

NEAR EAST UNIVERSITY

LOJİK DEVRELER

DERS NOTLARI

BMT 110

2016

İÇİNDEKİLER

1. SAYI SİSTEMLERİ
 2. SAYI SİSTEMLERİ ARASINDAKİ DÖNÜŞÜMLER
 3. SAYILARIN TÜMLENMESİ
 4. SAYILARIN KODLANMASI
 5. LOJİK KAPILAR, LOJİK KAPILARIN ELEKTRİKSEL KARŞILIKLARI
 6. BOOLE CEBRİ (BOOLEAN ALGEBRA)
 7. KOMBİNASYONLU (KARMAŞIK) LOJİK DEVRELER
 8. LOJİK İFADELERİN SADELEŞTİRİLMESİ
 9. LOJİK KAPILARIN ELEKTRONİK ELEMANLAR İLE ELDE EDİLMESİ
 10. TOPLAYICI DEVRELER (ADDERS)
 11. ÇIKARICI DEVRELER (SUBTRACTORS)
 12. ÇARPMA DEVRELERİ (MULTIPLIERS)
 13. SAYISAL KOMPARATÖR (COMPARATORS)
 14. KOD ÇÖZÜCÜLER (DECODERS)
 15. AZALTICI / VERİ SEÇİCİ DEVRELER (MULTIPLEXERS / DATA SELECTORS)
 16. ÇOĞULLAYICI / VERİ DAĞITICI DEVRELER (DEMULTIPLEXERS / DATA DISTRIBUTORS)
 17. KODLAYICILAR (ENCODERS)
 18. FLİP FLOP'LAR
 19. SAYICILAR (COUNTERS)
 20. DURUM TABLOSU ve DURUM DİYAGRAM YÖNTEMİ
-

1. SAYI SİSTEMLERİ

Günümüzde kullanılan sayı sistemi on tabanlı sayı sistemidir. Sayı sistemi ilk olarak Hintli matematikçiler tarafından M.S. 400'lü yıllarda kullanılmaya başlanmış ve M.S. 800'lü yıllarda ise Araplar tarafından kullanılmıştır. Bu sayı sistemi belli bir süre Arap sayı sistemi olarak adlandırılmıştır. Daha sonra Avrupa bilim adamları bu sayı sistemini önce sıfır rakamı olmadan kullanmış ve belli bir zaman sıfır rakamının kabulü ile çelişki yaşanmıştır. M.S. 1200 dolaylarında bugün var olan sayı sistemi kullanılmaya başlanmış ve onlu sayı sistemi adını almıştır. Onlu sayılarla mikroişlemcilerin ya da mikro bilgisayarların verileri işlemesi zor ve karmaşıktır. Sayısal elektronik devre düzeneklerine gereksinim hızlı bir şekilde artmış ve araştırmalar sayısal (ikili) veri işleme doğrultusunda gerçekleştirilmiştir. İkili sayılar iki rakamdan oluşur, dolayısıyla iki seviye ile işlem yapılır. Onlu sayılarda ise on farklı rakam olduğundan on seviyeli bilgi ve veri işleme gerçekleştirilmelidir. Bu durum devre düzenegi tasarımı ve gerçekleştirilmesi açısından ikili sayıların onlu sayılara göre elektronikte tercih edilmesinin açık göstergesidir.

1.1. 10 Tabanlı (Desimal) Sayılar

Günlük hayatta kullanılan sayılar 10 tabanlı sayı sistemine aittir. 10 sayısı sistemde kullanılan rakam sayısını verir.

$$a_i = \{0,1,2,3,4,5,6,7,8,9\}$$

Bir sayının ait olduğu sayı sisteminin anlaşılması için verilen sayının sağ alt köşesine kullanılan sayı sistemine ait taban sayısı yazılır. Onlu sayılarda her basamak o sayının mutlak değerini belirleyen bir ağırlığa sahiptir. Onlu sayılarda bu ağırlık 10'dur ve 10'un katları, ($10^0=1$, 10, 100, 1000, vs.) şeklinde yazılabilir. Verilen bir sayının toplam değeri, basamak ağırlıkları dikkate alınarak hesaplanabilir.

LSD (Least Signification Digit) (En az ağırlıklı basamak)

$$(1999)_{10} = X_3 \cdot 10^3 + X_2 \cdot 10^2 + X_1 \cdot 10^1 + X_0 \cdot 10^0 = 1 \cdot 10^3 + 9 \cdot 10^2 + 9 \cdot 10^1 + 9 \cdot 10^0 = 1000 + 900 + 90 + 9 = (1999)_{10}$$

MSD (Most Signification Digit) (En çok ağırlıklı basamak)

$$(568,25)_{10} = 5 \cdot 10^2 + 6 \cdot 10^1 + 8 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2} = 500 + 60 + 8 + 0,2 + 0,05 = (568,25)_{10}$$

1.2. 2 Tabanlı (Binary) Sayılar

Tabanı iki olan sayıların oluşturduğu bir sayı sistemidir. Sayı sistemleri içinde en sade sayı sistemidir.

$$a_i = \{0,1\}$$

Verilen iki tabanlı sayının on tabanlı karşılığını bulmak için her bit'i ağırlığına göre yazmak gerekir. Bit: İkili sayı sisteminde her bir basamağa verilen isimdir (Binary Digit). On tabanlı sayılarda olduğu gibi her bitin ikili düzende bir ağırlığı vardır. Bu ağırlıklar 2'nin katları

$(2^0=1,2,4,8,16,32,64,..vs.)$ şeklinde yazılır. İkili düzende en az ağırlıklı bit'in karşılığı LSB (Least Signification Bit) ve en çok ağırlıklı bit'in karşılığı MSB (Most Signification Bit)'dir.

Aşağıda verilen ikili düzendeki sayının değerinin belirlenmesi için ilgili bitlerin ağırlıkları hesaba katılmalıdır.

┌───┐ LSB (Least Signification Bit) (En az ağırlıklı bit)

$$(1101)_2 = X_3 \cdot 2^3 + X_2 \cdot 2^2 + X_1 \cdot 2^1 + X_0 \cdot 2^0 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = (13)_{10}$$

└───┘ MSB (Most Signification Bit) (En çok ağırlıklı bit)

$$(11101101)_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 128 + 64 + 32 + 0 + 8 + 4 + 0 + 1 = (237)_{10}$$

$$(1011001011)_2 =$$

$$(1001,1011)_2 =$$

4	Bit	= 1 Nibble		
8	Bit	= 1 Byte	= 1B	
16	Bit	= 2 Byte	= 1 Word	= 1W
32	Bit	= 4 Byte	= 2 Word	= 1 Long Word
1024	Byte	= 1 Kilo Byte	= 1KB	
1024	KB	= 1 Mega Byte	= 1MB	
1024	MB	= 1 Giga Byte	= 1GB	
1024	GB	= 1 Tera Byte	= 1TB	

1.3. Sekizli (Oktal) Sayılar

Tabanı sekiz olan sayıların oluşturduğu bir sayı sistemidir. Bu sayı sisteminin rakam kümesi;

$$a_i = \{0,1,2,3,4,5,6,7\}$$

Verilen sekiz tabanlı sayının on tabanlı sayı karşılığını bulmak için her basamağı ağırlığına göre yazmak gerekir. Sekiz tabanlı sayılarda ağırlıklar 8' in katları ($8^0=1,8,64,512,4096,..vs.$) şeklinde yazılır.

$$(276)_8 = X_2 \cdot 8^2 + X_1 \cdot 8^1 + X_0 \cdot 8^0 = 2 \cdot 8^2 + 7 \cdot 8^1 + 6 \cdot 8^0 = 128 + 56 + 6 = (190)_{10}$$

$$(42105)_8 = 4 \cdot 8^4 + 2 \cdot 8^3 + 1 \cdot 8^2 + 0 \cdot 8^1 + 5 \cdot 8^0 = 16384 + 1024 + 64 + 0 + 5 = (17477)_{10}$$

$$(7652)_8 =$$

$$(243,76)_8 =$$

1.4. Onaltılı (Heksadesimal) Sayılar

Tabanı on altı olan sayıların oluşturduğu bir sayı sistemidir. Heksadesimal sayı sisteminde 10 tabanlı sayılarda olan rakamlara ilave olarak altı rakam daha vardır. (A=10, B=11, C=12, D=13, E=14 ve F=15)

$$a_i = \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$$

Verilen 16 tabanlı sayının on tabanlı sayı karşılığını bulmak için her basamağı ağırlığına göre yazmak gerekir. 16 tabanlı sayılarda ağırlıklar 16' nın katları ($16^0=1, 16^1=16, 16^2=256, 16^3=4096, 16^4=65536, \dots$) şeklinde yazılır.

$$(10BA)_{16} = X_3 \cdot 16^3 + X_2 \cdot 16^2 + X_1 \cdot 16^1 + X_0 \cdot 16^0 = 1 \cdot 16^3 + 0 \cdot 16^2 + B \cdot 16^1 + A \cdot 16^0 = 4096 + 0 + 176 + 10 = (4282)_{10}$$

$$(CE2)_{16} = C \cdot 16^2 + E \cdot 16^1 + 2 \cdot 16^0 = 3072 + 224 + 2 = (3298)_{10}$$

$$(2A4F)_{16} =$$

$$(FA9,12)_{16} =$$

2. SAYI SİSTEMLERİ ARASINDAKİ DÖNÜŞÜMLER

2.1. On Tabanlı Sayı Sisteminden İki Tabanlı Sayı Sistemine Dönüşüm

Verilen sayı tam sayı ise ikiye bölünerek kalanlar kayıt edilir. Bu durumda kalan kısım ya 0 ya da 1'dir. Bölme işlemi bölümde 0 veya 1 görülünceye kadar devam ettirilir.

Onlu düzende verilen sayı reel sayı olabilir. Yani hem tam kısımdan ve hem de kesirli kısımdan oluşuyorsa veya sadece kesirli bir sayı ise bu durumda tam kısmı ayrı ve yukarıda ifade edildiği gibi yapılır. Kesirli kısım ise ayrı olarak ve sürekli 2 ile çarpılır ve tam kısımları kayıt edilir. Sayı, tam sayı (kesirsiz) olana kadar çarpma işlemine devam edilir. Tam kısımlar uzun süre sıfır çıkıyor burada işlem durdurulur. Diğer bir ifadeyle beş veya altı adım sonucunda tam kısımda hala sıfır varsa işlem bitirilir.

$$(248)_{10} = (..?)_2 = (11111000)_2$$

$$(0,625)_{10} = (..?)_2 = (0,101)_2$$

<u>Bölüm</u>	<u>Kalan</u>	<u>LSB</u>
248 / 2 = 124	0	↑
124 / 2 = 62	0	
62 / 2 = 31	0	
31 / 2 = 15	1	
15 / 2 = 7	1	
7 / 2 = 3	1	
3 / 2 = 1	1	

MSB

<u>Çarpım</u>	<u>Tam Kısım</u>	
0,625 * 2 = 1,250	1	↓
0,250 * 2 = 0,500	0	
0,5 * 2 = 1,000	1	

MSB
LSB

$$(9,8125)_{10}=(\dots\dots)_2$$

$$(38,375)_{10}=(\dots\dots)_2$$

2.2. On Tabanlı Sayı Sisteminden Sekiz veya On altı Tabanlı Sayı Sistemine Dönüşüm

On tabanlı sayıdan, sekiz tabanlı sayı veya on altı tabanlı sayıya geçişte onlu sayıdan ikiliye geçişte yapıldığı gibi tam kısım için taban sayısına bölme, kesirli kısım için taban sayısı ile çarpma işlemi yapılır.

$$(247)_{10}=(\dots\dots)_8$$

$$(0,513)_{10}=(\dots\dots)_8$$

$$(1367)_{10}=(\dots\dots)_{16}$$

2.2.1. İki Tabanlı Sayı Sisteminden Sekiz Tabanlı Sayı Sistemine Dönüşüm

Tabanı sekiz olan sayıların oluşturduğu sayı kümesindeki tüm sayılar $\{0,1,2,3,5,6,7\}$, iki tabanlı sayı sisteminde **üç bit** ile ifade edilmektedir $\{000, 001, 010, 011, 100, 101, 110, 111\}$. İki tabanlı sayı sisteminde verilen sayı virgülden sağa ve sola doğru olmak üzere üçer bitlik parçalara ayrılır. Son parçalar üç bit olmuyorsa sayısal değeri bozmayacak şekilde sıfır eklenir ve üç bite tamamlanır. Bu üç bitlik grupların sekiz tabanlı sayı sisteminde karşılığı yazılarak verilen sayı iki tabanlı sayı sisteminden sekiz tabanlı sayı sistemine çevrilmiş olur.

$$(1111110,01011)_2= \underline{00}(1 \ \underline{111} \ \underline{110}, \ \underline{010} \ \underline{11})\underline{0}_2=(176,26)_8$$

$$(101000111,00100011110)_2=(\dots\dots)_8$$

$$(1110011100,10011011)_2=(\dots\dots)_8$$

2.2.2. Sekizli Tabanlı Sayı Sisteminden İki Tabanlı Sayı Sistemine Dönüşüm

Daha önce anlatılan, iki tabanlı sayı sisteminden sekiz tabanlı sayı sistemine dönüşüm işleminin tam tersi yapılır. Verilen sekiz tabanlı sayıya ait rakamların ikili düzende üç bit olarak karşılığı yazılıp sayısal değer bozulmayacak şekilde yan yana sıralanırsa, dönüşüm işlemi doğrudan gerçekleşmiş olmaktadır.

$$(176,26)_8=(001 \ 111 \ 110, \ 010 \ 110)_2=(1111110,01011)_2$$

$$(451,3045)_8=(\dots\dots\dots)_2$$

$$(71023,76)_8=(\dots\dots\dots)_2$$

2.2.3. İki Tabanlı Sayı Sisteminden On Altı Tabanlı Sayı Sistemine Dönüşüm

Tabanı on altı olan sayıların oluşturduğu sayı kümesindeki tüm sayılar {0,1,2,3,5,6,7,8,9,A,B,C,D,E,F,}, iki tabanlı sayı sisteminde **dört bit** ile ifade edilmektedir {0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111}. İki tabanlı sayı sisteminde verilen sayı virgülden sağa ve sola doğru olmak üzere dörder bitlik parçalara ayrılır. Son parçalar dört bit olmuyorsa sayısal değeri bozmayacak şekilde sıfır eklenir ve dört bite tamamlanır. Bu dört bitlik grupların on altı tabanlı sayı sisteminde karşılığı yazılarak verilen sayı iki tabanlı sayı sisteminden son altı tabanlı sayı sistemine çevrilmiş olur.

$$(1111110,01011)_2 = \underline{0(111)} \underline{1110}, \underline{0101} \underline{1} \underline{000}_2 = (7E,58)_{16}$$

$$(1011000100,1010011)_2 = (\dots\dots)_{16}$$

$$(1001001111001,101001110)_2 = (\dots\dots)_{16}$$

2.2.4. Onaltı Tabanlı Sayı Sisteminden İki Tabanlı Sayı Sistemine Dönüşüm

Daha önce anlatılan, iki tabanlı sayı sisteminden on altı tabanlı sayı sistemine dönüşüm işleminin tam tersi yapılır. Verilen on altı tabanlı sayıya ait rakamların ikili düzende dört bit olarak karşılığı yazılıp sayısal değer bozulmayacak şekilde yan yana sıralanırsa, dönüşüm işlemi doğrudan gerçekleşmiş olmaktadır.

$$(34EA,B28)_{16} = (0011 \ 0100 \ 1110 \ 1010, \ 1011 \ 0010 \ 1000)_2 = (11010011101010,101100101)_2$$

$$(1C94,2F9)_{16} = (\dots\dots)_2$$

$$(A53B,762)_{16} = (\dots\dots)_2$$

3. SAYILARIN TÜMLENMESİ

Sayı sistemlerinde doğrudan çıkarma yapılacağı gibi tümleyen yöntemiyle de çıkarma yapılabilir. Tümleyen yöntemiyle çıkarma işlemi aslında bir toplama işlemidir. Bu işlemde bir üst basamaktan borç alınmaz. Her sayı sistemine ilişkin iki adet tümleyen bulunabilir. Bunlar; t sayı sisteminin tabanını göstermek üzere;

- ✓ t tabanına göre tümleyen
- ✓ (t-1) tabanına göre tümleyen olarak gösterilebilir.

3.1. t Tabanına Göre Tümleme

$$T_t = t^n - N$$

T_t : t tabanında verilen N sayısının yine t tabanında tümleyeni,

t: Taban yani sayı sisteminin adı,

N: Tümleyeni alınacak t tabanındaki sayı,

n: Sayının digit veya bit adedi (basamak sayısı),

Örnek: $(1963)_{10}$ Verilen sayının on tabanındaki tümleyenini bulunuz? $T_{10} = ?$

$$T_{10} = 10^4 - 1963 = 10000 - 1963 = 8037$$

Örnek: $(0,6914)_{10}$ Verilen sayının on tabanındaki tümleyenini bulunuz? $T_{10} = ?$

$$T_{10} = t^0 - N = 1 - N = 1 - 0,6914 = (0,3086)_{10}$$

3.2. (t-1) Tabanına Göre Tümleme

$$T_{t-1} = t^n - t^m - N$$

T_{t-1} : t tabanında verilen N sayısının yine (t-1) tabanında tümleyeni,

t: Taban yani sayı sisteminin adı,

N: Tümleyeni alınacak t tabanındaki sayı,

n: Tam kısmın basamak sayısı,

m: Kesirli kısmın basamak sayısı,

Örnek: $(31652)_{10}$ Verilen sayının dokuz tabanındaki tümleyenini bulunuz? $T_9 = ?$

$$T_9 = (10^5 - 1 - 31652) = (68347)_{10}$$

$$T_{10} = t^n - N = 100000 - 31652 = (68348)_{10}$$

$$T_{t-1} + 1 = T_t$$

Verilen sayı ile (t-1) tabanına göre elde edilen sayı toplandığında bütün basamaklarda (t-1) rakamları oluşur. Yukarıdaki örneği incelediğimizde, 9 tabanına göre tümleme işlemi yapıldığında elde edilen sayı ile verilen sayı toplandığında 9'lu rakamlar oluşur, yani toplama işlemi sonucunda her bir basamağın sayısal değerinin 9 olduğu görülmektedir.

$$31652 + 68347 = 99999$$

O halde bir sayının (t-1) tabanına göre tümleyeni bulmak için pratik bir yol geliştirilebilir. Verilen sayının (t-1) tabanına göre tümleyenini bulmak için her basamaktaki toplama işlemi sonucunda (t-1) tabanına ait rakamı verecek sayı bulunur, yani (t-1) tümleyen sayısı alınır. Bu sayı (t-1)'e göre tümleyendir. (t-1) tabanına göre tümleyene 1 eklendiğinde t tabanına göre tümleyen bulunur.

$$1963 + 8036 = 9999$$

3.3. İki Tabanlı Sayı Sisteminde Dört İşlem

<u>A + B = 2⁰ 2¹(Elde)</u>	<u>A * B = Sonuç</u>	<u>A - B = 2⁰ 2¹(Borç)</u>	<u>A / B = Sonuç</u>
0 + 0 = 0 0	0 * 0 = 0	0 - 0 = 0 0	0 / 0 = 0
0 + 1 = 1 0	0 * 1 = 0	0 - 1 = 1 1	0 / 1 = 0
1 + 0 = 1 0	1 * 0 = 0	1 - 0 = 1 0	1 / 0 = 0
1 + 1 = 0 1	1 * 1 = 1	1 - 1 = 0 0	1 / 1 = 1

Yukarıdaki bir bitlik toplama işlemindeki ilk üç seçenekte toplam sonucu bir bit iken son seçenek yani birinci ve ikinci bileşenin lojik "1" olduğu seçenekte ise işlem sonucu bir bit değil, iki bit'tir. Toplam sonucunu ifade eden bitin yanında bir ikinci bit ortaya çıkmıştır. Bu bit elde (carry) bit'idir ve temsil ettiği bit ağırlığı ise 2¹'dir.

Çarpma ve bölme işlemleri on tabanlı sayı sisteminde olduğu gibi yapılır.

Bir bitlik çıkarma işlemi yapıldığında küçük sayıdan büyük sayıyı çıkarmada işlem sonucu bir bit ile ifade edilemez. Çünkü üst basamaktan bir borç alınır ve bu borcun da bir bit ile ifade edilmesi gerekir. Dolayısıyla bu durum bir borç biti ile isimlendirilir. Borç bitinin bir olması işlem yapılan basamakta sonuç eksi olduğu anlaşılır. Daha doğrusu üst basamaktan bir borç alındığı anlamına gelmektedir. Büyük sayılardan küçük sayıları çıkarmak kolay olmasına rağmen küçük sayılardan büyük sayıları çıkarmak zor olmaktadır.

İki tabanlı sayı sisteminde n bitlik toplama işlemi on tabanlı sayı sisteminde olduğu gibi yapılır. Elde oluşursa elde bir sonraki bit'e (basamağa) eklenir.

$$(101101 + 10011)_2 = (..?)_2$$

101101	(45) ₁₀
+ 10011	(19) ₁₀

1000000	(64) ₁₀

$$(1100101 + 101110)_2 = (.....)_2$$

$$(10111000 + 10010)_2 = (.....)_2$$

3.4. t Tabanına Göre Tümlleme ile Çıkarma İşlemi

Doğrudan çıkarma işlemi yapılırken; büyük sayıdan küçük bir sayı çıkarıldığında sorun oluşmamaktadır. Fakat küçük bir sayıdan büyük bir sayıyı çıkarırken borç biti oluşmakta ve gerçek sonuç sayısal değer olarak çıkışa yansımamaktadır. Dolayısıyla aşağıdaki örnekte görüldüğü üzere 26' dan 29 çıkarıldığında sonucun -3' e çıkması beklenirken 61 çıkmıştır. Borç biti oluştuğundan dolayı da sonucun bit sayısı, sayıların (çıkarılan ve çıkan sayılar) bit sayılarından büyük olmuştur. Burada çıkan hatalı sonucu düzeltmek için sonucun t tabanına göre tümlemesini almak gerekmektedir. Sonucun t tabanına göre tümlemesi alındığında sonuç 3 çıkmaktadır. Sayının tümlemesi alındığından dolayı da sayı negatiftir.

$$(11101)_2 - (11010)_2 = (.....)_2$$

$$(11010)_2 - (11101)_2 = (.....)_2$$

$$\begin{array}{r} 11101 \ (29)_{10} \\ - 11010 \ (26)_{10} \\ \hline 00011 \ (3)_{10} \end{array}$$

$$\begin{array}{r} 11010 \ (26)_{10} \quad (4 \text{ bit}) \\ - 11101 \ (29)_{10} \quad (4 \text{ bit}) \\ \hline 111101 \ (61)_{10} \rightarrow ? \quad (5 \text{ bit}) \\ 000010 \ (t-1) \text{ tabanına göre tümleyen} \\ 000011 \ t \text{ tabanına göre tümleyen } (-) \end{array}$$

Çıkarma işlemi doğrudan değil de t tabanına göre tümleme alınarak da yapılabilir.

$$\begin{array}{r} M_t \text{ (Çıkarılan sayı)} \\ - N_t \text{ (Çıkan sayı)} \\ \hline \text{Sonuç} \end{array} \quad \rightarrow \quad \begin{array}{r} M_t \text{ (Çıkarılan sayı)} \\ + N_{Tt} \text{ (Çıkan sayının t tabanına göre tümleyeni)} \\ \hline \text{Sonuç} \end{array}$$

Bu yöntemle doğrudan çıkarma işlemi yapmak yerine çıkarılan sayı ile çıkan sayının t tabanına göre tümleyeni toplanır. Toplama işlemi sonucunda;

- ✓ Elde oluşmuşsa (veya toplama sonucunun bit sayısı M, N sayılarının bit sayısından büyük ise) ; Sonuç pozitifdir. Elde silinerek gerçek sonuca ulaşılır.
- ✓ Elde oluşmamışsa (veya toplama sonucunun bit sayısı M, N sayılarının bit sayısından büyük değilse) ; Sonuç negatiftir. Sonucun t tabanına göre tümleyeni alınarak gerçek sonuca ulaşılır.

$$\begin{array}{r} 1101 \ (13)_{10} \\ - 0101 \ (5)_{10} \\ \hline \end{array} \rightarrow \begin{array}{r} 1101 \\ + 1011 \\ \hline 11000 \text{ (Elde silindi)} \\ 1000 \ (8)_{10} \end{array} \quad \begin{array}{r} 1001 \ (9)_{10} \\ - 1011 \ (11)_{10} \\ \hline \end{array} \rightarrow \begin{array}{r} 1001 \\ + 0101 \\ \hline 1110 \text{ (Elde Yok)} \\ 0010 \ -(2)_{10} \end{array}$$

$$(10011)_2 - (1001)_2 = (\dots)_2$$

$$(10001)_2 - (11011)_2 = (\dots)_2$$

3.5. (t-1) Tabanına Göre Tümleme ile Çıkarma İşlemi

$$\begin{array}{r} M_t \text{ (Çıkarılan sayı)} \\ - N_t \text{ (Çıkan sayı)} \\ \hline \text{Sonuç} \end{array} \quad \rightarrow \quad \begin{array}{r} M_t \text{ (Çıkarılan sayı)} \\ + N_{T(t-1)} \text{ (Çıkan sayının (t-1) tabanına göre tümleyeni)} \\ \hline \text{Sonuç} \end{array}$$

Doğrudan çıkarma işlemi yapmak yerine çıkan sayının (t-1)'e göre tümleyeni alınıp çıkarılan sayı ile toplanır. Bu ara toplama işlemi sonucunda;

- ✓ Elde biti=1 (Elde var) ise; Sonuç pozitifdir. Elde silinir ve sonuç, 1 ile toplanarak gerçek sonuca ulaşılır.
- ✓ Elde biti=0 (Elde yok) ise; Sonuç negatiftir. Sonucun (t-1) tabanına göre tümleyeni alınarak gerçek sonuca ulaşılır.

$$\begin{array}{r}
 57639 \\
 - 65423 \\
 \hline
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 57639 \\
 + 34576 \\
 \hline
 92215 \text{ (Elde=0, Sonuç negatif, (t-1) tabanına göre tümleyen alınır.)} \\
 07784 \text{ (-7784)}
 \end{array}$$

$$\begin{array}{r}
 65423 \\
 - 57639 \\
 \hline
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 65423 \\
 + 42360 \\
 \hline
 107783 \text{ (Elde=1, Sonuç pozitif, Elde silinir ve sonuç 1 ile toplanır.)} \\
 07784 \text{ (7784)}
 \end{array}$$

$$(7654)_8 - (5347)_8 = (\dots)_8$$

$$(3216)_8 - (6241)_8 = (\dots)_8$$

$$(1010)_2 - (1110)_2 = (\dots)_2$$

$$(10111)_2 - (11001)_2 = (\dots)_2$$

4. SAYILARIN KODLANMASI

Hayatımızda kullanılan onlu sistemdeki sayılar, özel karakter ve harfler, dijital sistemlerde işlenebilmesi için ikili sayı sistemine dönüştürülmesi gerekir. Bilgileri dijital sistemlerde kullanmak ve üzerinde işlem yapmak için yapılan dönüştürme işlemine kodlama adı verilir. Başka bir deyişle kodlama iki küme arasında karşılığı tanımlanmış temel kurallar dizini olarak da tanımlanır. Kodlar kendi arasında dijital ve alfanumerik olmak üzere iki temel türde incelenebilir.

Kodlama Çeşitleri

- ✓ Dijital (sayısal) Kodlama
 - i) BCD (Binary Coded Decimal - İkili Kodlanmış Onlu Sayı Kodu) veya 8421 Kodlama
 - ii) Gray Kodu
 - iii) Artı-3 (Excess-3) Kodu
 - iv) 5'te 2 Kodu
 - v) Eşlik (Parity) Kodu
 - vi) Aiken Kodu
 - vii) Bar (Çubuk) Kodu (Bar-Code)
- ✓ Alfanumerik Kodlama
 - i) ASCII Kodu

4.1. BCD Kodlama

Binary Coded Decimal olarak bilinen bu kod on tabanlı sayıları ikili düzende kodlanması demektir. Bu kodlamada, on tabanlı sayının her bir basamağı (digit) için dörder bitlik ikili bir ifade yazılarak kodlama yapılır.

On Tabanlı Sayı	BCD Kodlama (8421)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Yukarıdaki tablodan da görüldüğü gibi 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 her bir basamağı 4 bitlik iki tabanlı sayı grupları şeklinde yazılır. Yazılan gruplar bir araya getirilince BCD kodlu sayı elde edilir.

$$(145)_{10} = (0001\ 0100\ 0101)_{BCD}$$

$$(5698)_{10} = (\dots\dots\dots)_{BCD}$$

$$(73204)_{10} = (\dots\dots\dots)_{BCD}$$

$$(1001011010000100)_{BCD} = (1001\ 0110\ 1000\ 0100)_{BCD} = (\dots\dots\dots)_{10}$$

$$(0111001010010110)_{BCD} = (\dots\dots\dots)_{BCD} = (\dots\dots\dots)_{10}$$

$$(100001110011010110010011)_{BCD} = (\dots\dots\dots)_{BCD} = (\dots\dots\dots)_{10}$$

4.1.1. BCD Kod ile Toplama İşlemi

BCD-Kodlama aritmetik işlemlerden toplama işleminde kullanılır. Onlu düzendeki toplama işlemi sonucunu elde etmek için karşımıza iki durum çıkar.

- ✓ Toplam sonucu < 10 ise ; toplam sonucu doğrudur ve aynısı alınır.

$$\begin{array}{r}
 0011 \\
 + 0100 \\
 \hline
 0111
 \end{array}
 \qquad
 \begin{array}{r}
 3 \\
 + 4 \\
 \hline
 7
 \end{array}
 \qquad
 \begin{array}{r}
 0111\ 0110 \\
 + 0001\ 0010 \\
 \hline
 1000\ 1000
 \end{array}
 \qquad
 \begin{array}{r}
 76 \\
 + 12 \\
 \hline
 88
 \end{array}$$

- ✓ Toplam sonucu > 9 ise ilgili basamaktaki işlem sonucu ortaya çıkan sayısal değere basamak düzeltme sayısı olarak 6 eklenir. Böylece ikili düzeni kullanarak onlu düzendeki sayılar arasında aritmetik toplama işlemi gerçekleşmiş olur.

$$\begin{array}{r}
1001 \\
+ 0110 \\
\hline
1111 \\
+ 0110 \\
\hline
10101 = (10110) = (1\ 0101) \\
= (0001\ 0101)_{BCD}
\end{array}
\qquad
\begin{array}{r}
9 \\
+ 6 \\
\hline
15
\end{array}
\qquad
\begin{array}{r}
0101\ 0100 \\
+ 0010\ 0111 \\
\hline
0111\ 1011 \\
+ 1\ 0110 \\
\hline
1000\ 10001 = (1000\ 0001)_{BCD}
\end{array}
\qquad
\begin{array}{r}
54 \\
+ 27 \\
\hline
81
\end{array}$$

$$132_{10} + 47_{10} = (\dots\dots\dots)_{BCD}$$

$$257_{10} + 192_{10} = (\dots\dots\dots)_{BCD}$$

4.2. GRAY Kodu

Dijital elektronik ve bilgisayar giriş çıkış işlemlerinde kullanılan Gray kodlama yöntemi, en az değişim kodlamadır. Bunun nedeni bir sayıdan diğerine geçerken yalnızca bir bitin konum değiştirmesidir. Örneğin; ikili (binary) kodlamada $(3)_{10} = (0011)_2$ değerinden $(4)_{10} = (0100)_2$ değerine geçerken üç bitin değeri aynı anda değişirken, gray kodlamada yalnızca bir bitin değeri değişir.

