosu elde edilir.

Şekil 10.26.b'de oluşturulan durum tablosunda çıkış bölümü yoktur. Mevcut durum ve giriş değişkenlerini kullanarak durum tablosunun doğruluk tablosu şeklinde düzenlendiği form, Şekil 10.27'de görülmektedir. Gerekli durum geçişlerini sağlayacak olan FF giriş koşulları her bir mevcut durum için sonraki durum ve giriş değerleri Şekil 10.27'den aktarılarak yazılabilir. Yazım sırasında, kullanılan FF'ye ait geçiş tablosundan faydalanılır (Şekil 10.26.c).



a) Durum Şeması.

		Sonraki Durum			
Mevcut Durum		X=0		X=1	
A	B	A	B	A	B
0	0	0	0	0	1
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	1	0	0

b) Durum Tablosu.

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

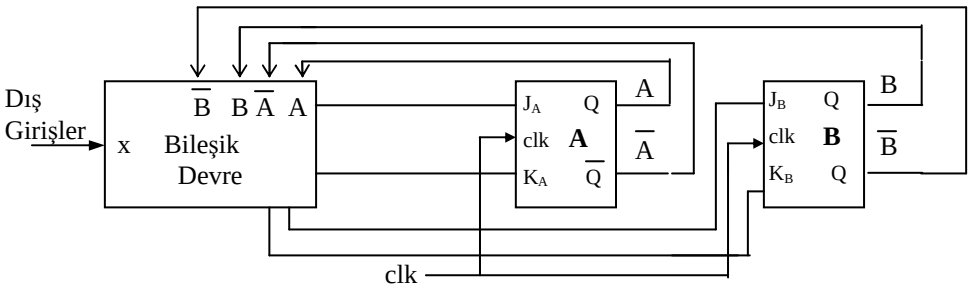
c) JK FF Geçiş Tablosu.

Şekil 10.26. Durum şeması verilen bir devrenin durum tablosunun oluşturulması.

Kullanılan FF'ler A - B, girişler ise J_A, K_A, J_B, K_B sembolleriyle gösterilir ve tasarlanan ardışıl devrede bulunacak bileşik devre blok olarak gösterilirse; Şekil 10.27'deki devre

oluşur. Bileşik devreye uygulanan girişler; harici giriş ve FF'lerin mevcut durum değerleridir.

Mevcut Durum		Giriş	Sonraki Durum		Flip-Flop Girişleri			
A	**B**	**x**	**A**	**B**	J_A	K_A	J_B	K_B
0	0	0	0	0	0	D	0	D
0	0	1	0	1	0	D	1	D
0	1	0	1	0	1	D	D	1
0	1	1	0	1	0	D	D	0
1	0	0	1	0	D	0	0	D
1	0	1	1	1	D	0	1	D
1	1	0	1	1	D	0	D	0
1	1	1	0	0	D	1	D	1

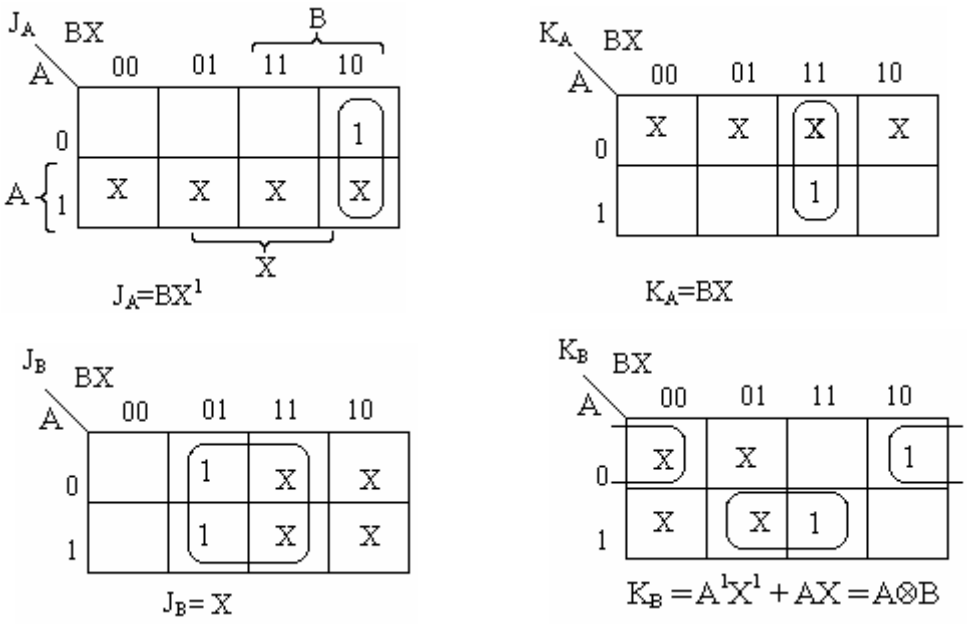


Şekil 10.27. Şekil 10.26'da verilen devrenin geçiş tablosu ve oluşturulacak devrenin blok şeması.

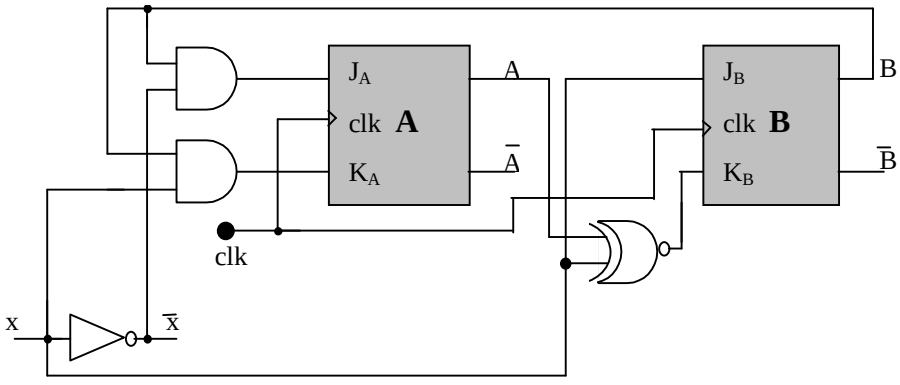
Tasarlanan devre için oluşturan geçiş tablosunda FF'lerin girişlerine uygulanması gerekli değerler belirlendikten sonra, tablodaki değerler Karnaugh haritalarına taşınır. Karnaugh haritalarında gruplandırma yapılır ve grupları temsil eden eşitlikler yazılır (Şekil 10.28). Yazılan eşitlikler, çizilecek devreyi temsil eder.

Elde edilen fonksiyonların lojik kapılar ve FF'ler kullanılarak çizilmesi ile, Şekil 10.29'daki devre oluşur.

Tasarım sırasında lojik devre çizme aşamasına kadar gerçekleştirilen aşamalardan bir kısmı kaldırılabilir. Örneğin; Karnaugh haritaları için gerekli bilgiler, doğrudan durum tablosundan elde edilebilir. Bu durumda işlemler kısalır.



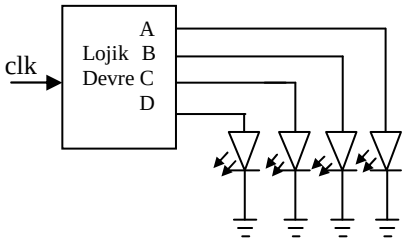
Şekil 10.28. İfadelerin Karnaugh haritalarına taşınması ve eşitliklerin yazılması.



Şekil 10.29. Tasarlanan devrenin mantık şeması.

Örnek 9: Şekil 10.30'da blok şeması ve çalışma programı verilen devreyi J-K tipi FF'ler kullanarak tasarlayalım.

Çözüm : Tasarımda 4 adet J-K FF kullanılacağından, J-K FF'ye ait geçiş tablosu yardımıyla J-K FF'lerin girişlerine ait geçiş değerleri bulunur (Şekil 10.31).



a) Blok şeması.

Clk	A	B	C	D
0	1	1	1	1
1	1	0	1	0
2	0	1	0	1
3	1	1	0	0
4	0	0	1	1
5	1	0	1	1
6	1	0	0	1
7	1	0	0	0
8	0	1	0	0
9	0	0	1	0
10	0	0	0	1

b) Oluşturulacak devrenin çalışma programı.

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

c) JK FF geçiş tablosu.

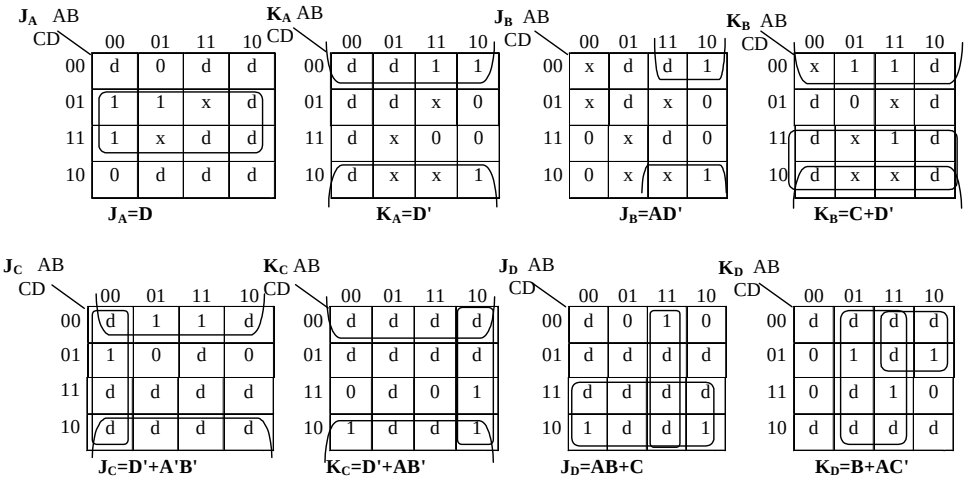
Şekil 10.30. Ardışıl devre tasarımı ve çalışma tablosu.

Clk	A	B	C	D	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D
0	1	1	1	1	d	0	d	1	d	0	d	1
1	1	0	1	0	d	1	1	d	d	1	1	d
2	0	1	0	1	1	d	d	0	0	d	d	1
3	1	1	0	0	d	1	d	1	1	d	1	d
4	0	0	1	1	1	d	0	d	d	0	d	0
5	1	0	1	1	d	0	0	d	d	1	d	0
6	1	0	0	1	d	0	0	d	0	d	d	1
7	1	0	0	0	d	1	1	d	0	d	0	d
8	0	1	0	0	0	d	d	1	1	d	0	d
9	0	0	1	0	0	d	0	d	d	1	1	d
10	0	0	0	1	1	d	1	d	1	d	d	0
ilk durum	1	1	1	1								

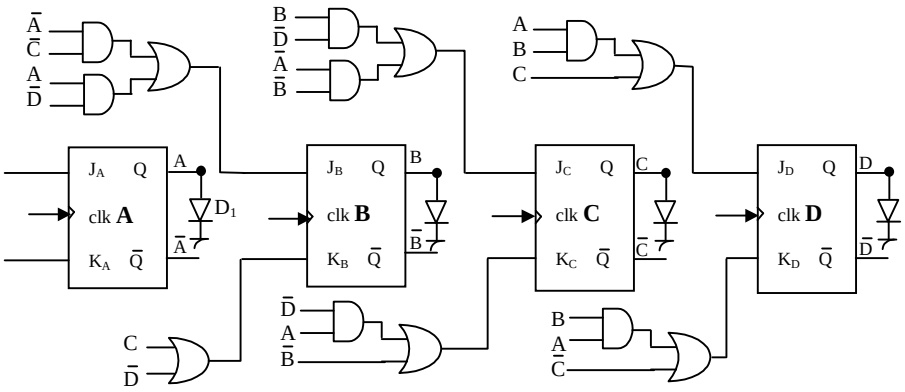
Şekil 10.31. Çalışma programına göre oluşturulan. FF'lere ait geçiş tabloları.

Geçiş tablosundaki bilgilerin Karnaugh haritalarına taşınması gerekir. Taşınma sırasında kullanılmayan kombinasyonlar 'farketmez' olarak alınırsa, Şekil 10.32'daki Karnaugh

haritaları ve lojik fonksiyonları elde edilir. Karnaugh haritalarında ‘d’ ile gösterilen ve FF’nin özelliğinden dolayı oluşan ‘farketmez’ durumu ile, kullanılmayan kombinasyonlardan dolayı oluşan ve ‘x’ ile gösterilen ‘farketmez’ durumu olmak üzere 2 türlü farketmez bulunmaktadır. Her iki ‘farketmez’ durumu aynı çerçevede değerlendirilebilir.



Şekil 10.32. Şekil 10.30’da verilen devre için elde edilen geçiş durumlarının Karnaugh haritalarına taşınması ve lojik eşitliklerin yazılması.



Şekil 10.33. Tasarlanan devrenin lojik şeması.

Elde edilen lojik eşitlikleri gerekleştirecek bileşik devre ve JK FF'ler birlikte çizilirse, istenilen çalışma programını gerçekleştirilecek ardışıl devrenin lojik şeması elde edilir (Şekil 10.33).

FF'ler kullanılarak gerçekleştirilen ve ardışıl devreler olan sayıcılar, kaydediciler vb devrelere ait tasarım şekilleri daha sonraki bölümlerde detaylı olarak incelenecektir.

Tekrarlama ve Çalışma Soruları

1. 'Ardışıl devreyi' tarif ederek, bileşik devre ile farkını açıklayınız.

2. Ardışıl devreleri, uygulanan sinyallerin zamanlamasına bağlı olarak gruplandırınız.

3. 'Tetiklemeli ardışıl devreleri' tanımlayınız.

4. Senkron ve asenkron ardışıl devreler arasındaki temel farkı açıklayınız.

5. Ardışıl devrenin davranışını etkileyen faktörler nelerdir?

6. Ardışıl devre analiz yöntemlerini sıralayınız.

7. Durum geçiş şeması yöntemini özetleyiniz.

8. Tam toplayıcı devresinin durum geçiş şemasını çıkarınız.

9. Durum tablosu oluşturma işlemini özetleyiniz.

10. Ardışıl toplayıcıda elde tablosu için durum geçiş şemasını oluşturunuz.

11. Ardışıl toplayıcı durum geçiş şemasından faydalanarak durum tablosunu oluşturunuz.

12. Ardışıl toplayıcı için oluşturulan durum tablosunu kullanarak durum denklemini çıkarınız. Bulduğunuz durum denklemini Karnaugh haritaları kullanarak sadeleştiriniz.

13. Durum denkleminde eşitliğin sağ ve sol taraflarının taşıdıkları anlamları açıklayınız.

14. '101' sayı öbeğini 'parola' olarak kabul eden bir sistemin durum şemasını ve durum tablosunu oluşturarak, sadeleştirme işlemini yapınız.

15. Ardışıl devre çözümleme işlemi aşamalarını blok şema kullanarak özetleyiniz.

16. Ardışıl devre tasarım aşamalarını maddeler halinde sıralayınız.

17. 'Durum indirgeme' terimini açıklayınız.

18. Durum indirgemesi, ardışıl devrelerin hangi durumları için geçerlidir?

19. Aşağıda verilen durum tablosundaki durum sayısını azaltınız.

Şimdiki durum	Gelecek durum (Sonraki)		Çıktı	
	X=0	X=1	X=0	X=1
A	B	C	0	1
B	C	D	1	0
C	B	C	0	1
D	D	A	1	0
E	B	C	0	1

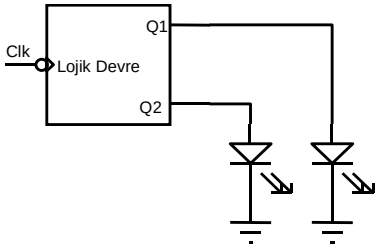
20. Aşağıda verilen durum tablosundaki durum sayısını azaltınız.

Şimdiki durum	Gelecek durum (Sonraki)		Çıktı	
	X=0	X=1	X=0	X=1
A	A	B	1	0
B	C	D	0	1
C	B	C	0	0
D	A	C	1	1
E	B	C	0	0
F	B	C	0	0

21. Ardışıl devrelerde durum ataması işlemini tartışınız.

22. Sırası ile 0'dan 10'a kadar tek sayıları çıkış olarak veren ardışıl devreyi JK FF'ler kullanarak gerçekleştiriniz.

23. Blok şeması ve çalışma tablosu verilen ardışıl devreyi RS FF'ler kullanarak tasarlayınız.



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

24. 0-7 arasındaki tek sayıları çıkış olarak veren ardışıl devreyi JK FF'ler kullanarak tasarlayınız.

25. 0-15 arasındaki çift sayıları sırası ile çıkış olarak veren ardışıl devreyi JK FF'ler kullanarak tasarlayınız.

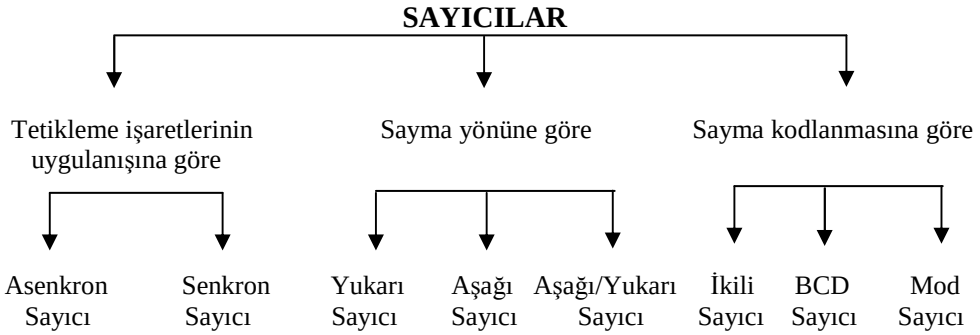
BÖLÜM 11

SAYICILAR

(COUNTERS)

Giriş

Giriş darbelerine bağlı olarak belirli bir durum dizisini tekrarlayan lojik devreler, **'sayıcı'** olarak adlandırılır. Çok değişik alanlarda kullanılan sayıcı devreleri, FF'lerin uygun şekilde bağlanmalarıyla elde edilir. Dijital ölçü, kumanda ve kontrol sistemlerinin en önemli elemanları olan sayıcıları, değişik referanslara göre sınıflandırmak ve adlandırmak mümkündür. Sayıcılar en genel şekli ile aşağıdaki gibi sınıflandırılabilir.



A- Sayıcıların tetikleme sinyallerinin uygulama zamanına göre sınıflandırılması:

Tetikleme sinyallerinin FF'lere uygulanış zamanına göre sayıcılar iki gruba ayrılır:

i- Asenkron (farklı zamanlı) sayıcılar,

ii- Senkron (eş zamanlı) sayıcılar.

Asenkron sayıcılarda, sayma işlemi için kullanılan tetikleme sinyali ilk FF'ye uygulanır. İlk FF'nin Q veya Q^{l} çıkışından alınan sinyal ile daha sonra gelen FF tetiklenir. Diğer bir deyişle; FF'ler birbirini tetiklerler.

Senkron sayıcılarda, tetikleme sinyalleri sayıcıyı oluşturan bütün FF'lere tek bir hattan aynı anda uygulanır. Bu durumda devrede bulunan tüm FF'ler birlikte tetiklenir.

B- Sayıcıların sayma yönüne göre sınıflandırılması:

Sayıcılar sayma yönüne göre üç grupta incelenebilir:

i- Yukarı / İleri sayıcılar (Up counters): Sayıcı 0'dan başlayıp yukarı doğru sayma işlemi gerçekleştiriyorsa, **'yukarı sayıcı'** denir.

ii- Aşağı / Geri sayıcılar (Down counters): Sayıcı belirli bir sayıdan başlayıp 0'a doğru sayma işlemi yapıyorsa, **'aşağı sayıcı'** olarak adlandırılır.

iii- Yukarı-Aşağı sayıcılar (Up-Down Counters): Sayıcı her iki yönde sayma işlemini gerçekleştirebiliyorsa, **'yukarı-aşağı sayıcı'** olarak isimlendirilir.

C- Sayıcıların sayma kodlanmasına göre sınıflandırılması:

Sayıcılar, sayıcının takip edeceği sayma dizisi referans alınarak gruplandırılabilir. Sayıcılar girişlerine uygulanan darbe miktarına bağlı olarak 2^n değişik durum alabilir. Diğer bir deyişle; n sayıdaki FF ile, 2^n sayıda sayma işlemi yapılır. Üç adet FF kullanan sayıcı 8 kademe, dört adet FF kullanan sayıcı 16 kademe sayma gerçekleştirir. Sayıcılar, sayabileceği maksimum değeri sayabileceği gibi, belirli bir değere kadar sayma yapabilir. Sayıcılar, sayılan dizinin kodlanmasına göre: İkili sayıcı, BCD sayıcı, Mod sayıcı vb. gruplara ayrılabilir.

Yapılan gruplandırmalarda; grubun birisi incelenirken, diğer grupların elemanlarının özellikleri ile karşılaşılabilir. Bu nedenle, tetikleme işaretlerinin uygulanması şekline göre sayıcıların sınıflandırılmasından başlanarak, yeri geldikçe diğer gruplandırmalar içerisinde bulunan alt gruplardan bahsedilecektir.

11.1. Asenkron Sayıcılar

Bir FF'nin çıkışının onu takip eden FF'nin girişini tetiklemek için kullanıldığı sayıcılar, **'asenkron sayıcılar'** olarak adlandırılır. Asenkron sayıcılar, **'dalgalı sayıcı'** veya **'seri sayıcı'** olarak ta isimlendirilir. Bu tip sayıcılarda FF'ler 'toggle' modunda çalışırlar, yani uygulanan her tetikleme sinyali ile durum değiştirirler.

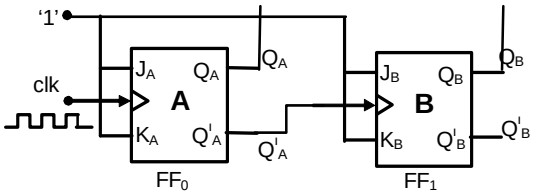
Asenkron sayıcıların önemli özelliklerinden (mahsurlarından) birisi, çalışma hızı yani yayılım gecikmesidir. Çünkü, 5 adet seri bağlı FF'nin kullanıldığı bir sayıcıda herbir FF'nin yayılım gecikmesi 10 nsn ise, devrede bulunan 5. FF'nin konum değiştirmesi için 5x10nsn = 50 nsn'lik bir zamanın geçmesi gerekir.

Asenkron sayıcıları, yukarı ve aşağı asenkron sayıcılar olarak sınıflandırmak mümkündür.

11.1.1. Asenkron Yukarı Sayıcı (İkili Dalgacık Sayıcı)

Asenkron yukarı sayıcı, devredeki ilk FF'den başlayarak, FF çıkışının bir sonraki FF'nin tetikleme girişine dizi şeklinde bağlanmasıyla elde edilir. Devrede bulunan FF'ler 'toggle' modunda çalıştıklarından, tetikleme sinyalinin yükselen kenarı ile ilgili FF'nin durumu 1'den 0'a yada 0'dan 1'e değişir.

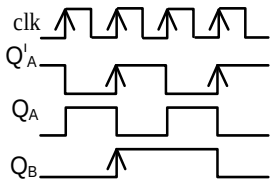
Şekil 11.1.a'da görülen asenkron sayıcıda, en düşük değerlikli FF olan ve gelen tetikleme sinyallerini alan A FF'si, ilk gelen tetikleme sinyalinin yükselen kenarı ile durum değiştirir ve Q_A çıkışı '1' olur. İkinci gelen tetikleme sinyali, A FF'sinin durumunu 1'den 0'a değiştirir. Bu anda A FF'sinin $\overline{Q_A}$ çıkışının bağlı olduğu B FF'si tetiklenir ve Q_B çıkışı '1' değerini alır.



(a) İki bitlik asenkron sayıcı devresi

Clk palsi	Q_B	Q_A
Başlangıç	0	0
1	0	1
2	1	0
3	1	1
4	0	0

(b) İki bitlik sayıcı çıkış değerleri

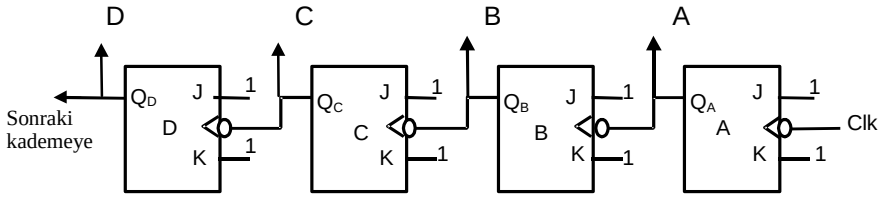


(c) İki bitlik sayıcı FF çıkış dalga şekilleri

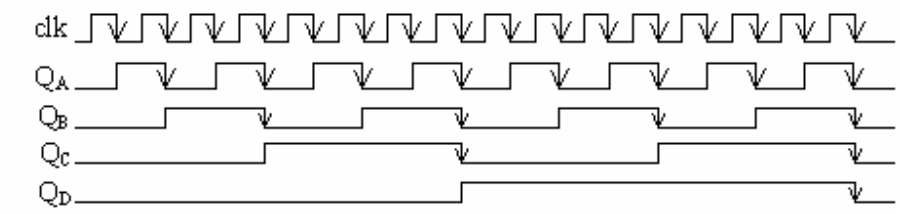
Şekil 11.1. Asenkron ikili sayıcı devresi ve dalga şekilleri.

Tetikleme sinyalleri ile FF'lerde oluşan çıkış değerleri tablo olarak gösterilirse, ikili sayma dizisinin oluştuğu görülür (Şekil 11.1.b). Bu nedenle devre, '**ikili sayıcı**' olarak adlandırılır. Oluşan olayların dalga şekli formunda çizilmesi ile, Şekil 11.1.c'de gözüken çıkış sinyal şekilleri oluşur. Oluşan dalga şekli, FF'lerde tetikleme sinyaline bağlı olarak oluşan çıkış değerleri ile ilgili açıklayıcı bilgi vermektedir. Sayma dizisinde, 4.tetikleme palsı ile çıkışların '00' değerlerini aldığı ve sayma işlemine tekrar başlanacağı görülebilir.

Açıklanan devre, 4 kademeli ikili sayma işlemini gerçekleştirir ve FF'ler pozitif kenar tetiklemeli olduğundan çıkışlardaki değişmeler tetikleme sinyalinin yükselen kenarında gerçekleşir. Negatif kenar tetiklemeli FF'ler kullanılması durumunda, Şekil 11.2.a'daki bağlantı oluşur. Her bir FF'nin Q çıkışının, bir sonraki FF'nin tetikleme sinyalini oluşturduğu Şekil 11.2.a'daki devrede oluşan olayları inceleyelim.



a) Dört bitlik asenkron yukarı sayıcı devresi prensip şeması.



b) Sayıcıda bulunan FF çıkışlarındaki dalga şekilleri.

Şekil 11.2. Dört bit asenkron yukarı sayıcı ve FF çıkışlarındaki dalga şekilleri.

Devrede bulunan FF'ler 'toggle' modunda çalıştıklarından, tetikleme sinyalinin düşen kenarı ile ilgili FF'nin durumu 1'den 0'a yada 0'dan 1'e değişir. En düşük değerlikli FF olan ve gelen tetikleme sinyallerini alan A FF'si, ilk gelen tetikleme sinyalinin düşen kenarı ile durum değiştirir ve Q_A çıkışı '1' olur. İkinci gelen tetikleme sinyalinin düşen kenarı A FF'sinin durumunu 1'den 0'a değiştirir. Bu anda A FF'sinin çıkışının bağlı olduğu B FF'si tetiklenir ve Q_B çıkışı '1' değerini alır.

Her tetikleme sinyali ile durum değiştiren A FF'si, dördüncü sinyalin sonunda B'yi tekrar tetikleyerek Q_B çıkışının 1'den 0'a düşmesine neden olur. Bu değişim C FF'sini tetikleyerek Q_C çıkışının '1' olmasını sağlar. Tetikleme sinyalleriyle FF'lerin çıkışlarında olan değişimler ve herbir FF'nin çıkışında oluşacak dalga şekli, Şekil 11.2.b'de görülmektedir. Şekilde görüldüğü gibi, FF'lerin çıkışı birer birer değişerek, sinyalin sayıcı boyunca bir dalgacık şeklinde yayılmasını sağlar.

Sayıcı devresindeki bir FF'nin giriş ve çıkış sinyallerinin karşılaştırılması durumunda; her bir FF çıkışındaki sinyalin frekansının, girişindeki sinyalin frekansının yarısı olduğu görülür. Bu durum, FF'nin frekans bölücü olarak kullanılabileceği imkanını ortaya çıkarır.

Sayıcıda kullanılan FF çıkışlarındaki dalga şekillerinden görüleceği üzere; sayıcı devresindeki her bir FF'de girişten uygulanan bilginin frekansı ikiye bölünmekte ve 4 kademeli bir sayıcıda girişten uygulanan sinyal 16'ya bölünmüş olarak elde edilmektedir (Şekil 11.2.b).

Açıklanan olaylar genel olarak; '**bir sayıcı devresindeki ilk FF'nin girişine uygulanan sinyalin frekansı, en son FF'nin çıkışında sayıcının sayabileceği maksimum değer kadar bölünmüş olarak elde edilir**' şeklinde özetlenebilir. Bu tanım ile, 40 KHz'lik bir tetikleme palsı uygulanan 4 kademeli bir sayıcı devresinde; 1. FF'nin çıkışında 20 KHz, 2. FF'nin çıkışında 10 KHz, 3. FF'nin çıkışında 5 KHz, 4.FF'nin çıkışında ise 2.5 KHz' lik bir sinyal elde edilir. En son FF çıkışında, tetikleme palsı 16'ya bölünmüş olur.

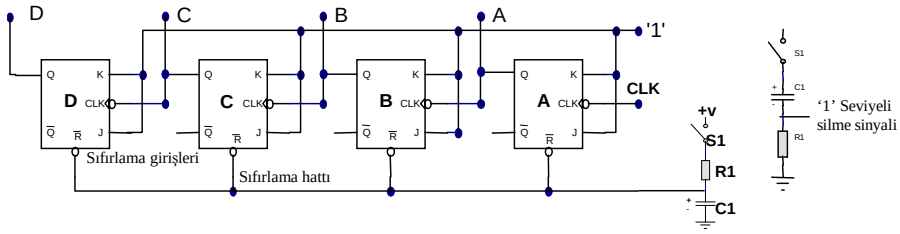
11.1.1.1. Sıfırlamalı ve Önkurmalı Asenkron Yukarı Sayıcılar

Şekil 11.2.a'da görülen devre 0'dan başlamakta ve yukarı doğru saymaktadır. Ancak FF'lerin ve buna bağlı olarak sayıcının başlangıçta sıfır olmaması durumunda, tüm FF'leri '0' konumuna kurmamız gerekir. Bu işlem, FF'lerin asenkron silme (clear-reset) girişinden faydalanarak gerçekleştirilir. Şekil 11.3.a'da sıfırlama devresi eklenmiş asenkron yukarı sayıcı devresi görülmektedir.

Sıfırlama girişine sahip sayıcıdaki S_1, R_1, C_1 elemanlarından oluşan sıfırlama devresi, sıfırlama işlemini gerçekleştirir. Sıfırlama işleminin gerçekleştirilmesi için, devredeki S_1 anahtarı devre çalışma anahtarı ile eşzamanlı bağlanmalıdır. Çalışma anahtarı ve sıfırlama devresindeki S_1 kapatıldığı anda, R_1-C_1 seri devresinden çok büyük bir şarj akımı geçer (çünkü C_1 boştur ve üzerindeki gerilim sıfırdır). Bu şarj akımı nedeniyle, R_1 direncinde büyük bir gerilim düşümü oluşur. R_1'in alt ucundaki '0' seviyeli bu gerilim, sıfırlama sinyali olarak FF'lerin sıfırlama girişlerine uygulanır ve tüm FF'lerin çıkışları sıfıra kurulur. Daha sonra C_1 kondansatörü şarj olur ve kondansatörde oluşan '+V' gerilimi sıfırlama girişlerini etkisiz yapar.

Kullanılan FF'lerin sıfırlama girişlerinin '1' seviyesinde etkili olması durumunda, sıfırlama devresindeki C_1 ile R_1 elemanlarının yer değiştirmesi gerekir (Şekil 11.3.b). Bu yeni devrede,

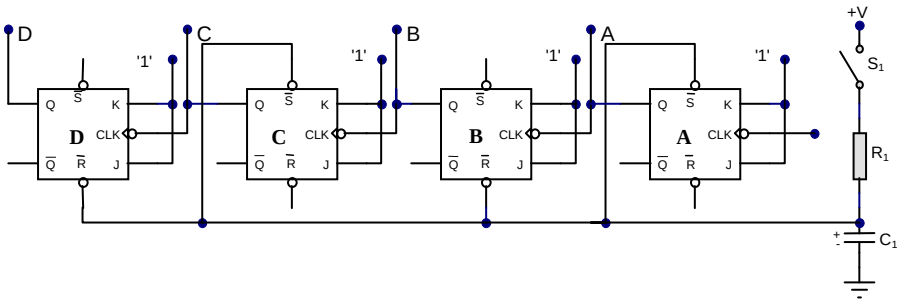
ilk şarj akımı sırasında tüm gerilim R_1 direnci üzerinde düşeceğinden devre sıfırlanır. Daha sonra kondansatörün şarj olmasıyla devreden akan akım sıfırlanacağından sıfırlama girişleri etkisizleşir. Böylece, devrede sayma işlemi başlamadan önce sıfırlama işlemi gerçekleştirilir.



Şekil 11.3. Sıfırlama girişine sahip asenkron yukarı sayıcı ve sıfırlama devresi.

Belirli tip Flip-flop'lar, sıfırlama girişleri (Reset-R) yanısıra önkurma (preset-S) girişlerine de sahiptir. Bu giriş aktif olunca, ilgili FF'nin çıkışı '1' değerine kurulur. Hem önkurma, hem de sıfırlama girişlerine sahip FF'ler kullanılarak, sayma işleminin istediğimiz kademeden (sayıdan) başlaması sağlanabilir. Belirli bir sayıdan başlayarak sayma işlemi yapan devreler, **'önkurmalı sayıcılar'** (preset sayıcılar) olarak adlandırılır.

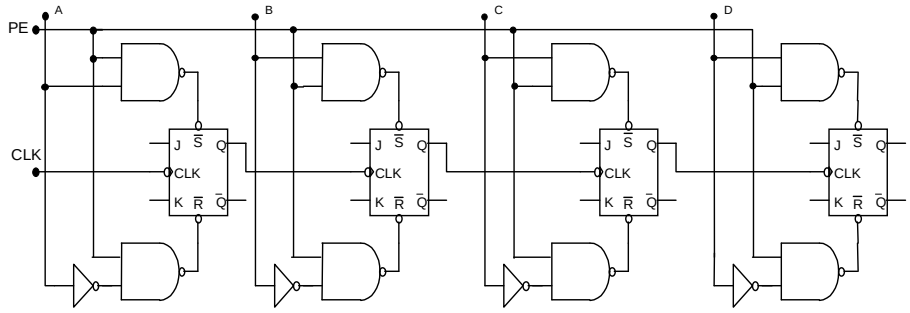
Şekil 11.4'de önkurma ve sıfırlama girişlerine sahip J-K FF'lerle oluşturulan ve '0101' ikili sayısından saymaya başlayan önkurmalı sayıcı devresi görülmektedir. Bu devrede, 1'e kurulması istenen FF'lerin önkurma girişleri ile, 0'a kurulması istenen FF'lerin sıfırlama girişleri birleştirilerek darbe üretici devreye bağlanır. Devreye gerilim uygulanması ile, devredeki FF'ler '0' veya '1' değerlerine kurulur ve sayıcı '0101' değerinden başlayarak yukarı doğru sayar.



Şekil 11.4. Belirli bir kademeden başlayan yukarı sayıcı devresi.

Sayma işleminin başlayacağı değeri isteğe göre ayarlayabileceğimiz devre örneği, Şekil 11.5'de görülmektedir. '**Çok seçenekli önkurmalı sayıcı'** adı verilen bu devrede, sayma

işleminin başlatılmasını istediğimiz değer ikili sayı olarak ABCD girişlerine uygulanır. Örneğin; sayma işlemine $(4)_{10}=(0100)_2$'den başlamak isteniyorsa, girişlere A=B=D=0 ve C=1 değerleri uygulanmalıdır. Devreye PE (Paralel Enable) sinyalinin uygulanması ile bilgiler paralel olarak FF'lere aktarılır. 'Clk' sinyalinin ilk FF'nin tetikleme girişine uygulanması ile, sayıcı kurulan değerden başlayarak yukarı doğru sayma işlemi yapar.



Şekil.11.5. Çok seçenekli önkurmalı sayıcı.

11.1.2. Asenkron Aşağı Sayıcı

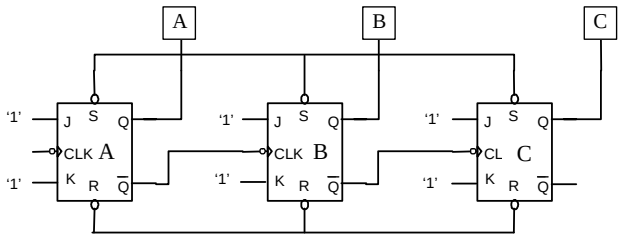
Belirli bir değerden başlayıp 0'a doğru sayma işlemi yapan ve bir FF'nin çıkışının daha sonraki FF'nin tetikleme girişi olarak kullanıldığı devrelere, **'Asenkron aşağı sayıcılar'** denir. Asenkron aşağı sayıcının asenkron yukarı sayıcıdan farkı: FF'ler arasındaki bağlantıda Q çıkışı yerine Q' çıkışlarının kullanılması ve sayma yönüdür. Bu fark dışında, asenkron yukarı sayıcı ve aşağı sayıcının çalışma prensipleri aynıdır.

Üç adet FF ile gerçekleştirilen aşağı sayıcı devresi Şekil 11.6.a'da görülmektedir. Prensip şeması verilen aşağı sayıcı devresinde, Şekil 11.6.b'de verilen sayım dizisi oluşur ve FF çıkışları Şekil 11.6.c'de tabloda verilen değerleri alırlar.

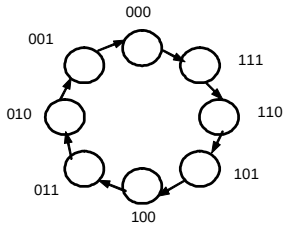
Devrenin çalışmasını incelemek için, başlangıçta tüm FF çıkışlarının '0' olduğu kabul edilir. FF'ler negatif kenar tetiklemeli olduklarından, ilk gelen tetikleme sinyalinin negatif kenarı ile A FF'si durum değiştirir. Q_A çıkışı 1'e giderken, $\overline{Q_A}$ çıkışı 1'den 0'a gider. Bu durumda B FF'sinin tetiklenmesi için uygun şart sağlanır. Bu nedenle B FF'si durum değiştirir ve Q_B çıkışı 0'dan 1'e giderken, $\overline{Q_B}$ çıkışı 1'den 0'a gider. $\overline{Q_B}$ çıkışı C FF'sini tetikler ve Q_C=1 olurken, $\overline{Q_C}$=0 değerini alır. Anlatılanlardan, ilk tetikleme sinyali ile tüm FF'ler için uygun tetikleme işareti oluştuğunu ve C=B=A=1 değerini aldığını bulabiliriz (Şekil 11.6.d).

İkinci gelen tetikleme sinyalinin düşen kenarında, A FF'si tekrar durum değiştirir ve Q_A=0, $\overline{Q_A}$=1 olur. Bu anda uygun tetikleme sinyali oluşmadığından, A dışındaki FF'lerde değişim olmaz. Çünkü, gerekli olan tetikleme sinyalleri oluşmaz. İkinci tetikleme palsı sonucunda, FF'ler A=0, B=C=1 değerlerine sahip olur.

Devam eden tetikleme sinyalleri ile durum değişiklikleri devam eder ve çıkışlarda Şekil 11.6.d'de gösterilen sinyaller oluşur. Çıkış sinyallerinden görüldüğü üzere, 8. pals ile tüm FF'lerin çıkışları '0' olur. Bu durumdan sonra gelen ilk tetikleme sinyali ile, FF'ler başlangıca döner ve geri sayma işlemi tekrar başlar. Aşağı sayıcılar, yukarı sayıcılar kadar yaygın olarak kullanılmazlar. Kullanım yerlerinden birisi, belirli sayıda sayma işlemi sonucunda sayma işleminin sona ermesinin istenildiği sistemlerdir. Bu sistemlerde, sayıcı ön kurma (preset) yardımıyla istenilen sayıya kurulur ve tetikleme sinyalleri ile geriye doğru sayma işlemi başlar. Sayma işleminin '0' değerine eriştiği durum tespit edilir ve sayma işlemi durdurulur. JK-FF'li TTL 7476 entegresi ve ön kurma işlemi yardımıyla, 15-0 arasında istenilen sayıdan geriye doğru sayma işlemi gerçekleştirilebilir.



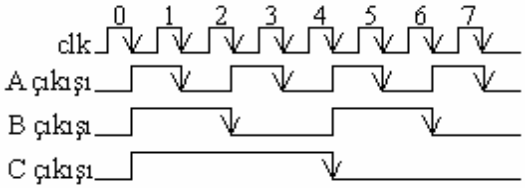
a) Aşagı sayıcı prensip şeması.



b) Aşağı sayıcı dizisi.

Sayı	C	B	A
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

c) Doğruluk tablosu.



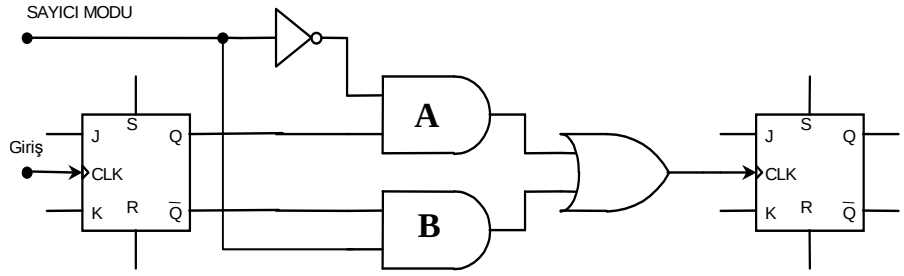
d) Asenkron aşağı sayıcı FF çıkışları dalga şekilleri.

Şekil 11.6. Aşağı sayıcı prensip şeması, sayma dizisi, doğruluk tablosu ve FF çıkış dalga şekilleri.

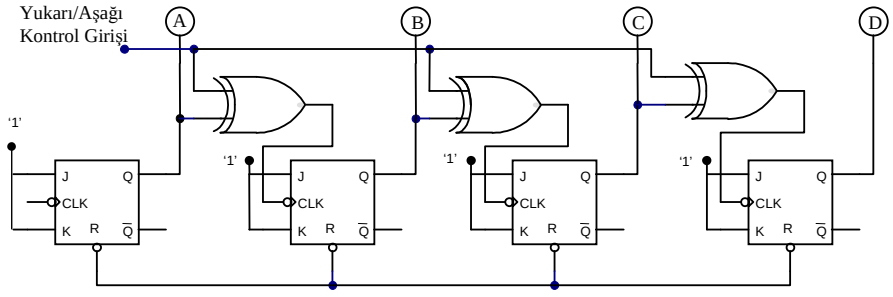
11.1.3. Yukarı - Aşağı Asenkron Sayıcılar

Yukarı sayıcı ve aşağı sayıcı yapısına sahip asenkron sayıcılar, küçük bir değişiklikle hem aşağı hem de yukarı sayıcı yapısında düzenlenebilirler. Yukarı / aşağı sayıcının yapısı, her FF çıkışına konan kontrol devresi haricinde bu kısma kadar anlatılan sayıcıların temel yapısından farklı değildir. Kontrol devresi ile, bir sonraki FF'nin darbe girişine bir önceki FF'nin Q yada Q' çıkışının bağlanması sağlanır.

Şekil 11.7'de görülen aşağı / yukarı sayıcı devresinde; sayma modu (count mode) girişindeki sinyal '1' ise, B kapısı aktif olarak $\overline{Q}$ çıkışının bir sonraki FF'nin tetikleme girişine uygulanmasını ve sayıcının aşağı doğru saymasını sağlar. Sayma modu girişine '0' uygulandığında ise, A kapısı aktif olarak Q çıkışını bir sonraki FF'nin tetikleme girişine uygular ve yukarı doğru sayma işlemi gerçekleşir.



Şekil 11.7. Aşağı / Yukarı Sayıcı Prensip Şeması.



Şekil 11.8. 0-15 arası yukarı / aşağı sayıcı devresi.

Asenkron yukarı / aşağı sayıcı devresinde kullanılan kontrol devresi yerine 'Özel-VEYA' veya 'Özel-VEYADEĞİL' kapısı kullanmak mümkündür. Hatırlanacağı üzere 'Özel-VEYA' kapısı farklılık kapısı, 'Özel-VEYADEĞİL' kapısı ise eşitlik kapısı olarak

çalışmaktadır. ‘Özel-VEYA’ kapısının sayıcı devresine Şekil 11.8’deki gibi bağlanmasıyla sayma işleminin yönü kontrol edilebilir. Kontrol girişindeki bilgi ‘1’ ise, ‘Özel-VEYA’ kapısının Q’dan gelen girişindeki bilginin tersi kapı çıkışında oluşur. Yani kontrol girişi ‘1’ seçilmişse, ‘clk’ girişi Q' değerine eşit olurken, kontrol girişinin ‘0’ seçilmesiyle ‘clk’ girişi Q değerine eşit olur. Kısacası, bir ‘Özel-VEYA’ kapısı sayıcı devresinde yukarı / aşağı sayma kontrolü olarak kullanılırsa; kontrol girişinin ‘0’ yapılmasıyla sayıcı yukarıya doğru, ‘1’ yapılmasıyla sayıcı aşağı doğru sayma işlemi gerçekleştirir.

11.1.4. Asenkron Sayıcılarla Belirli Bir Değere Kadar Sayma İşlemi

Buraya kadar anlatılan sayıcı devreleri ile, FF sayısına bağlı olarak 2^n değerine kadar sayma işlemi gerçekleştirilebilir (n=kullanılan FF sayısı). Diğer bir deyişle; sayıcı devreleri uygulanan tetikleme sinyaline bağlı olarak 2^n değişik durum alabilir, yani 2^n sayma yapabilir. Bir sayıcının bu şekilde tekrar yapmadan sayabildiği sayı miktarına, **‘sayıcının modu’** denir. Örneğin; Mod-8 sayıcı 7’ye, Mod-10 sayıcısı 9’a kadar sayar ve tekrar 0’a döner.

2^n değeri dışında sayma isteniyorsa, sayıcı tasarımında değişiklikler yapılması gerekir. Bu şekildeki bir saymayı gerçekleştirmek için gerekli işlem; sayılması istenen en son sayıyı tespit ederek, bu sayıdan sonra devreyi başlangıç noktasına döndürmektir. Bu işlem; **‘sıfırlama’** olarak isimlendirilir. Sıfırlama işlemi için iki farklı yöntem kullanılır: Dolaylı sıfırlama (modlara göre sıfırlama) yöntemi ve Doğrudan sıfırlama yöntemi.

11.1.4.1. Dolaylı Sıfırlamalı Asenkron Sayıcılar

Sayıcıları belirli bir değerden sonra sıfırlamak için yaygın olarak kullanılan yöntemlerden ilki, dolaylı sıfırlama yöntemidir. FF’lerin aldıkları durumlar ve kapı devreleri kullanılarak, FF’lerin sıfırlama girişleri yardımı ile sayıcı devresindeki sıfırlama işlemi gerçekleştirilmesi, **‘dolaylı sıfırlama’** olarak adlandırılır. **‘Modlara göre sıfırlama yöntemi’** olarak da adlandırılan bu yöntemde, sayma işlemini gerçekleştirecek devrenin tasarımı için aşağıdaki işlem sırası takip edilir.

1. Sayılması istenilen sayıların ikili sayı sistemindeki karşılıkları tablo halinde toplanır.
2. Tablonun en alt kısmına sayılacak en son sayıdan sonraki sayı yazılır.
3. Sayıcıda kullanılacak FF tipi ve adedi tespit edilerek, istenilen sayıcının prensip şeması çizilir.
4. Devreyi sıfırlayacak birleşik devre sayıcı devresine ilave edilerek lojik devre oluşturulur.

Bu işlem sırası, gerçekleştirilmek istenen devrenin özelliğine ve tasarımcının tecrübesine göre azaltılabilir.

Örnek 1: $(5)_{10}=(101)_2$'ye kadar sayma işlemi yapıp, tekrar başa dönen (sıfırlanan) sayıcı devresini tasarlayalım.

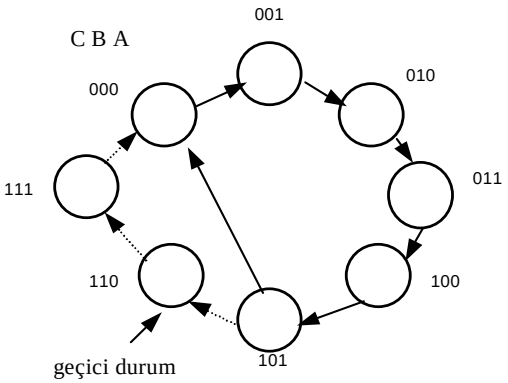
0-5 arasındaki 6 kademeli sayma dizisini sayan sayıcı için gerekli FF sayısı 3'tür. Sayma işlemi 6 aşamalı olduğundan bu devre, **'Mod-6 sayıcı'** devresi olarak adlandırılır. Böyle bir devrenin sayma dizisi ve durum değişim şeması Şekil 11.9.a'daki gibi olur.

Sayıcı devresi 6 kademeli sayma işlemi gerçekleştirdiğinden, 101 durumundan sonra sayma işleminin başlangıca dönmesi (sıfırlanması) gerekir. Sayma işlemindeki en son diziyi takip eden değer, sıfırlama işlemine yardımcı olması amacıyla geçici durum olarak kullanılır. Diğer bir değişle; Mod-6 sayıcıda sıfırlama işlemi, $(110)_2$ ile temsil edilen konum yardımıyla gerçekleştirilir.

Sıfırlama devresi olarak eklenecek birleşik devrenin oluşturulmasında, doğruluk tablosundaki değişkenlerin aldıkları değerlerden ve 'VEDEĞİL' kapısının özelliğinden faydalanılır (Şekil 11.10.a). Sıfırlama yapılması istenen değerde (dizide), 'VEDEĞİL' kapısının çıkışı '0' olacak şekilde bir bağlantı gerçekleştirilir. Çıkışı 1'den 0'a giden 'VEDEĞİL' kapısı, tüm FF'lerin sıfırlama girişlerine lojik '0' uygulayacağından devrenin konumu '000' değerine döner (Şekil 11.10.b). Bir kere sıfırlama işlemi gerçekleştirildikten sonra, 'VEDEĞİL' kapısının çıkışı değişse bile devre normal çalışmasına devam eder.

C	B	A
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0

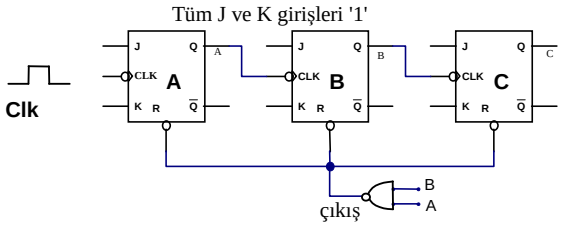
Sayıcıyı sıfırlamak için gerekli geçici durum



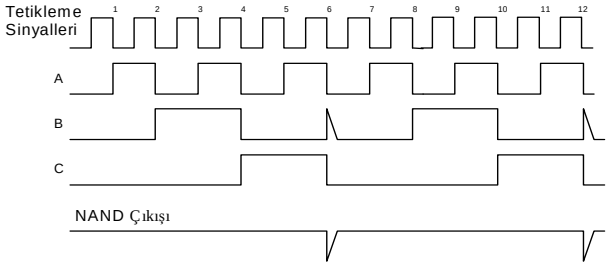
a) Mod-6 sayıcı sayma dizisi. **b)** Mod-6 sayıcı durum şeması.

Şekil 11.9. Mod-6 sayıcı sayma dizisi ve durum şeması.

Şekil 11.10.b'deki sinyal şekillerinden görüleceği üzere devrenin mahsuru; sıfırlama yapılan değerde sıfırlama işleminden önce meydana gelen kısa süreli sinyaldir. Bu sinyalin süresi çok kısadır ve bu nedenle sayma işleminin görüntülendiği göstergelerde fark edilemez. Ancak, kullanılan sayıcı devresinin başka bir lojik devreyi sürmesi durumunda bir probleme neden olabilir.



(a)



(b)

Şekil 11.10. MOD-6 sayıcı devresi ve sinyal şekilleri.

Bu arada not edilmesi gereken diğer bir nokta; MOD-6 sayıcının son FF'sinin çıkışındaki sinyalin frekansının giriş sinyal frekansının, 1/6'sı olduğudur. Bu durumda, MOD-6 sayıcısı girişine uygulanan sinyalin frekansını 6'ya böler diyebiliriz. Böyle bir devrede, en son FF'den elde edilen sinyalin '1' ve '0' değerlerinin çıkışta bulunduğu süre eşit olmayabilir. Örneğin; Şekil 11.10.b'deki sayıcı devresinde C FF'sinin çıkışı 4 tetikleme sinyali süresince '0' kalırken, 2 tetikleme sinyali süresince '1' kalmaktadır.

Örnek 2: 0-9 arasındaki sayıları saymak için, MOD-10 sayıcı tasarımını yapalım.

0-9 arasındaki sayıları sayarak tekrar başa dönen MOD-10'lu sayıcılar, **'onluk sayıcılar'** veya **'BCD sayıcılar'** olarak isimlendirilir. Çünkü bu sayıcılar, sayma sayıları olan 0-9 arasındaki sayıları saymakta ve sayma işlemi sırasında BCD kodlu ikili sayıları kullanmaktadırlar.

BCD kodlu 0000 - 1001 arasındaki 10 farklı konuma sahip sayıları sayma için kullanılacak sayıcı devresinde 4 adet FF kullanılması gerekir. Ancak 4 FF ile gerçekleştirilen devre 16'ya kadar sayar. Bu nedenle, devrenin belirli bir değerde sıfırlanması gerekir. Sayma işleminin bitirilip devrenin başlangıca döneceği değer, $(10)_{10} = (1010)_2$ bilgisidir. Bu bilginin anlamı; D=1, C=0, B=1 ve A=0 olduğu anda devrenin sıfırlanması gerektiğidir.

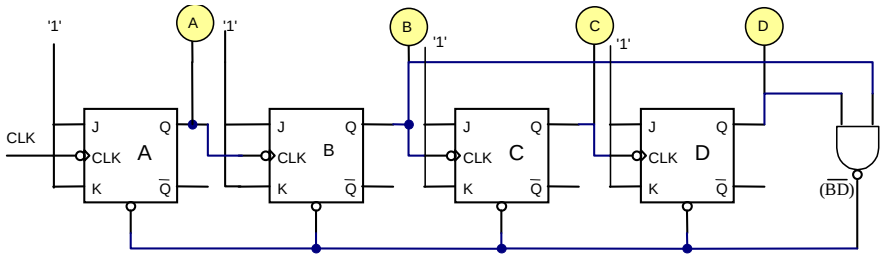
BCD kodunun ifade edildiği veya sayma işleminde FF'lerin çıkışlarının belirtildiği doğruluk tablosu incelenirse; B ve D çıkışlarının ilk kere 1010 bilgisi sırasında aynı anda '1' olduğu

görülür (Şekil 11.11.a). Bu nedenle; B ve D FF'lerinin çıkışlarını kullanan bir 'VEDEĞİL' kapısı ile sıfırlama işlemi gerçekleştirilebilir.

Anlatılanların ışığı altında çizilecek Mod-10 yani BCD sayıcı devresi, Şekil 11.11.b'de gösterilen bağlantıya sahip olur. Bu devrenin tek mahsuru kısa sürede olsa $(10)_{10}$ sayma durumunun göstergede gözükmesidir. Ancak çok kısa bir süre göstergede gözüken bilgi, insan tarafından fark edilemez.

Desimal Sayı	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
10	1	0	1	0

(a)



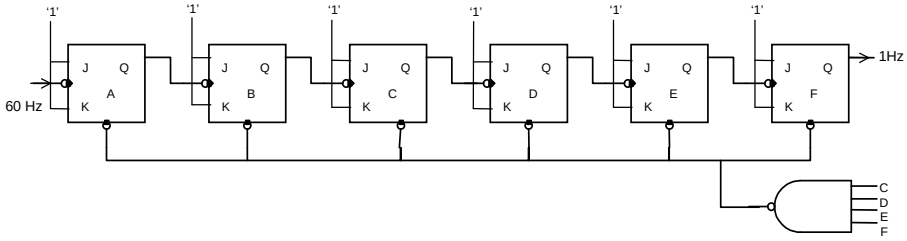
(b)

Şekil 11.11. Dolaylı sıfırlamalı asenkron BCD sayıcı devresi lojik seması.

Mod-10 sayıcısında FF'lerin çıkışında oluşan dalga şekilleri incelenirse, devrenin tetikleme sinyalinin frekansını 10'a böldüğü görülür. Bu nedenle frekansın 10'a bölünmesi istenilen durumlarda Mod-10 sayıcısı kullanılır.

Örnek 3: 60 Hz bir sinyalden, 1 Hz'lik sinyal elde etmemizi sağlayacak devreyi tasarlayalım.

Çözüm: 2^5=32 ve 2^6= 64 olduğundan; 60'a kadar sayma işlemini yapacak ve buna bağlı olarak frekansı 60'a bölecek devrede 6 adet FF kullanılması gerekir. Sayıcı, Şekil 11.12'de görülen yapıya sahip olur ve $(60)_{10}= (111100)_2$ bilgisine ulaştığı anda sıfırlama işlemi gerçekleşir. Bu durumda; C, D, E, F FF'lerinin çıkışları 'VEDEĞİL' kapısına bağlanır. Tüm girişlerinin '1' olduğu anda 'VEDEĞİL' kapısının çıkışı '0' olarak lojik devreyi sıfırlar.



Şekil 11.12. MOD-60 Sayıcı devresi lojik şeması.

11.1.4.2. Doğrudan sıfırlanmalı asenkron sayıcılar

2^n'den farklı sayma işlemi gerçekleştiren sayıcılar oluşturmanın diğer bir yöntemi, doğrudan sıfırlamalı sayıcı tasarlamaktır. Bu yöntemde devredeki sıfırlama işlemi, devrenin çalışma konumundan yararlanılarak gerçekleştirilir. Bu yöntem daha karmaşık ve tasarımı daha uzun olmakla beraber, modlara göre sıfırlama yöntemindeki sakıncayı ortadan kaldırır.

Doğrudan sıfırlamalı sayıcı tasarımında; kullanılacak FF tipi ve sayısı tespit edildikten sonra, FF'lerin çıkışlarının nasıl değiştiğini gösteren bir tablo oluşturulur. Oluşturulan tablo yardımıyla, FF'ler için etkili giriş sinyallerinin nereden alınacağı tespit edilir.

Örnek 4: 3 adet JK tipi negatif kenar tetiklemeli FF kullanarak, doğrudan sıfırlamalı Asenkron MOD-6 sayıcısı gerçekleştirelim.

