dışında, yukarıdakine benzer:

$$(\underbrace{10}_{2}\ \underbrace{1100}_{C}\ \underbrace{0110}_{6}\ \underbrace{1011}_{B}\ .\ \underbrace{1111}_{F}\ \underbrace{0010}_{2})_2 = (2C6B.F2)_{16}$$

Her bir ikili hane grubuna tekabül eden onaltılı (veya sekizli) hane, Tablo 1.1'deki değerler incelendikten sonra kolayca hatırlanacaktır.

Sekizli veya onaltılı sistemden ikiliye çevrim, yukarıdakinin tersi bir işlemle yapılır. Her bir sekizli hane bunun üç haneli ikili eşdeğerine çevrilir. Benzer bir şekilde, her onaltılı hane bunun dört haneli eşdeğerine çevrilir. Aşağıdaki örnek bu işlemi göstermektedir:

$$(673.124)_8 = (\underbrace{110}_{6}\ \underbrace{111}_{7}\ \underbrace{011}_{3}\ .\ \underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{100}_{4})_2$$

$$(306.D)_{16} = (\underbrace{0011}_{3}\ \underbrace{0000}_{0}\ \underbrace{0110}_{6}\ .\ \underbrace{1101}_{D})_2$$

İkili sayılarla çalışmak zordur, çünkü ondalık eşdeğerlerinden üç dört kat daha fazla hane gerektirir. Örneğin ikili 111111111111 sayısı, ondalık 4095 sayısına eşdeğerdir. Ne var ki bilgisayarlarda ikili sayılar kullanılır ve bazen operatörün veya kullanıcının makineyle doğrudan doğruya ikili sayılar yoluyla iletişim kurması gerekir. Bilgisayardaki ikili sistemi aynen koruyan, ancak söz konusu kişinin dikkate alması gereken hane sayısını azaltan bir yöntemde ikili sistemle sekizli veya onaltılı sistem arasındaki ilişki kullanılır. Bu yöntemle insan, sekizli veya onaltılı sayılarla düşünür ve makineyle doğrudan iletişim gerekli olduğu zaman ilgili değeri inceleyerek gerekli çevrimi yapar. Örneğin ikili 111111111111 sayısı 12 hanelidir ve sekizli sistemde 7777 (dört hane) veya onaltılı sistemde FFF (üç hane) olarak ifade edilir. İnsanlar kendi aralarında bilgisayardaki ikili sayılardan söz ederken, sekizli veya onaltılı sistem daha çok arzu edilir, çünkü eşdeğerde ikili sayıda gerekli olan basamak sayısının üçte veya dörtte biriyle daha anlaşılır ifade edilebilmektedir. Ancak insan makineyle iletişim kurarken (konsol anahtarları, gösterge lambaları veya *makine dilinde* yazılan bir program yoluyla), sekizli veya onaltılıdan ikiliye ve ikiliden bu sistemlere çevrim, kullanıcı tarafından incelenerek yapılır.

1.5 TÜMLEYENLER

Sayısal bilgisayarlarda tümleyenler, çıkarma işlemini sadeleştirmek için ve mantık işlemleri için kullanılır. Her r-tabanlı sistem için iki tip tümleyen vardır: (1) r'nin tümleyeni ve (2) $(r-1)$'in tümleyeni. Tabanın değeri yerine konduğu zaman bu iki tip, ikili sayılarda 2'nin ve 1'in tümleyeni veya ondalık sayılarda 10'un ve 9'un tümleyeni adını alır.

r'nin Tümleyeni

n haneli bir tamsayı kısmı bulunan r tabanında bir N pozitif sayı için, N'nin tümleyeni $N \neq 0$ için $r^n - N$ ve $N=0$ için 0'dır. Aşağıdaki sayısal örnekler tanımın açıklık kazanmasını sağlayacaktır:

$(52520)_{10}$'un 10'a tümleyeni = $10^5 - 52520 = 47480$.
Sayıdaki basamak sayısı $n = 5$.
$(0.3267)_{10}$'un 10'a tümleyeni = $1 - 0.3267 = 0.6733$.
Tamsayı yok, bu nedenle $10^n = 10^0 = 1$.
$(25.639)_{10}$'un 10'a tümleyeni = $10^2 - 25.639 = 74.361$.
$(101100)_2$'nin 2'ye tümleyeni = $(2^6)_{10} - (101100)_2 = (1000000 - 101100)_2$
$= 010100$.
$(0.0110)_2$'nin 2'ye tümleyeni = $(1 - 0.0110)_2 = 0.1010$

Tanımdan ve örneklerden, ondalık bir sayının 10'un tümleyeninin, bütün en az değerlikli sıfırları olduğu gibi bırakıp sıfırdan farklı ilk en az değerlikli haneyi 10'dan çıkararak ve daha sonra da diğer bütün değerlikli haneleri 9'dan çıkararak bulunabileceği açıktır. 2'nin tümleyeni ise bütün en düşük değerlikli sıfırları ve sıfırdan farklı ilk haneyi olduğu gibi bırakıp diğer bütün anlamlı (değerlikli) hanelerdeki 1'lerin yerine 0 ve 0'ların yerine 1 yazarak bulunabilir. r'nin tümleyeninin bulunması için üçüncü ve daha basit bir yöntem, $(r-1)$'in tümleyeninin tanımından sonra verilmiştir.

Her r tabanı (1'den büyük, ancak 1'e eşit olmayan) için sayıların r'nin tümleyeni mevcuttur ve yukarıdaki tanımdan elde edilebilir. Burada verilen örneklerde $r=10$ (ondalık) ve $r=2$ (ikili) sayılar kullanılmaktadır, çünkü bizi en çok ilgilendiren bu iki tabandır. Tümleyenin adı kullanılan sayının tabanıyla ilgilidir. Örneğin 11 tabanlı bir sayının $(r-1)$ tümleyenine, 10'un tümleyeni denir, çünkü $r=11$ için $r-1=10$'dur.

(r-1)'in Tümleyeni

n haneli bir tamsayı kısmı ve m haneli bir kesirli kısmı bulunan r tabanında bir N pozitif sayı için, N'nin $(r-1)$ tümleyeni $r^n - r^{-m} - N$ olarak tanımlanır. Aşağıda bazı nümerik örnekler verilmiştir:

$(52520)_{10}$'un 9'a tümleyeni = $(10^5 - 1 - 52520 = 99999 - 52520 = 47479$.

Kesir yok, bu nedenle $10^{-m} = 10^0 = 1$
$(0.3267)_{10}$'un 9'a tümleyeni = $(1 - 10^{-4} - 0.3267) = 0.9999 - 0.3267 = 0.6732$
Tamsayı yok, bu nedenle $10^n = 10^0 = 1$
$(25.639)_{10}$'un 9'a tümleyeni = $(10^2 - 10^{-3} - 25.639) = 99.999 - 25.639 = 74.360$
$(101100)_2$'nin 1'e tümleyeni = $(2^6 - 1) - (101100) = (111111 - 101100)_2$
$= 010011$
$(0.0110)_2$'nin 1'e tümleyeni = $(1 - 2^{-4})_{10} - (0.0110)_2 = (0.1111 - 0.0110)_2$
$= 0.1001$

Örneklerden, ondalık bir sayının 9'un tümleyeninin, her haneyi 9'dan çıkarmak suretiyle bulunduğunu gördük. İkili bir sayının 1'in tümleyenini bulmak çok daha kolaydır: 1'ler 0, 0'lar da 1 yapılır. (r-1) tümleyeni kolayca bulunduğu için, r'nin tümleyeni istendiğinde bunu kullanmak bazen daha elverişlidir. Tanımlardan ve örneklerden sağlanan sonuçların kıyaslanmasından, r^{-m}, en az anlamlı haneye eklendikten sonra (r-1)'in tümleyeninden r'nin tümleyeninin elde edilebileceği sonucu çıkar. Örneğin 10110100 sayısının 2'nin tümleyeni, bu sayının 1'in tümleyeni 01001011'e 1 eklenerek bulunabilir (=01001100).

Tümleyenin tümleyeninin sayıyı ilk değerine döndürdüğünü söylemekte yarar var. N sayısının r'nin tümleyeni = r^n-N ve (r^n-N)'in tümleyeni = r^n-(r^n-N) = N'dir; ve aynı şey 1'in tümleyeni için de geçerlidir.

r'nin Tümleyenleriyle Çıkarma İşlemi

İlkokullarda çıkarma işlemini öğretmek için yararlanılan doğrudan yöntemde ödünç kavramı kullanılır. Bu yöntemde, çıkan hane buna karşılık gelen çıkarılan haneden büyük olduğu zaman, bir üst haneden 1 ödünç alırız. Kalem kağıtla çıkarma yapılırken bu en kolay yol gibi gözükür. Çıkarma işlemi sayısal cihazlar yoluyla yapıldığı zaman bu yöntemin, tümleyenlerin ve aşağıda anlatılan toplamanın kullanıldığı yöntem kadar etkili olmadığı anlaşılır.

Her ikisi de r tabanında olan iki pozitif sayının birbirinden çıkarılması (M-N) aşağıdaki şekilde yapılabilir:

1. Çıkarılan M'i, çıkan N'in r tümleyenine ekleyin.
2. 1. adımda elde edilen sonuçta elde olup olmadığına bakın.
   (a) Elde varsa atın.
   (b) Elde yoksa, 1. adımda elde edilen sayının r tümleyenini alın ve önüne bir eksi işareti koyun.