İkili sayı sistemine kolayca çevrilmesi avantajıdır. En çok tercih edilen uygulama alanı olarak geri beslemeli sistemlerde konum denetimidir. Kodlama sıralamasında bir önceki sayısal kod ile bir sonraki sayısal kod arasında sadece tek bir bit’de farklılık olmasından dolayı konum belirleme işlemlerinde tercih edilir. Gray kodlamada basamakların sayı değeri yoktur.

On Tabanlı Sayı	İki Tabanlı Sayı	Gray Kodu
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

4.2.1. İkili Sayılardan Gray Koduna Geçiş

İkili sayı sisteminden gray koduna geçerken en ağırlıklı bit (MSB) hangi sayı olursa olsun doğrudan yazılır. Daha sonra her bit solundaki bit ile (bir üst bit ile bir alt bit) ardışık toplanarak bir sonraki basamağa yazılır. İşlem esnasında oluşan eldeler atılır. Bu işleme en az ağırlıklı bite kadar (LSB) devam edilir.

$$(1\ 1\ 1\ 1)_2 = (1\ 0\ 0\ 0)_{gray}$$

$$(1000)_2 = (\dots\dots\dots)_{gray}$$

$$(0111)_2 = (\dots\dots\dots)_{\text{gray}}$$

4.2.2. Gray Kodundan İkili Sayı Sistemine Geçiş

Gray kodlu bir sayıyı ikili sistemdeki sayı şekline dönüştürmek için, en ağırlıklı bit (MSB) doğrudan aşağı yazılır. Aşağı yazılan sayı ile bir sonraki basamakta bulunan sayı toplanarak önceki yazılan sayının yanına yazılır. Bu işleme en düşük değerlikli bite kadar devam edilir. İşlem esnasında oluşan eldeler atılır.

$$\begin{array}{cccc} (1 & 1 & 1 & 0)_{\text{gray}} \\ \downarrow & \nearrow & \nearrow & \nearrow \\ (1 & 0 & 1 & 1)_2 \end{array}$$

$$(1000)_{\text{gray}} = (\dots\dots\dots)_2$$

$$(0111)_{\text{gray}} = (\dots\dots\dots)_2$$

$$(1101)_{\text{gray}} = (\dots\dots\dots)_{10}$$

4.3. Artı-3 (Excess-3) Kodu

On tabanlı sayıya 3 eklenip BDC koda çevrilmesiyle, Artı-3 kodlama elde edilir. Aritmetik işlemlerde işlem kolaylığı sağladığı için kullanılır. İki veya daha fazla basamaklı sayılar +3 koduna dönüştürülürken her basamak için aynı işlem yapılır.

$$(7)_{10} = (1010)_{+3}$$

$$(24)_{10} = (0101 \ 0111)_{+3}$$

$$(138)_{10} = (0100 \ 0110 \ 1011)_{+3}$$

$$(4296)_{10} = (\dots\dots\dots)_{+3}$$

$$(25792)_{10} = (\dots\dots\dots)_{+3}$$

+3 kodunda on tabanlı sayıya dönüşüm için yukarıda anlatılan işlemin tersi yapılır.

$$(1001011010000100)_{+3} = (1001 \ 0110 \ 1000 \ 0100)_{+3} = (9 \ 6 \ 8 \ 4) = (6351)_{10}$$

$$(0111001110010110)_{+3} = (\dots\dots \dots\dots \dots\dots)_{+3} = (\dots\dots\dots) = (\dots\dots\dots)_{10}$$

$$(100001110011010110010011)_{+3} = (\dots\dots \dots\dots \dots\dots \dots\dots)_{+3} = (\dots\dots) = (\dots\dots\dots)_{10}$$



4.8. ASCII Kodu

ASCII (American Standard Code for Information Interchange) (Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi). Latin alfabesi üzerine kurulu 7 bitlik bir karakter setidir. İlk kez 1963 yılında ANSI tarafından standart olarak sunulmuştur.

ASCII'de 33 tane basılmayan kontrol karakteri ve 95 tane basılan karakter bulunur. Kontrol karakterleri metnin akışını kontrol eden, ekranda çıkmayan karakterlerdir. Basılan karakterler ise ekranda görünen, okuduğumuz metni oluşturan karakterlerdir. ASCII'nin basılan karakterleri aşağıda belirtilmiştir. Bütün büyük ve küçük harfler, rakamlar, noktalama işaretleri ve kontrol karakterleri bu kodlamada tanımlanmıştır.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Şekil. ASCII Kodları

128	Ç	144	É	160	á	176	☒	192	Ł	208	⌘	224	α	240	≡
129	ü	145	æ	161	í	177	☓	193	ł	209	⌞	225	β	241	±
130	é	146	Æ	162	ó	178	☑	194	Ł	210	⌗	226	Γ	242	≥
131	â	147	ô	163	ú	179		195	ł	211	⌘	227	π	243	≤
132	ã	148	ö	164	ñ	180	†	196	—	212	⌡	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	‡	197	+	213	⌢	229	σ	245	∫
134	â	150	û	166	•	182	‡	198	†	214	⌣	230	μ	246	+
135	ç	151	ù	167	◦	183	¶	199	¶	215	⌤	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	¶	200	⌢	216	⌥	232	Φ	248	◦
137	ë	153	Ö	169	ƒ	185	¶	201	⌣	217	⌦	233	⊙	249	•
138	è	154	Û	170	¬	186	¶	202	⌤	218	⌧	234	Ω	250	•
139	ì	155	◊	171	½	187	¶	203	⌞	219	■	235	δ	251	√
140	î	156	£	172	¾	188	¶	204	⌗	220	■	236	∞	252	∞
141	ï	157	¥	173	ı	189	¶	205	=	221	■	237	φ	253	z
142	Ä	158	€	174	«	190	¶	206	‡	222	■	238	ε	254	■
143	Å	159	ƒ	175	»	191	¶	207	⌤	223	■	239	∩	255	

Şekil. Genişletilmiş ASCII Kodları

5. LOJİK KAPILAR (LOGIC GATES)

Dijital (Sayısal) devrelerin tasarımında kullanılan temel devre elemanlarına Lojik kapılar adı verilmektedir. Her lojik kapının bir çıkışı, bir veya birden fazla girişi vardır. Lojik kapıların girişlerine, "Lojik 1" veya "Lojik 0" adı verilen seviyeler uygulanabilir. Girişlerinin durumuna göre lojik kapıların çıkışından "Lojik 1" veya "Lojik 0" gerilim seviyeleri gözlemlenir. "Lojik 0" seviye, 0 (sıfır) volt gerilimi temsil etmekte iken (Lojik 0 \equiv 0V). "Lojik 1" seviye ise, +5 volt gerilimi temsil etmektedir (Lojik 1 \equiv 5V).

Lojik kapıların girişlerine giriş seviyesinin uygulanması ("Lojik 1" ve "Lojik 0") için iki konumlu devre elemanı olan anahtar kullanılır. Anahtarın iki konumu vardır (anahtar açık ve anahtar kapalı). Genelde, anahtar açık durumu "Lojik 0", anahtar kapalı durumu ise "Lojik 1" olarak temsil edilmektedir.

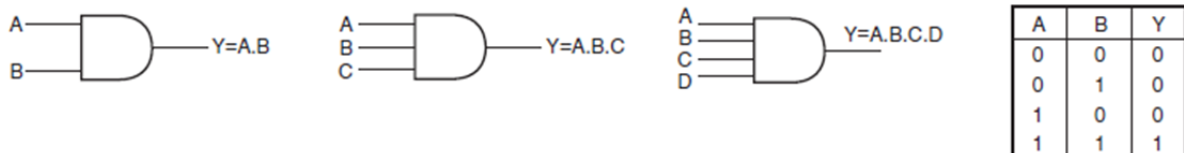


VE (And), VEYA (Or), DEĞİL (Not) olmak üzere üç temel lojik kapı vardır. Ayrıca bu üç temel kapıdan türetilmiş (VE-DEĞİL (Nand), VEYA-DEĞİL (Nor), ÖZEL VEYA (Ex-Or) ve ÖZEL VEYA-DEĞİL (Ex-Nor)) dört kapı ile birlikte toplamda 7 (yedi) adet lojik kapı bulunmaktadır.

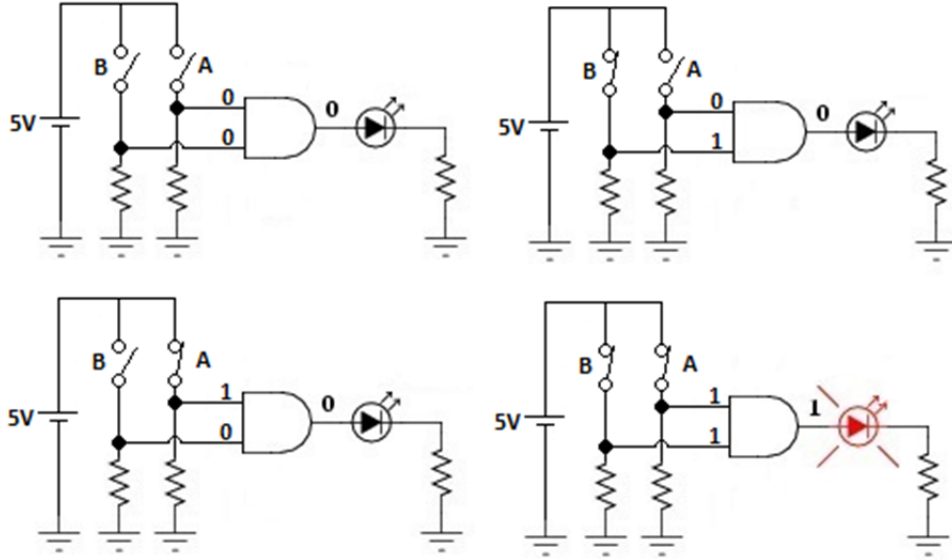
Bir Lojik kapının girişlerinin durumlarına bağlı olarak çıkışının ne olacağını gösteren tabloya doğruluk tablosu (truth table) adı verilir. Doğruluk tablosu n girişli bir lojik kapının, olası tüm giriş durumuna karşılık, lojik kapının çıkışının hangi değeri alacağını gösterir. n girişli bir lojik kapının, girişlerinin alabileceği 2^n adet durum vardır ve her durum doğruluk tablosunda bulunmalıdır.

5.1. VE KAPISI (AND GATE)

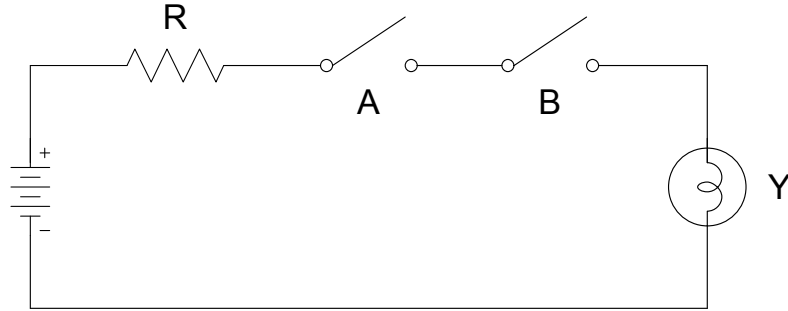
Aşağıda 2, 3 ve 4 girişli VE kapılarının sembolleri, lojik ifadeleri ve 2 girişli VE kapısına ait doğruluk tablosu görülmektedir. Doğruluk tablosu incelendiğinde, 2 girişli VE kapısının $2^2=4$ durumu olduğu görülmektedir. Doğruluk tablosundan da görüldüğü üzere VE kapısının her iki girişi "lojik 1" durumda iken çıkışı "lojik 1" olmaktadır.



2 girişli VE kapısına ait doğruluk tablosunun nasıl oluşturulduğu aşağıdaki şekillerde ayrıntılı olarak görülmektedir.

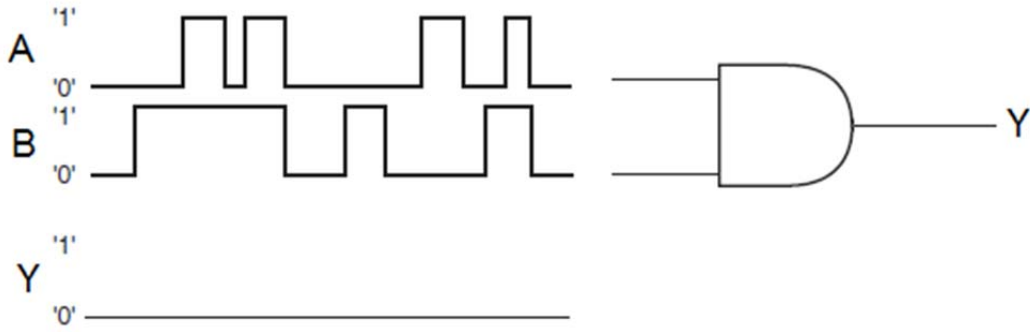


Aşağıda VE kapısının elektriksel eşdeğer devresi verilmiştir. VE kapısı matematiksel anlamda bir bitlik çarpma işlemini ifade ederken elektriksel anlamda anahtarların seri bağlanmasını ifade etmektedir. Aşağıdaki devrede, Y lambasının yanıyor olması “lojik 1”, Y lambasının sönmük olması ise “lojik 0” anlamına gelmektedir. Şekilden de görüleceği üzere Y lambasının yanması için her iki anahtarın (A ve B) kapalı olması gerekmektedir.

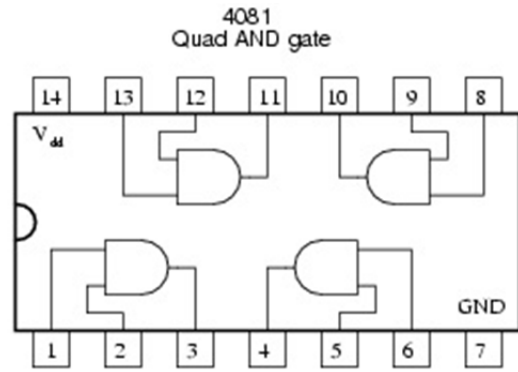
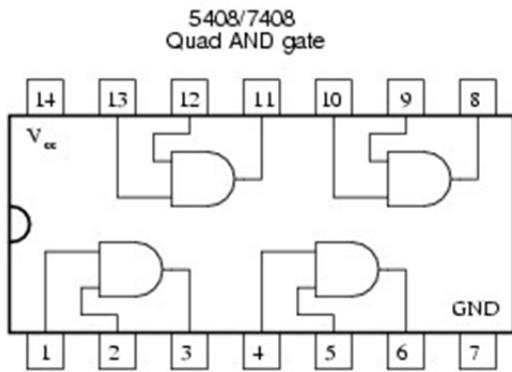


Soru: Sadece iki girişli VE kapısı kullanarak üç girişli VE kapısı nasıl elde ederiz?

Soru: iki girişli VE kapısının girişlerine A ve B sinyalleri uygulanmaktadır. VE kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.

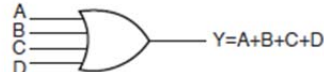
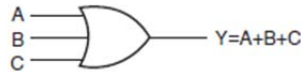
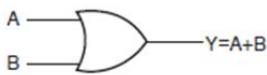


Aşağıda TTL teknolojisi ile üretilmiş 2 girişli VE kapı entegresinin (7408, 5408) ve CMOS teknolojisi ile üretilmiş 2 girişli VE kapı entegresinin (4081) iç yapısı verilmiştir.



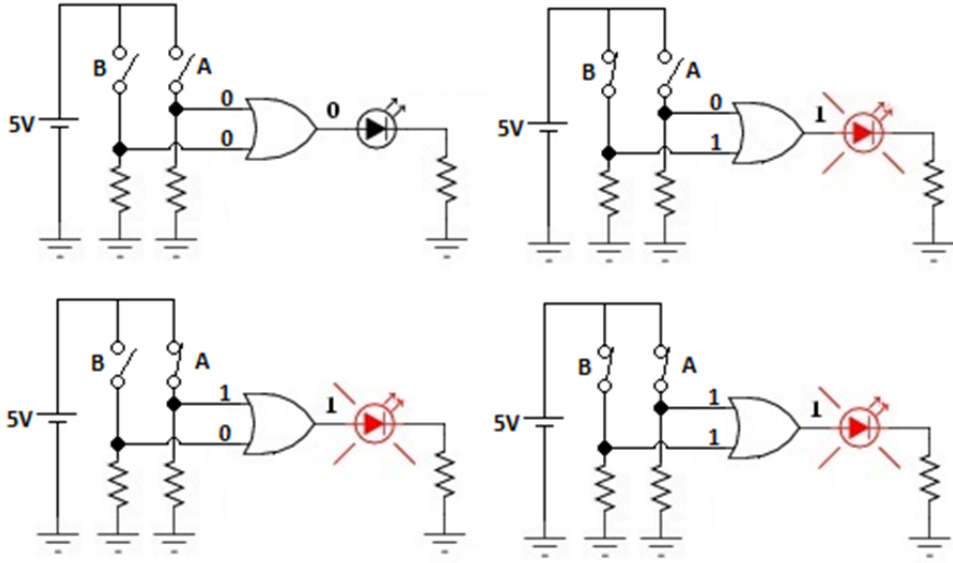
5.2. VEYA KAPISI (OR GATE)

Aşağıda 2, 3 ve 4 girişli VEYA kapılarının sembolleri, lojik ifadeleri ve 2 girişli VEYA kapısına ait doğruluk tablosu görülmektedir. Doğruluk tablosundan da görüldüğü üzere VEYA kapısının herhangi bir girişi "lojik 1" veya her iki girişi de "lojik 1" durumda iken çıkışı "lojik 1" olmaktadır.

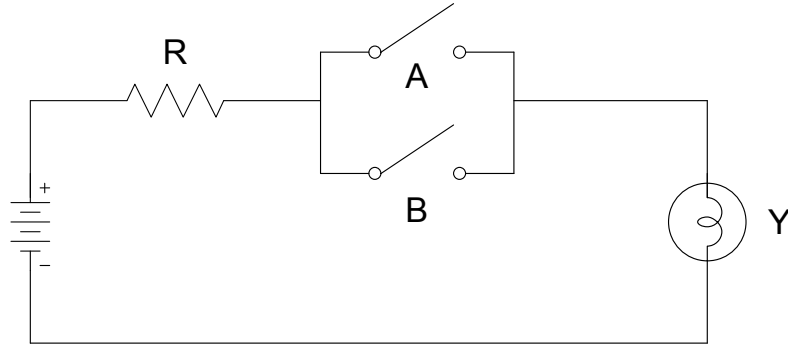


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

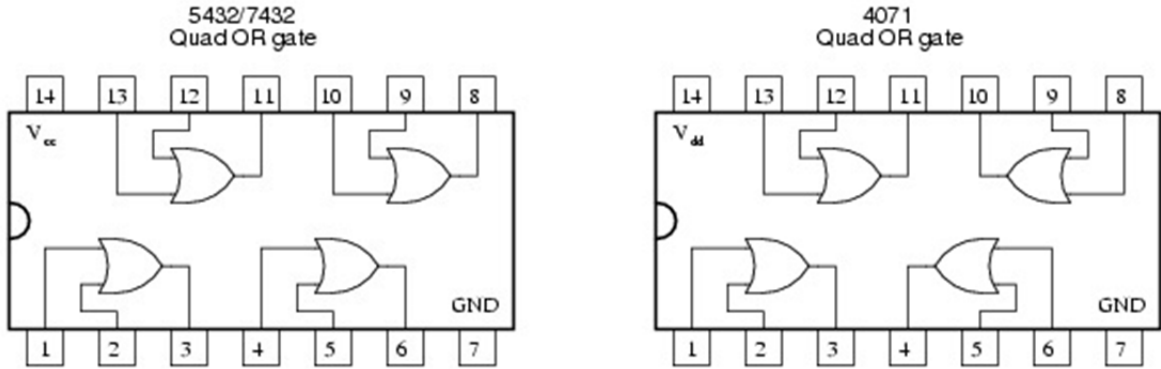
2 girişli VEYA kapısına ait doğruluk tablosunun nasıl oluşturulduğu aşağıdaki şekillerde ayrıntılı olarak görülmektedir.



Aşağıda VEYA kapısının elektriksel eşdeğer devresi verilmiştir. VEYA kapısı elektriksel anlamda anahtarların paralel bağlanmasını ifade etmektedir. Aşağıdaki devrede, Y lambasının yanıyor olması "lojik 1", Y lambasının sönmük olması ise "lojik 0" anlamına gelmektedir. Şekilden de görüleceği üzere Y lambasının yanması için anahtarlardan herhangi birinin (A ve B) veya ikisinin de kapalı olması gerekmektedir.

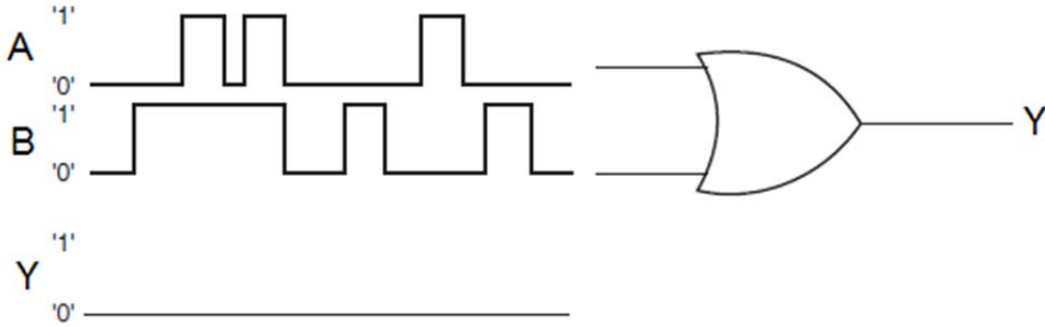


Aşağıda TTL teknolojisi ile üretilmiş 2 girişli VEYA kapı entegresinin (7432, 5432) ve CMOS teknolojisi ile üretilmiş 2 girişli VEYA kapı entegresinin (4071) iç yapısı verilmiştir.



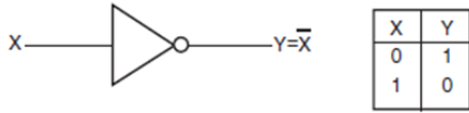
Soru: Sadece iki girişli VEYA kapısı kullanarak dört girişli VEYA kapısı nasıl elde ederiz?

Soru: iki girişli VEYA kapısının girişlerine A ve B sinyalleri uygulanmaktadır. VEYA kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.

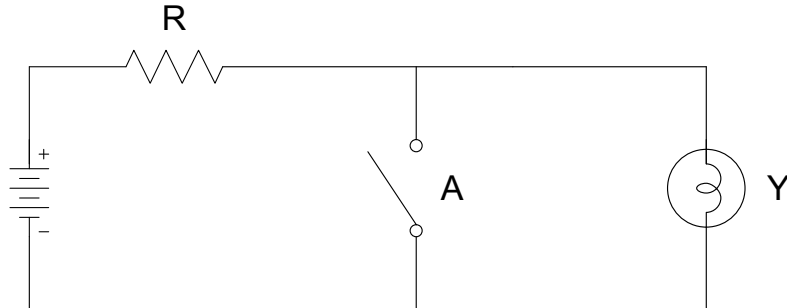


5.3. DEĞİL KAPISI (NOT GATE)

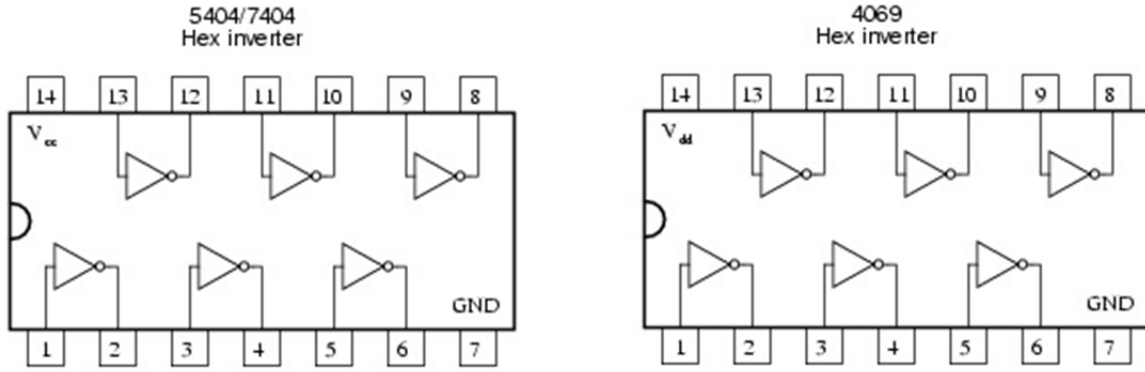
Aşağıda DEĞİL kapısının sembolü, lojik ifadesi ve doğruluk tablosu görülmektedir. Doğruluk tablosundan da görüldüğü üzere DEĞİL kapısının çıkışı, girişinin evriğidir (tümleyenidir).



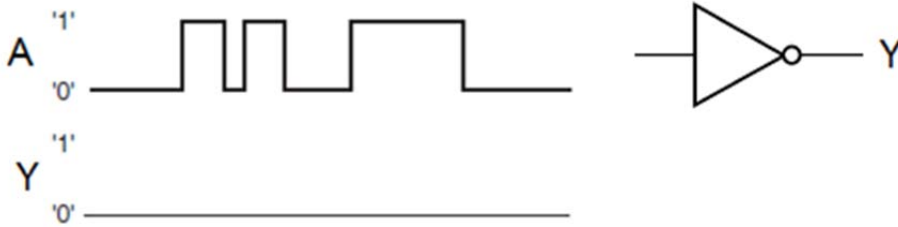
Aşağıda DEĞİL kapısının elektriksel eşdeğer devresi verilmiştir. DEĞİL kapısı veya DEĞİL işlemi (tümleyen alma işlemi) elektriksel anlamda anahtarın ve/veya anahtarların çıkışa (veya çıkışa bağlı lambaya) paralel bağlanmasını ifade etmektedir. Şekilden de görüleceği üzere Y lambasının yanması için A anahtarının açık olması gerekmektedir. A anahtarı kapalı olduğu zaman çıkış veya çıkışa bağlı lampa anahtar üzerinden kısa devre olmakta ve yanmamaktadır.



Aşağıda TTL teknolojisi ile üretilmiş DEĞİL kapı entegresinin (7404, 5404) ve CMOS teknolojisi ile üretilmiş DEĞİL kapı entegresinin (4069) iç yapısı verilmiştir.

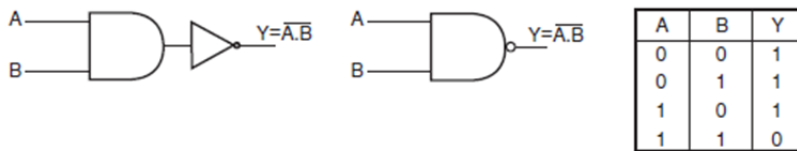


Soru: Aşağıda verilen DEĞİL kapısının girişine A sinyali uygulanmaktadır. DEĞİL kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.

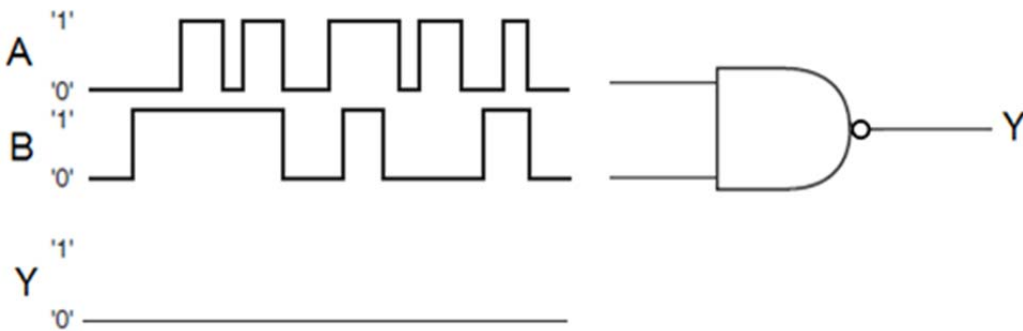


5.4. VE-DEĞİL KAPISI (NAND GATE)

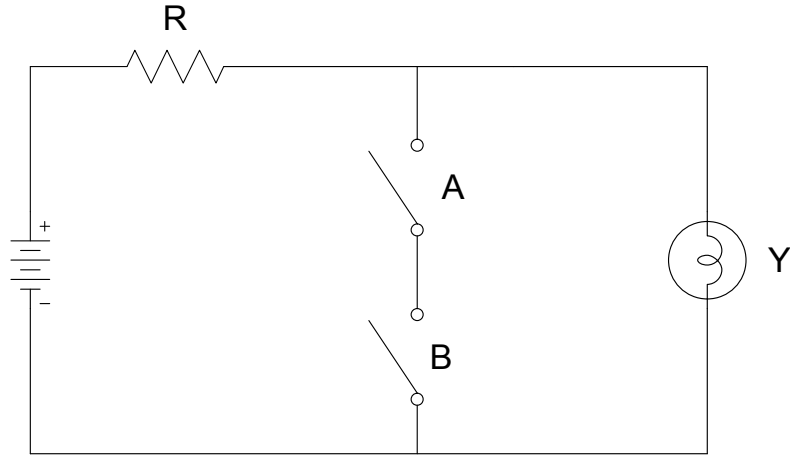
Aşağıda 2 girişli VE-DEĞİL kapısının sembolü, lojik ifadesi ve 2 girişli VE-DEĞİL kapısına ait doğruluk tablosu görülmektedir.



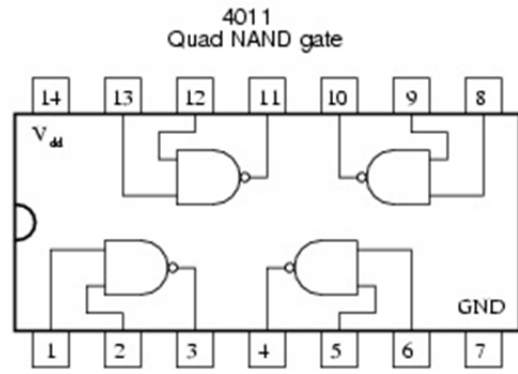
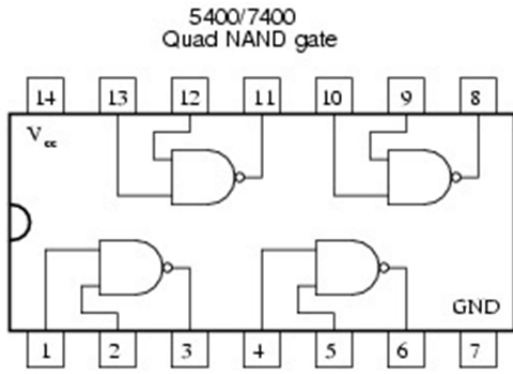
Soru: iki girişli VE-DEĞİL kapısının girişlerine A ve B sinyalleri uygulanmaktadır. VE-DEĞİL kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.



Aşağıda VE-DEĞİL kapısının elektriksel eşdeğer devresi verilmiştir.