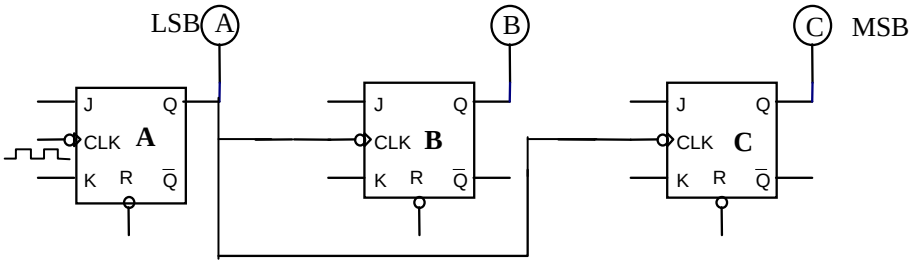
Tasarlanan devrenin doğruluk tablosu Şekil 11.13.a'da görülmektedir. Tablodan görüleceği üzere, her gelen tetikleme sinyali ile A FF'si konum değiştirir. A FF'sinin 1'den 0'a her konum değiştirmesinde B FF'si konum değiştirirken, C FF'si desimal 3'den 4'e giderken ve 5'den sıfıra geçerken durum değiştirmektedir. 3'den 4'e geçerken A ve B FF'lerinin çıkışları 1'den 0'a giderek, C FF'si için uygun tetikleme sinyali oluştururlar. Ancak 5'ten 0'a geçerken yalnızca A FF'si çıkışında 1'den 0'a düşüş olmaktadır. Bu durumda, iki koşulu yerine getiren sadece A FF'si olduğundan, A FF'si çıkışı C FF'sinin tetikleme girişine bağlanmalıdır. Anlatılanlar devre haline getirilirse, Şekil 11.13.c'deki prensip bağlantı oluşur.

Desimal Sayı	C	B	A	Sayılan Darbe
0	0	0	0	1.darbe
1	0	0	1	2.darbe
2	0	1	0	3.darbe
3	0	1	1	4.darbe
4	1	0	0	5.darbe
5	1	0	1	6.darbe
0	0	0	0	

a) Doğruluk tablosu.

Q_n	Q_{n+1}	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

b) J-K FF geçiş tablosu.



c) Mod-6 sayıcı lojik devresi prensip şeması.

Şekil 11.13. Doğrudan sıfırlamalı Mod-6 sayıcı doğruluk tablosu ve prensip lojik devresi.

Mod-6 sayıcı oluşturmak için tetikleme girişleri bağlantısı yapıldı. Ancak J-K girişlerine bağlantı yapılabilmesi için, JK FF geçiş tablosu yardımıyla JK geçişlerinin belirtilmesi gereklidir. JK geçişlerinin belirlenmesi sırasında; ilgili FF girişinde tetikleme sinyali yoksa, J ve K'nın değeri ne olursa olsun FF konum değiştirmez. J ve K'nın durumunun önemli olmadığı bu durumlar, J=d ve K=d olarak kabul edilir. İlgili FF'nin tetikleme girişinde bilgi olması durumunda FF konum değiştireceğinden, J ve K girişlerinin durumları Şekil 11.13.b'de verilen geçiş tablosuna göre belirlenir.

FF'lere ait J ve K girişlerinin değerleri tespit edilirken; FF'nin mevcut ve sonraki durumları kontrol edilir. Eğer iki durum arasında değişiklik var ise, ilgili FF girişinde tetikleme sinyali olması gerektiği açıktır (Şekil 11.13.a'daki oklarla gösterilen durumlar). Değişiklik olan durumlarda, gerekli değişikliği sağlayacak J ve K değerleri geçiş tablosundan faydalanılarak yazılır. Değişiklik olmayan durumlarda J ve K değerleri olarak 'd' atanır. Sayma dizisinde kullanılmayan durumlar; fark etmez olarak kabul edilir ve karnaugh haritasında 'x' ile gösterilir. Sayıcının en son sayma dizisinden sonra '000' değerine döneceği göz önünde bulundurularak geçiş tablosu tamamlanır (Şekil 11.14).

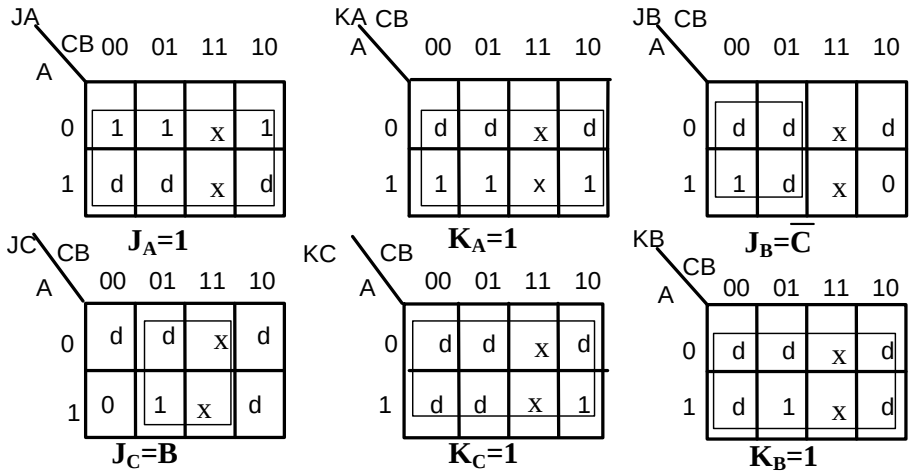
Desimal sayı	C	B	A	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	0	d	d	d	d	1	d
1	0	0	1	0	d	1	d	d	1
2	0	1	0	d	d	d	d	1	d
3	0	1	1	1	d	d	1	d	1
4	1	0	0	d	d	d	d	1	d
5	1	0	1	d	1	0	d	d	1
0	0	0	0						

İlk durum ←

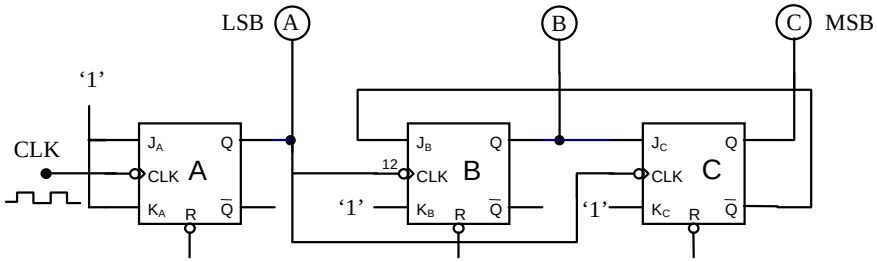
Şekil 11.14. Mod-6 sayıcıya ait JK geçiş tabloları.

Geçiş tablosunda bulunan herbir sütundaki değerler, sütunun ismi ile anılan Karnaugh haritasına taşınır. Karnaugh'da oluşan değerler gruplandırma işlemine tabi tutulur. Oluşan grupları temsil eden lojik eşitlikler yazılır.

Şekil 11.15'de J ve K giriş eşitliklerinin Karnaugh haritaları yardımıyla elde edilmesi görülmektedir. Karnaugh haritaları yardımı ile elde edilen eşitliklerin Şekil 11.13.c'deki prensip devreye uygulanması sonucu, Şekil 11.16'daki lojik devre elde edilir. Eşitliklerin temsil ettikleri anlamların prensip şema üzerine taşınmasında; J_A=1 ifadesi ilgili girişe '1' uygulanacağını gösterirken, J_B=C' ifadesi J_B girişine C' çıkışının bağlanacağını belirtir.



Şekil 11.15. Eşitliklerin Karnaugh haritalarıyla tespiti.



Şekil 11.16. Doğrudan sıfırlamalı asenkron Mod-6 sayıcı.

Örnek 5: Dört adet JK FF kullanarak, doğrudan sıfırlamalı asenkron Mod-10 sayıcı tasarlayalım.

Tasarlanan devrede önce doğruluk tablosu oluşturulup, tablo yardımı ile sayıcıda durum değiştirmeye etki eden sinyaller tespit edilmelidir. Oluşturulan doğruluk tablosu ve tespit edilen durum değişiklikleri, Şekil 11.17.a'da görülmektedir. Durum değişikliklerinde, dikey oklar FF'nin durum değiştirmesine neden olan 1'den 0'a değişimleri göstermektedir. Yatay okların başlangıç noktası sinyalin hangi FF'nin çıkışından alındığını, yatay okların ucu ise sinyalin hangi FF'ye etki ettiğini göstermektedir.

Tablonun incelenmesinden aşağıdaki yorumlar çıkartılır:

A FF'sinin çıkışı her gelen giriş sinyali ile durum değiştirdiğinden, giriş sinyalinin A FF'sinin tetikleme girişine uygulanacağı açıktır. A FF'sinin her 1'den 0'a durum değiştirmesinde B FF'si konum değiştirdiğinden, A çıkışının B FF'sinin tetikleme girişine uygulanması gerekir. Aynı şekilde, B FF'sinin 1'den 0'a durum değiştirmesi C FF'sinde değişikliğe neden olduğundan, B FF'si çıkışı C FF'sinin tetikleme girişine uygulanır.

D FF'sinin konumu, desimal sayının 7'den 8'e ve 9'dan 0'a geçişi sırasında değişmektedir. İlk konum değişikliği sırasında yalnızca A FF'sinin çıkışı değişmektedir. Bu durumda; A FF çıkışının, D FF'sinin tetikleme girişine bağlanmasıyla gerekli şartlar sağlanır.

Sistemde dört FF kullanılacağından, durum geçiş tablosunun oluşturulması ile, Şekil 11.17.b'deki prensip şema üzerindeki gerekli bağlantılar tamamlanabilir.

Devre durum geçiş tablosunun oluşturulması için, Şekil 11.17.a'daki tablo ve JK FF geçiş tablosu yardımıyla, her FF'nin J ve K girişlerine yapılacak bağlantıyı tespit etmek gerekir. FF'lerin **'uyarma fonksiyonları'** olarak bilinen durumların tespiti sırasında; ilgili FF'nin tetikleme girişinde bilgi olmaması halinde J ve K'nın alacağı değerler önemli olmadığından, J=d ve K=d olarak kabul edilir. FF'nin tetikleme girişinde 1'den 0'a değişen bilgi olması durumunda durum değişikliği söz konusu olduğundan, geçiş tablosu yardımıyla J ve K sütunlarının alacağı değerler tespit edilirse; Şekil 11.18'deki tablo oluşur.

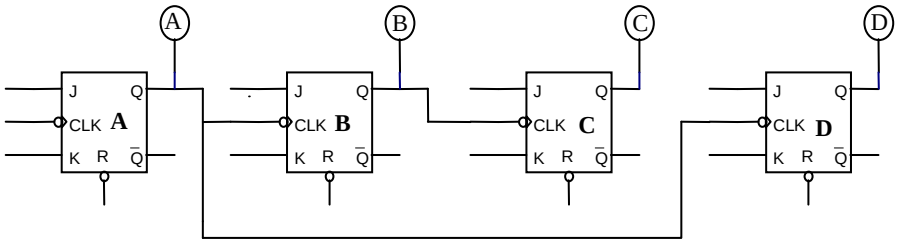
Desimal Sayı	D	C	B	A	Giriş darbeleri
0	0	0	0	0	1.darbe
1	0	0	0	1	2.darbe
2	0	0	1	0	3.darbe
3	0	0	1	1	4.darbe
4	0	1	0	0	5.darbe
5	0	1	0	1	6.darbe
6	0	1	1	0	7.darbe
7	0	1	1	1	8.darbe
8	1	0	0	0	9.darbe
9	1	0	0	1	10.darbe
0	0	0	0	0	

İlk Durum

MSB FF' una etki eden darbeler

LSB FF' una etki eden darbeler

(a)

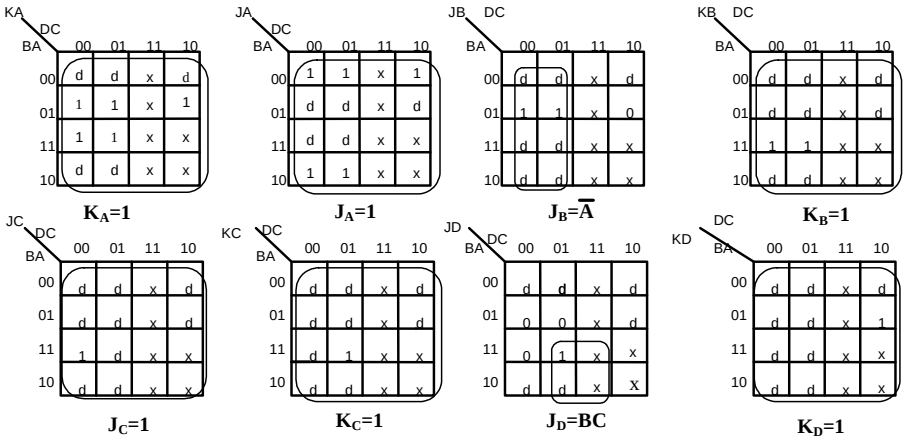


(b)

Şekil 11.17. Mod-10 sayıcı doğruluk tablosu ve prensip şeması.

	D	C	B	A	J_D	K_D	J_C	K_C	J_B	K_B	J_A	K_A	Sayılan Darbe
0	0	0	0	0	d	d	d	d	d	d	1	d	1.darbe
1	0	0	0	1	0	d	d	d	1	d	d	1	2.darbe
2	0	0	1	0	d	d	d	d	d	d	1	d	3.darbe
3	0	0	1	1	0	d	1	d	d	1	d	1	4.darbe
4	0	1	0	0	d	d	d	d	d	d	1	d	5.darbe
5	0	1	0	1	0	d	d	d	1	d	d	1	6.darbe
6	0	1	1	0	d	d	d	d	d	d	1	d	7.darbe
7	0	1	1	1	1	d	d	1	d	1	d	1	8.darbe
8	1	0	0	0	d	d	d	d	d	d	1	d	9.darbe
9	1	0	0	1	d	1	d	d	0	d	d	1	10.darbe
0	0	0	0	0									

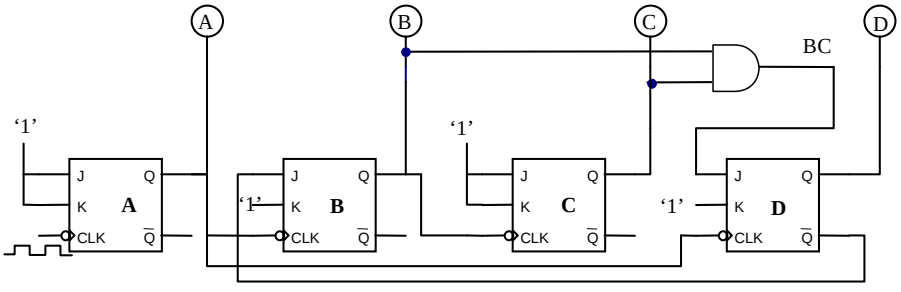
Şekil 11.18. Mod-10 sayıcı için J-K geçişleri tablosu.



Şekil 11.19. J-K eşitliklerinin karnaugh haritalarıyla tespiti.

Tespit edilen J ve K değerlerinin Karnaugh haritalarına taşınması, Karnough'ya taşınan '1' değerlerinin gruplandırılması ve gruplardan eşitliklerin yazılması ile Şekil 11.19'daki J ve K eşitlikleri elde edilir. Karnaugh haritasında; kullanılmayan durumlar (10 ile 15 arası) fark etmez durumu temsilen 'x' ile gösterilir.

Elde edilen J-K eşitliklerine göre Şekil 11.17.b'deki prensip şema tamamlanırsa, Şekil 11.20'deki lojik devre oluşur. Lojik devrenin çizilmesi sırasında; $J_B=A'$ ifadesi; J_B girişine A' çıkışının bağlanacağını gösterirken, $J_D=BC$ ifadesi; B ve C çıkışlarının 'VE' kapısından geçirilerek J_D girişine uygulanacağını belirtir.



Şekil 11.20. Doğrudan sıfırlamalı asenkron MOD-10 sayıcı lojik devresi.

11.2. Senkron Sayıcılar (Eşzamanlı Paralel Sayıcılar)

Asenkron sayıcılarda temel işlem; tetikleme sinyalinin ilk FF'ye uygulanması ve FF'lerdeki konum değişikliğinin bir yayılım gecikmesi sonucunda takip eden FF'lere sırası ile aktarılmasıdır. Bu durumda, FF'lerin konum değiştirmelerinin neden olduğu gecikme (bilgi aktarımı sırasında) sebebiyle bir zaman kaybı olmakta ve sayma hızı azalmaktadır. Bilgi aktarımı sırasında oluşan zaman kaybını azaltmak ve sayma hızını artırmak amacıyla, tetikleme sinyalinin tüm FF girişlerine aynı anda uygulandığı **'senkron sayıcı'** olarak adlandırılan sayıcılar geliştirilmiştir.

Senkron (eş zamanlı) kelimesi, herhangi bir devrede bulunan elemanların ve devrede oluşan olayların birbiri ile zaman ilişkisi içerisinde bulunduğunu belirtir. Senkron kelimesinin sayıcılar ile birlikte kullanılması, sayıcıda bulunan tüm FF'lerin tetikleme girişlerine aynı tetikleme sinyalinin uygulanması anlamına gelir.

Tüm FF'lerin tek bir tetikleme sinyali ile tetiklendiği senkron sayıcılarda, FF'ler kontrol girişlerinin durumlarına bağlı olarak konum değiştirirler.

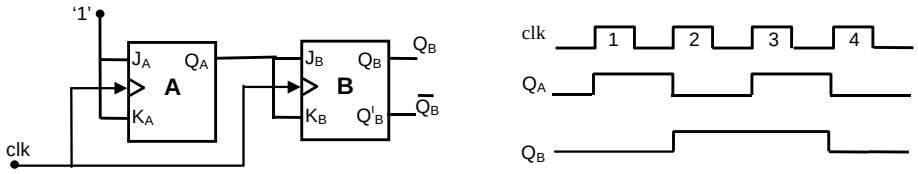
11.2.1. İki ve Üç Bitlik Senkron Yukarı Sayıcılar

Şekil 8.21.a'da, 2 bit ikili sayma dizisini gerçekleştiren senkron sayıcı devresi görülmektedir. Devrenin çalışmasını açıklamak için, başlangıçta sayıcının '00' durumunda, yani her iki FF'nin '0' konumunda bulunduğu kabul edilmelidir.

Şekil 8.21.a'daki senkron sayıcı devresine uygulanan ilk tetikleme sinyali ile 'toggle' modunda bulunan A FF'si tetiklenir ve Q_A çıkışı '1' olur. İlk tetikleme sinyali sırasında JK girişleri '0' olan B FF'si tetikleme sinyalinden etkilenmez.

İkinci tetikleme sinyali ile, hem 'toggle' modundaki A FF'si, hemde Q_A çıkışının '1' olması ile J_B girişinde '1' değerini alan B FF'si durum değiştirir. Bu anda çıkış uçlarında; A=O ve B=1 değerleri oluşur.

Üçüncü tetikleme sinyali ile A FF'si konum değiştirirken, JK girişlerinde '0' değerine sahip olan B FF'si konum değiştirmez. Sonuçta; A=1 ve B=1 değerleri oluşur.



(a) **(b)**

Şekil 11.21. İki bit senkron sayıcı şeması ve çıkış dalga şekilleri.

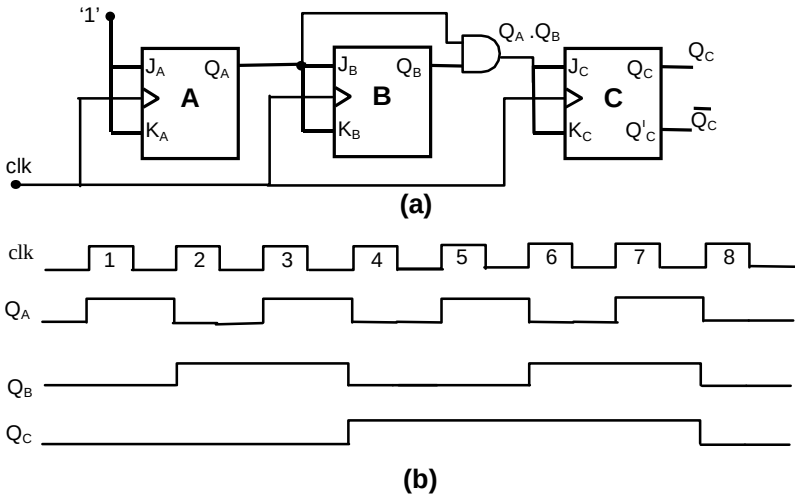
Uygulanan dördüncü tetikleme sinyali ile, JK girişlerinde ‘1’ değerine sahip A ve B FF’leri konum değiştirir. Her iki FF’nin çıkışı ‘0’ değerini alır ve başlangıç değerlerine dönülür.

Anlatılan olayların zamanlara bağımlı olarak dalga şeklinde çizilmesi ile Şekil 8.21.b’deki sinyal şekilleri oluşur. Sinyal şekillerinin çizilmesi sırasında, devrede oluşan yayılım gecikmeleri ihmal edilmiştir.

Üç FF kullanılması ve tetikleme sinyalinin tüm FF’lerin tetikleme girişlerine aynı anda uygulanması ile, üç bitlik senkron sayıcı elde edilir. Üç bitlik senkron sayıcının lojik şeması ve devrede oluşan dalga şekilleri Şekil 8.22’de görülmektedir.

Üç bitlik senkron sayıcıda bulunan FF’lerin durumlarının incelenmesinden; A FF’sinin her gelen tetikleme palsı ile durum değiştirdiği görülür. Bu durumda J_A ve K_A girişleri ‘1’ yapılarak, FF’nin ‘toggle’ modunda çalışması sağlanır.

B FF’si; 2, 4, 6 ve 8. tetikleme sinyalleri uygulandığı anda durum değiştirir ve bu zamanların tümünde A FF’si çıkışı ‘1’ değerine sahiptir. Bu durumda; B FF’sine ait J_B ve K_B girişleri Q_A çıkışına bağlanabilir. Bu bağlantı ile; Q_A=1 iken, gelen tetikleme sinyallerinde B FF’si konum değiştirir. Q_A’nın ‘1’ olmadığı durumlarda ise; tetikleme palslerinin B FF’sine etkisi olmaz.



Şekil 11.22. Üç bitlik senkron sayıcı lojik şeması ve dalga şekilleri.

Üç bitlik sayma işlemini özetleyen doğruluk tablosunun veya dalga şekillerinin incelenmesinden; C FF’sinin durum değiştirdiği anlarda Q_A ve Q_B değerlerinin ‘1’ olduğu bulunur. Bu durumda; Q_A ve Q_B çıkışlarının ‘VE’ kapısına uygulanması ve kapı çıkışının J_C-

K_C girişlerine bağlanması ile gerekli şartlar sağlanır. Q_A ve Q_B çıkışlarının '1' olduğu anlarda, J_C-K_C girişlerine '1' uygulanır. Bu anlarda gelen tetikleme sinyalleri ile C FF'si konum değiştirirken, diğer zamanlarda uygulanan tetikleme sinyalinin bir etkisi olmaz.

11.2.2. Dört Bitlik ve BCD Senkron Yukarı Sayıcılar

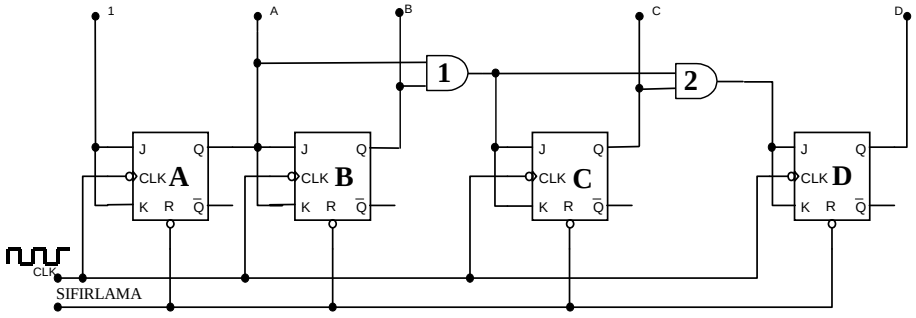
Şekil 11.23'de negatif kenar tetiklemeli FF'ler ile oluşturulan, dört bitlik senkron yukarı sayıcı lojik şeması görülmektedir. Şekilden görüleceği üzere; A, B ve C FF'lerinin bağlantılarının, üç bitlik senkron sayıcıdan farkı yoktur.

Devrenin çalışmasını açıklamak için, ilk anda tüm FF çıkışlarının '0' olduğu kabul edilir. İlk tetikleme sinyali ile, A FF'si konum değiştirerek '1' değerini alır ve B FF'sinin J-K girişlerini '1' yapar. Ancak bu konum değişikliği FF'nin yayılım gecikmesine bağlıdır. Örneğin; yayılım gecikmesi 20nsn ise, B FF'sinin JK girişlerindeki değer değişimi 20nsn sonra etkili olur. Yayılım gecikmesinin sonucu olarak, ilk tetikleme sinyali ile yalnızca A FF'si konum değiştirir, diğer FF'lerde bir değişiklik olmaz.

İkinci tetikleme sinyali ile, 'toggle' modu çalışma şartı sağlanan A ve B FF'leri etkilenir. Bu sinyal ile A FF'sinin çıkışı 1'den 0'a giderken, B FF'sinin çıkışı 0'dan 1'e değişir. Bu anda C ve D FF'lerinin J-K girişleri '0' olduğundan konum değiştirmezler. İkinci tetikleme sinyali sonucunda çıkışlardaki değerler; D=0, C=0, B=1, A=0 olduğundan, çıkış $(2)_{10}$ değerine eşittir.

Üçüncü tetikleme sinyali ile yalnızca 'toggle' modunda çalışan A FF'i durum değiştirir. Bu durum değişikliği ile D=0, C=0, B=1 ve A=1 değerleri oluşur. Bu değer $(3)_{10}$ sayısını temsil eder.

Dördüncü tetikleme sinyali ile, 'toggle' modunda bulunan B ve C FF'leri ile A FF'si birlikte konum değiştirdiklerinden, çıkışlardaki bilgiler; A=0, B=0, C=1 ve D=0 şeklini alır. Bu sinyal ile FF'lerin çıkışlarındaki değer, $(4)_{10}$ sayısına eşitlenir.



Şekil 11.23. Dört bitlik senkron sayıcı lojik şeması (Mod-16).

Devrede bulunan FF'lerin gelen sinyaller ile durumlarını değiştirmesi kontrol edilirse; devrede bulunan D FF'si, sayma dizisi sırasında yalnızca A=1, B=1 ve C=1 olduğu durumlarda 'toggle' moduna sahip olarak konum değiştirir. Bu şartların sağlandığı durumlar; ikili '0111' ve '1111' değerleridir.

Gelen tetikleme sinyalleri ile çalışmaya devam eden dört bitlik sayıcı, 16. sinyal sonucunda $(0000)_2$ başlangıç değerini alarak sayma işlemine tekrar başlar. Sayma sonucunda oluşan doğruluk tablosunun ve sinyal şekillerinin asenkron sayıcılarda elde edilenlerden hiçbir farkı yoktur.

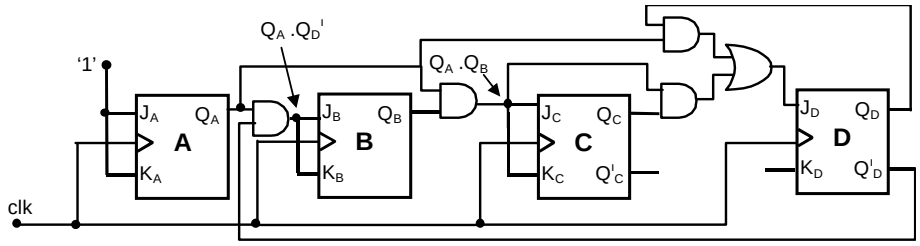
Dört bitlik senkron sayıcı devresinin bağlantılarındaki değişiklik ile oluşturulan BCD sayıcı devresi, '0000' ile '1001' durumları arasındaki sayma dizisini gerçekleştirir (Şekil 11.24). Sayma dizisi sırasında sayıcı '1001' değerini sayacağı anda '0000' başlangıç konumuna dönerek, tekrar saymaya başlar.

Sayma dizisini gerçekleştirecek devreyi oluşturmak için; FF'lerin durum değiştirdikleri anlarda sahip olunan çıkış değerleri incelenir. Şekil 11.24'deki sayma dizisinin incelenmesinden; A FF'sinin her tetikleme sinyali ile durum değiştirdiği ve bu nedenle A FF'sinin 'toggle' modunda çalışması gerektiği bulunur. A FF'sine ait J_A ve K_A girişlerinin, doğrudan '1' değerine bağlanması ile istenilen gereklilik oluşturulur.

Tetikleme Sinyali	**Q_D**	**Q_C**	**Q_B**	**Q_A**
Başlangıç	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10-Başlangıç	0	0	0	0

Şekil 11.24. BCD Senkron sayıcı sayma dizisi.

Q_B'ye ait durum değiştirme sütununun incelenmesinden; Q_A=1 ve Q_D=0 iken, gelen tetikleme sinyali ile B FF'sinin durum değiştirdiği bulunur. Bulunan durum değişiklikleri; $J_B=K_B=Q_A.\overline{Q}_D$ eşitliğini sağlayan lojik devre çıkışının J_B ve K_B girişlerine uygulanması ile sağlanabilir. Gerekli lojik devre, Q_A ve Q_D çıkışlarının 'VE' kapısına uygulanması ile oluşturulabilir (Şekil 11.25).



Şekil 11.25. BCD senkron yukarı sayıcı lojik şeması.

Sayma dizisi tablosundan, C FF'sindeki durum değişimlerinin Q_A=1 ve Q_B=1 durumlarından sonra oluştuğu görülür. Bu durumda J_C ve K_C girişleri için gerekli eşitlik; $J_C=K_C=Q_A.Q_B$ şeklinde yazılabilir. Gerekli eşitlik; Q_A ve Q_B çıkışlarının 'VE' kapısından geçirilerek ilgili girişlere uygulanması şeklinde gerçekleştirilebilir.

D FF'sine ait dizinin incelenmesinden; D FF'sinin iki kere durum değiştirdiği bulunur. İlk durum değişikliği sırasında FF'lerin durumları Q_A=1, Q_B=1 ve Q_C=1 iken, ikinci değişim sırasında Q_A=1 ve Q_D=1 olduğu görülür. Bu durumda J_D ve K_D girişleri için gerekli eşitlik;

$$J_D=K_D=Q_A.Q_B.Q_C+Q_A.\overline{Q_D}$$

şeklinde yazılabilir. Eşitliği sağlayacak lojik devrenin 'VE-VEYA' kapıları yardımı ile oluşturulup, J_D ve K_D girişlerine uygulanmasıyla bağlantı tamamlanır.

Tüm açıklamaların lojik devrede oluşturulması ile, Şekil 11.25'deki devre oluşur.

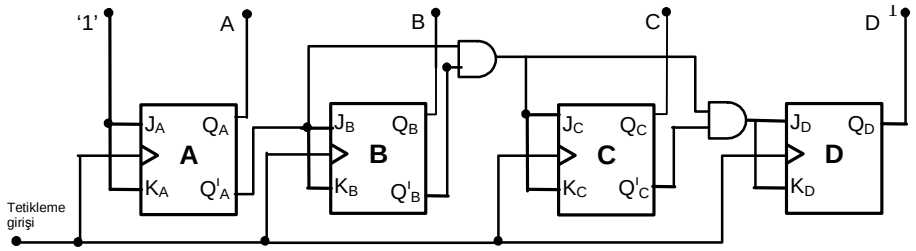
Asenkron sayıcılara göre daha fazla devre elemanına ihtiyaç duyan senkron sayıcılar, işlem hızı olarak daha üstün durumdadır. Piyasada çok sayıda TTL ve CMOS asenkron sayıcı entegresi bulunmaktadır. 74LS160/162 (senkron onluk sayıcı) ve 74HC101/163 (senkron MOD-16 sayıcı) entegreleri bunlardan ikisidir.

11.2.3. Senkron Aşağı Sayıcı

Yukarı sayan asenkron sayıcıları aşağı sayıcı yapmak için; FF'ler arasındaki bağlantının (bir sonraki tetikleme girişine) Q çıkışı yerini Q′ çıkışından alınarak gerçekleştirildiğini görmüştük. Senkron sayıcılarda aynı mantık, JK girişlerini sürmek için, Q′ tümleyen çıkışının kullanılması şeklinde uygulanır (Şekil 11.26).

Senkron yukarı sayıcıdan tek farkı; 'VE' kapılarına uygulanan sinyallerin Q′ tümleyen çıkışından alınması olan senkron aşağı sayıcıda, her gelen tetikleme sinyali ile en düşük

sıralı konumdaki FF konum değiştirir. Çıkışı düşük basamak değerine sahip FF'lerin konumunun '0' olması durumunda, çıkışı '1' olan ilk FF durum değiştirir. Örneğin; çıkışları DCBA=1100 olan bir aşağı sayıcıda, gelen tetikleme sinyali ile çıkış DCBA=1011 değerini alır. Bu örnekte A konum değiştirir, çünkü her gelen sinyal ile A konum değiştirir. B'nin konum değiştirmesinin nedeni, A'nın mevcut durumunun '0' (A′=1) olmasıdır. C konum değiştirir çünkü BA=00'dır (BA=1). D FF'si konum değiştirmez, çünkü alt sıralı FF'lerin değeri 100'dır. Bu değerlerin uygulandığı 'VE kapısı' çıkışı '0' olacağından, D FF'si tetikleme sinyalinden etkilenmez. Senkron aşağı sayıcıda yeni FF çıkışlarına göre oluşacak olaylar incelenirse, aşağı sayma dizisi elde edilir.



Şekil 11.26. Dört bitlik senkron aşağı sayıcı lojik şeması.

11.2.4. Senkron Yukarı - Aşağı Sayıcı

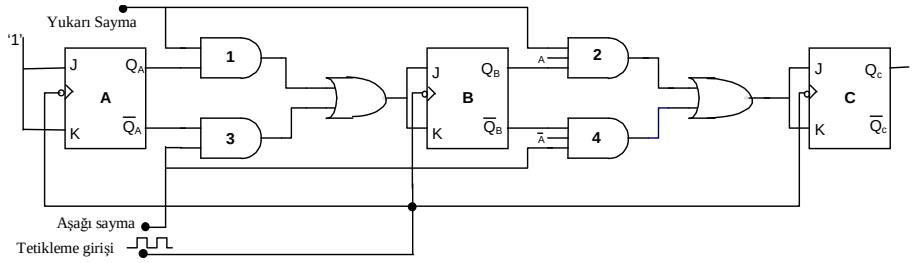
İleri ve geri sayma işlemi yapabilen senkron sayıcılar, Şekil 11.27'deki gibi tek bir devrede birleştirilebilir. Kontrol girişleri, FF'lerin çıkışının Q veya Q' çıkışından alınacağına karar vererek sayma yönünü (ileri sayma-geri sayma) belirler. Şekil 11.27'de görülen üç bitlik senkron sayıcı, yukarı sayma girişi '1' olduğu zaman 000'dan başlayarak 111'e doğru yukarı sayarken, aşağı sayma girişinin '1' olması durumunda 111 değerinden başlayarak 000'a doğru sayar.

Şekil 11.27'de görülen devrede, yukarı sayma=1 ve aşağı sayma=0 durumunda; 1 ve 2 nolu 'VE' kapıları çıkışları lojik '1' olurken, 3 ve 4 nolu 'VE' kapılarının çıkışları lojik '0' olur. Bu durumda, A ve B çıkışlarını takip eden FF'lerin J-K girişine Q_A ve Q_B değerleri aktarılır. Bu çıkışların herhangi birisinin '1' olması, çıkışın bağlı olduğu FF girişine '1' uygulanmasını sağlayacağından, uygulanan ilk tetikleme sinyali ile yukarı sayma işlemi gerçekleşir.

Yukarı sayma = 0 ve aşağı sayma =1 olduğu durumda ise; Q′ çıkışlarına bağlı olan kapı devreleri aktif olur. Bu durumda, Q'_A ve Q'_B çıkışları daha sonraki FF'lere ait J-K girişlerine aktarılacağından aşağı sayma işlemi gerçekleşir.

Yukarı ve aşağı sayma girişlerinin bir adet ‘DEĞİL’ kapısı ile birleştirilerek tek giriş şeklinde kullanılması mümkündür.

74LS190 ve 74HC150 entegreleri, piyasada çok sayıda bulunan aşağı-yukarı senkron sayıcılardan iki tanesidir. Her iki sayıcıda MOD-10 sayıcıdır.



Şekil 11.27. Senkron yukarı / aşağı sayıcı (Mod-8).

11.2.5. Senkron Sayıcıların Tasarımı

Senkron sayıcıların tasarımında, genel olarak doğrudan sıfırlamalı asenkron sayıcılarda açıklanan işlem sırası takip edilir. İki tasarım arasında tek fark; senkron sayıcılarda tetikleme girişlerine uygulanan sinyallerin tüm FF’ların tetikleme girişine aynı anda uygulanması nedeniyle, ‘doğrudan sıfırlamalı sayıcı’ tasarımında yapılan etkili darbe tespiti işlemine gerek olmamasıdır. Bu fark dışında asenkron ve senkron sayıcı tasarımındaki işlem sırası aynıdır.

Tasarımdaki işlem sırası aşağıdaki gibi özetlenebilir:

i- Tasarlanmak istenen sayıcıda kullanılacak FF tipi ve sayısı belirlenir.

ii- Sayıcının yukarı mı yoksa aşağı mı sayacağına karar verilerek, sayılmak istenen dizi bir tabloda sıralanır (Şekil 11.28.a). Sayıcı çıkışındaki değişimleri gösteren bu tabloya ‘doğruluk tablosu’ diyebiliriz. Çünkü sayıcının tüm fonksiyonlarını ve tetikleme sinyallerinde oluşan olayları tablodan görebiliriz.

Doğruluk tablosunda; her FF’nin alacağı bir sonraki durum, ilgili sayının karşısına (Q_{t+1}) şeklinde yeni sütun olarak belirlenebilir (Şekil 11.28.b).

iii- Doğruluk tablosu ve FF geçiş tablosu yardımıyla, her FF için giriş-geçiş değerleri (örneğin JK girişleri) tespit edilir.

iv- Her bir FF girişi için Karnaugh haritası hazırlanarak, elde edilen geçiş değerleri Karnaugh haritalarına taşınır. Bütün değerler Karnaugh haritasına taşındıktan sonra, boş kalan kutular ‘farketmez’ olarak işaretlenir.

v- Oluşan Karnaugh haritasında grublandırmalar yapılarak lojik eşitlikler yazılır.

vi- Basitleştirilmiş eşitliklerden senkron sayıcı lojik devresi çizilir.

Senkron sayıcıların takip ettikleri sayma dizisinin belirli bir sıra dahilinde olması şart değildir. Yani, senkron sayıcıları istediğimiz sayıları sayacak şekilde tasarlayabiliriz. Bu işlem için, saymak istediğimiz sayı dizisini doğruluk tablosunda belirtip, geçiş tablosunu sayılacak diziye göre oluşturmamız yeterlidir.

Özetlenen işlem basamaklarını örnek tasarımlar ile açıklayalım.

Örnek 6 : 0-1-2 sayı dizisini periyodik olarak sayan senkron sayıcıyı, tasarım işlem basmaklarına uyarak tasarlayalım.

i- Sayıcıda kullanılacak FF türü belirlenir: JK FF kullanılacak.

ii- Tasarlanan sayıcı, aynı sayma dizisini tekrarlamakta ve 2'den sonra '0' değerine dönmektedir. Bu durumda Şekil 11.28.a'daki sayma dizisi oluşur.

Tetikleme Sinyali	Desimal Değer	İkili Değer
1	0	0 0
2	1	0 1
3	2	1 0
4	0	0 0

a) Sayıcı sayma dizisi

Mevcut Durum Q_t		Sonraki Durum Q_{t+1}		Giriş-Geçiş Değerleri			
B	**A**	**B**	**A**	J_B	K_B	J_A	K_A
0	0	0	1	0	d	1	d
0	1	1	0	1	d	d	1
1	0	0	0	d	1	0	d

b) Sayıcıda giriş-geçiş değerlerinin oluşturulması

Şekil 11.28. Senkron sayıcı tasarım aşamaları.

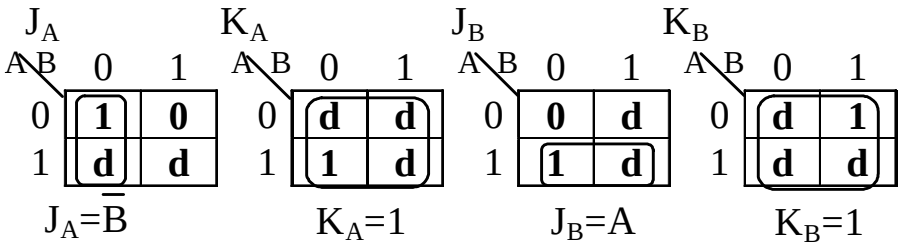
Sayıcının belirtilen diziyi sayabilmesi için gerekli FF sayısı tespit edilir: FF sayısını bulmanın en kolay yolu, sayma dizisindeki en büyük sayının kaç adet FF ile yazılabileceğini bulmaktır. Yapılan tasarımda en büyük sayı '2' olduğundan ve '10' şeklinde gösterilebildiğinden, FF sayısı 2 olarak bulunur.

iii- Mevcut durum ve sonraki durum değerlerinin sıralandığı doğruluk tablosuna JK FF geçiş tablosundan faydalanarak giriş-geçiş değerleri eklenir (Şekil 11.28.b).

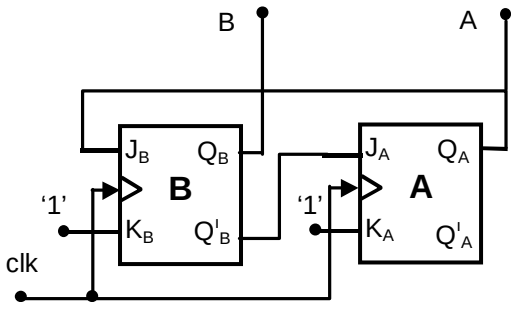
iv- Eklenen giriş-geçiş değerlerinin oluşturduğu fonksiyonları sadeleştirmek için Karnaugh şemaları hazırlanır. Oluşturulan tablodaki değerler, ilgili Karnaugh haritalarına taşınır (Şekil 11.29). Sayma dizisinde bulunmayan değerleri temsil eden hücrelere 'farketmez-d' değeri yazılır.

v- Karnaugh haritalarında gruplandırma yapılarak, lojik eşitlikler elde edilir (Şekil 11.29).

vı- Elde edilen eşitliklere ait bağlantıları içerecek lojik devre çizilir. Kullanılacak FF sayısı iki olduğundan, bu FF'ler çizilerek gerekli bağlantılar oluşturulur (Şekil 11.30).



Şekil 11.29. Senkron sayıcı tasarımında lojik eşitliklerin yazılması.



Şekil 11.30. 0, 1, 2 Sayma dizisini periyodik sayan senkron sayıcı lojik şeması.

Örnek 7: Mod–7 Senkron yukarı sayıcı lojik devresini tasarlayalım.

Tasarımda JK FF'ler kullanacağız. Mod–7 sayıcısında, 0'dan 6'ya kadar sayacak devre için üç adet FF kullanmamız gerekir. Sayıcının takip edeceği sayma dizisinin yer aldığı doğruluk

tablosu (sayma dizisi) ile, J–K FF geçiş tablosu yardımıyla belirlenen tüm FF'lere ait geçiş tablosu değerleri Şekil 11.31'deki gibi oluşur.

Tetikleme Sinyali	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	d	0	d	1	d
1	0	0	1	0	d	1	d	d	1
2	0	1	0	0	d	d	0	1	d
3	0	1	1	1	d	d	1	d	1
4	1	0	0	d	0	0	d	1	d
5	1	0	1	d	0	1	d	d	1
6	1	1	0	d	1	d	1	0	0
0	0	0	0						

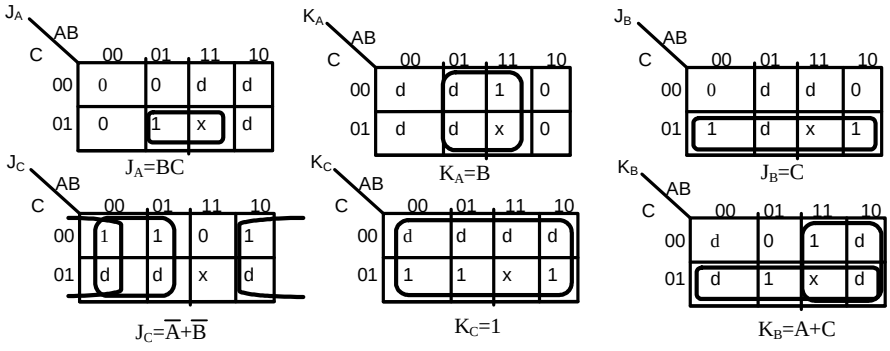
Q_n	Q_{n+1}	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

JK Geçiş Tablosu

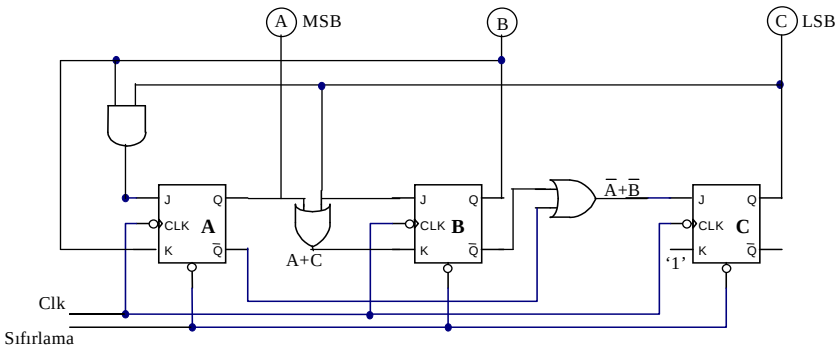
Şekil 11.31. MOD-7 senkron sayıcı doğruluk tablosu ve JK FF geçiş tablosu.

Doğruluk tablosundaki J ve K girişlerine ait değerlerin Karnaugh haritalarına taşınması ile, Şekil 11.32'de görülen Karnaugh şemalarındaki gruplar elde edilir. Grupların temsil ettiği lojik eşitliklerin yazılması ile, tasarlanan devrede oluşacak bağlantılar bulunur.

Elde edilen eşitliklerden faydalanılarak tasarlanan devrede gerekli bağlantılar oluşturulursa, Şekil 11.33'deki Mod-7 senkron sayıcı lojik şeması elde edilir.



Şekil 11.32. MOD-7 senkron sayıcı için J-K giriş eşitliklerinin yazılması.



Şekil 11.33. MOD-7 senkron sayıcı lojik devresi.

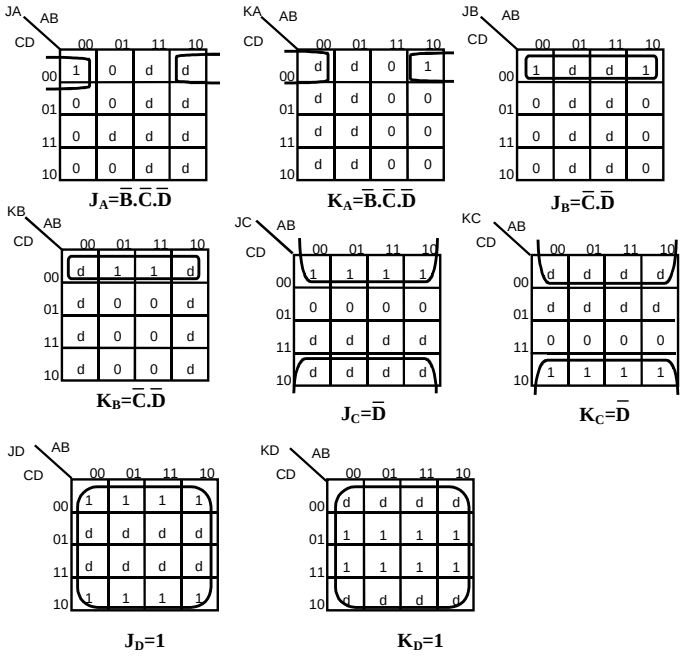
Örnek 8: MOD-16 senkron aşağı sayıcıyı JK FF kullanarak tasarlayalım.

Sayıcı devresi 16 basamaklı sayma dizisi sayacağından sayıcı yapısında, 4 adet J-K FF kullanılacaktır. Kullanılacak FF'lere ait sayma dizisi ve J-K girişleri için geçiş değerleri Şekil 11.34'de tablo halinde görülmektedir. Devrede D FF'si LSB'yi (En düşük değerlikli biti) temsil etmektedir.

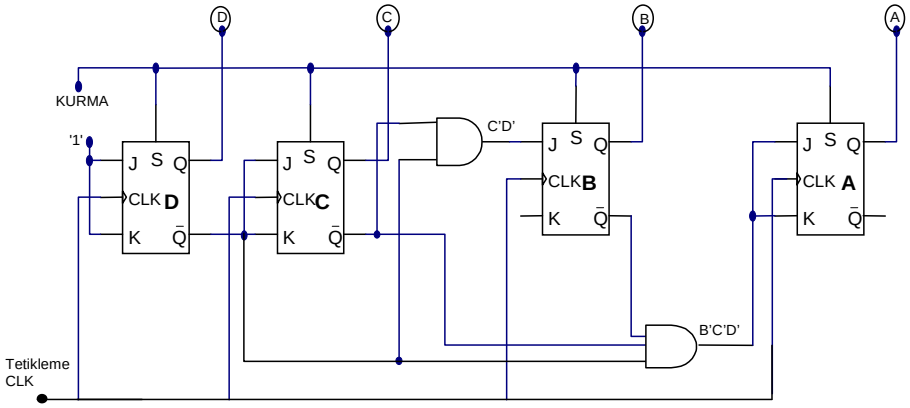
Tetikleme Sinyali	A	B	C	D	J_A	K_A	J_B	K_B	J_C	K_C	J_D	K_D	
0	1	1	1	1	d	0	d	0	d	0	d	1	15
1	1	1	1	0	d	0	d	0	d	1	1	d	14
2	1	1	0	1	d	0	d	0	0	d	d	1	13
3	1	1	0	0	d	0	d	1	1	d	1	d	12
4	1	0	1	1	d	0	0	d	d	0	d	1	11
5	1	0	1	0	d	0	0	d	d	1	1	d	10
6	1	0	0	1	d	0	0	d	0	d	d	1	9
7	1	0	0	0	d	1	1	d	1	d	1	d	8
8	0	1	1	1	0	d	d	0	d	0	d	1	7
9	0	1	1	0	0	d	d	0	d	1	1	d	6
10	0	1	0	1	0	d	d	0	0	d	d	1	5
11	0	1	0	0	0	d	d	1	1	d	1	d	4
12	0	0	1	1	0	d	0	d	d	0	d	1	3
13	0	0	1	0	0	d	0	d	d	1	1	d	2
14	0	0	0	1	0	d	0	d	0	d	d	1	1
15	0	0	0	0	1	d	1	d	1	d	1	d	0
0	1	1	1	1									

Şekil 11.34. MOD-16 senkron aşağı sayıcısının doğruluk tablosu ve J-K geçişlerinin tespiti.

Şekil 11.34'deki J-K geçiş değerleri karnaugh haritalarına taşınarak, Karnaugh haritalarında oluşan değerlere göre gruplandırma yapılır. Oluşturulan gruplardan Şekil 11.35'deki eşitlikler elde edilir.



Şekil 11.35. MOD–16 senkron aşağı sayıcının JK eşitliklerinin tespiti.



Şekil 11.36. Mod-16 senkron aşağı sayıcı lojik şeması.

Elde edilen eşitlikler temsil eden bağlantılar yapılırsa, Şekil 11.36'daki Mod-16 senkron aşağı sayıcı devresi oluşur.

Devreye çalışma gerilimi uygulandıktan sonra, kurma girişi ile tüm FF'ler '1111' konumuna kurulmalı ve daha sonra 'clk' işareti uygulanmalıdır. Bu işlemlerden sonra sayıcı $(1111)_2$'den $(0000)_2$'ye doğru saymaya başlar. Devrenin yukarı sayıcıdan farkı, Q′ çıkışlarının kullanılmasıdır.

Örnek 9: Şekil 11.37'de verilen doğruluk tablosundaki sıraya göre (1, 2, 4) sayma işlemini yapan ve bu işlemi tekrarlayan devreyi tasarlayalım.

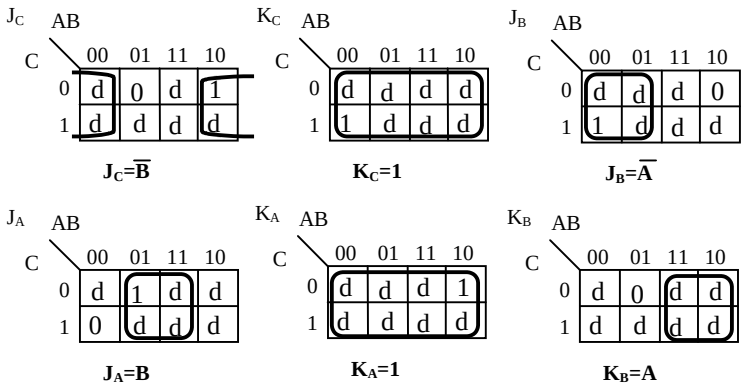
İlk işlem olarak; doğruluk tablosunda verilen sayma dizisi ve JK FF geçiş tablosundan faydalanarak, FF'lere ait geçiş değerleri tespit edilir (Şekil 11.37).

Tespit edilen FF geçiş değerlerin karnaugh haritalarına taşınmasıyla, lojik devrenin oluşmasını sağlayacak eşitlikler elde edilir (Şekil 11.38).

Önceki durum			Sonraki durum			FF durumları					
C	**B**	**A**	**C**	**B**	**A**	$\mathbf{J_C}$	$\mathbf{K_C}$	$\mathbf{J_B}$	$\mathbf{K_B}$	$\mathbf{J_A}$	$\mathbf{K_A}$
1	0	0	0	1	0	d	1	1	d	0	d
0	1	0	0	0	1	0	d	d	0	1	d
0	0	1	1	0	0	1	d	0	d	d	1

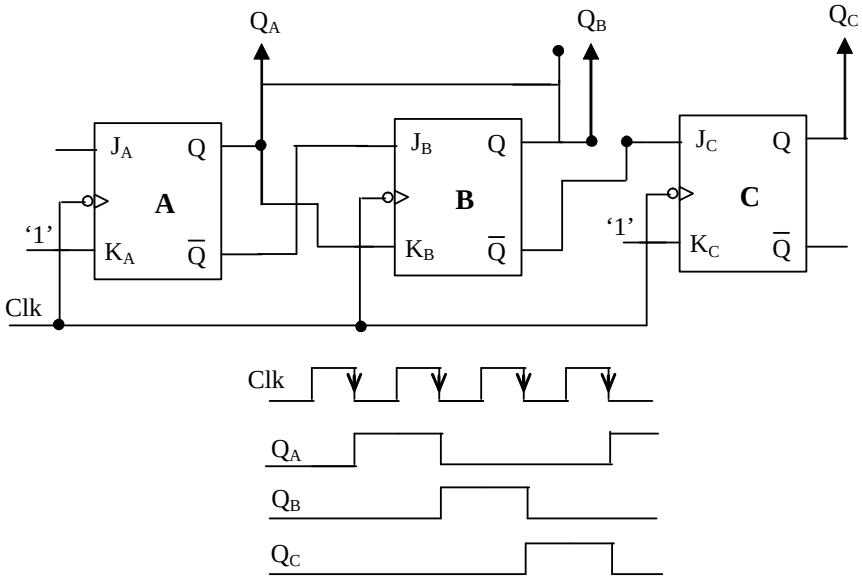
Qn	$\mathbf{Q_{n+1}}$	**J**	**K**
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

Şekil 11.37. JK geçiş tablosu yardımı ile sayıcı sayma dizisinin oluşturulması.



Şekil 11.38. 1, 2, 4 sayma dizisini sayacak senkron sayıcı devresine ait eşitliklerin yazılması.

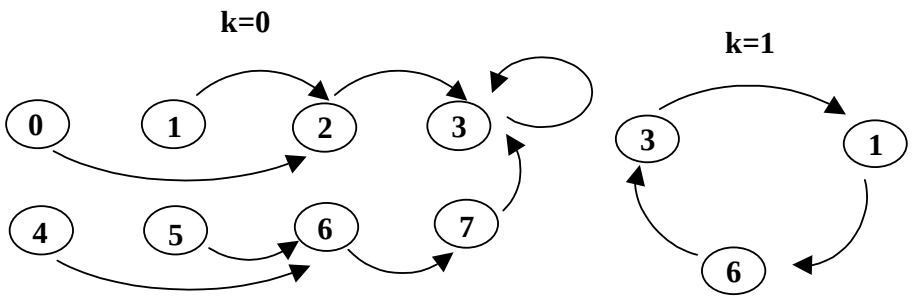
Elde edilen eşitlikleri gerçekleştirecek bağlantıların yapılması ile istenilen lojik devre oluşur (Şekil 11.39).



Şekil 11.39. 1, 2, 4 dizisini sayan sayıcı lojik devresi ve oluşan dalga şekilleri.

Örnek 10: J-K FF'ler ve en az sayıda 'VEDEĞİL' devresi kullanarak, Şekil 11.40'da verilen sayma işlemini gerçekleştiren senkron sayıcı devresini tasarlayalım.

Sayma işleminin yönü, kontrol girişi (k) olarak kullanılan girişe göre değişmektedir. Bu nedenle geçiş tablosu oluşturulurken, k=0 ve k=1 durumları ayrı ayrı değerlendirilmelidir (Şekil 11.40).

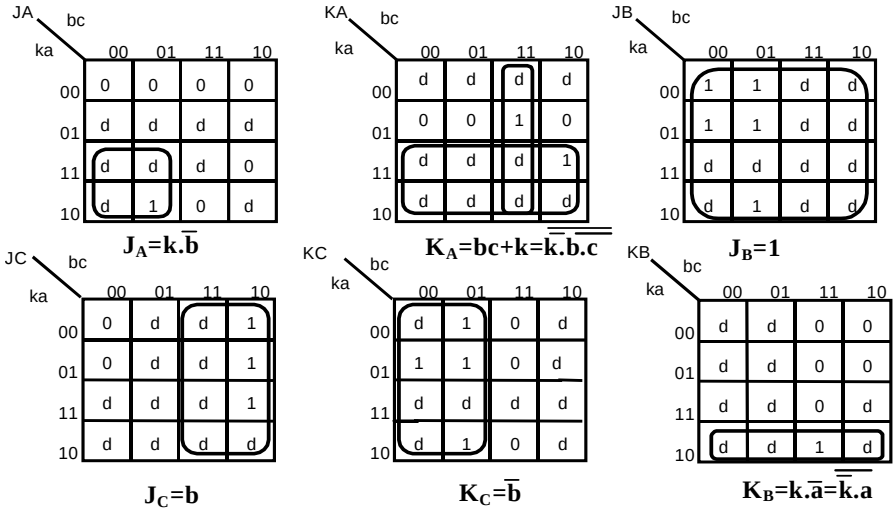


Şekil 11.40. Örnek sayma dizisi durum şeması.

İlk işlem olarak, önceki durum ve sonraki durum değerlerinden faydalanılarak sayma dizisi geçiş tablosu oluşturulur (Şekil 11.41).

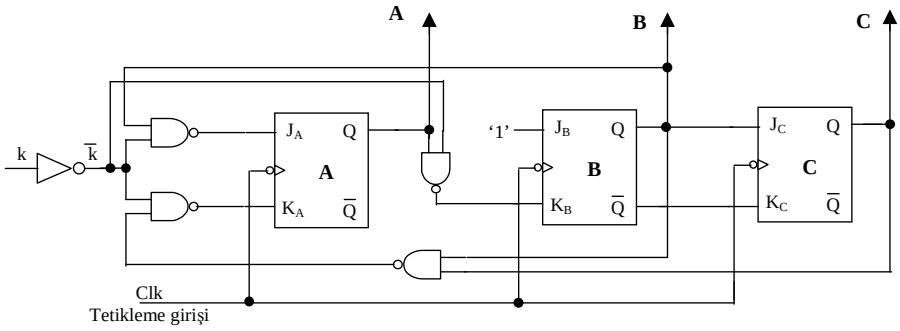
Önceki Durum				Sonraki Durum			FF Durumları					
k	**a**	**b**	**c**	**a**	**b**	**c**	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	0	d	1	d	0	0
0	0	0	1	0	1	0	0	d	1	d	d	1
0	0	1	0	0	1	1	0	d	d	0	1	d
0	0	1	1	0	1	1	0	d	d	0	d	0
0	1	0	0	1	1	0	d	0	1	d	o	1
0	1	0	1	1	1	0	d	0	1	d	d	1
0	1	1	0	1	1	1	d	0	d	0	1	d
0	1	1	1	0	1	1	d	1	d	0	d	0
1	0	0	0	d	d	d	d	d	d	d	d	d
1	0	0	1	1	1	0	1	d	1	d	d	1
1	0	1	0	d	d	d	d	d	d	d	d	d
1	0	1	1	0	0	1	0	d	d	1	d	0
1	1	0	0	d	d	d	d	d	d	d	d	d
1	1	0	1	d	d	d	d	d	d	d	d	d
1	1	1	0	0	1	1	d	1	d	0	1	d
1	1	1	1	d	d	d	d	d	d	d	d	d

Şekil 11.41. Örnek sayma dizisi geçiş tablosu.