Aşağıdaki örnekler yönteme açıklık getirecektir:

ÖRNEK 1-5: 10'un tümleyenini kullanarak 7253 - 3250 işlemini yapın.

M = 72532
N = 03250

N'in 10'a tümleyeni = 96750

$$\begin{array}{r} 72532 \\ +\ 96750 \\ \hline \text{son elde} \rightarrow 1\ \ 69282 \end{array}$$

cevap : 69282

ÖRNEK 1-6: $(3250 - 72532)_{10}$ işlemini yapın.

M = 03250
N = 72532

N'in 10'a tümleyeni = 27468

$$\begin{array}{r} 03250 \\ +\ 27468 \\ \hline \text{elde yok}\ \ 30718 \end{array}$$

cevap : - 69282 = - (30718'in 10'a tümleyeni)

ÖRNEK 1-7: Aşağıdaki ikili sayılarla M - N işlemini yapmak için 2'nin tümleyenini kullanın.

(a) M = 1010100
N = 1000100

N'in 2'ye tümleyeni = 0111100

$$\begin{array}{r} 1010100 \\ +\ 0111100 \\ \hline \text{son elde} \rightarrow 1\ \ 0010000 \end{array}$$

cevap : 10000

(b) M = 1000100
N = 1010100

N'in 2'ye tümleyeni = 0111100

$$\begin{array}{r} 1000100 \\ +\ 0101100 \\ \hline \text{elde yok}\ \ 1110000 \end{array}$$

cevap : - 10000 = - 1110000'in 2'ye tümleyeni

Bu işlem şu şekilde kanıtlanır: N sayısının r tümleyenine M eklenmesi ($M+r^n$-N) verir. n haneli bir tamsayı kısmı bulunan sayılarda r^n, (n+1). konumda bir 1'e eşittir (yani "elde" denen şeye). Hem M hem de N pozitif kabul edildiğinden:

(a) $(M + r^n - N) \geq r^n$ ise $M \geq N$, veya
(b) $(M + r^n - N) < r^n$ ise $M < N$ olur.

(a) durumunda cevap pozitiftir ve M-N'ye eşittir, ki bu da r^n eldesi atıldığı zaman doğrudan doğruya elde edilen şeydir. (b) durumunda cevap negatiftir ve -(N-M)'ye eşittir. Elde olmaması negatif olduğunu gösteriyordu. Cevap, ikinci tümleyeni alıp bir eksi işareti koyarak bulunmuştur:

$$-[r^n - (M + r^n - N)] = -(N - M).$$

(r-1) Tümleyeniyle Çıkarma

(r-1) tümleyeniyle çıkarma işlemi, aşağıda gösterilen ve "son elde" denen bir farkın dışında r'nin tümleyeniyle çıkarmayla tamamen aynıdır. r tabanlı ve her ikisi de pozitif olan M-N çıkarma işlemi aşağıdaki şekilde yapılır:

1. Çıkarılan M'i, çıkan N'in (r-1) tümleyenine ekleyin.

2. İlk adımda elde edilen sonuçta elde olup olmadığına bakın.
   (a) Elde varsa, en az anlamlı haneye 1 ekleyin (son elde).
   (b) Elde yoksa, 1. adımda elde edilen sayının (r-1) tümleyenini alın ve önüne eksi işareti koyun.

Bu işlemin kanıtı, r'nin tümleyeni için verilene çok benzemektedir ve alıştırma olarak öğrenciye bırakılmıştır. Aşağıdaki örnekler işleme açıklık kazandıracaktır:

ÖRNEK 1-8: 1.5 VE 1.6. örnekleri, 9'un tümleyenlerini kullanarak tekrarlayın.

(a) $M = 72532$
$N = 03250$
N'in 9'a tümleyeni $= 96749$

```
              72532
          +   96749
          1 / 69281
son elde            +
                  1
              69282
```

cevap : 69282

(b) $M = 03250$
$N = 72532$
N'in 9'a tümleyeni $= 27467$

```
              03250
          +   27467
eldesiz     / 30717
```

cevap : - 69282 = - (30717'in 9'a tümleyeni)

ÖRNEK 1-9: 1-7. örneği, 1'e tümleyeni kullanarak tekrarlayın.

(a) M = 1010100
N = 1000100
N'in 1'e tümleyeni = 0111011

$$
\begin{array}{rr}
 & 1010100 \\
+ & 0111011 \\
\hline
\text{son elde } 1 & 0001111 \\
+ & 1 \\
\hline
 & 0010000
\end{array}
$$

cevap : 10000

(b) M = 1000100
N = 1010100
N'in 1'e tümleyeni = 0101011

$$
\begin{array}{rr}
 & 1000100 \\
+ & 0101011 \\
\hline
\text{elde yok} & 1101111
\end{array}
$$

cevap : - 10000 = - (1101111'in 1'e tümleyeni)

1'e ve 2'ye Tümleyenlerin Kıyaslanması

1'e ve 2'ye tümleyenlerin kıyaslanması, her birisinin avantajlarını ve dezavantajlarını ortaya çıkaracaktır. 1'in tümleyeninin avantajı, yapılması gereken tek şey 1'leri 0 ve 0'ları 1 yapmak olduğu için, sayısal bilgisayarlar tarafından kolayca kullanılabilmesidir. 2'nin tümleyeni iki yoldan elde edilebilir: (1) 1'in tümleyeninin en küçük değerlikli hanesine 1 ekleyerek ve (2) en küçük değerlikli konumlardaki bütün 0'ları ve ilk 1'i olduğu gibi bıraktıktan sonra kalan diğer bütün 1'leri 0, 0'ları da 1 yaparak. İki sayının, tümleyenler yoluyla çıkarılması sırasında 2'nin tümleyeni daha avantajlıdır, çünkü sadece bir aritmetik toplama işlemi gereklidir. Elde kalması halinde 1'in tümleyeni iki aritmetik toplama gerektirir. 1'in tümleyeninin, iki aritmetik sıfıra sahip olmak gibi bir dezavantajı daha vardır: hanelerin hepsi 0'ken ve hepsi 1'ken. Bu olguya açıklık getirmek açısından, iki eşit ikili sayının birbirinden çıkarıldığını varsayalım: 1100-1100=0.

1'in tümleyenini kullanırsak:

$$
\begin{array}{rr}
 & 1100 \\
+ & 0011 \\
\hline
+ & 1111
\end{array}
$$

- 0000, elde etmek için tekrar tümle

2'nin tümleyenini kullanarak:

$$\begin{array}{r} 1100 \\ +\ 0100 \\ \hline +\ 0000 \end{array}$$

2'nin tümleyeninde sadece bir aritmetik sıfır varken, 1'in tümleyeninde sıfır eksi veya artı olabilir, bu da işleri karmaşıklaştırabilir.

Sayısal bilgisayarlardaki aritmetik işlemler için çok yararlı olan tümleyenler 8 ve 9. Bölümlerde ele alınmıştır. Ancak, 1'in tümleyeni de daha sonra gösterileceği üzere, mantık işlemlerinde işe yaramaktadır, çünkü 1'in 0'a ve 0'ın 1'e çevrilmesi, bir mantık tersleme işleminin eşdeğeridir. Buna karşılık 2'nin tümleyeni sadece aritmetik uygulamalarla ilişkili olarak kullanılmaktadır. Sonuç olarak şu kuralı benimsemek yerinde olacaktır: türü belirtilmeksizin *tümleyen* terimi aritmetik olmayan bir işlemle ilgili olarak kullanılmıyorsa bunun 1'in tümleyeni olduğu varsayılmaktadır.

1-6 İKİLİ KODLAR

Elektronik sayısal sistemlerde, iki ayrı değere sahip sinyaller ve iki kararlı duruma sahip devre elemanları kullanılmaktadır. İkili sinyaller, ikili devre elemanları ve ikili sayılar (haneler) arasında doğrudan bir benzeşim söz konusudur. Örneğin n haneli ikili bir sayı, her birinin 0'a veya 1'e eşit çıkış sinyal eşdeğeri olan n kadar ikili devre elemanıyla gösterilebilir. Sayısal sistemler sadece ikili sayıları değil, ayrıca diğer birçok ayrık bilgi ögelerini de temsil edebilmekte ve kullanabilmektedir. Bir nicelikler grubu içinde ayırdedilebilen her ayrık bilgi ögesi ikili bir kodla gösterilebilir. Örneğin *kırmızı*, tayfın ayrı bir rengidir. *A* harfi alfabenin ayrı (belirgin) bir harfidir.

Tanım gereği bir *bit*, bir ikili hanedir. Bir ikili kodla ilişkili olarak kullanıldığında, 0 veya 1'e eşit bir ikili niceliği gösterdiğini düşünmek daha doğrudur. Bir ikili koddaki 2^n'li bir grubu göstermek için, en az n sayıda haneye ihtiyaç vardır. Bunun nedeni n kadar bitin, 2^n kadar farklı şekilde düzenlenebilmesidir. Örneğin dört ayrı nicelikten oluşan bir grup, her bir nicelik aşağıdaki bit kombinasyonlarına karşılık gelecek şekilde iki bitlik bir kodla gösterilebilir: 00, 01, 10 ve 11. Sekiz ögeli bir grup, her bir öge şunlardan sadece ve sadece birisiyle temsil edilecek şekilde, üç bitlik bir kod gerektirir: 000, 001, 010, 011, 100, 101, 110 ve 111. Bu örneklerden, n kadar bitli bir kodun ayrı bit kombinasyonlarının (birleşimlerinin), ikili formda 0'dan (2^n-1)'e kadar sayılarak bulunabileceği sonucu çıkar. Kodlanacak grubun eleman sayısı, 2'nin katları olmadığı zaman bazı bit kombinasyonları boş (tahsis edilmeden) kalır. 0, 1, 2,.....9'dan oluşan on ondalık hane böyle bir gruba örnek teşkil eder. On ögeyi birbirinden ayıran ikili bir kodun en az dört bit içermesi gerekir; çünkü üç bit en fazla sekiz ögeyi birbirinden ayırabilir. Dört bit ise 16 farklı birleşim oluşturabilir (16 farklı şekilde dizilebilir), ama sadece on hane kodlandığı için, kalan altı kombinasyon boştur ve kullanılmaz.