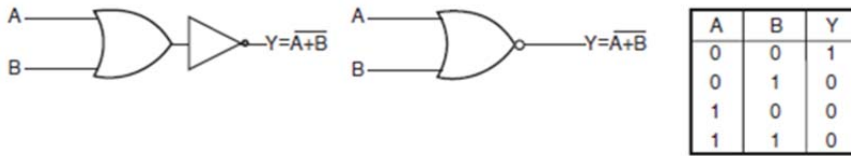


Aşağıda TTL teknolojisi ile üretilmiş 2 girişli VE-DEĞİL kapı entegresinin (7400, 5400) ve CMOS teknolojisi ile üretilmiş 2 girişli VE-DEĞİL kapı entegresinin (4011) iç yapısı verilmiştir.

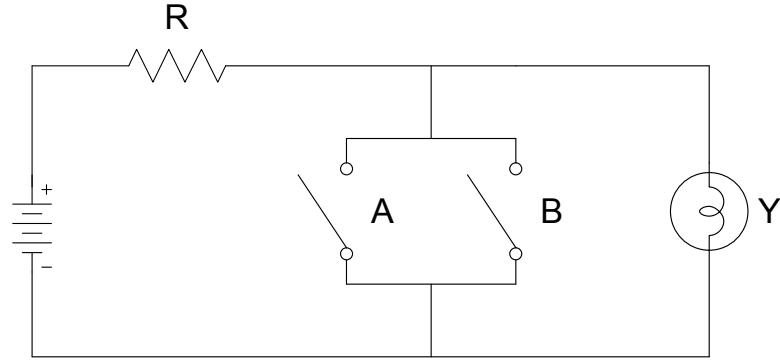


5.5. VEYA-DEĞİL KAPISI (NOR GATE)

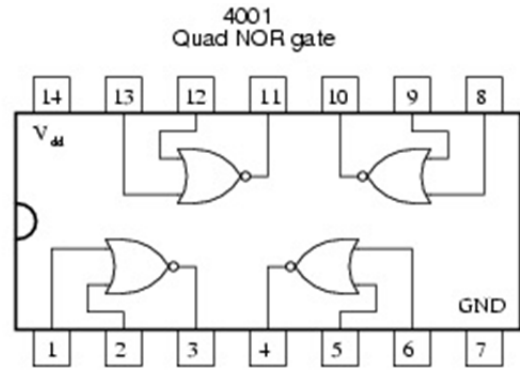
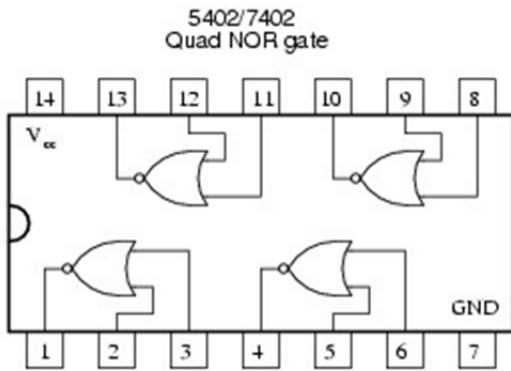
Aşağıda 2 girişli VEYA-DEĞİL kapısının sembolü, lojik ifadesi ve 2 girişli VEYA-DEĞİL kapısına ait doğruluk tablosu görülmektedir.



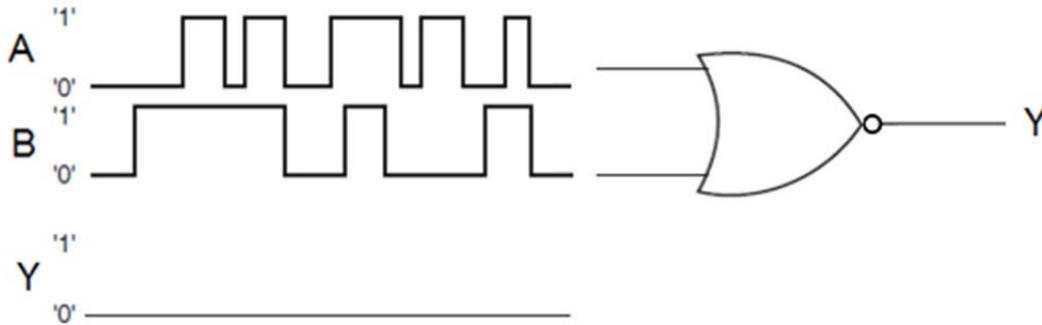
Aşağıda VEYA-DEĞİL kapısının elektriksel eşdeğer devresi verilmiştir.



Aşağıda TTL teknolojisi ile üretilmiş 2 girişli VEYA-DEĞİL kapı entegresinin (7402, 5402) ve CMOS teknolojisi ile üretilmiş 2 girişli VEYA-DEĞİL kapı entegresinin (4001) iç yapısı verilmiştir.

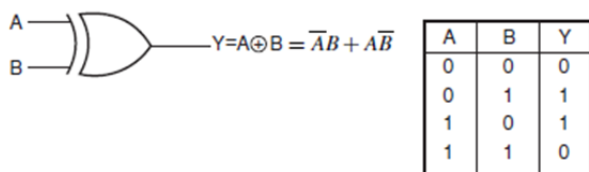


Soru: iki girişli VEYA-DEĞİL kapısının girişlerine A ve B sinyalleri uygulanmaktadır. VEYA-DEĞİL kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.

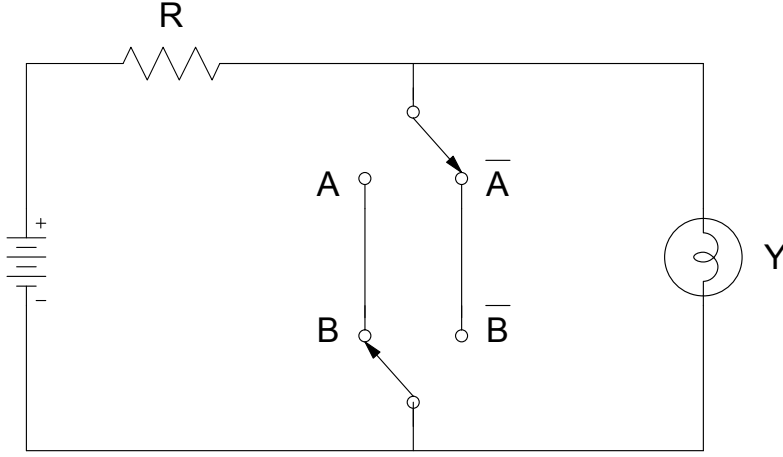


5.6. ÖZEL VEYA KAPISI (EX-OR GATE)

Aşağıda 2 girişli ÖZEL VEYA kapısının sembolü, lojik ifadesi ve 2 girişli ÖZEL VEYA kapısına ait doğruluk tablosu görülmektedir.

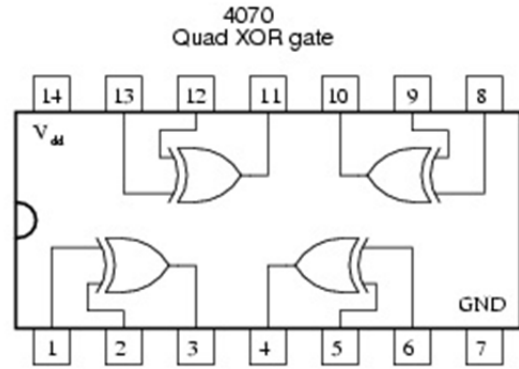
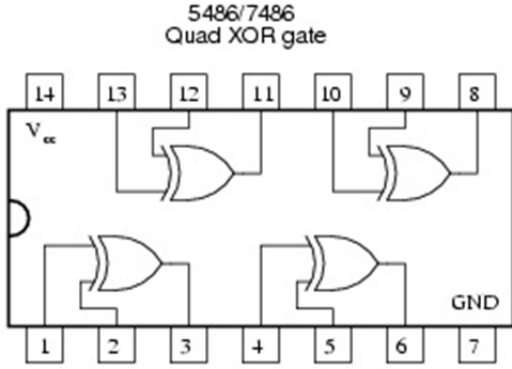


Aşağıda ÖZEL VEYA kapısının elektriksel eşdeğer devresi verilmiştir.

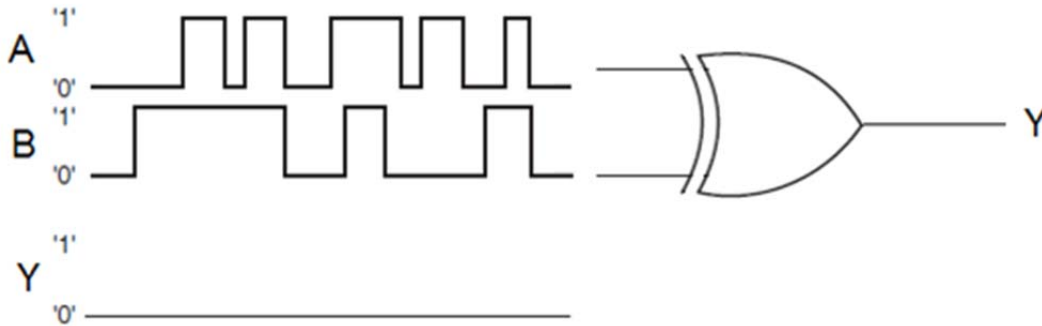


Doğruluk tablosundan da görüldüğü gibi kapı girişleri bir bit olarak düşünüldüğünde, devrenin çıkışına yansıyan Y lambasının yanma işlevi bir bit eşitsizlik devresi olarak da ifade edilir.

Aşağıda TTL teknolojisi ile üretilmiş 2 girişli ÖZEL VEYA kapı entegresinin (7486, 5486) ve CMOS teknolojisi ile üretilmiş 2 girişli ÖZEL VEYA kapı entegresinin (4070) iç yapısı verilmiştir.

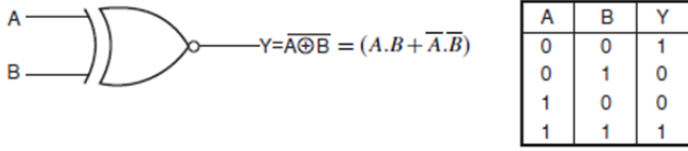


Soru: iki girişli ÖZEL VEYA kapısının girişlerine A ve B sinyalleri uygulanmaktadır. ÖZEL VEYA kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.

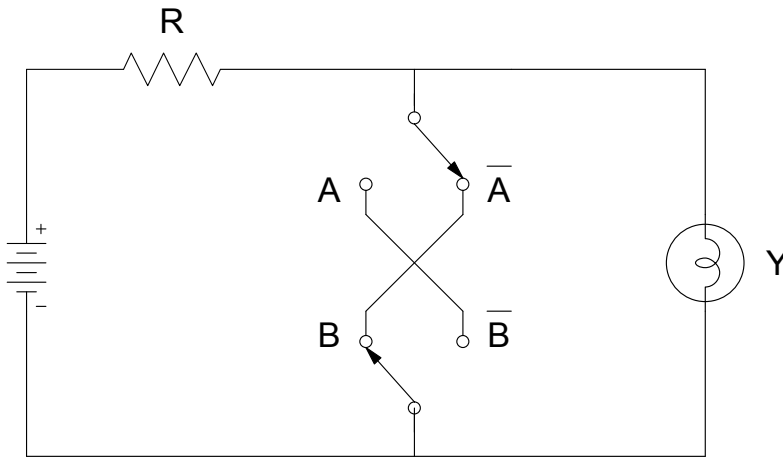


5.7. ÖZEL VEYA-DEĞİL KAPISI (EX-NOR GATE)

Aşağıda 2 girişli ÖZEL VEYA-DEĞİL kapısının sembolü, lojik ifadesi ve 2 girişli ÖZEL VEYA-DEĞİL kapısına ait doğruluk tablosu görülmektedir.

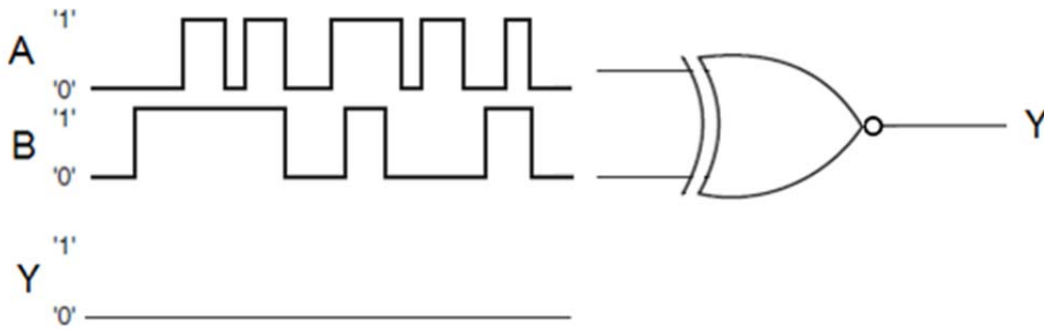


Aşağıda ÖZEL VEYA-DEĞİL kapısının elektriksel eşdeğer devresi verilmiştir.



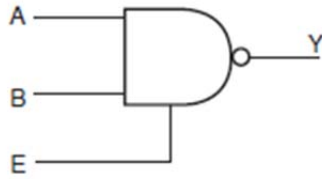
Doğruluk tablosundan da görüldüğü gibi kapı girişleri bir bit olarak düşünüldüğünde, devrenin çıkışına yansıyan Y lambasının yanma işlevi bir bit eşitlik devresi olarak da ifade edilir.

Soru: iki girişli ÖZEL VEYA-DEĞİL kapısının girişlerine A ve B sinyalleri uygulanmaktadır. ÖZEL VEYA-DEĞİL kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.



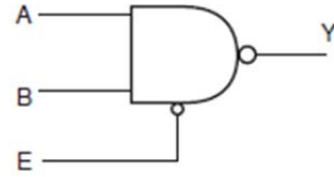
5.8. ÜÇ KONUMLU LOJİK KAPILAR (TRISTATE LOGIC GATES)

Bu konuya kadar anlatılan yedi lojik kapının “lojik 0” ve “lojik 1” olmak üzere iki çıkış durumu (seviyesi) bulunmaktadır. Ancak üç konumlu lojik kapıların, “lojik 0”, “lojik 1” çıkış durumları yanında Yüksek-Direnç çıkış durumu bulunmaktadır. Yüksek Direnç durumu lojik kapıda ekstra olarak bulunan Enable girişi ile kontrol edilmektedir. Bu konuya kadar anlatılan tüm kapılar (VE, VEYA, DEĞİL, VE-DEĞİL, VEYA-DEĞİL, ÖZEL VEYA, ÖZEL VEYA-DEĞİL) üç konumlu olarak üretilebilmektedir. Kapılarda bulunan Enable girişi, “lojik 0” aktif veya “lojik 1” aktif olabilmektedir. Aşağıda Enable girişi “lojik 0” aktif ve “lojik 1” aktif olan 2 girişli VE-DEĞİL (NAND) kapısının sembolleri ve doğruluk tabloları ayrı ayrı olarak verilmiştir.



A	B	E	Y
0	0	0	Y.D.
0	0	1	1
0	1	0	Y.D.
0	1	1	1
1	0	0	Y.D.
1	0	1	1
1	1	0	Y.D.
1	1	1	0

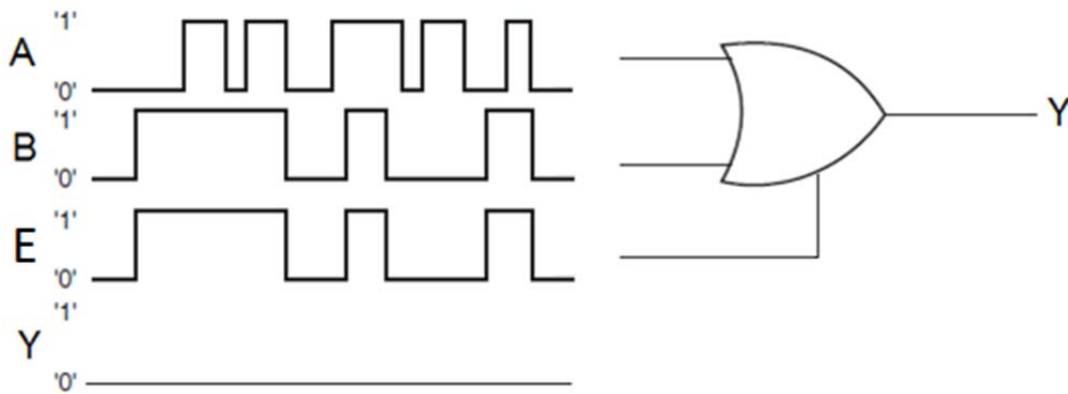
Y.D. = Yüksek Direnç Durumu



A	B	E	Y
0	0	0	1
0	0	1	Y.D.
0	1	0	1
0	1	1	Y.D.
1	0	0	1
1	0	1	Y.D.
1	1	0	0
1	1	1	Y.D.

Y.D. = Yüksek Direnç Durumu

Soru: Enable girişli VEYA kapısının girişlerine A, B ve E sinyalleri uygulanmaktadır. VEYA kapısının çıkışının (Y) zamana bağlı değişimini aşağıda belirtilen boşluğa ölçekli olarak çiziniz.



Soru: Üç girişli VEYA-DEĞİL (NOR) kapısı kullanarak DEĞİL (NOT) kapısı elde ediniz?

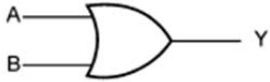
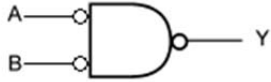
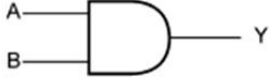
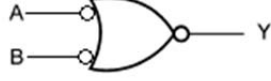
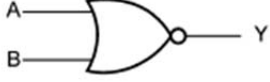



Soru: İki girişli VE-DEĞİL (NAND) kapısı kullanarak DEĞİL (NOT) kapısı elde ediniz?

Soru: İki girişli VE (AND) kapıları ve bir adet DEĞİL (NOT) kapısı kullanarak üç girişli VE-DEĞİL (NAND) kapısı elde ediniz?

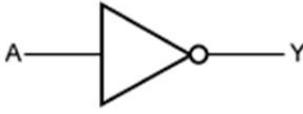

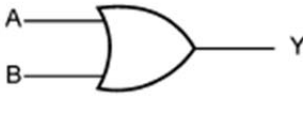
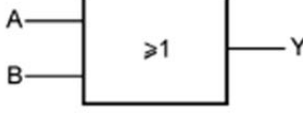
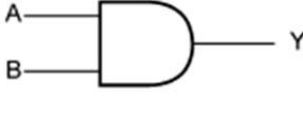

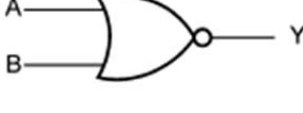
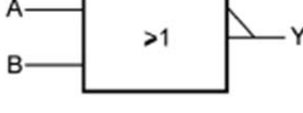



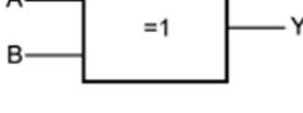

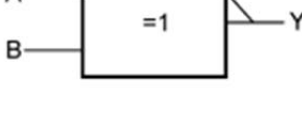
Soru: İki girişli VE-DEĞİL (NAND) kapıları kullanarak üç girişli VE-DEĞİL (NAND) kapısı elde ediniz?

Soru: İki girişli VE-DEĞİL (NAND) kapıları kullanarak iki girişli VE (AND) kapısı elde ediniz?

5.9. Lojik Kapılar Arasındaki İlişkiler

	Lojik Kapı	Lojik İfadesi ve Eşdeğer İfadesi	Eşdeğer Lojik Kapı
OR		$Y = A + B = \overline{\overline{A} \cdot \overline{B}}$	
AND		$Y = A \cdot B = \overline{\overline{A} + \overline{B}}$	
NOR		$Y = \overline{A + B} = \overline{A} \cdot \overline{B}$	
NAND		$Y = \overline{A \cdot B} = \overline{A} + \overline{B}$	

5.10. Lojik Kapıların Değişik Standartlarda Kullanılan Simgeleri

	IEEE Sembolü	ANSI Sembolü
NOT		
OR		
AND		
NOR		
NAND		
EX-OR		
EX-NOR		

Soru: İki girişli VE-DEĞİL (NAND) kapıları kullanarak iki girişli VEYA (OR) kapısı elde ediniz?

Soru: İki girişli VEYA-DEĞİL (NOR) kapıları kullanarak iki girişli VEYA (OR) kapısı elde ediniz?

Soru: İki girişli VEYA-DEĞİL (NOR) kapıları kullanarak iki girişli VE (AND) kapısı elde ediniz?

Soru: $Y = \overline{(A \cdot B + C \cdot D)}$ ifadesini lojik kapılar kullanarak çiziniz?

Soru: $Y = \overline{(A + B) \cdot (C + D)}$ ifadesini lojik kapılar kullanarak çiziniz?

Soru: Lojik kapılar kullanılarak elde edilmiş devrenin lojik ifadesini yazınız? $Y(A,B)=?$

6. BOOLE CEBRİ (BOOLEAN ALGEBRA)

1850'li yıllarda George Boole tarafından geliştirilen Boole Cebri, sayısal devrelerin analiz ve tasarımını sağlayan matematiksel teoridir. [Sayısal bilgisayar](#) devreleri uygulamasında, ikili değişkenler üzerine tanımlanan sayısal (dijital) operasyonları gösterir.

Boole Cebri ikili sayı sistemine dayanır. Bu sistemde yer alan '0' lar kapalı (off), yanlış (false) gibi ifadeleri, '1' ler ise açık (on), doğru (true) gibi ifadeleri temsil eder.

6.1. Boole Cebri Kuralları

6.1.1. Kural 1: 0 ve 1 ile yapılan işlemler

$$\begin{array}{ll} 0 \cdot 0 = 0 & 0+0 = 0 \\ 0 \cdot 1 = 0 & 0+1 = 1 \\ 0 \cdot A = 0 & 0+A = A \\ 1 \cdot 1 = 1 & 1+1 = 1 \\ 1 \cdot A = A & 1+A = 1 \end{array}$$

Örnek: $0+(A+B \cdot C+C \cdot D+E \cdot F)=1 \cdot (A+B \cdot C+C \cdot D+E \cdot F)=A+B \cdot C+C \cdot D+E \cdot F$

6.1.2. Kural 2: Benzerlik Kuralı (Identity Law)

$$\begin{array}{ll} A+A = A & A \cdot A = A \\ A+A+A+A+\dots+A = A & A \cdot A \cdot A \cdot A \dots A = A \end{array}$$

Örnek: $A \cdot (A \cdot B + C) = ?$

Örnek: $B \cdot (A + B + C) = ?$

Örnek: $(A \cdot A \cdot \bar{B} \cdot \bar{B} + C \cdot C \cdot C) \cdot (A \cdot A \cdot \bar{B} \cdot \bar{B} + A \cdot \bar{B} \cdot \bar{B} + C \cdot C) = ?$

6.1.3. Kural 3: Tümlenme Kanunu (Complementation Laws)

$$A + \bar{A} = 1 \qquad A \cdot \bar{A} = 0$$

6.1.4. Kural 4: Değişme Özelliği (Commutative Laws)

$$A + B = B + A \qquad A \cdot B = B \cdot A$$

6.1.5. Kural 5: Birleşme Özelliği (Associative Laws)

$$A+(B+C) = (A+B)+C = (A+C)+B = A+(C+B) \quad A.(B.C) = (A.B).C = (A.C).B = A.(C.B)$$

6.1.6. Kural 6: Dağılma Özelliği (Distributive Laws)

$$A.(B+C) = A.B + A.C \quad A+(B.C) = (A+B).(A+C)$$

6.1.7. Kural 7: Yutma Özelliği (Absorption Law or Redundancy Law)

$$A+A.B = A.(1+B) = A \quad A.(A+B) = A.A+A.B = A+A.B = A$$

6.1.8. Kural 8: Çift Negatif Kuralı (Double Negative Law - Involution Law)

$$\overline{\overline{A}} = A$$

6.1.9. Kural 9: De Morgan Kuralı

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

Örnek: $\overline{A.B + C.D + E.F} = ?$

Örnek: $\overline{(A+B).(C+D).(E+F)} = ?$

6.1.10. Kural 10:

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

İspat :

6.1.11. Kural 11: Basitleştirme Kanunu (Minimisation Law)

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

6.1.12. Kural 12: Konsensüs Teorisi (Consensus Theorem)

$$A.B + \bar{A}.C + B.C = A.B + \bar{A}.C \quad (A + B).(\bar{A} + C).(B + C) = (A + B).(\bar{A} + C)$$

Tablo 1. Konsensüs Teorisinin Doğruluk Tablosu ile ispatlanması

A	B	C	AB	$\bar{A}C$	BC	$AB + \bar{A}C + BC$	$AB + \bar{A}C$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	1
0	1	0	0	0	0	0	0
0	1	1	0	1	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	1	0	1	1	1

6.1.13. Kural 13: Transpozisyon Teorisi (Transposition Theorem)

$$A.B + \bar{A}.C = (A + C).(B + \bar{A}) \quad (A + B).(\bar{A} + C) = (A.C) + (B.\bar{A})$$

Tablo 2. Transpozisyon Teorisinin Doğruluk Tablosu ile ispatlanması

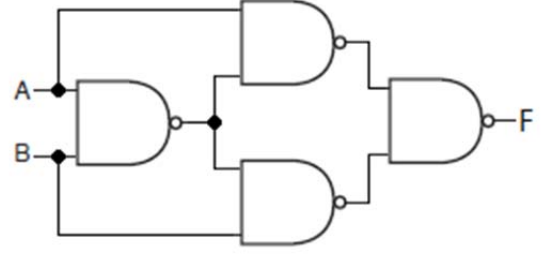
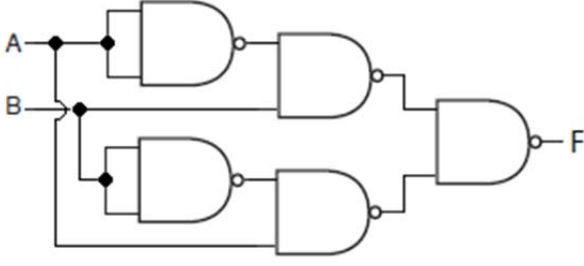
A	B	C	AB	$\bar{A}C$	A+C	$\bar{A}+B$	$AB + \bar{A}C$	$(\bar{A}+B)(A+C)$
0	0	0	0	0	0	1	0	0
0	0	1	0	1	1	1	1	1
0	1	0	0	0	0	1	0	0
0	1	1	0	1	1	1	1	1
1	0	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0
1	1	0	1	0	1	1	1	1
1	1	1	1	0	1	1	1	1

Örnek: En az sayıda 2 girişli VE-DEĞİL (NAND) kapıları kullanarak ÖZEL-VEYA (EX-OR) kapısı elde ediniz ve çiziniz?

$$F = \bar{A}.B + A.\bar{B}$$

I. Yöntem : $F = \bar{F} = ?$

II. Yöntem : $F = \bar{A}.B + A.\bar{B} = ?$



Örnek: $A + \overline{B} + C \cdot \overline{D}$ ifadesini Boole Cebri kurallarını kullanarak sadeleştiriniz?

Örnek: $F = A \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + B \cdot \overline{C} + A \cdot B \cdot C$ ifadesini Boole Cebri kurallarını kullanarak sadeleştiriniz?

Örnek: $F = \overline{A} \cdot B + \overline{A} \cdot \overline{B} + A \cdot B$ ifadesini Boole Cebri kurallarını kullanarak sadeleştiriniz?

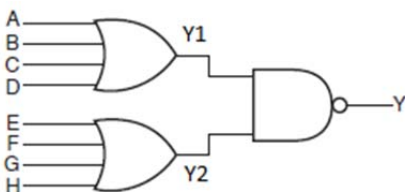
7. KOMBİNASYONLU (KARMAŞIK) LOJİK DEVRELER

Birden fazla lojik kapının kombinasyonlu (Karmaşık) olarak birbirine bağlanmasından oluşan devrelere Kombinasyonlu Lojik Devreler adı verilir. Bu devreler oluşturulurken veya devrenin çıkış fonksiyon değeri aranırken aşağıdaki sıralama izlenir.

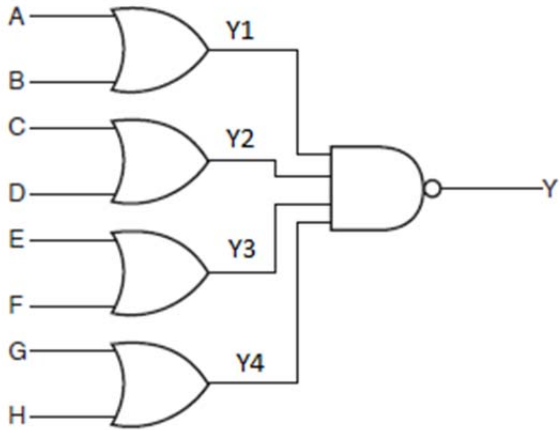
- Lojik kapılı devre verilmişse verilen lojik devreden yararlanarak lojikselse ifadenin çıkarılması,
- Verilen, anlatılan veya var olan olayın lojik ifadesinin çıkarılması
- Çıkarılan lojik ifadenin sadeleştirilmesi,
- Sadeleştirilmiş ifadenin lojik kapılarla gerçekleştirilmesi, şeklinde bir senteze tabi tutulur.

7.1. Lojik Kapı Devresi Verilen Sistemin Lojikselse İfadesinin Bulunması

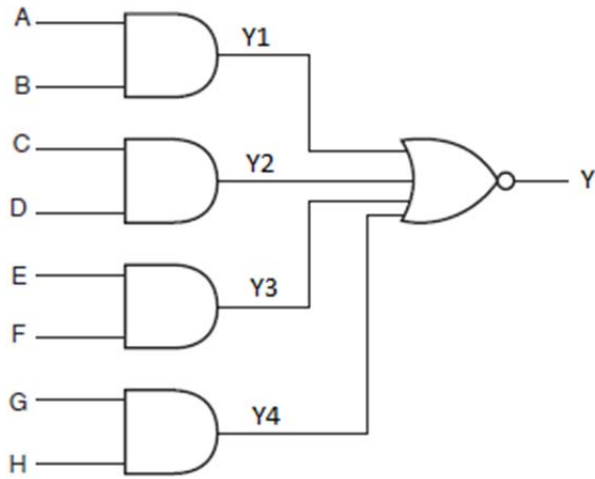
Örnek: Lojik kapı devresi verilen sistemin lojik ifadesini bulunuz?



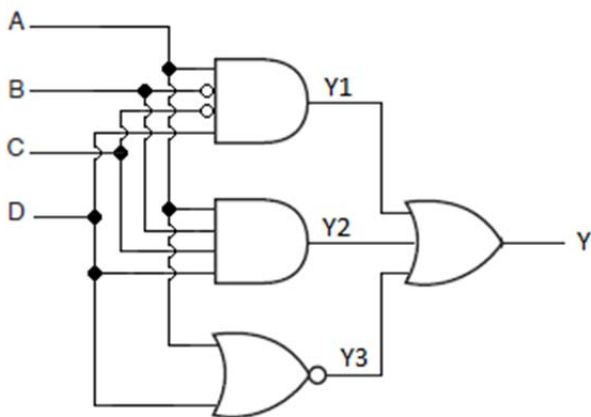
Örnek: Lojik kapı devresi verilen sistemin lojik ifadesini bulunuz?