Şekil 11.42. Sayıcı devresi giriş eşitliklerinin oluşturulması.

Oluşturulan geçiş tablosundaki değerlerin Karnaugh haritalarına taşınmasıyla, çizilecek lojik devrenin bağlantılarını gösteren eşitlikler elde edilir (Şekil 11.42). Eşitliklerin temsil ettikleri bağlantıların yapılması ile, istenen sayma dizisini gerçekleştirecek lojik devre oluşturulur (Şekil 11.43).



Şekil 11.43. Dizi sayıcı lojik devresinin çizilmesi.

11.2.6. Ön Kurmalı Sayıcılar

Belirli bir değerden başlatılarak, aşağı veya yukarı doğru sayma işlemi yapan devreler, **'Ön kurmalı sayıcılar'** olarak isimlendirilir.

Senkron sayıcı entegrelerinin birçoğu, sayma işleminin belirli bir değerden başlamasını sağlayan önkurma (preset) girişine sahiptirler. Önkurma işlemi, tetikleme sinyalinden bağımsız olarak (asynchronously) veya tetikleme sinyali ile eşzamanlı olarak (synchronously) gerçekleştirilebilir. Önkurma işlemi aynı zamanda **'sayıcının yüklenmesi'** (loading) olarakta adlandırılır.

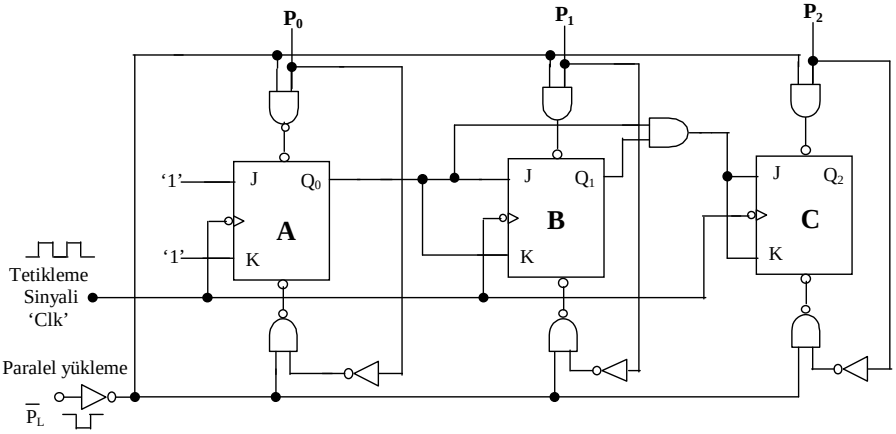
Şekil 11.44'de J, K ve Clk girişlerinin senkron çalışmayı sağlayacak şekilde bağlandığı, üç bitlik senkron yukarı sayıcı devresi görülmektedir. Devredeki kurma ve sıfırlama girişleri, asenkron ön kurma işlemini gerçekleştirebilecek şekilde bağlanmıştır.

Sayıcı devresine istenilen sayının yüklenmesi işlemi; P_0, P_1, P_2 paralel girişlerine yüklenmek istenen bilginin ve 'P_L' paralel yükleme girişine '0' değerinin uygulanması ile gerçekleştirilir. Bu işlem ile; P_0, P_1 ve P_2'ye uygulanan bilgiler sırasıyla A, B ve C FF'lerine aktarılır.

Bilgi aktarma işlemi; J, K ve 'Clk' girişlerinden bağımsız olarak gerçekleştirilir. Çünkü $\overline{P_L}$ girişi '0' olduğu sürece 'Clk' girişi etkisizdir. P_L'nin '1' değerine dönmesi ile, 'Clk' aktif hale gelir ve sayıcı kurulan sayma değerinden itibaren saymaya başlar. Örneğin; P_2=1, P_1=0, P_0=1 ve P_L=1 iken paralel girişlerin sayıcı devresine hiçbir etkisi yoktur ve 'Clk' sinyalleri ile sayıcı devresi sayma işlemini gerçekleştirir. P_L girişine '0' uygulanması ile, sayıcının

durumuna bakılmaksızın FF'ler Q_2=1, Q_1=0, Q_0=1 değerlerine kurulur. $\overline{P_L}$'nin '1' değerine dönmesi ile sayıcı 101'den başlayarak sayma işlemine devam eder.

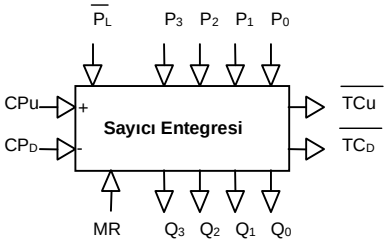
Önkurma işlemini gerçekleştirecek çok sayıda TTL ve CMOS entegresi bulunmaktadır. 74190, 74191, 74HC192, 74HC193 entegreleri bunlardan birkaçıdır.



Şekil 11.44. Asenkron önkurmalı senkron sayıcı devresi lojik şeması.

11.3. Sayıcı Entegreleri

Sayıcı devrelerin çok değişik uygulama alanları olması nedeniyle, çok sayıda sayıcı entegresi bulunmaktadır. Sayıcı entegrelerini sınıflandırmadan önce, sayıcı entegrelerde bulunan pinleri açıklayalım. Bu pinlerin hepsinin bir entegrede bulunmadığı durumlar olabileceği gibi, açıklanmayan farklı girişlerde bulunabilir. Burada genel pin fonksiyonları açıklanacaktır.



Pin adı	Açıklama
CP_U	Yukarı sayıcı tetikleme girişi
CP_D	Aşağı " " "
MR	Asenkron ana sıfırlama girişi (Aktif 1)
$\overline{P_L}$	Asenk. paralel yükleme girişi (Aktif 0)
P_0-P_3	Paralel veri girişleri
Q_0-Q_3	FF çıkışları
$\overline{TC_D}$	Aşağı sayıcı çıkışı terminali (Aktif 0)
$\overline{TC_u}$	Yukarı sayıcı çıkış terminali (Aktif 0)

Şekil 11.45. Sayıcı entegrelerinde bulunan pinler ve anlamları.

Sayıcı entegresinde bulunan pinlerin fonksiyonları aşağıdaki şekilde özetlenebilir.

CP_U ve CP_D Tetikleme girişleri: Sayıcı entegresi, bu iki girişten birisinin aktif olmasıyla sayma yönüne karar verir. CP_U'nun aktif olmasıyla sayma işlemi yukarı doğru yapılırken, CP_D'nin aktif olmasıyla sayma işlemi aşağı doğru gerçekleşir.

Ana Sıfırlama (Master Reset –MR): Ana sıfırlama ucu, aktif '1' asenkron girişidir ve sayıcıyı '0000' konumuna getirir. MR=1 olduğu sürece sayıcı '0000' konumunda sabit kalır.

Önkurma Girişi (Preset Input): Sayıcıdaki FF'ler, P_0-P_3 paralel veri girişlerine lojik bilgilerin uygulanması ve P_L girişindeki bilginin 1'den 0'a gitmesiyle kurulurlar. Bu kurma, asenkron önkurma işlemidir ve sayma işlemine üstünlüğe sahiptir. MR=1 olduğu durumda P_L'nin hiçbir etkisi yoktur.

Sayma Çıkışları: Sayma işleminde oluşan değerler Q_0-Q_3 arasındaki FF çıkışlarında görülür. Q_0 çıkışı LSB, Q_3 çıkışı MSB değerini temsil eder.

Terminal Sayma Çıkışları (TC_U): Büyük Mod'a sahip sayıcıları oluşturmada, yani iki veya daha fazla sayıcı entegresinin birlikte kullanılacağı durumlarda kullanılır. Yukarı sayma işleminde, düşük değerli sayıcının TC_u çıkışı, daha yüksek değerli sayıcının CP_u girişine bağlanır. Aşağı sayma işleminde, düşük değerlikli sayıcının TC_D çıkışı, yüksek değerlikli sayıcının CP_D girişine bağlanır.

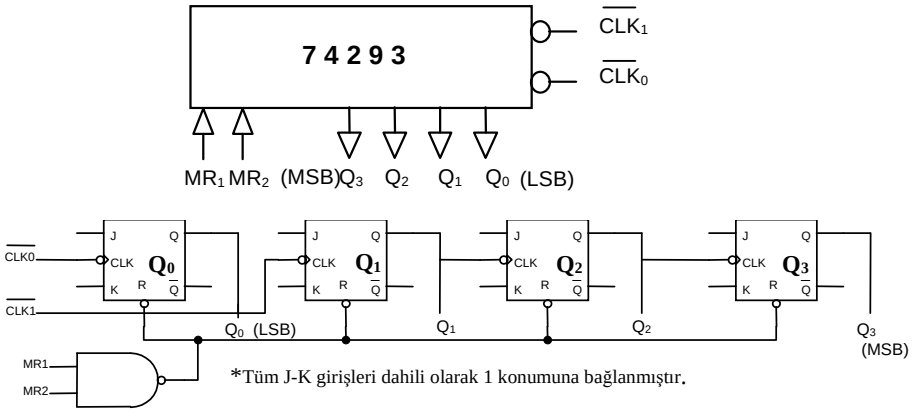
Yukarı sayıcı çıkış terminali olarak isimlendirilen TC_u çıkışı; yalnızca sayıcı '1111' değerinde ve CP_u=0 olduğu durumda '0' değerine sahip olur. Bir sonraki tetikleme sinyalinin yükselen kenarına kadar '0' olarak kalan TC_u çıkışı; yeni bir yükselen kenar ile '1' konumunu alırken, sayıcı '0000' konumuna geçer. Sayıcının 1111'den '0000' konumuna geçtiği anda TC_u'da oluşan 0'dan 1'e değişimin, daha yüksek değerlikli sayıcıya uygulanması ile yukarı doğru sayma işlemi sağlanır.

Aşağı sayıcı çıkış terminali olarak isimlendirilen TC_D çıkışı; sayıcının konumu '0000' değerini alıncaya ve CP_D=0 oluncaya kadar '1' konumunda kalır. Sayıcı çıkışı '0000' ve CP_D=0 iken, CP_D'ye gelen tetikleme sinyali ile sayıcı '1111' konumuna geçerken, TC_D=1 değerini alır. TC_D'de meydana gelen bu değişim, daha düşük değerlikli sayıcı olarak kullanılan ikinci sayıcı entegresini tetiklemek amacıyla kullanılır.

Sayıcı entegrelerinde bulunan pinleri genel olarak tanımladıktan sonra, sayıcı entegrelerini asenkron ve senkron sayıcı entegreleri gruplandırmaları altında inceleyelim.

11.3.1. Asenkron Sayıcı Entegreleri

Piyasada çok sayıda TTL ve CMOS asenkron sayıcı entegreleri bulunmaktadır. TTL 7493 (74LS93) ve bu elemanın eşiti 74293 entegreleri bunlardan ikisidir. CMOS ailesi içerisinde kullanılan bir asenkron sayıcı entegresi ise, Mod-128'e kadar sayma işlemi yapabilen 4024 entegresidir. Farklı üreticilerin ürettikleri aynı serideki entegreler arasında küçük farklılıklar bulunsa da, bu farklılıklar kolayca ortadan kaldırılabilir.

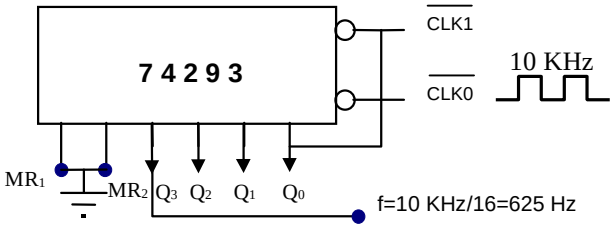


Şekil 11.46. 74293 asenkron sayıcı entegresi sembolü ve iç yapısı.

Hem TTL, hem de CMOS veri kataloglarında değişik asenkron sayıcı entegreleri bulmak mümkündür. Şekil 11.46'da, üretici veri kataloglarında bulunan 74293 entegresi sembolü ve entegreye ait lojik bağlantı görülmektedir. 74293 Sayıcı entegresinin çıkış pinlerinde Q_0-Q_3 olarak isimlendirilen 4 adet çıkış ve Clk_1-Clk_2 olarak isimlendirilen 2 adet tetikleme girişi mevcuttur. Her bir FF'nin sahip olduğu sıfırlama girişi, bir 'VEDEĞİL' kapısının çıkışına bağlı tek hat üzerinde birleştirilmiş ve 'VEDEĞİL' kapısının girişleri kontrol için MR_1 ve MR_2 pinleri haline getirilmiştir (Şekil 11.46). Kullanıcıya esneklik sağlaması amacıyla entegrenin içerisindeki FF'ler: Q_0 yalnız başına çalışacak şekilde ve Q_1-Q_3 3-bitlik sayıcı olarak işlem görecek şekilde bağlanmışlardır. 74293 entegresi uygulamalarına örnek olması açısından, örnek sayıcıları 74293 entegresi ile oluşturalım.

Örnek 11: 74293 entegresinin 10 KHz'lik tetikleme sinyali ile Mod-16 sayıcısı olarak işlem yapabilmesi için gerekli bağlantıyı gerçekleştirelim.

Mod-16 sayıcısı 4 adet FF kullanacağından, Q_0 çıkışının $\overline{Clk_1}$ girişine bağlanarak 4 FF'li bir sistem oluşturması gereklidir. 10 KHz'lik tetikleme sinyali ise Q_0 FF'sinin Clk_0 girişine uygulanmalıdır. Bu bağlantı ile, Mod-16 sayma işlemi yapan sayıcı (16'ya bölen) devre elde edilir.

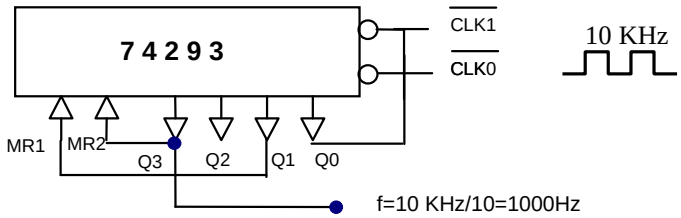


Şekil 11.47. 74293 entegresi ile oluşturulan Mod-16 sayıcı.

Örnek 12: 74293 entegresi ile Mod-10 sayıcısı oluşturmak için gerekli bağlantıyı gösterelim

Mod-10 sayıcısı oluşturmak için 4 adet FF gerektiğinden, Q_0 çıkışının $\overline{Clk}_1$ girişine bağlanması gerekir. Bununla beraber, sayıcının 0000-1010 değerleri arasında sayma yapması istendiğinden; sıfırlama için Q_1 ve Q_3 çıkışları MR_1 ve MR_2 girişlerine uygulanmalıdır. Bunun nedeni; yalnızca 1010 durumunda Q_1 ve Q_3 çıkışlarının aynı anda '1' olmasıdır. Q_1 ve Q_3 çıkışlarındaki bilgilerin birlikte '1' olduğu durumda, 'VEDEĞİL' kapısının çıkışı 1'den 0'a giderek sayma işlemini sıfırlar. Bu durumda ortaya çıkan bağlantı Şekil 11.48'de görülmektedir.

Elde edilen devreye 10 KH_Z'lik tetikleme sinyali uygulanması durumunda, Q_3 çıkışında 10 KH_Z / 10 = 1 KH_Z çıkış sinyali elde edilir.

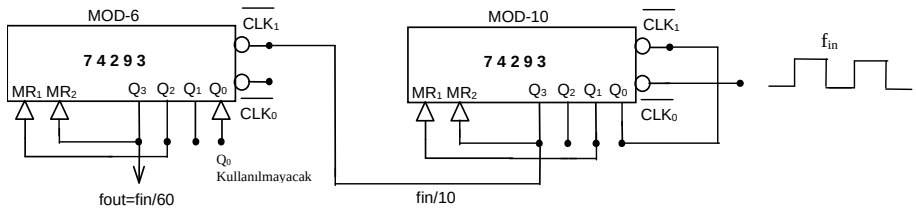


Şekil 11.48. 74293 entegresi ile oluşturulan Mod-10 sayıcı.

Örnek 13: Şekil 11.49'da verilen sayıcı devresinin analizini yaparak, çalışmasını açıklayalım.

Şekildeki devrede, girişe uygulanan sinyal iki kademede 60'a bölünür. Sağdaki devre Mod-10 sayıcı işlevi görerek, giriş sinyalini 10'a bölmekte ve bu entegrenin Q_3 çıkışından alınan bilgi Mod-6 sayıcı olarak çalışan ikinci entegrenin Clk_1 girişine uygulanmaktadır. Q_0 girişinin kullanılmadığı bu entegrede, uygulanan sinyalin frekans 6'ya bölünmektedir. Bu işlem sonucunda, soldaki entegrenin Q_3 çıkışında elde edilen sinyalin frekansı;

$$f_{out} = \frac{f_{in}/10}{6} = \frac{f_{in}}{60} \text{ olur.}$$

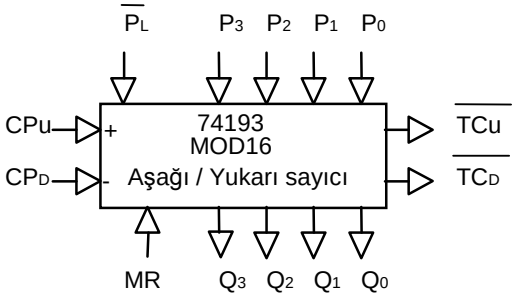


Şekil 11.49. İki adet 74293 entegresi ile oluşturulan MOD-60 sayıcı devresi.

Anlatılan 74293 entegresi, 0'dan başlayarak sayma işlemi yapma amacıyla kullanılmaktadır. Sayıcı devresinde ön kurma işleminin gerektiği durumlarda ön kurmalı sayıcı entegrelerinden faydalanılır. Ön kurma prensibi ile çalışan entegrelere örnek olarak; TTL 74160, 74161, 74163, 74193 ve CMOS 74HC162, 74HC163 entegreleri verilebilir.

Ön kurmalı sayıcı entegrelerinin parametrelerini ve entegre bağlantılarını incelemek için, 74193 sayıcı entegresi uygulamalarını örnek olarak inceleyelim. Bu entegreyi incelemek, diğer tipler hakkında genel bilgi verecektir.

Şekil 11.50.a'da lojik sembolü ve giriş / çıkış bağlantıları görülen 74193 entegresi; asenkron ana sıfırlama ve asenkron ön kurma özelliklerine sahip, senkron saymalı yukarı / aşağı sayıcı elemanıdır. 74193 entegresine ait çalışma tablosu ise Şekil 11.50.b'de verilmektedir. Yapılması istenen işlem, tablodaki değerlerden faydalanılarak gerçekleştirilir.



a) 74193 sayıcı entegresi sembolü

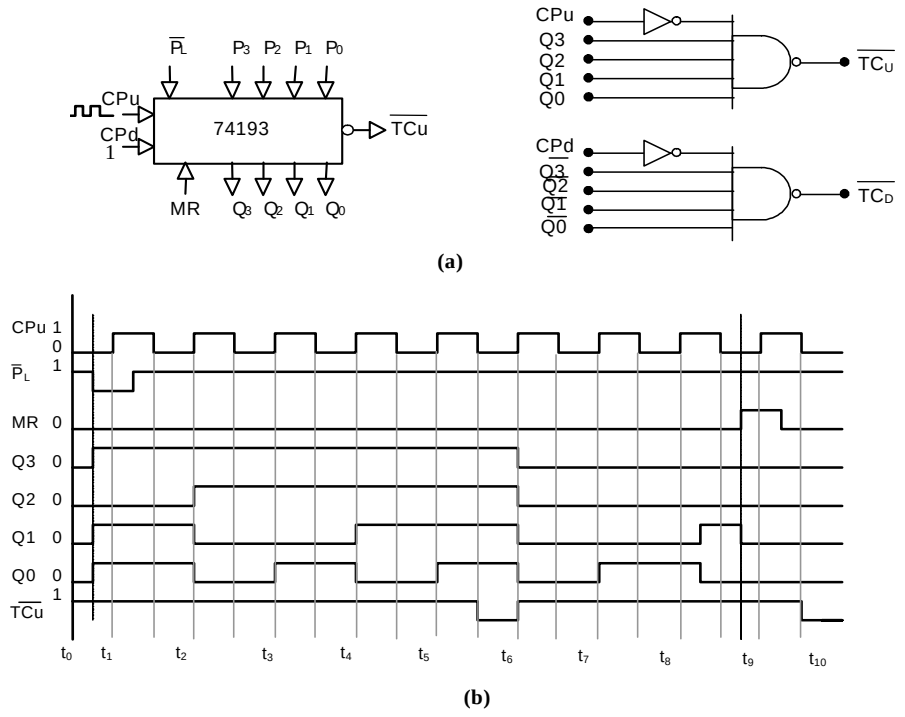
MR	**P_L**	**CP_U**	**CP_D**	**Çalışma Şekli**
H	X	X	X	Asenkron Sıfırlama
L	L	X	X	Asenkron Ön Kurma
L	H	H	H	Değişiklik Yok
L	H	↑	H	Yukarı Sayma
L	H	H	↑	Aşağı Sayma

H=1; L=0 ; X=Don't care ; ↑=PGT

b) Çalışma şekli seçme tablosu

Şekil 11.50. 74193 ön kurmalı aşağı/yukarı sayıcı sembolü ve çalışma tablosu.

Örnek 13: Şekil 11.51'de 74193 entegresinin yukarı sayıcı olarak kullanılması görülmektedir. Paralel veri girişlerine sabit olarak '1011' bilgisi ve CPu, P_L, MR girişlerine ise Şekil 11.51.b'de görülen bilgiler uygulanmaktadır. Sayıcının başlangıçta '0000' konumunda olduğunu kabul ederek, sayıcı çıkışında elde edilecek dalga şekillerini çizelim.



Şekil 11.51. 74193 entegresinin örnek bağlantısı ve oluşan dalga şekilleri.

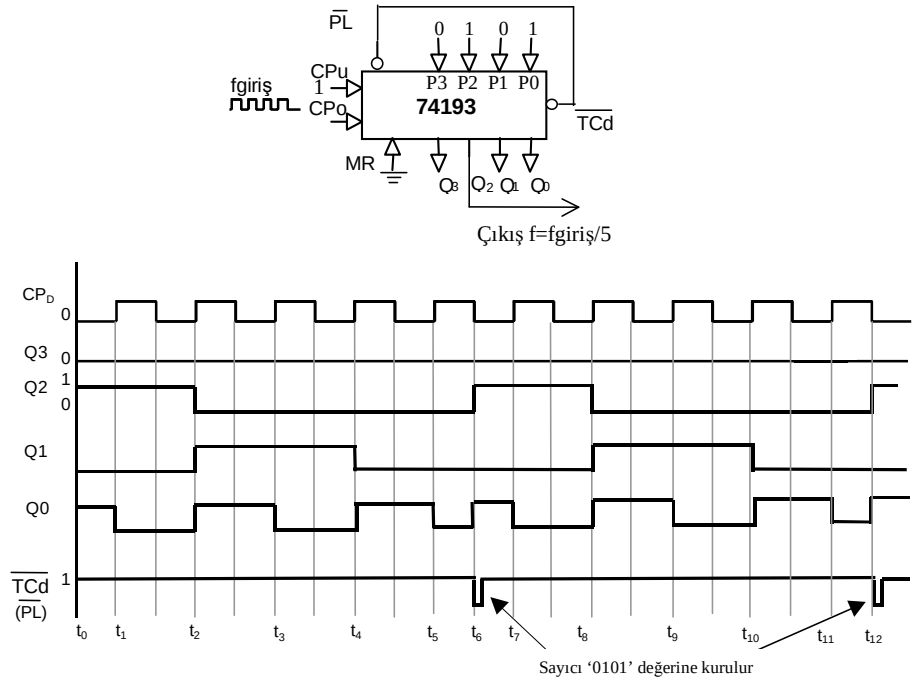
Başlangıçta (t_0 anında), FF'lerin hepsi '0' konumundadır. Bu durum $\overline{TC_u}$ çıkışının '1' olmasına neden olur. t_1 anından hemen önce $\overline{P_L}$ girişine '0' uygulandığından, FF'ler paralel girişlerdeki değerlere yani Q_3=1, Q_2=0, Q_1=1 ve Q_0 =1 değerleriyle yüklenirler. 't_1' anında CP_u girişinde yükselen kenar (PGT) olmasına rağmen, $\overline{P_L}$ pininin aktif '0' olması nedeniyle sayıcıda herhangi bir değişiklik olmaz. t_2, t_3, t_4, ve t_5 anlarında; tetikleme sinyalinin her yükselen kenarında sayıcı yukarı doğru sayar. t_5 anının yükselen kenarından sonra sayıcı, '1111' değerini alır. Ancak $\overline{TC_u}$ çıkışı, t_5 anında tetikleme sinyali 0'a gidene kadar değişmez. T_6 anında oluşan yükselen kenar ile $\overline{TC_u}$=1 değerini alırken, sayıcı '0000' değerine sahip olur.

t_7 ve t_8 anlarındaki yükselen kenarlarda sayma işlemi gerçekleştiren sayıcıda, t_9 anından önce '1' değerini alan MR girişi nedeniyle, sayıcı sıfırlanarak çıkışlar 0000 konumuna getirilir.

Örnek 14: 74193 entegresi kullanılarak değişik moddaki sayıların elde edilmesi:

Ön kurmalı sayıcılar, ek bir devreye ihtiyaç duymadan farklı moddaki sayıcıları elde etmek için kolayca şekillendirilebilir. Buna örnek olması bakımından Şekil 11.52'deki devreyi kullanabiliriz.

Aşağı sayıcı olarak kullanılan bu devrenin paralel yükleme girişlerine $(0101)_2$ değeri sabit olarak uygulanmaktadır. Devrede, TC_D çıkışı $\overline{P_L}$ girişine bağlandığına göre sayıcıdaki işlemleri inceleyelim.



Şekil 11.52. 74193 entegresi ile Mod-5 sayıcı işlemi ve oluşan dalga şekilleri.

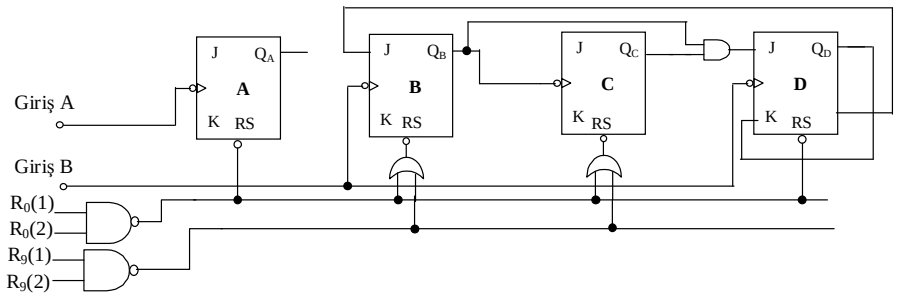
Sayıcıyı analiz etmek için, sayıcının başlangıçta t_0 anında '0101' değerine kurulduğunu ve aşağı sayma yapıldığını kabul edeceğiz.

Sayıcı devresi, $t_1 - t_5$ zamanları arasında geri sayma yaparak, t_5 anında '0000' değerine sahip olur. t_6 anında tetikleme sinyalinin 0'a geçişi TC_D'yi '0' yapar. Bu değer, $\overline{P_L}$ girişini aktif yaparak sayıcının '0101' konumunun kurulmasını sağlar. TC_D çok kısa bir süre '0' olarak kalır. Çünkü sayıcının '0101' değerine kurulması ve bu değeri koruması TC_D'nin '0' konumundan kurtulmasına bağlıdır. Bu nedenle TC_D'de yalnızca kısa süreli bir sinyal oluşur.

Anlatılan işlemler $t_7 - t_{12}$ arasında ve takip eden zaman aralıklarında tekrarlanır. Q_2 çıkışında meydana gelen değişmeleri inceleyecek olursak; Q_2 çıkışında meydana gelen değişimin tetikleme sinyalinde meydana gelen değişimin 1/5'i olduğunu görürüz. Bu durum, Q_2'nin frekansının tetikleme sinyali frekansının 5'te 1 olması sonucunu doğurur.

Değişebilir frekans bölücü devresi, Şekil 11.52'deki paralel veri girişlerine anahtarlar konularak elde edilebilir. Anahtarlar, oluşturulmak istenen frekans bölücünün değerine uygun olarak sayıcıya bilgi yükleyecek şekilde kurulurlar .

Sayıcı devrelerinde kullanılan diğer bir entegre tipi, iki adet bağımsız sayıcı içeren 7490 elemanıdır (Şekil11.53). Biri Mod-2, diğeri Mod-5 olan bu sayıcıların tetikleme girişleri birbirinden bağımsızdır. Şekil 11.53'de şemaları verilen bu sayıcıların, birbiriyle farklı bağlantıları sonucu 2'den 10'a kadar bölme işlemleri yapılabilir.



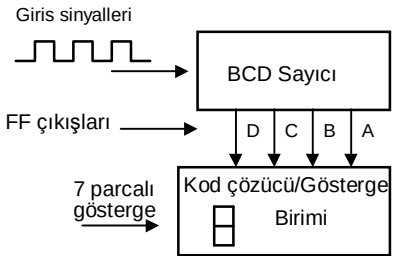
Şekil 11.53. 7490 sayıcı entegresi iç yapısı.

7490 entegresi ile BCD sayma için; Q_A çıkışı, giriş B'ye bağlanmalıdır. Sayma işleminin başlaması için, R_0 ve R_9 resetleme girişlerinin en az birer girişi '0' seviyeli olmalıdır.

7493 sayıcı entegresi, 7490 entegresine benzer şekilde, Mod-2 ve Mod-8 iki adet sayıcı içeren bir elemandır. Bu entegrede bulunan sayıcıların bağımsız veya kaskat bağlanmasıyla, 2'den 16'ya kadar istenilen MOD'da sayma işlemi yapılabilir.

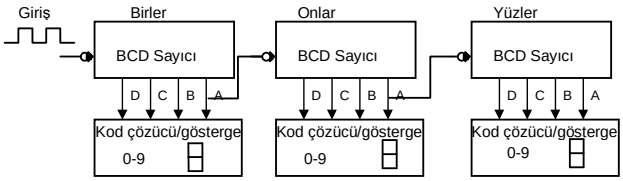
11. 4. Kaskat BCD Sayıcılar

BCD sayıcı, 0-9 arasındaki sayıları çıkış olarak veren 10 farklı çıkışa sahip bir devredir. 10 farklı çıkışın her birinin doğrudan bağlandığı bir LED ile sayma işlemi görüntülenebileceği gibi, yedi-parçalı (seven-segment) gösterge olarak isimlendirilen eleman yardımıyla sayıcı çıkışındaki bilgi izlenebilir. Sayıcı çıkışındaki bilginin göstergede izlenebilmesi için kod çözücü devreler kullanılır. Bu durumda, sayıcı işleminin oluşturulması ve göstergede görüntülenmesi için Şekil 11.54'deki devrenin oluşturulması gerekir. Bu devrede, BCD sayıcıda gerçekleştirilen sayma işlemi kod çözücü devre yardımıyla uygun forma dönüştürülür ve gösterge ile görüntülenir.



Şekil 11.54. BCD sayıcı blok şeması.

BCD sayıcı olarak isimlendirilen Şekil 11.54'deki devre, 0'dan 9'a kadar sayma işlemi yapıp görüntüleyerek tekrar 0'a döner. Desimal sayılardan daha büyük sayıları saymak ve görüntülemek için, BCD sayıcılar Şekil 11.55'deki gibi kaskat olarak bağlanır.



Şekil 11.55. BCD Sayıcıların 000-999 arasında sayması için kaskat bağlantısı.

Kaskat bağlantılı BCD sayıcılarda sayma işlemi aşağıdaki gibi özetlenebilir:

i- Başlangıçta '000' değerine kurulan sayıcı devresi, tetikleme sinyalleri ile 0'dan başlayarak saymaya başlar.

ii- Gelen her tetikleme sinyali ile sayma işlemi yapan sayıcı devresi, $(1001)_2$ değerine kadar sayma işlemi gerçekleştirir. $(1001)_2$ değerinde göstergede '009' onluk sayı okunur.

iii- Girişe uygulanan tetikleme sinyalinin onuncu palsında 1'ler basamağındaki sayıcı çıkışı 0'a dönerken, birler basamağını temsil eden FF onlar basamağındaki sayıcıya bir pals gönderir. Bu pals, onlar basamağındaki sayıcıda bir artmaya neden olur. Bu anda göstergede $(010)_{10}$ değeri okunur.

iv- Birler basamağındaki her on pals sonucunda gönderilen palslerin sayısı '10' değerine ulaşana kadar onlar basamağındaki sayma devam eder. Girişten uygulanan 99. tetikleme sinyali sonucunda göstergede gözüken değer 099'dur. Takip eden ilk tetikleme sinyal ile birler basamağındaki sayıcı onlar basamağındaki sayıcıya bir pals gönderip, 0'a döner. Bu palsı alan onlar basamağındaki sayıcı 0'a dönerken, aynı anda yüzler basamağındaki sayıcıya bir tetikleme sinyali gönderir. Bu anda göstergede '$(100)_{10}$' değeri okunur.

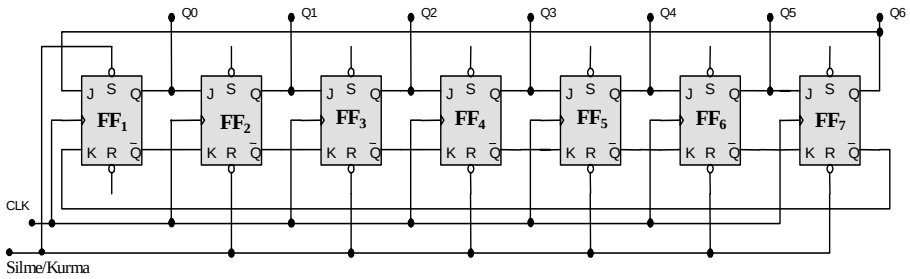
v- Anlatılan işlemler 999. sinyale kadar devam eder ve 1000. sinyal ile tüm sayıcılar ve göstergeler 0'a döner.

999'a kadar sayma işlemi gerçekleştiren bu devrede, yapılan sayma işleminin daha büyük değerleri sayabilmesi için yeni katların eklenmesi gerekir. BCD sayıcı olarak, BCD sayma işlemi için tasarlanan 7490 veya 74192 sayıcı entegreleri kullanılabileceği gibi, uygun bağlantı ile 74293 entegresi gibi elemanlarda kullanılabilir.

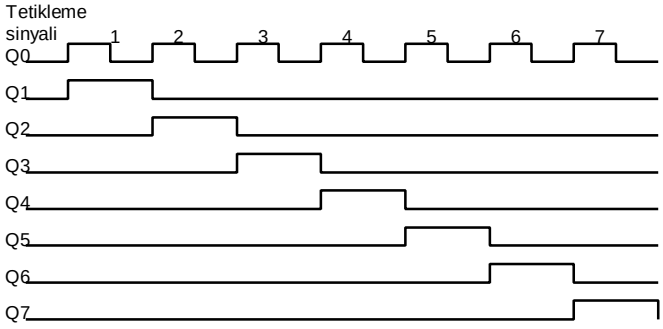
11.5. Halka ve Johnson Sayıcılar (Ring Counter and Johnson Counter)

Sayıcı girişinde kod çözme işleminin olmasının istendiği devrelerde, halka sayıcı (ring counter) olarak isimlendirilen devre kullanılır. Halka sayıcı devrede her bir sayma için bir adet FF kullanılır (Şekil 11.56). Örneğin; 7 sayan bir halka sayıcı için 7 adet FF kullanırken, 10 sayan bir halka sayıcı istenmesi durumunda 10 adet FF kullanılır. Herhangi bir anda bu FF'lerden sadece birinin çıkışı 1'dir (Şekil 11.57).

Halka sayıcı devresi, bir FF'nin Q ve Q^{I} çıkışlarının, bir sonraki FF'nin J ve K girişlerine bağlanmasıyla elde edilir. En sondaki FF'nin Q çıkışı ise ilk FF'nin girişine uygulanır.



Şekil 11.56. Halka sayıcı açık devresi.



Şekil 11.57. Halka sayıcı devresi dalga şekilleri.

Kurma / Silme girişi ile, FF_1 '1' değerine kurulurken, diğer tüm FF'ler sıfırlanır ve '0' değerine sahip olur. Tüm tetikleme sinyallerinin aynı kaynaktan elde edildiği devrede, ilk tetikleme sinyali ile J girişinde '1' bulunan FF_2 '1' değerine kurulurken, diğer FF'lerin çıkışı '0' dır. Gelen tetikleme sinyali ile Q çıkışının '1' olmasını sağlayan şartlara sahip olan FF'ler sırasıyla '1' konumuna geçer. FF'lerde Q=1 olması için gerekli olan J=1, K=0 olma şartı sırasıyla sağlandığından, çıkışlarda elde edilen bilgiler Şekil 11.57'deki gibi olur. Herhangi bir anda sadece tek çıkışı '1' olan ve girişine uygulanan tetikleme sinyallerini sayan bu devre, bir çeşit senkron sayıcı çalışması gerçekleştirir.

Halka sayıcıda yapılan işlemi küçük bir değişikle gerçekleştirilen diğer bir sayıcı türü, '**Johnson sayıcı'** devresidir. Bu devrenin halka sayıcı devresinden farkı; son FF'den ilk FF'ye yapılan geri besleme bağlantısının, Q çıkışı yerine $Q^{'}$ çıkışından alınmasıdır.

Şekil 11.58.a'da doğruluk tablosu, b'de sayma dizisi ve c'de açık şeması görülen Johnson sayıcıda, ilk anda tüm FF'lerin çıkışı 0'dır. İlk gelen tetikleme sinyali ile; J=1, K=0 değerine sahip olan FF_1'in Q çıkışı '1' değerini alır. FF_1'in çıkışı; FF_2'nin J=1, K=0 değerini almasını sağlar ve gelen ikinci sinyal ile FF_2'nin Q çıkışı '1' değerini alır. Bu iki çıkışın '1' değerini koruduğu anda gelen üçüncü tetikleme palsı ile J=1, K=0 değerine sahip olan FF_3'ün Q çıkışında '1' oluşur. Konum değiştirmeler bu şekilde devam ederek 4 sayma sonucunda tüm çıkışlar '1' olur. Şekil 11.58'de 4 sayma için gerçekleştirilen sayma dizisi, FF sayısının değiştirilmesi ile istenilen sayma işlemini gerçekleştirecek şekilde modellenebilir. Sayılmak istenen sayı dizisinin yarısı kadar FF'ye ihtiyaç vardır.

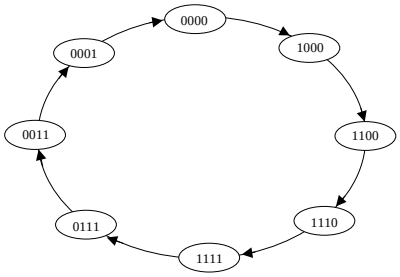
Şekil 11.58'de görülen devre, sekiz ayrı konumda çıkış verdikten sonra tekrar başa döner. Bu nedenle anlatılan sayıcı, Mod-8 sayıcısıdır. Bu örnekten görüldüğü üzere; Johnson sayıcı oluşturmak için gerekli FF sayısı, ring sayıcıların aksine sayılmak istenen değerin yarısı kadardır. Bununla beraber, Johnson sayıcı devresi her bir sayma durumunu tespit edecek (çözecek) lojik kapılara ihtiyaç duyar. Kullanılan FF sayısı ne olursa olsun, 2 girişli 'VE' kapısının kullanımı yeterli olur (Şekil 11.58.a). Doğruluk tablosunda girişleri açıklanan 8 adet iki girişli 'VE' kapının devreye bağlanması ile, her kapı belli bir durum anında yetkilenir ve kapı çıkışları birbirini izleyen 8 çıkış (halka sayıcısının aynısı) üretir.

Kapıların bağlanacağı FF'lerin tespiti düzenli bir sıra takip eder. Tüm çıkışların '0' olduğu anda, başta ve sonda bulunan FF'lerin çıkışlarının tümleyeni alınırken, tüm çıkışların '1' olduğu anda başta ve sonda bulunan FF'lerin normal çıkışları alınır. Diğer bütün durumlar, komşu halde bulunan 01 veya 10 değerlerine sahip FF'lerin çıkışlarının alınmasıyla çözümlenir.

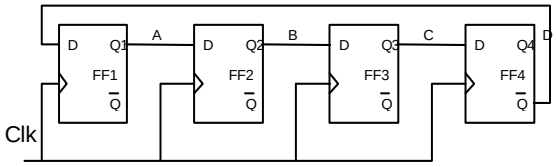
Devre karmaşıklığı bakımından halka sayıcı ile ikili sayıcılar arasında bir devre yapısına sahip olan Johnson sayıcılar, belirli uygulamalarda sağladıkları avantajlar nedeniyle tercih edilirler. Piyasada yaygın olarak bulunan Johnson sayıcılardan ikisi, tüm kod çözücü devreyi de içeren 4017 ve 4022 entegreleridir.

Tetikleme Palsı	Q_1	Q_2	Q_3	Q_4	Gerekli 'VE' kapısı
0	0	0	0	0	A'D'
1	1	0	0	0	AB'
2	1	1	0	0	BC'
3	1	1	1	0	CD'
4	1	1	1	1	AD
5	0	1	1	1	A'B
6	0	0	1	1	B'C
7	0	0	0	1	C'B

a) Doğruluk tablosu.



b) Sayma dizisi.



c) Açık şeması.

Şekil 11.58. MOD-8 Johnson sayıcı doğruluk tablosu, sayma dizisi ve ve açık şeması.

11.6. Sayıcı Uygulamaları

Çok çeşitli yapıda üretilen sayıcı devreleri çok farklı uygulama alanlarında kullanılmaktadır. En yaygın kullanım yeri olarak; sinyallerin frekanslarının ölçülmesi, elektriki sinyallerin frekanslarının bölünmesi, devir sayısının gösterilmesi, saat veya zaman ölçülmesi, vb. uygulama alanları sayılabilir. Sayıcı devrelerinin farklı uygulama alanlarında kullanılma prensiplerini inceleyelim.

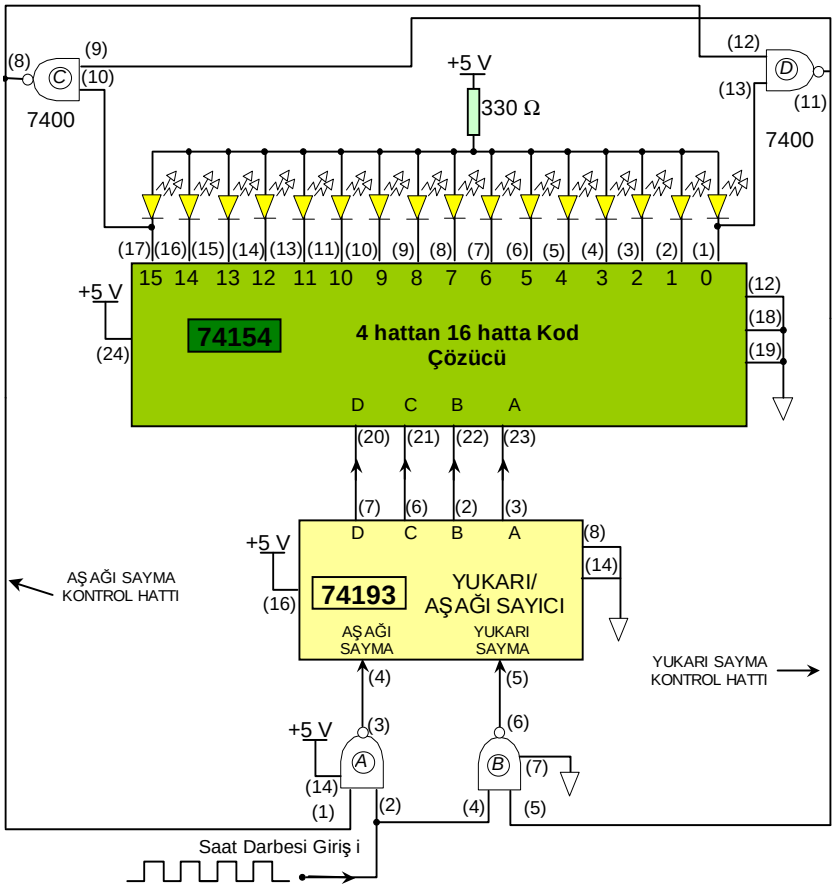
11.6.1. Sağa / Sola Yürüyen Işıklar

74193 yukarı / aşağı sayıcı entegreci kullanarak, sağa veya sola doğru yanan ışıklar oluşturmak mümkündür (Şekil 11.59).

74193 entegresi ile yukarı / aşağı saydırma işlemi yapılabilir. 0000 ile 1111 değerleri arasında yukarı/aşağı sayma yapan entegrenin çıkışı, 4 hattan 16 hatta kod çözücü (onaltılı kod çözücü) olan 74154 entegresine giriş olarak uygulanır. Bu durumda 16 LED'in sağa veya sola yanması kontrolü yapılabilir. Devredeki LED'lerin sağa / sola yanması işlemi, 'VE DEĞİL' kapıları A,B,C,D kapıları yardımı ile gerçekleştirilir.

Şekil 11.59'daki devrenin çalışmasını açıklayabilmek için; 'A' kapısının 1 nolu girişinin 'lojik 0' ve 'B' kapısının 5 nolu girişinin lojik '1' olduğunu kabul edelim. 'C' ve 'D'

kapılarının bağlantı şekillerinden dolayı, belirtilen iki noktadaki lojik değerler devamlı birbirinin tersidir. ‘B’ kapısının girişlerinden birisinin ‘lojik 1’ olması nedeni ile, tetikleme sinyali 74193 entegresinin ‘yukarı sayma’ girişinde gözükür ve sayıcı 0000’den başlayarak yukarı doğru saymaya başlar.



Şekil 11.59. Sağa / sola yürüyen ışık devresi.

Sayıcı entegresi çıkışında oluşan ikili değerler 74154 entegresine giriş olarak uygulanmaktadır. Kod çözücü entegresi çıkışları, girişlerdeki ikili değerlere bağlı olarak ‘lojik 0’ seviyesine düşer. ‘0’ seviyesine düşen çıkışa bağlı olan LED iletime geçerek ışık verir (yanar). Çıkışlar sağdan sola doğru aktif olduğundan, LED’ler sağdan sola doğru yanar.

Sayıcı çıkışındaki sayma değeri ‘1111’ değerine ulaşınca en soldaki LED yanar ve ‘C’ kapısının girişlerinden birisi ‘0’ olur. Girişlerinden birisi ‘0’ olan ‘C’ kapısının çıkışı ‘1’ olur ve ‘A’ kapısının ‘1’ nolu pine bağlı olan ucu ‘lojik 1’ olur. ‘A’ kapısının çıkışı ‘1’ değerini

alır ve 74154 entegresinin ‘aşağı sayma’ ucunu aktif hale getirir. Aynı anda, ‘C’ kapısının çıkışı ‘D’ kapısının girişini de ‘1’ yapar ve her iki girişinde ‘1’ olan ‘D’ kapısı çıkışında ‘lojik 0’ oluşur. ‘D’ kapısı çıkışında oluşan ‘0’ değeri, tetikleme sinyalinin ‘B’ kapısı üzerinden ‘yukarı sayma’ ucuna uygulanmasına engel olur. ‘D’ kapısının çıkışındaki ‘lojik 0’ değerinin ‘C’ kapısına giriş olarak uygulanması, kod çözücü entegrenin en solundaki çıkışının ‘0’ olmaması durumunda bile ‘aşağı sayma’ ucunun aktif olarak kalmasını sağlar.

Sayıcı entegresinin aşağı doğru sayması, kod çözücü çıkışlarının soldan sağa doğru aktif olmasını (lojik 0) ve LED’lerin soldan sağa doğru yanmasını sağlar. Sayma işlemi ‘0000’ değerine ulaştığı anda, ‘B’ kapısı çıkışı tekrar ‘yukarı sayma’ girişini aktif yapar ve LED’ler sağdan sola doğru yanmaya başlar.

11.6.2. Dijital Saat

Sayıcıların yaygın kullanıldığı yerlerden birisi, dijital (sayısal) saatlerdir (Şekil 11.60). Sayısal saatlerde; saat, dakika ve saniye olarak bulunulan an gösterilir.

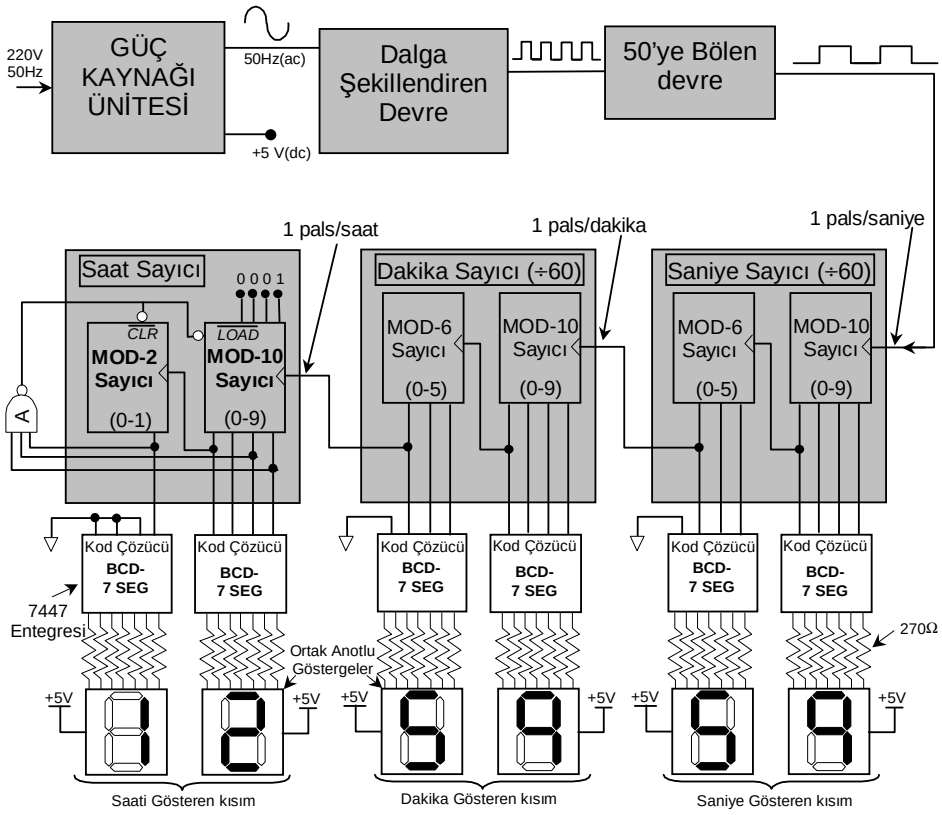
Şekil 11.60’daki dijital saat devresinin çalışması için gerekli besleme gerilimi ve sayma işlemi için gerekli tetikleme sinyali, 220 V / 50Hz şehir şebekesinden elde edilir. 220 V AC gerilim doğrultmaç devresi ile +5V DC gerilim şekline dönüştürülürken, 50 Hz’lik sinyal 1 Hz’lik tetikleme sinyali şekline dönüştürülür. Bu işlemler için; dalga şekillendirici ve 50’ye bölen devreler kullanılır.

Saat devresindeki saniye sayıcı devreye uygulanan tetikleme sinyali, saniye sayma işleminin yanında dakika ve saat sayma işlemleri içinde kullanılır.

Saniye sayıcı devre, her gelen tetikleme sinyali ile sayma işlemi yapan Mod-60 sayıcı devresidir. 60’a kadar sayma işlemi yapan saniye sayıcıdaki Mod-10 sayıcısı 9’dan 0’a dönerken, Mod 6 sayıcısına tetikleme sinyali gönderir. Saniye sayıcıdaki değer 59 olduğu durumda, sayma değeri 0’a döner ve bu anda dakika sayıcıdaki Mod-10 sayıcısına bir tetikleme sinyali gönderir.

Dakika sayıcıdaki devrede Mod-60 sayıcı devresidir. Mod-10 sayıcısı ile 9’a kadar sayma işlemi yapılırken, Mod-6 sayıcı ile sayma işleminin 59’a kadar olması sağlanır. Devredeki sayma işlemi 0’a dönerken, Mod-12 sayıcı olarak çalışan saat sayıcı devreye bir pals tetikleme sinyali gönderilir. Bunun uygulamadaki anlamı; 60 dakika sonucunda saat gösteren devrenin gösterdiği değerin ‘1’ artmasıdır.

Saat sayıcı devresi, Mod-10 ve buna bağlı olarak çalışan Mod-2 sayıcı devresi 12’den 13’e $(0001\ 0011)_{BCD}$ geçtiği zaman, ‘A’ kapısının (NAND kapısı) tüm girişleri lojik ‘1’ ve ‘A’ kapısı çıkışı lojik ‘0’ olur. ‘CLR’ girişine ‘0’ uygulanan Mod-2 sayıcısı 0’a giderken, veri yükleme girişine ‘0’ uygulanan Mod-10 sayıcısı ‘1’ değerini alır. Bu durum; saatin ‘1:00’ değerinden başlamasını, yani ‘12:59’ değerini gösterdikten sonra ‘1:00’ değerine dönmesini sağlar.



Şekil 11.60. Dijital saat devresi lojik şeması.

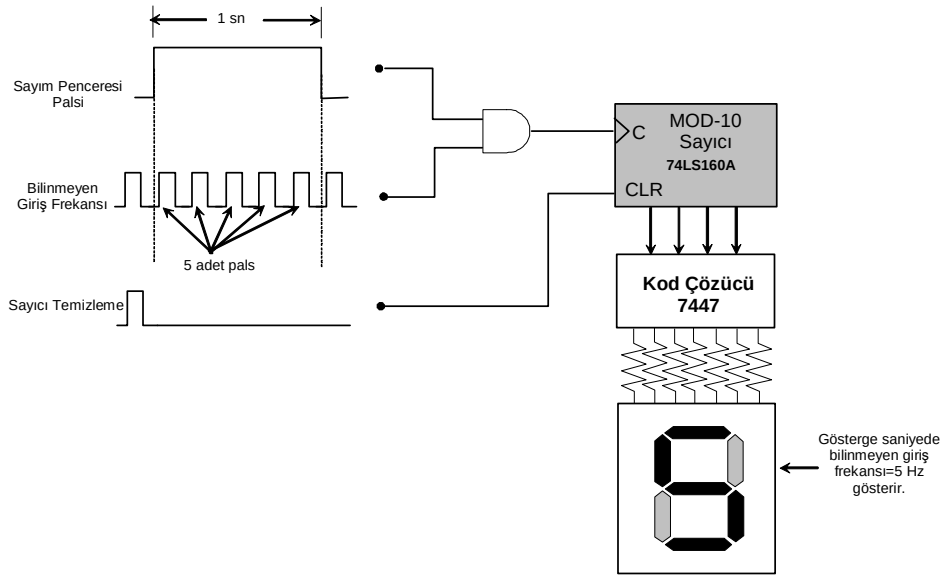
11.6.3. Frekans Sayıcı

Frekans sayıcı devresi, mantık devreleri olarak açıklanan sayıcı, kaydedici, kod çözücü, frekans bölücü, vb. devrelerinin bir arada kullanıldığı dijital sistemlere en iyi örneklerdir. Temel eleman olarak sayıcının kullanılması nedeni ile, frekans sayıcı devresi sayıcı devrelerine uygulama olarak kabul edilebilir.

Frekans sayıcı devresinde, frekans ölçümü işleminden önce devrede bulunan sayıcıların temizlenmesi (sıfırlanması) gerekir. Frekans ölçme işleminde, sayım penceresi palsi (count window pulse) ile bilinmeyen giriş frekansı bir 'VE' kapısı girişlerine uygulanır (Şekil 11.61). 'VE' kapısı, bir kontrol anahtarı gibi görev yapar. Çünkü 'VE' kapısının sayım penceresi palsı uygulanan girişinde '0' olması ile kapı açık anahtar özelliği gösterirken, girişin '1' olması ile kapı kapalı anahtar özelliği gösterir. Kapının kapalı anahtar özelliği göstermesi ile, bilinmeyen giriş frekansı sayıcı devresinin tetikleme girişine uygulanır. Şekil

11.61'deki devrede, sayım penceresi palsı süresinde bilinmeyen giriş sinyali olarak 5 pals'lik bir sinyal uygulandığından, Mod-10 sayıcı 5 kere tetiklenir ve çıkışında '0101' değeri oluşur. Sayıcı çıkışında oluşan değerin, BCD'den yedi parçalı göstergeye kod çözücü devresine uygulanması ile göstergede '5' değeri okunur. Sayım penceresi palsı süresi olarak 1 sn seçilmesi nedeni ile, okunan sinyalin frekansı 5 Hz'dir.

Bilinmeyen giriş sinyalinin sayım sinyali palsinin aktif olması sırasında örneklenmesi nedeni ile sayım penceresi palsi, 'örnekleme palsı' olarak da isimlendirilmektedir.

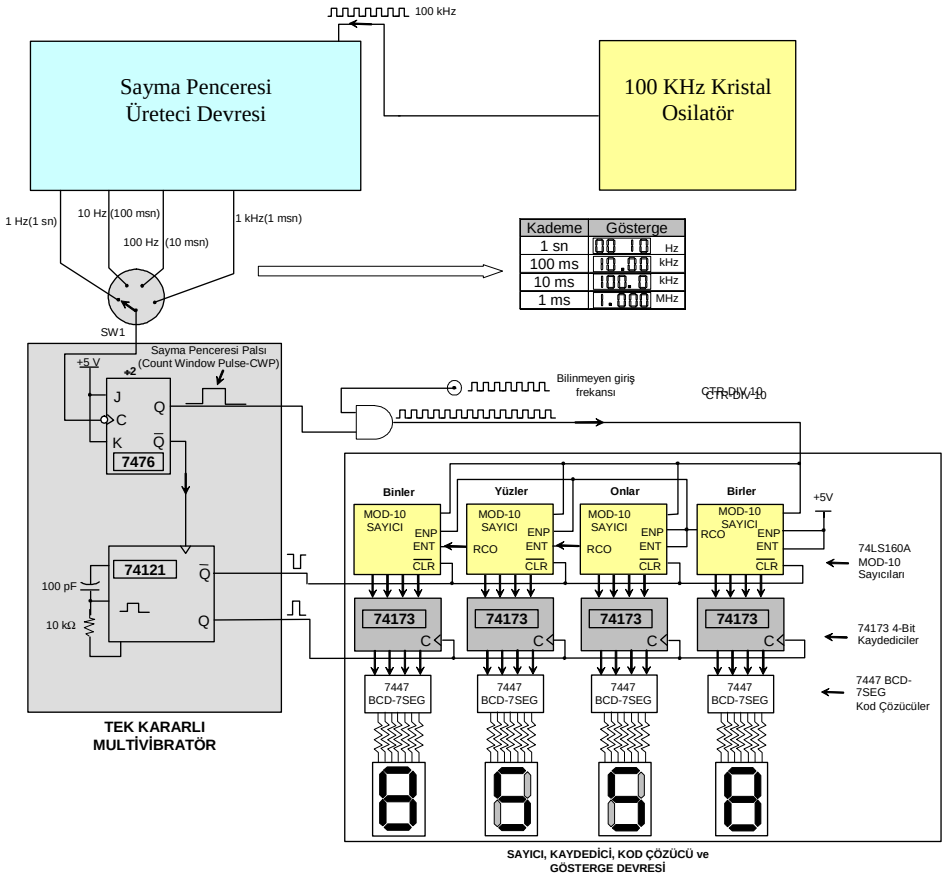


Şekil 11.61. Frekans sayıcı devresi prensip şeması.

