Kombinasyonel Mantık Devreleri

Karnaugh Map ve Temel Bileşenler

Karnaugh Map

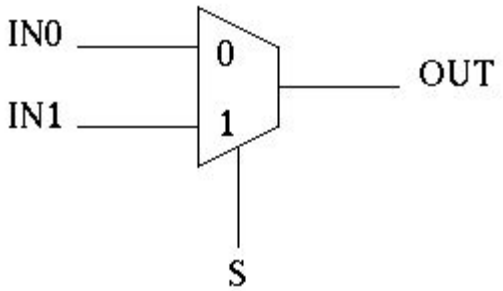
- Kombinasyonel devrelerin tasarımında kullanılır
- İstenen bir işlemin Boolean formülünün çıkarılmasında kullanılır
- 5 inputa kadar kağıt üzerinde hesaplamak için kullanılabilir
- Kısaca k-map olarak söylenir
- K-map'in özel bir çiziliş yöntemi vardır. Belli bir girdi değeri aynı olan hücreler birbirinin yanına gelecek şekilde çizilir

Karnaugh Map ile Tasarım

- İstenen işlemin girdilerini ve çıktılarını belirle
- Her bir çıktı için ayrı bir doğruluk tablosu (Truth table) hazırla
- Girdi sayısına uygun K-map çizilir
- K-map üzerine beklenen çıktı değeri (0, 1) yazılır, varsa X (don’t care) değerleri yazılır.
- En genel ifadeyi verecek gruplamalar yapılır
- Her grup için Boolean ifade yazılıp bu ifadeler OR işlemine sokulur
- Devre çizilir

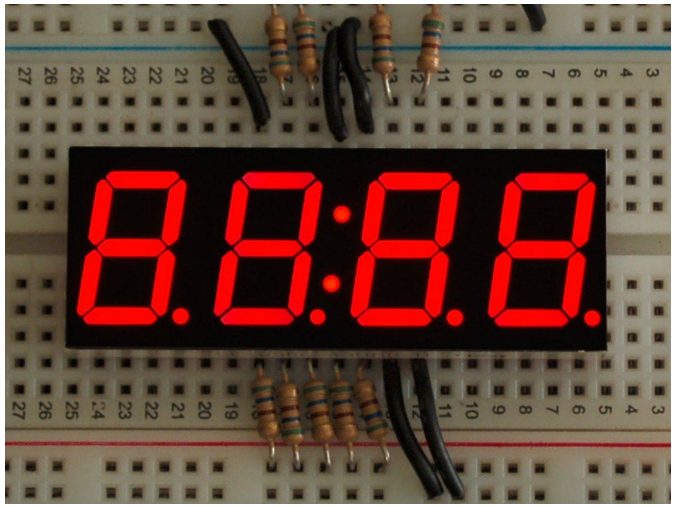
Tasarım Örneği - 1

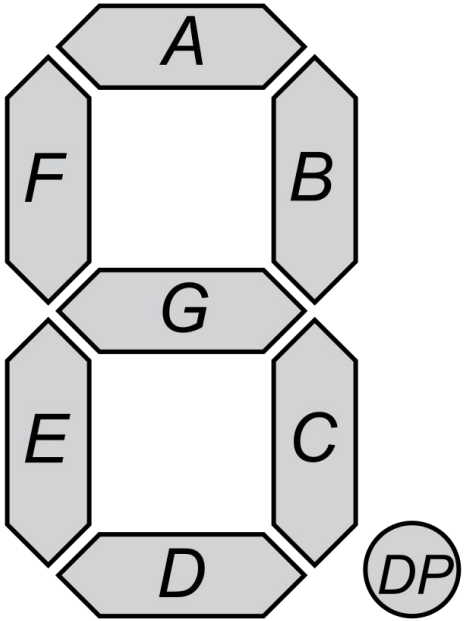
Aşağıda tasarlanması istenen elektronik bileşen verilmiştir. S girdisi 0 iken OUT değeri IN0 değerini, S girdisi 1 iken OUT değeri IN1 değerini vermelidir. Yani S girdisi kullanılarak 2 girdiden biri seçilerek çıkışta verilmelidir.