Örnek: Lojik kapı devresi verilen sistemin lojik ifadesini bulunuz?



Örnek: Lojik kapı devresi verilen sistemin lojik ifadesini bulunuz?



$f(A_0) = ?$

$f(A_1) = ?$

$f(A_2) = ?$

$f(A_3) = ?$

$f(A_4) = ?$

$f(A_5) = ?$

Örnek: On tabanlı sayı ailesine ait rakamları Artı-3 (Excess-3) koduna çeviren sistemi lojik kapılar ile tasarlayınız?

GİRİŞ $[X]_{10}$					ÇIKIŞ $[X]_{+3}$				
X_{10}	B3	B2	B1	B0	A3	A2	A1	A0	X_{+3}

$f(A_0) = ?$

$f(A_1) = ?$

$f(A_2) = ?$

$f(A_3) = ?$

8. LOJİK İFADELERİN SADELEŞTİRİLMESİ

Elde edilen veya verilen lojik ifadelerin en kısa şekli en ideal durumdur. Çünkü ifadenin uzunluğu oranında bunu gerçekleştirecek lojik devredeki eleman sayısı artacaktır. Dolayısıyla devrenin maliyeti artacaktır. Bundan dolayı bazı işlem kuralları uygulanarak lojik ifadeler sadeleştirilir.

8.1. Cebirsel (Klasik) Sadeleştirme

Boole cebirinde anlatılan kurallar uygulanarak yapılan sadeleştirme şeklidir.

Örnek: $A.\bar{B} + B.\bar{C} + A.B.\bar{C} + \bar{A}.C$ ifadesini sadeleştiriniz?

$$A.\bar{B}.(C + \bar{C}) + (A + \bar{A}).B.\bar{C} + A.B.\bar{C} + \bar{A}.(B + \bar{B}).C$$

$$A.\bar{B}.C + A.\bar{B}.\bar{C} + A.B.\bar{C} + \bar{A}.B.\bar{C} + A.B.\bar{C} + \bar{A}.B.C + \bar{A}.\bar{B}.C$$

$$A.\bar{B}.C + A.\bar{B}.\bar{C} + A.B.\bar{C} + \bar{A}.B.\bar{C} + \bar{A}.B.C + \bar{A}.\bar{B}.C$$

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

Örnek: $\bar{C}.D + B.\bar{C}.\bar{D} + A.\bar{B}.C + \bar{A}.\bar{B}.C.D + A.\bar{B}.D + A.\bar{B}.\bar{C}.\bar{D}$ ifadesini sadeleştiriniz?

$$(\bar{A}.\bar{B} + A.\bar{B} + \bar{A}.B + A.B).\bar{C}.D + (A + \bar{A}).B.\bar{C}.\bar{D} + A.\bar{B}.C.(D + \bar{D}) + \bar{A}.\bar{B}.C.D +$$

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

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

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

Lojikel işlem kuralları uygulanarak yapılan sadeleştirme, hem zor ve hem de hesaplama süresi uzun olduğundan dolayı genelde **KARNAUGH** haritasıyla sadeleştirme kullanılır.

8.2. MIN ve MAX Terimler

MIN VE MAX TERİMLER

C	B	A	MAX - TERİM	MIN - TERİM	SIRA
0	0	0	$M_0 = A + B + C$	$m_0 = \bar{A} \cdot \bar{B} \cdot \bar{C}$	0
0	0	1	$M_1 = \bar{A} + B + C$	$m_1 = A \cdot \bar{B} \cdot \bar{C}$	1
0	1	0	$M_2 = A + \bar{B} + C$	$m_2 = \bar{A} \cdot B \cdot \bar{C}$	2
0	1	1	$M_3 = \bar{A} + \bar{B} + C$	$m_3 = A \cdot B \cdot \bar{C}$	3
1	0	0	$M_4 = A + B + \bar{C}$	$m_4 = \bar{A} \cdot \bar{B} \cdot C$	4
1	0	1	$M_5 = \bar{A} + B + \bar{C}$	$m_5 = A \cdot \bar{B} \cdot C$	5
1	1	0	$M_6 = A + \bar{B} + \bar{C}$	$m_6 = \bar{A} \cdot B \cdot C$	6
1	1	1	$M_7 = \bar{A} + \bar{B} + \bar{C}$	$m_7 = A \cdot B \cdot C$	7

Yukarıdaki tablodan görüldüğü gibi çarpım işlemi (And işlemi) MIN terimleri, toplam işlemi (OR işlemi) MAX terimleri ifade etmektedir.

$$F(A, B, C, D) = \sum_m(0, 1, 5, 9) = m_0 + m_1 + m_5 + m_9 \rightarrow \text{Çarpımların toplamı SOP (Sum of Products)}$$

$$F(A, B, C, D) = \prod_M(4, 5, 7, 10) = M_4 \cdot M_5 \cdot M_7 \cdot M_{10} \rightarrow \text{Toplamların çarpımı POS (Product of Sums)}$$

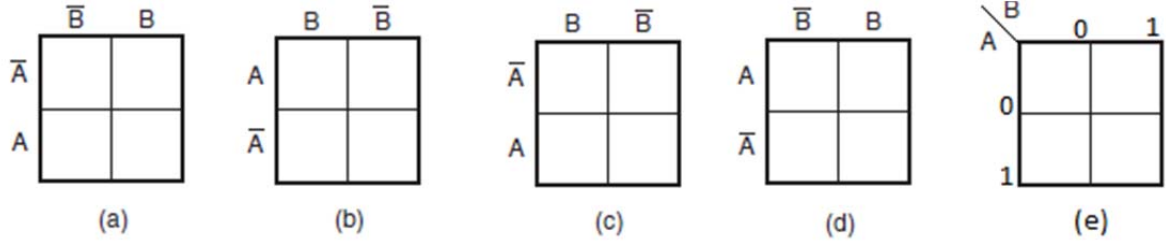
$$M_i = \overline{m_i}, m_i = \overline{M_i} \rightarrow m_2 = \overline{M_2} = \overline{A + \bar{B} + C} = \bar{A} \cdot B \cdot \bar{C}$$

8.3. Karnaugh Haritası ile Lojik İfadelerin Sadeleştirilmesi

Karnaugh haritası lojik ifadelerin grafiksel olarak gösterilmiş halidir. Başka bir ifade ile bazı özellikleri olan bir tablodur. Bu özellikler sayesinde lojik ifadeler kısa süre içerisinde en sade hale getirilir. Karnaugh haritasında sadeleştirilen ifade genelde en kısa şekildir. Başka yol ve yöntemle daha fazla sadeleştirme yapılamaz. Max ve Min ifadeler doğrudan karnaugh haritasına aktarılabilir. Ayrıca doğruluk tablosunda bulunan sonuçlar da kolayca karnaugh haritasına aktarılıp sadeleştirilebilir.

8.3.1. İki değişkenli Karnaugh Haritası

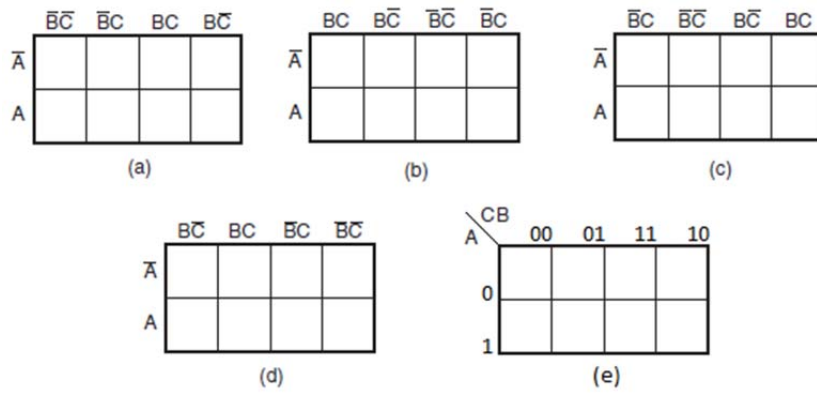
Karnaugh haritasını çizmek için kare şeklindeki tabloyu değişken sayısı ile orantılı olarak karelere ayırmak gerekir. Değişken sayısı n ise kare sayısı 2^n formülünden bulunabilir. Dolayısıyla n=2 için 4 küçük kareye bölünmesi gerekir. Karnaugh haritasının kenarlarındaki satır ve sütunlara değişkenler cinsinden aşağıdaki şekilde isimler verilir.



Yukarıdaki şekilden de görüldüğü iki değişkenli karnaugh haritası 5 farklı şekilde gösterilmektedir. Lojik devreler dersinde (e) seçeneğindeki gösterim kullanılacaktır.

8.3.2. Üç değişkenli Karnaugh Haritası

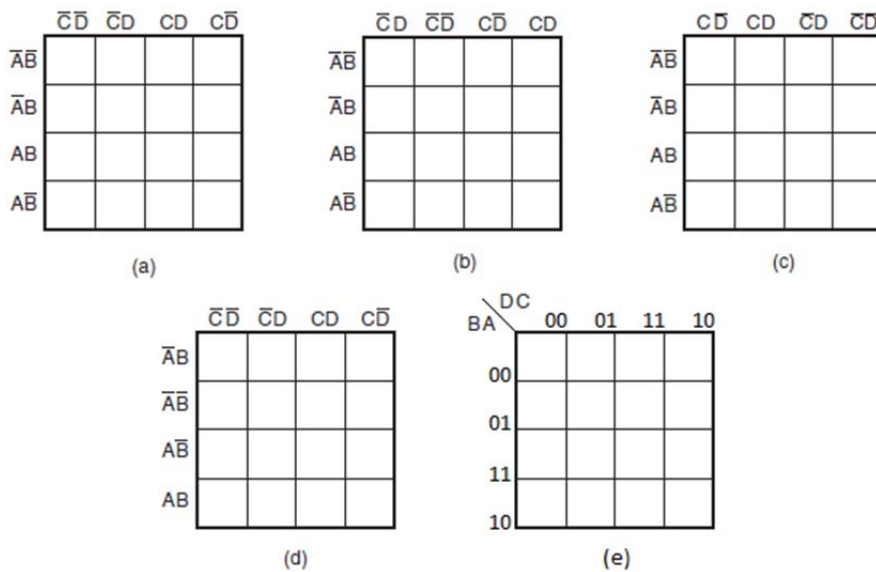
Değişken sayısı $n=3$ olduğu için tablonun 8 küçük kareye bölünmesi gerekir.



Yukarıdaki şekilden de görüldüğü üç değişkenli karnaugh haritası 5 farklı şekilde gösterilmektedir. Lojik devreler dersinde (e) seçeneğindeki gösterim kullanılacaktır.

8.3.3. Dört değişkenli Karnaugh Haritası

Değişken sayısı $n=4$ olduğu için tablonun 16 küçük kareye bölünmesi gerekir.



Yukarıdaki şekilden de görüldüğü dört değişkenli karnaugh haritası 5 farklı şekilde gösterilmektedir. Lojik devreler dersinde (e) seçeneğindeki gösterim kullanılacaktır.

Karnaugh haritasındaki küçük karelerin değişkenler cinsinden neler ifade ettiği ve küçük karelerin numaraları aşağıdaki şekilde verilmiştir.

Dec	Binary			
	(MSB)			(LSB)
Sıra	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
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

DC \ BA	00	01	11	10
00	0000	0100	1100	1000
01	0001	0101	1101	1001
11	0011	0111	1111	1011
10	0010	0110	1110	1010

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Örnek: Aşağıda doğruluk tablosunda elde edilen lojikel sonucu (F) karnaugh haritasına yerleştiriniz?

Dec	Binary				F
	(MSB)			(LSB)	
Sıra	D	C	B	A	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
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	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Örnek: $F = A.B.\bar{C}.\bar{D} + \bar{A}.B.\bar{C}.\bar{D} + A.\bar{B}.C.D + A.\bar{B}.C.\bar{D} + \bar{A}.\bar{B}.C.D + A.\bar{B}.C.D$
Verilen lojik ifadeyi karnaugh haritasına yerleştiriniz?

		DC			
		00	01	11	10
BA					
	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

		DC			
		00	01	11	10
BA					
	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

Örnek: $F = \bar{C}.D + B.\bar{C}.\bar{D} + A.\bar{B}.C + \bar{A}.\bar{B}.C.D + A.\bar{B}.D + A.\bar{B}.\bar{C}.\bar{D}$
Verilen lojik ifadeyi karnaugh haritasına yerleştiriniz?

Örnek: $F(A, B, C, D) = \sum_m(0, 1, 5, 9, 11, 15)$ Verilen MIN ifadeyi karnaugh haritasına yerleştiriniz?

		DC			
		00	01	11	10
BA					
	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

		DC			
		00	01	11	10
BA					
	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

8.3.4. Karnaugh Haritası Farketmez (Don't Care) İfadesi

Lojik fonksiyonlarda ifadelerden bazıları fonksiyonun alacağı değere bağlı değildir. Yani bu ifadeler ne olursa olsun fonksiyonun sonucuna etki etmez. Böyle ifadelere farketmez (Don't Care) denir. Karnaugh haritasında Don't Care ifadeler 'X' ile gösterilir. Bu ifadeler karnaugh haritasındaki gruplamalarda 0 veya 1 olarak alınabilir. Karnaugh haritasındaki her bir Don't Care ifade gruplamada olmak zorunda değildir. Sabit 0 veya 1 için geçerli olan bir gruba mutlaka ait olma zorunluluğu Don't Care için geçerli değildir.

Örnek: $F(A, B, C, D) = \sum_m(0, 1, 5, 9, 11, 15) + d(2, 3, 7)$ Verilen ifadeyi karnaugh haritasına yerleştiriniz?

8.3.5. Karnaugh Haritasında Grublama Yapılırken Dikkat Edilecek Hususlar

- 1- Üst ve alt kenarlar katlandığında karelerin içindeki doğruluk değerleri üst üste gelirse birbirine komşu olmaktadır.
- 2- Sağ ve sol köşeler katlandığında karelerin içindeki doğruluk değerleri üst üste gelirse birbirine komşu olmaktadır.
- 3- Köşeler katlandığında karelerin içindeki doğruluk değerleri üst üste gelirse dört köşe komşu olmaktadır.
- 4- Grublama yaparken Lojik 0' lar yada Lojik 1' ler gruplandırılır.
- 5- Lojik 1' ler gruba alınırken de Karnaugh haritasındaki Lojik 1' lerin tamamı seçilmeli, açıkta Lojik 1 kalmamalıdır. Lojik 1' ler gruplandığı zaman çıkan lojikselle ifade, **Çarpımların Toplamı** (SOP) şeklinde olmaktadır.
- 6- Lojik 0' lar gruba alınırken Karnaugh haritasındaki Lojik 0' ların tamamı seçilmeli, açıkta Lojik 0 kalmamalıdır. Lojik 0' lar gruplandığı zaman çıkan lojikselle ifade, **Toplamların Çarpımı** (POS) şeklinde olmaktadır.
- 7- Don't Care ifadeler 'X', karnaugh haritasındaki gruplamalarda 0 veya 1 olarak alınabilir. Karnaugh haritasındaki her bir Don't Care ifade gruplamada olmak zorunda değildir.
- 8- Grublama yaparken 2^n li gruplar (1, 2, 4, 8, 16, vs. gruplar) oluşturulmalıdır. Bu rakamların dışında yapılan ve ifade edilen gruplar yanlış gruplama olur.
- 9- Gruplamada ilk önce 2^n değerinin en büyük değerine sahip gruplar oluşturulur. Oluşturulamıyorsa bir sonraki küçük 2^n li gruplar oluşturulur.
- 10- Karnaugh haritasının tamamı bir grup oluyorsa fonksiyon sıfır veya bir'dir.
- 11- Karnaugh haritasındaki sabit bir veya sıfır birden fazla grupta yer alabilir.

Örnek: $F = A.\bar{B}.\bar{C} + A.C + \bar{A}.B.\bar{C} + A.B$ Verilen lojik ifadeyi karnaugh haritası ile sadeleştiriniz?

		CB			
		00	01	11	10
A	0	0	2	6	4
	1	1	3	7	5

Örnek: $F(A, B, C, D) = \sum_m(1, 2, 5, 6, 9) + d(10, 11, 12, 13, 14, 15)$ Verilen ifadeyi karnaugh haritası ile sadeleştiriniz?

		DC			
		00	01	11	10
BA	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

		DC			
		00	01	11	10
BA	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

Örnek: $F(A, B, C, D) = \sum_m(0, 2, 3, 5, 6, 7, 8, 10, 11, 14, 15)$ Verilen ifadeyi karnaugh haritası ile sadeleştiriniz?

Örnek: $F(A, B, C, D) = \sum_m(2, 4, 6, 8, 9, 10, 12, 13, 14)$ Verilen ifadeyi karnaugh haritası kullanarak;

a) Lojik 1' leri gruplandırarak sadeleştiriniz ve çiziniz?

b) Lojik 1' leri gruplandırarak sadeleştiriniz ve sadece NAND kapıları kullanarak çiziniz?

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

c) Lojik 0' leri gruplandırarak sadeleştiriniz? (I ve II. Yöntemi kullanarak sadeleştiriniz)

d) Lojik 0' leri gruplandırarak sadeleştiriniz ve sadece NOR kapıları kullanarak çiziniz?

Örnek: $F(A, B, C, D) = \sum_m(8, 9, 10, 11, 12, 13, 14, 15)$ Verilen ifadeyi karnaugh haritası ile sadeleştiriniz?

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

DC \ BA	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Örnek: $F(A, B, C, D) = \prod_M(0, 1, 2, 3, 4, 5, 6, 7)$ Verilen ifadeyi karnaugh haritası ile sadeleştiriniz?

Örnek: $F = ACD + AB\bar{D} + BC\bar{D} + A\bar{C}\bar{D}$ Verilen ifadeyi karnaugh haritası ile sadeleştiriniz?

		DC			
		00	01	11	10
BA	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

		DC			
		00	01	11	10
BA	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

Örnek $F = \bar{A}C\bar{D} + A\bar{D} + B\bar{C}\bar{D} + AB\bar{C} + A\bar{C}D + A\bar{B}\bar{C}\bar{D}$ Verilen ifadeyi karnaugh haritası ile sadeleştiriniz?

8. LOJİK İFADELERİN SADELEŞTİRİLMESİ (Devam...)

Açıklama :

1) Minimum terimler kullanıldığı zaman (Karnaugh haritasında 1' ler gruplandırıldığı zaman) elde edilen lojik ifade Çarpımların Toplamı şeklinde olmaktadır. Bu ifade lojik kapılar ile gerçekleştirildiği zaman AND-OR lojik kapı bileşimi ile gerçekleştirilmektedir. Eğer istenirse aynı lojik ifade NAND-NAND, OR-NAND veya NOR-OR lojik kapı bileşimleri ile gerçekleştirilebilir.

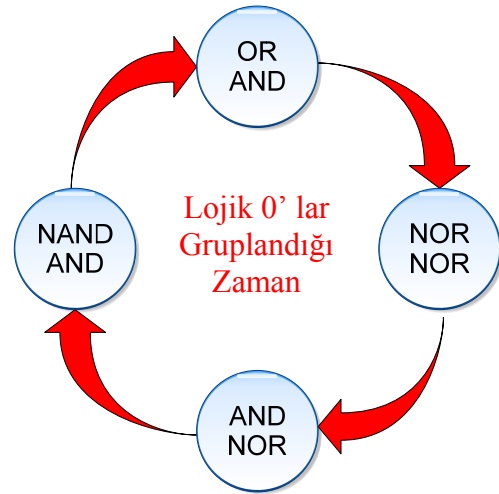
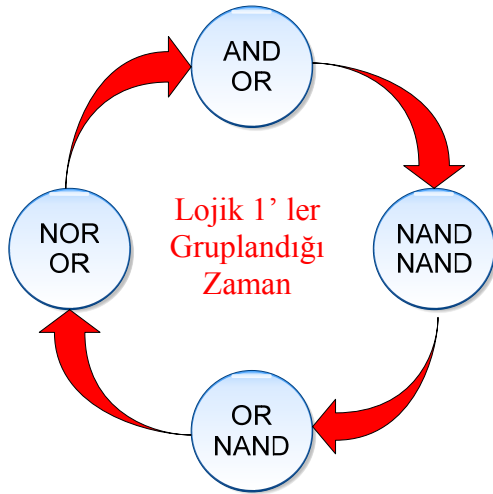
Örnek:

$$F = \bar{A}.B + A.\bar{C} \quad \text{---> AND-OR}$$

$$F = \bar{F} = \overline{\bar{A}.B + A.\bar{C}} = \overline{\bar{A}.B} . \overline{A.\bar{C}} \quad \text{---> NAND-NAND}$$

$$F = \overline{\bar{A}.B} . \overline{A.\bar{C}} = \overline{(A + \bar{B})} . \overline{(\bar{A} + C)} \quad \text{---> OR-NAND}$$

$$F = \overline{(A + \bar{B})} . \overline{(\bar{A} + C)} = \overline{(A + \bar{B}) + (\bar{A} + C)} \quad \text{---> NOR-OR}$$



2) Maksimum terimler kullanıldığı zaman (Karnaugh haritasında 0' lar gruplandırıldığı zaman) elde edilen lojik ifade Topamların Çarpımı şeklinde olmaktadır. Bu ifade lojik kapılar ile gerçekleştirildiği zaman OR-AND lojik kapı bileşimi ile gerçekleştirilmektedir. Eğer istenirse aynı lojik ifade NOR-NOR, AND-NOR veya NAND-AND lojik kapı bileşimleri ile gerçekleştirilebilir.

Örnek:

$$F = (A + B).(A + C) \quad \text{---> OR-AND}$$

$$F = \bar{F} = \overline{(A + B).(A + C)} = \overline{(A + B)} + \overline{(A + C)} \quad \text{---> NOR-NOR}$$

$$F = \overline{(A + B)} + \overline{(A + C)} = \overline{(\bar{A}.\bar{B})} + \overline{(\bar{A}.\bar{C})} \quad \text{---> AND-NOR}$$

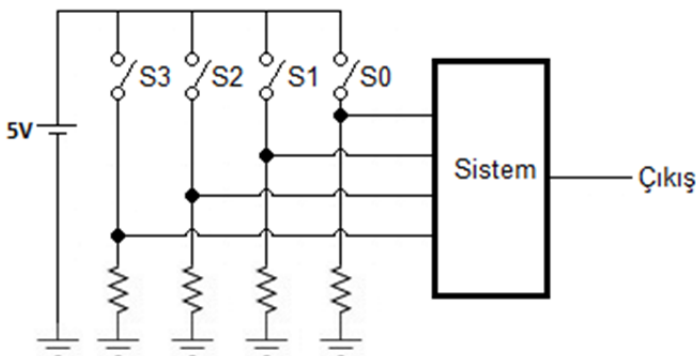
$$F = \overline{(\bar{A}.\bar{B})} + \overline{(\bar{A}.\bar{C})} = \overline{(\bar{A}.\bar{B}).(\bar{A}.\bar{C})} \quad \text{---> NAND-AND}$$

Örnek: Bir sendikanın yönetim kurulunda dört temsilci bulunmaktadır. 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 belli kararlar almaktadırlar. Sistem sonucun geçerli olup olmadığını hesaplayarak, geçerli ise bir lambanın yanmasını sağlamaktadır. Temsilcilerin temsil ettiği üye sayıları; A temsilcisi: 100 üye, B temsilcisi: 130 üye, C temsilcisi: 210 üye, D temsilcisi: 120 üye şeklindedir. Bir sonucun geçerli olabilmesi için salt çoğunluğun kabul etmesi gerekmektedir. Sisteme ait devreyi Karnaugh haritası kullanarak NAND kapılarıyla tasarlayınız?

Dec Sıra	Binary				Sonuç
	(MSB) D	C	B	(LSB) 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	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

DC BA	00	01	11	10
	00	0	4	12
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Örnek:

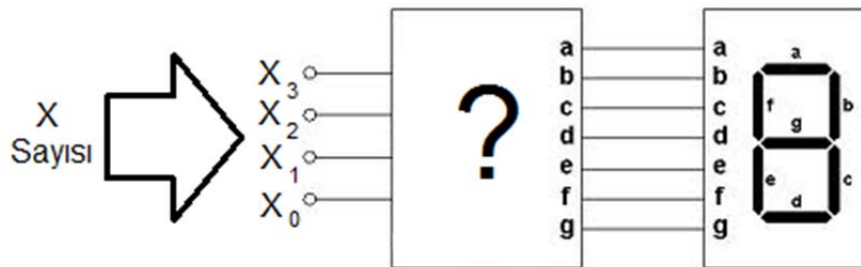


Yandaki şekilde görülen devre bir fotokopi makinesinin kontrol devresidir. Makinenin içerisinde kağıdın yolu üzerinde 4 adet anahtar bulunmaktadır. Sistem herhangi iki anahtarın kapatılması durumunda çıkış vermektedir. Sisteme ait devreyi Karnaugh haritası kullanarak NOR kapılarıyla tasarlayınız?

Dec	Binary				Çıkış
	(MSB)			(LSB)	
Sıra	S3	S2	S1	S0	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	
11	1	0	1	1	
12	1	1	0	0	
13	1	1	0	1	
14	1	1	1	0	
15	1	1	1	1	

	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Örnek: 2 tabanlı olarak verilen $X (X_3 X_2 X_1 X_0)$ sayısını, 10 tabanlı sayıya (1 Basamaklı = 1 Digit) dönüştüren devreyi tasarlayınız?

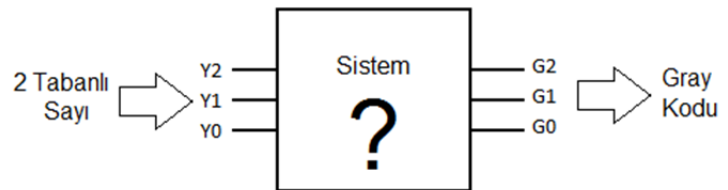


Dec	Binary				Çıkış						
	(MSB)			(LSB)	a	b	c	d	e	f	g
Sıra	X3	X2	X1	X0							
0	0	0	0	0							
1	0	0	0	1							
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							

	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

Örnek: 3 Bitlik iki tabanlı olarak verilen Y sayısını ($Y_2 Y_1 Y_0$), GRAY koduna dönüştüren devreyi lojik kapılar ile tasarlayınız?

Giriş			Çıkış		
Y2	Y1	Y0	G2	G1	G0
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	1			



9. LOJİK KAPILARIN YARIİLETKEN ELEMANLAR İLE ELDE EDİLMESİ

Lojik kapılar direnç, diyot, transistör, küçük değerli kondansatör, FET, MOSFET, vb. elektronik devre elemanlarından oluşmaktadır. Aynı tipte veya farklı elektronik elemanların bir paket haline getirilmesi ile oluşan yeni elemana, '**entegre**' (Integrated Circuit : IC) adı verilir.

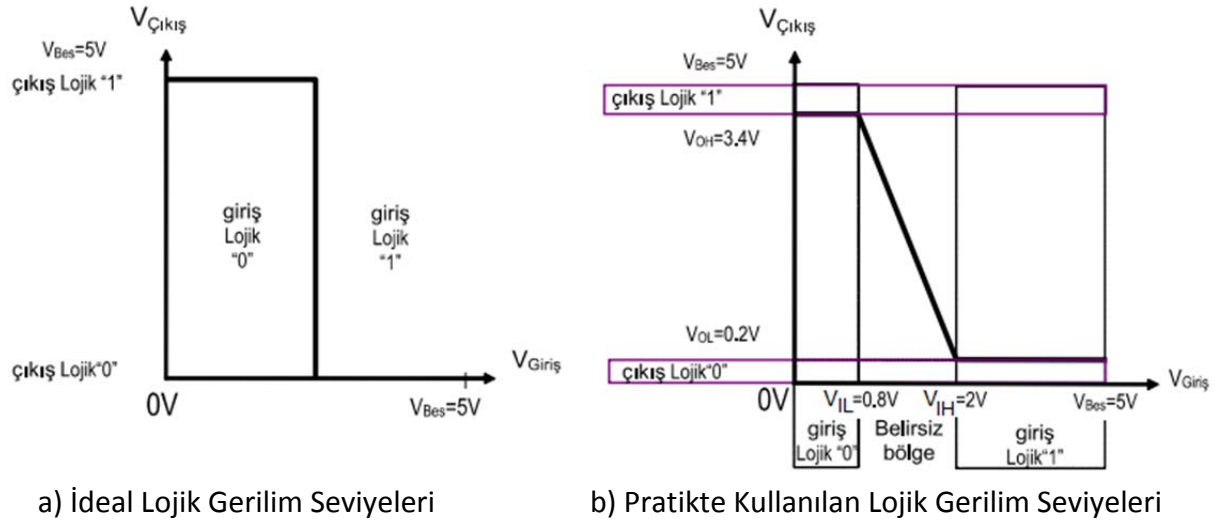
Entegreler imal edildikleri elemanlara göre;

- i) Diyot Lojik Ailesi (DL : Diode Logic Family)
- ii) Direnç-Transistör Lojik Ailesi(RTL : Resistor-Transistor Logic Family)
- iii) Diyot-Transistör Lojik Ailesi (DTL : Diode-Transistor Logic Family)
- iv) **Transistör-Transistör Lojik Ailesi** (TTL : Transistor-Transistor Logic Family)
- v) Emiter Kuplajlı Lojik Ailesi (ECL : Emitter Coupled Logic Family)
- vi) PMOS Lojik Ailesi
- vii) NMOS Lojik Ailesi
- viii) **CMOS Lojik Ailesi** (CMOS : Complementary MOS Family)

isimlerini almaktadırlar. Bu lojik ailelerinden TTL ve CMOS ailesi en çok kullanılan lojik kapı entegreleridir.

9.1 Lojik Gerilim Seviyeleri

Lojik gerilim seviyeleri; Lojik '0' veya Lojik '1' lojik sinyallerinin seviyelerini temsil eden gerilim değerleridir. Aşağıda ideal ve pratikte kullanılan lojik gerilim seviyeleri görülmektedir.