Şekil 11.61'deki devrede Mod-10 sayıcı kullanılması nedeni ile, 10 Hz'e kadar frekans ölçümü mümkündür. Daha yüksek frekansları ölçebilmek için, ek lojik devrelerin eklenmesi gerekir. Şekil 11.62'deki devrede, çok kademeli frekans sayıcı devresi lojik şeması görülmektedir.

Frekans sayıcı devre ile ölçülen frekansın hassasiyeti, sayım penceresi palsinin hassasiyetine bağlıdır. 100 KHz'lik bir kristal osilatör kullanımı ile, 1 KHz'lik frekansın ölçülebilmesi için yeterli hassasiyet sağlanabilir. Kristal osilatörden üretilen 100 KHz'lik sinyal, sayma penceresi üretici devresinde (74L5160A) kaskat bağlı sayıcı devrelerde bölünerek; frekans ölçümünde 1 Hz, 10 Hz, 100 Hz ve 1KHz ölçme kademelerinin oluşturulmasını sağlar. SW çok konumlu anahtar (komülatör) ile, ölçüm yapılacak frekans kademesi seçilir.

Seçilen frekans kademesine bağlı olarak, ikiye bölücü FF devresinin (7476) çıkışında sayma penceresi palsi oluşur. Sayma penceresi palsi ile bilinmeyen giriş frekansı bir ‘VE’ kapısına uygulanır. Şekildeki devrede ‘VE’ kapısı çıkışında sayma penceresi palsinin genişliğine bağlı olarak palsler oluşur. ‘VE’ kapısı çıkışından elde edilen 14 pals, dört kademeli sayıcı devresine uygulanır.



Şekil 11.62. Frekans sayıcı devresi lojik şeması.

Sayıcı devresinde, palsler sayılarak ölçülmek istenen frekans değeri bulunur. Birler basamağını temsil eden Mod-10 sayıcı ‘9’ değerine ulaştığı anda, onlar basamağını temsil eden sayıcıya bir pals gönderir ve 0’a döner. İncelenen devrede, sayma işlemi 14’e kadar devam eder ve bu durumda ölçülmek istenen sinyalin frekansı 14 Hz olarak okunur.

Sayıcı devresi çıkışı, BCD olarak kaydedicilere, kaydedicilerin çıkışları ise yedi parçalı göstergeleri süren BCD’den yedi parçalı göstergeye kod çözücü devresine uygulanır. Diğer

taraftan, tek kararlı multivibratörün Q' çıkışı kaydedicilerin tetikleme girişine uygulanır. Bu pals ile, sayıcı devresinden kaydedicilere yüklenen değer saklanır. Kaydedici çıkışlarındaki değerler, kod çözücü devrede yedi parçalı göstergede gözükecek şekle dönüştürülür ve değerin bir süre gözükmesi kaydediciler yardımıyla sağlanır.

Tekrarlama ve Çalışma Soruları

1. 'Sayıcıyı' tarif ediniz

2. Sayıcıları sayma yönüne ve sayma kodlamasına göre sınıflandırınız.

3. Asenkron ve senkron sayıcıları tarif ediniz.

4. Aşağı, yukarı ve aşağı / yukarı sayıcıları tanımlayınız.

5. İki bitlik dalgacık sayıcı şemasını ve çıkış dalga şeklini çizerek, devrenin çalışmasını açıklayınız.

6. Düşen kenar tetiklemeli 4 FF kullanarak, 0-15 arasında sayan yukarı sayıcı devresini çiziniz ve devrenin çalışmasını açıklayınız.

7. FF'ler ile frekans bölme işleminin genel prensibini açıklayınız

8. Asenkron yukarı sayıcıda, aktif '1' resetleme girişine sahip FF'ler kullanıldığına göre, sayıcıyı sıfırlayacak devreyi ekleyiniz.

9. Ön kurmalı sayıcıyı şekille açıklayınız.

10. Çok seçenekli ön kurmalı sayıcı nedir?

11. Asenkron aşağı sayıcıyı tanımlayınız.

12. Üç bitlik asenkron aşağı sayıcının çalışmasını dalga şekilleri yardımı ile açıklayınız.

13. 110 değerinden başlayarak aşağı doğru sayma işlemi gerçekleştiren devreyi çizerek çalışmasını anlatınız.

14. 4 FF ve 'Özel-VEYA' kapıları kullanarak 0-15 arasında sayma yapabilen aşağı yukarı sayıcı devresi tasarlayarak, çalışma prensibini özetleyiniz.

15. Sayıcılarda kullanılan 'Mod' terimini açıklayınız.

16. Sayılarda dolaylı sıfırlama yöntemini tanımlayınız.

17. Dolaylı sıfırlama yöntemi kullanılan bir sayıcı tasarımında takip edilecek işlemleri sıralayınız.

18. Mod-5 sayıcısını dolaylı sıfırlama yöntemi kullanarak tasarımlayınız.

19. Mod-12 sayıcısını dolaylı sıfırlama yöntemi kullanarak tasarımlayınız.

20. Sayıcılarda doğrudan sıfırlama yönteminin genel prensipleri nelerdir?

21. Doğrudan sıfırlamalı Mod-10 sayıcısını tasarlayınız.

22. Doğrudan sıfırlamalı Mod-12 sayıcısını tasarımlayınız.

23. İlk FF'nin tetikleme girişine uygulanan sinyalin frekansını 25'e bölen asenkron sayıcı devresini tasarımlayınız.

24. 'Senkron sayıcı' terimindeki 'senkron' kelimesi hangi anlamda kullanılmıştır.

25. İki bit senkron yukarı sayıcı devresini şekille birlikte açıklayınız.

26. Dört bitlik senkron yukarı sayıcı devresinin lojik şemasını çizerek çalışma prensibini açıklayınız.

27. BCD senkron sayıcı devresinin şeklini çizerek, çalışmasını dalga şekli yardımı ile açıklayınız.

28. Senkron aşağı sayıcı temel prensibini açıklayınız.

29. Dört bitlik senkron aşağı sayıcı devresini lojik şemasını çizerek, çalışmasını özetleyiniz.

30. Üç bitlik senkron yukarı –aşağı sayıcı devresini şekille anlatınız.

31. Senkron sayıcı tasarım aşamalarını sıralayınız.

32. Mod-9 senkron sayıcı tasarımını yapınız.

33. Mod-12 senkron sayıcı tasarımını yapınız.

34. 1, 2, 4, 5 sayma dizisini sayan senkron sayıcı devresini tasarlayınız.

35. 5, 3, 2, 1 sayma dizisini tekrarlayan senkron sayıcı devresini tasarlayınız.

36. 'Sayıcının yüklenmesi' terimini açıklayınız.

37. Sayıcı entegrelerinde genelde bulunan pinleri şema üzerinde gösteriniz ve pinlerin anlamlarını özetleyiniz.

38. Sayıcı entegrelerinde bulunan terminal sayma çıkışlarının amacını ve çalışma şekillerini açıklayınız.

39. Piyasada bulunan asenkron sayıcı entegrelerine örnekler veriniz.

40. 74193 asenkron sayıcı entegre sembolünü çiziniz.

41. 74193 asenkron sayıcı entegresi iç yapısını şekille açıklayınız.

42. 74193 entegresi ile Mod-25 sayıcı devresini oluşturunuz.

43. Girişine uygulanan frekansı 50'ye bölen sayıcı devresini 74193 ile oluşturunuz.

44. 99’a kadar sayan kaskat sayıcının şemasını çizerek çalışma prensibini özetleyiniz.

45. 9999’a kadar sayan sayıcıyı BCD sayıcılar ile oluşturunuz.

46. ‘Ring sayıcıyı’ tanımlayınız.

47. 7 FF kullanan ring sayıcı devresini çizerek, çalışma prensibini özetleyiniz.

48. ‘Johnson sayıcı’yı tanımlayınız.

49. Mod-6 Johnson sayıcı şemasını çiziniz.

50. Ring ve Johnson sayıcılar arasındaki farkı özetleyiniz.

BÖLÜM 12

KAYDEDİCİLER

(REGISTERS)

Giriş

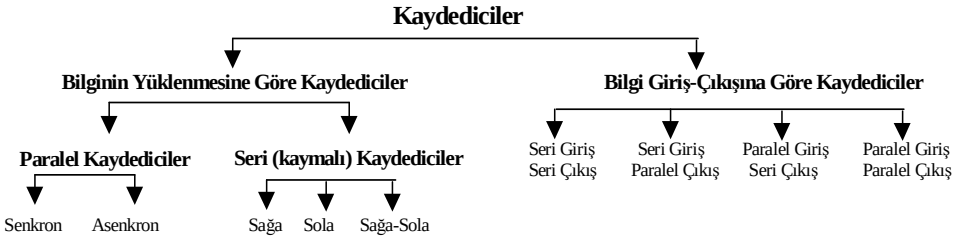
Lojik kapılar ve FF'lerden oluşan, ikili bilgileri geçici olarak saklamak için kullanılan devreler, **'kaydediciler'** (registers) olarak adlandırılır. Kaydedicilerin sayıcılardan tek farkı; kaydedicilerin sayıcılarda olduğu gibi belirli bir sayma dizisini devamlı tekrarlamamasıdır. Kaydediciler; bilgisayarlarda bilgi depolama, ikili toplayıcı / çıkarıcı devrelerde bilgi tutma ve bilgi transferi gibi işlemlerde, sayıcılarda bellek birimlerinde, vb. yerlerde kullanılırlar.

Kaydediciler, bilginin yüklenişine ve bilgi-giriş çıkış şekline göre sınıflandırılabilirler (Şekil 12.1).

Kaydedicilerin bilginin yüklenişine göre sınıflandırılmasında; tüm bilgilerin FF'lere aynı anda yüklendiği kaydediciler **'paralel kaydediciler'** (parallel registers) olarak isimlendirilirken, bilgilerin tek-tek yüklendiği ve kaydırıldığı kaydediciler **'seri kaydediciler'** (serial registers) veya **'kaymalı kaydediciler'** (shift registers) olarak isimlendirilir.

Kaydediciler bilgi giriş-çıkış şekline göre sınıflandırılırken; bilginin seri veya paralel olarak giriş ve çıkışlarda oluşabileceği referans alınır. Bu referansa göre 4 farklı durum oluşur: seri giriş-seri çıkış, seri giriş-paralel çıkış, paralel giriş-seri çıkış, paralel giriş-paralel çıkış.

Daha farklı özelliklere göre de sınıflandırılabilecek kaydediciler, bilgi yüklenmesi ve bilgi giriş-çıkış şekli referans alınarak Şekil 12.1'deki gruplara ayrılabilirler.



Şekil 12.1. Kaydedicilerin sınıflandırılması.

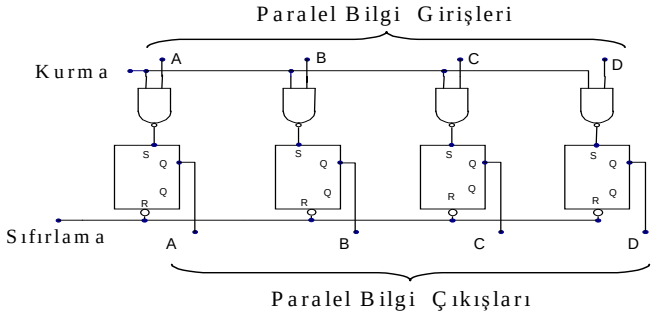
12.1. Kaydedicilerin bilginin yüklenmesine göre sınıflandırılması

Kaydediciler, bilginin yüklenmesi referans alınarak paralel kaydediciler ve seri kaydediciler şeklinde alt sınıflara ayrılırlar.

12.1.1. Paralel Kaydediciler

Bütün bilgilerin aynı anda FF'lere yüklendiği kaydediciler, **'paralel kaydediciler'** olarak adlandırılır. Paralel kaydedicilere bilgi yükleme işlemi, senkron veya asenkron olarak gerçekleştirilir. Tetikleme girişi kullanılmadan, bilgilerin kurma girişi yardımı ile yüklendiği kaydedici devreleri **'asenkron paralel kaydedici'** olarak isimlendirilirken, bilgilerin yüklenmesi için tetikleme girişlerinin kullanıldığı devreler **'senkron paralel kaydedici'** olarak adlandırılır.

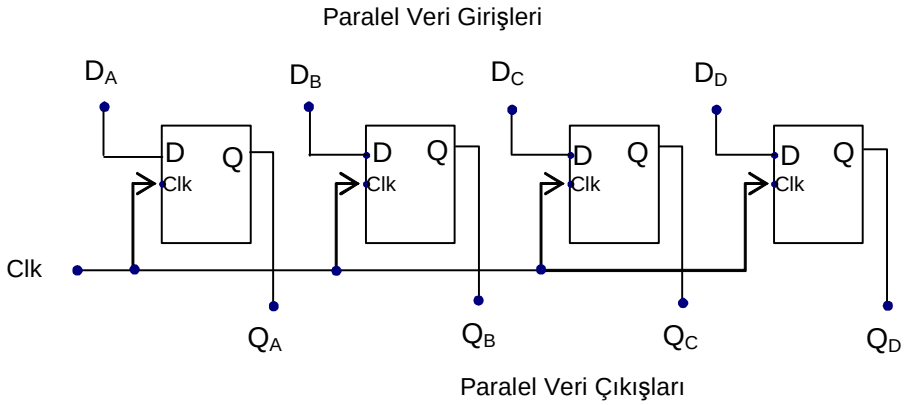
Yeni bilgi girişi için tetikleme girişinin kullanılmadığı asenkron paralel kaydedici devresinde, her yeni bilginin yüklenmesinden önce FF'lerin sıfırlanması gerekir (Şekil 12.2).



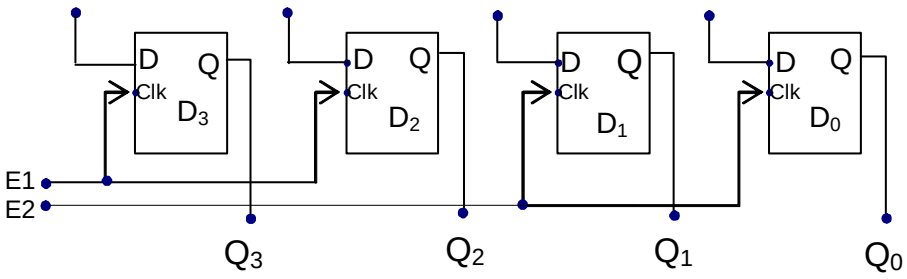
Şekil 12.2. Asenkron paralel kaydedici prensip şeması.

Bilgi yüklenmeden önce yapılması gerekli sıfırlama işlemi, sıfırlama girişinin (R) aktif yapılmasıyla gerçekleştirilir. FF'lerin enerjilenmesinden sonra, yüklenecek bilgiler bilgi girişleri olarak kullanılan 'VEDEĞİL' kapılarına uygulanır. Kurma girişinin '1' yapılması ile istenen bilgiler kaydediciye aktarılır. Bunun anlamı; bilgi girişlerine uygulanan bilginin kaydediciye yüklenmesini istediğimiz anda, kurma girişinin '1' yapılmasının yeterli olmasıdır. Kaydediciye yüklenen bilgiler, bir sonraki sıfırlama işlemine kadar FF'lerde saklanır.

Giriş bilgilerinin kaydedicilere yüklenmesi için gerekli tetikleme sinyallerinin aynı anda-aynı kaynaktan uygulandığı paralel kaydediciler, '**senkron paralel kaydediciler**' olarak isimlendirilir. D tipi FF'lerin kullanıldığı bu tip kaydedicilerde, 'Clk' girişinin '1' yapılması ile girişlerden uygulanan bilgiler FF'lere yüklenebilir ve çıkıştan alınabilir. 'Clk' girişinin '0' yapılması durumunda ise, girişten uygulanan bilgi çıkıştan alınamaz (Şekil 12.3).



Şekil 12.3. Senkron paralel kaydedici devresi.



Şekil 12.4. 74LS75 Senkron paralel kaydedici entegresi prensip şeması.

Senkron paralel kaydedicilere örnek olarak 74LS75 4-bit kaydedici entegresi verilebilir. Yapısında D tipi FF'lerin kullanıldığı bu entegre, iki adet iki bit veya bir adet dört bit kaydedici olarak kullanılabilir (Şekil 12.4). Entegrede, **'enable-E'** girişlerin beraber bağlanması ile bütün FF'lerin aynı anda çalışmaya hazır hale gelmesi sağlanır.

12.1.2. Seri (Kaymalı) Kaydediciler

Bilgilerin kaydedici devresine seri olarak yüklendiği ve her tetikleme sinyali ile önceki bilgilerin sağa veya sola kaydırıldığı devre, **'seri kaydedici'** veya **'kaymalı kaydedici'** olarak adlandırılır. Kayma işlemi uygulanan tetikleme sinyali ile gerçekleştirildiği için, kaydedicilerde kullanılan tetikleme sinyali **'kaydırma sinyali'** olarak da adlandırılabilir. Bilginin seri olarak gönderilmesi tercih edilen yerlerde, kaymalı kaydedici (shift register) devrelerden faydalanılır.

Seri kaydedicilerde, ilk FF haricindeki FF'lerin çalışma konumu bir önceki FF'nin çalışmasına göre belirlenir. Diğer bir ifade ile, her tetikleme palsı ile bir önceki FF'deki bilgiler sonraki FF'lere aktarılır.

Girişte bulunan seri bilginin tetikleme sinyali ile senkronizeli olarak çıkışa aktarıldığı kaydedicilerde istenen sayıda FF kullanılabilir. Kullanılacak FF sayısına uygun entegreler seçilerek, oluşturulmak istenen devre gerçekleştirilir.

Kaymalı kaydediciler, bilginin kaydırılma yönüne göre isimlendirilirler. Kaydediciler bilgi kaydırılma yönüne göre üç grup altında incelenebilir:

i- Sağa kaymalı kaydediciler,

ii- Sola kaymalı kaydediciler,

iii- Sağa-sola kaymalı kaydediciler.

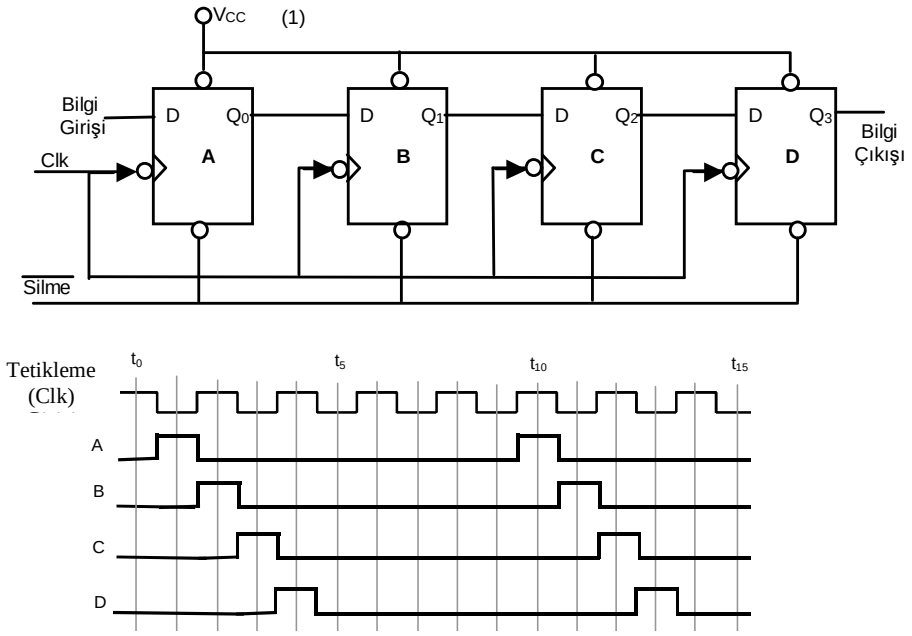
12.1.2.1. Sağa Kaymalı Kaydediciler

Bilgilerin seri olarak yüklendiği ve yüklenen bilgilerin kaydırma sinyalleri ile FF'lerde sağa doğru hareket ettiği kaydedici devreleri, **'sağa kaymalı kaydedici'** olarak isimlendirilir.

D tipi FF'lerle oluşturulan kaymalı kaydedici devresinde, ilk tetikleme sinyali ile A FF'ine '1' değeri yüklenir (Şekil 12.5). Bir sonraki tetikleme sinyali ile, B FF'ine A FF'indeki '1' değeri aktarılır.

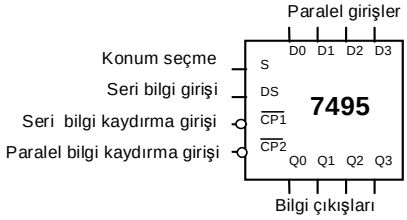
C FF'i 't_3' anındaki tetikleme sinyali ile, D FF'i ise 't_4' anındaki tetikleme sinyali ile '1' konumuna kurulurlar. t_2, t_3, t_4 anlarında giriş bilgisi '0' olduğundan ve 4 adet FF kullanıldığından, dört tetikleme palsı bilgilerin baştan sona kaydırılmasını sağlar. t_6 anında tüm FF'lerin çıkışı 0'a kurulmuş olur.

Pratikte kullanılan kaydedici entegreler TTL 7495 entegresi gibi hem seri, hem de paralel girişe sahip olabilir. Bu kaydedicilerde seri yada paralel girişten girilecek herhangi bir bilgi çıkıştan alınabilir.

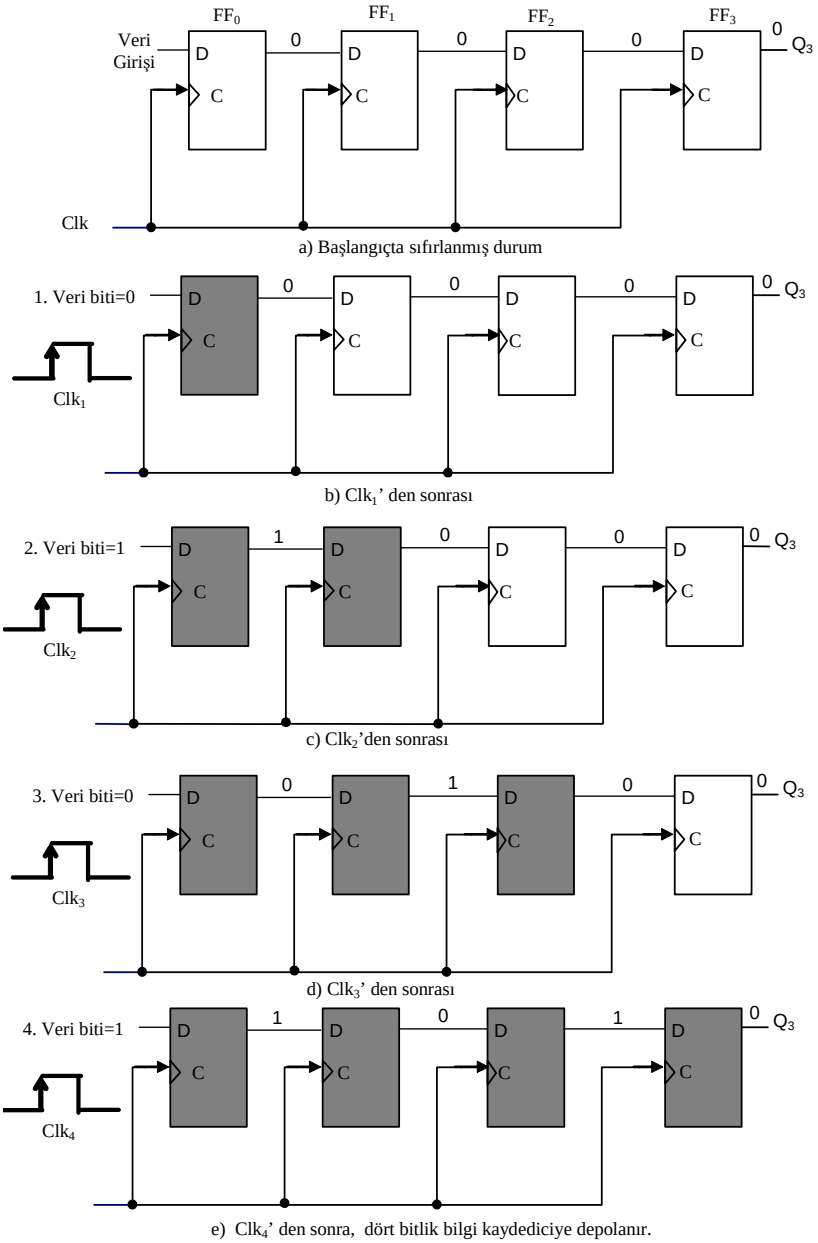


Şekil 12.5. 7474 Entegreleri ile oluşturulan 4-bit kaymalı kaydedici ve zaman diyagramları.

Şekil 12.6'da lojik sembolü görülen 7495 entegresinde, seri bilgi girişi için CP_1, paralel bilgi girişi için ise CP_2 kontrol girişi kullanılır. S (Select – seçme) girişinin '1' yapılmasıyla CP_2 kullanılırken, S girişinin '0' yapılması ile CP_1 aktif hale getirilir. CP_1'in aktif yapılmasıyla, ilk gelen tetikleme palsı ile birlikte Ds girişindeki (seri bilgi girişi) bilgi Q_0 çıkışına aktarılır. Daha sonraki tetikleme palsleri ile, bilgi Q_0'dan Q_1'e, Q_1'den Q_2'ye, Q_2'den Q_3'e aktarılır. Sonuçta, girişten uygulanan bilgi sağa doğru kaydırılmış olur.



Şekil 12.6. 7495 Entegresinin lojik sembolü.



Şekil 12.7. Sağa kaymalı kaydediciye bilgi yüklenmesi.

Örnek 1: Şekil 12.7.a'da görülen 4 bit kaymalı kaydedici devresine $(1010)_2$ bilgisinin yüklenmesi sırasında oluşan olayları inceleyelim:

Devrede 4 FF bulunması nedeni ile, devre 4 bitlik bilgi saklama kapasitesine sahiptir (Şekil 12.7.a). İlk anda tüm FF'lerın çıkışının '0' olduğu kabul edilen devreye '1010' bilgisinin yüklenmesi işlemi, en sağdaki bitten başlanarak yapılır. En sağdaki bitin '0' olması nedeni ile, FF_0'ın D girişine '0' uygulanır. İlk gelen tetikleme (kaydırma) sinyali ile '0' bilgisi FF_0'a yüklenir (Şekil 12.7.b).

Sağdan ikinci bit '1' olduğundan, FF_0'ın D girişine uygulanması gereken yeni değer 1'dir. FF_0'a '1' uygulanması durumunda; ilk gelen kaydırma sinyali ile birlikte FF_0=1 değerini alırken, Q_0 çıkışı FF_1'in D girişine uygulandığından FF_1= 0 olur (Şekil 12.7.c).

Üçüncü bitin '0' olması nedeni ile, FF_0 girişine '0' uygulanır ve tetikleme sinyali ile FF_0= 0 değerine kurulur. Bu anda; Q_0 çıkışı FF_1'in D girişine '1' değeri uyguladığından FF_1 = 1 ve Q_1 çıkışı FF_2'nin D girişine '0' uyguladığından FF_2=0 değerlerine sahip olur. Diğer bir değiş ile; FF_0'daki bilgi FF_1'e, FF_1'deki bilgi ise FF_2'ye aktarılır. Bu işlemler ile sağdan üç bitlik bilgi kaydedici devreye yüklenir (Şekil 12.7.d).

Yüklenecek bilgideki dördüncü bit '1' olduğundan, FF_0'ın D girişine '1' uygulanır. Kaydırma sinyali ile FF_2'deki bilgi FF_3'e kaydırılır. Sonuçta, FF_0 = 1, FF_1 = 0, FF_2 = 1 ve FF_3 = 0 değerlerine sahip olur. (Şekil 12.7.e.). Bunun anlamı; dördüncü bit sonunda 4 bitlik sayının kaydırıcı devresine yüklenmesidir.

4 bitlik sağa kaymalı kaydedici devresindeki bilgileri çıkıştan almak istiyorsak; 4 kaydırma palsını FF_0'a uygulamamız ve FF_3'ün Q_3 çıkışındaki bilgiler her kaydırma palsı sonunda okumamız gereklidir. İlk tetikleme palsı ile en sağdaki biti temsil eden değer Q_3'te görülürken, dördüncü pals ile en soldaki değer Q_3'de gözükür.

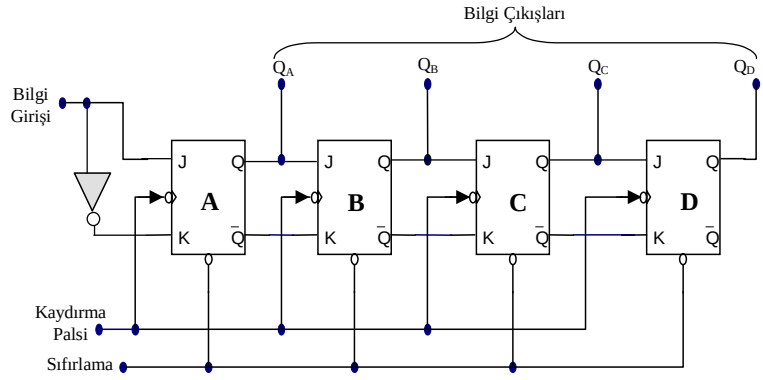
Örnek 2: 4 FF'ten oluşan sağa kaymalı kaydedici devresine $(1000)_2$ bilgisinin yüklenmesi sırasında oluşan olayları inceleyelim.

Başlangıçta tüm FF'lerin '0' olduğu kabul edilirse, ilk gelen tetikleme palsı ile A FF'ine '1' değeri yüklenir. Bu anda diğer çıkışlar '0'dır.

İkinci tetikleme palsı ile A FF'ine '0' yüklenirken, A'da bulunan '1' bilgisi B'ye aktarılır ve Q_B=1 konumunu alır.

Üçüncü tetikleme palsı ile yüklenen bilgi '0' olduğundan, Q_A=Q_B=Q_D=0 ve Q_C=1 olur.

Dördüncü pals ile C'deki bilgi D FF'ine kaydırılarak Q_D=1 olurken, diğer FF'ler '0' değerini alır. Bu anlatılanlar tablo haline getirilirse, Şekil 12.8.b'deki tablo oluşur.



(a)

		MBS			LBS	
Bilgi girişi	**Kaydırma palsı**	**A**	**B**	**C**	**D**	**Onlu Değer**
X	Reset	0	0	0	0	0
1	1	1	0	0	0	8
0	2	0	1	0	0	4
0	3	0	0	1	0	2
0	4	0	0	0	1	1

(b)

Şekil 12.8. Sağa kaymalı kaydedici devresi ve $(1000)_2$ bilgisinin yüklenmesi sırasında oluşan çıkışlar.

Örnek 3: 3 adet RS FF'ten oluşan sağa kaymalı kaydedici devresine '0110' bilgisini uygulayarak, 5. kaydırma sinyali sonucunda devredeki FF'lerde bulunan değerleri inceleyelim.

İlk anda tüm FF'ler '0' konumdadır. Bu konumdan başlayarak oluşan olayları açıklayıp tablo halinde özetleyelim.

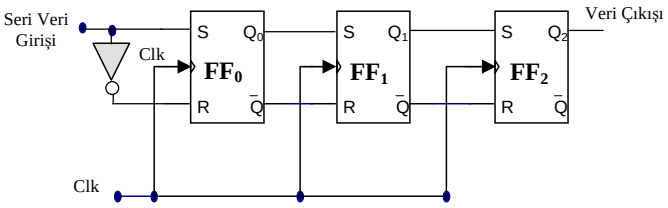
RS FF'lerin D tipi FF gibi davranabilmesi için; S ve R girişlerinin 'DEĞİL' kapısı ile birleştirilmesi gerekir (Şekil 12.9.a). 'DEĞİL' kapısı ile, her iki girişe aynı değerin uygulanması engellenir ve RS FF'in S girişindeki değeri takip etmesi sağlanır.

FF'lere uygulanan ilk değer '0' dır. Birinci kaydırma palsı ile tüm FF'ler '0' durumunu korurken, ikinci pals ile FF_0 çıkışı '1' değerine kurulur.

Üçüncü kaydırma palsı ile FF_0'a 1 yüklenirken, diğer FF'ler FF_1= 1 ve FF_1= 0 değerlerine sahip olur.

Dördüncü kaydırma sinyali ile FF_0'a '0' değeri yüklenirken, FF_0'daki '1' değeri FF_1'e, FF_1'deki '1' değeri de FF_2'ye kaydırılır.

Beşinci kaydırma sinyali ile FF_0= 0 değerini alırken, FF_1= 0 ve FF_2= 1 değerlerine yüklenir (Şekil 12.9.b).



(a)

Kaydırma Palsı	Seri Veri Girişi	FF_0	FF_1	F
-	-	0	0	(
1	0	0	0	(
2	1	1	0	(
3	1	1	1	(
4	0	0	1	1
5	0	0	0	1

(b)

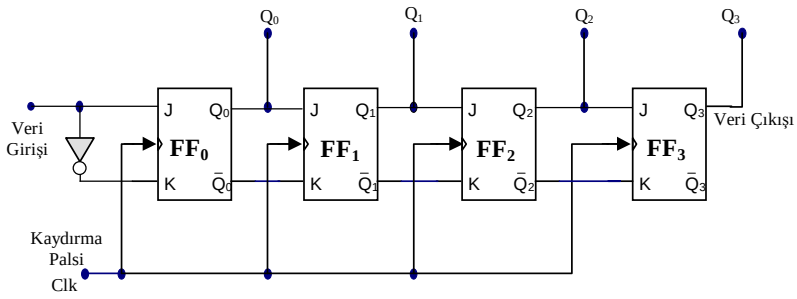
Şekil 12.9. Bilgilerin seri olarak sağa kaymalı kaydediciye yüklenmesi.

Örnek 4: J-K FF'ler ile oluşturulan 4 bitlik sağa kaymalı kaydedici devresini çizerek, çizilen devreye '1001' değerlerinin uygulanması durumunda 6. kaydırma palsı sonucunda FF'lerde oluşacak değerleri tablo halinde gösterelim.

JK FF'lerin çalışma prensibi; JK=00 veya JK=11 konumlarında kaydedici olarak kullanımını zorlaştırır. Bu nedenle, J-K girişleri bir DEĞİL kapısı ile birleştirilirse; JK=01 veya JK=10 olarak 2 durum söz konusu olur. JK FF, bu konumlarda D tipi FF gibi çalışarak; J=0 ise Q=0 ve J=1 ise Q=1 olacak şekilde bir çalışma gösterir (Şekil 12.10.a).

Devreye yüklenmesi gereken bilgiler sırası ile FF_0'ın veri girişine uygulanırsa, dördüncü pals sonucunda FF'lerde; FF_0= 1, FF_1= 0, FF_2= 0 ve FF_3= 1 değerleri oluşur.

Kaydediciye yüklenen değerler, her kaydırma sinyali ile sağa kayar ve altıncı pals sonucunda Şekil 12.10.b'deki değerler ortaya çıkar.



(a)

Kaydırma Palsı	Seri Veri Girişi	FF_0	FF_1	FF_2	FF_3
-	-	0	0	0	0
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	1	1	0	0	1
5	0	0	1	0	0
6	0	0	0	1	0

(b)

Şekil 12.10. Dört bitlik kaymalı kaydediciye bilgi yüklenmesi.

12.1.2.2. Sola Kaymalı Kaydediciler

Bilginin sola kaydırılmasına karşılık gelen işlemi gerçekleştiren seri kaydedici devresi, **'sola kaymalı kaydedici'** olarak isimlendirilir. Uygulanan bilginin sola doğru kaydırılarak FF'lere aktarıldığı sola kaymalı kaydediciler, D veya J-K tipi FF'ler kullanılarak gerçekleştirilebilir.

Şekil 12.11'de, J-K FF'lerle oluşturulmuş sola kaymalı kaydedici devresi görülmektedir. Bu devreye $(1000)_2$ bilgisinin yüklenmesi işlemini inceleyelim.

İlk gelen tetikleme sinyali ile, ilk bilgi olan '1' değeri en küçük değerli biti temsil eden D FF'una yüklenir ve Q_D=1 olur. Bu anda diğer FF çıkışları '0' durumundadır.

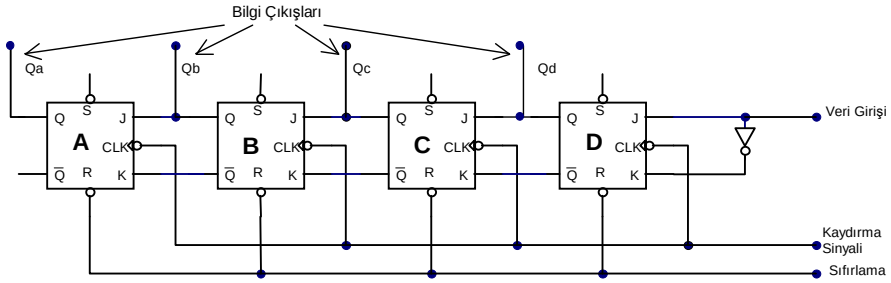
İkinci tetikleme sinyali ile girişten uygulanan bilgi '0' olduğundan Q_D=0 olurken, D FF'indeki bilgi C'ye aktarılır ve Q_C=1 değerini alır.

Üçüncü gelen tetikleme palsı ile Q_B=1 olurken, Q_D ve Q_C FF'leri '0' değerini alır.

Dördüncü tetikleme sinyali ile D'ye yüklenen yeni bilgi '0' olduğundan, Q_D=0 değerini alır. Bu anda Q_A=1 olurken, diğer tüm çıkışlar '0' değerine sahiptir.

Devrenin tetikleme palsleri ile aldığı sonuçlar tablo haline getirilirse, Şekil 12.11.b'deki tablo oluşur.

Bir bilginin sola kaymalı kaydedicilere yüklenmesi, FF sayısı ve yüklenecek bilginin basamak sayısına bağlıdır. Bilginin tümü FF'lere yüklendikten sonra, gelecek her tetikleme palsı ile sola kayma işlemi devam eder ve yeni değerler '0' olarak kabul edilir.



(a)

Bilgi Giriş	Kaydırma Palsı	MSB A	B	C	LSB D	Onlu değer
		0	0	0	0	
1	1	0	0	0	1	1
0	2	0	0	1	0	2
0	3	0	1	0	0	4
0	4	1	0	0	0	8

(b)

Şekil 12.11. JK FF'lerle sola kaymalı kaydedici devresi ve $(1000)_2$ bilgisinin yüklenmesi işleminin tablo olarak gösterimi.

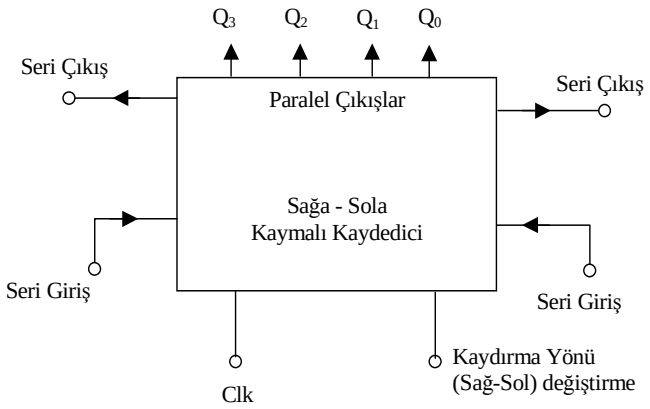
12.1.2.3. Sağa - Sola Kaymalı Kaydediciler

Normalde bilginin sağa veya sola sadece bir yönde kaydırıldığı kaydedicilerin yanında, bazı durumlarda bilginin sağa veya sola kaydırılması istenebilir. Kaymanın yönünün, eklenen kayma yönü girişi ile belirlendiği sağa-sola kaydırmalı kaydedici devresi, 74194 gibi elemanlarla gerçekleştirilebilir.

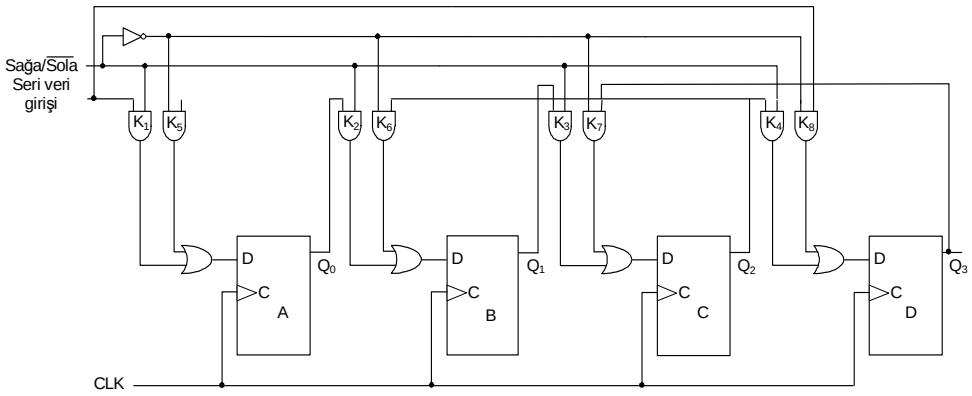
Çarpma ve bölme işlemleri, kaydedicilerdeki bilginin sağa veya sola kaydırılması ile gerçekleştirilebilir. Örneğin; $(0111)_2 = (7)_{10}$ sayısı bir sola kaydırılırsa, bu sayının iki katı olan $(1110)_2 = (14)_{10}$ sayısı elde edilir. Bu durumda, herhangi bir sayının sola bir basamak kaydırılması sayının 2 ile çarpılması demektir. $(1000)_2 = (8)_{10}$ sayısının bir sağa kaydırılması ile, $(0100)_2 = (4)_{10}$ sayısı elde edilir ve bu işlem sayının 2'ye bölünmesine karşılık gelir.

Bu işlemleri gerçekleştirmek için RS, JK ve D tipi FF'lerin kullanıldığı kaydedicilerde, kullanılan FF'lerin kurma ve silme girişlerinin- bulunması gerekir.

Sağa - sola kaymalı kaydedici devresi blok şeması, Şekil 12.12.a'da görülmektedir. Kaydırma yönü '1' iken, seri girişten uygulanan bilgi sağa doğru yer değiştirir. Kaydırma yönü girişine uygulanan '0' bilgisi ile, seri bilgi girişine uygulanan bilgi sola doğru kaydırılır.



(a)



(b)

Şekil 12.12. Sağa - sola kaymalı kaydedici blok şeması ve lojik devresi

Entegreye, bilginin kaydırılma yönünü belirleyen sinyal ile birlikte, veri girişinin uygun olan taraftan uygulanması gerektiği unutulmamalıdır.

Blok şeması verilen sağa / sola kaydırma işlemi, Şekil 12.12.b'deki lojik devre ile gerçekleştirilebilir. Bilgi kaydırma yönüne, kontrol girişi olarak düşünülebilecek 'sağa / sola' kaydırma girişinin değerine göre karar verilir.

Sağa / sola kaydırma yönü seçici girişinin '1' yapılması ile, sağa kaydırma işlemi gerçekleştirilir. Kontrol girişinin '1' olması seri veri girişinin K_1 kapısının çıkışında gözükmesini ve bilginin A FF'sine uygulanmasını sağlar. A FF çıkışındaki bilgi, K_2 üzerinden B FF'sine aktarılır ve bu şekilde bilgi D FF'sine doğru kaydırılır. Bu işlem, sağa kaydırma işlemine karşılık gelir.

Kaydırma yönü girişinin '0' yapılması ile, seri veri girişi K_8 kapısı üzerinden D FF'sine uygulanır. D FF'i çıkışındaki bilgi, bir sonraki kaydırma palsı ile K_7 üzerinden C FF'sine uygulanır. C FF'sindeki bilgi K_6 üzerinden B FF'sine, B FF'sindeki bilgide K_5 üzerinden A FF'sine aktarılır. Anlatılan olaylar, sola kaydırma işlemini ortaya çıkarır.

Sağa kaydırma işleminde çıkış Q_3'den alınırken, sola kaydırma işleminde Q_0'dan alınır.

12.2. Bilgi Giriş - Çıkışına Göre Kaydedicilerin Sınıflandırılması

Farklı tip kaydediciler, bilginin kaydediciye yüklenmesi ve bilginin çıkışlardan alınması şekline göre gruplandırılabilirler. Bu gruplandırmada dört farklı kaydedici tipi ortaya çıkar:

i- Seri giriş - seri çıkışlı kaydediciler,

ii- Seri giriş - paralel çıkışlı kaydediciler,

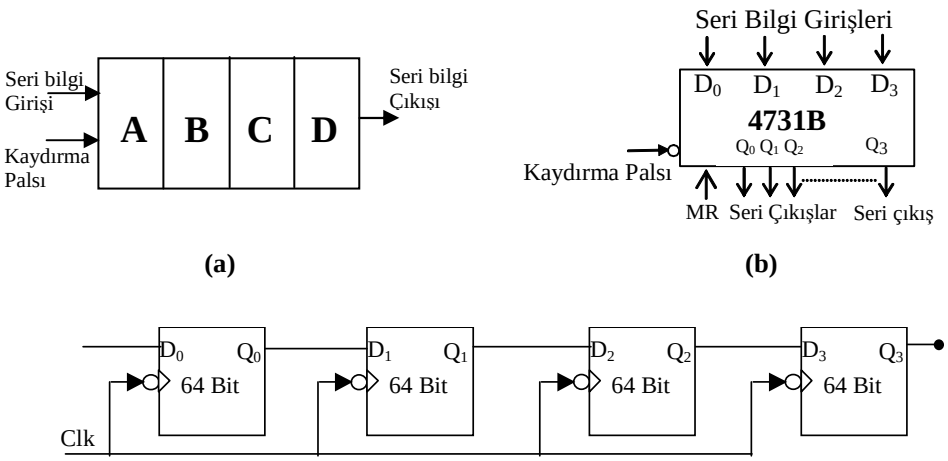
iii- Paralel giriş - seri çıkışlı kaydediciler,

iv- Paralel giriş - paralel çıkışlı kaydediciler.

12.2.1. Seri Giriş – Seri Çıkışlı Kaymalı Kaydediciler

Sağa veya sola kaymalı kaydedicilerde, ilk FF'nin veri girişinden uygulanan bilgi son FF'nin çıkışından alınırsa, **'seri giriş - seri çıkışlı'** kaymalı kaydedici elde edilir. Şekil 12.5 ve 12.7'de anlatılan olaylar, seri giriş ve seri çıkışın kullanıldığı kaymalı kaydedici sistemine örnek olarak verilebilir. Bilgi transferinde ortaya çıkan gruplar, Şekil 12.12'deki prensip şema ile anlatılabilir.

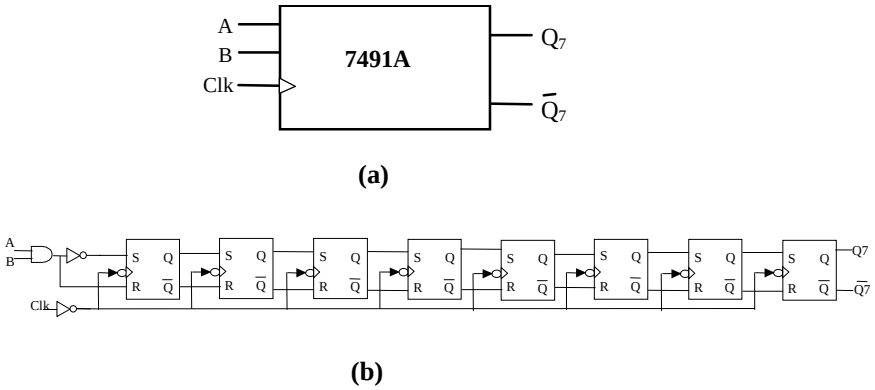
Şekil 12.13.a'da bilginin seri olarak A'dan girilip, çıkışın yine seri olarak D'den alındığı seri giriş - seri çıkışlı kaymalı kaydedici devresi görülmektedir. CMOS 4731B entegresi, seri giriş – seri çıkış işlemini gerçekleştiren bir elemandır (Şekil 12.13.b). 4731 entegresinde 4 adet 64 bit kaymalı kaydedici bulunmaktadır. Bu entegredeki kaydediciler bağımsız şekilde 64 biti saklayan kaydedici olarak kullanılabileceği gibi, 4 tanesi seri bağlanarak 256 bit kaydedici olarak kullanılabilir (Şekil 12.13.c).



Şekil 12.13. Kaydedicide seri giriş - seri çıkış bilgi transferi ve 4731B entegresinin 256-bit kaydedici olarak kullanılması.

Bilgilerin seri olarak girilip seri olarak çıkıştan alındığı diğer bir entegre, 7491A 8 bit kaymalı kaydırıcı entegresidir (Şekil 12.14.a). Yapısında R-S FF'lerin kullanıldığı bu entegrenin iç yapısı Şekil 12.14.b'de görülmektedir.

7491 entegresinde iki adet veri girişi bulunmaktadır (A ve B). Veri girişlerden birinden uygulanır iken, diğer girişin '1' değerine sahip olması gerekir. Entegreden veri çıkışı, Q_7 veya tersi olan Q_7 çıkışlarının birisinden alınabilir.



Şekil 12.14. 7491A entegresi lojik sembolü ve iç yapısı.

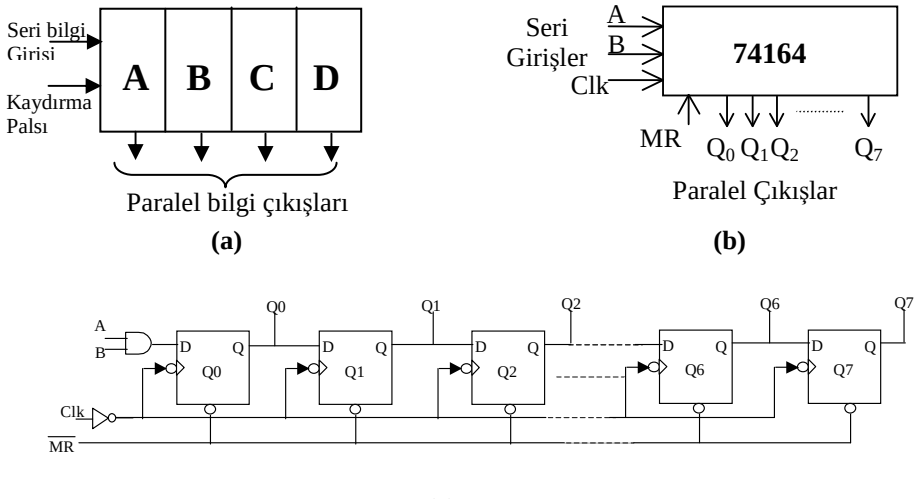
12.2.2. Seri Giriş - Paralel Çıkışlı Kaymalı Kaydediciler

Bilginin A FF'inden seri olarak girilip, çıkışların A, B, C, D FF'lerinden aynı anda paralel olarak alındığı devre, '**seri giriş – paralel çıkışlı'** kaymalı kaydedici olarak isimlendirilir (Şekil 12.15.a). Bu yapıdaki devre ile, tek-tek girilen bilgilere istenilen anda ulaşmak ve FF'lerde bulunan bilgileri birlikte (aynı anda) okumak mümkündür. Seri giriş – paralel çıkışlı kaymalı kaydedici devresi, bir hattan gelen bilgileri birden çok hatta dağıtma işlemini de gördüğünden demultiplexere benzetilebilir.

74164 entegresi, her bir FF çıkışına harici olarak ulaşılabilen 8 bit seri giriş – paralel çıkış işlemi sağlayan kaymalı kaydedici entegresidir (Şekil 12.15.b).

Açık şeması Şekil 12.15.c'de görülen 74164 entegresinde bulunan A ve B girişlerine sahip 'VE' kapısı, tek bir seri giriş yerine A ve B girişlerinin seri olarak kullanılmasını sağlar. Tek bir giriş kullanılması durumunda, ikinci girişin '1' değerine bağlanması gerekir.

Entegredeki MR girişi, tüm FF'lerin aynı anda senkron olarak sıfırlanmasını sağlamak için kullanılır.



Şekil 12.15. Seri giriş - paralel çıkış bilgi transferi ve 74164 seri giriş - paralel çıkışlı 8-bit kaymalı kaydedici.

12.2.3. Paralel Giriş - Seri Çıkışlı Kaymalı Kaydediciler

Giriş bilgilerinin paralel girişler yardımıyla aynı anda tüm FF'lere yüklendiği, buna karşılık çıkışların seri olarak tek-tek okunduğu kaymalı kaydedici, **'Paralel giriş – seri çıkışlı**

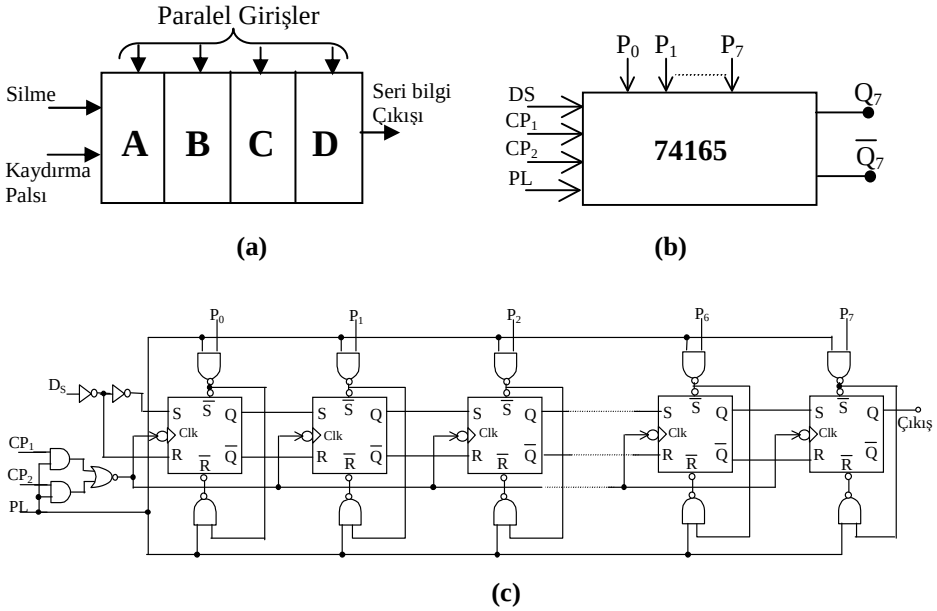
kaymalı kaydedici' olarak isimlendirilir. Bilgilerin doğrudan FF'lere yüklenebildiği bu devrelerde, çıkış olarak yalnızca en sondaki FF'in çıkışına erişilebilir (Şekil 12.16.a).

Paralel giriş - seri çıkışlı kaymalı kaydedici devresi, birden fazla girişteki bilgileri tek bir hatta belirli sıra ile vermesi nedeniyle multiplexer'e benzetebilir.

74165 entegresi, paralel giriş – seri çıkışlı 8 bit kaymalı kaydedici entegrelerine bir örnektir (Şekil 12.16.b). Bu entegrede yalnızca Q_7 FF'sinin çıkışına erişilebilir. Bu entegredeki her bir FF, bilgilerin paralel olarak yüklenmesini sağlayan girişlere (kurma ve silme) sahiptir (Şekil 12.16.c).

CP_1 ve CP_2 girişlerinden birisinin kayma işlemi tetikleme girişi olarak kullanabileceği 74165 entegresinde, Ds girişi ile seri bilgi girişi mümkündür.

P_L yükleme girişinin '1' yapılması ile, paralel girişlerdeki bilgilere bağlı olarak 'VEDEĞİL' kapıları iletime geçerek, duruma hazır hale gelir. FF'ler, 'VEDEĞİL' kapılarının çıkışının alacağı değere göre yüklenirler (sıfırlanır veya kurulurlar).



Şekil 12.16. Paralel giriş - seri çıkışlı kaymalı kaydedici ve 74165 seri giriş - paralel çıkışlı 8 bit kaymalı kaydedici iç yapısı.

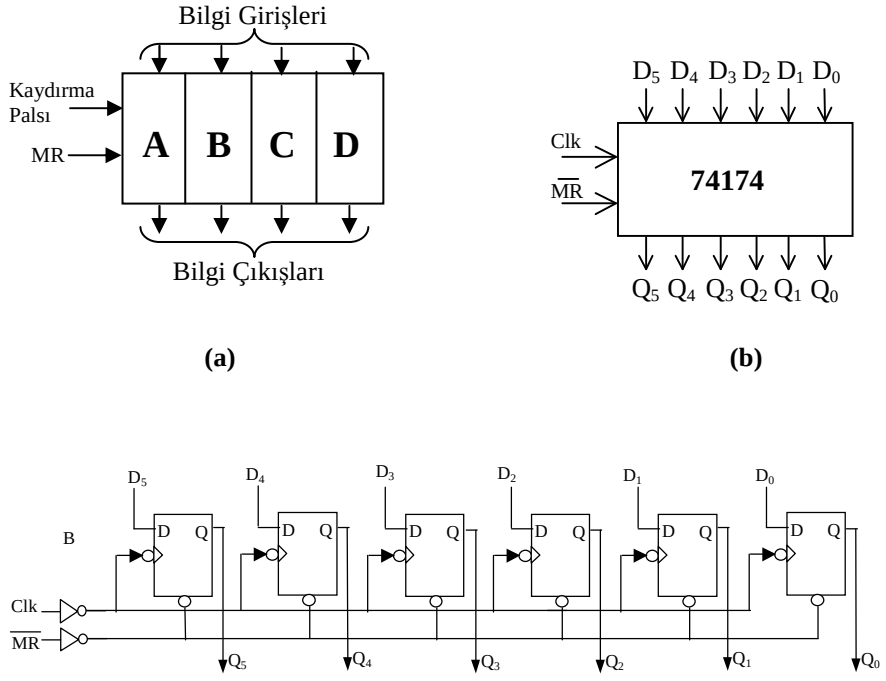
Paralel girişteki değer '1' ise, asenkron kurma (S') girişine bağlı olan 'VEDEĞİL' kapısı çıkışı '0' olur ve 'VEDEĞİL' kapısına bağlı bulunan FF '1' değerine kurulur. Paralel girişteki değer '0' ise; paralel girişe bağlı 'VEDEĞİL' kapısı çıkışı '1' değerini alır. '1'

değerine sahip çıkışın bağlı olduğu asenkron silme (R') girişine bağlı ‘VEDEĞİL’ kapısının her iki girişi ‘1’ ve buna bağlı olarak çıkışı ‘0’ olacağından, ilgili FF ‘sıfırlama’ işlemine tabi kalır ve ‘0’ olur. Bu şekilde, girişlerdeki bilgiler paralel olarak kaymalı kaydedici devresine yüklenir.

FF’lere yüklenen bilgiler, herbir kaydırma sinyali ile sağa doğru kaydırılır. Her kaydırma işlemi sonucunda çıkıştan bir bitlik bilgi okunur. Okunan bilgiler, FF’lere yüklenen bilgileri temsil ettiklerinden, paralel bilgiler seri olarak okunmuş olur. Bu şekildeki veri dönüşümü, bilgisayarda işlenen verilerin seri port (RS-232) yardımıyla iletilmesi için kullanılır.

12.2.4. Paralel Giriş - Paralel Çıkışlı Kaymalı Kaydediciler

Bilgilerin aynı anda paralel olarak yüklenmesini ve belirli işlemlerden sonra tüm çıkışlara aynı anda eriştirilmesini sağlayan kaydedici devreleri, **‘paralel giriş – paralel çıkışlı kaymalı kaydedici’** olarak isimlendirilir (Şekil 12.17.a).



Şekil 12.17. Paralel giriş - paralel çıkışlı kaydedici devresi, 74174 kaydedici entegresi sembolü ve açık şeması.