2^n kadar ayrı niceliği kodlamak için gerekli *minimum* bit sayısının *n* olmasına rağmen, ikili bir kod için kullanılabilecek *maksimum* bit sayısı diye bir sınır yoktur. Örneğin ondalık on hane on bitle kodlanabilir ve her ondalık haneye dokuz 0'lı ve bir 1'li bir bit birleşimi tahsis edilebilir. Bu ikili kodda 6 hanesi 0001000000 bit birleşimiyle gösterilebilir.

Ondalık Kodlar

Ondalık haneler için ikili kodlar en az dört bit gerektirmektedir. Dört veya daha fazla biti olası on ayrı birleşimle düzenleyerek çok çeşitli kodlar elde edilebilir. Buna birkaç örnek Tablo 1.2'de verilmiştir.

Tablo 1.2 Ondalık haneler için ikili kodlar

Ondalık sayı	(BCD) 8421	Artık-3	84-2-1	2421	(İki birli) 5043210
0	0000	0011	0000	0000	0100001
1	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

BCD (ikili kodlu ondalık), doğrudan bir ikili eşdeğer düzenlemesidir. İkili bitlere, konumlarına göre ağırlık vermek mümkündür. BCD kodunun ağırlıkları 8, 4, 2 ve 1'dir. Örneğin 0110 bit düzenlemesi, ondalık hane 6'yı gösterecek şekilde ağırlıklarla yorumlanabilir, çünkü 0x8+1x4+1x2+0x1=6. Örneğin 8, 4, -2, -1 kodunda olduğu gibi, ondalık bir koda negatif ağırlıklar vermek de mümkündür. Bu durumda 0100 bit birleşimi ondalık 2 sayısı olarak yorumlanır, çünkü 0x8+1x4+1x(-2)+0x(-1)=2. Tabloda verilen diğer iki kodun ağırlıkları 2421 ve 5043210'dır. Bazı eski bilgisayarlarda kulanılan bir ondalık kod, artık-3 kodudur. Bu ağırlıksız bir koddur; kod düzenlemesi, 3 eklendikten sonra ilgili BCD değerinden elde edilmektedir.

Sayılar sayısal bilgisayarda ya ikili olarak, ya da ikili bir kod vasıtasıyla ondalık olarak temsil edilir. Veriler belirlenirken kullanıcı verileri ondalık şeklinde vermek ister. Girilen ondalık sayılar bilgisayarın içinde ondalık bir kod vasıtasıyla saklanır. Her bir ondalık hane en az dört adet ikili saklama elemanı gerektirir. İkili olarak temsil edilen sayılarla makine içinde aritmetik işlem yapılacağı zaman ondalık sayılar ikiliye çevrilir. Sayıların hepsi tamamen kodlu haldeyken de doğrudan doğruya ondalık aritmetik işlemleri yapmak da mümkündür. Örneğin 395 ondalık sayısı ikiliye çevrildiği zaman 110001011'e eşittir ve dokuz ikili haneden oluşur. Aynı sayı makinede BCD koduyla temsil edildiği zaman her bir ondalık hane için dört bit olmak üzere toplam 12 bitlik yer işgal eder:001110010101. İlk dört bit 3'ü, ikinci dört bit 9'u ve son dört bit de 5'i temsil eder.

Ondalık bir sayının ikiliye *çevrilmesi* ile ondalık bir sayının ikili *kodlanması* arasındaki farkı anlamak çok önemlidir. Çevrimden elde edilen bitler ikili hanelerdir. Kodlamadan elde edilen bitler ise kullanılan kodun kurallarına göre düzenlenen 1 ve 0 birleşimleridir. Bu nedenle sayısal bir sistemdeki bir 1'ler ve 0'lar dizisinin bazen ikili bir sayıyı, bazen de belli bir ikili kodla belirlenen başka türlü bir ayrık bilgi niceliğini gösterebileceğini kavramak son derece önemlidir. Örneğin ondalık sayının 0 ila 9 arasında bir tamsayı olduğu sürece BCD kodu, hem bir kod hem de doğrudan bir ikili çevrim olarak seçilmiştir. 9'dan büyük sayılar durumunda çevrim ve kodlama tamamen farklıdır. Bu kavram o kadar önemlidir ki, başka bir örnekle tekrarlamakta yarar vardır. Ondalık 13 sayısının ikili çevrimi 1101'dir; buna karşılık aynı sayının BCD kodu ise 00010011'dir.

Tablo 1.2'de verilen beş adet ikili kod arasında BCD, kullanılması en doğal olanı gibi gözükmektedir; gerçekten de en sık rastlanan kod biçimidir. Tabloda verilen diğer dört bitli kod arasında, BCD'de bulunmayan ortak bir özellik vardır. Artık-3 ile 2,4,2,1 ve 8, 4, -2, -1, kendini tümleyen kodlardır; yani, 1'ler 0'la ve 0'lar 1'le değiştirilerek ondalık sayının 9 tümleyeni kolayca bulunur. Örneğin ondalık 395 sayısı 2, 4, 2, 1 kodunda 001111111011 olarak gösterilir. Bunun 9'a tümleyeni 604 ise 110000000100 ile gösterilir, ki bu da ilkindeki 1'leri 0, 0'ları da 1 yaparak kolayca elde edilir. Bu özellik, makinede ondalık sayılarla (ikili kodlu) aritmetik işlemler yapılırken yararlıdır; çıkarma işlemi ise 9'un tümleyeniyle hesaplanır.

Tablo 1.2'deki iki birli kod, hata-tespit özellikli yedi bitli bir koda örnek teşkil eder. Her ondalık hane, ilgili ağırlıklı sütunlara konan beş adet 0 ve iki adet 1'den oluşur. Sayısal sistemlerin ikili 1'i belli bir sinyalle ve ikili 0'ı başka bir sinyalle gösterdiği dikkate alınırsa, bu kodun hata-tespit özelliği anlaşılabilir. Sinyallerin bir yerden başka bir yere iletimi sırasında hata oluşabilir. Bir veya daha fazla bitin değeri değişebilir. Alıcı tarafındaki bir devre, gelen sinyaldeki 1 sayısının ikiden fazla (veya az) olduğunu belirleyebilir; ve alınan bitlerin birleşimi olabilecek birleşimlere uymuyorsa, bir hata olduğu belirlenir.

Hata Tespit Kodları

İster darbe modülasyonlu sinyaller şeklinde, ister sayısal bilgisayar girişi veya çıkışı şeklinde olsun, ikili bilgiler, iletişim ortamında örneğin kablo veya radyo dalgaları vasıtasıyla iletilebilir. Fiziksel iletişim ortamına giren her türlü harici gürültü (parazit), bit değerlerini 0'dan 1'e ve 1'den 0'a değiştirir. İletişim sırasında ortaya çıkan hataları tespit etmek amacıyla bir hata-tespit kodu kullanılabilir. Bu şekilde bulunan hata düzeltilemez, ancak varlığı gösterilir. Genel yöntem, hataların sıklığını gözlemektir. Hatalar iletilen bilginin tamamı üzerinde belirgin bir etki yaratmaksızın rastgele, arada bir ortaya çıkıyorsa, ya hiç bir şey yapılmaz, ya da hatalı mesaj tekrar gönderilir. Buna karşılık hatalar, alınan bilgilerin anlamını bozacak sıklıkta ortaya çıkıyorsa, sistemde arıza olup olmadığı kontrol edilir.

Tablo 1.3 Eşlik bit üretimi

(a) Mesaj	P (tek)	(b) Mesaj	P (çift)
0000	1	0000	0
0001	0	0001	1
0010	0	0010	1
0011	1	0011	0
0100	0	0100	1
0101	1	0101	0
0110	1	0110	0
0111	0	0111	1
1000	0	1000	1
1001	1	1001	0
1010	1	1010	0
1011	0	1011	1
1100	1	1100	0
1101	0	1101	1
1110	0	1110	1
1111	1	1111	0

Bir *eşlik biti*, toplam 1 sayısını tek veya çift yapmak için mesaja eklenen bir bittir. Dört bitlik bir mesaj ve P eşlik biti Tablo 1.3.'te verilmiştir. (a) şıkkında P, bütün 1'lerin toplamı, tek (beş bitin tamamında) olacak şekilde seçilmiştir. (b)'de ise P, bütün 1'lerin toplamı çift olacak şekilde seçilmiştir. Bilginin bir yerden başka bir yere iletimi sırasında eşlik biti aşağıdaki şekilde kullanılır. Gönderen tarafta mesaj (örneğimizde ilk dört bit), gerekli P'nin üretildiği bir "eşlik üretme" devresine uygulanır. Eşlik biti de dahil olmak üzere mesaj, hedefe gönderilir. Alıcı tarafta ise bütün bitler (örneğimizde beş bit), kabul edilen eşlemeyi kontrol etmek amacıyla bir "eşlik devresine" uygulanır. Bulunan eşlik, kabul edilen eşlik değerine uymuyorsa hata var demektir. Eşlik yöntemi, bir, üç veya diğer tek sayılı hata birleşimini tespit eder. Çift sayılı hata birleşimi tespit edilemez. Eşlik üretimi ve kontrolüne ilişkin daha ayrıntılı bir tartışma Bölüm 4.9'da bulunabilir.