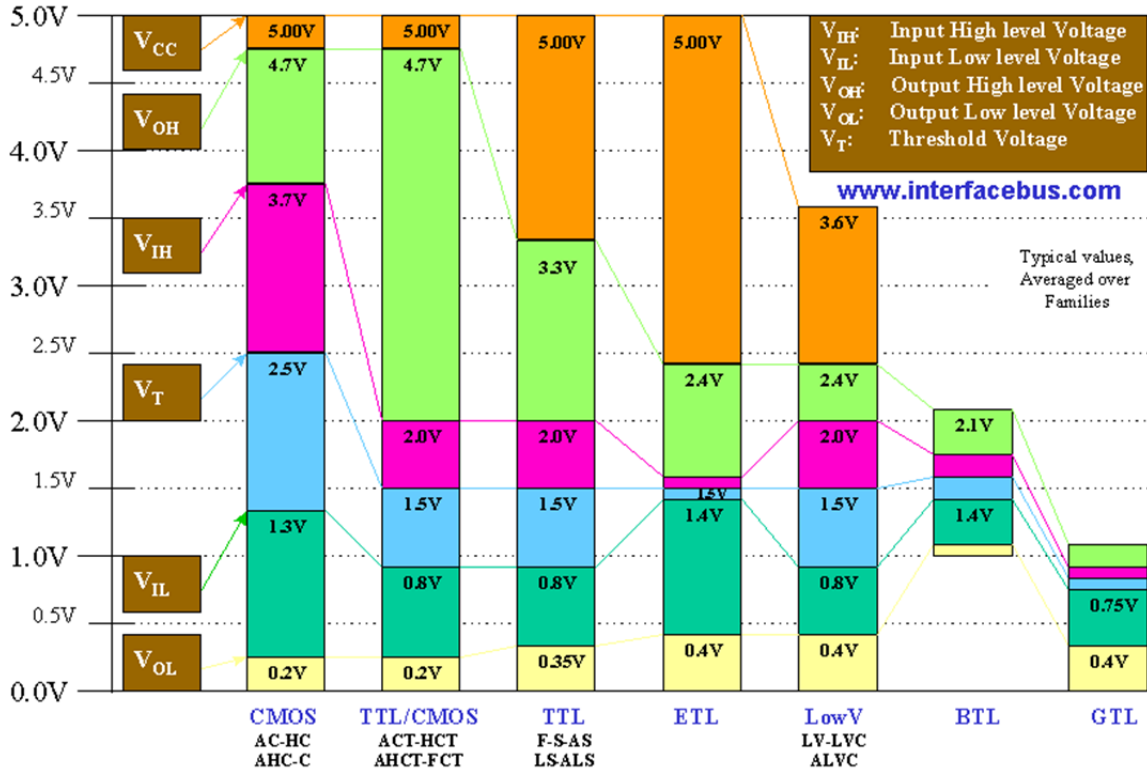
Lojik '1' seviyesi 'H: High', lojik '0' seviyesi 'L: Low', giriş (I: Input) ve çıkış (O: Output) ile temsil edilmektedir.

V_{IL} (V input low) : Girişin lojik '0' olarak algılanması için uygulanabilecek en yüksek gerilim değeridir.

V_{IH} (V input high) : Girişin lojik '1' olarak algılanması için uygulanabilecek en düşük gerilim değeridir.

V_{OL} (V output low) : Çıkışın lojik '0' olarak algılanması için uygulanabilecek en yüksek gerilim değeridir.

V_{OH} (V output high) : Çıkışın Lojik '1' olarak algılanması için uygulanabilecek en düşük gerilim değeridir.



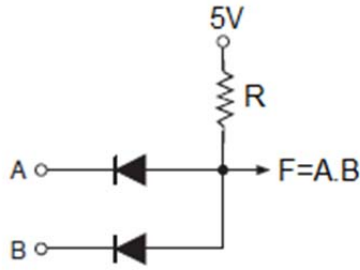
9.2 Lojik Kapıların Yarıiletken Elemanlar İle Elde Edilmesi İlgili Örnek Devreler

Örnek : Şekildeki DL (Diode Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapaıya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}

Yarıiletken elemanlar ile elde edilen kapı devreleri çözülürken ;

- Verilen devrede belirtilen giriş sayısına bağlı olarak bir doğruluk tablosu oluşturulur.
- Doğruluk tablosundaki her durum için girişler devreye uygulanır ve bu durumdaki çıkışa ait gerilim değeri hesaplanır.
- Doğruluk tablosundaki girişler devreye uygulanırken girişin lojik değerine (lojik '0' veya lojik '1') karşılık gelen gerilim değeri devreye verilmelidir.
- Devrede bulunan diyot elemanlarının eşik gerilimleri 0.7V, transistör elemanlarının base devresindeki diyot eşik gerilimi 0.7V ve doyum bölgesindeki kollektör-emiter gerilimi 0.2V alınmalıdır.
- Çıkışa ait hesaplanan gerilim değerlerine göre çıkışın lojik değeri (lojik '0' veya lojik '1') belirlenir.

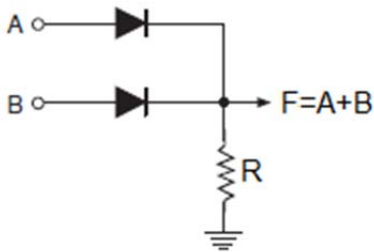
vi) Çıkışa ait lojik değerlerine göre verilen devrenin hangi lojik kapiya ait olduđu tespit edilir.



Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

A ve B girişlerine aynı anda lojik '1' uygulandığında F noktasının (çıkış) gerilimi 5 volt olmaktadır. Çünkü hem D1 diyotu ve hem de D2 diyotunun anot ve katodu arasında diyotu iletme geçirecek gerekli gerilim oluşmadığından diyotlar kesimde olurlar. Dolayısıyla doğruluk tablosuna lojik '1' yazılır. Doğruluk tablosundaki diğer durumlarda F noktasının gerilimi $V_F = 0.7$ volt olmaktadır. Bu durumlarda çıkış noktasının gerilim seviyeleri 0.7V olduğu için doğruluk tablosuna lojik '0' yazılır. Doğruluk tablosu incelendiğinde devrenin bir VE (AND) kapiı gibi çalıştığı görülmektedir.

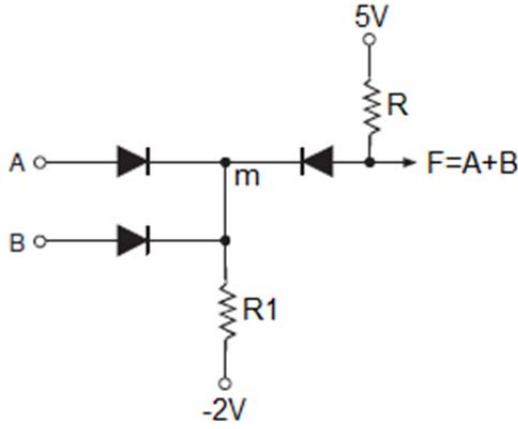
Örnek : Şekildeki DL (Diode Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

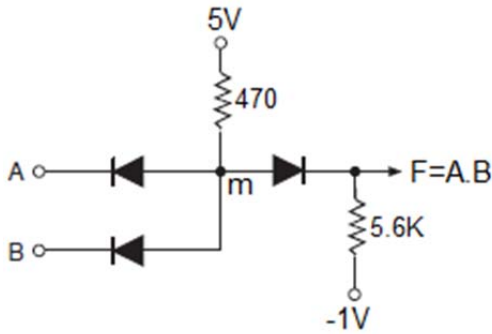
A ve B girişlerine aynı anda lojik '0' uygulandığında F noktasının (çıkış) gerilimi 0 volt olmaktadır. Çünkü hem D1 diyotu ve hem de D2 diyotunun anot ve katodu arasında diyotu iletme geçirecek gerekli gerilim oluşmadığından diyotlar kesimde olurlar. Dolayısıyla doğruluk tablosuna lojik '0' yazılır. Doğruluk tablosundaki diğer durumlarda F noktasının gerilimi $V_F = 5 - 0.7 = 4.3$ volt olmaktadır. Bu durumlarda çıkış noktasının gerilim seviyeleri 4.3V olduğu için doğruluk tablosuna lojik '1' yazılır. Doğruluk tablosu incelendiğinde devrenin bir VEYA (OR) kapiı gibi çalıştığı görülmektedir.

Örnek : Şekildeki DL (Diode Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.} {Geliştirilmiş OR Kapısı}



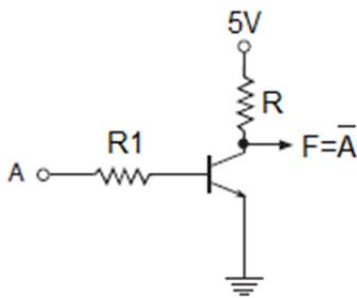
Giriş		Çıkış		
A	B	m (V)	F (V)	F
0	0			
0	1			
1	0			
1	1			

Örnek : Şekildeki DL (Diode Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.} {Geliştirilmiş AND Kapısı}



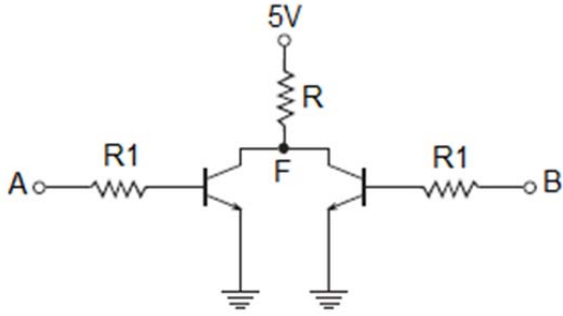
Giriş		Çıkış		
A	B	m (V)	F (V)	F
0	0			
0	1			
1	0			
1	1			

Örnek : Şekildeki RTL (Resistor-Transistor Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



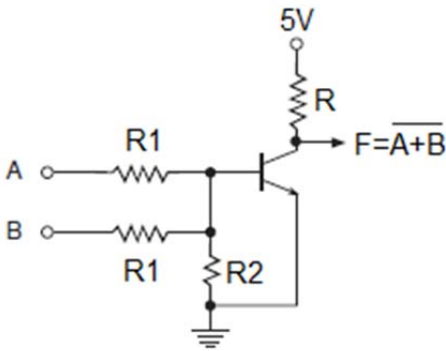
Giriş		Çıkış	
A	F (V)	F	
0			
1			

Örnek : Şekildeki RTL (Resistor-Transistor Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



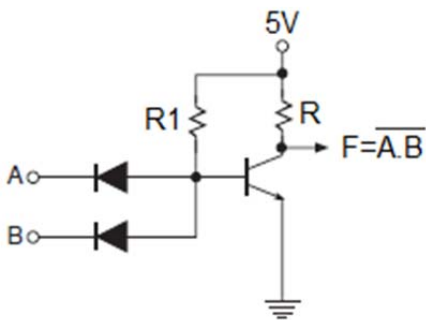
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki RTL (Resistor-Transistor Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



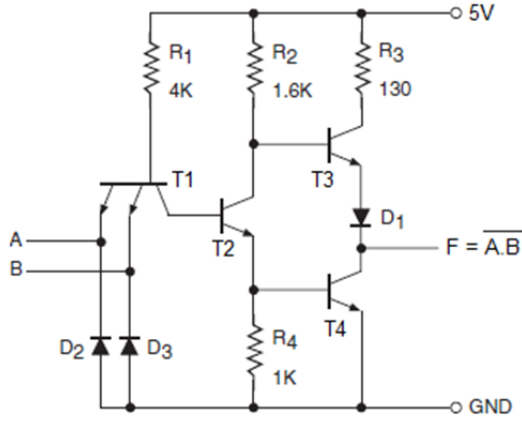
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki DTL (Diode-Transistor Logic) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? NOT: Devrede bulunan diyotlar GE diyottur. {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



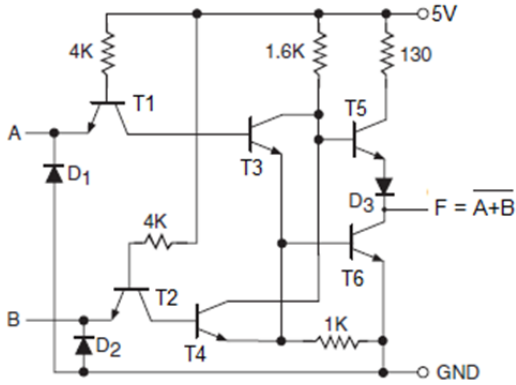
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki TTL (Transistor-Transistor Logic – Çok (Multi) Emiter Girişli Transistör) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



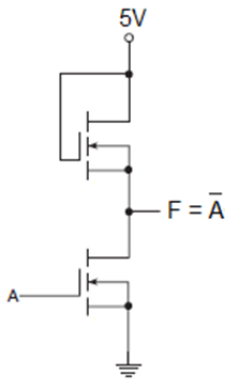
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki TTL (Transistor-Transistor Logic – Çok (Multi) Emiter Girişli Transistör) devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



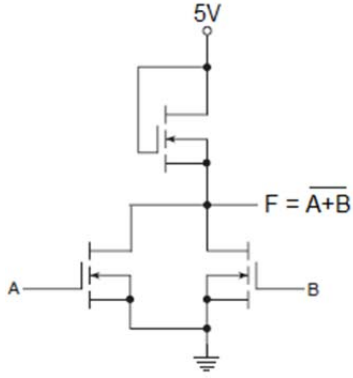
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki NMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



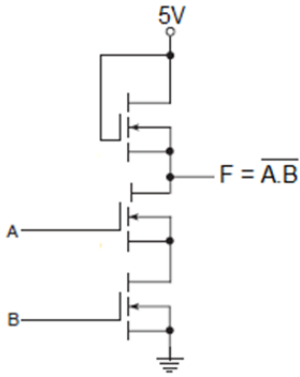
Giriş	Çıkış	
A	F (V)	F
0		
1		

Örnek : Şekildeki NMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



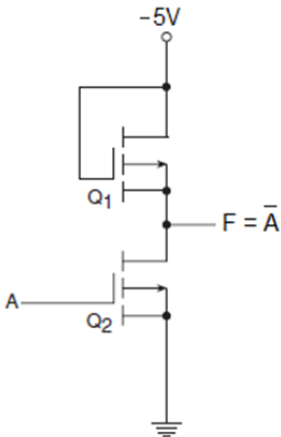
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki NMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



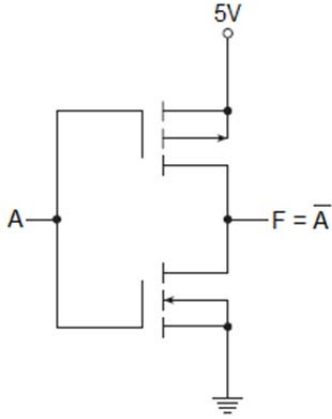
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki PMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv -5V, lojik '1' \equiv 0V kabul edilmiştir.}



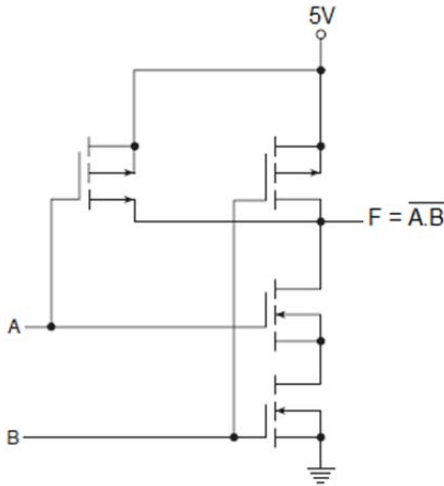
Giriş		Çıkış	
A		F (V)	F
0			
1			

Örnek : Şekildeki CMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



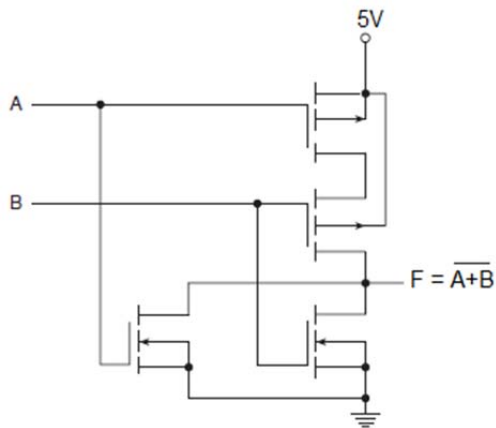
Giriş		Çıkış	
A	F (V)	F	
0			
1			

Örnek : Şekildeki CMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



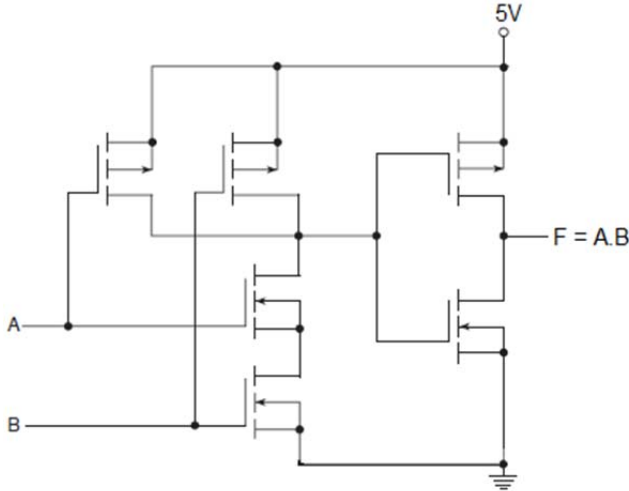
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki CMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



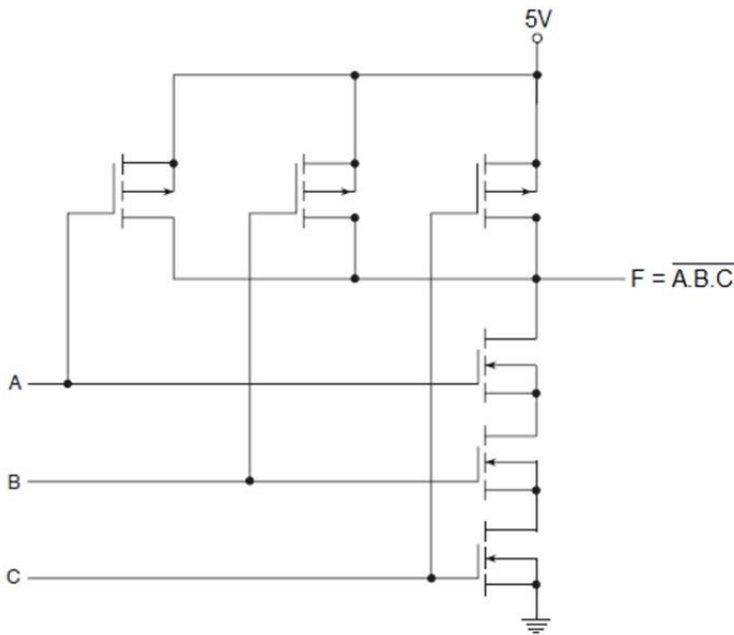
Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki CMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



Giriş		Çıkış	
A	B	F (V)	F
0	0		
0	1		
1	0		
1	1		

Örnek : Şekildeki CMOS devresine ait doğruluk tablosunu doldurarak devrenin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}



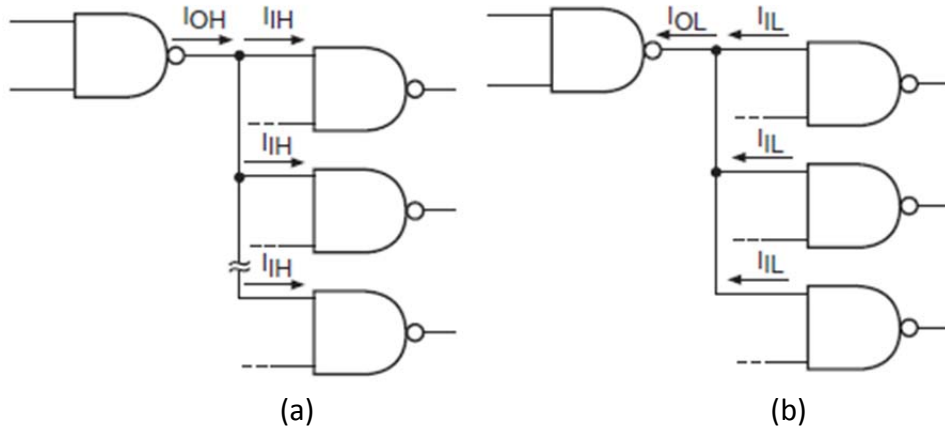
Giriş			Çıkış	
A	B	C	F (V)	F
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Fan-In : Fan-In teriminin literatürde iki farklı tanımı bulunmaktadır. Bu tanımlar;

i) Bir lojik kapının giriş sayısı. (Fan-In is the number of inputs of an electronic logic Gates)

ii) Bir tek lojik kapının kabul edeceği maksimum giriş sayısı. (Fan-In is a term that defines the maximum number of digital inputs that single logic gate can accept. Most TTL Gates have one or two inputs although some have more than two. A typical logic gate has a Fan-In of 1 or 2.)

Fan out : Bir lojik kapı çıkışının besleyebileceği lojik giriş sayısı. (The number of logic gate inputs that can be driven from the output of a single logic gate without causing any false output is called fan-out.)



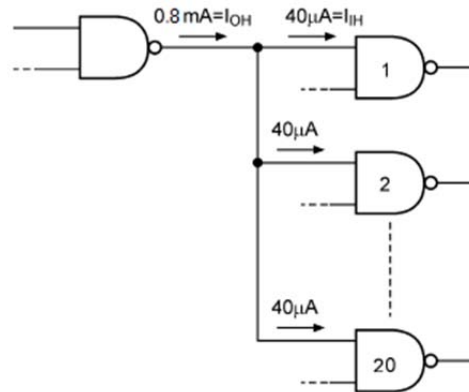
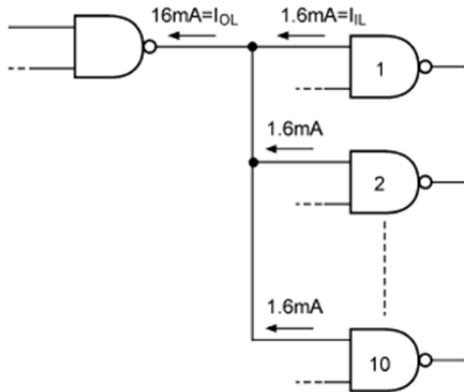
a) TTL lojik kapılarında; lojik '1' durumundaki bir kapının girişi bağlı olduğu bir önceki kapının çıkışından $40\mu\text{A}$ akım çeker.

b) TTL lojik kapılarında; lojik '0' durumundaki bir kapının girişi bağlı olduğu bir önceki kapıya 1.6mA akım verir.

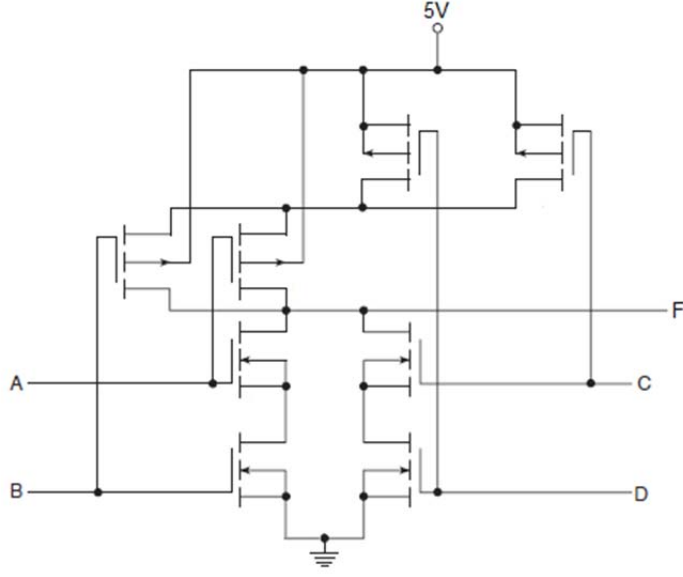
Örnek : 7400 lojik kapısı çıkışında lojik '0' durumunda 16mA kapıdan içeri akmaktadır. Lojik '1' durumunda ise $800\mu\text{A}$ akar. Bu kapının Fan-Out değeri kaçtır?

Lojik '0' için Fan-Out = $16\text{mA} / 1.6\text{mA} = 10$

Lojik '1' için Fan-Out = $800\mu\text{A} / 40\mu\text{A} = 20$



ÖDEV : Şekildeki CMOS devresinin hangi lojik kapiya ait olduğunu belirleyiniz? {Devrede; lojik '0' \equiv 0V, lojik '1' \equiv 5V kabul edilmiştir.}

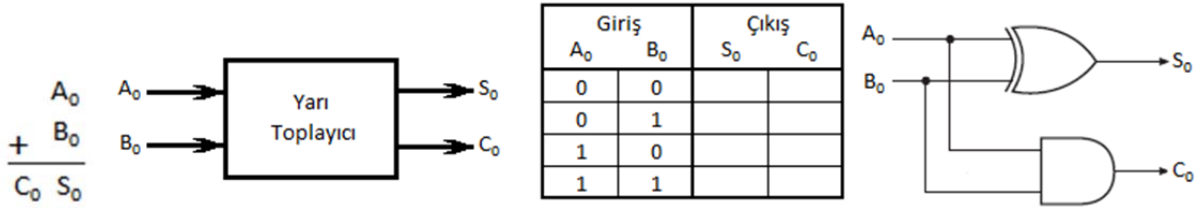


10. TOPLAYICI DEVRELER (ADDERS)

İki tabanlı sayı sisteminde, iki sayının toplamı seri veya paralel toplayıcılar ile gerçekleştirilebilir. Paralel toplayıcı devrelerinde bütün bitler aynı anda uygulanır ve sonucun çıkışa yansması bütün bitlerde aynı anda gerçekleşir. Seri toplayıcıda ise, iki tabanlı sayı sistemindeki sayıların önce en az ağırlıklı bitleri devrenin girişine gelir ve bunların toplamı olan bit çıkışa yansıtılırken, bir sonraki ağırlıklı bitlere eklenecek olan elde ise bir başka çıkış olarak görülür. Bu nedenle seri toplama yapabilmek için, her bir bite ilişkin elde bitinin bir yerde saklanarak kendinden sonraki bitlerin toplamına eklenmesi gerekir. Bu saklama işlemi bellek özelliği olan tutucu (latch) devreler ile yapılır. Bu sebeple seri toplayıcılar ardışıl devre olarak gerçekleştirilebilir.

Toplayıcı devreleri aynı zamanda yarı toplayıcı (half adder) ve tam toplayıcı (full adder) olmak üzere ikiye ayrılırlar.

i) Yarı Toplayıcı devreler iki farklı iki tabanlı sayıyı toplar ve toplama sonucunda herhangi bir bitte oluşmuş olan elde'yi bir sonraki bite aktarmaz. Bundan dolayı toplama işlemi eksik yapılmış olur. Başka bir ifade ile eldesiz toplama yarı toplayıcı ile gerçekleştirilir. Devrenin iki girişi (A_0 , B_0) ve iki çıkışı (S_0 , C_0) vardır.



Doğruluk tablosundan yararlanarak S_0 ve C_0 lojik ifadeleri aşağıdaki gibi yazılır.

$$S_0 = \overline{A_0} \cdot B_0 + A_0 \cdot \overline{B_0} = A_0 \oplus B_0 \qquad C_0 = A_0 \cdot B_0$$

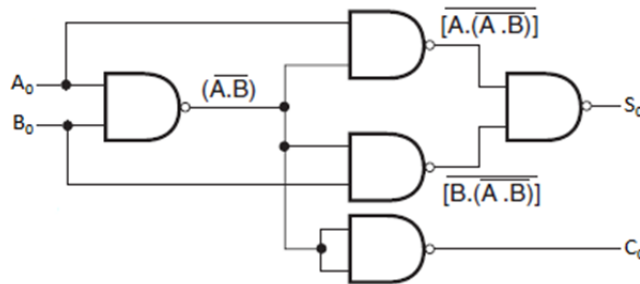
Örnek : Yarı toplayıcı devresini sadece **NAND** kapısı kullanarak ve **beş** kapıyla gerçekleştiriniz.

$$S_0 = \overline{A_0} \cdot B_0 + \overline{A_0} \cdot A_0 + A_0 \cdot \overline{B_0} + \overline{B_0} \cdot B_0 = A_0 \cdot (\overline{A_0} + \overline{B_0}) + B_0 \cdot (\overline{A_0} + \overline{B_0})$$

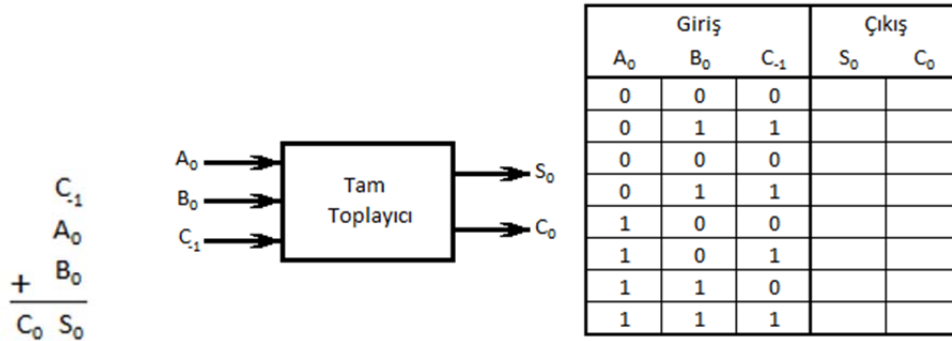
$$S_0 = A_0 \cdot \overline{A_0 \cdot B_0} + B_0 \cdot \overline{A_0 \cdot B_0}$$

$$S_0 = \overline{\overline{S_0}} = \overline{\overline{A_0 \cdot A_0 \cdot B_0} + \overline{B_0 \cdot A_0 \cdot B_0}} = \overline{\overline{A_0 \cdot A_0 \cdot B_0} \cdot \overline{B_0 \cdot A_0 \cdot B_0}}$$

$$C_0 = \overline{\overline{C_0}} = \overline{\overline{A_0 \cdot B_0}}$$



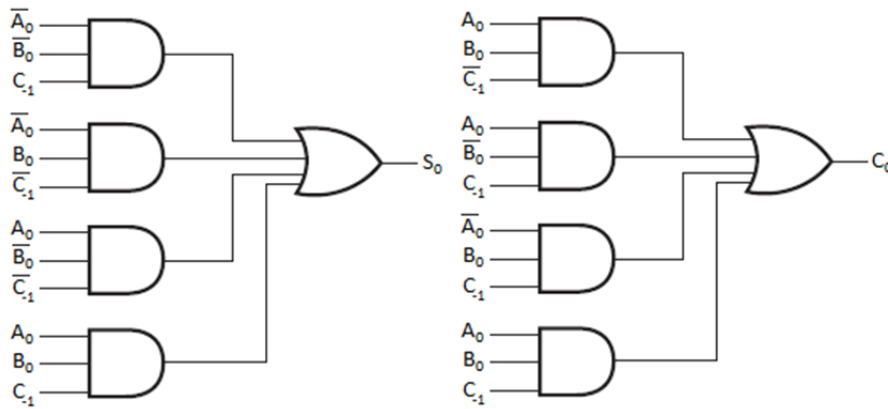
ii) **Tam toplayıcı** devrelerde ise bir önceki basamakta oluşabilecek elde işleme katılır. Bu sebeple devreye bir elde girişi eklenir. Başka bir ifade ile eldeli toplama işlemi tam toplayıcı ile gerçekleştirilir. Tam toplayıcı, üç giriş (A, B, C_{in}) ve iki çıkışı (S, C) olan bir devredir.