Kaymalı kaydedici, yalnızca paralel giriş – paralel çıkış özelliğine sahip olabileceği gibi (74174 entegresi), hem seri hem de paralel girişe ve paralel çıkışa sahip olabilir (74178 entegresi vb.). Şekil 12.17.b'de görülen 74174 entegresi ile, 6 bitlik bilgi aynı anda D_0 - D_5 girişlerinden devreye yüklenip, Q_0-Q_5 paralel çıkışlarından okunabilir. Kaydedicilerin bu kullanım şekli, bilgileri kısa süreli saklamak için bellek olarak kullanılmalarına uygun bir yapı oluşturur.

Paralel girişlerdeki bilgilerin yüklenmesi için 'Clk' girişi kullanılırken, tüm FF'lerin sıfırlanması için MR girişi kullanılır. Normalde çıkışlar paralel olarak okunmasına rağmen, uygun bağlantı ve yalnızca Q_5 çıkışının kullanılması ile bilgilerin çıkıştan seri olarak okunması mümkündür (Şekil 12.17.c).

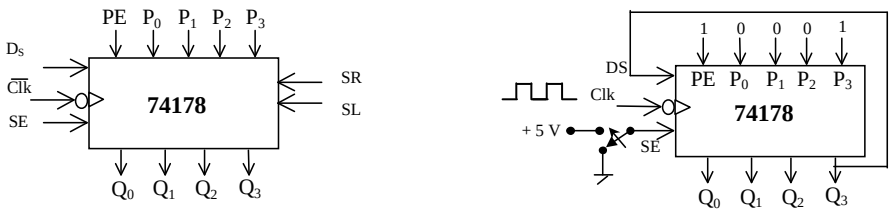
Paralel giriş - paralel çıkışa sahip diğer bir entegre 74195 entegresidir. 74195 entegresine bilgilerin seri olarakta yüklenmesi mümkündür.

12.2.5. İki Yönlü Kaymalı ve Farklı Giriş - Çıkış Kombinasyonlu Kaydediciler

Kaydedici entegreleri yapılan gruplandırmalardan yalnızca birisine giren işlemleri yapabileceği gibi, gruplandırmalardaki işlemlerden birden fazlasını gerçekleştirebilirler. **'Üniversal Kaydediciler'** olarak isimlendirilen bu elemanlarda, farklı fonksiyonlardan istenileni seçme işlemi eklenen yeni girişler ile sağlanır.

Bu tür kaydedicilerde, kayma yönü istenilen yönde ayarlanabileceği gibi, bilgi giriş - çıkış şekli de kullanılacak devreye göre şekillendirilebilir.

İki yönlü kaydırma işlemini gerçekleştiren entegrelere örnek olarak 74178 entegresi verilebilir.



(a) **(b)**

Şekil 12.18. 74178 entegresi lojik sembolü ve entegrenin halka sayıcı olarak kullanılması için gerekli bağlantı.

Hem seri, hem de paralel bilgi girişine imkan tanıyan 74178 kaydedici entegresi lojik sembolü, Şekil 12.18.a'da görülmektedir. PE (paralel enable - paralel yetki) ve SE (serial enable – seri yetki) girişleri ile bilgi giriş şekli seçilir. P_0 – P_3 girişlerinin paralel girişler olarak kullanıldığı entegrede, D_S girişi ile bilgilerin seri olarak yüklenmesi mümkündür. Çıkışlar ise paralel olarak Q_0 – Q_3'den alınır. SE=1 ve PE=0 olduğu anda seri bilgi girişi yetkilendirilirken, SE=0 ve PE=1 durumunda paralel yükleme gerçekleştirilir.
Entegredeki kaydırma işleminin yönü SR ve SL girişleri ile belirlenir. Seri bilgi girişinin SL (Shift Left) uygulaması ile bilgi sola doğru kaydırılırken, giriş bilgisinin SR'ye (Shift Rigth) uygulanması ile bilgi sağa doğru kaydırılır. Sola kaymada çıkış Q_0'dan alınırken, sağa kaymada Q_3'den alınır.

Bilgilerin paralel olarak yüklenebildiği ve seri bilgi girişi ile sağa kaydırma işleminin gerçekleştirildiği 74178 entegresi, halka sayıcı olarak kullanılabilir. Bu durumda, Q_3 çıkışının D_S girişine (seri giriş) bağlanması gerekir (Şekil 12.18.b). PE = 1, SE = 0 ve paralel girişler $(0001)_2$ değeriyle yüklendikten sonra, SE = 1 ve PE = 0 yapılması durumunda her gelen tetikleme palsı ile kaydedicideki bilgi bir bit kaydırılır.

12.3. Kaymalı Kaydedici Uygulamaları

Kaymalı kaydedicilerin farklı çalışma şekilleri bulunması nedeni ile, çok çeşitli uygulama alanlarında kullanılmaları mümkündür. Bunlardan birkaç tanesini inceleyelim.

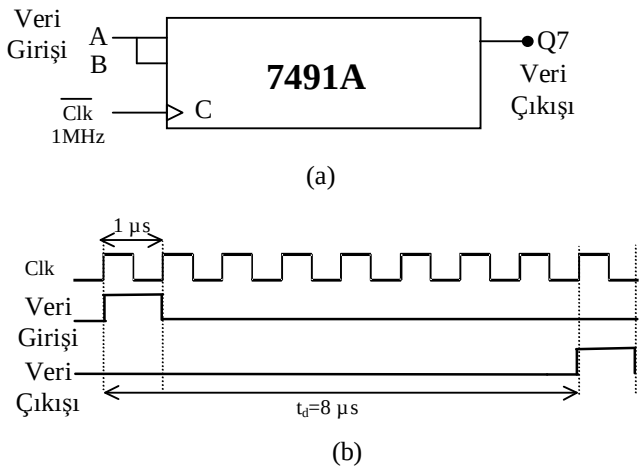
12.3.1. Zaman Gecikmesi

Seri giriş - seri çıkışlı kaymalı kaydediciler, kaydedicide kullanılan durum sayısı ve tetikleme sinyalinin fonksiyonu olarak üretilen zaman gecikmesi nedeni ile, zaman gecikmesi üretmek amacıyla kullanabilirler.

Seri giriş - seri çıkışa sahip 8 bitlik 7491A kaymalı kaydedici entegresi, Şekil 12.19.a'daki gibi bağlantı ile zaman gecikmesi üretici olarak kullanılabilir. A ve B seri girişlerinin birlikte kullanıldığı devreye, seri veri girişinin uygulanması durumunda, ilk gelen kaydırma sinyali ile veri ilk FF'ye yüklenir (Şekil 12.19.b).

Yüklenen bilgi, her bir kaydırma sinyali ile kaydırılarak, belirli bir süre sonucunda çıkış FF'sinde okunur. Devrede 1 MHz'lik kaydırma sinyali uygulanması durumunda, entegrede 8 FF bulunması nedeni ile, zaman gecikmesi 8x1 μsn = 8 μsn olarak oluşur.

Kaydırma sinyalinin frekansını değiştirmek sureti ile, oluşan zaman gecikmesinin süresi değiştirilebilir. Ayrıca kaymalı kaydedicileri kaskat bağlayarak, daha uzun zaman gecikmeleri elde etmek mümkündür.



Şekil 12.19. Kaymalı kaydedicinin zaman gecikmesi üreteci olarak kullanılması.

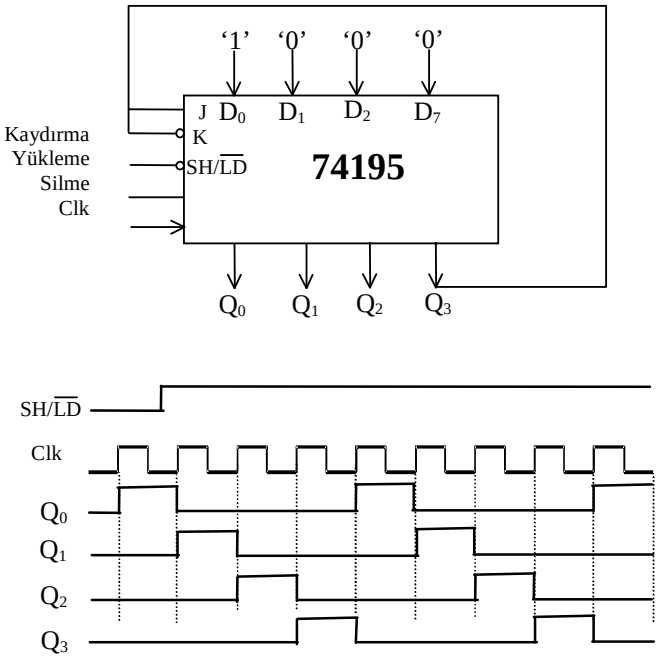
12.3.2. Kaymalı Kaydedicinin Halka Sayıcı Olarak Kullanılması

Seri giriş - seri çıkışa sahip kaymalı kaydedicide, çıkışın seri bilgi girişine bağlanması ile, **'Halka Sayıcı'** devresi elde edilir. Seri giriş-paralel çıkışa sahip kaymalı kaydedicide ise, en yüksek değerli biti temsil eden çıkışın seri girişe bağlanması ile halka sayıcı devresi oluşturulur.

74195 dört bitlik kaymalı kaydedici entegresi, Şekil 12.20.a'daki gibi bağlanarak, halka sayıcı olarak kullanılabilir.

Entegrede gerekli bağlantı (en yüksek değerli çıkışın seri girişe bağlanması) yapıldıktan sonra $(1000)_2$ bilgisi paralel yükleme girişlerine uygulanırsa, ilk tetikleme sinyali ile birlikte bilgiler paralel olarak entegreye yüklenir. Bilginin yüklenmesinden sonra uygulanan kaydırma sinyalleri ile, çıkıştaki bilgi Şekil 12.20.b'deki zaman diyagramında görülen değişimi gösterir. Bu durumda devre, halka sayıcı devresi işlevini gerçekleştirir.

Kaymalı kaydedici devresinde, çıkıştan girişe doğru olan bağlantının en son FF'nin Q çıkışı yerine $\overline{Q}$' den alınarak yapılması ile Johnson sayıcı devresi elde edilir. Bu bağlantı ile kaymalı kaydedicinin Johnson sayıcı olarak kullanılabileceği ortaya çıkar.



Şekil 12.20. 74195 Entegresinin halka sayıcı olarak bağlantısı ve 4 bitlik halka sayıcı çıkış sinyal şekilleri.

12.3.3. Seriden Paralele Veri Dönüşümü

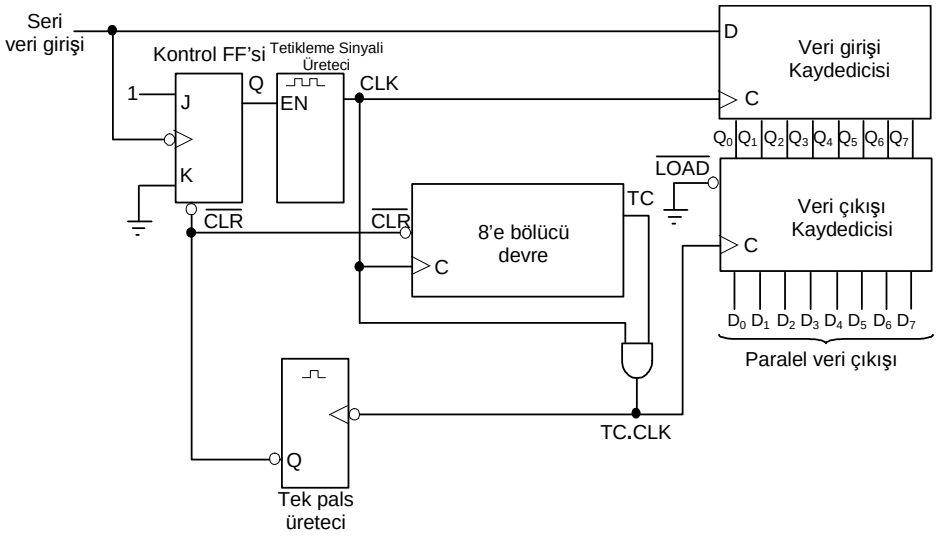
Seri veri iletişimi, bir sayısal sistemden diğerine veri iletişimi sırasında gerekli bağlantının hat sayısını azaltmak amacıyla kullanılır. Örneğin, 8 bitlik bir bilginin seri olarak iletilmesi tek bir hattan yapılabilirken, aynı bilginin paralel olarak iletilmesi için 8 hat gereklidir. Bu özellik nedeni ile, bilgisayarlar ve mikroişlemciler arasındaki veri iletişiminin çok hızlı olmasının gerekmediği durumlarda, seri haberleşme kullanılır.

Bilgisayar veya mikroişlemci temelli sistemlerde veri iletiminin seri olarak yapılması durumunda, seri olarak iletilen bilgilerin işlenebilmesi için paralel bilgilere dönüştürülmesi gerekir. Seriden paralele veri dönüşümü, kaymalı kaydediciler kullanılarak yapılabilir.

Seri - paralel veri dönüşümü sırasında, seri bilgi ile birlikte başlangıç ve bitişi temsil eden bilgilerin iletilmesi gerekir. Bu nedenle, seri bilgi iletiminde başlangıcı göstermek için bir bitlik '0' bilgisi, bitişi göstermek için iki bitlik '1' bilgisi kullanılır.

Şekil 12.21'de, iki kaymalı kaydedici kullanılarak gerçekleştirilen, seri-paralel veri dönüştürücü devresi prensip şeması görülmektedir. Basitleştirilmiş devreye Şekil 12.22'de görülen 8 bitlik veri kısmı bulunan (toplam 11 bit) seri bilginin uygulanması durumunda

devrenin çalışmasını özetleyelim: Başlangıç bilgisi olarak kullanılan bitin gelmesi ile, kontrol FF'si '1' değerine kurulur ve bu değer 'Clk' sinyali üretecini yetkilendirir. Yetkilendirilen 'Clk' sinyali üreteci, tetikleme sinyali üretmeye başlar. Üretilen sinyal, veri girişi kaydedici ve 8'e bölücü devrelere uygulanır.

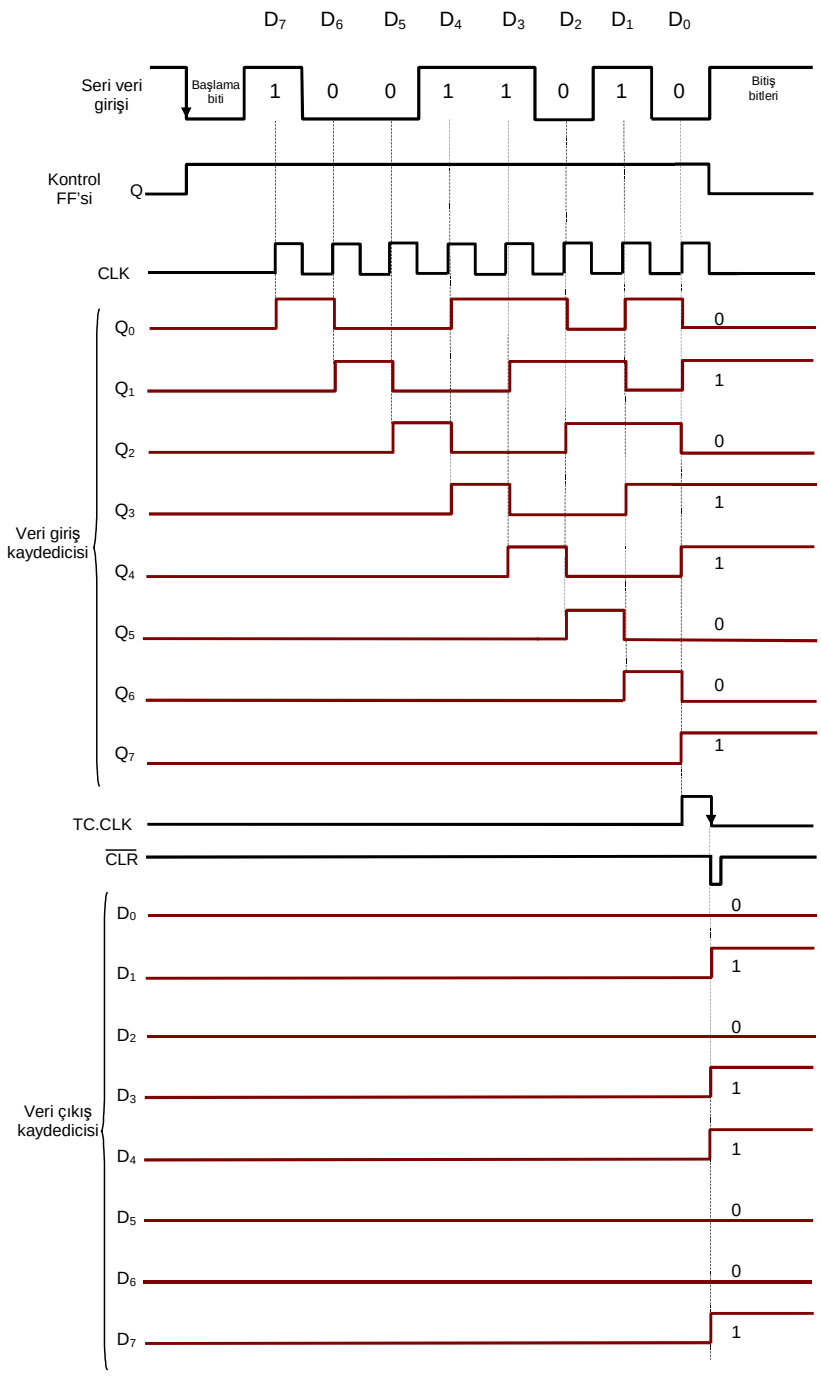


Şekil 12.21. Kaymalı kaydedici kullanılarak yapılan seri – paralel veri dönüşümü prensip şeması.

Tetikleme sinyali üreteci devresinde üretilen 'Clk' sinyali, seri veri ile aynı frekansa sahiptir. Üretilen 'Clk' sinyali ile, ilk veri biti veri kaydedicisine yüklenir. 8 bitlik (D_7-D_0) seri bilgi, veri giriş kaydedicisinde seri olarak kaydırılır. Takip eden Clk sinyalleri ile, seri olarak gelen bilgiler sırası ile veri giriş kaydedicisine yüklenir.

8. tetikleme sinyali (Clk) sonucunda, 8'e bölücü devrenin TC çıkışındaki 'VE' kapısının çıkışı kısa bir süre '1' değerini alarak tetikleme palsı oluşturur. Oluşan tetikleme palsı (TC-Clk), veri çıkış kaydedicisini tetikler ve veri giriş kaydedicisi çıkışlarındaki 8 bitlik bilginin çıkış kaydedicisi paralel girişlerinden kaydediciye yüklenmesini sağlar. 'TC-Clk' tetikleme palsı, aynı anda tek-pals üretecini tetikler. Tetiklenen sinyal üreteci kısa süreli bir pals üretir. Üretilen pals, 8'e bölücü devredeki sayıcıyı ve aynı anda kontrol FF'ini sıfırlayarak tetikleme sinyali üretecini yetkisizlendirir.

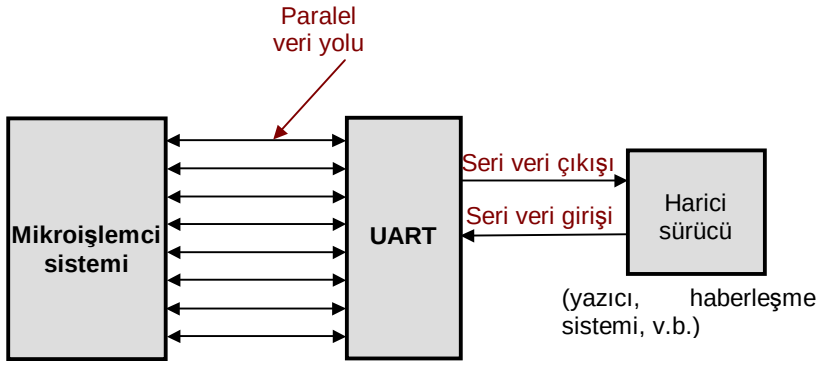
Seri – Paralel dönüştürücü devre, açıklanan olaylardan sonra diğer bir 11 bitlik bilgiyi almaya hazırdır. Devre ilk gelecek başlangıç bitini bekler.



Şekil 12.22. Seri – Paralel veri dönüşümü işlemi zamanlama diyagramları.

12.3.4. Universal Asenkron Alıcı Verici

Bilgisayar ve mikroişlemci temelli sistemler veri gönderimi ve alımı işleminde genellikle paralel haberleşme sistemini kullanırlar. Bununla beraber, bu sistemler çevre birimleri ile veri alma / gönderme işleminde veri haberleşme şekli olarak seri formu kullanırlar. Paralel ve seri haberleşme şekilleri arasında dönüşümü sağlamak amacıyla, 'Universal asenkron alıcı – verici (Universal Asynchronous Receiver Transmitter – UART)' olarak isimlendirilen arabirim elemanı kullanılır (Şekil 12.23).

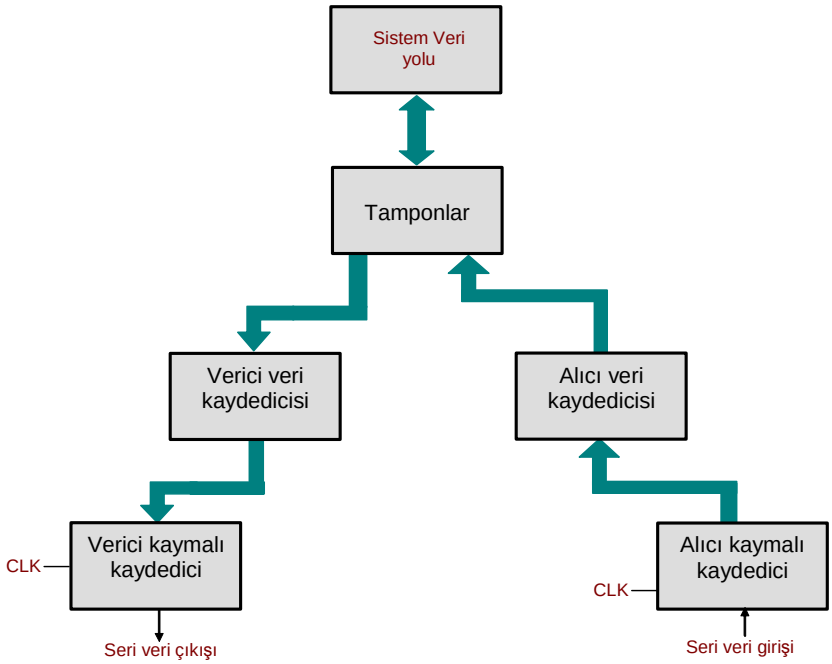


Şekil 12.23. Universal asenkron alıcı verici arabirim elemanı kullanımı.

UART arabirim elemanı, seri – paralel veri dönüşümü kısmında açıklanan seri – paralel veri dönüştürücü ile birlikte, paralel – seri veri dönüştürücü devresi içerir (Şekil 12.24). Veri yolu, UART ile mikroişlemcili sistem arasında veri taşınmasını sağlayan paralel hatlar setidir. Tampon (buffer), veri kaydediciler ile veri taşıtı arasında arabirim elemanıdır.

Seri formattaki verileri alan 'UART', aldığı veriyi paralel şekle dönüştürür ve oluşan bilgiyi sistem veri yolu üzerine yerleştirir. Seri-paralel dönüştürme işleminde Şekil 12.21'de verilen seri giriş-paralel çıkışlı kaydedici devresi kullanılabilir. Ayrıca, sistem veri yolundan paralel formattaki veriyi alan 'UART', aldığı veriyi seri forma dönüştürür ve veriyi çevre birim elemanlarına gönderir.

Paralel-seri veri dönüşümü için Şekil 12.16'da verilen paralel giriş-seri çıkışlı kaydedici devresi kullanılabilir.



Şekil 12.24. Temel ‘UART’ blok şeması.

Tekrarlama ve Çalışma Soruları

1. ‘Kaydedicileri’ tanımlayınız.
2. Kaydedicileri bilginin yüklenmesine göre ve bilgi giriş / çıkışına göre sınıflandırınız.
3. ‘Paralel kaydediciyi’ tanımlayınız.
4. Senkron paralel kaydedici devresi çizerek çalışma prensibini kısaca açıklayınız.
5. Asenkron paralel kaydedici devresinin çalışma prensibini açıklayınız.
6. ‘Seri kaydedici’ terimini tanımlayarak,’kaydırma sinyali’nin kullanılma yerini açıklayınız.
7. Seri kaydedicileri, bilgi kaydırma yönüne göre sınıflandırınız.
8. Sağa kaydırmalı kaydedici’yi tanımlayarak, sağa kaydırma işlemini şekille açıklayınız.
9. Hem seri, hemde paralel veri girişine sahip entegrelerde oluşan olayları örnek üzerinde açıklayınız.

10. Üç FF'ye sahip bir sağa kaymalı kaydedicide bilgi yüklenmesi işlemini şekille açıklayınız.

11. Dört FF'den meydana gelen sağa kaymalı kaydediciye, '1000' bilgisinin giriş bilgisi olarak uygulanması durumunda 7. tetikleme palsı sonucunda FF'lerde meydana gelecek değerleri tablo olarak gösteriniz.

12. 'Sola kaymalı kaydedici' devresini tanımlayarak, çalışmasını şekille açıklayınız.

13. Sola kaymalı kaydedici devresine (4 FF bulunan) $(1011)_2$ bilgisinin yüklenmesi işlemini şekille açıklayınız.

14. $(1011)_2$ bilgisinin yüklü olduğu sola kaymalı kaydedici devresine uygulanan 2. tetikleme sinyali sonunda yeni yüklenen bilgileri bulunuz.

15. Sağa / Sola kaymalı kaydedici devresinde gerçekleştirilebilecek olan aritmetik işlemleri açıklayınız.

16. Sağa / Sola kaydırma işlemini blok şema üzerinde açıklayınız.

17. Sağa / Sola kaydırıcı devresi lojik devresini çizerek, çalışmasınız açıklayınız.

18. Bilgi giriş – çıkışına göre kaydedicileri sınıflandırınız.

19. Seri giriş – seri çıkışlı kaydedici prensip şemasını çizerek, çalışmasını açıklayınız.

20. 7491 entegresi genel özelliklerini açıklayınız.

21. Seri giriş – paralel çıkışlı kaydedici devresini şekille açıklayınız.

22. 74164 entegresinin genel özelliklerini blok şema ile açıklayınız.

23. Paralel giriş – seri çıkışlı kaydedici devresini kısaca açıklayınız.

24. 74165 entegresi iç yapısını çizerek, kullanımını özetleyiniz.

25. Paralel giriş – paralel çıkışlı kaymalı kaydedici prensibini açıklayınız.

26. 74174 entegresi genel özelliklerini kısaca özetleyiniz.

27. 'Universal Kaydedicileri' tanımlayınız.

28. 74178 entegresi özelliklerini özetleyiniz.

29. 74178 entegresini, paralel yükleme ve sola kaydırma işlemini gerçekleştirecek şekilde tasarlayınız.

30. 74178 entegresini halka sayıcı olarak bağlayınız, çalışma prensibini özetleyiniz.

31. Kaymalı kaydedici uygulama alanlarına 3 adet örnek veriniz.

32. Kaymalı kaydedicinin zaman gecikmesi amacıyla kullanılmaasını örnek devre ile açıklayınız.

33. Kaymalı kaydedicinin halka sayıcı olarak kullanılmasını örnekle açıklayınız.

34. Kaymalı kaydedici devresi ile Jonhson sayıcı devresi oluşturarak, çalışma prensibini özetleyiniz.

35. Bilginin seriden paralele dönüşümünün gerekli olduğu yerler nerelerdir?

36. Seri – Paralele veri dönüşümünü gerçekleştiren devrenin prensip şemasını çizerek, çalışmasını özetleyiniz.

37. Paralel - Seri veri dönüşümü için gerekli devrenin prensip şemasını çiziniz.

38. Üniversal asenkron alıcı – verici (UART) elemanının işlevini açıklayınız.

39. 'UART' elemanının blok şemasını çizerek, yaptığı işlemi özetleyiniz.

40. Kaymalı kaydedicinin klavye kodlayıcı olarak kullanımını araştırınız.

BELLEK (HAFIZA) DEVRELERİ

(MEMORIES)

Giriş

Bilgisayarlarda ve programlar ile çalışan endüstriyel devrelerde kullanılan programların veya programların çalıştırılması sırasında işlenen verilerin ikili bir yapıda saklanması amacıyla kullanılan devreler veya elemanlar, **'bellek / hafıza devreleri'** olarak isimlendirilir. Diğer bir deyişle; bilgilerin geçici veya daimi olarak saklandığı birimler 'bellek devreleri' olarak isimlendirilir. Bilgisayarın bir elemanı olduğu dijital sistemlerin analog sistemlere üstünlüklerinden birisi, büyük hacimli bilgileri küçük alanlarda uzun süreli saklayabilme özelliğidir.

Dijital sistemlerde bellek elemanı olarak; flip-flop'lar (FF), FF'lerin değişik bağlantısıyla oluşturulan kaydedici devreleri (yarı iletken elemanlar), kondansatör ve bilgi saklama kapasiteli diğer elektronik elemanlar / devreler (manyetik elemanlar / ortamlar) kullanılır. Bellek devresinin sayısal sistemin içinde bulunduğu durumlarda yarı iletken bellekler kullanılırken, bilgilerin taşınması istenen durumlarda manyetik veya optik belleklerden faydalanılır. Hızlı bellek elemanları olan FF'ler ve kaydediciler yardımıyla, bilgilerin birimler arasında hızlı olarak transferi mümkündür. Bilgisayarlarda dahili olarak yapılan işlemlerde yaygın olarak kullanılan FF'li kaydedicilerin yanında, küçük güç tüketiminin gerektiği yerlerde bellek olarak kondansatörler kullanılabilir.

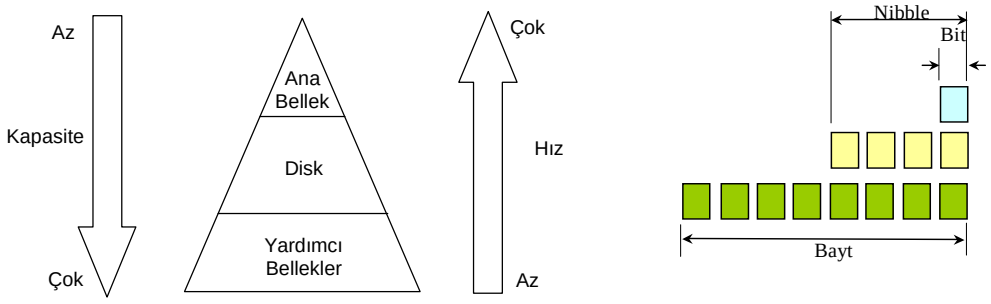
Yarı iletken elemanlardan oluşan FF temelli bellekler, hızın önemli olduğu yerlerde ana bellek olarak kullanılır. Dahili olarak kullanılan ana bellek devresi, bilgisayar programının çalışması ve programın çalışma sırasında gerekli verilerin sağlanması işlemlerine yardımcı olur. Bu durumda bellekler konusunda yeni terimler oluşur. Bellek çeşitlerini genel olarak inceledikten sonra, bellekler ile ilgili temel terimleri açıklayalım.

Yapıldıkları malzeme şekline göre; yarı iletken, optik bellekler ve manyetik bellekler olarak gruplandırılabilen bellekler, yerleştirildiği yer olarak; dahili ve harici bellekler, yaptıkları işlemler açısından; ana bellek ve yardımcı bellek olarak, bellek bölgesine erişim şekline göre; rasgele veya sıralı erişimli bellekler olarak gruplandırılabilir.

Bilgisayar merkezi işlem birimi ile doğrudan irtibatlı, bilgisayara dahili olarak yerleştirilen, temelde yarı iletken elemanların oluşturduğu bellek devreleri; yerleştirildikleri yer olarak **'dahili bellek'**, yaptıkları işlem nedeniyle **'ana bellek'** olarak isimlendirilir.

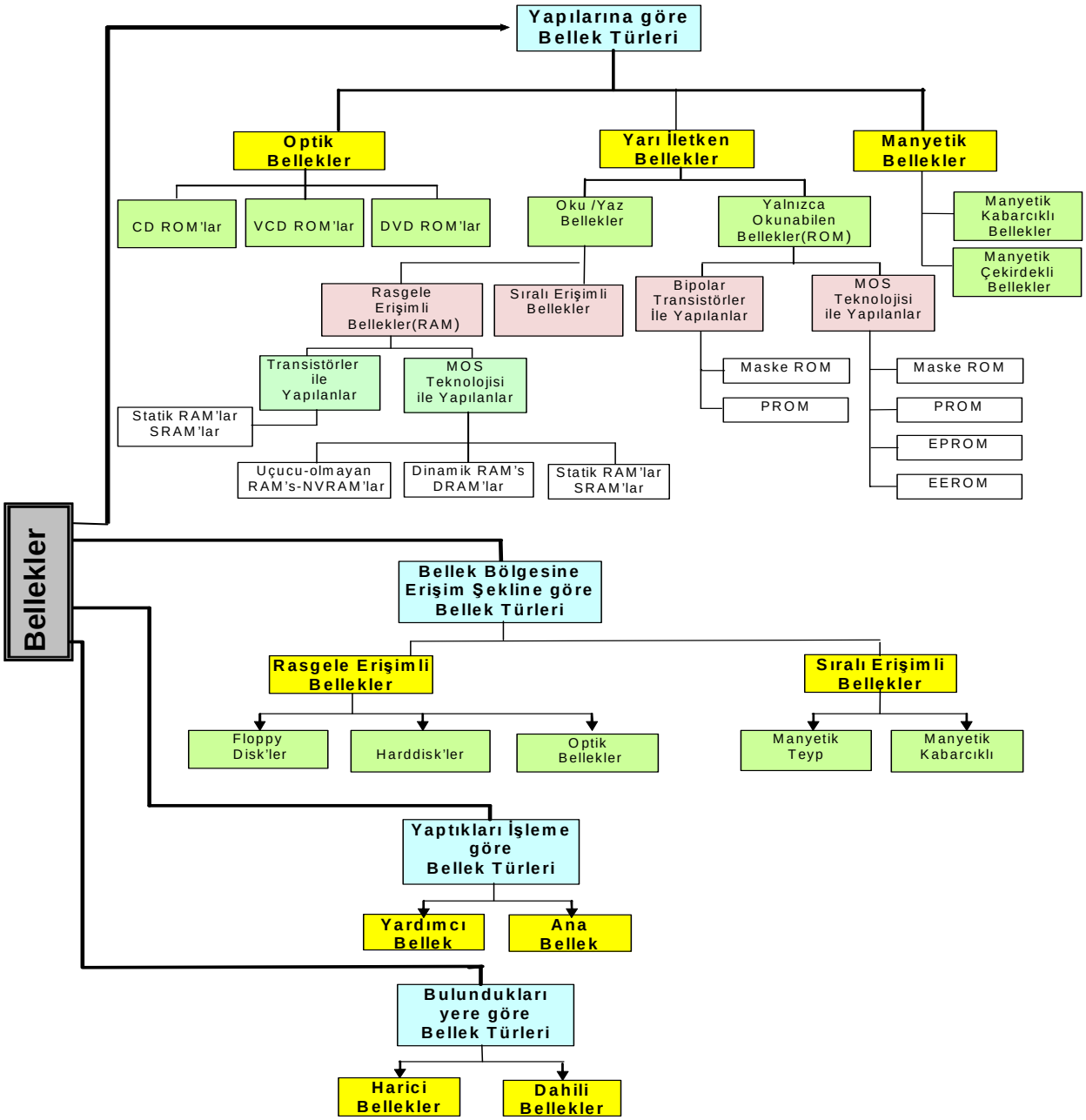
Çok büyük hacimli bilgilerin saklanması amacıyla kullanılan ve genelde bilgisayarın dışında oluşturulan bellek elemanları, yerleştirildikleri yer nedeniyle **'harici bellek'** olarak isimlendirilirken, yaptıkları işlem açısından **'yardımcı / yedek bellek'** olarak isimlendirilirler. Yardımcı bellekte saklanan bilgiler, bilgisayarın ihtiyacı olduğu anlarda bilgisayara yüklenerek kullanılırlar.

Yardımcı bellek olarak; manyetik disk veya manyetik teyp, floppy disketler, manyetik kabarcıklı bellekler (magnetik bubble memory - MBM) veya birleşik diskler (compact disks - CD) kullanılır. Bit başına maliyeti dahili belleklere göre daha ucuz olan harici (yardımcı) bellekler, çok uzun süreli bilgi saklamada tercih edilirler. Manyetik bilgi saklama prensibine göre çalışan, ancak yarı iletken yapıya sahip manyetik kabarcıklı bellekler (MBM'ler) yavaş çalışmaları nedeni ile dahili bellek olarak kullanılamazlar. Şekil 13.1'de dahili / ana belleklerin hız ve kapasite açısından karşılaştırılması görülmektedir.



Şekil 13.1. Ana bellek ve yardımcı belleklerin hız / kapasite açısından karşılaştırılması, belleklerde bayt yapısı.

Farklı yapı ve farklı kullanım yerlerine sahip bellekleri, Şekil 13.2'deki gruplar altında inceleyerek, her grubun temel özelliklerini açıklayalım. Bununla beraber, önce kullanacağımız terimleri, kavramları ve belleklerde yapılan temel işlemleri tanımlayalım.



Şekil 13.2. Belleklerin sınıflandırılması ve bellek çeşitleri.

13.1. Belleklerde Kullanılan Terimler ve Yapılan Temel İşlemler

Bellekler ile ilgili temel terimleri açıklamak / anlamak, bellek ile ilgili işlemlerin daha iyi anlaşılmasına yardımcı olacaktır.

Bit : '0' veya '1' değerini alabilen ikili sayı BİT (Binary Digit) olarak isimlendirilir. BİT, sayısal sistemlerde en küçük bilgi birimidir.

Nibble : Dört bitin bir araya gelerek oluşturduğu bilgi grubu 'nibble' olarak isimlendirilir (Şekil 13.1.b). Bir bayt, dört bitlik iki gruba ayrılabilir ve her grup **'nibble'** olarak adlandırılır.

Bayt (Byte) : 8 bitlik bilginin açıklanması için kullanılan özel terim.

Bazı kaynaklarda, nadirende olsa, bayt terimi farklı uzunluklardaki bitleri tanımlamak için kullanılsa da, genel kabul baytın 8 bit olduğudur.

Bellek kelimesi (Memory Word) : Aynı tip verileri veya bir komutu temsil etmek amacıyla kullanılan bir bellek içerisindeki bir grup bit.

16 bitlik bir kelimeyi saklamak amacıyla kullanılan 16 FF'nin oluşturduğu kaydedici, bellek kelimesi için örnek olabilir. Bellek kelimesi uzunluğu, bilgisayarların ölçüsüne bağlı olarak 4 bit– 64 bit arasında değişir. Bununla beraber genel kabul, 16 bit yani iki baytlık bilginin kelime (word) olarak isimlendirilmesi şeklindedir.

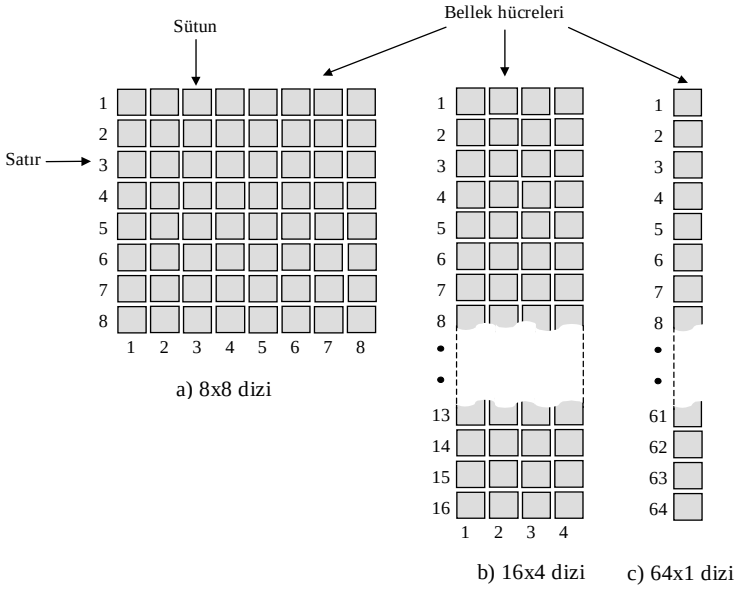
Bellek Hücresi (Memory Cell) : Tek bir bitlik bilgiyi (0 veya 1) saklayabilmek için kullanılan eleman, devre veya cihaz.

Bellek hücresine örnek olarak; bir FF, şarj edilmiş bir kondansatör, manyetik teyp veya disketteki tek bir spot verilebilir.

Bellek Dizisi (Memory Array) : Bellek hücrelerinin bir araya gelerek oluşturduğu yapı.

Bellek hücreleri dizileri oluştururken farklı kombinasyonlarda bir araya gelebilirler (Şekil 13.3). Bellek dizilerinde, hücrelerin yan yana bir araya gelerek oluşturduğu yapı **'satır'**, düşey doğrultuda bir araya gelerek oluşturduğu yapı **'sütun'** olarak isimlendirilir.

64 hücreden oluşan bir bellek farklı şekillerde organize edilebilir. Hücrelerin kare olacak şekilde bir araya gelmesi ile 8x8 dizi oluşur ve toplam 64 bitlik bellek kapasitesini belirtir (Şekil 13.3.a). Aynı sayıda hücre, 16x4 dizi oluşturacak şekilde (Şekil 13.3.b) veya 64x1 dizi şeklinde yerleştirilebilir (Şekil 13.3.c).



Şekil 13.3. Bellek hücrelerinin farklı kombinasyonlar ile dizi şeklinde düzenlenmesi.

Kapasite (Capacity) : Tüm bellek sisteminde veya belirli bir elemanda saklanabilecek bit sayısını belirtmek için kullanılan terim.

Şekil 13.3'de bulunan belleklerin tümünde kapasite 64 bittir. 4096 tane 20-bit kapasiteli bellekten bahsettiğimiz zaman, belleğin toplam kapasitesi;

4096x20=81.920 bit

demektir ve 4096x20 bit olarakta ifade edilebilir.

Bir bellegin kapasitesi, 2^{10}=1024 biti ifade eden '1K' kısaltması ile açıklanabilir. Bu durumda, 4Kx20 bir bellek ile 4096x20 kapasiteli bellek ifade edilir.

Daha büyük kapasiteli belleklerin geliştirilmesi ile, 2^{20}=1.048.576 biti ifade eden '1 Mega–1M' ve 2^{30}=1024 M biti ifade eden '1Gega-1G' sınıflandırılmaları yapılmıştır. Yapılan sınıflandırmaya göre, 2Mx8 bir bellek ile, 2.097.152x8 kapasiteli bellek ifade edilmektedir.

Örnek 1: 5Mx8 ve 1Mx16 şeklinde ifade edilen belleklerin, hangisinde daha çok bilgi saklanabileceğini bulalım.

5Mx8 = 5x1.048.576x8 = 41.943.040 bit,

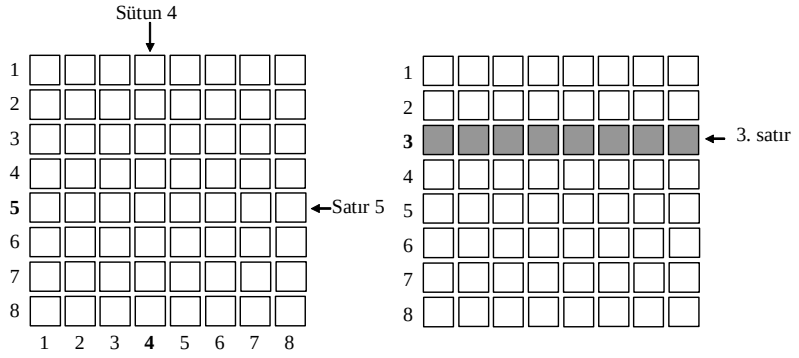
1Mx16 = 1.048.576x16 = 16.777.216 bit.

Bu durumda, 5Mx8 kapasiteli bellek daha çok bilgi saklayabilir.

'Depolama yoğunluğu' terimi, entegrelerin bilgi saklama kapasitelerini karşılaştırmak amacıyla kullanılır. İki bellek entegresinin karşılaştırılmasında; daha çok bilgi saklama kapasitesine sahip olan belleğin 'depolama yoğunluğu' daha yüksek olarak açıklanır.

Adresler (Address) : Bellek dizisinde bir hücrenin veya kelimenin bulunduğu bölgeyi / yeri ifade eden sayı.

Bir bellek elemanı veya sistemde saklanan her bir hücre / kelime belirli bir adrese sahiptir. Örneğin; Şekil 13.4.a'da bulunan hücrenin adresi satır ve sütun olarak ifade edilir (satır 5, sütun 4). Şekil 13.4.b'de bulunan hücrelerin adresi ise yalnızca 3. satır olarak tanımlanır. Bu açıklamalardan, bellek adresini tanımlama şeklinin bellek organizasyonu ile ilişkili olduğu sonucunu çıkarabiliriz.



a) Tek bir hücrenin adresi. **b)** Satır olarak çok sayıda hücrenin adresi.

Şekil 13.4. Bellek adres bölgelerinin tanımlanması.

Adresler sekizli, onlu veya onaltılı olarak ifade edilseler de, ikili sayılar ile açıklanırlar. Şekil 13.5'de sekiz kelimeyi saklamak amacıyla kullanılan ve her bir kelimenin saklandığı adresin 3-bitlik ikili sayıyla temsil edildiği bellek yapısı görülmektedir. Bellekteki belirli bir kelime bölgesinden bahsettiğimiz zaman, bölgeyi tanımlamak için kullanılan adres kodunu kullanırız.

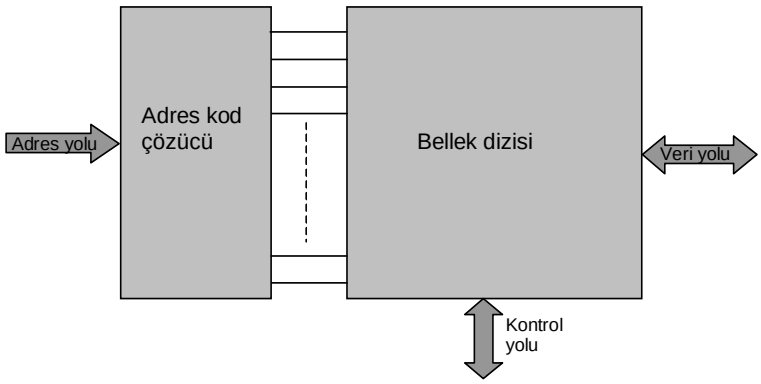
Adres kodları belleklerde bulunan adres hatları ile belleğin adres girişlerine uygulanır.

Adresler	Bilgi
000	Kelime 0
001	Kelime 1
010	Kelime 2
011	Kelime 3
100	Kelime 4
101	Kelime 5
110	Kelime 6
111	Kelime 7

Şekil 13.5. Her bir kelime belirli ikili adrese sahiptir.

Yollar / Taşıtlar (Buses) : Mikroişlemcili bir sistem içerisindeki birimler veya mikroişlemcili sistem ile çevre elemanları arasında iletişimi sağlayan hatlar.

Birimler arasında iletişimi sağlayan ve **'yol'** veya **'taşıt'** olarak adlandırılan hatlar, veri iletişimi amacıyla kullanılıyorsa, **'veri yolu'** olarak isimlendirilir. Veri yolu ile tek yönlü veya çift yönlü iletişim mümkündür (Şekil 13.6). Veriler 8 bitten oluşuyor ise veri yolunda 8 hat bulunur. Veri yolunda 8, 16 veya 32 hat bulunabilir.



Şekil 13.6. Belleklerde iletişim ve yollar.

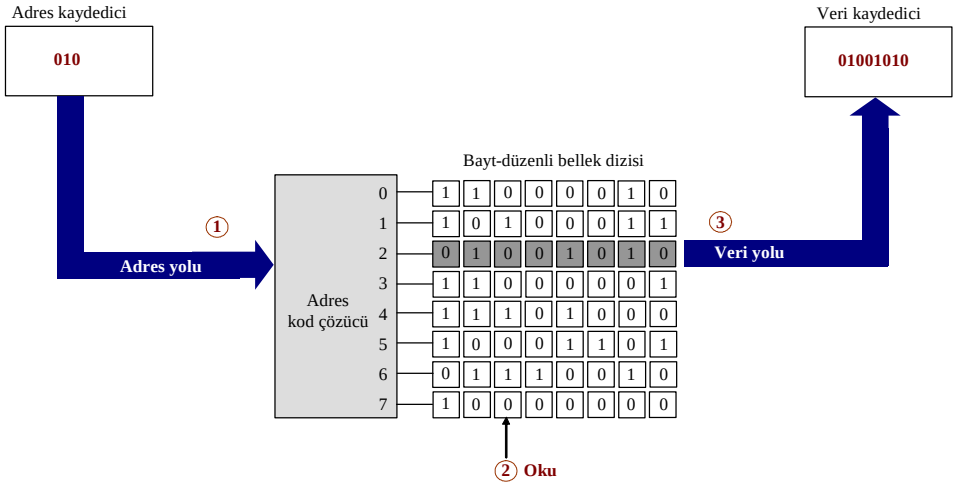
Bellekler üzerinde yapılan işlemleri belirlemek ve belleği yetkilendirmek amacıyla kullanılan bilgileri taşıyan hatlar, **'kontrol yolu'** olarak tanımlanır. Belleklerde kontrol

yolunda bulunan hat sayısı; belleğin çeşidine, kapasitesine, yapısına, vb. özelliklere bağlı olarak değişir.

Belleklerin adreslerini temsil eden bilgileri taşıyan hatlar, **'adres yolu'** olarak isimlendirilir (Şekil 13.6). Adres yolundan gelen adres bilgileri, adres kod çözücü devrelerde çözülerek, bellekte ilgili bölgenin seçilmesini sağlar. Adres yolunda bulunan hatların sayısı, bellek kapasitesi ile ilişkilidir. 4 hatlık yoldan gelen bilgi ile 16 bellek bölgesi tanımlanabilirken (2^4=16), 8 hat ile 256 bellek bölgesi temsil edilebilir (2^8=256) ve 16 adres hattı ile 64K (2^{16}=65.536=64K) bellek bölgesi adreslenebilir.

Okuma İşlemi (Read Operation) : Belirli bir bellek adresinde saklanan ikili bilginin belirlenip, farklı bir elemana aktarılması işlemi.

Şekil 13.7'deki 2 nolu bellek kelimesini kullanmak istediğimiz zaman, '010' nolu adreste okuma işlemi gerçekleştirmemiz gerekir. Okuma işlemi, bazı durumlarda kelimenin bellekten alınması nedeniyle **'alma (fetch)'** olarak ta adlandırılır.



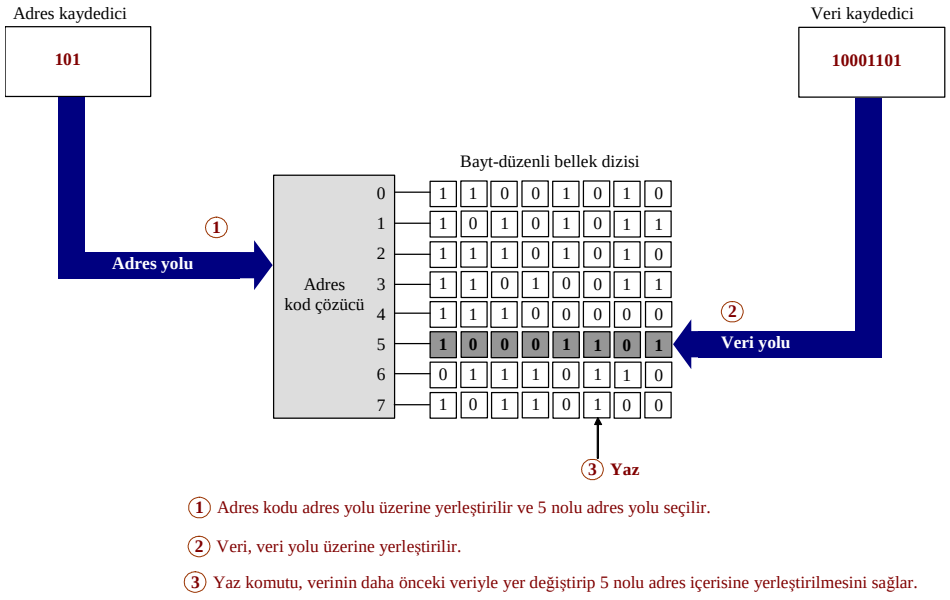
Şekil 13.7. Belleklerde okuma işlemi.

Bir bellek bölgesinde bulunan veriyi okumak için, adres kaydedicisinde bulunan ve verinin okunacağı bellek bölgesini (adresi) temsil eden bilgiler adres yolu üzerine yerleştirilir. Adres yolu üzerindeki veri, adres kod çözücü tarafından çözülerek ilgili adres bölgesi bulunur.

'OKU' komutu ile, bulunan adres bölgesindeki verinin kopyası veri yolu üzerine yerleştirilir ve veri kaydedicisine depolanır. Bellek bölgesinden veri okunması işleminde, bellekte bulunan veride bir değişiklik olmaz.

Yazma İşlemi (Write Operation) : Belirli bir bellek bölgesine yeni bir bilginin yerleştirilmesi veya bir adres bölgesinde saklanan bilgilerin yeni bilgilerle yer değiştirmesi işlemi.

Bir bellek bölgesine verinin konulabilmesi için, adres kaydedicisindeki değer adres yolu üzerine yerleştirilir (Şekil 13.8). Adres yolu üzerindeki değer, adres kod çözücü tarafından çözülür ve ilgili bellek bölgesi seçilir. Bu arada 'YAZ' komutu işleme konularak, veri kaydedicisinde bulunan veri seçilen bellek bölgesine yerleştirilir (Şekil 13.8). Bir bellek bölgesine 'YAZ' komutu ile yeni bir bilgi yazılması, eski bilginin kaybolmasına neden olur.



Şekil 13.8. Belleklerde yazma işlemi.

Erişim Zamanı (Access Time) : Bir bellek elemanının işlem hızı ölçüsüdür ve bir okuma işlemini gerçekleştirmek için gerekli toplam zamanı ifade eder. Diğer bir deyişle; bir belleğin yeni bir adres bilgisi alması ile verinin bellek çıkışında hazır olduğu durum arasında geçen süre, **'Erişim zamanı - t_{ACC}'** olarak isimlendirilir.

Uçucu Bellek (Volatile Memory) : Saklama işlemi için elektrik enerjisinin gerektiği bellek tipi. Elektrik kaynağının kesilmesi ile bellekte saklanan tüm bilgi silinir. Yarı iletken belleklerin çoğu uçucu iken, tüm manyetik bellekler uçucu olmayan (nonvolatile) özelliğe sahiptirler.

Rastgele Erişimli Bellek (Random Access Memory - RAM) : Okuma veya yazma işlemi sırasında bellek bölgesine erişim için, çalışılan bellek bölgesinin konumunun hiçbir etkisinin olmadığı bellek türü. Diğer bir değişle, tüm bellek bölgeleri için okuma veya yazma işlemi erişim zamanının aynı olduğu bellek çeşidi. Yarı iletken ve manyetik nüveli belleklerin çoğu RAM'dır. RAM'lar takip eden kısımlarda detaylı olarak incelenecektir.

Sıralı Erişimli Bellek (Sequential Access Memory – SAM) : Erişim zamanının çalışılan bellek bölgesinin konumuna bağlı olarak değişim gösterdiği bellek türü. Bir bilgi, bilginin saklandığı adrese kadar ki tüm bellek bölgelerinin sırası ile kat edilmesi sonucunda bulunur. Bu işlem, RAM tipi belleklere göre çok daha uzun bir erişim zamanı gerektirir.

SAM tipi belleklere örnek olarak manyetik teyp'ler, disk ve manyetik kabarcıklı bellekler gösterilebilir. Bu tip belleklerdeki işlem, içinde şarkılar bulunan bir kasette istediğimiz bir şarkıya erişmek için yaptığımız hızlı erişim işlemine benzetilebilir. İstediğimiz şarkıya erişim zamanı kasetlerde şarkının kaydedildiği bölgeye bağlı olduğu gibi SAM belleklerde biligye erişim, bilginin bulunduğu yere bağımlıdır.

Oku / Yaz Bellekler (Read / Write Memory - RWM) : Okuma ve yazma işlemlerinin benzer işlem aşamaları ile gerçekleştirildigi her hangi bir bellek türü. Bilgilerin yazılıp, daha sonra okunabildikleri tüm bellek çeşitlerine verilen genel ad. RAM bellekler bu gruba girer.

Yalnızca Okunabilen Bellekler (Read Only Memory – ROM) : Yazma işleminin yalnızca bir kere ve üreticiler tarafından gerçekleştirildiği bellek türü. ROM bellekteki bilgiler istenilen sayıda okunabilir. Yazma işleminin bir kereden fazla gerçekleştirilebildiği ROM tiplerinde; yazma işlemi okuma işlemine göre çok daha karmaşık olduğundan, yazma işlemi çok sık olarak yapılmaz. Çok değişik türlerin bulunduğu ROM belleklerin hepsi uçucu olmayan tiptedir ve elektrik enerjisi kesilse dahi bilgiler saklanmaya devam eder. ROM'lar takip eden kısımlarda detaylı olarak incelenecektir.

Statik Bellek Elemanları (Static Memory Devices) : Saklanan bilginin elektrik enerjisi uygulandığı sürece yeniden yazmaya (tazelemeye) gerek olmadan saklandığı yarı iletken bellek elemanları.

Dinamik Bellek Elemanları (Dinamic Memory Devices) : Saklanan bilginin uzun süre sabit olarak saklanamadığı, elektrik enerjisi uygulanması yanında, bilginin belirli aralıklarla tazelenmesini (refresh) gerektiren bellek türü.

Dahili / Ana Bellek (Internal / Main Memory) : Komutların ve CPU'nun üzerinde çalıştığı verilerin saklandığı, bilgisayarın ana belleği (Main memory). Dahili bellekler,

bilgisayarlarda bulunan en hızlı belleklerdir ve genellikle yarı iletken elemanlarından yapılırlar. Bilgisayarlarda bulunan RAM bellekler, dahili belleklerdir.

Harici / Yedek Bellek (External or Auxiliary Memory) : Genellikle bilgisayarın dışında bulunan ve yavaş çalışan bellek türü. Yardımcı bellek olarak ta isimlendirilen bu tip bellekte, dahili belleğe göre çok daha büyük hacimli bilgiler saklanabilir. Dahili belleklere göre daha yavaş olan bu bellekler, uçucu olmayan özelliktedirler. Manyetik teypler ve diskler yaygın olarak kullanılan yardımcı bellek türleridirler.

Temel terimleri özetledikten sonra, daha sonraki konuların anlaşılmasına yardımcı olması açısından belleklerde yapılan genel işlemleri inceleyelim.

13.2. Genel Bellek İşlemleri

Bellek işlemleri her bir bellek tipi için farklı olsa da, genel işlemlerin prensipleri tüm bellek çeşitleri için aynıdır. Genel olan işlemleri açıklamak, belirli tip bellek elemanlarındaki işlemleri anlamaya yardımcı olacaktır. Önce bellek işlemlerini gerçekleştirmek için gerekli giriş / çıkış işlemlerini özetleyelim.

i- Okuma veya yazma işlemi için erişilecek adresi seç,

ii- Okuma veya yazma işlemini seç,

iii- Yazma işlemi ise, bellekte saklanacak bilgiyi sağla,

iv- Okuma işlemi ise, bellekten gelen veriyi çıkış terminallerinde tut,

v- Adres girişlerine ve oku / yaz komutlarına cevap vermek için belleğin yetkilendirme girişini uygun sinyal ile uyar.