Yansımalı Kod

Sayısal sistemler sadece ayrık biçimdeki verileri işleyebilecek şekilde tasarlanabilir. Birçok fiziksel sistem kesintisiz veri çıkışı sağlar. Bu verilerin, sayısal sisteme uygulanmadan önce, sayısal veya ayrık şekle dönüştürülmesi gerekir. Sürekli veya analog bilgiler, bir analog-sayısal dönüştürücü vasıtasıyla sayısal şekle dönüştürülür. Analog verilerden çevrilen sayısal verileri göstermek için bazı durumlarda Tablo 1.4'teki yansımalı kodu kullanmak yerinde olacaktır. Yansımalı kodun ikili sayılara kıyasla avantajı, yansımalı kodla yazılan bir sayının, bir sayıdan diğerine geçerken sadece bir bit değişmesidir. Tipik bir yansımalı kod uygulaması, bir mil (şaft) konumunun sürekli (kesintisiz) değişmesi analog verilerle temsil edilmesidir. Mil, bölümlere ayrılır ve her bölüme bir numara verilir. Bitişik bölümler bitişik yansımalı kodlu sayılarla temsil edildiği taktirde, herhangi iki bölümü birbirinden ayıran çizgide hata sezilirse belirsizlik azaltılmış olur. Tablo 1.4'te verilen yansımalı kod, bu türden çok sayıda olası koddan sadece birisidir. Farklı bir yansımalı kod elde etmek için, istenilen bit birleşimiyle başlayıp, iki sayının kod tahsisinin aynı olmaması koşuluyla, istenilen rastgele tarzda her defasında sadece bir biti 1'den 0'a veya 0'dan 1"e çevirmek yeterlidir. Yansımalı kod *Gray* kodu olarak da bilinmektedir.

Alfanümerik Kodlar

Sayısal bilgisayarlardaki birçok uygulama, hem sayılardan hem de harflerden oluşan verilerin işlenmesini gerektirir. Örneğin milyonlarca poliçe hamili bulunan bir sigorta şirketinde dosyaların işlenmesi için sayısal bir bilgisayar kullanılabilir. Poliçe sahibinin adını ikili sistemle göstermek için, alfabeye karşılık gelen bir ikili koda ihtiyaç vardır. Buna ek olarak, aynı ikili kodun, ondalık sayıları ve diğer özel karakterleri de göstermesi gerekir. Alfanümerik (bazen *alfamerik* olarak kısaltılır) bir kod, on ondalık haneden, alfabenin 26 harfinden ve $ gibi çeşitli özel sembollerden oluşan bir elemanlar grubunun iki kodudur. Bir alfanümerik gruptaki toplam eleman sayısı 36'dan fazladır. Bu nedenle en az altı bitle kodlanması gerekir ($2^6=64$, ama $2^5=32$ yetersiz).

Altı bitlik bir alfanümerik kodun olası düzenlemelerden birisi Tablo 1.5'te "iç (makine) kodu" altında verilmiştir. Bu kod, az sayıda değişiklikle, birçok bilgisayarda alfanümerik kodların makinede gösterimi için kullanılmaktadır. 64'ten fazla karakteri (küçük harfler ve sayısal bilgilerin iletimi için kullanılan özel

Tablo 1.4 Dört bitlik yansımalı kod

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

kontrol karakterleri) kodlama ihtiyacı, yedi ve sekiz bitli alfanümerik kodların geliştirilmesine yol açmıştır. Bu kodlardan birisi ASCII (Amerikan Standard Bilgi Değişim Kodu), diğeri ise EBCDIC (Genişletilmiş BCD Değişim Kodu) olarak bilinmektedir. Tablo 1.5'te verilen ASCII kodu, yedi bitten oluşur, ancak pratik açıdan sekiz bitli bir koddur, çünkü eşlik için her zaman sekizinci bir bit eklenir. Ayrık bilgiler delikli kartlar yoluyla aktarıldığı zaman, alfanümerik karakterler için 12 bitlik bir ikili kod kullanılır. Delikli bir kartta 80 sütun ve 12 satır vardır. Her bir sütunda alfanümerik bir karakter, ilgili satırlarda açılan deliklerle gösterilir. Bir deliğin varlığı 1 ve yokluğu da 0 olarak algılanır. 12 satır, üstten başlayarak 12, 11, 0, 1, 2....9 olarak işaretlenir. İlk üçü *zon* (alan) olarak adlandırılırken, kalan dokuz satır *nümerik* olarak adlandırılır. Tablo 1.5'teki 12 bitlik kart kodunda deliklerin açıldığı (ki bu delikler 1'e karşılık gelir) satırlar gösterilmiştir. Diğer boş satırların 0 olduğu varsayılmıştır. 12 bitli kart kodu, kullanılan bitlerin sayısı açısından etkisizdir. Bilgisayarların çoğu, giriş kodunu altı bitlik bir iç koda çevirir. Örnek olarak, "John Doe" adının iç kodla gösterimi şu şekildedir:

100001	100110	011000	100101	110000	010100	100110	010101
J	O	H	N	boşluk	D	O	E

Tablo 1.5 Alfanümerik Karakter Kodları

Karakter	6-Bitlik iç kodu	7-Bitlik ASCII kodu	8-Bitlik EBCDIC kodu	12-Bitlik kart kodu
A	010 001	100 0001	1100 0001	12,1
B	010 010	100 0010	1100 0010	12,2
C	010 011	100 0011	1100 0011	12,3
D	010 100	100 0100	1100 0100	12,4
E	010 101	100 0101	1100 0101	12,5
F	010 110	100 0110	1100 0110	12,6
G	010 111	100 0111	1100 0111	12,7
H	011 000	100 1000	1100 1000	12,8
I	011 001	100 1001	1100 1001	12,9
J	100 001	100 1010	1101 0001	11,1
K	100 010	100 1011	1101 0010	11,2
L	100 011	100 1100	1101 0011	11,3
M	100 100	100 1101	1101 0100	11,4
N	100 101	100 1110	1101 0101	11,5
O	100 110	100 1111	1101 0110	11,6
P	100 111	101 0000	1101 0111	11,7
Q	101 000	101 0001	1101 1000	11,8
R	101 001	101 0010	1101 1001	11,9
S	110 010	101 0011	1110 0010	0,2
T	110 011	101 0100	1110 0011	0,3
U	110 100	101 0101	1110 0100	0,4
V	110 101	101 0110	1110 0101	0,5
W	110 110	101 0111	1110 0110	0,6
X	110 111	101 1000	1110 0111	0,7
Y	111 000	101 1001	1110 1000	0,8
Z	111 001	101 1010	1110 1001	0,9
0	000 000	011 0000	1111 0000	0
1	000 001	011 0001	1111 0001	1
2	000 010	011 0010	1111 0010	2
3	000 011	011 0011	1111 0011	3
4	000 100	011 0100	1111 0100	4
5	000 101	011 0101	1111 0101	5
6	000 110	011 0110	1111 0110	6
7	000 111	011 0111	1111 0111	7
8	001 000	011 1000	1111 1000	8
9	001 001	011 1001	1111 1001	9
boşluk	110 000	010 0000	0100 0000	deliksiz
.	011 011	010 1110	0100 1011	12,8,3
(	111 100	010 1000	0100 1101	12,8,5
+	010 000	010 1011	0100 1110	12,8,6
$	101 011	010 0100	0101 1011	11,8,3
*	101 100	010 1010	0101 1100	11,8,4
)	011 100	010 1001	0101 1101	11,8,5
−	100 000	010 1101	0110 0000	11
/	110 001	010 1111	0110 0001	0,1
,	111 011	010 1100	0110 1011	0,8,3
=	001 011	011 1101	0111 1110	8,6