Doğruluk tablosundan yararlanılarak S₀ ve C₀ lojik ifadeleri aşağıdaki gibi yazılır.

$$S_0 = \overline{A_0} \cdot \overline{B_0} \cdot C_{-1} + \overline{A_0} \cdot B_0 \cdot \overline{C_{-1}} + A_0 \cdot \overline{B_0} \cdot \overline{C_{-1}} + A_0 \cdot B_0 \cdot C_{-1}$$

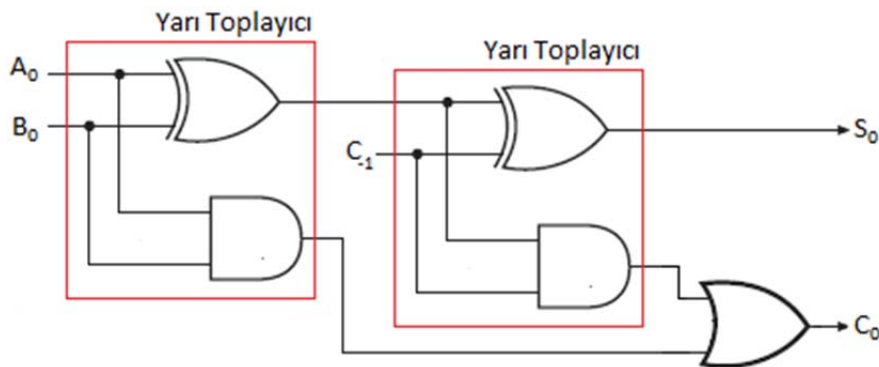
$$C_0 = A_0 \cdot B_0 \cdot \overline{C_{-1}} + A_0 \cdot \overline{B_0} \cdot C_{-1} + \overline{A_0} \cdot B_0 \cdot C_{-1} + A_0 \cdot B_0 \cdot C_{-1}$$



S₀ ve C₀ lojik ifadeleri klasik sadeleştirme yöntemleri ile sadeleştirilerek S₀ ve C₀ için aşağıdaki lojik ifadeler elde edilir.

$$S_0 = C_{-1} \cdot (\overline{A_0} \cdot \overline{B_0} + A_0 \cdot B_0) + \overline{C_{-1}} \cdot (\overline{A_0} \cdot B_0 + A_0 \cdot \overline{B_0}) = A_0 \oplus B_0 \oplus C_{-1}$$

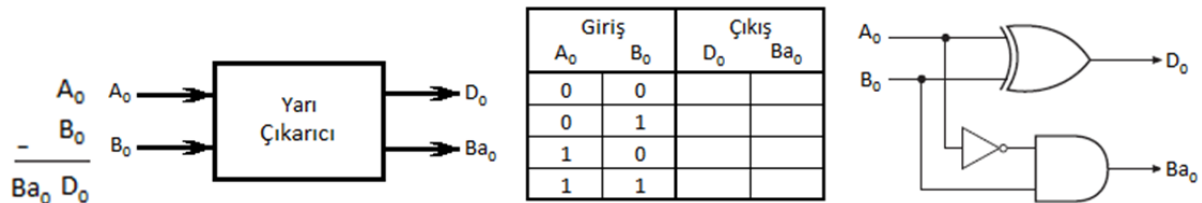
$$C_0 = A_0 \cdot B_0 \cdot (\overline{C_{-1}} + C_{-1}) + C_{-1} \cdot (A_0 \cdot \overline{B_0} + \overline{A_0} \cdot B_0) = A_0 \cdot B_0 + C_{-1} \cdot (A_0 \oplus B_0)$$



11. ÇIKARICI DEVRELER (SUBTRACTORS)

Çıkarıcı devreleri yarı çıkarıcı (half subtractor) ve tam çıkarıcı (full subtractor) olmak üzere ikiye ayrılırlar.

i) Yarı Çıkarıcı devre iki farklı iki tabanlı sayıyı çıkarır. Bir önceki basamakta oluşabilen borcu işleme katmaz. Bundan dolayı çıkarma işlemi eksik yapılır. Çıkarıcı devrenin iki girişi (A_0, B_0) ve iki çıkışı (D_0, Ba_0) vardır.

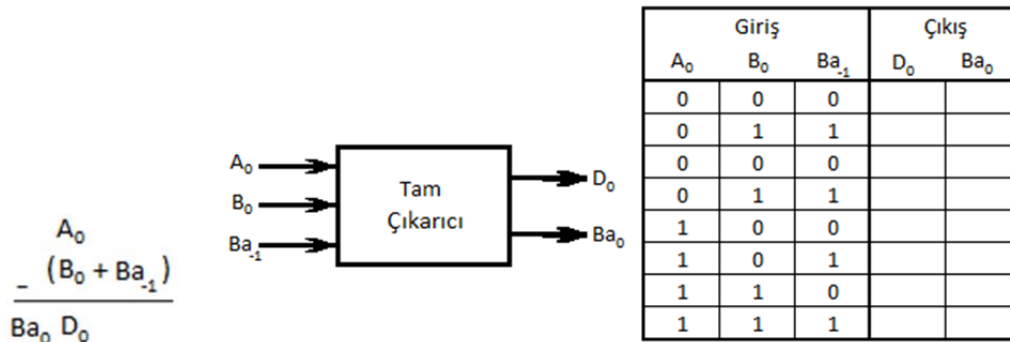


Doğruluk tablosundan yararlanarak D_0, Ba_0 lojik ifadeleri aşağıdaki gibi yazılır.

$$D_0 = \overline{A_0} \cdot B_0 + A_0 \cdot \overline{B_0} = A_0 \oplus B_0$$

$$Ba_0 = \overline{A_0} \cdot B_0$$

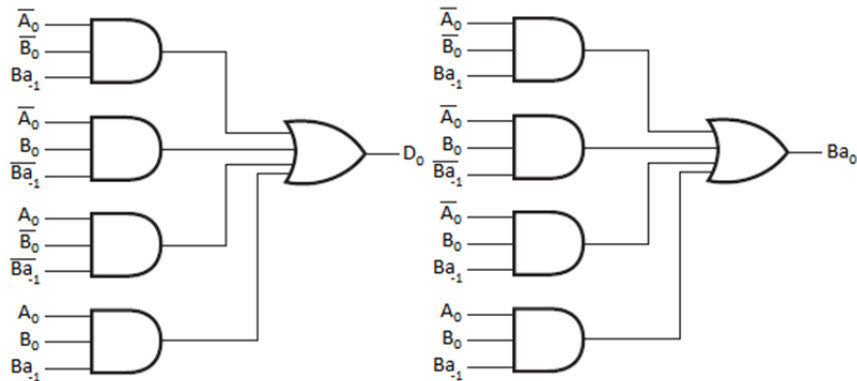
ii) Tam çıkarıcı devrede ise bir önceki basamakta oluşabilecek borç da işleme katılır. Bu sebeple devreye bir borç girişi eklenir. Tam çıkarıcı, üç girişi (A_0, B_0, Ba_{-1}) ve iki çıkışı (D_0, Ba_0) olan devredir.



Doğruluk tablosundan yararlanarak D_0, Ba_0 lojik ifadeleri aşağıdaki gibi yazılır.

$$D_0 = \overline{A_0} \cdot \overline{B_0} \cdot Ba_{-1} + \overline{A_0} \cdot B_0 \cdot \overline{Ba_{-1}} + A_0 \cdot \overline{B_0} \cdot \overline{Ba_{-1}} + A_0 \cdot B_0 \cdot Ba_{-1}$$

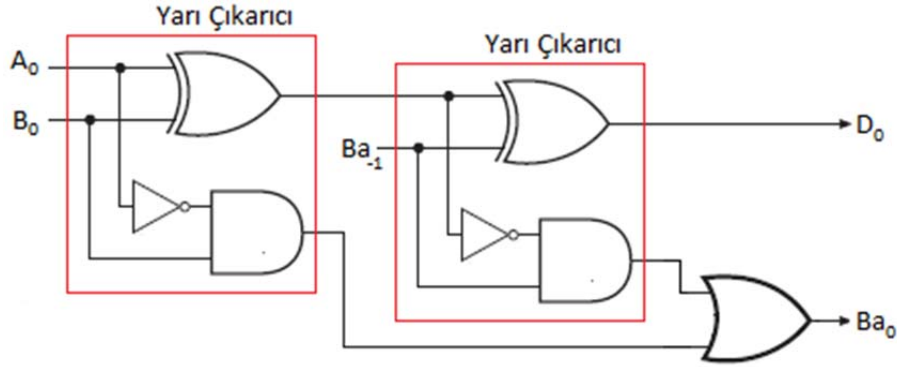
$$Ba_0 = \overline{A_0} \cdot \overline{B_0} \cdot Ba_{-1} + \overline{A_0} \cdot B_0 \cdot \overline{Ba_{-1}} + \overline{A_0} \cdot B_0 \cdot Ba_{-1} + A_0 \cdot B_0 \cdot \overline{Ba_{-1}}$$



D_0, Ba_0 lojik ifadeleri klasik sadeleştirme yöntemleri ile sadeleştirilerek D_0, Ba_0 için aşağıdaki lojik ifadeler elde edilir.

$$D_0 = Ba_{-1} \cdot (\overline{A_0} \cdot \overline{B_0} + A_0 \cdot B_0) + \overline{Ba_{-1}} \cdot (\overline{A_0} \cdot B_0 + A_0 \cdot \overline{B_0}) = A_0 \oplus B_0 \oplus Ba_{-1}$$

$$Ba_0 = \overline{A_0} \cdot B_0 \cdot (\overline{Ba_{-1}} + Ba_{-1}) + Ba_{-1} \cdot (\overline{A_0} \cdot \overline{B_0} + A_0 \cdot B_0) = \overline{A_0} \cdot B_0 + Ba_{-1} \cdot (\overline{A_0} \oplus \overline{B_0})$$

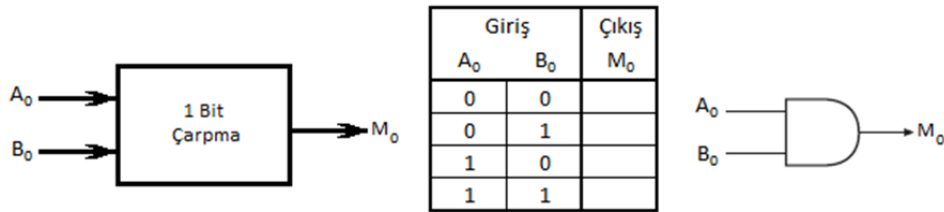


12. ÇARPMA DEVRELERİ (MULTIPLIERS)

Çarpma devreleri 1bitlik (1x1), 2 bitlik (2x2), n bitlik (nxn) gibi bit sayısı ile ifade edilmektedir. Örnek olması açısından 1 ve 2 bitlik çarpma devreleri incelenmiştir.

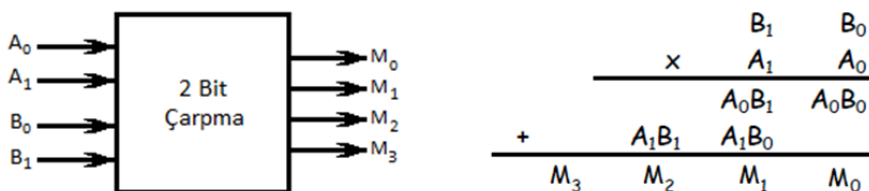
i) 1 Bit Çarpma Devresi (1x1 Multiplier)

Bir bitlik çarpma devresinin doğruluk tablosu incelendiği zaman, 1 bitlik çarpma işleminin aslında bir VE (AND) kapısı olduğu görülmektedir.

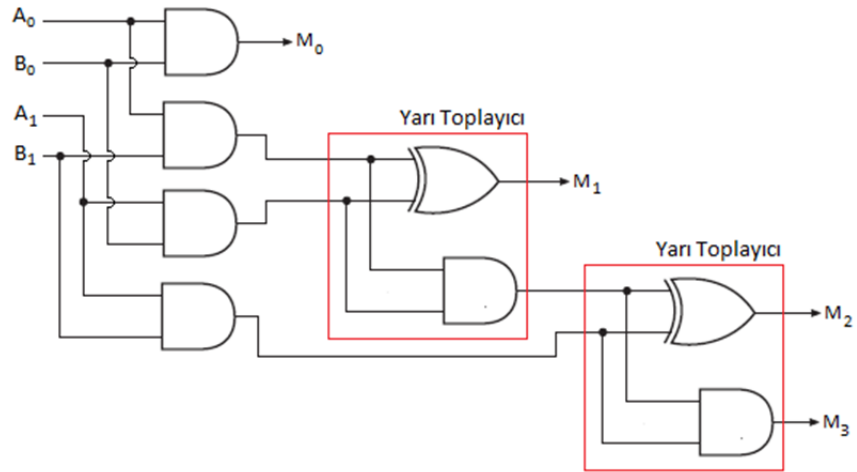


ii) 2 Bit Çarpma Devresi (2x2 Multiplier)

2 Bit çarpma devresi için de doğruluk tablosu oluşturulup daha sonra doğruluk tablosundan çarpma devresine ait lojik ifadeler elde edilebilirdi. Ancak aşağıda görüldüğü gibi matematiksel bir çarpma işlemi yapıyor gibi işlemler yapılmış ve gerekli formüller elde edilmiştir. Daha sonra elde edilen bu formüller kullanılarak sonuca gidilmiştir.

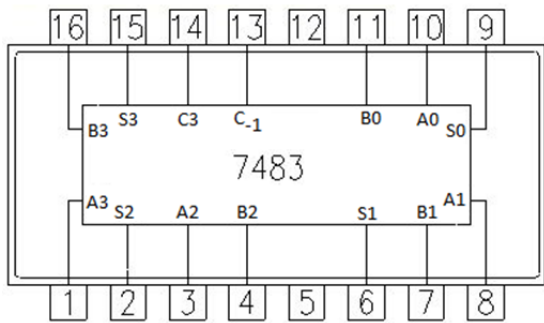


Bu yöntem kullanılarak (3x3), (4x4) .. (nxn) gibi çarpma devreleri kolaylıkla elde edilebilir.

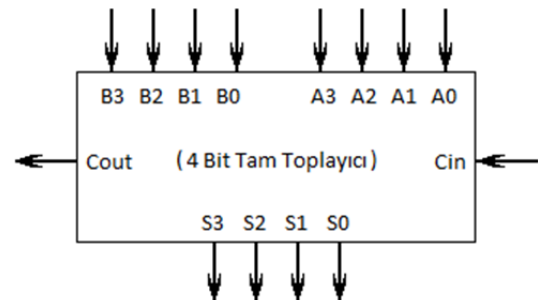


Örnek : 1 bit tam toplayıcı kullanarak 2 bit tam toplayıcı devre elde ediniz?

Örnek : 1 bit tam toplayıcı kullanarak 4 bit paralel tam toplayıcı devre elde ediniz?



Şekil A. 7483 – 4 Bit Tam Toplayıcı Devresi



Şekil B. 4 Bit Tam Toplayıcı Blok Şeması

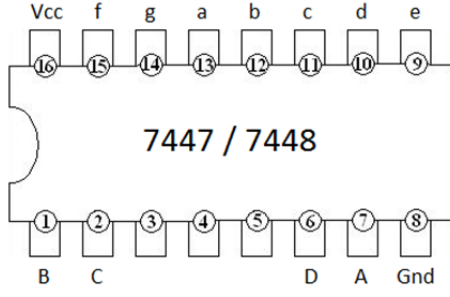
Örnek : 4 bit tam toplayıcı kullanarak 8 bit tam toplayıcı devre elde ediniz?

Örnek : 1 bit tam çıkarıcı kullanarak 2 bit tam çıkarıcı devre elde ediniz?

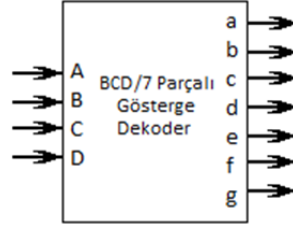
Örnek : 1 bit tam çıkarıcı kullanarak 4 bit tam çıkarıcı devre elde ediniz?

Örnek : Sadece VE (AND) kapısı ve ÖZEL-VEYA (EX-OR) kapısı kullanarak 1 Bit yarım toplama ve çıkarma işlemi yapabilen devreyi tasarlayınız?

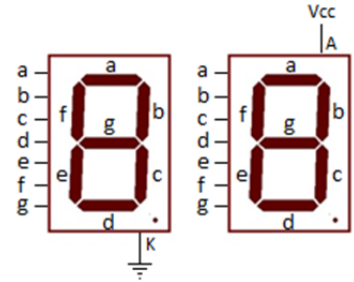
Örnek : 1 Basamaklı (1-Dijit) on tabanlı iki sayının, BCD kodlarını (karşılıklarını) toplayabilen devreyi tasarlayınız? Toplam sonucu 7-Parçalı gösterge üzerinde gözlenecektir.



Şekil A. 7447/7448 Dekoder Entegre



Şekil B. Dekoder Blok Şeması



Şekil C. Ortak Katot/Ortak Anot 7 Parçalı Gösterge

Örnek : 4 Bit Tam toplayıcı kullanarak, 4 Bitlik iki sayıyı toplayabilen ve çıkarabilen devreyi tasarlayınız?

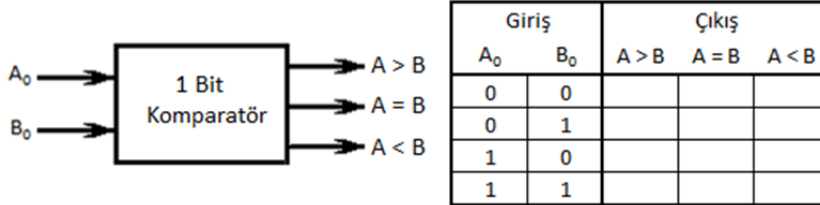
Örnek : 2 Basamaklı (2-Dijit) on tabanlı iki sayının, BCD kodlarını (karşılıklarını) toplayabilen devreyi tasarlayınız?



13. SAYISAL KOMPARATÖR (KARŞILAŞTIRICI) DEVRELERİ (COMPARATORS)

Sayısal (Dijital) karşılaştırıcı devreler de 1 bitlik komparatör, 2 bitlik komparatör, n bitlik komparatör gibi bit sayısı ile ifade edilmektedir. Sayısal karşılaştırıcı devreler iki sayıyı (A,B) karşılaştırıp, karşılaştırma sonucunda (A>B), (A=B) ve (A<B) olmak üzere 3 çıkış verir.

13.1. 1 Bitlik Komparatör

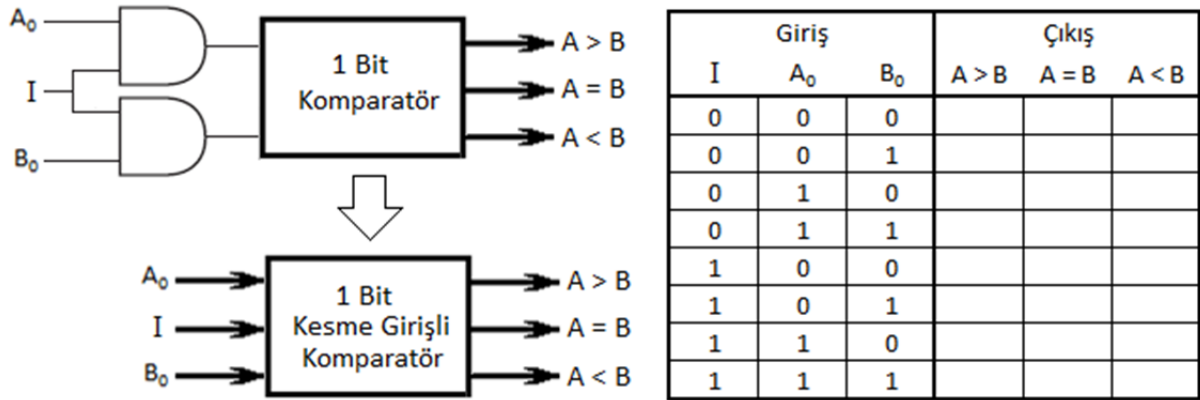


$f(A > B) = ?$

$f(A = B) = ?$

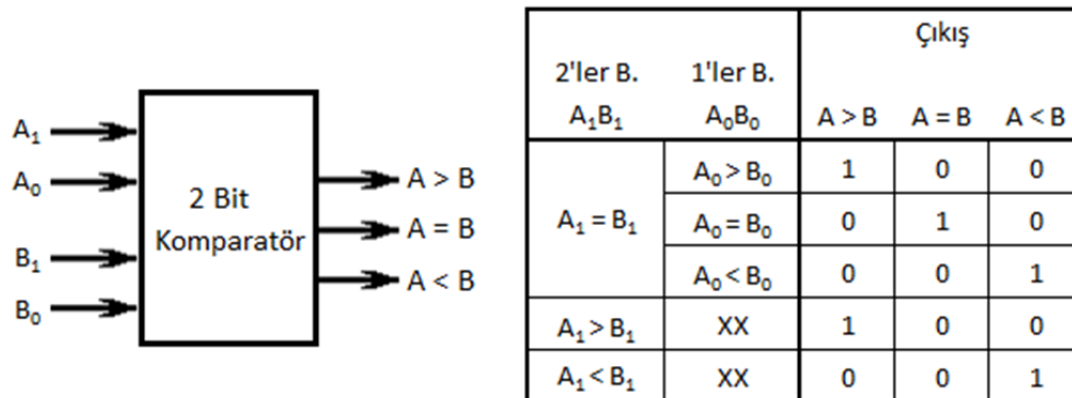
$f(A < B) = ?$

13.2. 1 Bitlik Kesme Girişli Komparatör



13.3. 2 Bitlik Komparatör

2 Bitlik Komparatör devresi için doğruluk tablosu oluşturulup daha sonra doğruluk tablosundan (A>B), (A=B) ve (A<B) çıkışlarına ait lojik ifadeler elde edilebilirdi. Ancak 1 Bitlik kesme girişli komparatör kullanarak da 2, 3,.. n bitlik komparatör elde edilebilir.



14. KOD ÇÖZÜCÜ DEVRELER (DECODERS)

Kod çözücü devreler, her bir giriş değerine karşılık bir çıkış elde etmek için kullanılır. Daha önce yapılmış olan şifrelemeyi çözen devredir. Kod çözücü devrenin girişinde n adet giriş var ise, devrenin 2^n adet çıkışı olur. Kod çözücü devreler "Lojik 0" aktif kod çözücü ve "Lojik 1" aktif kod çözücü olmak üzere ikiye ayrılırlar. Kod çözücü devreler giriş sayılarına bağlı olarak 1×2 , 2×4 , 3×8 , ..., $n \times 2^n$ (Decoder) kod çözücü olarak isimlendirilirler. Bazı kod çözücü devrelerde, kod çözücü devreyi aktif/pasif hale getirecek Yetkilendirme (ENABLE) girişi bulunabilir. Kod çözücü devrelerde bulunan Yetkilendirme (ENABLE) girişleri "Lojik 0" veya "Lojik 1" aktif olabilir.

"Lojik 0" aktif kod çözücü devrede, her giriş durumuna karşılık sadece bir çıkış "Lojik 0" olur. Diğer çıkışlar "Lojik 1" seviyede çıkış verirler.

"Lojik 1" aktif kod çözücü devrede ise, her giriş durumuna karşılık sadece bir çıkış "Lojik 1" olur. Diğer çıkışlar "Lojik 0" seviyede çıkış verirler.

Örnek : 2 girişli "Lojik 0" aktif kod çözücü devre tasarlayınız?

Giriş		Çıkış			
B	A	Q0	Q1	Q2	Q3
0	0				
0	1				
1	0				
1	1				

f(Q0)= ?

f(Q1)= ?

f(Q2)= ?

f(Q3)= ?

Örnek : 2 girişli “Lojik 1” aktif kod çözücü devre tasarlayınız? NOT: “Lojik 1” aktif Yetkilendirme (ENABLE) girişi olmalıdır.

Giriş			Çıkış			
E	B	A	Q0	Q1	Q2	Q3
0	X	X				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

f(Q0)= ?

f(Q1)= ?

f(Q2)= ?

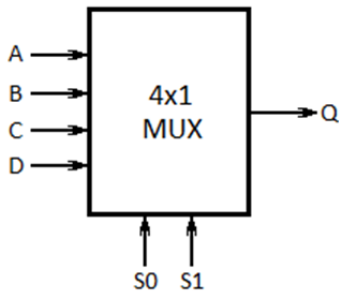
f(Q3)= ?

15. AZALTICI / VERİ SEÇİCİ DEVRELER (MULTIPLEXERS / DATA SELECTORS)

Çeşitli sayıda mevcut olan giriş verilerinden, istenilen veriyi seçip tek bir çıkış üzerinden dışarıya veya başka bir verinin girişine yansıtan devrelere azaltıcı / veri seçici (multiplexer / data selector) adı verilir. n adet seçici sinyal sayısı olan veri seçici devrenin 2^n adet girişi vardır. Dolayısıyla veri seçici devrelerin (multiplexer) n adet seçici girişi, 2^n adet girişi ve 1 adet çıkışı bulunur. Veri seçici, devreler giriş verilerini zamana bağlı olarak sırayla çıkışa yansıtabilmektedirler. Multiplexer devreler seçici sinyal sayısına bağlı olarak 2x1 MUX, 4x1 MUX, 8x1 MUX, ..., 2^n x1 MUX şeklinde isimlendirilirler.

Multiplexer devrelerinde, devreyi aktif/pasif hale getirecek “Lojik 0” veya “Lojik 1” aktif Yetkilendirme (ENABLE) girişi bulunabilir.

Örnek : 2 adet seçici sinyal girişi olan multiplexer devresi tasarlayınız?



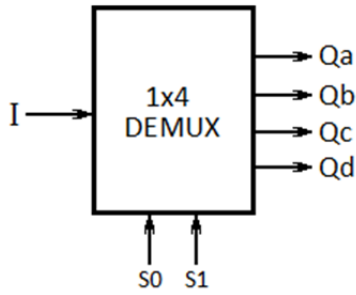
f(Q)= ?

16. ÇOĞULLAYICI / VERİ DAĞITICI DEVRELER (DEMULTIPLEXERS / DATA DISTRIBUTORS)

Devrenin girişine gelen veriyi, seçici sinyalin durumuna göre ilgili çıkışa gönderen devrelere **Demultiplexer** adı verilir. Demultiplexer devrelerinde, n adet seçici girişi, 2^n adet çıkış ve 1 adet giriş bulunur. Demultiplexer devreler seçici sinyal sayısına bağlı olarak 1x2 DEMUX, 1x4 DEMUX, 1x8 DEMUX, , 1×2^n DEMUX şeklinde isimlendirilirler.

Demultiplexer devrelerinde, devreyi aktif/pasif hale getirecek "Lojik 0" veya "Lojik 1" aktif Yetkilendirme (ENABLE) girişi bulunabilir.

Örnek : 2 adet seçici sinyal girişi olan demultiplexer devresi tasarlayınız?



$$f(Qa) = ?$$

$$f(Qb) = ?$$

$$f(Qc) = ?$$

$$f(Qd) = ?$$

17. KODLAYICI DEVRELER (ENCODERS)

Kodlayıcı devreler (Encoders), adından da anlaşılacağı gibi verileri kodlamak amacıyla kullanılır. Kod çözücü devrelerin (Decoders) tersi işlem yaparlar. Kodlayıcı devrenin girişinde 2^n adet giriş var ise, devrenin n adet çıkışı olur. Kodlayıcı devreler "Lojik 0" aktif kodlayıcı ve "Lojik 1" aktif kodlayıcı olmak üzere ikiye ayrılırlar. Kodlayıcı devreler giriş sayılarına bağlı olarak 2x1, 4x2, 8x3,, $2^n \times n$ (Encoder) kodlayıcı olarak isimlendirilirler. Bazı kodlayıcı devrelerde, kodlayıcı devreyi aktif/pasif hale getirecek Yetkilendirme (ENABLE) girişi bulunabilir. Kodlayıcı devrelerde bulunan Yetkilendirme (ENABLE) girişleri "Lojik 0" veya "Lojik 1" aktif olabilir.

Örnek : 4 girişli "Lojik 1" aktif kodlayıcı devre tasarlayınız?

Giriş				Çıkış	
I0	I1	I2	I3	A0	A1
X	X	X	1		
X	X	1	0		
X	1	0	0		
1	0	0	0		

$$f(A0) = ?$$

$$f(A1) = ?$$

Örnek : 4 girişli “Lojik 1” aktif kodlayıcı devre tasarlayınız? NOT: “Lojik 1” aktif Yetkilendirme (ENABLE) girişi olmalıdır.

E	Giriş				Çıkış	
	I0	I1	I2	I3	A0	A1
0	X	X	X	X		
1	X	X	X	1		
1	X	X	1	0		
1	X	1	0	0		
1	1	0	0	0		

Örnek : 1 adet kod çözücü (decoder) ve 2 adet VEYA (OR) kapısı kullanarak 1 bitlik tam toplayıcı tasarlayınız?

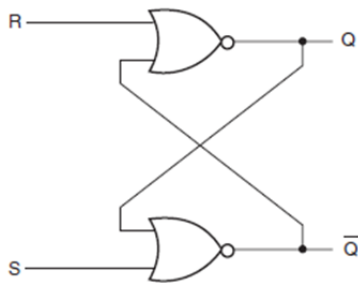
Örnek : 3x8 kod çözücü (decoder) devresi kullanarak 4x16 kod çözücü (decoder) tasarlayınız?

18. FLİP FLOP'LAR (FLIP FLOPS)

Flip Flop'lar iki kararlı elektriksel duruma sahip olan elektronik devrelerdir. Devrenin girişlerine uygulanan işarete göre çıkış bir kararlı durumdan diğer (ikinci) kararlı duruma geçer. Dijital (sayısal) elektronik devrelerinde bu durumlar lojik '0' ve lojik '1' olarak ifade edilmiştir. Standart lojik kapıların kendi başlarına veri saklama yeteneği yoktur. Birkaç lojik kapının birbirine bellek özelliği olacak şekilde bağlanmasıyla bir flip flop yani bilgi saklama ve depolama devresi elde edilebilir. Lojik kapılardan oluşturulan flip flop'lar dijital elektronikte en çok kullanılan bellek elemanlarıdır. Giriş sayısına, girişlerinin isimlerine ve tetikleme işaretlerinin uygulanış biçimlerine göre flip flop'lar adlandırılmaktadır. Ancak tüm flip flop'ların bir düz (Q) ve bir de evrik (\bar{Q}) olmak üzere iki çıkışı bulunmaktadır.

18.1. RS Flip Flop

a) VEYA-DEĞİL (NOR) Kapısıyla Elde Edilmiş RS Flip Flop;

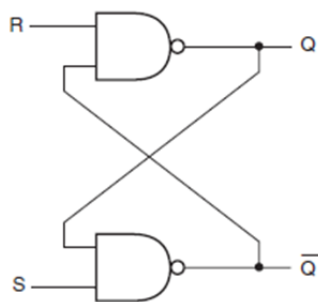


Giriş		Çıkış
R	S	Q_{n+1}
0	0	Değişmez / Depolama
0	1	1
1	0	0
1	1	Yasak Durum

Giriş		NOR
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

VEYA-DEĞİL (NOR) kapısıyla elde edilmiş flip-flop devresinde R (Sil / Reset) ve S (Kur / Set) girişine aynı anda lojik '1' uygulandığından flip flop devresinin iki çıkışı da (Q, \bar{Q}) aynı anda lojik '0' olmaktadır. Bilindiği gibi flip-flop'ların birbirinin tersi olan iki çıkışı vardır. Bu sebeple çıkışların her ikisinin birden aynı anda lojik '0' olma seçeneği yoktur. Bu nedenle bu durum Yasak Durum olarak isimlendirilir.

b) VE-DEĞİL (NAND) Kapısıyla Elde Edilmiş RS Flip Flop;



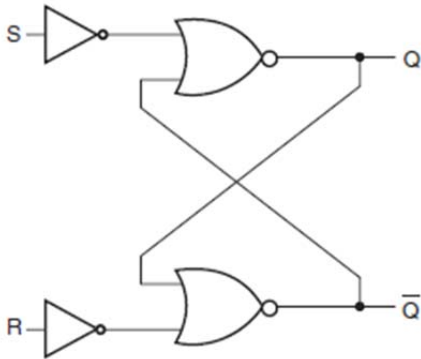
Giriş		Çıkış
R	S	Q_{n+1}
0	0	Yasak Durum
0	1	1
1	0	0
1	1	Değişmez / Depolama

Giriş		NAND
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

VE-DEĞİL (NAND) kapısıyla elde edilmiş flip-flop devresinde R (Sil / Reset) ve S (Kur / Set) girişine aynı anda lojik '0' uygulandığından flip flop devresinin iki çıkışı da (Q, \bar{Q}) aynı anda lojik '1' olmaktadır. Bilindiği gibi flip-flop'ların birbirinin tersi olan iki çıkışı vardır. Bu sebeple çıkışların her ikisinin birden aynı anda lojik '1' olma seçeneği yoktur. Bu nedenle bu durum Yasak Durum olarak isimlendirilir.