Tasarım Örneği - 2

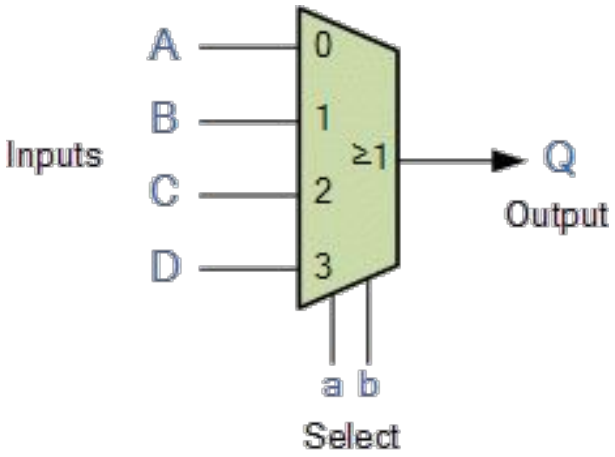
BCD to 7-segment decoder





Bileşenler

Multiplexer (MUX)



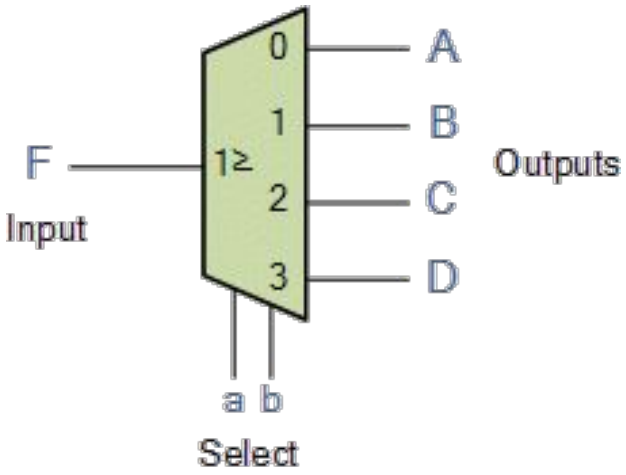
Multiplexer birden fazla girdi arasından bir tanesinin seçilerek çıkışa yönlendirilmesini sağlar.

A, B, C ve D girdilerdir. 1 veya 0 olabilirler.

a ve b hangi girdinin seçileceğini belirleyen seçim sinyalleridir. Örneğin ab = 10 = 2 ise C girdisindeki değer Q çıktısında verilir.

*Multi-bit inputs can be selected.

Demultiplexer (Demux)



Multiplexer'ın tersidir. Girdi sinyali seçilen çıkışlardan birine yönlendirilir. Diğer çıkışlar 0 verir.

F girdidir. 1 veya 0 olabilirler.

a ve b girdinin hangi çıktıya yönlendirileceğini belirleyen seçim sinyalleridir. Örneğin ab = 11 = 3 ise girdi D çıktısında verilir.

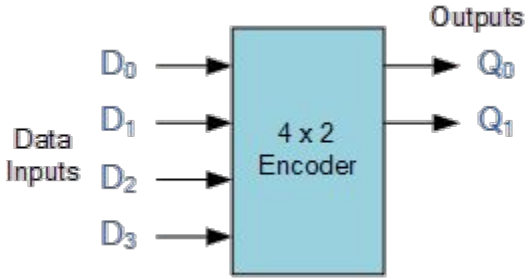
One-hot ve Binary Encoding (Kodlama)

Bir değeri farklı amaçlar için farklı şekillerde gösterebiliriz. Örnek:

Decimal (Onluk)	Binary (İkili)	One-hot
0	00	0001
1	01	0010
2	10	0100
3	11	1000

One-hot kodlamada değerlerden 1 tanesi her zaman 1’dir. Birden fazla veya hiç olmadığı durum olmaz.

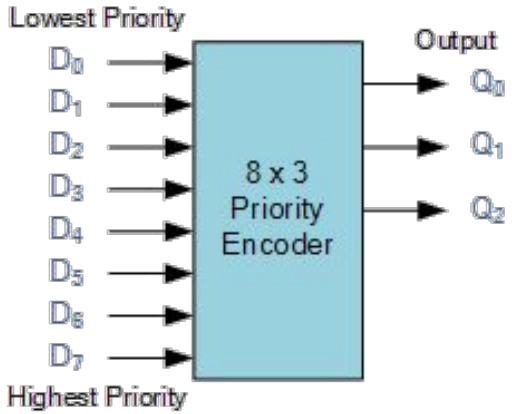
Binary Encoder



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

One-hot encoded değerleri binary-encoded değerlere dönüştürür. Yani hangi girdiden 1 geldiğini bize o girdinin numarasını vererek gösterir. Örneğin: D2=1 ve diğer girdileer 0 ise, Q1Q0 çıktısı 10 = 2 olur. D2'nin numarası ile aynı değer.