1.7 İKİLİ SAKLAMA VE KAYDEDİCİLER

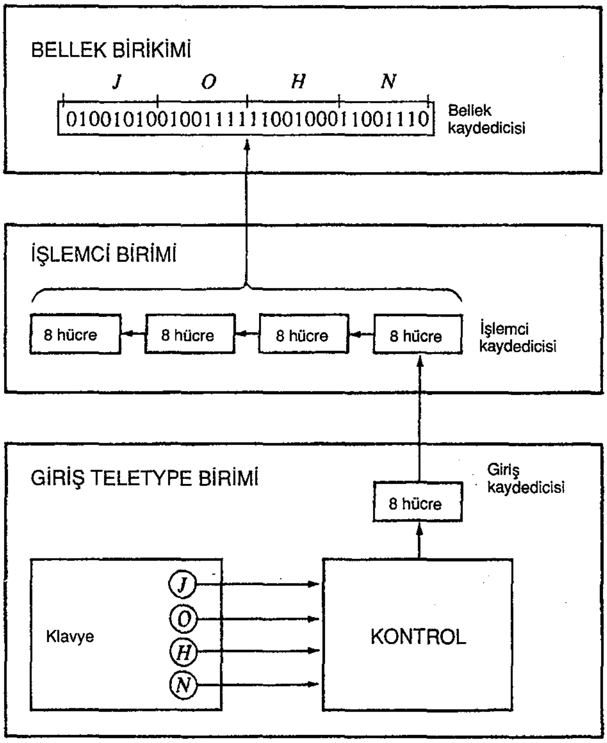
Sayısal bir bilgisayardaki ayrık bilgi ögelerinin, belli bir bilgi saklama ortamında fiziksel bir varlığı olması gerekir. Buna ek olarak, ayrık bilgi ögeleri ikili biçimde gösterildiği zaman, bilgi saklama ortamının, ayrı ayrı (bağımsız) bitleri saklamak için ikili saklama elemanları içermesi gerekir. Bir *ikili hücre*, iki kararlı duruma sahip olan ve bir bitlik bilgi saklayabilen bir aygıttır. Hücrenin girişi, hücreyi iki durumdan birine çeken (ayarlayan) uyarma sinyalleri alır. Hücrenin çıkışı ise iki durumu birbirinden ayıran fiziksel bir niceliktir. Bir hücre, bir kararlı durumdayken saklanan bilgi 1 ve diğer kararlı durumdayken saklanan ilgi 0'dır. İkili hücrelere örnek olarak elektronik flip-flop devreler, belleklerde kullanılan demir (ferrit) çekirdekler ve bir kartta delinen veya delinmeyen konumlardır.

Kaydediciler

Kaydedici, bir ikili hücreler grubudur. Bir hücre bir bitlik bilgi sakladığı için, bundan, n hücreli bir kaydedicinin n bit içeren ayrık bilgi niceliklerini saklayabileceği sonucu çıkar. Bir kaydedicinin *durumu*, 1'lerin ve 0'ların n katı kadar bir sayıdır; burada her bit kaydedicideki bir hücrenin durumunu gösterir. Kaydedicinin *içeriği* ise kaydedicide saklı bilgilere uygulanan yorumun bir fonksiyonudur. Örneğin aşağıdaki 16 hücreli kaydediciyi ele alalım:

1	1	0	0	0	0	1	1	1	1	0	0	1	0	0	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Fiziksel olarak kaydedicinin, her birinde 1 veya 0 değerlerinin saklandığı 16 adet ikili hücreden oluştuğu düşünülebilir. Kaydedicide saklanan bit düzenlemesinin (konfigürasyonunun) şekildeki gibi oluğunu varsayalım. Kaydedicinin durumu 16-katlı sayı 1100001111001001'dir. Açık olduğu üzere n kadar hücresi bulunan bir kaydedici olası 2^n durumundan birinde olabilir. Şimdi kaydedicinin içeriğinin bir ikili tamsayıyı gösterdiği varsayılırsa, kaydedicinin 0'dan 2^{16}-1'e kadar herhangi bir ikili sayıyı saklayabileceği açıktır. Elimizdeki örnekte kaydedicinin içeriği, ondalık 50121 sayısının eşdeğeridir. Ama kaydedicinin sekiz bitli kodla gösterilen alfanümerik karakterleri sakladığı varsayılırsa, kaydedicinin içeriği anlamlı herhangi iki anlamlı karakteri gösterir (tahsis edilmeyen bit birleşimleri anlamlı bilgi göstermez). EBCDIC kodunda yukarıdaki örnek iki karakteri gösterir: C (soldaki sekiz bit) ve I (sağdaki sekiz bit). Öte yandan, kaydedicinin içeriği dört bitli bir kodla gösterilen dört ondalık hane olarak yorumlanırsa, kaydedicideki veri dört haneli bir ondalık sayı olur. Artık-3 kodunda bu örnek ondalık 9096 sayısıdır. Kaydedici içeriği BCD'de anlamsızdır, çünkü 1100 bit birleşimi herhangi bir

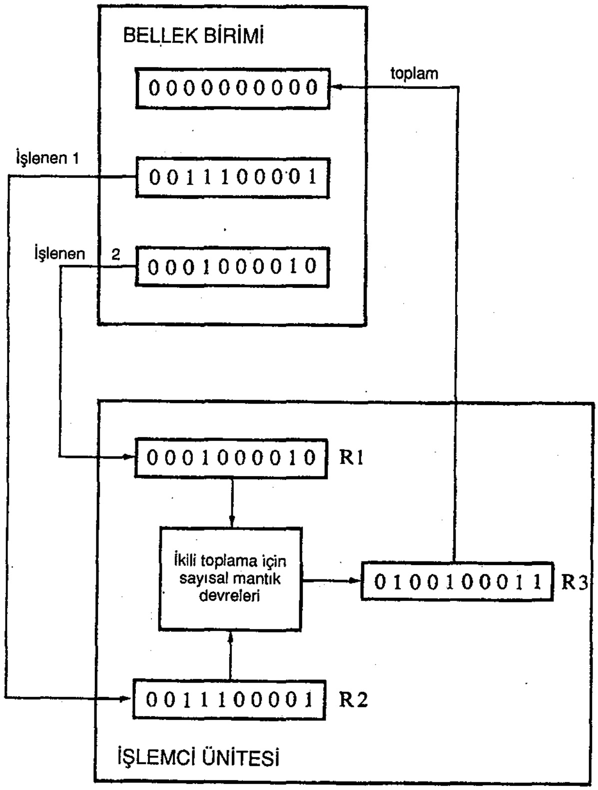


Şekil 1-2 Kaydedicilerle bilgi aktarımı

ondalık sayıya tahsis edilmemiştir. Bu örnekten, bir kaydedicinin bir veya daha fazla ayrık bilgi ögesini saklayabileceği ve aynı bit birleşiminin, farklı bilgi öge tipleri için farklı şekilde yorumlanabileceği sonucu çıkar. Kullanıcının kaydedicilerde anlamlı bilgiler saklaması ve bilgisayarın, bu bilgileri saklanan bilgilerin *türüne* göre işlemesi önemlidir.

Kaydedici Aktarımı

Sayısal bir bilgisayar, kaydedicileriyle tanımlanır. Bellek ünitesi (Şekil 1-1), sayısal bilgilerin saklanması için kullanılan binlerce kaydedicinin toplamından başka bir şey değildir. İşlemci ünitesi, üzerinde işlemlerin yapıldığı işlenenleri saklayan çeşitli kaydedicilerden oluşur. Kontrol ünitesinde çeşitli bilgisayar sıralarının kaydının tutulduğu çeşitli kaydediciler kullanılır; her giriş veya çıkış birimlerinde, birime veya birimden aktarılan bilgileri saklamak için en az bir kaydedicinin bulunması gerekir. Sayısal sistemlerde temel bir işlem olan *kaydediciler-arası bir işlem*, bir kaydedicide bulunan bilgilerin başka bir kaydediciye aktarılmasından oluşur. Şekil 1.2'de, kaydediciler arasında bilgi



Şekil 1- 3 İkili bilgi işleme örneği

aktarımı ve ikili bilgilerin bir teletype klavyesinden bellek ünitesindeki bir kaydediciye aktarılması gösterilmiştir. Giriş teletype ünitesinde bir klavye, bir kontrol ünitesi ve bir giriş kaydedicisi bulunduğu varsayılmıştır. Her tuşa basıldığında, kontrol ünitesi, eşdeğer sekiz bitlik alfanümerik karakter kodunu giriş kaydedicisine girer. Kullanılan kodun, tek-eşli sekizinci bite sahip bir ASCII kodu olduğunu varsayacağız. Giriş kaydedicisinden gelen bilgi, bir işlemci kaydedicisinin sekiz adet en az anlamlı hücresine aktarılır. Tuşlara tekrar basıldığında kontrol ünitesinin yeni bir sekiz bitlik kod girmesini mümkün kılmak için giriş kaydedicisi her aktarımdan sonra silinir. İşlemci her sekiz bitli karakterin işlemci kaydedicisine aktarılmasından önce, bir önceki karakter yeni karakterin solundaki bitişik sekiz hücreye kaydırılır. Dört karakterin transferi tamamlandığı zaman işlemci kaydedicisi dolar ve kaydedicinin içeriği bir bellek kaydedicisine aktarılır. Şekil 1.2'deki bellek kaydedicisinde saklanan bilgiler, ilgili dört tuşa basıldıktan sonra JOHN karakterlerinin aktarılmasıyla oluşmuştur.

Ayrık bilgi niceliklerini ikili formda işlemek için bilgisayarın (1) işlenecek verileri tutan kaydediciler ve (2) ayrık bit bilgilerini işleyecek devre elemanlarıyla donatılması gerekir. Verileri tutmak için kullanılan en yaygın aygıt kaydedicidir. İkili değişkenlerin işlenmesi sayısal mantık devreleri vasıtasıyla yapılır. Şekil 1.3'te, 10 bitlik iki adet ikili sayının toplanması işlemi verilmiştir. Şemada, normalde binlerce kaydediciden oluşan bellek ünitesinin sadece üç kaydedicisi gösterilmiştir. İşlemci ünitesinin şekildeki bölümü R1, R2 ve R3 ile gösterilen üç kaydedici ile R1 ve R2'deki bitleri işleyen ve aritmetik toplamlarına eşit olan ikili sayıyı R3'e aktaran sayısal mantık devrelerinden oluşmaktadır. Kaydedici bellekleri bilgi saklamasına karşılık iki işleneni işleme kapasitesinden yoksundur. Ne var ki bellekte saklanan bilgiler işlemci kaydedicilerine aktarılabilmektedir. İşlemci kaydedicilerinde elde edilen sonuçlar, tekrar ihtiyaç duyuluncaya kadar saklanmak üzerine bellek kaydedicilerine geri aktarılabilmektedir. Şemada, iki bellek kaydedicisinden R1 ve R2'ye aktarılan iki işlenenin içeriği görülmektedir. Sayısal mantık devrelerinin bulduğu toplam, R3 kaydedicisine aktarılır. Bu noktada R3'ün içeriği bellek kaydedicilerinden birine geri aktarılabilir.