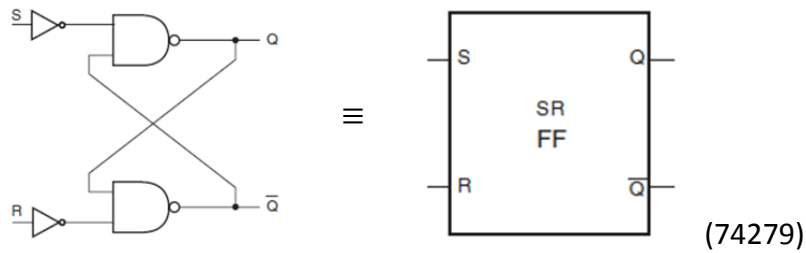
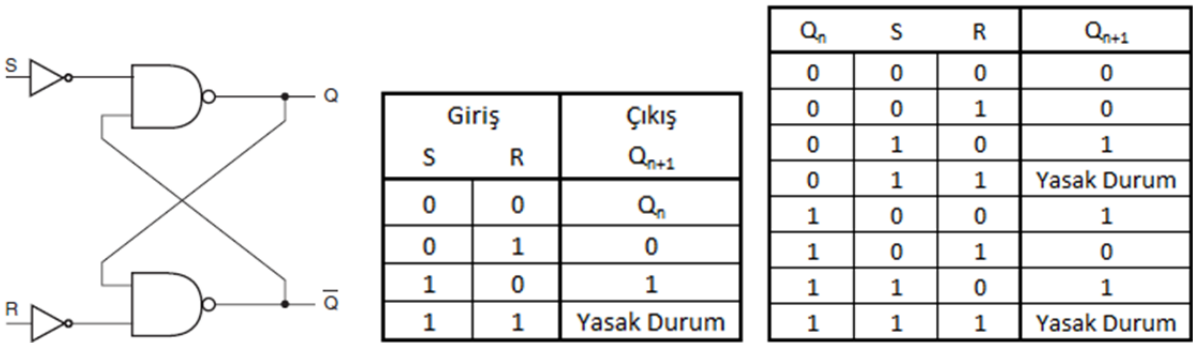
18.2. SR Flip Flop

a) VEYA-DEĞİL (NOR) Kapısıyla Elde Edilmiş SR Flip Flop;



VEYA-DEĞİL (NOR) kapılı RS flip flop' un girişlerine DEĞİL (NOT) kapısı bağlandığında SR flip flop elde edilmektedir.

b) VE-DEĞİL (NAND) Kapısıyla Elde Edilmiş SR Flip Flop;

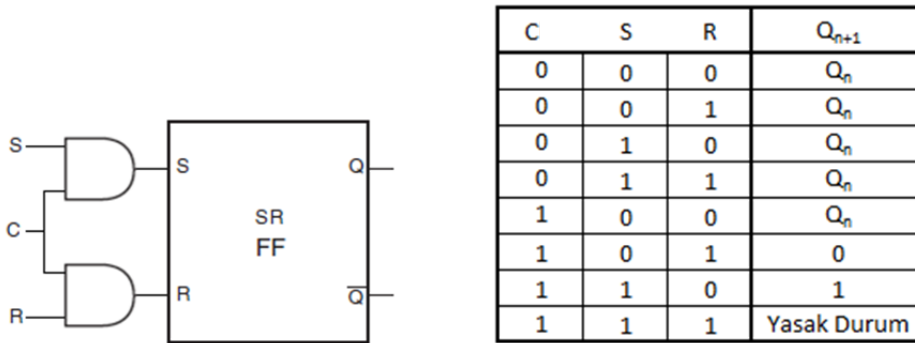


VE-DEĞİL (NAND) kapılı RS flip flop' un girişlerine DEĞİL (NOT) kapısı bağlandığında SR flip flop elde edilmektedir.

$$Q_{n+1} = \overline{Q_n} \cdot S \cdot \overline{R} + Q_n \cdot \overline{S} \cdot \overline{R} + Q_n \cdot S \cdot \overline{R} = \overline{R} \cdot [\overline{Q_n} \cdot S + Q_n \cdot (\overline{S} + S)] = \overline{R} \cdot [\overline{Q_n} \cdot S + Q_n]$$

$$Q_{n+1} = \overline{R} \cdot [S + Q_n]$$

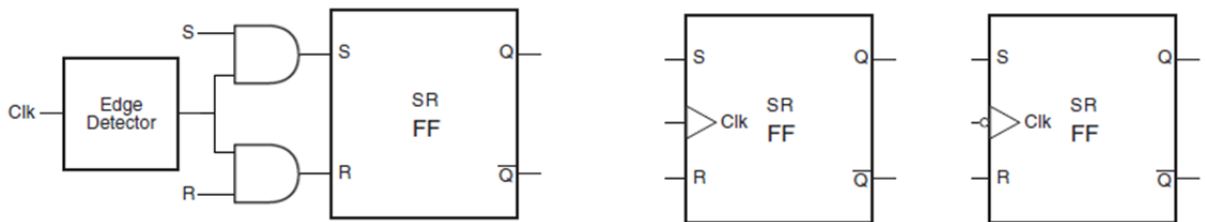
18.3. Saat Kumandalı SR Flip Flop



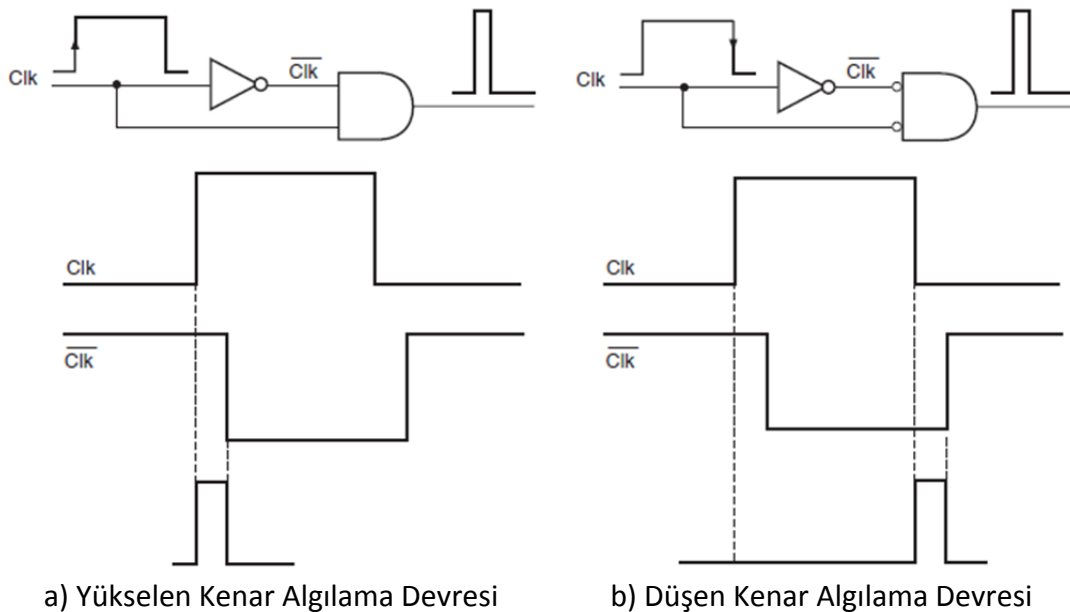
SR flip flop devresinin bir giriş işareti daha olabilmesi için yukarıdaki devre kullanılabilir. C girişi lojik '0' olduğu sürece flip flop devresinin çıkışı en son durumunu korur. C girişi lojik '1' olduğu durumda ise SR flip flop devresi normal SR flip flop devresi gibi çalışır.

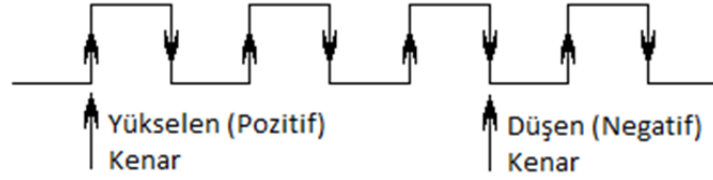
18.4. Kenar Tetiklemeli (Edge Triggered) SR Flip Flop

Yukarıda anlatılan C giriş değişkenine kenar algılama (Edge Triggered) devresi eklendiği zaman, SR flip flop kenar tetiklemeli flip flop haline gelmektedir. Kenar tetiklemeli flip flop'lar yükselen (pozitif) kenar tetiklemeli ve düşen (negatif) kenar tetiklemeli olarak ikiye ayrılırlar.



> işareti saat (clock) girişinin yükselen kenar tetiklemeli, •> işareti ise saat girişinin düşen kenar tetiklemeli olduğunu göstermektedir.



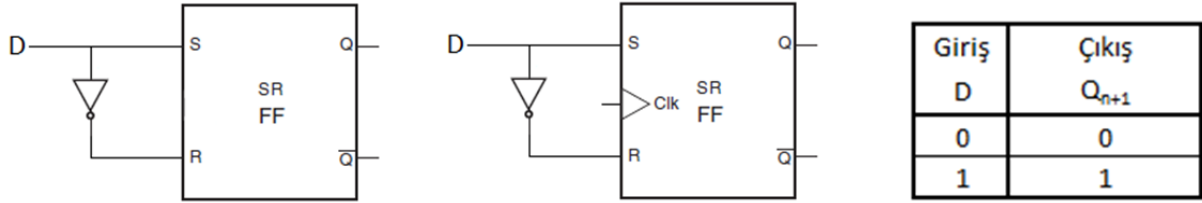


Yükselen kenar tetiklemeli SR flip flop devrelerinde saat girişinin lojik "0" dan lojik "1" e yükseldiği an S ve R girişlerinin durumları flip flop' un çıkışını belirler. Yükselen kenar aktiflik anı geçtiği zaman S ve R girişlerinde oluşan lojik değişimin çıkışa etkisi olmaz.

Düşen kenar tetikleme girişli SR flip flop devresinde ise saat girişinin lojik "1" den lojik "0" seviyesine indiği an S ve R girişlerinin durumları flip flop' un çıkışını belirler. Düşen kenar aktiflik anı geçtiği zaman S ve R girişlerinde oluşan lojik değişimin çıkışa etkisi olmaz.

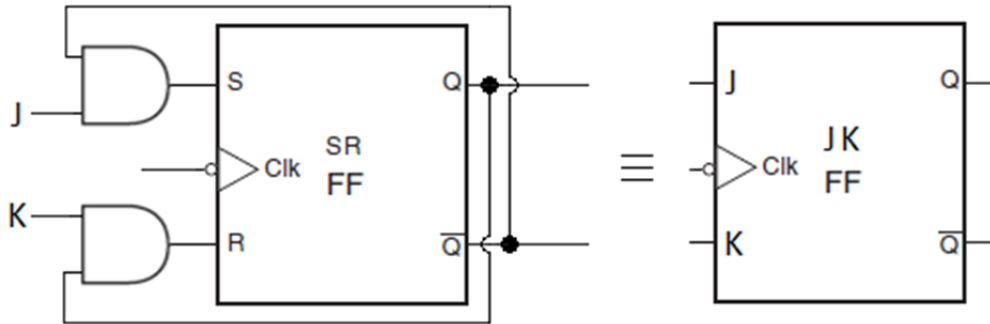
18.5. D Tipi SR Flip Flop

SR flip flop devresinin iki girişi arasına aşağıda görüldüğü gibi DEĞİL (NOT) kapısı bağlandığı zaman devrenin giriş sayısı bir olur ve farklı bir flip flop elde edilmiş olur. Elde edilen tek giriş D harfi ile gösterilir.



18.6. JK Flip Flop

SR flip flop devresinin girişlerine şekildeki VE (AND) kapısı eklenip çıkışlarının da girişe çapraz bağlanmasıyla JK flip flop elde edilir.



JK flip flop' un doğruluk tablosundan görüldüğü gibi giriş değişken ihtimallerinde yasak durum seçeneği yoktur. Böylece JK flip flop ile yasak durum seçeneği olmayan bir flip flop elde edilmektedir. JK flip flop'un saat (clock) işaretinin uygulanış şekline göre yükselen ve düşen kenar tetiklemeli olmak üzere iki farklı tipi vardır.

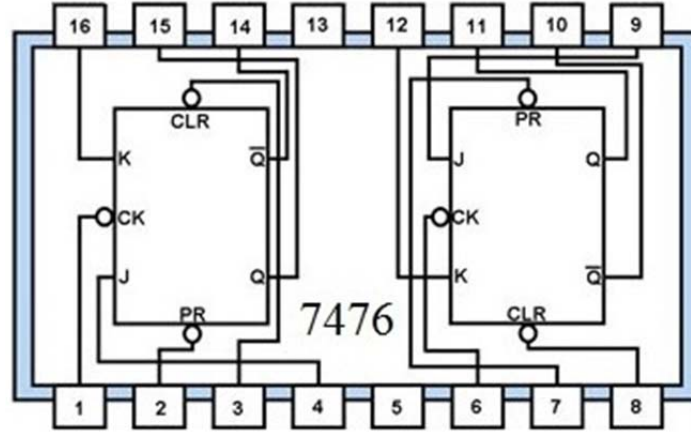
Giriş		Çıkış Q_{n+1}
J	K	
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

Q_n	J	K	Q_{n+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

JK flip flop' un doğruluk tablosundan yararlanılarak çıkış fonksiyonu aşağıdaki şekilde elde edilir.

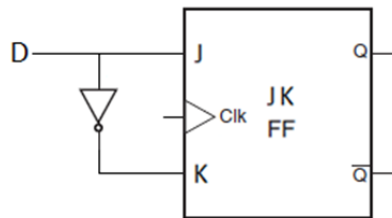
$$Q_{n+1} = \overline{Q_n} \cdot J \cdot \overline{K} + \overline{Q_n} \cdot J \cdot K + Q_n \cdot \overline{J} \cdot \overline{K} + Q_n \cdot J \cdot \overline{K} = \overline{Q_n} \cdot J \cdot (\overline{K} + K) + Q_n \cdot \overline{K} \cdot (\overline{J} + J)$$

$$Q_{n+1} = \overline{Q_n} \cdot J + Q_n \cdot \overline{K}$$



18.6. D Tipi JK Flip Flop

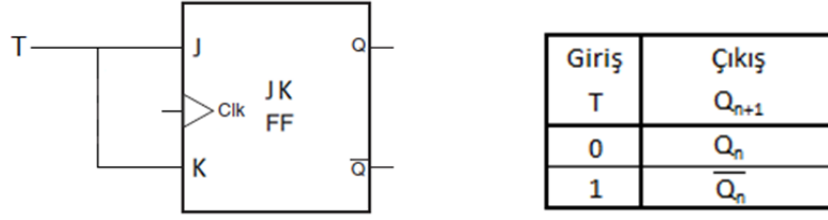
JK flip flop devresinin iki girişi arasına aşağıda görüldüğü gibi DEĞİL (NOT) kapısı bağlandığı zaman devrenin giriş sayısı bir olur ve D Tipi JK flip flop elde edilmiş olur. Elde edilen tek giriş D harfi ile gösterilir.



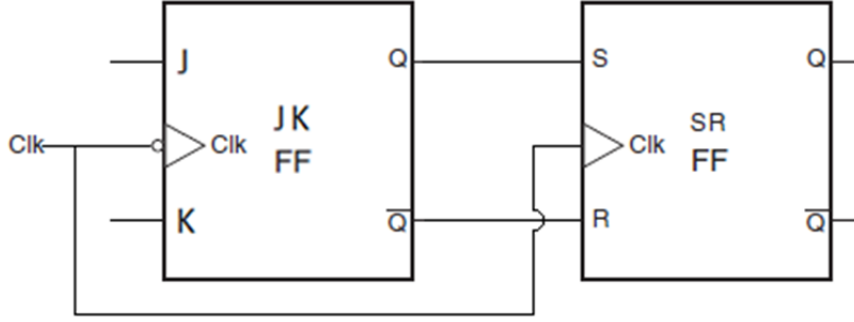
Giriş	Çıkış
D	Q_{n+1}
0	0
1	1

18.7. T Tipi JK Flip Flop (Toggle Flip Flop)

JK flip flop devresinin iki girişi birbirine bağlanıp tek bir giriş elde edildiği zaman T Tipi JK flip flop elde edilmiş olur. Elde edilen tek giriş T harfi ile gösterilir.



18.8. İki Kenar Tetiklemeli (Master-Slave) Flip Flop' lar

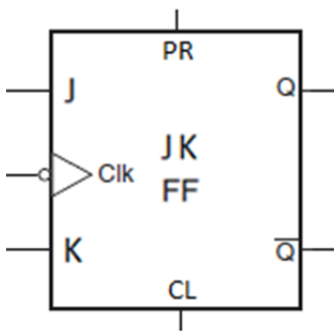


İki kenar tetiklemeli flip flop'larda master olarak kullanılacak flip flop, JK Flip-Flop olması gerekir. Çünkü SR flip flop' da var olan yasak durum ihtimali ortadan kalkmalıdır. Slave için SR flip flop kullanılması yeterlidir. Burada JK flip flop düşen kenar tetiklemeli, SR flip flop ise yükselen kenar tetiklemelidir.

18.8. Flip Flop' lardaki PRESET (PR) ve CLEAR (CL) Girişler

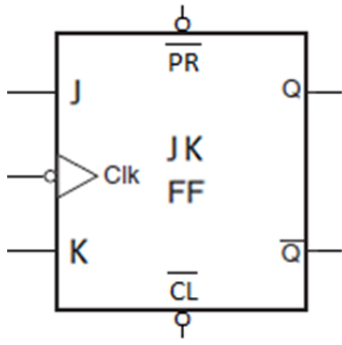
PRESET ve CLEAR girişleri aşağıdaki şekillerden görüldüğü gibi lojik "0" aktif veya lojik "1" aktif olabilirler.

PRESET : Preset girişi aktif olduğu zaman J, K ve Clk girişlerine bakılmaksızın flip flop çıkışı lojik "1" olur. Preset girişi pasif olduğu zaman flip flop çıkışı J, K ve Clk girişlerine bağlı olarak çıkış verir.



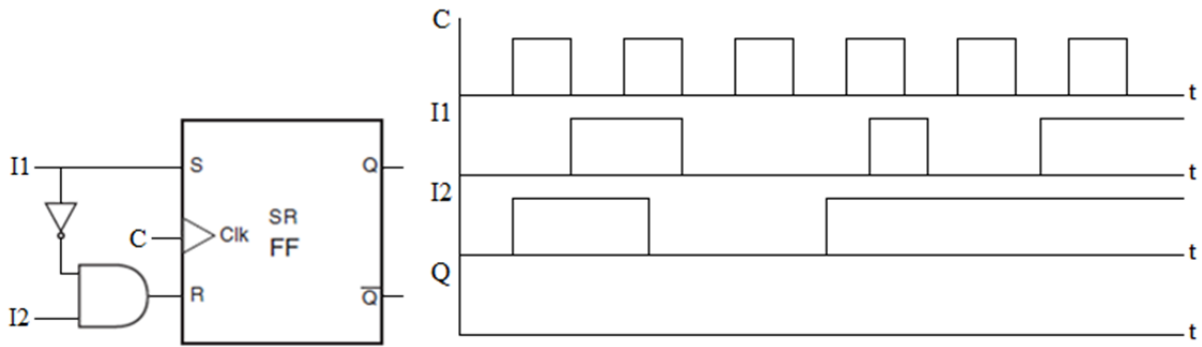
Giriş					Çıkış
PR	CL	Clk	J	K	Q_{n+1}
1	0	X	X	X	1
0	1	X	X	X	0
1	1	X	X	X	Tanımsız
0	0	↓	0	0	Q_n
0	0	↓	1	0	1
0	0	↓	0	1	0
0	0	↓	1	1	$\overline{Q_n}$

CLEAR : Clear girişi aktif olduğu zaman J, K ve Clk girişlerine bakılmaksızın flip flop çıkışı lojik "0" olur. Clear girişi pasif olduğu zaman flip flop çıkışı J, K ve Clk girişlerine bağlı olarak çıkış verir.

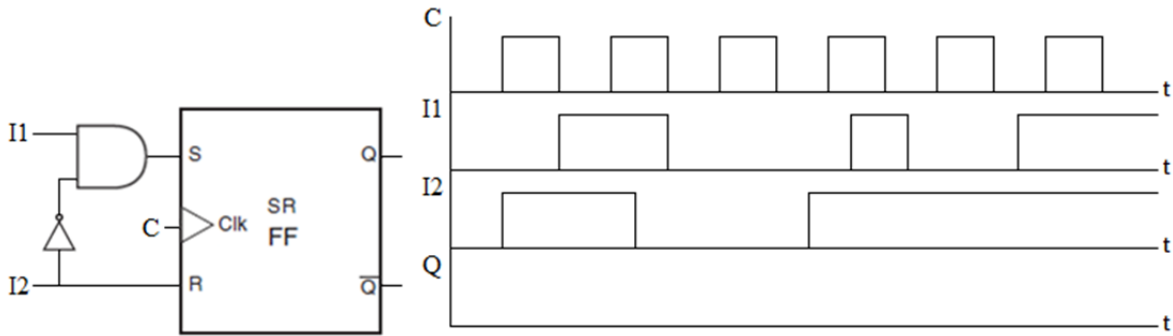


		Giriş			Çıkış
\overline{PR}	\overline{CL}	Clk	J	K	Q_{n+1}
0	1	X	X	X	1
1	0	X	X	X	0
0	0	X	X	X	Tanımsız
1	1	↓	0	0	Q_n
1	1	↓	1	0	1
1	1	↓	0	1	0
1	1	↓	1	1	$\overline{Q_n}$

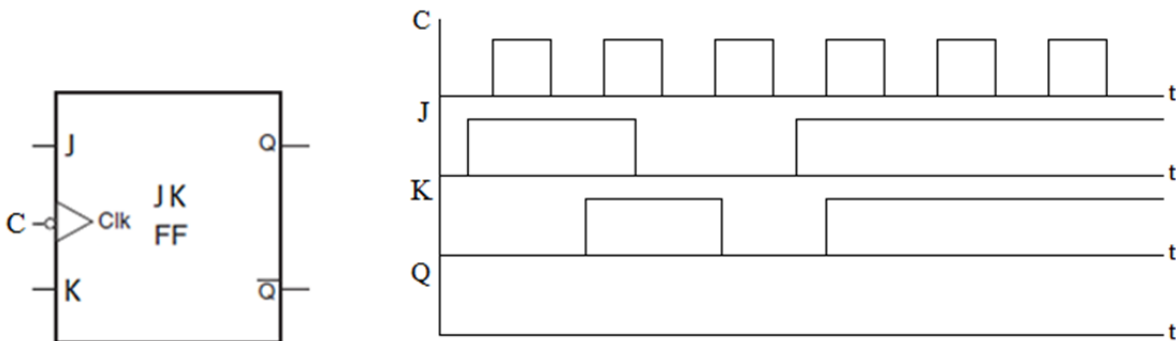
Örnek : Şekilde verilen devreye (Set öncelikli SR flip flop) I1, I2 ve C sinyalleri uygulandığı zaman flip flop çıkışının (Q) zamana göre değişimini çiziniz?



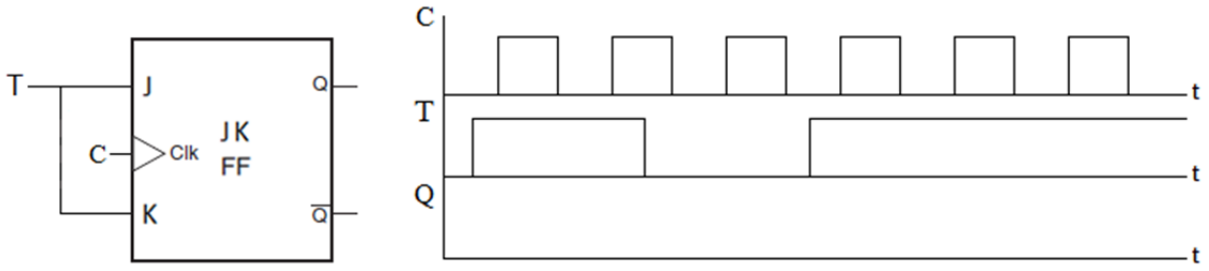
Örnek : Şekilde verilen devreye I1, I2 ve C sinyalleri uygulandığı zaman flip flop çıkışının (Q) zamana göre değişimini çiziniz?



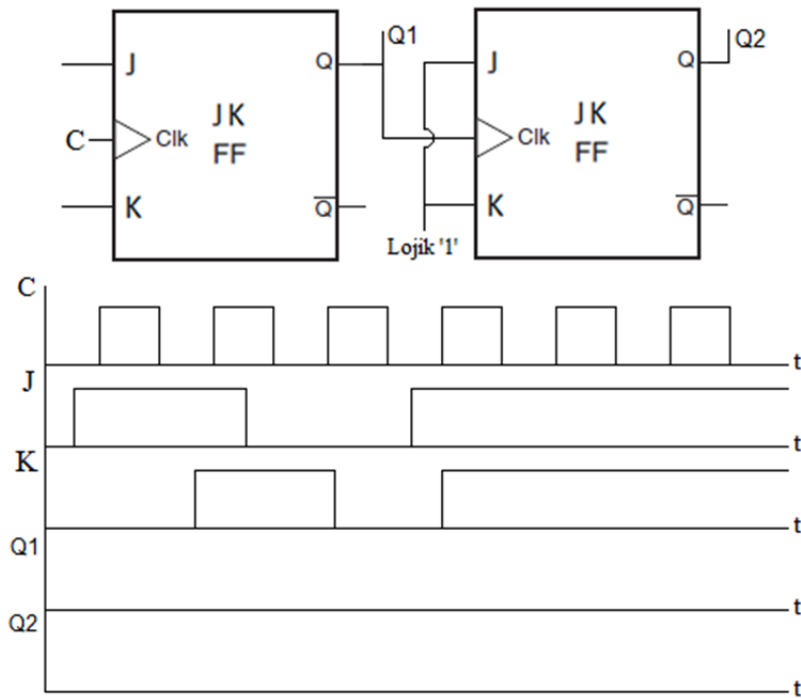
Örnek : Şekilde verilen devreye J, K ve C sinyalleri uygulandığı zaman flip flop çıkışının (Q) zamana göre değişimini çiziniz?



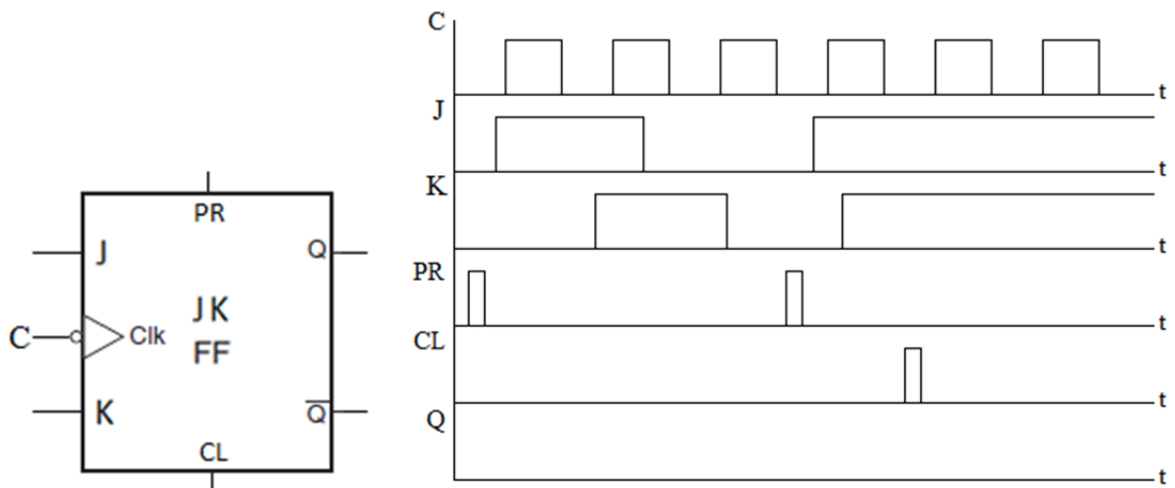
Örnek : Şekilde verilen devreye J, K ve C sinyalleri uygulandığı zaman flip flop çıkışının (Q) zamana göre değişimini çiziniz?



Örnek : Şekilde verilen devreye J, K ve C sinyalleri uygulandığı zaman flip flop çıkışlarının (Q1, Q2) zamana göre değişimlerini çiziniz?



Örnek : Şekilde verilen devreye J, K ve C sinyalleri uygulandığı zaman flip flop çıkışlarının (Q1, Q2) zamana göre değişimlerini çiziniz?



19. SAYICILAR (COUNTERS)

Giriş işaretlerine bağlı olarak belirli çıkışları sayısal ortamda periyodik olarak tekrarlayan lojik devrelere, sayıcı devreler adı verilir. Dijital (Sayısal) elektronikte çok geniş uygulama alanı olan sayıcı devreler flip flop'ların uygun bağlanmasıyla oluşturulurlar. Sayıcılar belli bir zaman diliminde darbe sayısının sayılması, dönen mekanizmanın hızının sayısal olarak tespit edilmesi, mikroişlemci devrelerinde zamanlayıcı (timer), sayıcı (counter), darbe genişlik modülasyon (PWM) üretici gibi yerlerde kullanılır.

Sayıcılar en genel şekli ile aşağıdaki gibi sınıflandırılabilir.

- a) Saat (clock) işaretinin uygulanış biçimine göre;
 - i) Asenkron (Farklı zamanlı / Seri) Sayıcılar
 - ii) Senkron (Eş zamanlı / Paralel) Sayıcılar
- b) Sayma biçimine (yönüne) göre;
 - i) İleri (Yukarı) sayıcılar (UpCounter),
 - ii) Geri (Aşağı) sayıcılar (Down Counter),
 - iii) İleri/Geri (Yukarı/Aşağı) sayıcılar (Up/Down Counter),
- c) Sayma kodlamasına göre;
 - i) İki tabanlı sayıcılar,
 - ii) BCD sayıcılar,
 - iii) Mod sayıcılar (Mod5, Mod10 sayıcı gibi),

19.1. Asenkron Sayıcılar

Asenkron sayıcılarda, sayma işlemi için kullanılan saat işareti birinci flip flop'a uygulanır. Birinci flip flop'un düz veya evrik çıkışından alınan sinyal ile daha sonra gelen flip flop tetiklenir. Birinci flip flop dışında kalan diğer flip flop'ların saat işaretleri bir önceki flip flop'un çıkışından gelir. Diğer bir deyişle flip flop'lar birbirlerine seri veya kaskat bağlanırlar. Bu sebepten dolayı asenkron sayıcılar, saat (clock) işareti farklı zamanlı olan sayıcılardır.