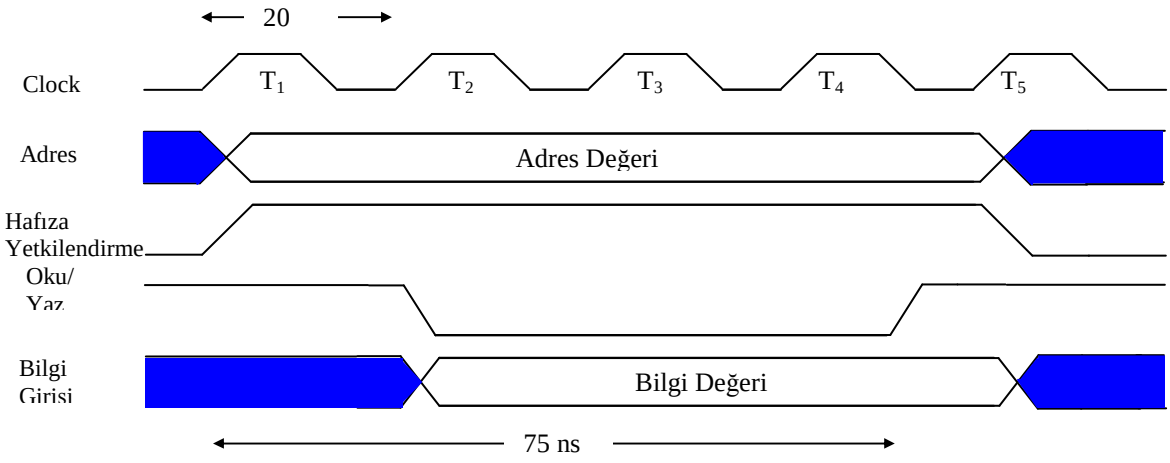
Okuma/Yazma işlemlerinden uygun olanın seçilmesi için R/W girişine ve entegrenin seçilmesi için entegre seçme (Chip Select) girişine uygun sinyallerin uygulanması gerekir. Bu işlemleri gerçekleştirmek için bellek entegresine Tablo 13.1'de özetlenen sinyaller uygulanır.

Entegre Seçme (CS)	Oku/Yaz (R/W)	Bellek İşlemi
0	X	İşlem Yok
1	0	Yazma İşlemi
1	1	Okuma İşlemi

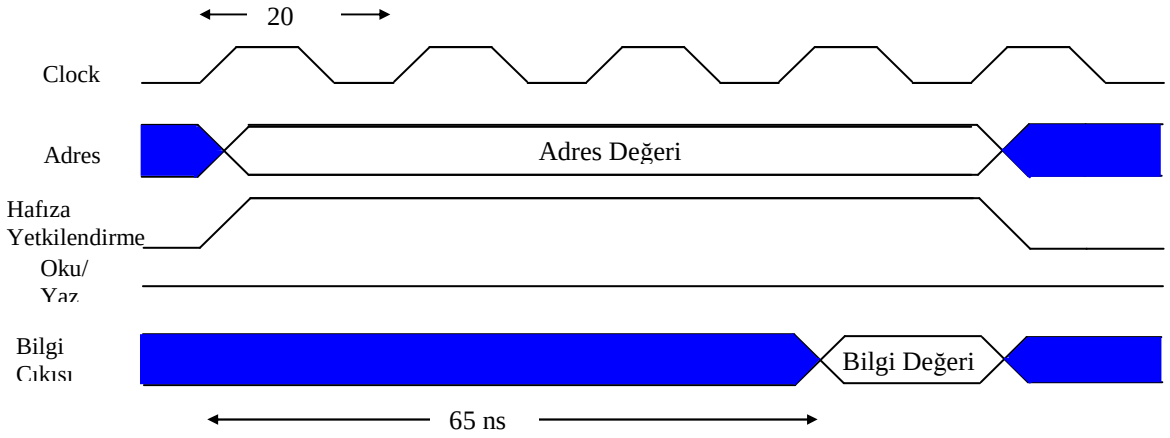
Tablo 13.1 Bellek entegresinde bulunan kontrol sinyalleri / girişleri.

Entegre seçme girişinin (CS) aktif olmaması durumunda entegrede hiçbir işlem yapılmaz. CS girişinin aktif olması ile, okuma veya yazma işlemlerinden birisi seçilebilir.

Bellek entegresindeki işlemler, Merkezi İşlem Birimi (MİB) veya bellek yönetim birimi (MMU) gibi harici kontrol birimleri tarafından kontrol edilir. Kontrol işleminde MİB tetikleme sinyali değil, bellek entegresine kontrol sinyallerinin uygulama zamanları referans alınır. Bununla beraber, okuma ve yazma işlemleri sistem tetikleme (saat) sinyali ile senkronize edilir. Bunun anlamı; bellek entegresi erişim zamanının ve okuma/yazma işlemlerinin belirli sayıda saat sinyali periyodu için MİB ile uyumlu çalışmasıdır.



a) Yaz Saykılı



b) Oku Saykılı

Şekil 13.9 Bellek okuma/yazma işlemleri zamanlama dalga şekilleri.

Örnek olarak; 50 MHz tetikleme frekansı ile çalışan, her bir saat periyodunun 20 nsn (1 nsn = 10^{-9} sn) olduğu MİB'de; erişim zamanının 65 nsn ve yazma zamanının 75 nsn olması durumunda oluşan olayları inceleyelim.

Belleğe erişim için harcanan toplam periyot süresinin erişim zamanından büyük (veya en azından eşit) olması gereklidir. MİB saat periyodunun 20 nsn ve erişim zamanının 75 nsn olması nedeni ile gerekli periyot sayısı dört olarak bulunur.

Şekil 13.9'da, 50 MHz MİB saat sinyali ile birlikte 75 nsn yazma saykıl zamanı ve 65 nsn erişim zamanına sahip bellek entegresinin bellek saykıl zamanlama şeması görülmektedir. Şekil 13.9.a'da her biri 20 nsn'lik T_1, T_2, T_3 ve T_4 saykılları sırasında gerçekleştirilen 'yazma' işlemi detaylandırılmaktadır. Yazma işleminde, MİB tarafından adres ve veri girişi sağlanır. T_1 sinyalinin yükselen kenarına bellek entegresi yetkilendirilir ve adres entegreye uygulanır. T_2 saykılının yükselen kenarında ise, yazılacak veri entegreye uygulanır. T_2 saykılı sırasında, R/W pininde oluşan değişiklik okuma işleminin yapılması için gerekli bilgiyi sağlar. Yanlış değerlerin yazılmasına engel olmak için, veriler adres bilgisinden sonra uygulanır ve R/W sinyali belirli bir süre '0' değerinde tutulur. R/W sinyali '1' değerine değiştikten sonra adres ve veri bilgilerinin belirli bir süre daha devam etmesi gerekir. 5 nsn'lik bu süre, yeni kontrol sinyalleri ile yeni T_1 sinyalinin işlem yapmasını sağlar.

Şekil 13.9.b'de, adres ve bellek yetkilendirme sinyallerinin uygulandığı 'okuma' işlemi gösterilmektedir. T_1 sinyalinin yükselen kenarında adres ve bellek yetkilendirme sinyalleri entegreye uygulanır. Adres ve bellek yetkilendirme sinyali uygulandıktan 65 nsn sonra erişilen bellek bölgesindeki bilgi veri yolunda gözükür. Bu veri, bir sonraki T_1 sinyali sırasında sistemde bulunan kaydedicilere aktarılır.

Bellek işlemlerini gerçekleştirmek için, bellek entegrelerinin uygun giriş / çıkış ayaklarına (pinlerine) sahip olması gerekir. Şekil 13.10.a'da 32 adet 4 bitlik bilgiyi saklama kapasitesine sahip, 32x4 bir belleğin basitleştirilmiş temel işlevleri gösterilmektedir.

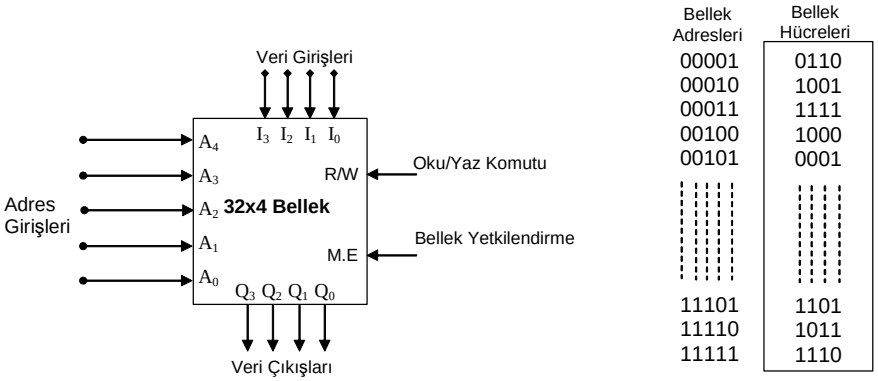
Açıklanan eleman 4 bitlik olduğundan, I_0-I_3 giriş hatlarına ve Q_0-Q_3 çıkış hatlarına sahiptir. Yazma işlemi sırasında bilgiler veri girişleri yardımıyla belleğe yüklenirken, okuma işlemi sırasında bellekten okunan bilgiler veri çıkışlarından alınır. Bellek entegresinde bulunan bağlantıları kısaca özetleyelim.

Adres Girişleri : Bellek 32 kelimeyi depoladığından dolayı, 00000'dan 11111'e kadar ki ikili sayılarla ifade edilen 32 farklı adrese, diğer bir deyişle 32 bilgi saklama bölgesine sahiptir. Adres bölgelerinden herhangi birisine erişmek için, A_0-A_4 ile temsil edilen adres girişlerine 5 bitlik adres kodu uygulanır.

Genelde, 2^N kelime saklama kapasitesine sahip bir belleğin, 'N' sayıda adres girişinin olması gerekir. Bu adres girişleri yardımıyla ulaşılabilen bellek bölgelerine farklı bilgiler yazılabilir veya bu bölgelerde bulunan bilgiler okunabilir (Şekil 13.10.b).

R/W′ Girişi : R/W′ giriş hattı, bellekte yapılacak oku / yaz işlemini belirler. Bazı entegrelerin ayrı-ayrı girişler ile belirlediği bu işlemler, tek bir girişin kullanıldığı durumlarda; R/W^{I}=1 olması ile okuma işlemi, R/W^{I}=0 olması ile yazma işlemi gerçekleştirir.

Bellek Yetkilendirme : Birçok bellek sistemi, sistemin tümünün veya bir kısmının girişlere karşı tepkisiz kalmasını sağlayan bellek yetkilendirme (Memory Enable-ME) girişine sahiptir. Farklı bellek entegrelerinde farklı isimlerle temsil edilen bu girişe uygun sinyal uygulanması ile, belleğin R/W^{I} ve adres girişlerine karşı tepkisiz olması sağlanır. Yetkilendirme girişi, belleklerin bir araya gelerek büyük bellek sistemlerini oluşturduğu düzeneklerde önemli bir işleve sahiptir.



(a) **(b)**

Şekil 13.10. 32x4 bellek şeması ve bellek hücrelerinin adresleri / içerikleri.

Bellek entegreleri sahip oldukları özelliklere ve kapasitelerine göre farklı sayıda veri giriş / çıkış ve adres giriş hatlarına sahip olabilirler. Örneğin; 4Kx8 kapasitesinde bir belleğin;

i- Sahip olacağı veri giriş ve veri çıkış hatları sayısını,

ii- Sahip olacağı adres hattı sayısını,

iii- Byte olarak sahip olacağı saklama kapasitesini bulalım.

İstenilen değerleri sırası ile bulacak olursak;

i- Kelime uzunluğu 8 olduğundan, veri giriş ve veri çıkışları 8 hatta sahip olmalıdır.

ii- Bellek 4K bilgi saklayabildiğinden; 4 x 1024 = 4096 kelime saklama kapasitesine sahiptir. 4096 bellek adresi olması gerektiğinden, $4096=2^{12}$ eşitliğinden 12-bit adres kodu olması sonucunu buluruz.

iii- 1 Byte = 8 bit olduğundan, bellek 4096 Byte saklama kapasitesine sahiptir.

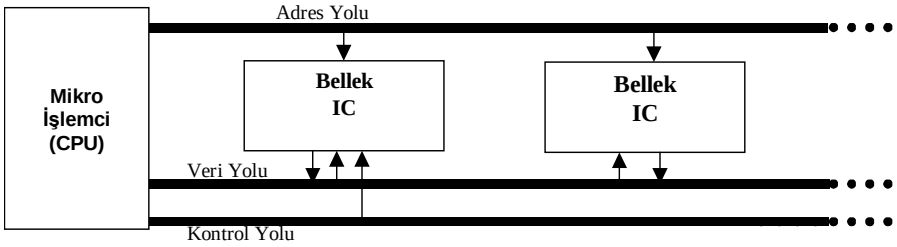
Burada kısaca açıkladığımız bellek sistemlerinde kullanılan yollar (buses), belleğin bilgisayarların içerisinde kullanıldığı durumlarda bellek ile CPU arasında haberleşme sağlarken, haberleşme sırasında taşıdıkları bilgiye göre isimlendirilirler. Daha önce açıklandığı üzere; taşınılan bilgiye göre üç tip yol ortaya çıkar: veri yolu, adres yolu ve kontrol yolu.

CPU'nun gerçekleştireceği okuma / yazma işleminde önemli bir rol oynayan üç tip yol, Şekil 13.11'deki gibi gösterilebilir. Yolların her birisi farklı sayıda hattı içerir ve hatların sayısı kullanıldığı sisteme göre değişiklik gösterir.

Adres Yolu (Address Bus) : Tek yönlü bu adres yolu ile, CPU'nun ikili adres çıkışları bellek entegresine taşınır.

Veri Yolu (Data Bus) : Çift yönlü bu veri yolu ile, CPU ile bellek arasında veri akışını sağlar.

Kontrol Yolu (Control Bus) : Kontrol yolu ile, kontrol sinyalleri ($R/\overline{W}$, ME, vb.) CPU'dan bellek entegrelerine taşınır.



Şekil 13.11. CPU ile bellekleri bağlayan üç yol bağlantısı.

13.3. Bellekleri Sınıflandırılması – Bellek Çeşitleri

Buraya kadar açıklanan terimlerden ve belleklerle ilgili verilen genel bilgilerden faydalanarak, bellekleri farklı bakış açılarıyla farklı şekillerde gruplandırabiliriz. Gruplandırma bakış açısına bağlı olarak değişim gösterse de, tüm gruplandırmalarda genel olarak Şekil 13.2'deki bellek çeşitlerini görebiliriz.

Gruplandırmada ortaya çıkan sınıflar, birbirinin alt sınıflarında yer alabilirler. Örneğin; harici bellek olarak manyetik veya yarı iletken bellekler kullanılabilirken, diğer taraftan yardımcı bellekler dahili veya harici olarak yerleştirilebilir. Bu karmaşıklığı bir yana

bırakarak bellekleri yapılarına göre inceleyip, bu inceleme sırasında diğer gruplardaki bellekleri açıklayalım.

Bellekler yapılarına göre;

i- Manyetik bellekler,

ii- Optik bellekler,

iii- Yarı iletken bellekler,

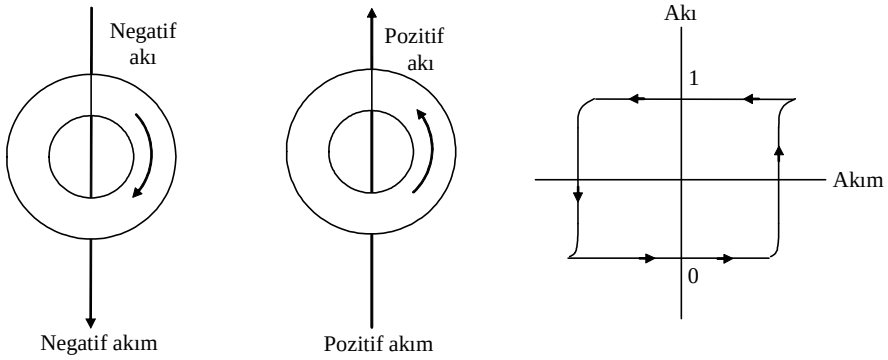
olarak üç grup altında incelenebilir.

13.4. Manyetik Bellekler

Manyetik bellekler, saklama işleminin manyetik olarak gerçekleştirildiği ve bilgilerin geçici olmayacak şekilde (nonvolatile) tutulduğu bellek türleridir. Çeşitli yapı şekillerine sahip manyetik bellekleri alt gruplara ayırarak açıklayalım.

13.4.1. Manyetik Çekirdekli Bellekler

İkili bilgilerin saklanması için manyetik çekirdeklerin kullanıldığı, uçucu olmayan RAM belleklerdir. Manyetik çekirdekli bellekler (Magnetic Core Memories), yarı iletken belleklerin yaygın olarak kullanımından önce bilgisayarlar tarafından dahili bellek olarak kullanılmaktaydılar. Temel manyetik çekirdekli bellek hücresi, manyetik malzemeden yapılmış bir elemandır.



a) Lojik 0 saklanması **b)** Lojik 1 saklanması **c)** Histeriz döngüsü

Şekil 13.12. Bir bitin manyetik çekirdekte saklanması.

Akım, manyetik akı ve gerilim şeklinde üç fiziksel özelliğin kullanıldığı manyetik çekirdekli bellekte, saklama işlemi çekirdeğin ortasından geçen bir sargıya bağlıdır. Bu sargıdan bir akım geçtiği zaman, geçen akımın yönüne bağımlı olarak çekirdek içersinde manyetik akı (flux) oluşur. Manyetik akının kalıcılığı nedeniyle, akım kesilse dahi çekirdek içerisinde magnetizasyon sabit kalır. Magnetizasyonun mümkün olan iki yönü (N ve S kutupları), 1 ve 0 değerlerini temsil etmek için kullanılır. Saat ibresinin tersi yönünde magnetizasyon 1 değerini, saat ibresi yönündeki magnetizasyon ise 0 değerini ifade eder. Şekil 13.12'de çekirdek üzerindeki sargıdan geçen akımın yönü ile, '0' ve '1' değerlerinin oluşması görülmektedir.

Manyetik çekirdekli bellek sistemleri 100ns-500nsn erişim zamanına sahiptirler ve eski mini bilgisayar ve mainfame sistemlerde kullanıldılar. Fakat fiziksel büyüklükleri ve karmaşık arabirim devreleri nedeni ile kullanımdan kalktılar.

13.4.2. Manyetik Kabarcıklı Bellekler (Magnetic Bubble Memories-MBM)

Manyetik kabarcıklı belleklerde, ikili bilgi manyetik malzemeden yapılmış ince bir film üzerinde ince kabarcıklar şeklinde saklanır. Kabarcıkların olması lojik 1, olmaması lojik 0 değerini temsil eder. MBM'ler sıralı erişimli belleklerdir ve veri bilgilerin dış dünya ile irtibatını sağlayan bir toplama (pick-up) noktasından hızla geçer.

Manyetik kabarcıklı belleklerin en önemli avantajı, kalıcı özellikte olmalarıdır. Elektrik enerjisi kesilse dahi, saklanan veri tutuldukları noktada sabit kalarak kaybolmazlar. Diğer kalıcı bellek tiplerinden (ROM-PROM-EPROM) farklı olarak, MBM belleklerde yazma ve okuma işlemi aynı kolaylıkta gerçekleşir. Kalıcı tip bellekler olan teyp ve disk belleklerle karşılaştırılmaları durumunda, MBM sistemi hareketli parçaları olmadığından dolayı sessiz ve daha hızlıdır.

Çok küçük güç tüketimine sahip bu tip bellekler, dahili bellek olarak kullanılabilmeleri için gerekli hıza sahip değillerdir (Erişim zamanı 1~ 2 msn). Manyetik teyp ve disk kaydedicilere göre 100 kere daha hızlı olan bu tip bellekler, gerekli sürüm devrelerinin pahalı olması nedeniyle yaygın olarak kullanılmazlar. Fiyatların düşmesi ile, MBM belleklerin floppy disk sistemlerinin yerini aldığını görmemiz mümkün olacakken, daha ucuz ve kolay üretilen sabit bellekler nedeniyle yaygınlaşmadan ortadan kalkmışlardır.

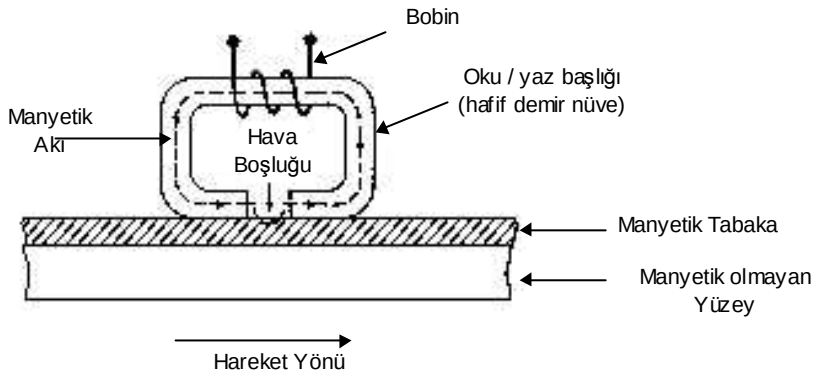
13.4.3. Manyetik Teyp ve Disk Bellekler

Teyp ve disk devrelerinde bulunan oku/yaz başlığı magnetik bir malzemenin yüzeyinde hareket ederek magnetik spot'u okuma ve yazma işlemini geçekleştirir. Teyp ve disk'lerdeki ikili bilgilerin hem kaydedilmesinde hem de okunmasında aynı temel prensip kullanılır.

Şekil 13.13'de, hareketli bir magnetik yüzeyde temel kaydetme işlemi görülmektedir.

Şekildeki okuma / yazma başlığı, etrafına bobin sarılı yumuşak demirden bir çekirdek ve küçük bir hava aralığından oluşur. Yazma işleminde, bobinlerden bir akım akarak çekirdekte manyetik hatlar oluşturur. Bu hatlar, manyetik akı için çok yüksek dirence (reluktansa) sahip hava aralığıyla karşılaşıncaya kadar devam eder. Yüksek direnç, akı hatlarının yolunu değiştirip hareketli yüzeyin manyetik kaplamasına doğru yönlenmesine neden olur ve bobinlerdeki akım hareketli yüzeyde manyetiklenmiş bölgeler oluşturur. Manyetiklenmiş bölgeler oku / yaz başlığı geçtikten sonrada manyetikliğini devam ettirir.

Okuma işlemi, yazma işleminin tersidir ve okuma işlemi sırasında bobin sensor hattı olarak kullanılır. Manyetik yüzey oku / yaz başlığı altında hareket ederken, manyetik bölgeler hava aralığında ve bobinlerde bir akı oluşturur. Bobin akısındaki bu değişiklik, sensor bobinlerinde bir gerilim oluşturur ve bu gerilim, '1' veya '0' olarak kabul edilir.



Şekil 13.13. Hareketli manyetik yüzeyde kayıt işleminde kullanılan temel parçalar.

13.4.4. Sabit Disk Sistemleri

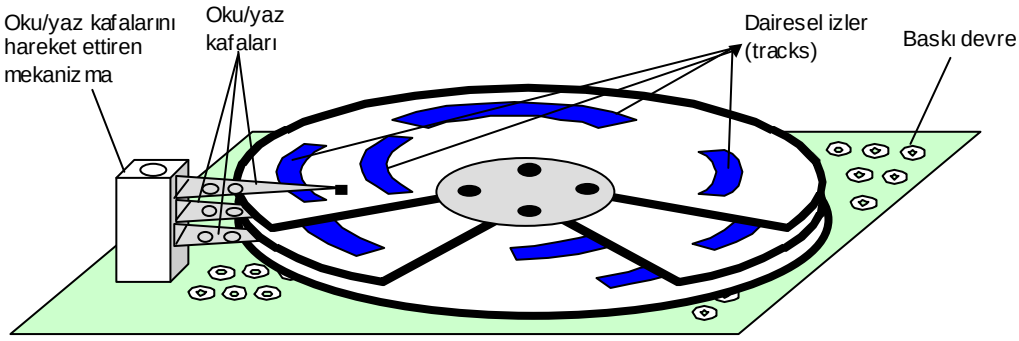
Sabit disk (Hard disk) sistemlerinde veri, her iki tarafı manyetik malzeme ile kaplanmış alüminyum disk üzerindeki dairesel izlere (track'lara) kaydedilir. 3,5 inch'ten 20 inch'e kadar ölçülerde olan hard diskler, 1000-7200 devir/sn'lik bir hızla dönebilir. Hala hazırdaki sabit disklerde, verinin bulunduğu bölgeye okuma veya yazma işlemi yapmak üzere ulaşmak için gerekli zaman yaklaşık 6.3msn-50msn arasında değişmektedir. Ancak bu değerler her yeni ürün ile küçülmektedir.

Sabit diskler, Şekil 13.14'de görüldüğü gibi otomatik bir pikabın eksen mili üzerine dizilmiş plaklara benzerler. Bilgiler, plaklar üzerine sıralı, indeks sıralı veya rasgele erişimli olarak depolanırlar. Depolanan bilgileri okumak veya yazmak amacıyla kullanılan birim, 'manyetik disk sürücü' olarak adlandırılır.

Sabit disk sürücü sistemi, Şekil 13.14'de görüldüğü gibi çok sayıda oku / yaz kafasına sahip olabilir. Oku / yaz kafaları, diskler arasında bulunarak plağın her iki yüzeyinde bulunan dairesel izler üzerinde bilgileri kaydetmek veya okumak için kullanılırlar. Oku / yaz kafalarının bağlı bulunduğu erişim kolu aldığı komutlara göre hareket ederek kafaları bilginin okuma veya yazılacağı bölgeler üzerinde konumlandırır.

Oku / yaz kafalarını hareket ettiren mekanizma, oku / yaz kafalarının plaklar üzerinde hareket etmesini sağlayan uyarıcı sistemdir. Uyarıcı sisteme bağlı olan kafalar dairesel izler üzerinde hareket eder.

Baskı devre, sabit diskin bağlı olduğu diğer sistemler ile uyarıcı sistemler arasındaki bilgi akışını sağlar. Baskı devre üzerinde bulunan elemanlar ile, plakaların dönme ve bilgi okuma hızları kontrol edilir.



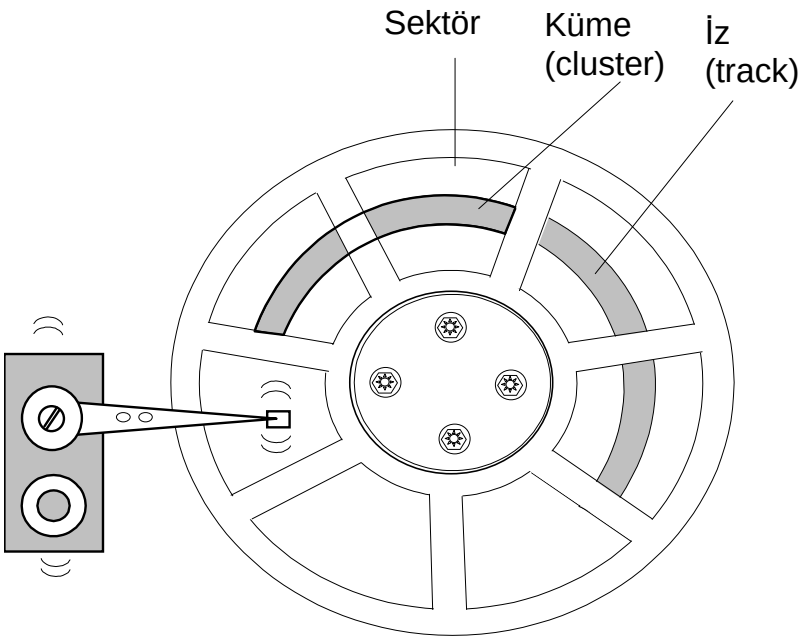
Şekil 13.14. Hard disk sisteminin çalışma prensibi.

Sabit diskler genelde sabit olarak monte edilirler ve normal kullanımda hareket etmezler. Bununla beraber hareketli olan ve taşınabilen sabit diskler bulunmaktadır ve bunlar 'disk paketleri' (disk cartridges) olarak isimlendirilirler. Veri yoğunluğu; sabit disklerde plaka üzerinde bir inç kareye düşen veri miktarıdır. Veri yoğunluğunun büyük olması sabit diskin performansını arttırsa da, sabit disklerin önünde duran en büyük engel veri yoğunluğunun çok daha fazla arttırılamamasıdır. İlki 1956 yılında IBM tarafından geliştirilen sabit diskler 2KB/inc^2 veri yoğunluğuna sahipti. 24 inc'lik sabit disklerden 50 adedi bir arada kullanılarak 5MB'lık depolama yeteneği oluşturuluyordu. 2000 yılı başında 17GB/inc^2 veri yoğunluğuna ulaşılırken, 2002 yılı itibari ile 40GB/inc^2 veri yoğunluğuna erişildi ve kısa süre içerisinde 100GB/inc^2 veri yoğunluğuna erişilmesi hedefleniyor. Ayrıca, yakın gelecekte sabit disklerin yerini alabilecek yeni teknolojiler üzerinde çalışılıyor. Geleneksel manyetik depolama yöntemlerinin yerini alacak bu teknolojiler ile 2002 yılı itibari ile 300GB/inc^2 veri yoğunluğuna sahip 2.5 inc'lik sabit diskler üretiliyor.

13.4.5. Floppy Disket Sistemleri

Manyetik disketler ile aynı yapıda olan ve **‘floppy disket’** veya kısaca **‘disket’** olarak adlandırılan bellekler, tek bir plaka yapısındaki plastik malzemenin üzerinin manyetik malzeme ile kaplanmasıyla elde edilir. Her iki yüzeyine kayıt yapılabilen (douple-sided recording) disketler, 5.25 inch veya 3,5 inch floppy disket ölçülerinde ve 720 KBayt veya 1,44 MBayt kapasitesinde imal edilirler. Şu anda yaygın olarak 3.5 inch’lik büyüklükte ve 1,44 MBayt kapasitesinde disketler kullanılmaktadır.

Sert plastik muhafaza içerisinde bulunan disketlerde, herhangi bir bellek bölgesine erişim zamanı hard disklere göre 10 kat daha fazladır ve bilgi aktarımı çok daha yavaştır. Ancak, düşük fiyatları ve taşınabilir olmaları floppy disketlerin en önemli avantajlarıdır.



Şekil 13.15. Floppy disketlerde bilgilerin saklandığı iz ve sektörlerin oluşturulması.

Disketlerin kullanılabilmeleri için imalatlarından sonra formatlama olarak adlandırılan işleme tabi tutulması gerekir. **‘Formatlama’** işlemi ile disketin üzerindeki izler (track) ve sektörler (sectors) belirlenir. Oluşturulan iz ve sektör sayısı, disketin kapasitesine göre değişir. Şekil 13.15’de görülen kısımlara ayrılan disketin istenilen adresine, istenilen bilgi yüklenebilir.

13.5. Optik Disk Bellekler

Optik disk bellekler, yeni bir bellek teknolojisidir ve piyasada etkin (dominant) olmaya aday gözükmektedir. Çalışma prensibi, çok ince lazer ışınının disk üzerinde yansıması ve 'scattering' işlemine dayanır. Disket üzerinde yanma olayı sonucu oluşan mikroskobik çukurlar (pits) '1' değerini, düzlükler (lands) ise '0' değerini temsil etmek için kullanılır (Şekil 13.16). Spiral olarak hareket eden izler ile temsil edilen veriler, eski plakalarda olduğu gibi, lazer ışınları tarafından pikaplardaki iğnenin sesi okuması gibi okunur. Okunan bilginin yönü içeriden dışarıya doğrudur.

Önemli bir özelliği çok yüksek bilgi saklama kapasitesi olan optik disk belleklerde, erişim zamanı ve veri transfer oranı hard disklerle kıyaslanacak kadar iyi bir durumdadır. Bu bellekler, yazılabilmeleri durumuna göre farklı isimlerle anılırlar.

Yalnızca okunabilen diskler, optik ROM (Optical ROM-OROM) veya kompakt disk-ROM (Compact Disk ROM-CD ROM) olarak adlandırılırlar. Daha geniş bellek kapasitesine sahip optik diskler ise, 'çok yönlü dijital disk-DVD' olarak adlandırılır. Bu tip bellekler, çok büyük programları veya sözlük / ansiklopedi gibi geniş hacimli bilgileri saklamada kullanılırlar. Optik disk bellekleri; CD'ler ve DVD'ler başlıkları altında inceleyelim.

13.5.1. CD-ROM Bellekler

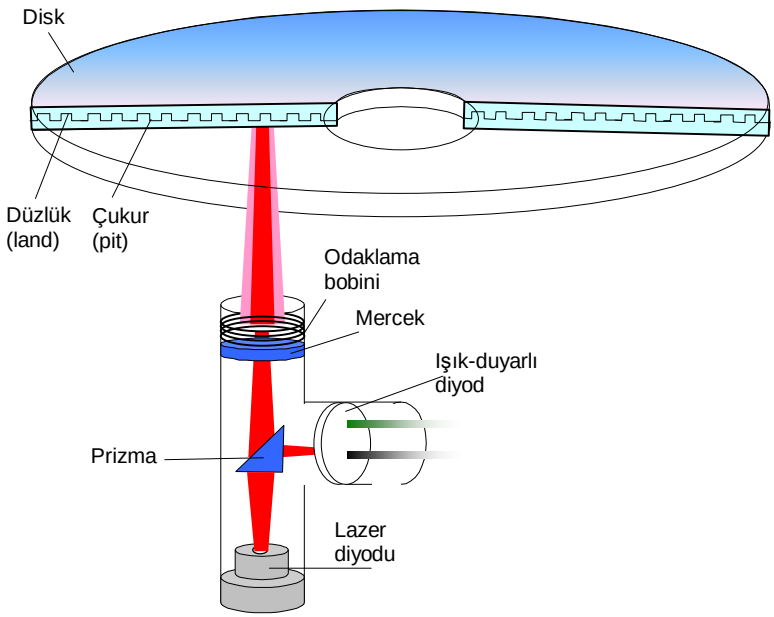
Sony ve Philips'in işbirliği ile geliştirilen CD'ler başlangıçta seslerin dijital formatta kaydedilmesi için geliştirildi. Müzik formatı ile kullanılmaya başlayan CD'leri film izleme formatı ile 'video CD', bilgisayar uygulamaları için 'CD-ROM', resim uygulamaları için 'photo CD' ve oyun uygulamaları için 'oyun CD' takip etti. Yalnızca bir kere bilgi yazılabilen optik diskler, 'bir kere yaz-çok kere oku bellekler' (write once read many-WORM) olarak isimlendirilirler. Yeni geliştirilen yapım teknolojileri ile üretilen yeniden yazılabilir (rewritable) CD ROM'lara, istenildiği kadar yaz-sil işlemi yapılabilmekte ve bu CD'ler floppy disketler veya manyetik hard diskler gibi kullanılmaktadır.

Optik disk bellekler, günümüzde çok kullanılan bellek türlerinden birisi durumundadır. Bunun nedenlerinden birisi geniş bellek kapasitesi ise, bir diğeri ucuz bir yedekleme birimi olmasıdır. Standart CD-ROM'larda yaklaşık 680 MBayt veri depolanabilmektedir. CD'lerin yapısında başlangıçtan beri olan gelişmeler, farklı standartların oluşmasına neden oldu. Oluşan farklı standartlar, farklı renklerde kitaplar olarak adlandırılmaktadır:

- Ses CD'leri tanımlayan 'Red Book',
- Veri CD'leri oluşturmak için 'Yellow Book',
- Eşzamanlı dosyaları saklamak için kullanılan CD-i'ler (CD Interactive) için 'Green Book',

- Yeniden yazılabilir CD'ler (CD-R) için 'Orange Book',
- Video-CD'ler için 'Write Book',
- CD Plus'lar için 'Blue Book'.

CDROM'lar ile ilgili standartların detayları, bellekler ile ilgili kitaplarda bulunabilir.



Şekil 13.16. Optik disk belleklere (CDROM) bilgi yazılması işlemi.

CD'ler üzerindeki bilgiler, yaklaşık 6 km uzunluğundaki spiral şeklinde içten dışa doğru ilerleyen CD yüzeyinde yer alır. CD'nin başlangıcı (ilk izi) olan 'Lead-in-area'nın en iç kısmında TOC (Table of Contents) olarak adlandırılan ve sabit sürücüdeki FAT (file allocation table) görevini yapan bir bölüm bulunur. TOC bölgesi, hangi verinin yüzeyin neresinde bulunabileceği bilgisini içerir. Bu bilgilerin bulunduğu başlangıç bölümü bir iz genişliğindedir. Bu bölümü takip eden 99 iz'e ses, resim veya veri kaydedilebilir. İzler ile, herbiri 2352 Byte'dan oluşan sektörler oluşturulmuştur. CD'nin son kısmında ise CD'nin son kısmı olduğunu bildiren ve 0'lardan oluşan bilgiyi içeren 'Load-Out-Area' bulunur.

CD ROM'lara bilgi yazılması işlemi karmaşık bir sistemi gerektirir. İşlem temelde, düz bir yüzey üzerine lazer ile çukurlar (pits) oluşturulması esasına dayanır. Çukur ve düzlükler iki farklı seviye olarak '0' ve '1' bilgilerini temsil etmek amacıyla kullanılır (Şekil 13.16).

Hepsi 1.2 mm kalınlığında ve 12 cm çapında olan CD-ROM'larda bilgiler, CD-ROM'un polikarbonat katmanına çok küçük çukurlar halinde basılır ve oluşan çukurlar 'pit' olarak isimlendirilir. Oluşturulan pitler yaklaşık 0,12 mikron derinliğinde, 0,5 mikron genişliğinde ve 0,8-3,5 mikron uzunluğundadır. Çukurların arasında bulunan yüzeye 'düzlük' (land) adı verilir.

Okuma işlemi sırasında CD-ROM'un yüzeyi lazer ışık demeti tarafından taranır. Tarama sırasında düzlük ve çukurlardan yansıyan ışık demeti farklılık gösterir. Bu farklılık ışığa duyarlı diyot tarafından algılanarak, '0' ve '1' şeklinde ifade edilen sayısal veriler şekline dönüştürülürler (Şekil 13.16).

CD-ROM'lar bir kere yazılabilen yapıda üretildikleri gibi, yeniden yazılabilir(CD-R) yapıda da üretilmektedirler. İki CD arasında yapı olarak farklılık bulunmaktadır: CD-ROM'larda yansıma yüzeyi alüminyumdan yapılırken, CD-R'lerde yansıma yüzeyi altından yapılmaktadır.

13.5.2. Dijital Video Diskleri (DVD'ler) – Digital Versatile Disks

CDROM'lar (kısaca CD) geniş bellek kapasiteleri nedeni ile büyük hacimli bilgilerin saklanmasına ve taşınmasına imkan tanıdılar. Diskette taşınması mümkün olmayan büyük hacimli bilgilerin CD'lere kaydı mümkün oldu. 680 MByte bilgi depolama kapasitesi ile CD ROM'lar bilgisayar yazılımları yanı sıra, filmlerin kaydının bile kaydedilebileceği ortamlar oluşturdular.

Bilgilerin, bir disk üzerine ince lazer demeti ile oluşturulan çukurların durumuna göre kaydedildiği CD'ler kısa sürede çok yaygın olarak kullanılmaya başladı. Ancak, CD ROM'ların yerini kısa sürede DVD'ler aldı. DVD'ler başlangıçta Dijital video ortamı için tasarlanmış olduklarından, **'Dijital Video Disk'** olarak tanındılar. Film üreticileri, ürünlerini DVD ortamına kısa sürede adapte ederek, birçok filmi DVD ortamında piyasaya sürdüler. DVD'nin asıl ismi ise, **'çok yönlü dijital disk'** anlamına gelen **'Dijital Versatile Disk'**tir. DVD bu ismini, iki yönlü (dual side) bilgi kaydedilmesi ve her yönünde iki katman (dual layer) bulunması nedeni ile almaktadır (Şekil 13.17). DVD'lerde birinci katmandaki bilgi içeriden dışarıya doğru okunurken, ikinci katmandaki bilgi dışarıdan içeriye doğru okunur. Bu özellik, katmanlar arasında değişim sırasında lazer kafasının hareketi için zaman kaybedilmemesini sağlar.

Bilgi saklama prensibi olarak CD'lere benzeyen DVD (Dijital Video Disk yada Dijital Versatile Disc) belleklerin CD'lerin yerini hızlı bir şekilde alması nedenleri;

i - Aynı alanda çok daha geniş hacimli bilgilerin saklanabilmesi,

ii-DVD çalıcı cihazların geleceğin resim kaydetme sistemi olarak nitelendirilmesi,

olarak sıralanabilir.

DVD'ler üzerindeki kapasitenin çok yüksek ve veri aktarma hızının daha süratli olması, CD'ler ile yapılamayan işlemlerin yapılmasına imkan tanımaktadır.

13.5.3. DVD – ROM Sürücülerin Çalışma Prensibi

DVD'ler, üzeri polimer kayıt katmanıyla kaplı oluklardan oluşur. DVD'ler çalışma prensibi olarak CD'ler ile büyük benzerlik taşır. DVD'ler de 120 mm çapında ve 1,2 mm kalınlığındadır. Her iki elemanda kullanılan temel prensip aynıdır. Her iki sistemde de çok ince bir lazer demeti, disk üzerinde mikroskobik çukurlar ile ifade edilen sayısal bilgiyi okur.

DVD teknolojisinin sırrı, CD'de kullanılan tek katman yerine DVD'de çok katman kullanılabilmesi (Şekil 13.17) ve DVD'lerde disk üzerinde oluşturulan çukurların CD'lere göre çok daha küçük boyutta olmasıdır (Şekil 13.18). CD'lerde verilerin oluşturduğu çukurlar 0,834 mikron uzunluğunda iken, DVD'ler de bu boyut 0,4 mikrondur. Ayrıca, CD üzerindeki spiral iz 1,6 mikron aralıklarla yer alırken, DVD üzerinde 0,74 mikron mesafe vardır. Bu ölçüler aynı boyuttaki DVD'de, CD'ye göre 7 kat daha fazla bilgi saklayabilme kapasitesi sağlar. Disk üzerindeki çukurların küçük boyutlu olması, DVD'lerde çukurları tarayan lazer demetinin CD'lerde kullanılan lazer demetine göre çok daha ince olması sonucunu doğurur.



a) En basit haliyle tek taraflı ve tek katmanlı bir DVD'nin yapısı.



b) Tek taraflı ve çift katmanlı DVD'nin yapısı. 1.2 mm'lik kalınlığa sahip diskte iki farklı katman var.

Şekil 13.17. Tek katmanlı ve çift katmanlı DVD'nin yapısı ile farklı DVD'lerin bilgi saklama kapasiteleri.

Bunun yanında, DVD'ler de kullanılan lazer tipi, CD'ler de olduğu gibi kızılötesi değil kırmızı lazerdir. Diğer bir değişle; CD'ler de 770-830 mikron arasında daha boyu, DVD'ler de ise 635-650 mikron arasında dalga boyu kullanılır. Düzlüklere ve çukurlara doğru yönlendirilen ışık demeti, çukurlardan daha az düzlüklerden ise daha fazla yansıtılır. Yansıtılan bu ışığın miktarına göre dijital veri elde edilir.

DVD'ler de tek yüz ve tek katman kullanılması durumunda 4.76 GBayt bilgi depolanabilirken, tek yüz çift katmanda 8,56 GBayt, çift yüz tek katmanda 9,46 GBayt ve son olarak iki yüz - her yüzde çift katman kullanılması durumda 17.6 GBayt'a kadar bilgi saklama kapasitesine ulaşılmaktadır. CD'de toplam kapasitenin ortalama 680 MBayt olduğu düşünülürse, eskiden birkaç CD'ye sığabilen yazılımların tek bir DVD'ye yerleştirilebileceği bulunur. Bu kapasite; tek bir DVD'ye iki saatlik bir filmin tamamının 8 farklı lisanda dublaj ve 32 farklı lisanda alt yazı seçeneği ile kaydedilmesi anlamına gelmektedir. Bu özellikleri nedeni ile, DVD'ye en büyük ihtiyaç eğlence ve film sektöründen geliyor.



Şekil 13.18. CD-ROM ve DVD-ROM'daki bilgi taşıyan çukurların karşılaştırılması.

DVD sürücülerin geriye doğru teknoloji uyumluluğu kısa sürede yaygınlaşmasının sebeplerinden biri olarak verilebilir. DVD sürücülerde, CD-ROM'lar sorunsuz olarak kullanılabilir. CD-ROM'ları okumak için, DVD'leri okumada kullanılan yüksek yoğunluklu lazer demeti 'hologram lens' yardımıyla CD-ROM'ları okuyabilecek hale getirilir (Şekil 13.19) . Bu durumda, aynı lazer demeti hem CD-ROM, hem de DVD-ROM'ları okumak için kullanılır. Ayrıca, aynı lazer demeti DVD'lerdeki farklı katmanlara odaklanarak, odaklandığı katmanı okumak için kullanılır. Üstteki katman yarı geçirgen (transparent) iken, alt katman standart yansıtıcı yapıdadır. Şekil 13.20'da, CD ve DVD'lerin özellikleri karşılaştırılmaktadır.



Şekil 13.19. CD-ROM ve DVD-ROM'ların aynı mekanizma ile okunması.

Özellik	**CD-ROM**	**DVD-ROM**
Disk çapı	120 mm	120 mm
Disk kalınlığı	1,2 mm	1,2 m (0,6x2
İz kalınlığı	1,6 mikron	0,4 mikron
Lazer dalga boyu	770-830 mm	635-650 mm
Lazer tipi	Kızıl ötesi	Kırmızı lazer
Veri kayıt yüzü	1	1 veya 2
Veri katmanı	1	1 veya 2
Kapasite/katman	682 MB	4,700 MB
Kapasite/...	682 MB	4.7-8,5 GB
Toplam kapasite	682 MB	17 GB
Veri aktarım hızı 1x	153. 6KBps	1385 KBps
Veri okuma hızı	307.2 KBps	2770 KBps

Şekil 13.20. CD-ROM ile DVD-ROM'un karşılaştırılması.

DVD cihazlar kullanım yerlerine göre; DVD video, DVD audio, DVD-ROM veya DVD-RAM olarak gruplara ayrılmaktadır.

DVD'lerde kullanılan lazer demetinin yoğun olması yani daha kısa dalga boyuna sahip olması nedeni ile çizik ve kirli DVD'lerde problem oluşacağı düşünülebilir. Ancak DVD'lerde kullanılan RS-PC (Reed Solomon Product Code) isimli özel hata düzeltme

tekniği ile oluşabilecek hataların önüne geçilir.

DVD teknolojisinin hala hazırdaki en büyük sakıncası, ev kullanıcıları için kayıt olanaklarının pahalı olmasıdır.

Şekil 13.18. CD-ROM ve DVD-ROM'ların aynı mekanizma ile okunması.

Özellik	**CD-ROM**	**DVD-ROM**
Disk çapı	120 mm	120 mm
Disk kalınlığı	1,2 mm	1,2 m (0,6x2
İz kalınlığı	1,6 mikron	0,4 mikron
Lazer dalga boyu	770-830 mm	635-650 mm
Lazer tipi	Kızıl ötesi	Kırmızı lazer
Veri kayıt yüzü	1	1 veya 2

Veri katmanı	1	1 veya 2
Kapasite/katman	682 MB	4,700 MB
Kapasite/...	682 MB	4.7-8,5 GB
Toplam kapasite	682 MB	17 GB
Veri aktarım hızı 1x	153. 6KBps	1385 KBps
Veri okuma hızı	307.2 KBps	2770 KBps

Şekil 13.19. CD-ROM ile DVD-ROM'un Karşılaştırılması

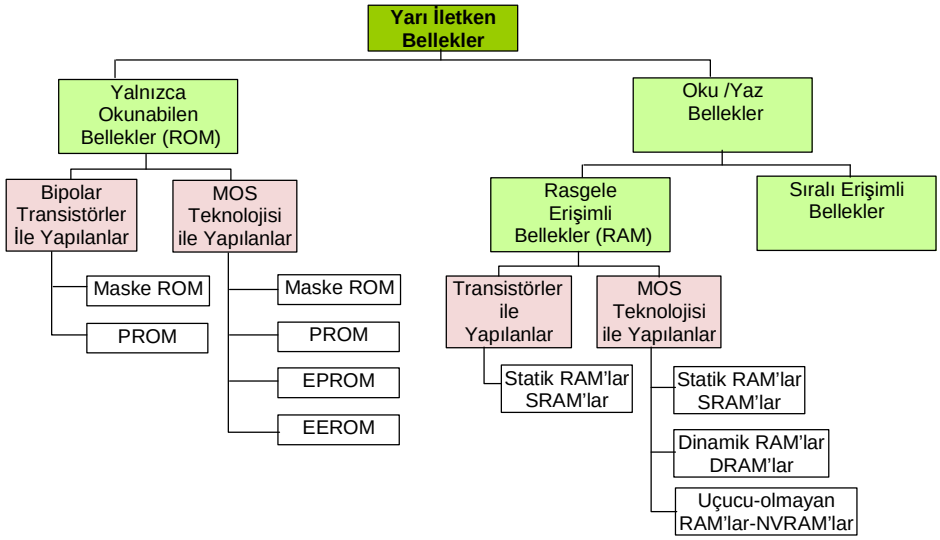
13.5. Yarı İletken Bellekler

Bilgisayarlarda ana bellek olarak kullanılan ve çeşitli entegreler şeklinde imal edilen yarı iletken bellekleri başlıca iki grupta toplayabiliriz.

i- Yalnız okunabilen bellekler (Read Only Memories – ROM)

ii- Okunabilen / yazılabilen bellekler (Read / Write Memories – RWM)

Şekil 13.21'de gösterilen çeşitli alt gruplara ayrılabilen ve kompleks bir yapıya sahip olan yarı iletken belleklerin özelliklerini ayrı ayrı inceleyelim.



Şekil 13.21. Yarı iletken bellek çeşitleri.

13.6.1. Yalnızca Okunabilen Bellekler (ROM'lar)

Adından da anlaşılacağı üzere, yalnızca bilgi okunması yapılabilen belleklerdir. Bilginin sabit olarak tutulması istenilen ve sık olarak değişmesine gerek olmayan durumlarda / yerlerde kullanılırlar. Bilgisayarlar, yazar kasalar, güvenlik sistemleri ve ev aletleri ROM'ların kullanım yerlerinden bir kaçıdır.

ROM belleklere bilgi, üretici tarafından üretim sırasında veya daha sonra elektriksel olarak yüklenir. Yüklenen bilgiler uçucu değildir ve elektrik kesilse dahi silinmez. Bilgi yükleme işlemi, ROM'un programlaması veya yakılması olarak isimlendirilir.

ROM belleklerin oluşturulmasında en basit yöntemlerden birisi, Kod çözücü ve diyotlar kullanılarak oluşturulan diyot-ROM devresidir (Şekil 13.22). Şekildeki devrede, sekiz satırdan her birinde dört bit bilgi saklanabilir. Bir hattın seçilmesi ile, diyotlar ile saklanan dört bitlik bilgi çıkışlarda (D_3, D_2, D_1, D_0) gözükür.

Kod çözücü devre yardımıyla '0' nolu satırın seçilmesi durumunda; katotlarına (-) gerilim uygulanan diyotlar iletime geçer ve iletime geçen diyotların bağlı olduğu çıkışlarda (D_3, D_2, D_1) lojik '0' bilgisi gözükürken, D_0 çıkışında lojik '1' değeri gözükür. Bu durumda çıkışta; '0001' değeri okunur. Kod çözücü devresinin 5 nolu satırı seçmesi durumunda, diyotların iletime geçmesi nedeni ile çıkışta '1010' değeri okunur.

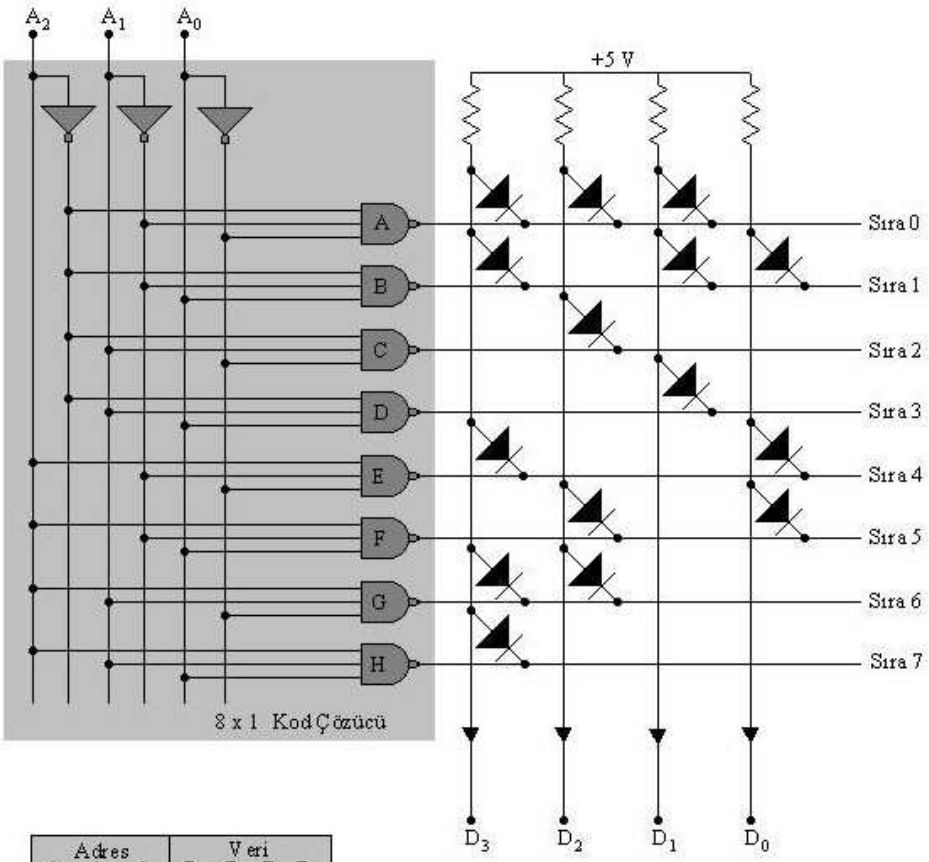
Şekil 13.22' deki doğruluk tablosunda, sekiz satırda okunacak değerler sıralanmaktadır. Satırları belirten değerler 'adres' olarak isimlendirilirken, satırların içerikleri 'veri' olarak tanımlanır. Adres seçme işlemi için kullanılan kod çözücü devresine üç adres seçme bilgisi uygulanır (A_2, A_1, A_0). Adres girişlerindeki değerlere göre kod çözücü devre çıkışlarından biri aktif olur.

Bazı tip ROM'lar sadece bir kez programlanabilme özelliğinde iken, bir kısmı elektriksel olarak silinip tekrar programlanabilirler. Programlanabilen ve silinebilen ROM'lar aşağıdaki alt gruplara ayrılırlar:

i- Programlanabilen ROM – PROM (Programmable Read Only Memory),

ii- Ultraviole ışınla silinip programlanabilen ROM-EPROM (Erasable PROM-EPROM),

iii- Elektrikle silinip programlanabilen ROM-EEPROM-E^2PROM (Electrically Erasable PROM).



Adres			Veri			
A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	0	1
0	0	1	0	1	0	0
0	1	0	1	0	1	1
0	1	1	1	1	0	1
1	0	0	0	1	1	0
1	0	1	1	0	1	0
1	1	0	0	0	1	1
1	1	1	0	1	1	1

(b)

Şekil 13.22. Dahili kod çözücülü ROM devresi lojik şeması.

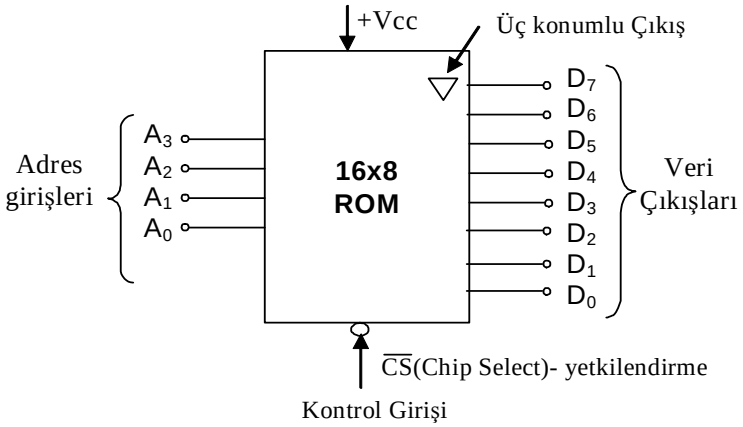
Boş olarak temin edilen bu üç tür ROM'lar kullanıcı tarafından istenilen şekilde programlanabilirler.

Şekil 13.23'de blok şema ile gösterilen ROM bellekler, üç farklı tip sinyal grubu girişine sahiptir: Adres girişleri, kontrol girişleri ve veri girişleri.

Şekil 13.23'deki ROM, 2^4= 16 farklı adres girişine sahip olduğundan 16 kelime saklayabilir. Her bir kelime 8 bit (1 bayt) olduğundan 8 veri çıkışı vardır ve bunun sonucu olarak bellek, 16x8 ROM olarak tanımlanır.

ROM belleklerin çoğunun veri çıkışı üç konumlu (tristate) yapıdadır. Bu özellik, bellek genişletilmesi sırasında birden fazla ROM entegresinin aynı veri yolunu (bus) kullanmasına imkan tanır.

$\overline{CS}$ (chip select) olarak ta isimlendirilen kontrol girişi, bir yetkilendirme (enable) girişidir ve ROM'un çıkışının yetkili veya yetkisiz olmasını belirler. Bazı üreticiler, kontrol girişini CS yerine, CE (chip enable) veya OE (Output enable) olarak göstermektedirler. Kontrol girişlerinin sayısı birden fazla olabilir. Bu girişlerden birisi, ROM'un kullanılmadığı durumlarda stand-by durumuna gelmesini sağlar.

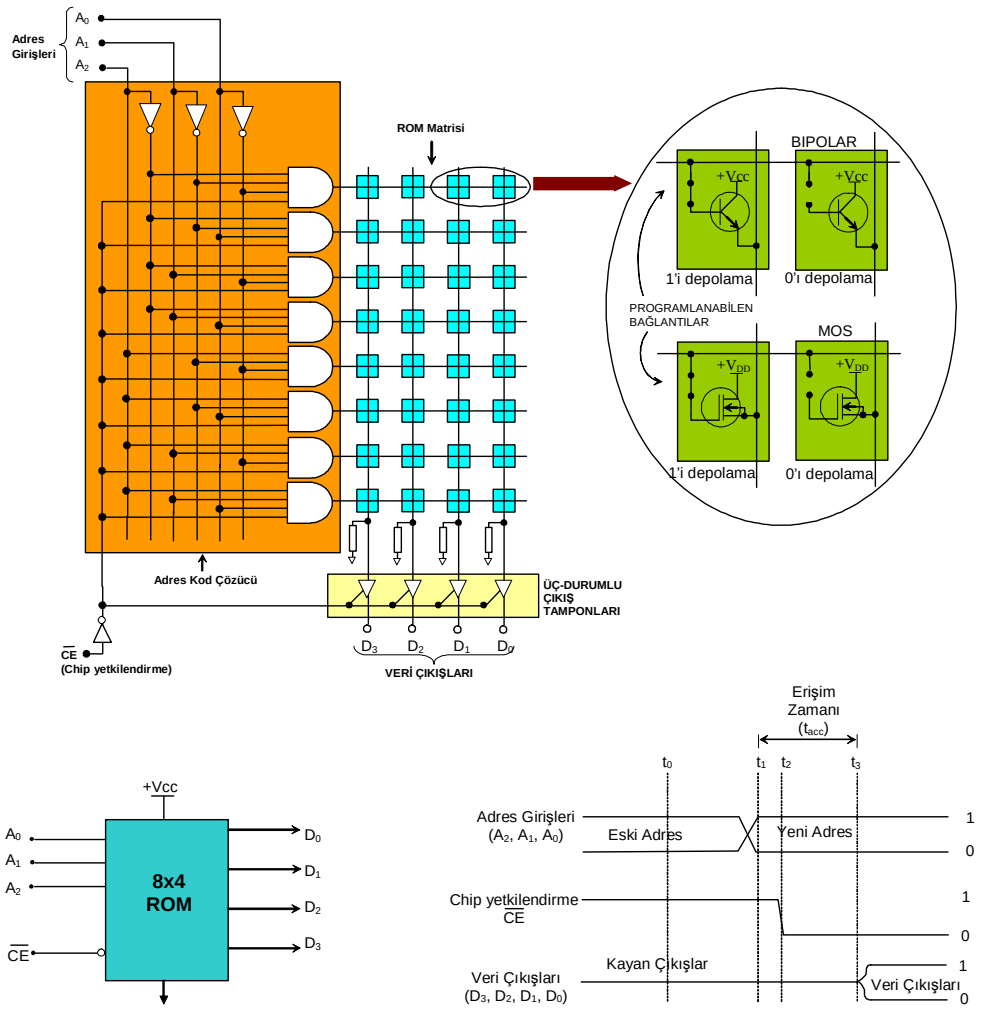


Şekil 13.23. ROM blok şeması.