Son iki örnek, sayısal bir sistemin çok basit bir yöntemle gerçekleştirilen bilgi akış yeteneklerini göstermektedir. Sistemin kaydedicileri, ikili bilgilerin saklanması ve tutulması için gerekli temel ögelerdir. Sayısal mantık devreleri bilgileri üzerinde işlem yapar. Sayısal mantık devreleri ve işlem yapabilme yetenekleri bir sonraki bölümde anlatılacak. Kaydediciler ve bellek ise 7. Bölümde ele alınmaktadır.

1-8 İKİLİ MANTIK

İkili mantık, iki ayrık değer alabilen değişkenleri ve mantıksal anlam taşıyan işlemleri ele alır. Değişkenlerin alabileceği iki değer farklı şekillerde adlandırılabilir (örneğin *doğru* ve *yanlış*, *evet* ve *hayır*, vs.), ancak buradaki amacımız için bit terimleriyle düşünmek ve 1 ve 0 değerlerini kullanmak yerinde olacaktır. İkili bilgilerin kullanılmasını ve işlenmesini matematiksel bir yöntemle anlatmak için ikili mantık kullanılmaktadır. Bu sayısal sistemlerin analizi ve tasarımı için özellikle uygundur. Örneğin Şekil 1.3'teki ikili aritmetik işlemleri yapan sayısal mantık devrelerini ifade etmenin en uygun şekli ikili değişkenler ve mantık işlemleridir. Bu kısımda anlatılacak ikili mantık, Boole cebiri denen bir cebirin eşdeğeridir. İki değerli Boole cebiri 2. Bölümde daha ayrıntılı olarak ele alınacaktır. Bu bölümün amacı, Boole cebirine kısa bir hatırlatma yapmak ve sayısal mantık devreleriyle ikili sinyaller arasındaki ilişkilerini ortaya koymaktır.

İkili Mantığın Tanımı

İkili mantık, ikili değişkenlerden ve mantıksal işlemlerden oluşur. değişkenler, A, B, C, x, y, z, vb. harflerle gösterilir; burada her değişken ancak ve ancak olası iki ayrı değerden birini alabilir: 1 ve 0. Üç temel mantık işlemi vardır: VE (AND), VEYA (OR) ve DEĞİL (NOT).

1. VE: Bu işlem, bir noktayla veya bir işlemcinin bulunmamasıyla gösterilir. Örneğin $x \cdot y = z$ veya $xy = z$, "x VE y eşit z" olarak okunur. Mantıksal VE işlemi, ancak ve ancak $X=1$ ve $y=1$ ise $z=1$'dir şeklinde yorumlanır; aksi taktirde $z=0$'dır. (x, y ve z'nin ikili değişkenler olduğunu ve sadece 1'e veya 0'a eşit olabileceğini unutmayın.)

2. VEYA: Bu işlem artı işaretiyle gösterilir. Örneğin $x+y=z$, "x VEYA y eşit z" olarak okunur; bu da $x=1$ *veya* $y=1$ *veya* hem $x=1$ hem de $y=1$ olması halinde $z=1$ olduğu anlamına gelir. Hem x hem de y 0'a eşitse, bu durumda $z=0$'dır;

3. DEĞİL: Bu işlem bir apostrofla (bazen de tireyle) gösterilir; Örneğin $x'=z$ (veya $\bar{x}=z$), "x z'ye eşit değil" olarak okunur; bu da z'nin, x'in değili olduğu anlamına gelir. Başka bir değişle $x=1$ ise $z=0$ olacaktır; buna karşılık $x=0$ ise $z=1$ olacaktır.

İkili mantık ikili aritmetiğe benzer; VE ve VEYA, sırasıyla çarpma ve toplama işlemleriyle bazı benzerlikler gösterir. Aslında, VE ve VEYA için kullanılan semboller, çarpma ve toplama için kullanılanlarla aynıdır. Ne var ki ikili mantığın ikili aritmetikle karıştırılmaması gerekir. Bir aritmetik değişkeninin, birçok haneden oluşabilen bir sayıyı gösterdiği anlaşılmalıdır. Oysa bir mantık değişkeni her zaman için ya 1 ya da 0 olur. Örneğin ikili aritmetikte 1+1 = 10'ken ("bir artı bir eşit 2" okunur), ikili mantıkta 1+1=1 olur ("bir VEYA bir eşit bir" olarak okunur).

x ve y değerlerinin her birleşimi için, mantık işleminin tanımıyla belirlenen bir z değeri vardır. Bu tanımlar, *doğruluk tabloları* kullanılarak özet şeklinde verilebilir. *Doğruluk tablosu, değişkenlerin alabileceği olası bütün bileşimleri içeren ve değişkenlerin alabileceği değerlerle işlem sonucu arasındaki ilişkiyi gösteren bir tablodur.* Örneğin x ve y değişkenli VE ve VEYA işlemlerinin

Tablo 1.6 Mantıksal işlemlerin doğruluk tablosu

VE		VEYA		DEĞİL	
x y	$x \cdot y$	x y	$x+y$	x	x'
0 0	0	0 0	0	0	1
0 1	0	0 1	1	1	0
1 0	0	1 0	1		
1 1	1	1 1	1		

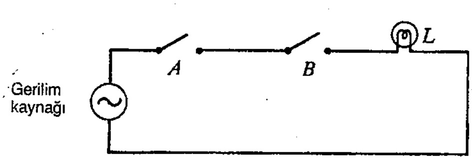
doğruluk tablosu, çiftler halinde birleştirilmesi durumunda değişkenlerin alabileceği olası bütün değerler yazılarak elde edilebilir. Daha sonra her bir birleşim için elde edilen işlem sonucu ayrı bir sırada yazılır. VE, VEYA ve DEĞİL işlemlerinin doğruluk tabloları Tablo 1.6'da verilmiştir. Bu tablolar işlemlerin tanımını açıkça göstermektedir.

Anahtarlama Devreleri ve İkili Sinyaller

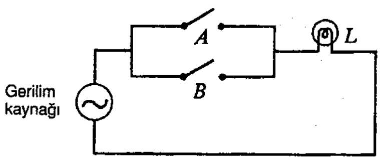
İkili değişkenlerin kullanılmasına ve ikili mantık uygulamasına, Şekil 1.4'teki basit anahtarlama devreleri örnek gösterilebilir. A ve B el (manuel) anahtarlarının, anahtar açıkken 0'a ve kapalıyken 1'e eşit değerler alan iki ikili değişkeni gösterdiğini varsayalım. Benzer bir şekilde L lambası da yanıkken 1'e ve sönükken 0'a eşit olan üçüncü bir ikili değişkeni göstersin. Anahtarlar seri bağlı ise, *A ve B* kapalıyken ışık yanar. Buna karşılık paralel bağlıysa ışık, A veya B kapalıyken yanar. Bu iki devrenin, ikili mantıkta VE ve VEYA işlemlerinin yardımıyla sırasıyla şu şekilde ifade edilebileceği açıktır:

$L = A \,.\, B$ Şekil 1-4(a)'daki devre için
$L = A + B$ Şekil 1-4(b)'deki devre için

Elektronik sayısal devrelere bazen *anahtarlama devreleri* de denir, çünkü bu devreler, bir transistör gibi aktif elemanı ileten (anahtar kapalı) veya iletmeyen (anahtar açık) durumdaki bir anahtar gibi davranır. Anahtarı elle değiştirmek yerine, elektronik bir anahtarlama devresi, aktif elemanın iletim veya iletmeme durumunu kontrol etmek için ikili sinyalleri kullanır. Gerilim veya akım gibi elektrik sinyalleri, sayısal bir sistemde, iki durumdan birisinde bulunur (geçişler hariç). Örneğin gerilimle çalışan devreler, mantık-1 veya mantık-0'a eşit olan bir ikili değişkeni gösteren iki ayrı gerilim seviyesine tepki verir. Örneğin belli bir sayısal sistemde mantık-1, nominal değeri 3 volt olan bir sinyal ve mantık-0 da anma değeri 0 volt olan bir sinyal olarak tanımlanabilir. Şekil 1.5'te de görüleceği üzere, gerilim düzeyinin anma değerinden sapması kabul edilebilir bir limit vardır. Kabul edilebilir bölgeler arasındaki ara bölgeden sadece durum geçişleri sırasında