19.1.1. Asenkron Sayıcı ile İleri Sayma işlemi için ;

Sayıcı devresinde kullanılan flip flop'ların saat (clock) girişleri düşen kenar tetiklemeli ise ikinci ve daha sonraki flip flop'ların saat işareti bir önceki flip flop'un düz çıkışından (Q) alınırsa **veya** devrede kullanılan flip flop'ların saat (clock) girişleri yükselen kenar tetiklemeli ise ikinci ve daha sonraki flip flop'ların saat işareti ise bir önceki flip flop'un evrik çıkışından (\bar{Q}) alınır; ileri sayma işlemi gerçekleşir.

19.1.2. Asenkron Sayıcı ile Geri Sayma işlemi için ;

Sayıcı devresinde kullanılan flip flop'ların saat (clock) girişleri yükselen kenar tetiklemeli ise ikinci ve daha sonraki flip flop'ların saat işareti bir önceki flip flop'un düz çıkışından (Q) alınırsa **veya** devrede kullanılan flip flop'ların saat (clock) girişleri düşen kenar tetiklemeli ise

ikinci ve daha sonraki flip flop'ların saat işareti ise bir önceki flip flop'un evrik çıkışından (\bar{Q}) alınır; geri sayma işlemi gerçekleşir.

Saat İşareti (Clock)	2. ve daha sonraki FF'lar Düz Çıkıştan Besleniyorsa	2. ve daha sonraki FF'lar Evrik Çıkıştan Besleniyorsa
Yükselen Kenar	Geri Sayıcı	İleri Sayıcı
Düşen Kenar	İleri Sayıcı	Geri Sayıcı

Asenkron sayıcıların ortak özellikleri ;

a) Sayma işlemi için kullanılan saat işareti birinci flip flop'a uygulanır. Birinci flip flop dışında kalan diğer flip flop'ların saat işaretleri, sayma işlemine göre (ileri/geri) bir önceki flip flop'un düz veya evrik çıkışından gelir.

b) Sayıcıda bulunan tüm flip flop'ları J ve K girişleri lojik '1' seviyeye bağlanır.

c) Sayma işleminde atlama yok ise sayıcı devresindeki tüm flip flop'ların Preset ve Clear girişleri pasif yapılır.

ç) Sayma işleminde atlama var ise sayıcı devresindeki bazı flip flop'ların Preset / Clear girişleri aktif yapılır.

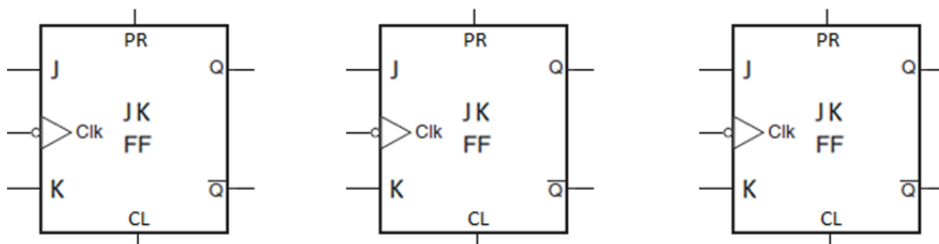
d) Sayma işleminde atlama var ise atlama işlemi için sayıcı devresinde VE (AND) kapısı veya VE-DEĞİL (NAND) kapısı kullanılır.

e) Sayıcı devresinde kullanılan flip flop'ların Preset / Clear girişleri lojik '1' aktif ise atlama işlemi için VE (AND) kapısı; Preset / Clear girişleri lojik '0' aktif ise atlama işlemi için VE-DEĞİL (NAND) kapısı, kullanılır.

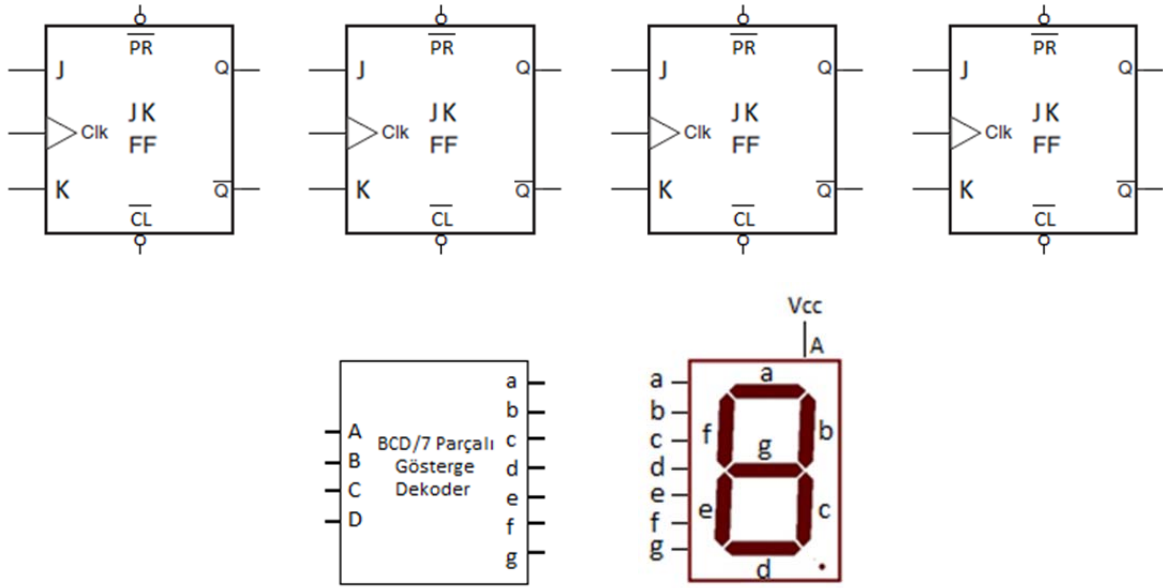
f) Asenkron sayılardaki sayma işlemlerinde genelde bir (1) atlama işlemi yapılır. Fakat istenirse sayma işlemlerinde birden fazla atlama yapılabilir.

g) Senkron sayıcılar tasarlanırken doğruluk tablosu oluşturulur ve lojik hesaplamalar veya karnaugh haritası kullanılarak sadeleştirme işlemleri yapılır. Asenkron sayıcıların tasarımı senkron sayıcılara göre daha kolaydır.

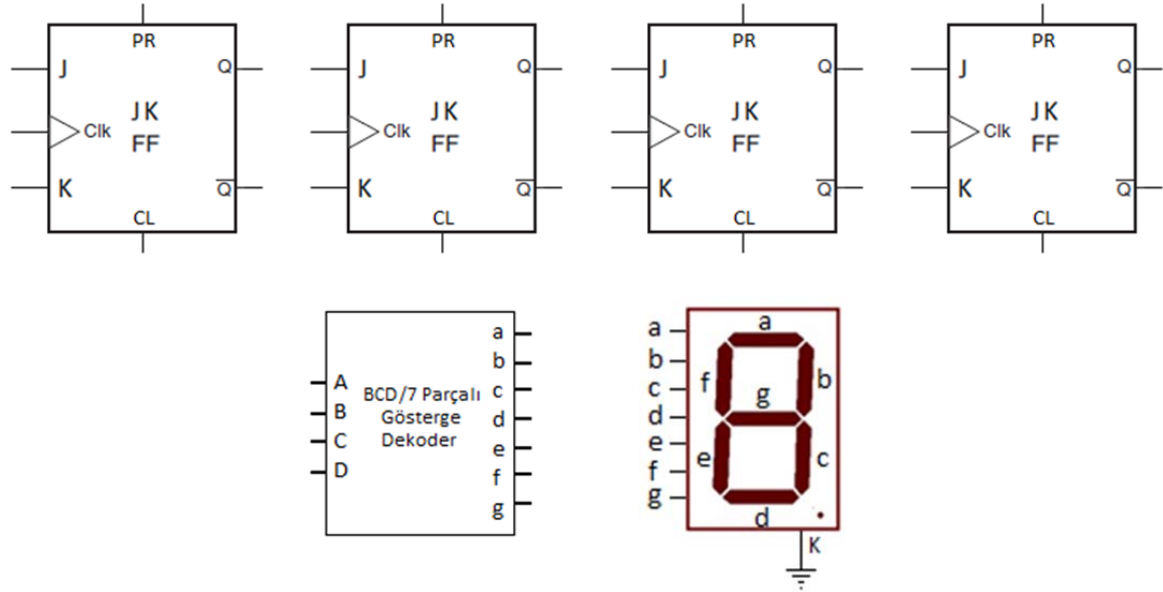
Örnek 1: 0, 7, 6, 5, 4, 3, 2, 1, 0, 7, 6, şeklinde geri sayan sayıcı tasarlayınız?



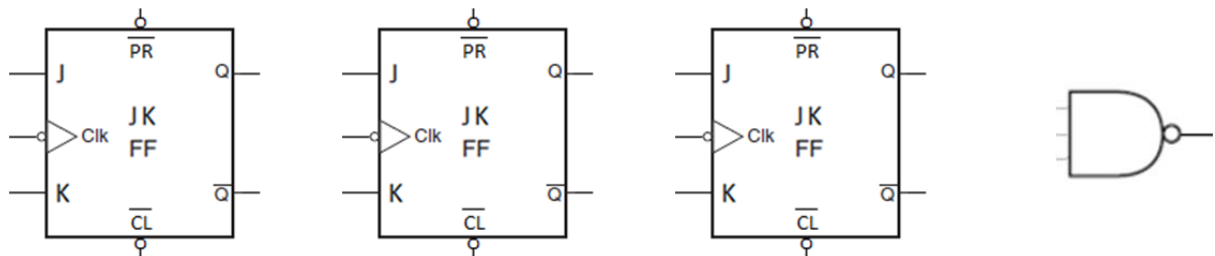
Örnek 2: Mod10 ileri sayan sayıcı tasarlayınız? Sayma işlemi ortak anodlu display üzerinden gözlemlenecektir.



Örnek 3: 4 bit geri sayan sayıcı tasarlayınız? Sayma işlemi ortak katodlu display üzerinden gözlemlenecektir.



Örnek 4: 0, 7, 6, 2, 1, 0, 7, 6, 2, şeklinde geri sayan sayıcı tasarlayınız?



19.2. Senkron Sayıcılar

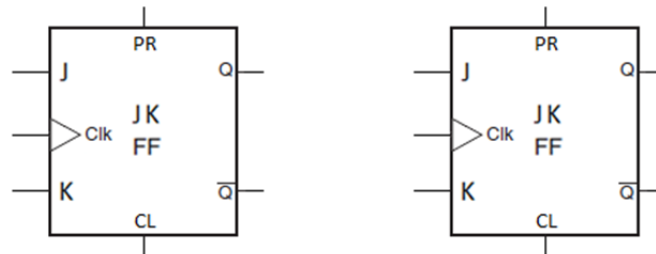
Senkron sayıcılarda, saat işareti sayıcıyı oluşturan bütün flip flop'lara tek bir hattan aynı anda uygulanır. Bu durumda devrede bulunan tüm flip flop'lar aynı anda tetiklenir. Başka bir ifade ile senkron sayıcılardaki tüm flip flop'ların saat işaretleri ortaktır.

Senkron sayıcılar tasarlanırken sayması istenen sayılar göz önüne alınarak doğruluk tablosu oluşturulur. Doğruluk tablosu oluşturulurken aşağıda verilen tablodan yararlanır.

Q_n	Çıkış		Giriş	
	Q_n	Q_{n+1}	J	K
0	→	0	0	X
0	→	1	1	X
1	→	0	X	1
1	→	1	X	0

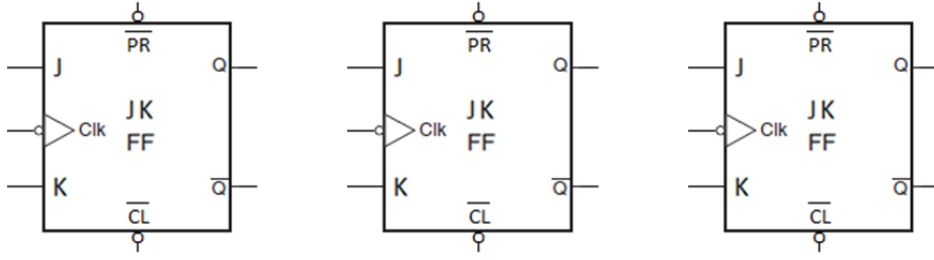
Oluşturulan doğruluk tablosuna göre kullanılacak flip flop'ların J, K girişleri için lojik ifadeler elde edilir. Flip flop'ların J ve K girişleri için elde edilen lojik ifadeler klasik sadeleştirme veya karnaugh haritası gibi yöntemler kullanılarak sadeleştirilir. Sadeleştirilen lojik ifadelere göre bağlantılar yapılır. Flip flop'larda boş kalan PR, CL girişlerini pasif hale getiren bağlantılar da yapılarak senkron sayıcı tasarımı yapılmış olur.

Örnek 5: 0, 1, 2, 3, 0, 1, şeklinde ileri sayan senkron sayıcı tasarlayınız?



Sayı	FF Çıkışları		FF Girişleri			
	Q1	Q0	J1	K1	J0	K0
0						
1						
2						
3						
0						

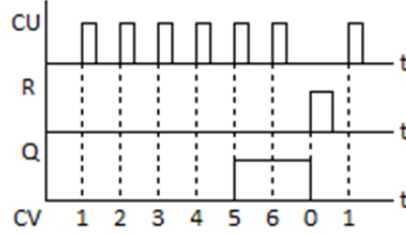
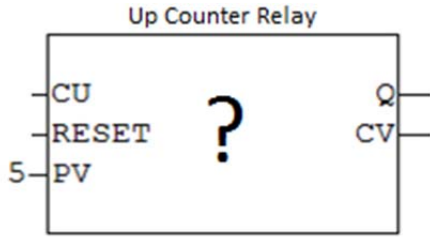
Örnek 6: 0, 1, 5, 4, 6, 0, 1, şeklinde sayan senkron sayıcı tasarlayınız?



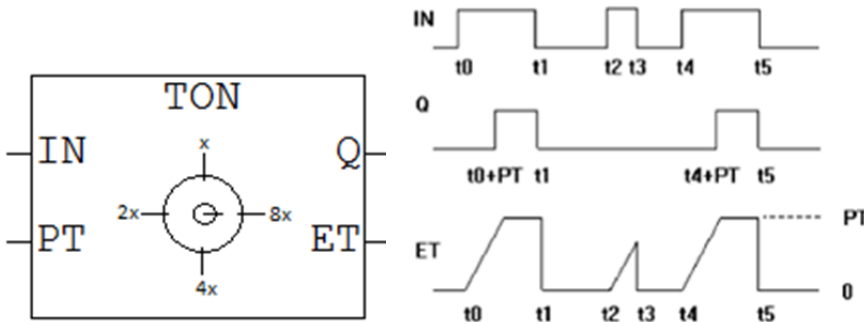
Sayı	FF Çıktıları			FF Girişleri					
	Q2	Q1	Q0	J2	K2	J1	K1	J0	K0
0									
1									
5									
4									
6									
0									

Örnek 7: 0 – 99 arası ileri sayan sayıcı tasarlayınız? Sayma işlemi ortak katodlu display üzerinden gözlemlenecektir. Devrenin bir sıfırlama (Reset) butonu bulunacak ve bu butona basıldığında sayı sıfırlanmalıdır. Ayrıca devrenin bir yetki (Enable) girişi olacak ve bu giriş var ise sayıcı saymaya başlayacaktır.

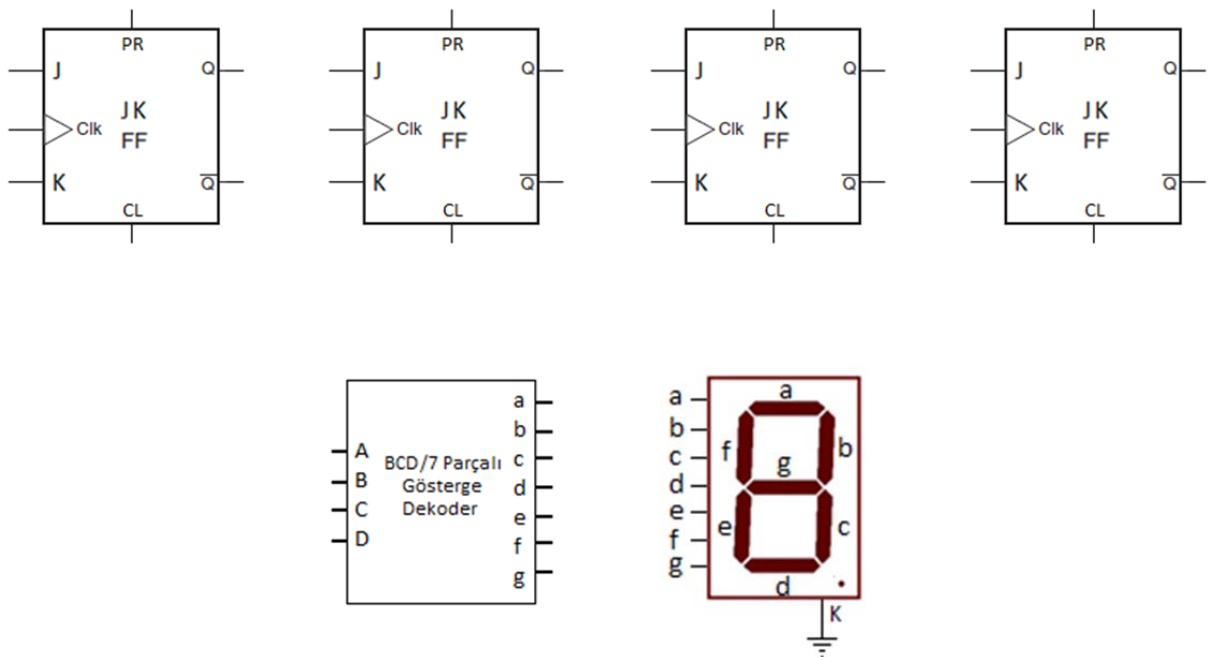
Örnek 8: 0 – 9 arası sayabilen ileri (Yukarı) sayıcı rölesi (Up Counter Relay) tasarlayınız? Blok şekilde, görüldüğü gibi rölenin 2 dijital girişi (CU, RESET), 1 dijital çıkışı (Q), 1 sayı girişi ve anlık olarak saydığı sayıyı gösteren 1 sayı çıkışı bulunmaktadır. Sayıcı zaman diyagramında belirtilen şekilde çalışmalıdır. Ayrıca sayıcı anlık değeri 9 olduktan sonra CU girişi gelse bile artmamalı veya sıfırlanmamalıdır.



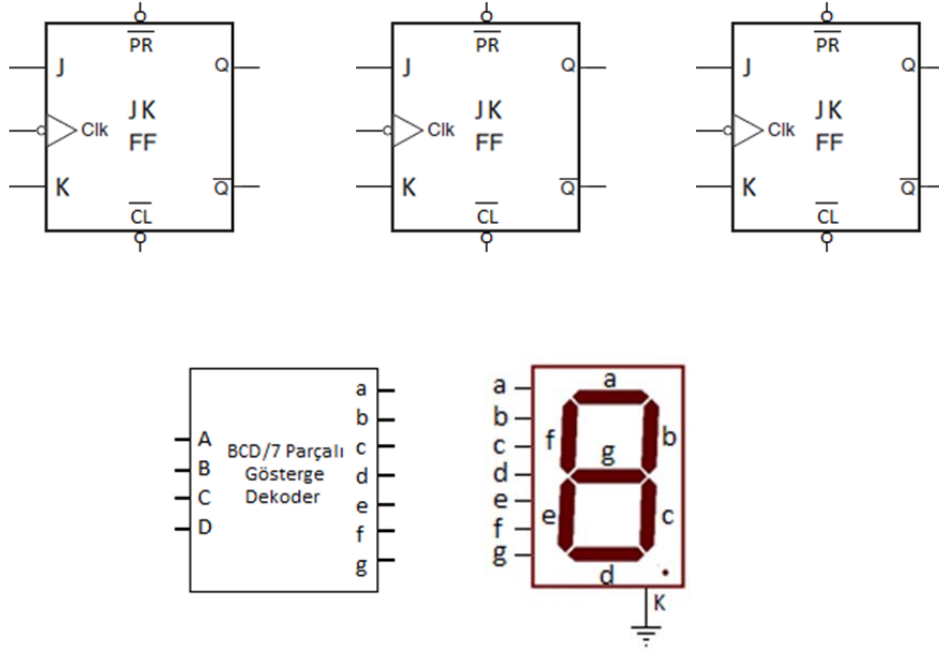
Örnek 9: 5 bitlik elektronik düz zaman rölesi (On Delay Timer - TON) tasarlayınız? Blok şekilde görüldüğü gibi rölenin 1 dijital girişi (IN), 1 dijital çıkışı (Q), 1 zaman girişi ve anlık olarak zamanı gösteren 1 zaman çıkışı bulunmaktadır. Röle zaman diyagramında belirtilen şekilde çalışmalıdır. Zaman rölesi 4 farklı zaman dilimlerini sayabilmelidir (x, 2x, 4x, 8x). Zaman rölesi sayabileceği en büyük zaman değerine geldiğinde zaman değeri artmamalı veya sıfırlanmamalıdır.



Örnek 10: 4 bit ileri-geri sayan sayıcı tasarlayınız? Sayma işlemi ortak katodlu display üzerinden gözlemlenecektir. { 3 girişli sayıcı [Clock, İleri giriş, geri giriş] veya 2 girişli sayıcı [İleri clock girişi, geri clock girişi] şeklinde iki farklı tasarım yapılabilir. }



Örnek 11: 3 bit ileri sayan sayıcı tasarlayınız? Sayma işlemi ortak katodlu display üzerinden gözlemlenecektir. Tasarlanacak sayıcının 1 veya 2 ek girişi olmalıdır. Bu girişlerin durumuna göre sayıcı normal ileri sayabilmeli, ileri tek sayıları sayabilmeli veya ileri çift sayıları sayabilmelidir.



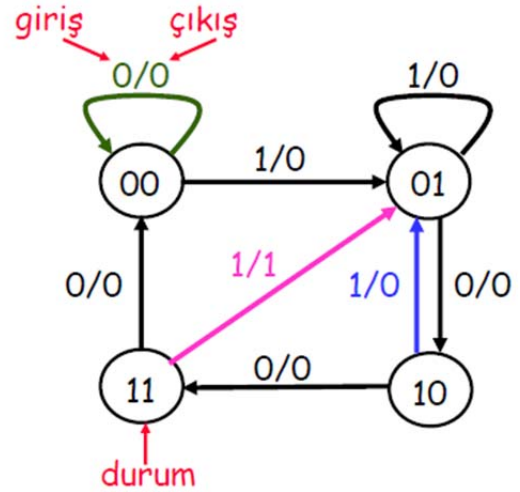
ÖDEV 1: Mikroişlemcilerde kullanılan 8 bit zamanlayıcı (timer) tasarlayınız? Zamanlayıcı 4 farklı zaman dilimlerini sayabilmelidir (x, 2x, 4x, 8x). Kesme (Interrupt) biti aktif ise zaman değeri her sıfırdan geçtiğinde kesme (interrupt) çıkışı vermeli ve kendini sıfırlamalıdır. Bir giriş yardımıyla zaman değeri istenildiği zaman sıfırlanabilmeli, diğer bir giriş yardımıyla zamanlayıcı çalıştırılıp durdurulabilmelidir.

ÖDEV 2: Ödev 1' de bahsedilen zamanlayıcıyı kullanarak mikroişlemcilerde kullanılan PWM (Pulse Width Modulation – Darbe Genişlik ayarı) modülü tasarlayınız?

20. DURUM TABLOSU ve DURUM DİYAGRAM YÖNTEMİ (STATE TABLE and STATE DIAGRAM METHOD)

Kombinasyonel (Karmaşık) devrelerde doğruluk tablosu girişler ile çıkışlar arasındaki ilişkiyi göstermektedir. Ardışıl devrelerde durum tablosu kullanılmaktadır. Durum tablosunda; girişler ile flip flop'ların şimdiki durumların sol tarafta, çıkış ile flip flop'ların gelecek durumlarının ise sağda sağ tarafta bulunmaktadır. Durum diyagramı çizilerek de durum tablosunu grafiksel olarak ifade edebiliriz. Aşağıda durum tablosu ve durum diyagramına ait bir örnek görülmektedir.

Şimdiki Durum		Giriş	Gelecek Durum		Çıkış
Q ₁	Q ₀	X	Q ₁	Q ₀	Z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1

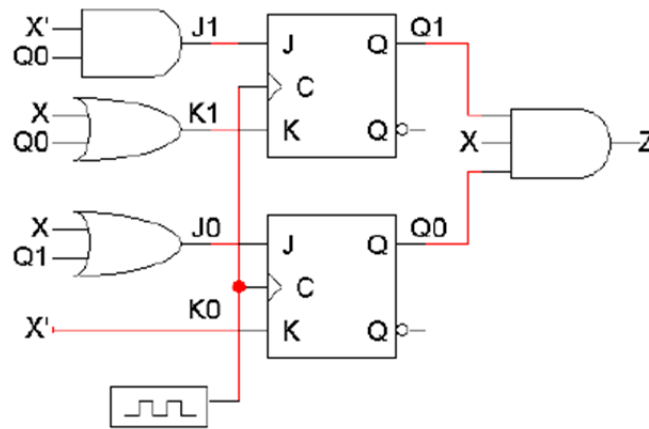


Durum diyagramı uygulanırken;

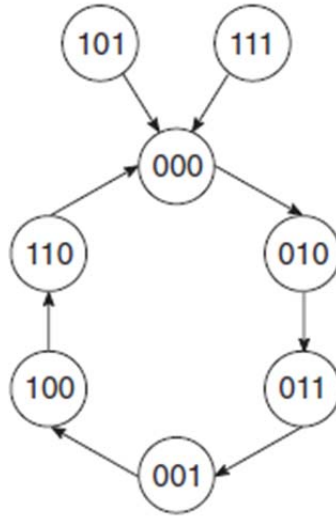
- Eğer n adet flip flop varsa, diyagramda 2^n adet durum (nod) olmalıdır.
- Eğer m adet giriş varsa, her bir durumda (nodda) 2^m adet çıkış oku olmalıdır.

Yukarıda verilen örneği incelersek;

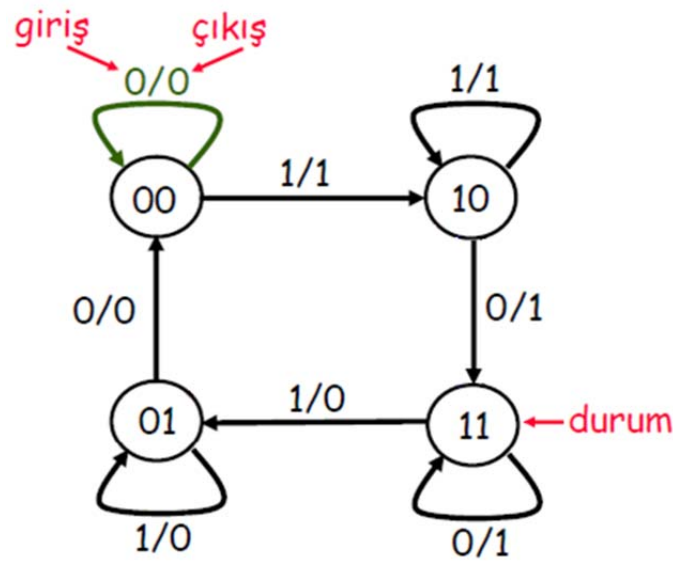
- 4 durum olduğu için devrede 2 adet flip flop olmalıdır.
- Her durumda 2 çıkış oku olduğu için devrenin 1 adet girişi olmalıdır.



Durum diyagramı sayıcı tasarlarken de kullanılmaktadır. Örneğin 0, 2, 3, 1, 4, 6, 0 sayan bir sayıcı durum diyagramıyla aşağıdaki gibi gösterilmektedir.



Örnek 1: Bir buton ile bir motorun kontrolü yapılacaktır. Butona bir kez basıldığında motor çalışacak, Butona ikinci kez basıldığında motor duracaktır. Gerekli devreyi durum tablosu ve durum diyagramı kullanarak tasarlayınız?



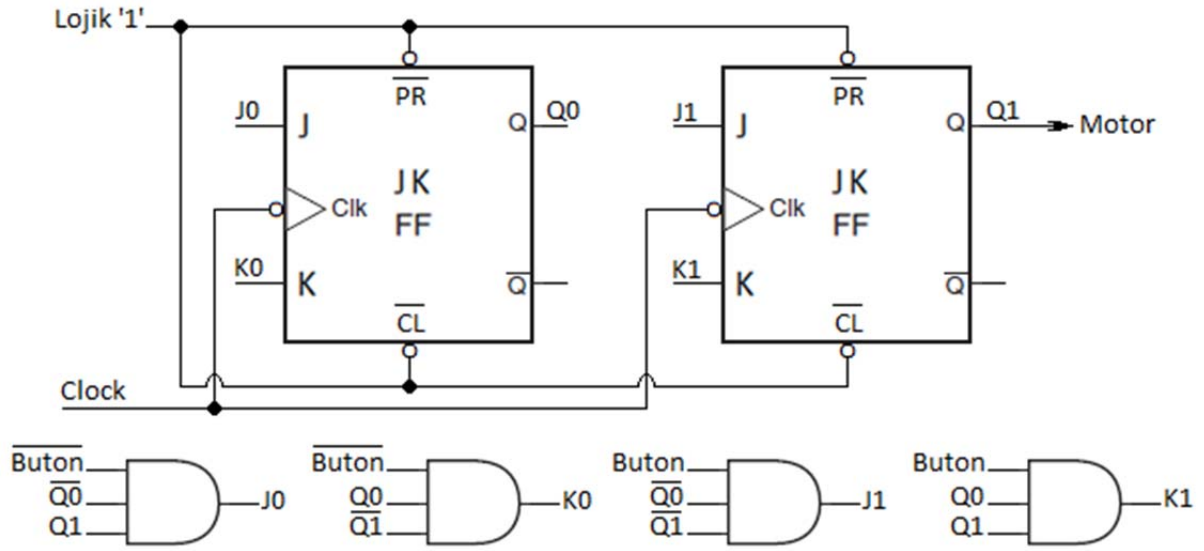
Şimdiki Durum		Giriş	Gelecek Durum		Flip Flop Girişleri				Çıkış
Q1	Q0	Buton	Q1	Q0	J1	K1	J0	K0	Motor
0	0	0	0	0	0	X	0	X	0
0	0	1	1	0	1	X	0	X	1
1	0	0	1	1	X	0	1	X	1
1	1	1	0	1	X	1	X	0	0
0	1	0	0	0	0	X	X	1	0

$$J0 = \bar{X} \cdot \bar{Q0} \cdot \bar{Q1}$$

$$K0 = \bar{X} \cdot Q0 \cdot \bar{Q1}$$

$$J1 = X \cdot \bar{Q0} \cdot \bar{Q1}$$

$$K1 = X \cdot Q0 \cdot Q1$$



ÖDEV 1: Bir buton yardımıyla bir doğru akım motoru dönüş yönünün kontrolü yapılacaktır. Butona 1. kez basıldığında motor saat yönünde dönecek, butona 2. kez basıldığında motor duracak, butona 3. kez basıldığında motor saat yönünün tersi yönünde dönecek ve butona 4. kez basıldığında motor duracaktır. Bu durum periyodik olarak devam edecektir. Gerekli devreyi durum tablosu ve durum diyagramı kullanarak tasarlayınız?