Blok şemada, oku / yaz (read / write - R/W) girişi yoktur. Çünkü normal işlem sırasında ROM'a bilgi yazılması mümkün değildir. Bilgilerin okunması sırasında iki işlemin yapılması gerekir: Uygun adres girişlerinin uygulanması ve kontrol girişlerinin aktif hale getirilmesi. Örneğin; ROM içerisindeki '0111' adresindeki bilgileri okumak istiyorsak; $A_3A_2A_1A_0$=0111 değerlerini adres girişlerine uygulamamız ve $\overline{CS}$ girişini '0' yapmamız gerekir. Bu durumda ilgili adresteki bilgi, veri çıkışlarında gözükür. $\overline{CS}$'nin '1' olması durumunda, çıkış yüksek empedans (Hi-Z) konumunda olur.

Şekil 13.24'de yetkilendirme girişinin durumuna göre 8x4 ROM bellek entegresinde

‘okuma’ işlemi özetlenmektedir. Adres hatlarından gelen bilgilerin adres girişlerine uygulanması ve yetkilendirme girişine uygun sinyalin verilmesi ile, bellekten okunan bilgi ‘erişim zamanı’ kadar sonra veri çıkışlarında gözükür. Şekildeki zamanlama şemasında t_1 ve t_3 arasında geçen zaman, **‘erişim zamanını’** ifade eder. ROM tipi belleklerin erişim zamanı, transistorlerle oluşturulanlarında 10 nsn ile 50 nsn arasında değişirken, MOS teknoloji ile oluşturulanlarında 35 nsn ile 500 nsn arasında değişir.

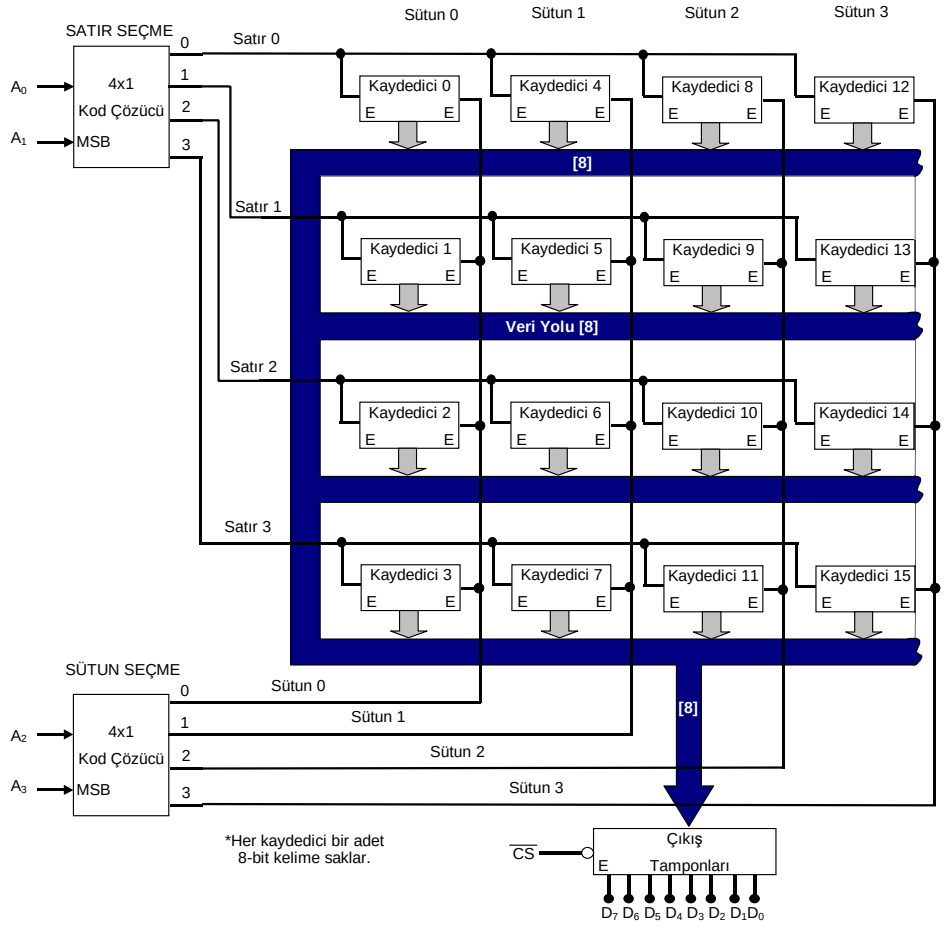


Şekil 13.24. ROM belleklerde okuma işlemi aşamaları.

13.6.1.1. ROM Belleklerin Yapısı

ROM bellekler dört temel parçadan oluşur: Satır kod çözücü, sütun kod çözücü, kaydediciler ve çıkış tamponları. Çok karışık bir yapıya sahip olan ROM entegreler, Şekil 13.25'deki gibi basitleştirilebilir.

16x8 kapasitedeki bir ROM belleğin yapısında bulunan genel kısımları Şekil 13.25'i referans alarak kısaca açıklayalım.



Şekil 13.25. 16x8 ROM'un yapısı.

i- Kaydediciler : Kaydediciler, ROM içerisinde veri depolamak için kullanılır. Her bir kaydedici, kelime uzunluğuna eşit sayıda bellek hücresi (örneğin, 8 bit veya 16 bit) içerir. Bellek hücreleri genelde kare matris dizisi şeklinde düzenlenir ve satır / sütun numaraları ile tanımlanır. Her bir kaydedicide bulunan veri çıkışları, tüm devreyi dolaşan dahili veri yoluna bağlanır. Her bir kaydedici, iki adet yetkilendirme (E) girişine sahiptir ve verinin yola konulabilmesi için her ikisinin '1' konumunda olması gerekir.

ii- Adres Kod çözücüleri : Adres kod çözücülüleri, **'satır ve sütun kod çözücüleri'** olarak isimlendirilir. Adres kodu olarak isimlendirilen $A_3A_2A_1A_0$ girişlerindeki bilgiler, dizideki hangi kaydedici bilgisinin veri yolu üzerine yerleştirileceğini belirler. A_1A_0 adres bitleri; satır seçme amacıyla satır kod çözücüsüne uygulanırken, A_3A_2 adres bitleri; bir sütunun seçilmesini sağlamak için sütun seçme kod çözücüsüne uygulanır. Adres bilgilerinin uygulanması ile, yalnızca bir kaydedici adres girişleri tarafından sütun ve satır olarak seçilir ve seçilen bu kaydedici yetkilendirilir.

Örneğin; '1101' adres girişleri ile, A_3A_2=11 olur ve 3. sütunu seçen hat aktif duruma gelir. A_1A_0 = 01 değerleri ile 1. satır aktif duruma gelir ve her iki E (yetkilendirme) girişi '1' olan 13 nolu kaydedici, bilgisini taşıta koyar.

iii- Çıkış Tamponları : Adres girişleri tarafından yetkilendirilen kaydedici, bilgilerini veri yoluna yerleştirir ve bu veriler, veriyi harici veri çıkışlarına aktaran çıkış tamponlarına uygulanır. $\overline{CS}$'nin '0' olması ile veriler çıkışlara aktarılırken, $\overline{CS}$'nin '1' olması ile tamponlar Hi-Z durumuna gelir ve D_0 -D_7 çıkışları oluşmaz.

Bellek entegresinde bulunan kısımları açıkladıktan sonra, bellek yapısı ile ilgili diğer özellikleri inceleyelim.

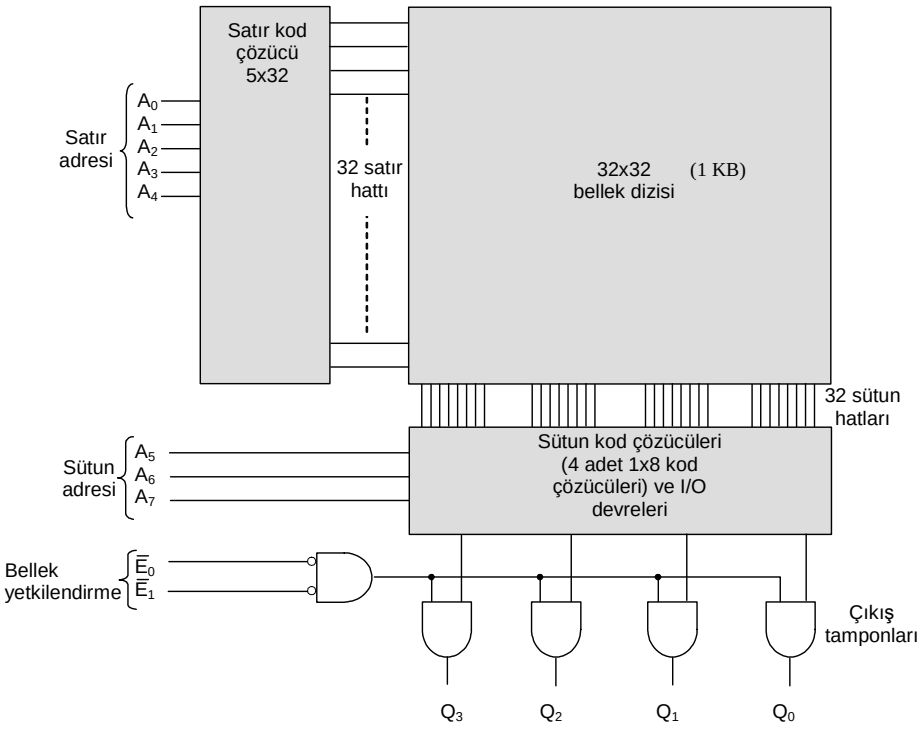
Şekil 13.25'de gösterilen yapı birçok entegre için benzerdir, ancak bellek kısımları depolanan bilginin sayısına göre şekil alır. Örneğin; Intel 2708 entegresi M0S ROM yapıdadır ve 1024 adet 8-bit bilgiyi depolar. Bu bilgiler, 64*16 dizisi şeklinde saklanır.

Örnek 1 : 4 KByte bilgiyi saklamak amacıyla kullanılan ve kare bellek dizisi şeklinde olan bir ROM'un yapısını açıklayalım:

4 KB bilgi 4x1024 = 4096 Byte olduğundan, ROM'da 4096 adet 8 bitlik veri saklanabilir. Bu durumda, her biri 8 bit saklama kapasitesinde olan 4096 adet kaydedici bulunması gerekir. $4096=64^2$ olduğundan, kaydedici dizisi 64x64 yapısında olur ve bellekte 64 satır–64 sütun bulunur. 64'de 1 kod çözücü için (satırlar için) 6 adres girişi gerekir. Aynı şekilde sütunlar içinde 6 adres girişi bulunur ve 2^{12}= 4096 olduğundan, 4096 farklı adres ortaya çıkar.

Burada anlatılan ROM tipi belleklerde, satır ve sütunlarda bulunan kod çözücü elemanlarla belirli bir adrese ulaşılır. Ulaşılan adreste bulunan bilgiler çıkışa aktarılır. Bilgilerin saklanmasında diyot-transistör veya MOSFET'lerden faydalanılır.

Örnek 2 : Şekil 13.26'da 32x32 matrisine göre çalışan ve 1024 bitlik bellek hücresi içeren, tipik bir ROM entegresinin blok şeması görülmektedir. 8 bitlik adres (A_0 ..A_7) yardımı ile bellekte bulunan 4 bitlik nipple'ler okunur. Bu entegrede, 8 bitlik adresten 5 tanesi ($A_0A_1A_2A_3A_4$) 32'de 1 çıkışa sahip kod çözücü yardımı ile uygun adresi belirler (satır olarak- 32xX). Kalan 3 adres girişi ile, ($A_5A_6A_7$) 4 adet 8'de 1 kod çözücülerden birisi seçilir. 4 adet 8'de 1 kod çözücünün çıkışı, bellek yetkilendirme (BY) girişleri yardımıyla yüksek empedanslı çıkış haline dönüştürür ve bu belleğin kapasitesini genişletme durumlarında kolaylık sağlar.



Şekil 13.26. 32x32 matrisli 1024 bit kapasiteli ROM'un blok şeması.

13.6.1.2. ROM Bellek Çeşitleri

Programlama, silinme ve yeniden programlama bakış açısı ile farklı yapıda ROM bellekler bulunmaktadır:

i- Maske programlı ROM'lar,

ii- Programlanabilir ROM'lar,

iii- Silinebilir-programlanabilir ROM'lar,

iv- Elektriksel olarak silinip programlanabilir ROM'lar.

Bu grupları, aynı prensibe göre çalışanları aynı başlık altında olmak üzere inceleyelim.

i- **Maske Programlı ROM'lar (MROM) :**

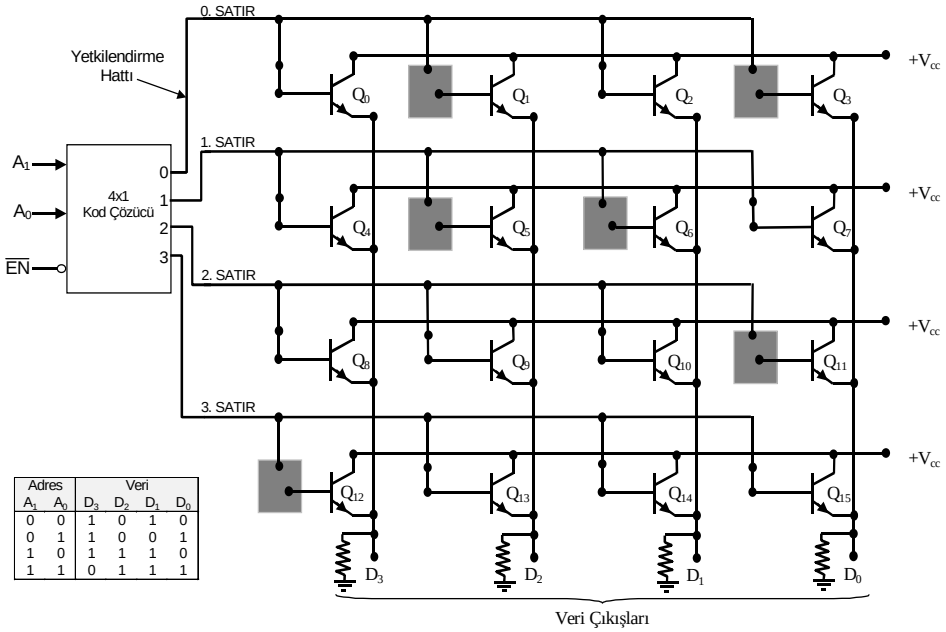
Maske programlı ROM bellek (MROM), tüketici isteğine bağlı olarak üretici tarafından programlanmış depolama bölgelerine sahiptir. Entegre içerisinde elektriksel bağlantı, mask olarak isimlendirilen negatif fotoğrafik malzemenin kullanılması ile sağlanır.

ROM içerisinde depolanan her bir bilgi grubu (seti) için özel bir maske gerektiğinden ve bu maskeler pahalı olduğundan, bu tip ROM bellekler yalnızca aynı tip bilgilerin (belirli matematiksel tablolar ve göstergeler için karakter jeneratör kodları gibi) ROM içerisinde saklanması durumunda kullanılır. Bu tip belleklerin en önemli dezavantajı; depolanan verinin değiştirilmesi gerektiğinde, bellek yapısının buna imkan tanımamasıdır. Bunun yanında, çok büyük hacimli belirli tip verileri saklamak için kullanılacak en ekonomik bellek türü, MROM belleklerdir. Maske programlı belleklerin mahsurunu ortadan kaldırmak amacıyla, kullanıcı tarafından programlanabilen ROM bellekler geliştirilmiştir.

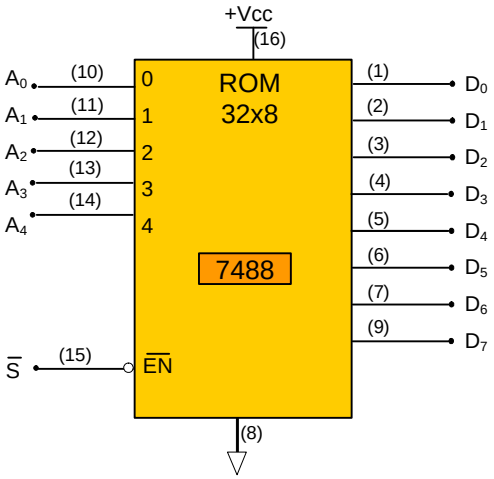
MROM belleklerde bilgi saklamak için diyot ve transistörlerden faydalanılır. Diyot ve transistörler, '0' veya '1' değerini saklayacak şekilde şekillendirilir. Şekil 13.27'de dört sütundan oluşan ve her bir sütunun dört hücre içerdiği, 16 bellek hücreli küçük bir bipolar MROM belleğin yapısı görülmektedir. Her bir hücrenin NPN tipi bir transistörle ifade edildiği bu sistemde, transistörler kollektörü şase tipte bağlamışlardır ve beyz giriş, emiter ise çıkış olarak kullanılmaktadır. Her bir satır 4 bit kaydedici olarak düşünülebilir ve transistor beyzlerinin ilgili satırın yetkilendirme hattına bağlanması 1'i temsil ederken, bağlanmaması 0'ı temsil eder. Her bir beyz bağlantısının durumu, tüketicinin isteğine uygun olarak üretim sırasında fotoğrafik maske tarafından kontrol edilir.

Şekil 13.27'den görüleceği üzere, birden fazla transistörün çıkışı (emiteri) aynı sütun hattına bağlıdır ve bu problem olarak düşünülebilir. Ancak herhangi bir anda yalnızca bir satıra bağlı eleman aktif olacağından, bu durum bir problem oluşturmaz. Hangi satırın aktif olacağı, A_1A_0 adres girişlerinin durumuna göre kod çözücü tarafından belirlenir. Kod çözücü devreye uygulanan yetkilendirme ($\overline{EN}$) girişinin '1' yapılması ile, kod çözücü çıkışları '0' konumunu alır ve tüm transistörler beyz gerilimi olmaması nedeniyle kesin durumuna

geçerler. Bu durumda tüm veri çıkışları ‘0’ konumundadır. Yetkilendirme girişinin ‘0’ olması durumunda, adres girişlerinin durumuna bağlı olarak uygun satır belirlenir ve ilgili verinin çıkışlarda oluşması sağlanır.



Şekil 13.27. 16 bellek hücreli MROM belleğin yapısı.



Şekil13.28. 32x8 kapasiteli 7488 entegresinin lojik şeması.

Bipolar MROM belleklerin en yaygın olanlarından birisi, 256x4 bellek yapısına sahip 74187 entegresidir. Diğer taraftan, 7488A ROM entegresi 32x8 kapasiteli bipolar ROM yapıda ve TMS 47256 entegresi 32Kx8 kapasitede NMOS yapıda belleklerdir. Şekil 13.28'de, 32x8 kapasiteli 7448 ROM entegresinin lojik şeması görülmektedir.

MOSFET yapıdaki bellekler, üç konumlu çıkışları nedeniyle bilgisayar veri taşıtlarına kolay bağlanırlar ve 'stand-by' durumları nedeniyle belleğin kullanılmadığı anlarda çok az güç tüketimine sahip olurlar.

ii- Programlanabilir ROM'lar (PROM)

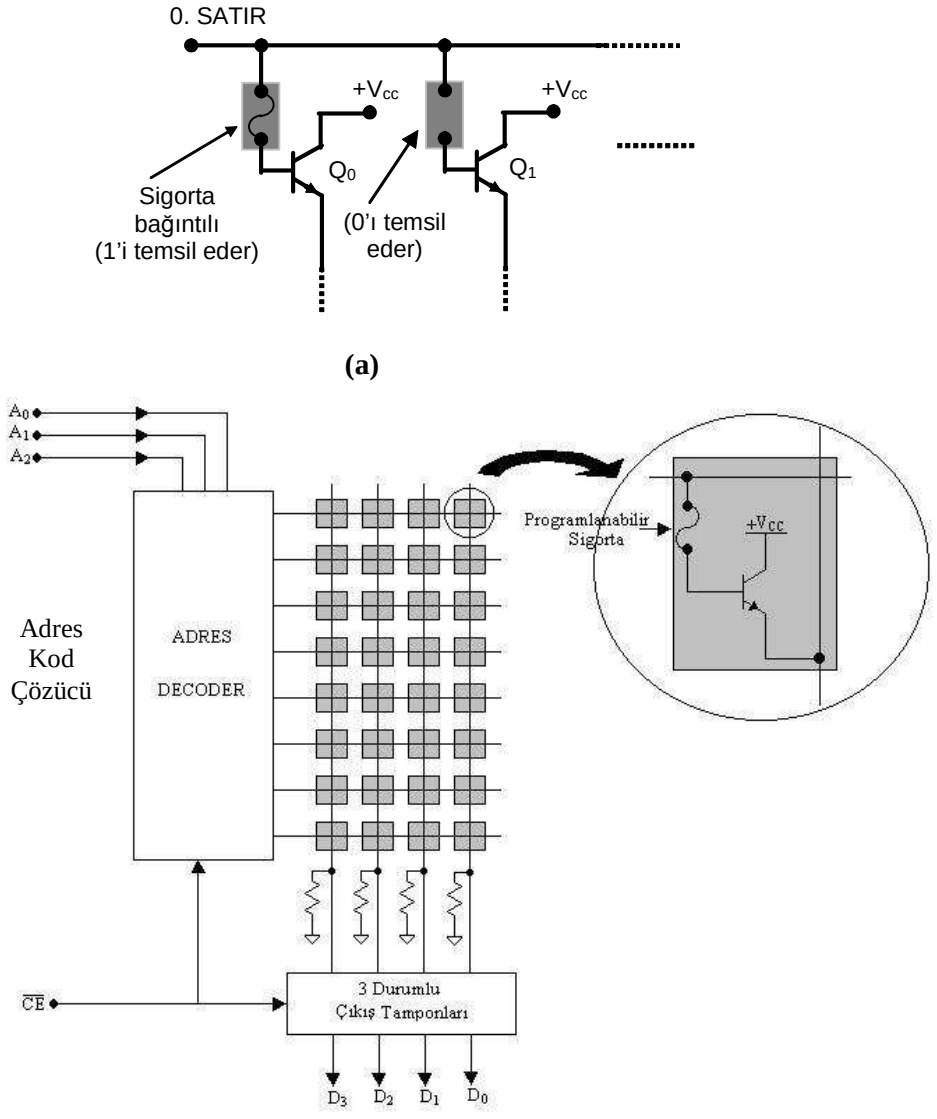
Maske – programlı ROM'ların çok pahalı ve yalnızca çok geniş hacimli uygulamalarda uygulanabilir olması nedeniyle; daha küçük çaplı uygulamalar için, üretim sırasında programlanmayan, kullanıma göre tüketici tarafından programlanabilen 'sigorta bağıntılı' (fusible-link) PROM'lar geliştirilmiştir. Bu tip bellekler, MROM bellekler gibi bir kere programlanınca silinemez ve yeniden programlanamaz. Bu nedenle, programlamada bir hata oluştuğu veya içerisindeki programın değişmesi gerektiği durumlarda, MROM tipi belleklerin atılması gerekir. Bu özellik nedeniyle, bu tip bellekler 'bir kere programlanabilir ROM'lar' olarak adlandırılır. Maske programlı ROM'lara üstünlüğü, kullanıcı tarafından programlanabilmesi olan PROM'ların, programlama işleminde sigortaları attırmak için 50 mA'lik akım kullanılır.

Sigorta bağlantılı PROM, yapı olarak MROM'lara benzer ve '1' ve '0' durumlarını oluşturmak için sigortalı kısım olduğu gibi bırakılır (1'i temsil eder) veya açık devre (0'ı temsil eder) haline getirilir (Şekil 13.29.a). Üretici tarafından Şekil 13.29.b'de görülen şekilde imal edilen PROM'lar, saklanmak istenen veriye göre şekillendirilirken; sigortanın artmasını sağlayacak bir akımı oluşturacak gerilim kontrollü olarak devreye uygulanır. Devrede oluşan akım, bir sigortanın atması şeklinde etki oluşturarak açık devre oluşmasına neden olur. Açık devre olan kısım tekrar eski haline dönüştürülemez. Hala hazırda, PROM bellek elemanlarının bilgisayar yardımıyla istenilen şekilde programlanmasını sağlayan çok sayıda programlayıcı devresi bulunmaktadır.

PROM'un programlanması ile oluşan yapı, PROM'un kapasitesine göre şekillenir. 256 Bit kapasitesindeki bir PROM'da; her bir hatta 8 bitlik bilgi saklanması durumunda, adres kod çözücü beş giriş hattına karşılık $2^5 = 32$ çıkış hattını içerir. Her bir çıkış hattı ile seçilen bölgede 8 bitlik bölgeye ulaşıldığından PROM'un kapasitesi 32*8 = 256 bit olur

PROM entegreler bipolar transistör yapısında (düşük yoğunluk, yüksek hız ve yüksek güç tüketimi) veya MOS yapısında (yüksek yoğunluk, düşük hız ve düşük güç tüketimi) imal edilmektedirler. Bu durumda, Bipolar transistorler kullanılarak imal edilen PROM'lar, MOS

teknolojisi kullanan PROM'lara göre daha hızlı ve ucuzdur. Buna karşılık, MOS teknolojisi kullanılan entegreler, daha yüksek bellek kapasitesine sahip olabilir.



Şekil 13.29. PROM'larda bir hücreyi programlamak için kullanılan sigorta bağıntıları.

74186 entegresi; 64x8 kapasitesinde, erişim zamanı 50 nsn olan popüler bir PROM entegresidir. TBP285166 entegresi; 2Kx8 kapasitesinde, TMS27PC256 entegresi ise, 32Kx8 kapasitesinde, 120 nsn erişim zamanına ve 'stand-by' konumunda 1.4 mW güç tüketimine sahip PROM tipi entegrelerdir.

PROM'ların kullanım yerine örnek olarak, çamaşır makinesi, mikro dalga fırın, bulaşık makinesi, vb. uygulama yerleri verilebilir.

iii- Silinebilir – Programlanabilir ROM (Erasable Programmable ROM–EPROM) :

Silinip-programlanabilir ROM bellekler (EPROM), kullanıcı tarafından programlanabilen ve istenildiği durumlarda silinerek tekrar programlama imkanı tanıyan bellek türleridir. Bir kere programlanan EPROM'da, saklanan bilgi uçucu değildir ve depolanan bilgi sonsuza kadar korunur. EPROM'un programlanması için özel programlama devresi gereklidir ve programlama devresi için 10 ~ 25 V arasında bir gerilim kullanılır. Bu gerilim ile, entegrenin içerisindeki transistörler 0'ı temsil etmesi için iletim (ON) konumuna getirilir veya 1'i temsil etmesi için olduğu konumda bırakılır.

Her bir adresin, içerisinde saklanacak veriye göre programlandığı EPROM entegresi içerisindeki bilgilerin silinmesi istenildiği durumlarda; entegrenin üzerindeki pencere açılarak EPROM'a yaklaşık 20 dakika ultraviyole (UV) ışık uygulanır. Ultraviyole ışık EPROM içerisinde bir akım oluşturarak, silikon elemanın yapısını imalat sırasındaki konumuna (tüm transistörler OFF) geri dönderir ve tüm hücreler '1' değerini alır. Programlanmış bir EPROM'un yalnızca belirli bir kısmını yeniden programlama olanağı yoktur. Silme işlemi ile tüm bilgi saklama hücreleri silinir ('1' duruma getirilir) ve silinen EPROM tüm olarak tekrar programlanabilir. EPROM programlandıktan sonra üzeri 'sticker' ile kapatılır. Silinmesi için üzeri açılarak entegrenin üzerindeki pencereden ultraviyole ışığın içeri girmesi sağlanır.

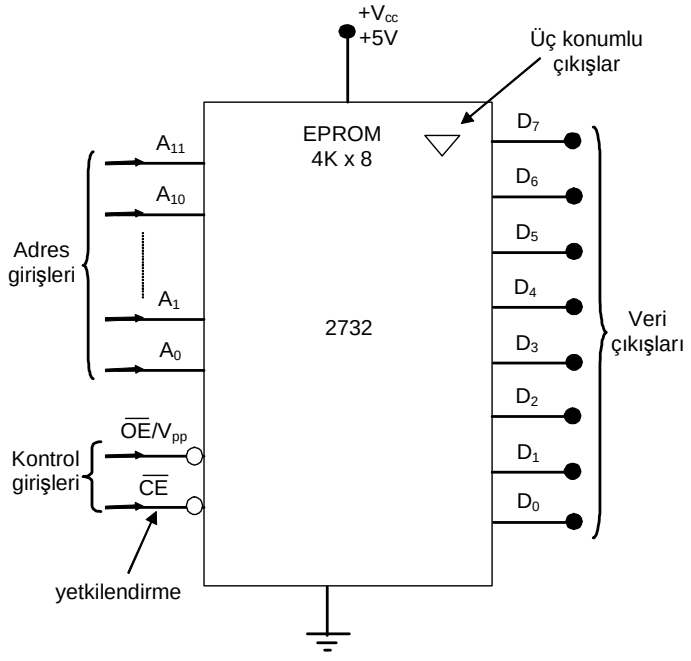
Piyasada çok farklı kapasiteye ve erişim zamanına sahip EPROM elemanları bulunmaktadır. Piyasada 27XX serisi EPROM'lar yaygın olarak kullanılmaktadır. 27XX serisi EPROM'lar 8 bit genişliğinde bilgi depolama kapasitesindedirler. Şekil 13.30'da yaygın olarak kullanılan 27XX serisi entegrelerin bellek yapısı ve bellek kapasitesi tablo şeklinde verilmektedir.

2732 EPROM entegresi; 4Kx8 kapasitesinde, +5V güç kaynağı ile çalışan 12 adres girişine (2^{12}=4096 olduğundan) ve 8 veri çıkışına sahip bir bellek elemanıdır. 2732'de iki kontrol girişi bulunur (Şekil 13.31). $\overline{CE}$; yetkilendirme kontrol girişi entegrenin güç tüketimini azalmayı sağlayan 'standby' modunda çalışma için kullanılırken, $\overline{OE}$ / V_{pp}; kontrol girişi iki farklı amaç için kullanılır. OE çıkış yetkilendirme (Output Enable) girişi; entegrenin veri çıkış tamponlarını kontrol için kullanılır ve entegrenin bir mikroişlemcinin veri yoluna karmaşıklık oluşmadan bağlanmasını sağlar. V_{pp} ise, programlama işlemi sırasında gerekli özel programlama gerilimidir (21V). V_{pp} gerilimi yardımıyla, programlama modunda

EPROM'un içerisine yeni veri girişi sağlanır. Programlama işlemi ile EPROM içerisindeki her bir adrese 8 bitlik kelime yerleştirilir.

EPROM	Bellek Yapısı	Bellek Kapasitesi
2708	1024*8	8192
2716	2048*8	16384
2732	4096*8	32768
2764	8192*8	65536
27128	16384*8	131072
27256	32768*8	262144
27512	65536*8	524288

Şekil 13.30. Yaygın kullanılan 27XX serisi EPROM entegreleri.



Şekil 13.31. 2732 EPROM entegresinin lojik sembolü.

Programlama işleminde; adres girişlerine bilgi yazılacak bellek bölgesini seçecek değerler ve veri girişlerine saklanmak istenen değerler uygulanırken, Vpp girişine 21V programlama gerilimi ve yetkilendirme girişine entegreyi seçmek için '0' değeri uygulanır. İki kontrol girişinin (Vpp=21V, CE=0) 50 msn entegreye uygulanması gerekir. Aynı işlemin tüm bellek bölgeleri için tekrarlanması gerekir.

Okuma işleminde; 'CE=0' bilgisi yanında 'OE=0' bilgisinin uygulanması gerekir. Aynı anda içeriği okunmak istenen bellek bölgesini temsil eden değerler adresleme girişlerine uygulanır. 2732 entegresi aktif durumda iken 500 mW güç tüketirken, 'standby' durumunda 175 mW güç tüketir.

EPROM içerisine yerleştirilecek verilerin en hızlı ve uygun yerleştirme şekli, bilgisayar ve EPROM programcısı yardımıyla yapılan veri yüklenmesi yöntemidir. EPROM belleklerin iki dezavantajları; silinmek için soketinden çıkarılma ve tek bir bitin değiştirilmesi için tüm entegrenin yeniden programlanması zorunluluklarıdır.

iv- Elektriksel Olarak Silinebilir PROM (Electrically Erasable PROM – EEPROM)

EPROM bellek türlerinin sahip olduğu iki dezavantajı ortadan kaldıran ve EPROM'un gelişmiş şekli olarak 1980'li yıllarda imal edilen bellek türü, kısaca EEPROM olarak isimlendirilen ve elektrik ile silinebilen PROM tipi belleklerdir. EEPROM belleklerde ultraviyole ışığa ve entegrenin bulunduğu devreden çıkarılmasına gerek yoktur.Bu tip belleklerde, MOSFET bellek hücresinin kanalı (drain) üzerine ince bir oksit (oxide) bölgesi eklenerek, elektriksel olarak silinebilme özelliği kazandırılır.

Programlama işleminde; MOSFET'in kapı ve kanal uçları arasına 10 msn süre ile 21V'luk bir gerilim uygulanarak, kapı-kanal uçlarında bir şarj oluşması sağlanır. Gerilim kesilse dahi oluşan şarj devam eder ve bilgi saklanır. Uygulanan gerilimin ters çevrilmesi, oluşan şarjın ortadan kalkmasına ve hücredeki bilginin silinmesine neden olur. Şarj işlemi çok küçük akım gerektirdiğinden, EEPROM'un programlanması ve silinmesi işlemi entegre devrede iken gerçekleştirilebilir.

EEPROM belleklerin EPROM'lara göre üstünlükleri;

i- Elektriksel olarak silinebilme kabiliyeti,

ii- Bellek dizisini içerisindeki bölgelerin birbirinden bağımsız olarak silinip programlanabilmesi,

iii- EPROM'da ortalama 50 msn olan bir bellek bölgesini silme süresinin 10 msn olması,

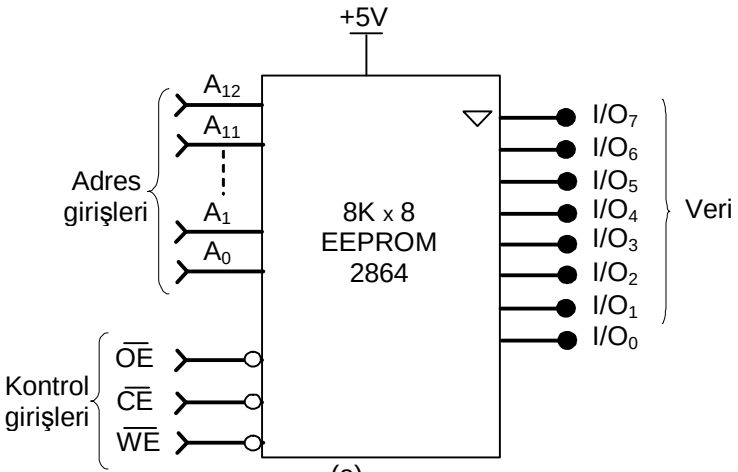
şeklinde sıralanabilir. EEPROM'ların en önemli mahsuru ise; EPROM'lara göre daha pahalı olmalarıdır.

Ayrıca EEPROM bellekler 21V/10msn bir sinyal ile doğrudan silinebilir. Bununla beraber,

EPROM bellekler EEPROM'lara göre daha ucuzdurlar ve daha büyük bellek yoğunluğuna sahiptirler.

Intel 2816 bellek entegresi; 2Kx8 kapasitesinde 250 nsn erişim zamanına sahip ilk EEPROM'dur. EEPROM'lar zaman içerisinde kapasite ve erişim zamanı olarak geliştirilmişlerdir. Şekil 13.32'de sembolü ve işlem modları görülen 8264 EEPROM bellek entegresi, 8Kx8 kapasitesine sahiptir. Yüksek gerilim (21V) üreteci entegrenin içerisindedir ve bu nedenle +5V besleme geriliminden başka bir besleme devresine ihtiyaç yoktur.

Bu bellekte veri pinleri I/0 olarak isimlendirilmiştir ve işlem moduna göre veri girişi veya veri çıkışı için kullanılırlar. Üç adet kontrol girişi, kullanılan işlem modunu belirler: CE girişi entegreyi yetkilendirmek (yetkisizlendirmede entegre standby moduna geçer) için kullanılır~~ken~~, OE girişi veri çıkış tamponlarını yetkilendirmek için kullanılır. Okuma işlemi sırasında OE'nin '0' yapılması ile çıkış tamponları yetkilendirilirken, yazma işlemi sırasında OE=1 yapılm~~ası~~ ile çıkış tamponları yetkisizlendirilirler ve veriler I/0 pinlerine uygulanabilir. WE girişi ile yazma (write) veya okuma (read) modunun birisi seçilir. Yazma modunda; WE = 0 girişi ile birlikte, yazılması istenilen veriler I/0 pinlerine uygulanır (Şekil 13.32.b).



(a)

	Girişler			
Mod	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	Çıkışlar
OKUMA	V_{IL}	V_{IL}	V_{IH}	$VERİ_{ÇIKIŞI}$
YAZMA	V_{IL}	V_{IH}	V_{IL}	$VERİ_{GİRİŞİ}$
STANDBY	V_{IH}	X	X	High Z

(b)

Şekil 13.32. 2864 EEPROM bellek entegresi sembolü ve işlem modları.

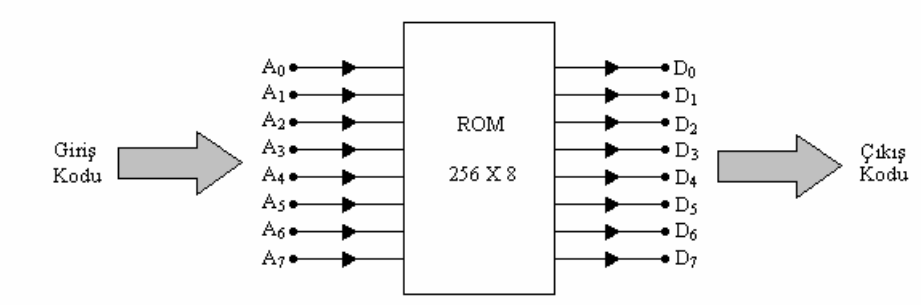
13.6.1.3. ROM Bellek Uygulama Alanları

ROM bellekler, bilgilerin kaybolmadan ve sık olarak değiştirilmeden uzun süre korunması istenilen uygulamalarda yaygın olarak kullanılırlar. ROM uygulama alanlarından bazılarını aşağıdaki şekilde özetleyebiliriz.

i- Mikrobilgisayar Programlarının Saklanması (Firmware) : Mikrobilgisayarlar ROM'un yaygın olarak kullanıldığı uygulamalardan birisidir. Mikrobilgisayar içeren cihazların (örneğin, elektronik oyunlar, yazar kasalar, elektronik teraziler, vb.) kontrol programlarını saklamak amacıyla ROM'lar kullanılır. PC bilgisayarlarda ise, işletim sistemlerine ait programları ve programlama dillerine ait yorumlayıcıları (interpreter) saklamak amacıyla ROM'lar kullanılır. ROM'larda saklanan ve değiştirme imkanı olmayan mikrobilgisayar programları **'firmware'** olarak isimlendirilirken, oku / yaz belleklerde (RWM'lerde) saklanan ve kolayca değiştirilebilen programlar **'software'** olarak isimlendirilir.

ii- Veri Tabloları (Data Tables) : ROM'lar sık olarak değişiklik gerektirmeyen veri tablolarını saklamak için kullanılır. Veri tablolarına örnek olarak, trigonometrik tablolar ve kod dönüştürme tabloları gösterilebilir. Trigonometrik fonksiyonların bulunduğu ' look-up' tablolarını içeren ROM entegreleri piyasada hazır halde bulunmaktadırlar. Örneğin; 'National Semiconductor' firmasının ürettiği MM4220BM entegresi, O^0 ile 90^0 arasındaki açıların sinüslerini veren tabloyu içerir. 128x8 olarak dizayn edilen bu ROM, 7 adres girişine ve 8 veri çıkışına sahiptir. Her bir adreste, $0,7^0$ aralıklarla açıların sinüs değerleri bulunur.

iii- Veri Çevirici – Dönüştürücüler (Data Converter) : Veri dönüştürücü devresi, bir kod ile ifade edilen veriyi alarak, farklı bir kodla çıkış üreten devredir. Kod dönüştürme yapmanın en kolay yöntemlerinden birisi, ROM kullanımıdır. ROM kullanarak yapılan dönüştürme işleminde, belirli bir adreste depolanan bilgi, dönüştürülmesi istenen yeni kodda bir veri çıkışı üretir (Şekil 13.30). Giriş kodu, ROM entegresinin adres girişlerine uygulanır ve giriş kodları çözülerek bellek içerisindeki bir bellek bölgesi seçilir. Seçilen bellek bölgesindeki bilgi çıkışa aktarılır.

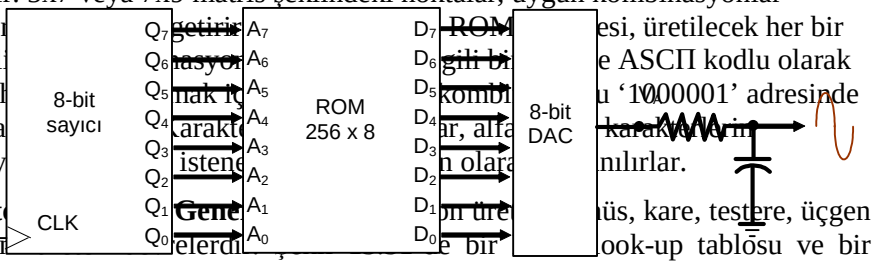


Şekil 13.30. ROM entegrenin kod çevirici olarak kullanılması.

TTL 74185 entegresi; 6-bit binary girişi için Binary-BCD kod dönüşümünü yapan bilgileri içeren ROM bellektir. Piyasada, 7 bit ASCII kodunu 7 bit EBCDIC'ye çeviren veya bu çevirimin tersini yapan entegreler bulunmaktadır.

iv- Karakter Üreteçleri/Keyboard (Character Generators):

Bir ekranda oluşturulan alfasayısal karakterleri (harfler, rakamlar, vb.) incelediğimiz zaman, karakterlerin oluşmasında küçük noktaların (dot'ların) durumlarından faydalandığını görürüz. Oluşturulan karakterlerin şekline göre noktalardan bir kısmı parlak iken, bir kısmı karanlık olarak kalır. 5x7 veya 7x9 matris şeklindeki noktalar, uygun kombinasyonlar oluşturarak karakter getirisi, üretilecek her bir karakter için gerekli asyon e ASCII kodlu olarak saklar. Örneğin; A harfi mak için u '1000001' adresinde ASCII 1000001 olarak karakter ar, alfasayısal karakterlerin görüntülenmesi veya istenen n olarak kullanılırlar.



v- Fonksiyon Üreteci (Function Generator): Genellikle sinüs, kare, testere, üçgen vb. dalga formlarını elerdi e bir look-up tablosu ve bir dijital-analog çevirici (DAC) kullanılarak sinüs dalgasının nasıl üretildiği görülmektedir. ROM, her biri farklı bir dalga şekli değerine karşılık gelen 256 farklı 8 bitlik değeri depolamaktadır. 8 bitlik sayıcı, ROM için ardışıl (sequential) adres girişleri üretmek amacıyla bir tetikleme sinyali tarafından devamlı olarak tetiklenir. Sayıcı çıkışı 256 farklı adresi tanımlayabildiğinden, ROM devresi DAC için 256 farklı veri çıkışı verir. DAC çıkışı, veri noktalama karşılık gelen 256 farklı analog gerilim değerini ifade eden dalga şeklinde olur. Alçak geçiren filtre, düzgün bir sinyal şekli elde etmek için DAC çıkışındaki değerleri yuvarlar ve şekli düzeltir.

Şekil 13.31. Bir ROM'un ve bir DAC'ın kullanıldığı fonksiyon üreteci devresi.

13.6.2. Yarıiletken Oku / Yaz Bellekler (RWM'lar)

Lojik bir devreye yerleştirildikten sonra sonsuz sayıda okuma veya yazma işleminin gerçekleştirebildiği bellekler, **'oku / yaz (RWM) bellekler'** olarak isimlendirilir. Bilgilerin sabit olarak kaydedildiği ROM'lardan farklı olarak, oku/yaz belleklerde bilgiler geçici olarak saklanırlar. Bir bellek bölgesine bilginin saklanması 'yazma' işlemine karşılık gelirken, bir bellek bölgesinde saklanan bilginin alınması işlemi 'okuma' işlemini temsil eder.

Oku/yaz bellekler genel olarak, sıralı erişimli bellekler ve rasgele erişimli bellekler olarak iki gruba ayrılır. RWM'ler daha çok rasgele erişimli bellekler (RAM) olarak tanınırlar. RAM bellekleri geniş şekilde incelenmeden önce, sıralı ve rasgele erişimli belleklerin genel özelliklerini özetleyelim.

13.6.2.1 Sıralı Erişimli Oku / Yaz Bellekler

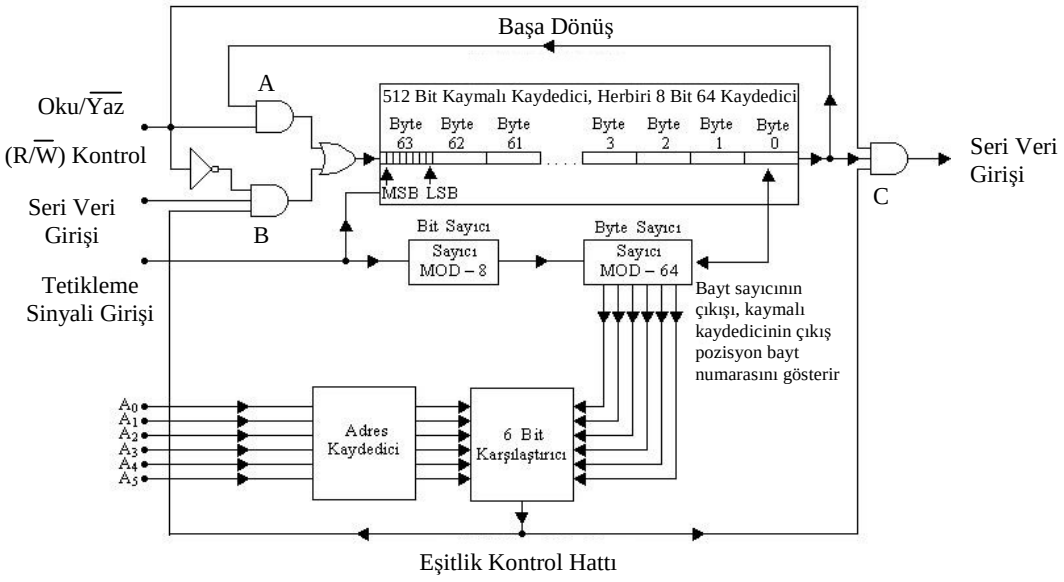
Daha önceki bölümlerde açıklanan kaymalı kaydediciler yapısında çalışan bellekler, **'sıralı erişimli bellekler'** veya **'seri bellekler'** (Sequential Access Memories – SAM) olarak isimlendirilirler. Şekil 13.32'de, 64 bayt'lık (64x8 bit= 512 bit) bilgiyi saklama kapasitesine sahip SAM belleğin yapısı görülmektedir.

Bellekte saklanacak bayt'lar 0 ile 63 arasındaki sayılar ile belirtilir ve 512 bitlik bilgi kaydediciler içerisinde seri olarak saklanır. 8 bitlik bilgileri seri olarak saklamak için, her bir baytın saklanacağı bellek bölgesini temsil eden 6 bitlik adres bilgisi kullanılır. 6 bitlik adres bilgisi adres kaydedicisine yerleştirilir ve karşılaştırıcının girişlerinden birisine uygulanır.

Kaymalı kaydedici devresine uygulanan her 'clock' sinyali ile, kaymalı kaydedicideki bilgi bir bit sağa kaydırılır. 'Clock' sinyali aynı anda Mod-8 bit sayıcıya uygulanarak 8 bit sağa kayma tamamlanır. Mod-8 bit sayıcı devresinin çıkışı bayt-sayıcı devresine (Mod-64) uygulanarak her 8 bitlik (1 bayt) işlem sonucunda artması sağlanır. Bu işlem ile, bayt-sayıcı devresi kaydırılan bilginin bayt numarasını sayar ve çıkışlarında işlem yapılan bayt numarasını gösterir.

Örneğin; '001001' nolu (desimal 9) adres bölgesine erişmek için, '001001' bilgisi adres kaydedicisine yüklenir ve karşılaştırıcıya uygulanır. 9 nolu adres bölgesindeki bilgiye erişmek için (okuma işlemi), oku / yaz kontrol girişi '1' yapılır ve bu değer A ile C isimli 'VE' kapılarını aktif yapar. Kaymalı kaydedicide bulunan veri baytı sağa kaydırılır ve 9 nolu bayt kaymalı kaydedicinin çıkışında gözükünceye kadar, 'A' isimli 'VE' kapısı girişine '1' uygulanır. Bayt sayıcının içeriği ile kaymalı kaydedicinin çıkışındaki bayt numarası aynı olduğundan (bayt sayıcının içeriği 9'dur), karşılaştırıcının her iki girişi eşittir ve karşılaştırıcı çıkışı aktif (lojik '1') olur. Çıkıştaki bu sinyal C kapısını yetkilendirir ve 9 nolu bellek bölgesinde bulunan bilginin (8 bit) seri olarak bellek dışına kaydırılmasını sağlar. 8 adet 'clock' sinyali oluştuktan sonra, 9 nolu bellek bölgesinin içeriğinin tamamı bellek çıkışında gözükür ve bayt sayıcının içeriği artar.

8 bitlik bir bilgiyi 9 nolu adres bölgesine kaydetmek (yazmak) için; ‘9’ sayısı adres kaydedicisine yüklenir ve kaymalı kaydedicideki bilgiler 9 nolu bayt kaymalı kaydedicinin çıkışına gelinceye kadar kaydırılır. Bu durumda, bayt sayıcının değeri ‘9’ olur ve karşılaştırıcı çıkışındaki ‘eşitlik kontrol hattı’ aktif (1) olarak B ve C kapılarına uygulanır. Bu anda oku/yaz kontrol hattının ‘0’ olması nedeni ile, A ve C kapıları yetkisizlendirilirken B kapısı yetkilendirilir. Yazılacak olan 8 bitlik bilgi kaydırılarak 9 nolu bellek bölgesine yerleştirilir. Yazma işlemi sırasında, daha önce bulunan bilgi A kapısının aktif olmaması nedeni ile geri yüklenemez ve kaybolur.



Şekil 13.32. 64 x 8 sıralı erişimli oku / yaz belleğin yapısı.

Açıklanan işlemlerden, seri belleklerde bir bellek bölgesine erişmek için ulaşılacak bellek bölgesine kadar tüm bellek bölgelerinin sıra ile adımlanması (geçilmesi) gerektiği görülür. Bu nedenle, seri bellekler **‘sıralı erişimli bellekler’** (Sequential access memory-SAM) olarak isimlendirilir.

Sıralı erişimli belleklerin sakıncası, bellek erişim zamanının büyük olmasıdır. Bununla beraber, yüksek hızlı erişimin gerekli olmadığı, düşük maliyetin önemsendiği uygulamalarda SAM bellekler kullanılmaktadır. Manyetik kabarcıklı bellekler, özel uygulamalarda kullanılan SAM bellek yapısındaki belleklerdir.

13.6.2.2 Rasgele Erişimli Bellekler (RAM)

Her hangi bir bellek bölgesine erişimin doğrudan sağlanabildiği ve tüm bellek bölgelerine

erişimin eşit zaman aldığı bellekler, **‘paralel bellekler’** veya **‘rasgele erişimli bellekler’** (Random Access Memomories – RAM) olarak isimlendirilir. RAM belleklerin yapısında bulunan kod çözücü devresinin çıkışı bir bellek bölgesini yetkilendirir ve kullanıcıya rasgele bir bellek bölgesine erişme imkanı sunar.

İsmini, bilgi saklanan bölgeye erişim şeklinden alsa da, günlük kullanımda RAM denildiği zaman okunabilen / yazılabilen (Read / Write Memory-RWM) bellek çeşidi kastedilmektedir. Bu durumda RAM belleğin tarifini yaparsak; programları veya bilgileri geçici süre saklamak için kullanılan, saklanan bilgilerin sonsuz kere değiştirilebildiği (silinip - yenisinin yazılabildiği), okuma ve yazma işlemlerinin aynı kolaylıkla yapılabildiği, enerjisinin kesilmesiyle bilgilerin kaybolduğu (volatile) bellek türü, **‘RAM bellek’** olarak isimlendirilir.

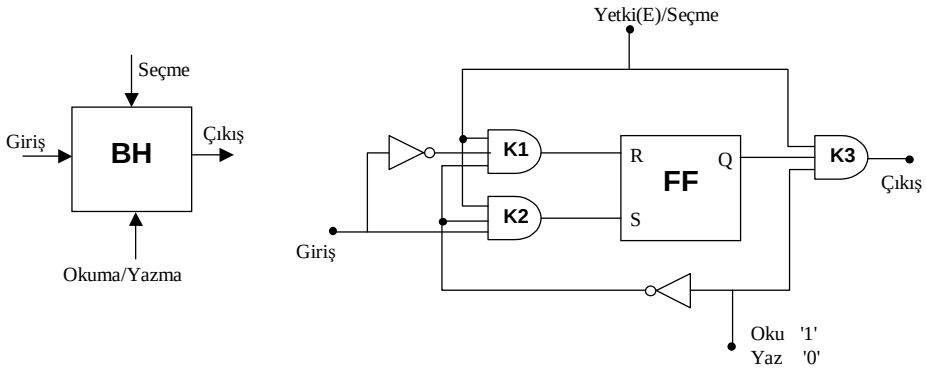
Rasgele erişimli bellekler küçük erişim zamanları olması nedeni ile sıralı erişimli belleklere göre daha yaygın olarak kullanılırlar. ROM bellekler yapı olarak rasgele erişimli bellek yapısında olmasına rağmen, rasgele erişim terimi yalnızca oku / yaz bellekler ile (RAM) birlikte anılmaktadır.

RAM’ların en yaygın kullanım yeri, bilgisayarda programların çalışması sırasında bilgilerin geçici olarak saklanması olduğundan, okuma ve yazma işlemlerinin çok hızlı olarak gerçekleştirilmesi gereklidir. RAM belleklere erişim sistemi ve okuma / yazma kolaylığı, RAM belleğe bu özelliği kazandırır.

RAM belleklerin özelliklerini açıklamada, ROM bellekler kısmında kullanılan terimlerden faydalanılır. Örneğin; RAM belleklerin kapasiteleri, ROM belleklerde olduğu gibi, 1K, 4K, 8K, 16K, 64K,... 1M, 4M, 16M, 32M, 64M vb. şekilde tanımlanır ve her bir kelime 4, 8, 16, 32 veya 64 biti içerebilir. RAM bellekleri bir arada kullanarak, kapasitesini veya kelime uzunluğunu değiştirmek mümkündür. Bu işlemleri açıklamadan önce, RAM belleklerde bilgi saklanması işlemini ve RAM belleğin iç yapısını inceleyelim.

13.6.2.3. RAM Belleklerde Bilgi Saklanması ve RAM Belleklerin İç Yapısı

Yarıiletken yapıdaki RAM belleklerde temel bilgi saklama elemanı, Flip-Flop (FF)’dur. Bir bitlik bilginin saklanması için, transistörlerden oluşan '1' adet FF kullanılır. Temel elemanı FF olan '1' bitlik bilgi saklama devresi, **'bellek hücresi - BH'** olarak isimlendirilir. Bellek hücresinde; giriş, çıkış, oku / yaz, seçme girişlerinin bulunması gerekir (Şekil 13.34.a).

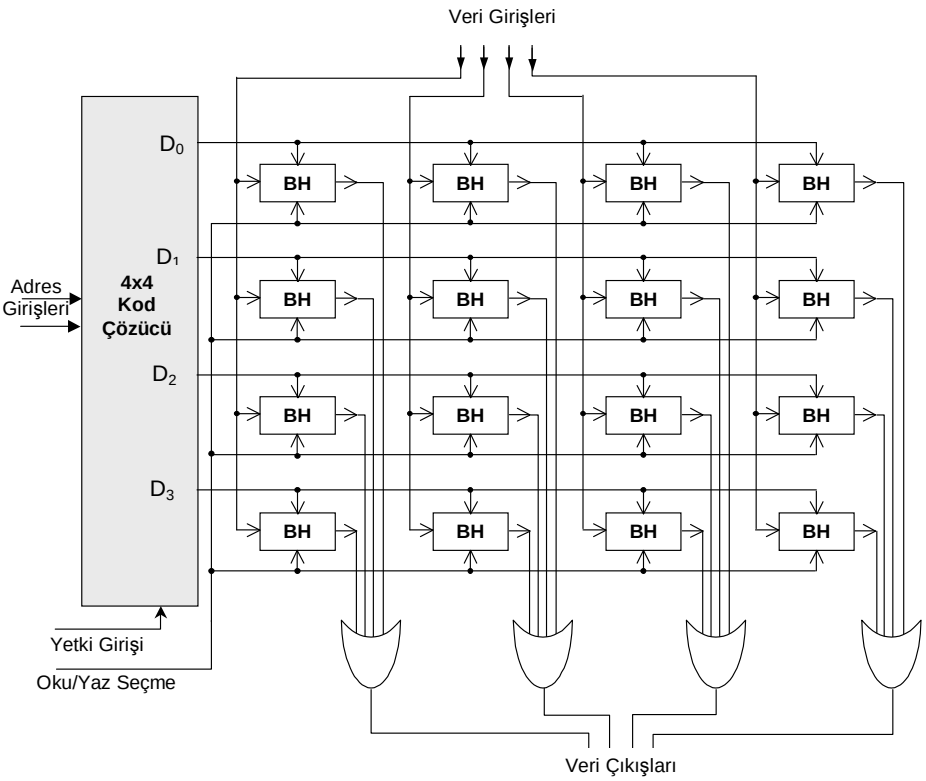


a) Bellek hücresi blok şeması. **b)** Bellek hücresi iç yapısı.

Şekil 13.34. RAM bellek hücresi blok şeması ve lojik devresi.

Bellek hücresinde bilgi yazma veya bulunan bilgiyi okuma işlemlerinden hangisinin yapılacağına, oku / yaz girişinin değeri ile karar verilir (Şekil 13.34.b). Oku / yaz girişine ‘1’ uygulanması durumunda; K3 nolu ‘VE’ kapısı, FF Q çıkışındaki bilgiyi çıkış ucunda gösterecek girişlere sahip olur. Oku / yaz girişinin ‘0’ olması durumunda ise, K1 veya K2 kapısı girişteki bilgiyi FF’e yükler. Bilgi yüklenmesi, hücreye girişteki bilginin yazılması demektir.

Yetki veya seçme girişi olarak isimlendirilen giriş ile, ilgili hücrenin seçilmesi (yetkilendirilmesi) sağlanır. Şekil 13.34.b’deki devrede, ‘seçme’ girişinin '1' yapılması ile bellek hücresi seçilir. Yetki girişi sayısı birden fazla olabilir. Ayrıca, kullanılan FF’in tetikleme sinyali girişi bulunması durumunda, bellek hücresine tetikleme sinyali girişi eklenir.