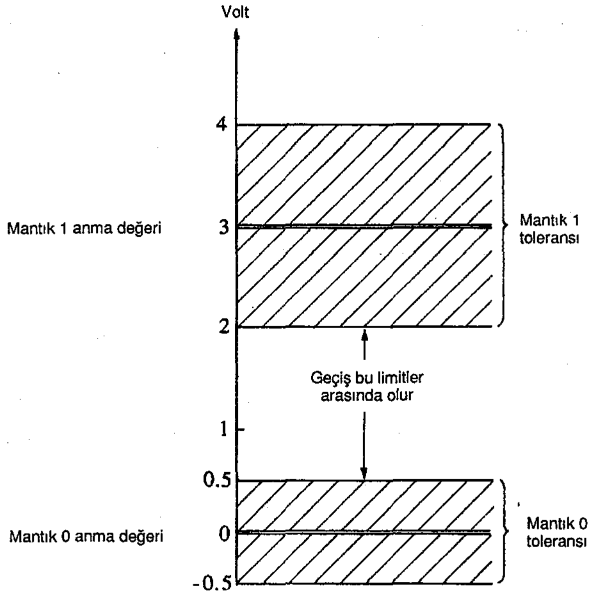


(a) Seri anahtarlar - mantık VE



(b) Paralel anahtarlar - mantık VEYA

Şekil 1.4 İkili mantığı gösteren anahtarlama devreleri



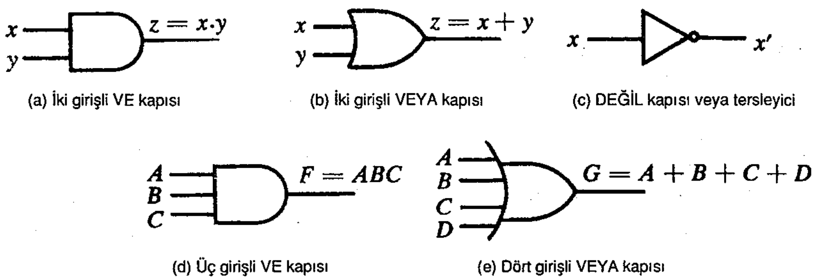
Şekil 1.5 İkili sinyaller örneği

geçilir. Sayısal devrelerin giriş uçları, kabul edilebilir tolerans limitleri dahilinde ikili sinyaller alır ve çıkış ucunda belirlenen toleranslar dahilinde kalan ikili sinyallerle tepki verir.

Mantık Kapıları

Elektronik sayısal devrelere *mantık devreleri* de denilmektedir, çünkü uygun girişle, mantıksal işlem (manipulasyon) yolları oluştururlar. Hesaplama veya kontrol için istenen her bilgi, çeşitli mantık devreleri birleşimlerinden, her biri bir değişkeni temsil eden ve bir bitlik bilgi taşıyan ikili sinyaller geçirmek suretiyle işlenebilir. VE, VEYA ve DEĞİL mantık işlemlerini yapan mantık devreleri ve sembolleri Şekil 1.6'da verilmiştir. *Kapı* denen bu devreler, giriş mantık koşullarının yerine getirilmesi halinde mantık-1 veya mantık-0 çıkış sinyali üreten donanım bloklarıdır. Aynı devre tipi için dört farklı isim kullanılmasına dikkat edin: sayısal devreler, anahtarlama devreleri, mantık devreleri ve kapılar. Bu isimlerin hepsi yaygın olarak kullanılmaktadır, ancak biz bu devreleri VE, VEYA ve DEĞİL kapıları olarak anacağız. İkili bir sinyali terslediği için DEĞİL kapısına bazen *tersleyici devre* de denir.

Şekil 1.6'daki iki giriş kapısındaki x ve y giriş sinyalleri, olası dört durumdan birisinde bulunabilir: 00, 01, 11, ya da 10. Bu giriş sinyalleri, VE ve VEYA



(a) İki girişli VE kapısı (b) İki girişli VEYA kapısı (c) DEĞİL kapısı veya tersleyici

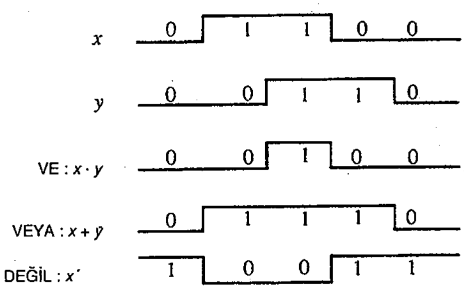
(d) Üç girişli VE kapısı (e) Dört girişli VEYA kapısı

Şekil 1.6 Sayısal mantık devrelerinin sembolleri

kapılarının çıkış sinyalleriyle birlikte Şekil 1.7'de verilmiştir. Şekil 1.7'deki zamanlama diyagramları, her bir devrenin olası dört giriş ikili birleşiminin herbirine verdiği cevabı göstermektedir. DEĞİL kapısına "tersleyici" denmesinin nedeni, x sinyali (tersleyicinin girişi) ile x' sinyalinin (tersleyicinin çıkış) karşılaştırılmasından açıkça anlaşılacaktır.

VE ve VEYA kapılarının ikiden fazla girişi olabilir. Şekil 1.6'da, üç girişli bir VE kapısıyla dört girişli bir VEYA kapısı verilmiştir. Üç girişli VE kapısı, üç giriş sinyalinin tamamının mantık 1 olması halinde bir mantık 1 çıkışıyla tepki vermektedir. Girişlerden birinin mantık 0 düzeyinde olması halinde çıkış mantık 0 olmaktadır. Dört girişli OR kapısı, girişlerden herhangi birisinin mantık 1 olması halinde mantık 1 tepkisi (cevabı) vermektedir. Bütün giriş sinyallerinin mantık 0 olması halinde bu kapının çıkışı da mantık 0 olmaktadır.

İkili mantığın matematik sistemi daha çok Boole veya anahtarlama cebiri olarak bilinmektedir. Karmaşık sayısal devre ağlarının çalışmasını anlatmak için bu cebir rahatlıkla kullanılmaktadır. Sayısal sistem tasarımcıları, devre şemalarını cebirsel ifadelere ve cebirsel ifadeleri devre şemalarına dönüştürmek için Boole cebiri kullanmaktadır. 2. ve 3. Bölümler, Boole cebiri, özellikleri ve işlem yeteneklerine ayrılmıştır. 4. Bölümde, kapı devreleri arasındaki ilişkileri matematiksel olarak ifade etmek için Boole cebirinin nasıl kullanıldığı ele alınmıştır.



Şekil 1.7 Şekil 1.6'daki (a), (b) ve (c) kapıları için giriş-çıkış sinyalleri

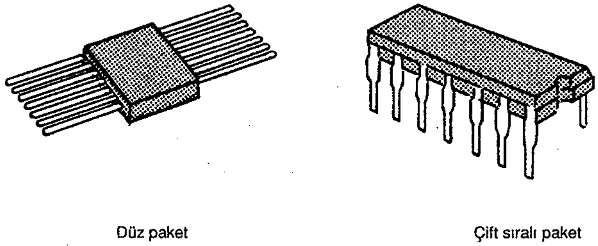
1.9 ENTEGRE DEVRELER

Sayısal devreler değişmez olarak entegre (tümleşik) devrelerden yapılmaktadır. Bir entegre devre (IC olarak kısaltılır), silisyumdan üretilen, transistörler, diyodlar, dirençler ve kondansatörler gibi çeşitli elektrik elemanları içeren ve *yonga* adı verilen küçük bir yarıiletken kristaldir. Yonga içindeki çeşitli elemanlar birbirine bağlanarak elektronik bir devre oluşturulur. Yonga metal veya plastik bir ambalaj içine yerleştirilir ve bağlantılar dış pinlere (bacaklara) kaynaklanarak IC elde edilir. Entegre devreler, birbirinden ayrılabilir aksamlardan oluşan diğer elektronik devrelerden farklıdır, çünkü IC'deki müstakil aksamlar (bileşenler) çıkarılamaz veya ayrılamaz ve ambalajın içindeki devreye sadece harici pinler vasıtasıyla erişilebilir.

Entegre devreler, Şekil 1.8'de gösterildiği gibi, *düz* paket ve *çift-sıralı paket* (DIP) olmak üzere iki tür ambalaj içinde üretilmektedir. Bunlardan çift-sıralı paket, düşük fiyatı ve devre kartlarına (board) kolay takılması nedeniyle en yaygın kullanılanıdır. IC pakedinin kılıfı plastik veya seramiktir. Paketlerin çoğu standart ebatlara sahiptir ve pin (bacak) sayısı 8 ila 64 arasında değişmektedir. Her IC pakedinin üzerinde, tanım amaçlı nümerik bir kod vardır. Her satıcı, çeşitli ürünlerle ilgili gerekli verileri sağlayan bir veri kitabı veya katalogu yayınlar.

IC pakedinin ebadı çok küçüktür. Örneğin dört VE kapısı, 20x8x3 milimetre boyutlarındaki 14-bacaklı bir çift sıralı paket içine yerleştirilir. Tam bir mikroişlemci, 50x15x14 mm boyutlarındaki 40-bacaklı bir çift-sıralı paket içine sığmaktadır.

Entegre devreler, ebatlarının çok küçük olmasının yanısıra, ayrık aksamlı elektronik devrelere kıyasla daha başka avantajlara ve yararlara da sahiptir. IC'lerin maliyeti çok düşüktür, bu da kullanımı ekonomik kılmaktadır. Güç tüketimlerinin çok az olması sayısal sistemlerin çalışmasını daha ekonomik kılmaktadır. Arızaya karşı yüksek bir güvenirliğe sahiptir, bu nedenle sayısal sistemler onarıma pek ihtiyaç duymaz. Çalışma hızı daha yüksektir, bu da bunları yüksek hızlı işlemler için uygun kılmaktadır. Entegre devrelerin kullanılması harici kablo bağlantılarının sayısını azaltmaktadır, çünkü bağlantıların birçoğu pakedin içinde yapılmıştır.



Şekil 1.8 Entegre devre paketleri

Bütün bu avantajlardan ötürü sayısal sistemler her zaman için entegre devrelerle inşa edilmektedir.

Entegre devreler iki genel kategoride toplanır: *lineer (doğrusal) ve sayısal.* Doğrusal IC'ler, yükselteç ve gerilim karşılaştırıcı türünden elektronik işlevler sağlamak amacıyla sürekli (kesintisiz) sinyallerle çalışır. Sayısal entegre devreler ise birbirine bağlı sayısal kapılardan oluşur ve ikili sinyallerle çalışır. Burada sadece sayısal entegre devreleri ele alacağız.

IC teknolojisi geliştikçe, tek bir silisyum yongası üzerine yerleştirilebilen kapı sayısı önemli ölçüde artmaktadır. Az sayıda dahili kapıya sahip IC'ler ile onlarca veya yüzlerce kapıya sahip IC'ler arasındaki fark, pakedin küçük-, orta- veya büyük-ölçekli entegrasyon (tümleşim) devresi olmasıyla tanımlanır. Tek paket içindeki birkaç mantık kapısı, küçük-ölçekli bir entegre (SSI) aygıtı oluşturur. IC'nin orta-ölçekli (MSI) olarak sınıflandırılabilmesi için, komple bir mantık fonksiyonunu yerine getirmesi ve 10 ila 100 arasında kapıya sahip olması gerekir. Büyük ölçekli entegre (LSI) aygıtı bir mantık fonksiyonunu yerine getirir ve 100'den fazla mantık kapısına sahiptir. Ayrıca tek yonga üzerinde binlerce kapıya sahip çok büyük ölçekli entegre (VLSI) aygıtlar da mevcuttur.

Bu kitap boyunca ele alınan birçok sayısal devre şeması, müstakil kapılarının ve bağlantılarının ayrıntılarına kadar gösterilmiştir. Bu tür şemalar, belli bir fonksiyonun mantık yapısını göstermek açısından yararlıdır. Ne var ki pratikte fonksiyonun bir MSI veya LSI aygıtından alınabileceğini ve kullanıcının, ara kapıların giriş ve çıkışlarına değil, sadece harici giriş ve çıkışlara erişebileceini unutmamak gerek. Örneğin sistemine bir kaydedici eklemek isteyen bir tasarımcı büyük bir ihtimalle bir fonksiyonu şemalarda gösterilen müstakil sayısal devrelerden tasarlamak yerine mevcut bir MSI devresini kullanarak uygulamayı tercih edecektir.

REFERANSLAR

1. Richard, R.K., *Arithmetic Operations in Digital Computers.* New York: Van Nostrand Co., 1955.
2. Flores, I., *The Logic of Computer Arithmetic.* Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1963
3. Chu, Y., *Digital Computer Design Fundamentals.* New York: McGraw-Hill Book Co., 1962, Chaps. 1 and 2.
4. Kostopoulos, G. K., *Digital Engineering.* New York: John Wiley & Sons, Inc., 1975, Chap. 1.
5. Rhyne, V. T., *Fundamentals of Digital Systems Design.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973, Chap. 1.

PROBLEMLER

1.1. Ondalık ilk 20 sayıyı 3 tabanında yazın.

1.2. Aşağıdaki sayıları, ondalık sisteme çevirmeksizin verilen tabanda toplayın ve çarpın.
(a) $(1230)_4$ ve $(23)_4$
(b) $(135.4)_6$ ve $(43.2)_6$
(c) $(367)_8$ ve $(715)_8$
(d) $(296)_{12}$ ve $(57)_{12}$

1.3. Ondalık 250.5 sayısını 3, 4, 7, 8 ve 16 tabanlarına çevirin.

1.4. Şu ondalık sayıları ikili sayılara çevirin: 12.0625, 10^4, 673.23 ve 1998.

1.5. Şu ikili sayıları ondalığa çevirin: 10.10001, 101110.0101, 1110101.110, 1101101.111.

1.6. Aşağıdaki sayıları verilen tabandan istenen tabana çevirin:
(a) ondalık 225.225'i ikili, sekizli ve onaltılıya
(b) ikili 11010111.110'ı ondalık, sekizli ve onaltılıya
(c) sekizli 623.77'yi ondalık, ikili ve onaltılıya
(d) onaltılı 2AC5.D'yi ondalık, sekizli ve ikiliye

1.7. Aşağıdaki sayıları ondalık sisteme çevirin:

(a) $(1001001.011)_2$	(e) $(0.342)_6$
(b) $(12121)_3$	(f) $(50)_7$
(c) $(1032.2)_4$	(g) $(8.3)_9$
(d) $(4310)_5$	(h) $(198)_{12}$

1.8. Aşağıdaki ikili sayıların 1 ve 2'nin tümleyenlerini bulun: 1010101, 0111000, 0000001, 10000, 00000.

1.9. Aşağıdaki ondalık sayıların 9 ve 10 tümleyenlerini bulun: 13579, 09900, 90090, 10000, 00000.

1.10. $(935)_{11}$ sayısının 10"a göre tümleyenini bulun.

1.11. (1) 10'un tümleyenini ve (2) 9'un tümleyenini kullanarak aşağıdaki ondalık sayılarla çıkarma işlemini yapın. Cevapları doğrudan çıkarma işlemiyle kontrol edin.

(a) 5250 - 321 (c) 753 - 864

(b) 3570 - 2100 (d) 20 - 1000

1.12. (1) 2'nin tümleyenini ve (2) 1'in tümleyenini kullanarak aşağıdaki ikili sayılarda çıkarma işlemini yapın. Cevapları doğrudan çıkarma işlemiyle kontrol edin.

(a) 11010 - 1101 (c) 10010 - 10011

(b) 11010 - 10000 (d) 100 - 110000

1.13. 1.5. Bölümde verilen işlemin, (r-1) tümleyeniyle iki sayının çıkarılması için geçerli olduğunu kanıtlayın.

1.14. Ondalık hanelerde (a) 3,3,2,1 ve (b) 4,4,3, -2 ağırlıklı kodları için, her bir ondalık hanenin 9 tümleyeninin, 1'ler 0 ve 0'lar bir yapılarak elde edileceği olası bütün tabloları belirleyin.

1.15. Ondalık 8620 sayısını (a) BCD ile, (b) artık-3 koduyla, (c) 2,4,2,1 koduyla ve (d) ikili sayı olarak gösterin.

1.16. İkili bir kodda, ondalık on hanenin her birisini göstermek için on bit kullanılmaktadır. Her hane, dokuz adet 0 ve bir adet 1 ile kodlanmaktadır. Örneğin 6'nın kodu 0001000000'dır. Geri kalan ondalık hanelerin ikili kodunu bulun.

1.17. 5421 ağırlıklarını kullanarak 12 taban haneleri için ağırlıklı ikili kodu bulun.

1.18. Mesajın 8, 4, -2, -1 kodunda ondalık on haneden oluşması halinde üretilen tek-eşlik bitini bulun.

1.19. Tablo 1.4'te verilen yansımalı kodun, verilenin dışındaki diğer iki bileşimini (kombinasyonunu) bulun.

1.20. 6-tabanlı hanelerinin tamamını göstermek için öyle bir ikili kod bulun ki kodun bitlerindeki 1'leri 0, 0'ları da 1 yapığımız zaman 5'in tümleyenini elde edelim.

1.21. 52 oyun kağıdını düzenli bir şekilde ikili olarak kodlayın. Minimum sayıda bit kullanın.

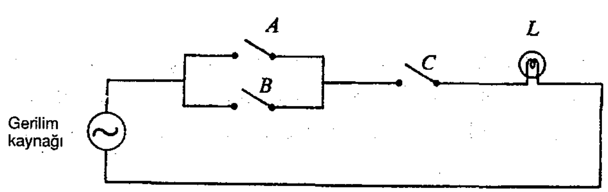
1.22. Tablo 1.5'te verilen yedi ASCII bitinden ve en anlamlı konumdaki bir eşlik bitinden oluşan sekiz bitli bir kodla adınızı soyadınızı yazın. Kelimeler arasına boşluk ve kısaltmalardan sonra nokta koyun.

1.23. (a) $(295)_{10}$ sayısını ikili formda, (b) 295 ondalık sayısını BCD formunda ve (c) XY5 karakterlerini EBCDIC formunda gösteren 24-hücreli bir kaydedicinin düzenlemesini (konfigürasyonunu) gösterin.

1.24. 12 hücreli bir kaydedicinin durumu 010110010111'dir. (a) BCD formunda üç ondalık haneyi (b) artık-3 kodunda üç ondalık haneyi, (c) 2,4,2,1 koduyla üç ondalık haneyi ve (d) Tablo 1.5'teki makine (iç) koduyla iki karakteri göstermesi halinde bu kaydedicinin içeriği nedir?

1.25. Toplanan iki ikili sayının ondalık eşdeğerlerinin 257 ve 1050 olması halinde Şekil 1.3'teki kaydedicilerin tümünün içeriğini bulun. (Kaydedicilerin 11 hücreli olduğunu varsayın.)

1.26. Aşağıdaki anahtarlama devresini ikili mantık sembolleriyle ifade edin.



1.27. Şekil 1.6'daki F ve G çıkışlarının sinyallerini gösterin (Şekil 1.7'dekine benzer bir şemayla). A, B, C ve D girişleri için keyfi ikili sinyaller kullanın.