Şekil 13.35. 4x4 RAM bellek organizasyonu.

Bellek hücreleri bir arada kullanılarak, kelime saklama kapasitesine sahip hücre grupları oluşturulur. Bellek hücrelerinden oluşan grupların bir arada kullanılması ile geniş kapasiteli bellek oluşur.

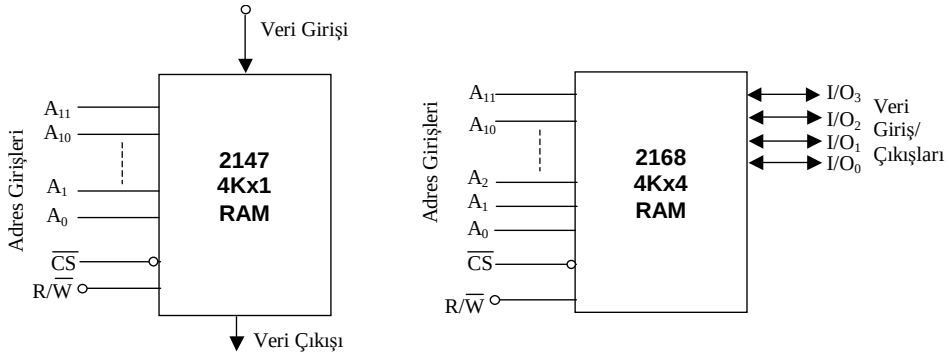
Şekil 13.35'de, herbir adresin dört bellek hücresi içerdiği (kelime uzunluğu 4 olan) RAM belleğin iç yapısı görülmektedir. Şekli basitleştirmek amacıyla bellek hücreleri blok olarak gösterilmektedir.

Yetkilendirme girişi ile aktif hale getirilen 4x4 kod çözücü devresi, adres girişlerinden gelen bilgiye göre kod çözme işlemi gerçekleştirir. Adres giriş değerlerine göre kod çözücü çıkışlarından birisi '1' değerini alır ve ilgili bellek hücrelerini yetkilendirir.

Yetkilendirilen bellek hücreleri, yazma veya okuma işlemi için hazırdır. Seçilen bellek hücreleri, oku / yaz girişindeki değere göre uygun olan işlemi gerçekleştirir.

Okuma işlemi seçilirse, hücrelerdeki bilgiler devredeki 'VEYA' kapıları üzerinden veri çıkışlarında gözükür. Okuma işlemi, bellek hücrelerindeki FF'lerde bulunan değerlerin alınmasıdır.

Yazma işleminin seçilmesi durumunda, veri girişlerinden uygulanan bilgiler (1 veya 0) ilgili bellek hücresine yüklenir. Yükleme işlemi, bellek hücresinde bulunan FF'nin '1' veya '0' değerine kurulmasıdır.



Şekil 13.36. 2147 ve 2168 RAM bellek entegrelerinin sembolleri.

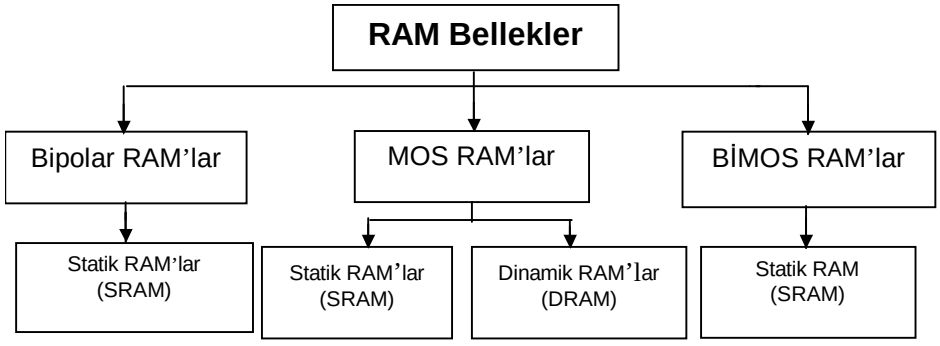
RAM bellek entegrelerinde veri giriş ve çıkış pinleri ayrı - ayrı bulunabileceği gibi, ortak kullanılacak şekilde de olabilir. Ortak pinlerin veri giriş ve çıkışı için kullanılması durumunda, entegrede gerekli pin sayısı azalır ve bu durum özellikle geniş kelime uzunluğuna sahip belleklerde kolaylık sağlar.

2147 RAM bellek entegresi; 4Kx1 kapasitesinde, veri giriş ve çıkışları için farklı pinlerin kullanıldığı bellek elemanıdır (Şekil 13.36.a). Seçme girişi ile (aktif '0') entegre yetkilendirilerek, adres girişlerindeki kombinasyonun temsil ettiği bellek hücresi seçilir.

2168 RAM bellek entegresi ise, 4Kx4 kapasitesinde, ortak veri giriş / çıkış pinlerine sahip bellek elemanıdır (Şekil 13.36.b). R/W pinindeki bilgiye göre, ortak pinler belleğe bilgi yazmak veya bellekteki bilgileri okumak için kullanılır.

13.6.2.4. RAM Çeşitleri

Yarı iletken RAM'lar bipolar transistör veya MOS teknolojileri kullanılarak imal edilirler. Ayrıca bu iki teknolojinin üstün taraflarının kullanılarak imal edildiği için 'BİMOS' olarak isimlendirilen üçüncü bir RAM yapım teknolojisi bulunmaktadır. Bu durumda, RAM bellekleri kullanılan malzemeye göre üç gruba ayırmak mümkündür: Bipolar RAM'lar, MOSRAM'lar, BİMOS RAM'lar (Şekil 13.37).



Şekil 13.37. Yarı iletken RAM çeşitleri.

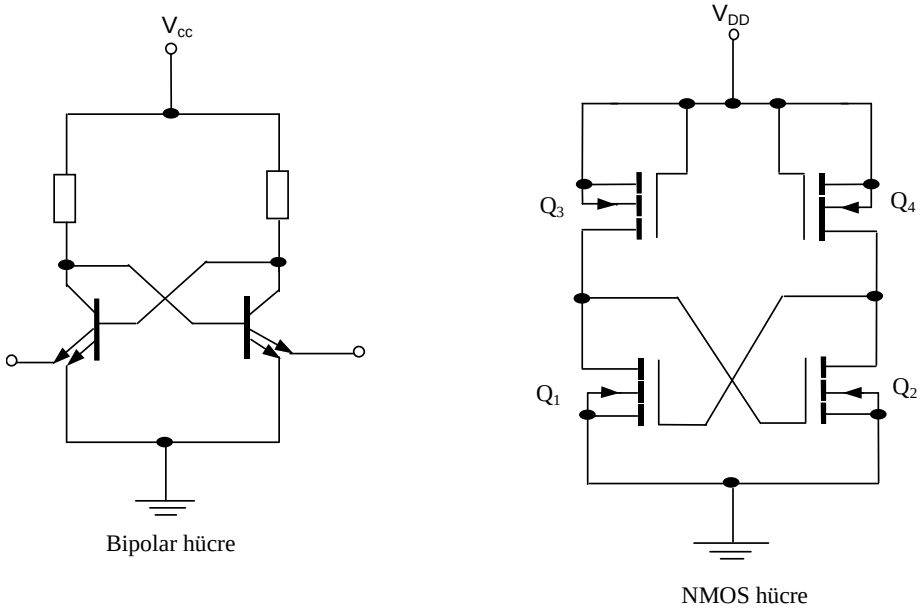
Yarı iletken RAM bellekler, bilgi saklama için enerji gereksinimi duyma durumlarına ve bilgi saklayabilme sürelerine göre iki gruba ayrılabilir: Statik RAM'lar ve

dinamik RAM'lar. Her ikiside 'uçucu-volatile' yapıda olan bu belleklerde enerji kesildiği zaman bilgi kaybolur. Farklı bilgi saklama tekniğine sahip RAM belleklerin iç yapısını ve bilgi saklama şekillerinin inceleyelim.

13.6.2.5. Statik RAM'lar (SRAM)

Bilgi saklama elemanı olarak FF'ler kullanan ve enerji verildiği sürece bilgiyi sonsuza kadar saklayabilen RAM bellekler, **'statik RAM bellekler'** olarak isimlendirilir. Statik bellekteki bilgi, yeni bilgi kaydedilmesi veya enerjinin kesilmesi durumunda kaybolduğundan, uçucu (volatile) bellekler olarak düşünülürler.

Yapım malzemesi olarak bipolar transistör kullanan bipolar RAM'lar ve bipolar transistör - MOS eleman birleşimi kullanan BİMOS RAM'lar, statik RAM sınıfına girerler. Bununla beraber SRAM'ların büyük bir kısmında NMOS ve CMOS elemanlar kullanılır. Statik RAM'larda bipolar transistör kullanımı hızı artırırken, MOS teknoloji kullanımı ile kapasite artar ve güç tüketimi azalır.

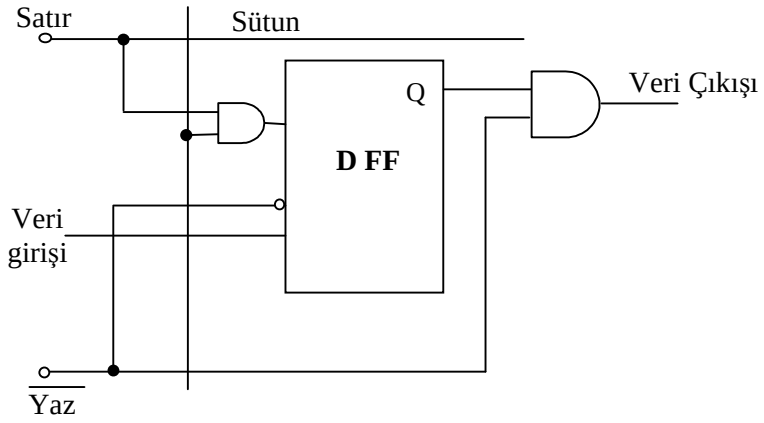


Şekil 13.38. Bipolar ve NMOS statik RAM hücrelerinin yapısı.

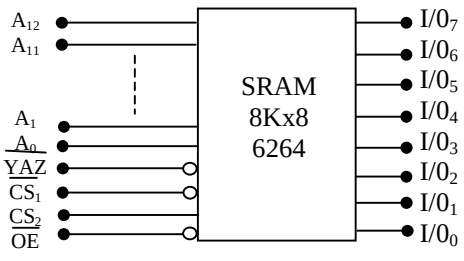
Şekil 13.38'de bipolar transistör ve NMOS elemanlar kullanılarak oluşturulan SRAM bellek hücreleri görülmektedir. Bipolar transistörle yapılan bellek hücresi iki transistör ve iki direnç kullanılırken, NMOS bellek hücresinde dört adet NMOS eleman kullanılır. Bipolar teknoloji, daha kompleks bir yapıya sahip olduğundan ve

direnç kullanımı gerektirdiğinden bellek içerisinde daha geniş yer kaplarlar. MOS teknolojide ise, dirençler yerine MOS elemanlar kullanıldığından daha az yer kaplarlar. MOS teknolojisi yerine CMOS kullanılması durumunda tek fark; direnç olarak kullanılan Q3-Q4 elemanları yerine, PMOS elemanları kullanılmasıdır.

Farklı malzeme kullanılarak imal edilebilen SRAM bellek hücresi, Şekil 13.39'da gösterilen giriş / çıkış uçlarına sahiptir. Bellek hücresinin sütun ve satırının '1' yapılması ile ilgili bellek hücresi seçilir. Yaz ucunun (0) olması durumunda, giriş bilgisi hücreye '1' veya '0' olarak kaydedilir. Yaz girişinin '1' olması durumunda ise hücre giriş bilgisinden etkilenmez, ancak hücrede saklanan bilgi veri çıkış hattında gözükür.



Şekil 13.39. SRAM hücresi genel lojik şeması.



(a)

Çalışma Modu	Girişler				
	Yaz	$\overline{CS_1}$	$\overline{CS_2}$	$\overline{OE}$	I/O Uçları
Oku	1	0	1	0	Veri çıkış
Yaz	0	1	1	X	Veri giriş
Çıkış Yetkisiz	1	X	X	1	Yüksek Z
Seçilmeyen	X	1	X	X	Yüksek Z
Enerji Yok	X	X	0	X	

(b)

Şekil 13.40. CMOS 6264 SRAM bellek entegresinin sembolü ve çalışma tablosu.

SRAM ve ROM bellek entegreleri arasındaki temel fark; SRAM entegrelerin veri girişlerine ve oku / yaz kontrol girişlerine sahip olmasıdır. CMOS 6264 entegresi; 8Kx8 bilgi saklama kapasitesinde, 100 nsn okuma / yazma süresine ve 'standby' durumunda 0,1 mW enerji tüketimine sahip SRAM bellek entegresidir (Şekil 13.40.a). Entegrede, 2^{13}=8192=8K bellek kapasitesi olması nedeni ile 13 adet adres girişi ve 8 adet bilgi giriş / çıkışı bulunur. Entegrede bulunan kontrol girişlerinin (4 adet) durumları elemanın çalışma durumunu tespit etmek için kullanılır (Şekil 13.40.b).

CS (Entegre seçme-chip select) uçlarının aktif olması ile entegre seçilir. Entegrenin seçilmesi için, her iki girişin aktif olması zorunludur (CS_1=0, CS_2=1).

Entegredeki YAZ ($\overline{WE}$) girişinin '0' olması ile, RAM içerisine yazma işlemi seçilir. YAZ girişinde '1' olması durumunda ise, 'okuma işlemi seçilir ve OE (çıkış yetki-output enable) ucunun '0' olması ile bilgi çıkışta gözükür. Entegre seçilmediği durumlarda, entegre düşük güç harcama durumundadır ve diğer uçların hiçbir etkisi yoktur.

13.6.2.6. Dinamik RAM Bellekler (DRAM)

Bilgi saklama elemanı olarak FF yerine küçük değerli MOS kondansatörün (birkaç pikofarad) kullanıldığı bellek, **'Dinamik RAM'** olarak isimlendirilir (Şekil 13.41.a).

MOS teknoloji kullanılarak imal edilen dinamik RAM'ların üstünlükleri;

i- Basit yapıda ve büyük bilgi saklama kapasitesinde olmaları,

ii- Düşük güç tüketimi,

iii- Hızlı işleme hızı,

iv- Bir entegre içerisine çok sayıda bellek hücresi yerleştirilebilmesi nedeni ile düşük maliyet,

olarak sıralanabilir.

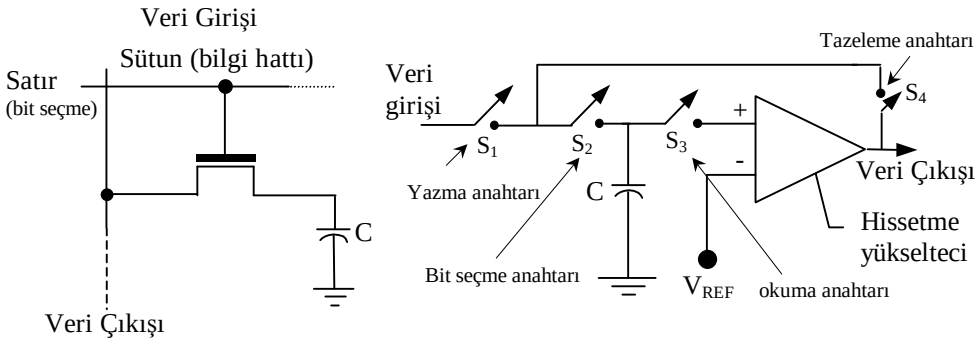
Dinamik RAM'ların sakıncası ise; bilginin kondansatörde saklanması nedeni ile, saklanan bilginin belirli bir süre sonucunda kaybolmasını engellemek için gerekli devrelerdir.

DRAM'lerde bilgi kaybolmasını engellemek için, bilgi saklamak amacıyla kullanılan hücrelerin belirli (periyodik) zamanlarda yenilenmesi gerekir. Bilgilerin yenilenmesi işlemi, **'tazeleme'** (refreshing) olarak isimlendirilir. Hücrelerin tazeleme zamanı, 2-10 msn olarak değişir. Tazeleme işlemi, yeni devreler gerektirir ve devreler DRAM işlemini karmaşık hale getirir.

Şekil 13.41.b'de basitleştirilmiş tazeleme devreli DRAM hücresi yapısı görülmektedir. Şekilde bulunan S_1-S_4 anahtarları normalde MOSFET elemanlardır. Anahtar

ile temsil edilen MOSFET'ler farklı adres kod çözücü çıkışları ve R/W sinyali tarafından kontrol edilir. Devredeki kondansatör, temel bilgi saklama elemanıdır. Yapı olarak tek bir MOS transistör ve bir kondansatörden oluşan DRAM hücresinde transistör bir anahtar gibi çalışır (Şekil 13.41.b). DRAM bu yapısı ile, yapı olarak RAM'lara göre çok daha basittir. Küçük alana büyük miktarlarda bilgi saklama kapasitesi ve düşük güç tüketimi gibi özellikleri nedeni ile DRAM'lar sahsi bilgisayarda dahili bellek olarak kullanılırlar.

DRAM bellek hücreleri, dizinler şeklinde sıralanarak geniş ölçekli bellekler oluşturulur. Örneğin; 128x128 dizinlerin oluşturulması ile 16384=16K bellek hücresi ortaya çıkar. Şekil 13.42'de 16Kx1 DRAM entegresinin iç yapısı görülmektedir. 16K bellek bölgesinin seçimi için 14 adet adres giriş hattına gereksinim vardır (2^{14}=16384). Adres hatlarından A_0-A_6 arası olanlar satırları seçmek için kullanılırken, A_7-A_{13} arasındaki hatlar sütunları seçmek için kullanılır.



(a) **(b)**

Şekil 13.41. MOS tekniği ile oluşturulan DRAM hücresi ve tazeleme devreli DRAM.

Piyasada 1024Kx1 (1Mx1) ve 256Kx4 kapasiteye sahip DRAM'lar bulunmaktadır. 1-bit ve 4-bit uzunluğa sahip DRAM'lar aynı hücre düzenine sahiptirler. Tek fark, her bir pozisyonun 4 hücre içermesi ve her bir adres seçme kombinasyonunda dört hücre grubunun beraber seçilmesidir.

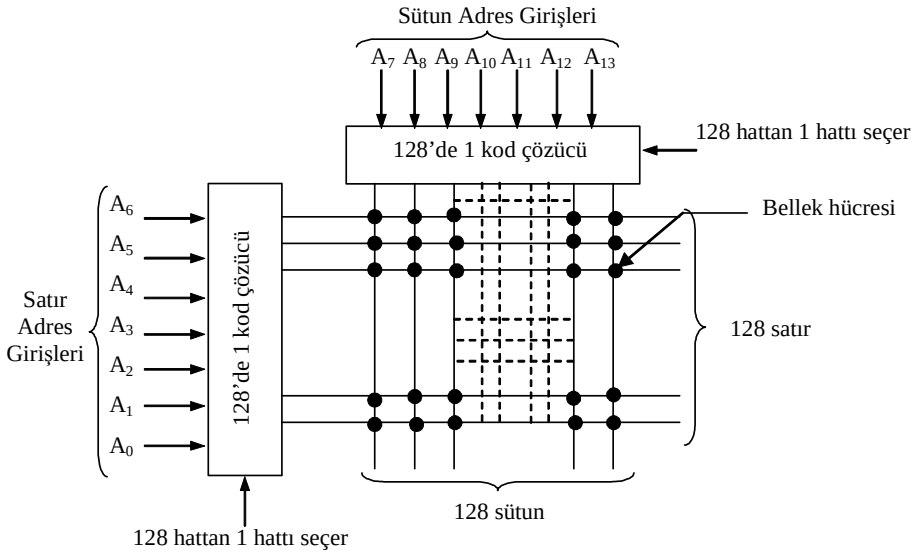
DRAM Belleklerde bellek tazeleme işlemi:

DRAM belleklerde bilgi okuma / yazma ve bellek tazeleme işlemlerini Şekil 13.41.b'yi kullanarak özetleyelim.

Bellek hücresine bilgi yazma işleminde; adres kod çözücüden ve oku / yaz devresinden gelen

sinyaller, S_1 ve S_2 anahtarlarını kapatırken S_3 ve S_4 anahtarlarını açarlar. Bu durum, veri girişine ‘C’ kondansatörünü bağlar. ‘C’ kondansatörüne uygulanan ‘1’ bilgisi kondansatörü şarj ederken, ‘0’ lojik bilgisi kondansatörü deşarj eder. Açık olan S_3 anahtarı nedeni ile, ‘C’ kondansatörü devrenin çıkış kısmından ayrılır.

İdeal durumda ‘C’ kondansatörünün sonsuz olarak şarjlı kalması gerekir. Ancak anahtar yerine kullanılan MOS elemanların iç dirençleri üzerinde oluşan sızıntı akımı, belirli bir süre sonra kondansatördeki bilginin kaybolmasına sebep olur. Hücreden bilgi okuma işlemi sırasında S_2, S_3 ve S_4 anahtarları kapalı iken, S_1 anahtarı açık durumdadır. Bu bağlantılar, saklanan bilginin hissetme yükseltecine bağlanmasını sağlar. Hissetme yükselteci, referans olarak uygulanan gerilim ile kondansatörden gelen gerilimi karşılaştırır ve veri çıkışı için 0 veya 5 volt gerilim üretir. Üretilen çıkış gerilimi S_2 ve S_4 anahtarları üzerinden ‘C’ kondansatörüne uygulanarak kondansatördeki verinin tazelenmesini (şarj veya deşarj ederek) sağlar. Diğer bir deyişle; bir bellek hücresinde bulunan veriyi her okuma işleminde bellek hücresi tazelenir.



Şekil 13.42. 16Kx1 DRAM belleğin hücre düzenlemesi.

Ortalama 2 msn’lik periyotlarla tazeleme işlemi yapılan 16Kx1 DRAM bellekte, her 122 μsn’lik (2 msn / 16.384 = 122 μsn) sürede ‘okuma’ işleminin yapılması gerekir. Bu süre, hızlı DRAM’lerde bile çok kısa zaman aralığıdır ve normal okuma işlemleri sırasında

mümkün değildir. Bu nedenle, DRAM bellek entegrelerinde oluşan her bir ‘okuma’ işleminde okunan bellek hücresiyle aynı sırada bulunan tüm hücreler tazelenir. Bu özellik, tüm belleği tazelemek için gerekli okuma işlemi sayısını azaltır. Sonuçta, 2 msn’lik zamanlama dilimi içerisinde yalnızca 128 satırın okunması işlemini gerçekleştirir.

Bir bellekte bulunan tüm satırların okunması işlemi, uygun şekilde tasarlanan bir sayıcı yardımıyla gerçekleştirilir. ‘Tazeleme sayıcısı’ olarak isimlendirilen yukarı sayıcı ile, ilk bellek bölgesinin bulunduğu satırdan başlanarak tüm belleklere ait satırların temsil edildiği satır numaraları taranır. Örneğin; 128 satırın taranması için 7 bit’lik tazeleme sayıcısı gerekir ve sayıcı ‘0000000’ değerinden başlayarak saymaya başlar (0 nolu satır tazelenir). Sayıcı, ‘1111111’ değerine kadar sayarak, 127 nolu satıra kadarki bellek hücrelerini tazeler. Sayıcı kullanılarak yapılan tazeleme işlemi ortalama 50 µsn içerisinde tamamlanabilir.

13.6.2.7. RAM Bellek Uygulamaları

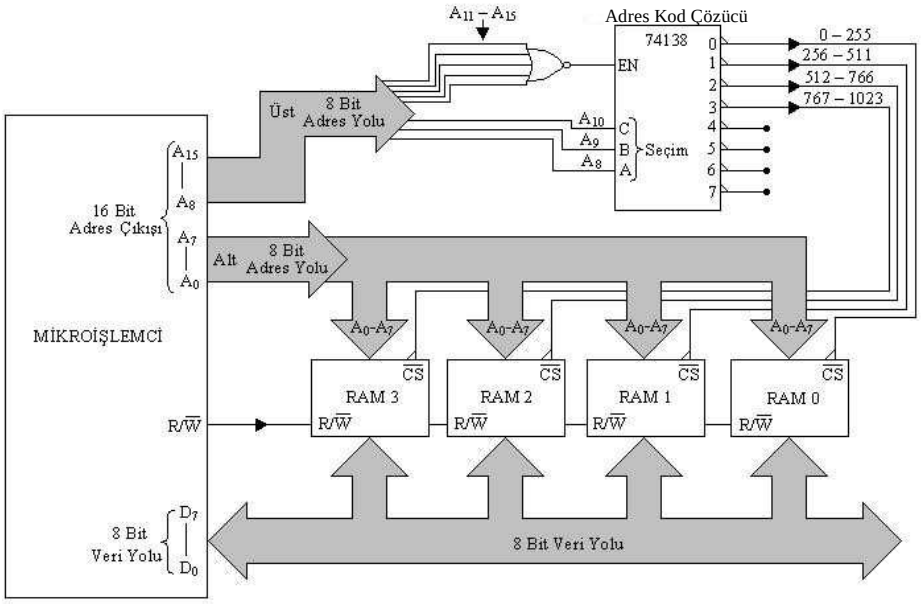
RAM belleklerin uygulama alanları açısından, dinamik RAM (DRAM) bellekler statik RAM (SRAM) belleklere göre daha geniş bir kullanım alanına sahiptir. Bunun nedeni, DRAM’ların daha küçük ölçülere, daha ucuz fiata ve daha az güç tüketimine sahip olmalarındandır. Bununla beraber, statik RAM’lar daha küçük erişim zamanına sahiptir ve ‘tazeleme’ devresine gereksinim yoktur.

SRAM’lar küçük ölçüde (<64K) oku/yaz bellek gereksinimi olan yerlerde veya hızlı bellek erişiminin önemli olduğu durumlarda kullanılırlar. DRAM’lar ise geniş ölçekli oku/yaz bellek gereken yerlerde tercih edilirler.

RAM belleklerin en yaygın kullanım yeri mikroişlemcili sistemlerdir. RAM bellekler, şahsi bilgisayarlardan dijital osilaskoplara, dijital kontrol sistemlerinden klavye ve görüntüleme elemanlarına kadar çok farklı yerlerde kullanılmaktadır.

Şekil 13.43’de, mikroişlemcili sistemlerde kullanılan RAM’ların yapısı görülmektedir. Mikroişlemcili sistemde bulunan adres yolu ve veri yolunu kontrol eden ve kontrol sinyalleri üreten mikroişlemci yardımıyla bir bellek bölgesindeki bilgi okunabilir veya bir bellek bölgesine bilgi yazılabilir.

Mikroişlemcili sistemde bulunan veri yolu kullanılarak veriler sistem içerisinde iletilir. Tüm birimlerin aynı veri yolunu kullanması nedeni ile, veri yolu bir anda yalnızca bir birim tarafından veri yazmak veya veri okumak için kullanılabilir. İletilecek verinin veri yolu üzerine yerleştirilmesinden önce, erişilmesi istenen bellek bölgesini temsil eden adres değeri adres yoluna yerleştirilir.



Şekil 13.43. RAM belleklerin mikroişlemcili sistemlerde kullanılması.

Şekil 13.43'deki devrede, her biri 256x8 bit bilgi saklama kapasitesine sahip dört adet RAM entegresi bulunmaktadır. RAM entegrelerinden hangisinin yetkilendirileceğine, mikroişlemci tarafından üretilen adres değeri ile karar verilir. Yetkilendirilecek RAM entegresine yüksek değerli adres yolunda bulunan değerler ile karar verilirken, yetkilendirilen RAM entegresinde erişilecek bölge düşük değerli adres yolunda bulunan değerler ile belirlenir. Adres kod çözücü olarak kullanılan 74138 entegresi çıkışlarındaki değerlere göre; 0 – 255 arasındaki bellek bölgelerini içeren RAM 0, 256 – 511 arasındaki bellek bölgelerinin içeren RAM 1, 512 – 766 arasındaki bellek bölgelerini içeren RAM 2 veya 767 – 1023 arasındaki bellek bölgelerini içeren RAM 3 entegrelerinden birisi seçilir. Seçme işlemi için; adres yolunda bulunan A_8, A_9 ve A_{10} hatları kod çözücü 74138 entegresinin seçme girişlerine uygulanırken, $A_{11} – A_{15}$ arasındaki hatlar kod çözücünün 'yetkilendirme' girişine uygulanır. $A_{11} – A_{15}$ hatlarının tümünün '0' olması ile kod çözücü entegresi yetkilendirilir ve A_8-A_9-A_{10} hatlarının değerlerine göre kod çözücünün çıkışında oluşan değerler uygun olan RAM entegresinin seçilmesini sağlar.

13.6.2.8. 'FLASH' Bellekler

Geniş bellek hücresi kapasitesine sahip oku / yaz bellek olan 'Flash bellekler', uçucu olmayan (nonvolatile) belleklerdir. 'Flash' belleklerde yüksek yoğunluğun (büyük bilgi depolama kapasitesi) nedeni ; bellek hücresi olarak MOS transistörün kullanılmasıdır.

Hücrede depolanacak bilginin ‘0’ veya ‘1’ olması, MOS’un kapısı (gate) üzerinde şarj oluşması / oluşmaması sonucunu doğurur.

İdeal bir bellekte bulunması gerekli; çok bilgi saklama, küçük erişim zamanı, uçucu olmama, okuma / yazma imkanı, yüksek işlem hızı, bit başına düşük maliyet gibi özelliklere sahip ‘Flash’ bellekler, diğer bellek tipleri ile karşılaştırılınca üstünlüklere sahiptir. Çünkü, diğer bellek türleri belirtilen özelliklerin bir kısmına sahip iken, Flash bellekler istenen tüm özelliklere sahiptirler.

Flash bellekler kolayca yeniden programlanabilirler. Çünkü oku / yaz özelliğine sahiptirler. Flash bellekler, ROM bellek türleri gibi uçucu olmayan belleklerdir ve enerji kesilince dahi saklanan bilgiyi korurlar.

ROM’lara göre kolayca yeniden programlanabilme üstünlüğüne sahip Flash bellekler, SRAM’lara göre yüksek bilgi kapasitesine sahip olma ve uçucu olmama üstünlüklerine sahiptirler. DRAM bellekler ile karşılaştırılan ‘Flash’ belleklerin, tazelenme işlemine gerek duymama ve düşük güç tüketimi gibi üstünlüklere sahip olduğu görülür.

Flash bellekler her bir byte için 10 μsn yazma zamanına sahiptir. Bu değerler hızlı EEPROM’larda 5 msn ve EPROM’larda 100 μsn’dir.

13.6.2.9. Bilgisayarlarda Kullanılan Bellek Çeşitleri

Bilgisayarlarda ve özellikle PC’lerde kullanılan bellekler, ROM ve RAM belleklerin farklı yapım şekilllerinde ihtiyaçlara göre geliştirilmesi ile oluşturuluyor. Belirli bir süreç içerisinde bilgisayarlarda kullanılan belleklerin genel özellikleri aşağıdaki şekild özetleyebiliriz.

SRAM (Statik Random Access Memory): Devreye akım sağlandığı sürece bilgilerin kaybolmadığı, 20nsn’lik hıza sahip, pahalı fakat hızlı, önbelleklerde tercih edilen bellek türü, önbelleklerde tercih edilen bellek türü. Bir bitliğk bilginin sakllanması için iki adet transistör kullanılır ve ikinci transistör ilk transistörün çıkışını kontrol eder.

DRAM (Dynamic RAM): Bir bitlik bilgiyi saklamak için tek bir transistörün kullanıldığı, SRAM’e göre daha basit bir yapıda ve dahya ucuz, saklanan bilgilerin sürekli tazelenmesine ihtiyaç duyulan bellek tütü. Ucuz ve SRAM’e göre daha yavaş olması nedeni ile bellek entegrelerinde kullanılır.

EDO-RAM (Extended Data output RAM): DRAM’lerin geliştirilmesi ile oluşturulan, 50 ile 70 nsn’lik hızlarda ve 66 MHz’ çalışan bellek türü. Bellek yuvalarına ikili gruplar halinde takılması gerekiyor.

VRAM(Video RAM):Görüntü bilgilerini hızlı bir şekilde işlemek ve aktarmak için kullanılan bellek türü. İki ayrı giriş/çıkış birimine sahip ve bunlardan birisi görüntü bilgisini oluşturmak için kullanılırken, ikincisi bilgileir monitöre göndermek için kullanılır. İkinci

port band genişliğini arttırmasına rağmen iki katına çıkarmaz ve sadece yüksek çözünürlükte veya rengin söz konusu olması durumunda etki eder.

WRAM(Windows RAM): VRAM belleklerin geliştirilmesi ile elde edilir ve eklenen yeni devreler yardımı ile özel görüntü fonksiyonlarının daha hızlı gerçekleştirilmesi sağlanır. Özellikle animasyonlarda etkin olan WRAM bellekler, ekran kartlarında kullanılıyorlar.

SGRAM (Synchronous Graphics RAM): Verileri tek tek yerine bloklar halinda alarak işleyen ve bu özelliği nedeni ile okuma/yazma performansını önemli ölçüde arttıran bellek türü. Özellikle grafik kartlarında kullanılıyor ve performansı arttırıyor.

SDRAM(Synchronous DRAM): Mikoişlemci sistem frekansı ile belleğin belleğin çalışma frekansının birbiriyle senkronizeli çalışması için geliştirilen bellek türü. Belleğe giden komutlarda hızlanma ve sistem performansında artış sağlayan bellek türü. Piyasada PC 66 , PC 100, PC 133 olarak üç farklı çeşidi bulunuyor. SDRAM'lere eklenen 'bellek kontrolü' devresi ile performansın arttığı belleklerde bilgiye erişim süresi ortalama 10nsn.
DDR-RAM (Double Data Rate RAM): SD-RAM'lerin geliştirilmiş şekli olan DDR-RAM'ler, SDRAM II veya DDR-SDRAM olarakta isimlendiriliyor. Çalışma frekansları yüksek olan bu belleklerde 100 MHz'de çalışan PC 200 ve
133 MHz'de çalışan PC 266 olmak üzere iki çeşit bulunuyor. SDRAM'lerde 3,3V olan çalışma geriliminin 2,5V'a indirilmesi ile mobil bilgisayarlarda tercih ediliyor.

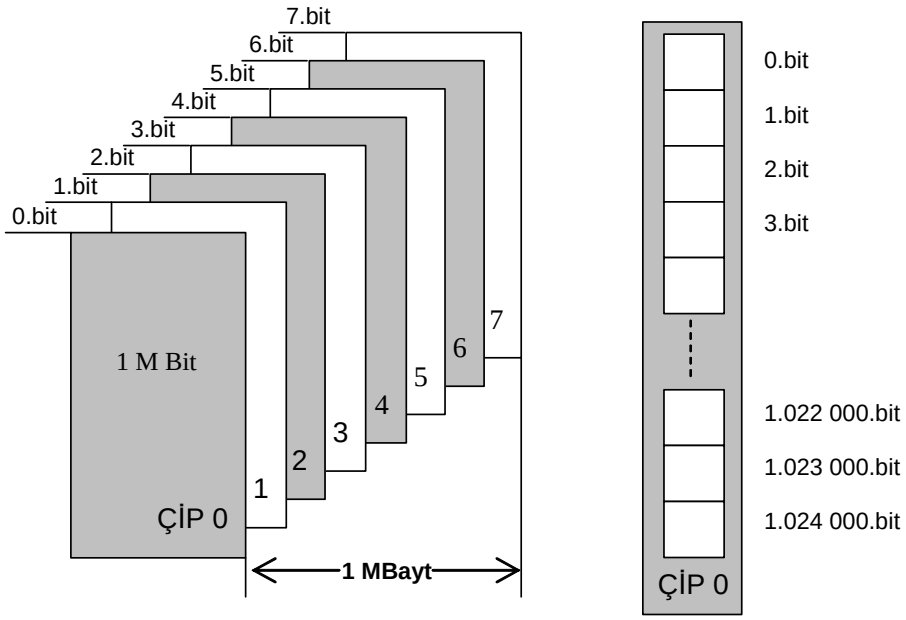
RAMBUS: İş istasyonlarında yürütülen 3D grafik, video ve animasyonlarda tercih edilen bellek türü. RDRAM, Concurent RDRAM ve Direct RDRAM olmak üzere üç farklı yapıda üretiliyorlar. En basit yapıya sahip RDRAM'ler, DDR-SDRAM'lerdeki gibi düşen ve yükselen kenarları veri iletişimi için kullanılıyor. Concurent RAMBUS, RDRAM'in geliştirilmiş bir türü, veri transferinin senkron olarak gerçekleştirildiği Concurent RAMBUS'larda 60 MHz'lik çalışma frekansında saniyede 600MBayt'lık veri transferi transferi gerçekleştirebiliyor. Direct RAMBUS'lar ise en gelişmiş çeşit ve 800 Mhz'lik çalışma frekansında saniyede 1,6 Gbayt'lık veri iletebiliyor.

13.7. SIMM Bellek Kartları

1980'li yıllarda çift sıralı paket (Dual intine Package-DIP) yapısında 256 KBit kapasiteli bellek entegrelerinin bir arada kullanılması ile oluşturulan 256 KBayt RAM bellekler yeterli iken, günümüzde 128 MByte normal RAM bellek kapasitesi olarak görülmektedir. Bilgisayarlarda ihtiyaç duyulan RAM bellek miktarının her geçen gün artması, üreticileri yeni tasarımlara zorlamıştır. Bellek entegrelerinin bir arada kullanılması ile oluşturulan büyük kapasiteli bellekler çözüm olsa da, çok sayıda bellek entegresinin mikroişlemci ve giriş/çıkış birimlerinin bulunduğu ana kartlar üzerine yerleştirilmesi, bilgisayar ana kartının boyutunu büyütmektedir. Bu problemin üstesinden gelmek ve dar alanda büyük bellek kapasitesi oluşturmak amacıyla, bellek entegrelerini bir sıra halinde bir küçük kart üzerinde dizildiği bellek modülleri geliştirildi. Geliştirilen bellek modülü, tek sıralı bellek modülü

(Single In-line Memory Module-SIMM) olarak adlandırılır. Bu tip bellek kartlarında giriş-çıkış uçları baskılı devre kartı üzerindedir.

SIMM modüllerinde başlangıçta düşük kapasiteli bellek entegreleri kullanılırken, daha sonraları yüksek kapasiteli bellek entegreleri kullanıldı. 8 adet 1MBit kapasiteli bellek entegresinin bir arada kullanılması ile 1Mbit x 8= 8Mbit'lik bellek elde edilir. Şekil 13.44'de 8 MBit'lik belleğin SIMM bellek kartı üzerinde oluşumu görülmektedir.



Şekil 13.44. 1 MBit'lik bellek entegresi iç yapısı ve SIMM bellek kartının oluşumu.

1 MBit'lik bellek entegresinde 1.024.000 bit bilgi sıra halinde kaydedilebilir. Bilgiler bayt şeklinde saklandığından, 8 adet 1 MBit bellek entegresinin bir arada kullanılması ile 1 MBayt bellek oluşturulur (Şekil 13.44). SIMM kartlarında, saklanması gereken bilgiden 1 bit fazla bilgi bulundurulur. Fazladan eklenen bu bit **'eşlik biti'** olarak adlandırılır.
Örneğin; 1 MBit'lik entegreler ile 1 MBayt'lık bellek oluşturmak için, 8 adet yerine 9 adet entegre kullanılır. 8 adet entegre saklanacak 8 biti tutmak için kullanılırken, 9 bitin saklandığı 9. Entegre eşlik biti için kullanılır. 9. Entegrede saklanan bit, sekiz entegrede saklanan bilginin kaç adet lojik '1' içerdiğini belirtir. Sistemde çift (even) eşlik biti kullanılıyorsa, bellekte saklanan lojik 1'lerin sayısı çift olacak şekilde eşitlik biti eklenir. Bellekteki 8 bitte bulunan lojik 1'lerin sayısı tek ise, eşlik biti olarak '1' eklenirken, lojik 1'lerin sayısı çift ise eşlik biti olarak '0' eklenir.

Açıklandığı şekilde eşlik biti kullanan bellekler **'pariteli bellek'** olarak tanımlanır. Paritesiz

bellek gereken yerlerde pariteli bellek kullanılması durumunda, bilgisayar eşlik bitini temsil eden entegreyi görmez.

Elektronik teknolojisinde yapılan gelişmeler sonucunda, SIMM bellek modüllerinin yerine iki kat kapasiteye sahip **'Çift Sıralı Bellek Modülleri'** (Dual In-line Memory Module - DIMM) geliştirilmektedir. DIMM bellek kartlarında kullanılan DRAM entegrelerin dört tarafında da uç bulunmakta ve belleği meydana getiren katmanlar ve dolayısıyla toplam bellek kapasitesi artırılmaktadır.

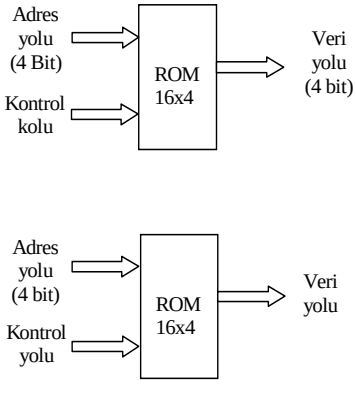
13.8. Bellek Ölçülerinin Genişletilmesi

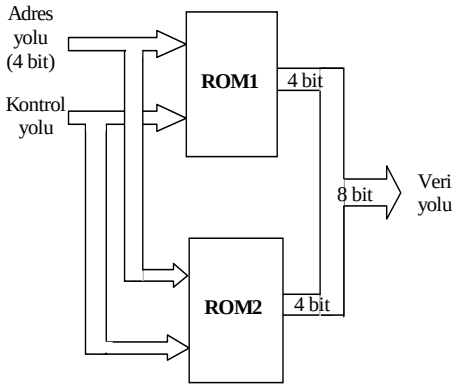
Bellekler, her bir adreste saklanan kelime uzunluğu (bit sayısı) veya saklanabilecek kelime miktarı açısından genişletilebilirler. Her iki genişletme işlemini birlikte yapmakta mümkündür.

13.8.1 Bellek Kelime Uzunluğunun Genişletilmesi

Bir bellekte saklanabilecek kelimenin uzunluğunu arttırmak için, veri yolundaki bitlerin (hatların) sayısını arttırmak gereklidir. Örneğin; elimizde bulunan 4 bit kelime uzunluğuna sahip bellek entegreleri ile (Şekil 13.45.a), 8 bitlik kelime uzunluğuna sahip bellekler üretmek mümkündür (Şekil 13.45.b).

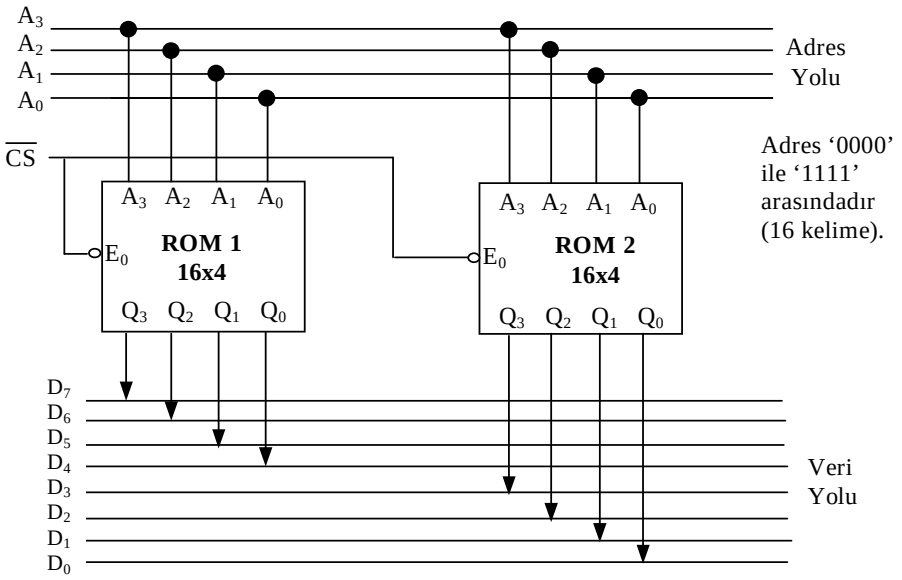
Kelime uzunluğunu artırmak amacıyla iki entegre birlikte kullanılmasına rağmen, gerekli olan adres hattı sayısında bir değişiklik yoktur. Diğer bir değişle, bellek yapısı aynı bellek adresi sayısına sahiptir (Şekil 13.46). Adres yolu üzerine bir adresin yerleştirilmesi ile, iki entegrede bulunan aynı numaraları adresler seçilir ve her iki adreste bulunan değerler, veri yoluna yerleştirilir. Veri yolunda 8 hattın bulunması ve entegrelerdeki 8 bitlik bilginin aynı anda veri yoluna yerleştirilmesi nedeni ile kelime uzunluğu 8 bite yükseltilmiş olur. Herbir adreste saklanan bilgi ise 8 bite yükselmiştir.





a) İki ayrı 16x4 ROM bellek entegresi. **b)** İki adet 16x4 ROM ile 16x8 ROM oluşumu.

Şekil 13.45. İki adet 16x4 ROM kullanarak 16x8 bellek elde edilmesi.

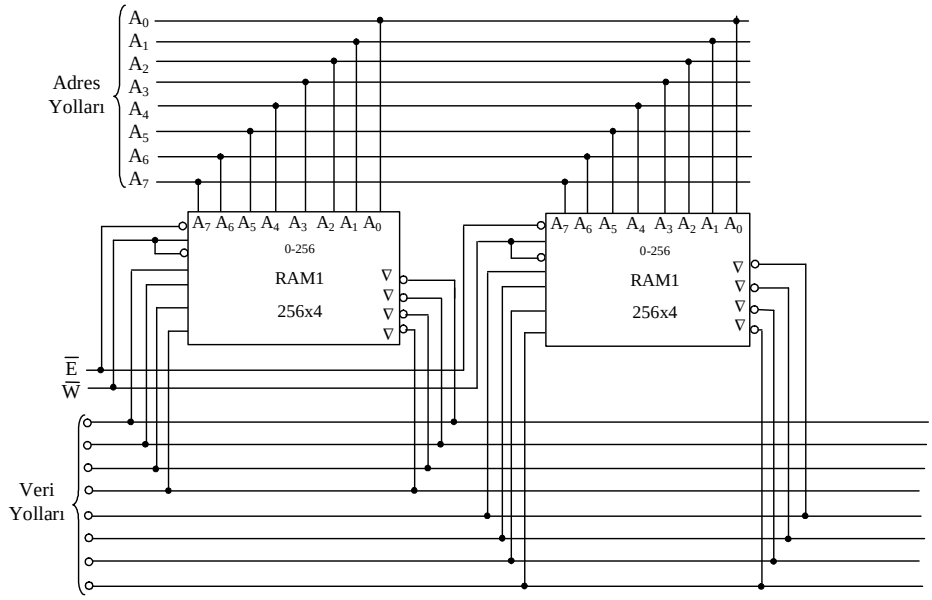


Şekil 13.46. ROM'ların kelime uzunluğunun genişletilmesi için gerekli bağlantı.

Örnek 3 : İki adet 256x4 RAM bellek kullanarak, 256x8 kapasitesinde bellek üretmek için gerekli bağlantıyı yapalım.

RAM bellekler, adres hattı bağlantısı olarak ROM bellekler ile aynı bağlantıya sahiptir. İki entegredeki aynı isimli adreslerin seçilmesi ile kelime uzunluğu artırılır (Şekil 13.47).

ROM belleklerden farklı olarak; oku / yaz kontrol girişlerinin yanı sıra, yazma girişlerine sahiptir. Yazma işlemi için W=0 uygulanır ve belleğe yazılacak bilgiler veri yoluna yerleştirilir. Aynı veri yolu yazma işlemi sırasında yazılacak verilerin belleğe girilmesi veya okuma işlemi sırasında bellekteki bilgilerin alınması için kullanılır. Yazma işlemi sırasında, üç konumlu çıkış yetkisizlendirilir ve çıkışların etkilenmesi engellenir (Şekil 13.47).



Şekil 13.47. İki adet 256x4 RAM ile 256x8 RAM bellek oluşturulması.

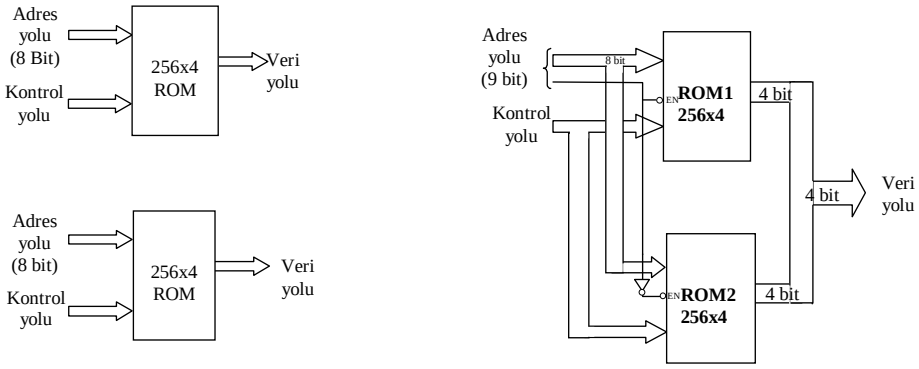
13.8.2. Bellek Adres Bölgesi Sayısının Arttırılması

Belleklerde saklanabilecek kelime sayısının arttırılması için, adres sayısının arttırılması gereklidir. Adres sayısının arttırılması için ise, adresleri temsil eden ve adres kod çözücüsüne uygulanan bitlerin sayısının arttırılması gerekir. Adres bit sayısının arttırılması entegrelerin birlikte kullanılması ve entegrelerin uygun şekilde seçilmesi ile sağlanır.

Birbirinden bağımsız olarak kullanılabilen 256x4 ROM bellekleri kullanarak (Şekil 13.48.a), 512x4 kapasitesinde bir bellek oluşturalım.

Kullanılan 256x4 ölçüsündeki belleklerde, 256 adet adres bölgesi bulunur ve bölgeleri temsil etmek amacı ile 8 bit kullanılır. Oluşturulmak istenen 512x4 kapasitesindeki yeni bellekte ise, 9 adet adres hattı bulunması gerekir (Şekil 13.48.b). 9. adres hattı olarak, bellek entegresinin ‘yetkilendirme’ girişi kullanılır ve ‘yetkilendirme’ girişi ile uygun olan entegre aktif hale getirilir. ‘Yetkilendirme’ girişi ile aktif hale getirilen entegredeki bellek bölgeleri seçilir ve seçilen adres bölgelerine erişilebilir.

İki belleğin bir arada kullanılması ile Şekil 13.49'daki lojik bağlantı ortaya çıkar. Oluşan yeni devrede, veri taşıtında dört hat bulunur ve her bir adres bölgesinde dört bitlik bilgi kaydedilebilir. Bunun anlamı; adres bölgelerinde bellekte saklanan kelime uzunluğunda bir değişiklik yoktur.



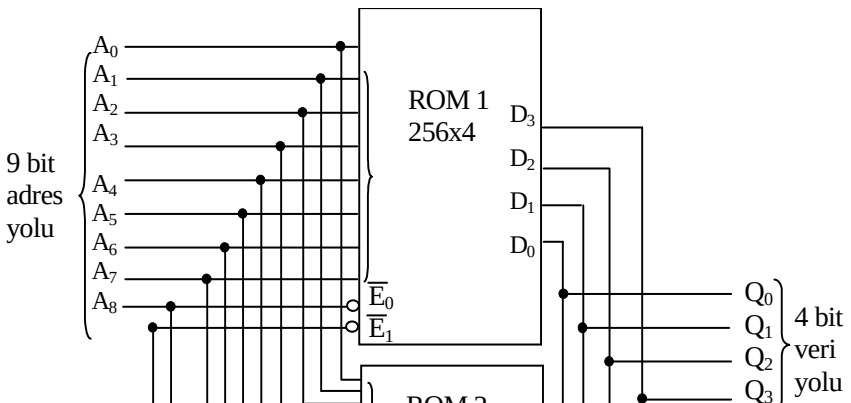
a) 256x4 ROM bellekler. **b)** İki adet bellek kullanılarak 512x4 bellek oluşturulması.

Şekil 13.48. Belleklerde adres sayısının genişletilmesi.

Entegredeki E_1 girişleri birleştirilerek, genel kontrol girişi olarak kullanılır. E_0 girişleri ise, 'A_8' adres girişi gibi bağlanır ve A_8'in değerine göre ROM1 veya ROM2 entegrelerinden birisi seçilir(Şekil 13.49).

A_8 girişi '0' yapıldığı zaman 'ROM1' seçilir ve bu entegre içerisindeki 256 adres bölgesine erişilebilir. A_0-A_7 adres bitlerinin durumuna göre 'ROM1' içerisindeki adresler seçilir. A_8=1 olması durumunda, 'DEĞİL' kapısında terslenen ve E_0 girişine uygulanan değer, 'ROM2' entegresini aktif hale getirir. 'ROM2' aktif olmasından dolayı, A_0-A_7 girişlerine uygulanan değere göre 'ROM2' içerisindeki adres bölgeleri seçilebilir.

ROM bellekler gibi, RAM belleklerinde bilgi saklama kapasitesinin artırılması mümkündür. RAM belleklerin kapasite artırımına örnek olması açısından, iki adet 16x4 RAM bellek kullanarak, 32x4 RAM bellek elde edilmesini inceleyelim.



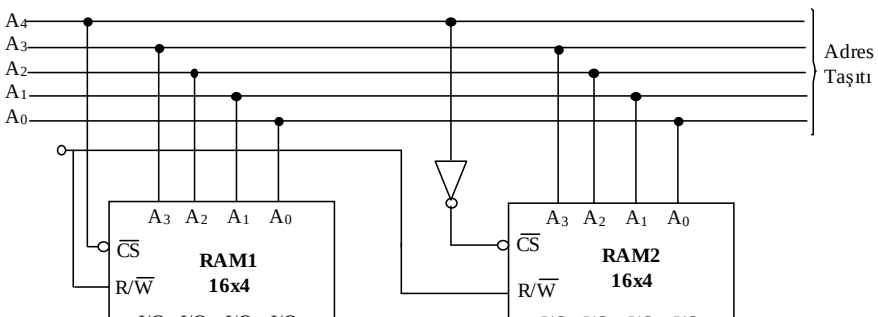
Şekil 13.49. ROM Belleklerde adres sayısının arttırılması.

Kullanılacak RAM bellekler dört bit kelime uzunluğuna sahip olduğundan, entegrelerin I/O girişleri ortak olarak bağlanabilir. Bellek entegreleri sıra ile yetkilendirilerek seçildiklerinden, çıkışların aynı taşıtlara bağlanması bir problem oluşturmaz. CS girişlerinin 'DEĞİL' kapısı üzerinden birlikte bağlanması, iki entegrenin aynı anda aktif olmasını engeller.

İki adet 16x4 RAM'ın Şekil 13.50'deki gibi bağlanması ile oluşan 32x4 kapasitesindeki bellekte, 32 adres bölgesi bulunur. 32 adres bölgesini temsil etmesi için gerekli 5. Adres hattı (A_4), CS girişinin kullanılması ile sağlanır. A_4 girişinin değerine göre RAM belleklerden birisi seçilir. A_0-A_3 adres girişlerinin değerlerine göre, seçilen entegredeki 16 adres bölgesinden birisine erişilir. Erişilen adres bölgesindeki bilgi veri yoluna yerleştirilir.

A_4=0 olması ile RAM1 seçilir ve RAM1 içerisindeki adresler aktif olur. Aktif olan bölgelerde R / W girişinin değerine bağlı olarak yazma veya okuma işlemi yapılabilir.

A_4=1 ise, RAM2 seçilir ve RAM2 içerisindeki adreslerde işlem yapılabilir. Seçilmeyene entegrenin veri çıkışları yüksek dirence sahiptir ve veri taşıtları üzerinde etkisi yoktur.



Adres bölgeleri 00000 – 01111 → RAM1
10000 – 11111 → RAM2

Şekil 13.50. İki 16x4 RAM bellek kullanarak 32x4 RAM bellek oluşturulması.

Tekrarlama ve Çalışma Soruları

1. Bellekleri tarif ediniz.
2. Bellekleri kullandıkları yere ve yapım malzemesine göre sınıflandırınız.
3. Belleklerde kapasite hız ilişkisini açıklayınız.
4. Yardımcı bellek elemanlarını sıralayınız.
5. Belleklere bilgi yerleştirilmesi yöntemleri nelerdir?
6. ‘Bellek hücresi’ ve ‘Bellek kelimesi’ terimlerini açıklayınız.
7. ‘Bellek dizisi’ terimini açıklayarak 16x4 ve 12x8 bellek dizileri oluşturunuz.
8. ‘Bellek kapasitesi’ teriminin tarifini yapınız.
9. ‘Byte’ ve ‘Nibble’ kaç bitlik bilgiyi tarif etmektedir.
10. ‘6Mx8’ ve ‘10Mx16’ kapasiteli belleklerde saklanabilecek bilgi miktarını bulunuz.
11. Belleklerde adres bölgesini bulma işlemini özetleyiniz.
12. ‘Belleklerde erişim zamanı’ terimini açıklayınız.
13. Belleklerde okuma işlemini özetleyiniz
14. Bellek entegrelerinde hangi girişler bulunur.

15. Belleklerde bulunan taşıt çeşitleri ve işlevleri nedir?
16. Bellekleri farklı bakış açıları ile sınıflandırınız.
17. ‘Manyetik bellekleri’ açıklayınız.
18. Manyetik bellek çeşitleri nelerdir?
19. Optik disk belleklerin genel çalışma prensiplerini özetleyiniz.
20. Yalnızca okunabilen belleklerin (ROM) yapısını açıklayınız.
21. Silinip-programlanabilir ROM’ların çalışma prensibini açıklayarak, çeşitlerini sıralayınız.
22. En yeni teknoloji ile üretilen bellek türü nedir?
23. ‘Bellek yetkilendirme’ işlemini tanımlayınız.
24. RAM bellek uygulama alanlarına örnek veriniz.
25. RAM belleklerin özelliklerini özetleyiniz.
26. RAM bellek iç yapısını şekille açıklayınız.
27. 4x4 RAM bellek organizasyonunu şekille kısaca açıklayınız.
28. RAM çeşitlerini özetleyiniz.
29. Statik RAM’ların genel özelliklerini açıklayınız.
30. SRAM hücresi lojik şemasını çizerek, kısaca özetleyiniz.
31. Dinamik RAM temel hücresinin genel özellikleri nelerdir?
32. DRAM’larda ‘bellek tazeleme’ işlemini özetleyiniz.
33. 8Kx1 DRAM bellek hücresinin hücre düzenlemesini şekille açıklayınız.
34. ROM belleklerde bulunan genel kısımlar nelerdir.
35. Flash bellekleri diğer bellek türleri ile karşılaştırınız.
36. Adres kod çözücülerin işlevi nedir?
37. Belleklerin ölçülerinin genişletilmesi işlemini özetleyiniz.
38. 4x4 ROM bellekler kullanarak 4x8 bellek elde ediniz.
39. 16x8 ROM bellekler kullanarak 16x16 bellek elde ediniz.
40. 16x4 RAM bellekler ile, 16x16 bellek elde ediniz.
41. RAM ve ROM belleklerde kelime uzunluğunun artırılması için işlem farkı varmıdır?

42. 16x16 ROM bellekler kullanarak 32x16 bellek oluşturunuz.
43. 16x16 RAM bellekler ile 32x32 bellek oluşturulabilir mi?
44. E^2EPROM belleklerin diğer ROM bellek türlerine üstünlükleri nelerdir.
45. ROM uygulama alanlarına 3 örnek veriniz.
46. ROM belleklerin veri tabloları oluşturmak amacıyla kullanımını açıklayınız.
47. Transistör kullanan SRAM ile MOS ve CMOS kullanan SRAM'ler arasındaki yapı farkı nedir?
48. 16Kx1 RAM entegresinin hücre düzenlemesini çiziniz.
49. Flash belleklerin genel özellikleri nelerdir?
50. Dinamik belleklerde bellek tazeleme işlemi hangi işlem sırasında yapılır?