Priority Encoder

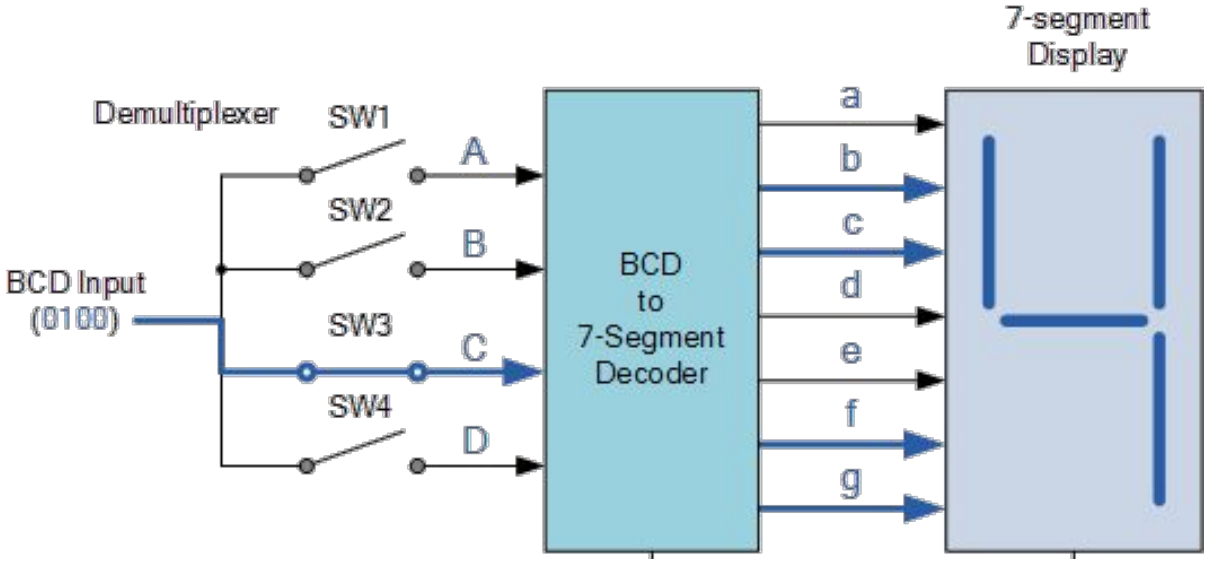


Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	x	0	0	1
0	0	0	0	0	1	x	x	0	1	0
0	0	0	0	1	x	x	x	0	1	1
0	0	0	1	x	x	x	x	1	0	0
0	0	1	x	x	x	x	x	1	0	1
0	1	x	x	x	x	x	x	1	1	0
1	x	x	x	x	x	x	x	1	1	1

X = dontcare

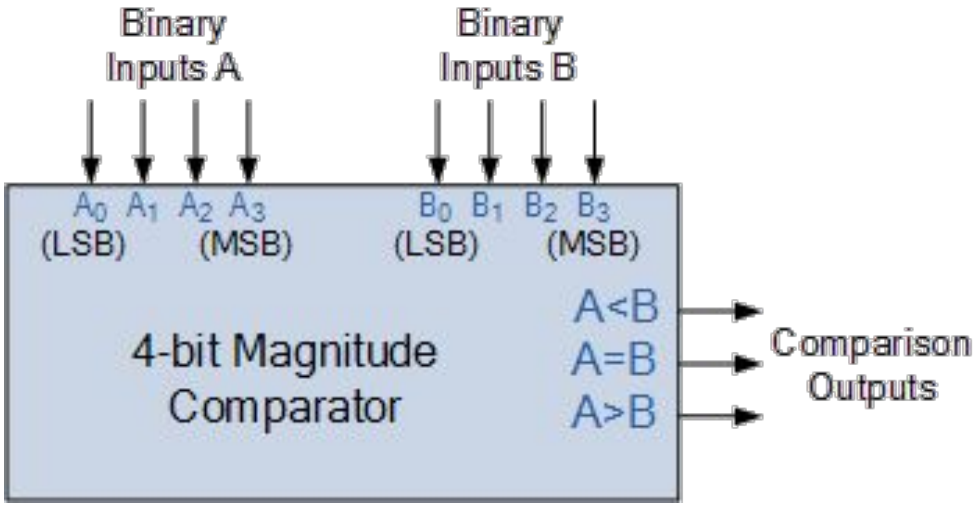
Binary encoder birden fazla girdinin 1 olduğu durumda hatalı sonuç verebilir. Priority Encoder girdi numarası büyük olan girdiye öncelik vererek bu sorunu çözer. Örneğin D2=1 ise D0 ve D1'in değeri ihmal edilir ve sonucu etkilemez.

Display Encoder



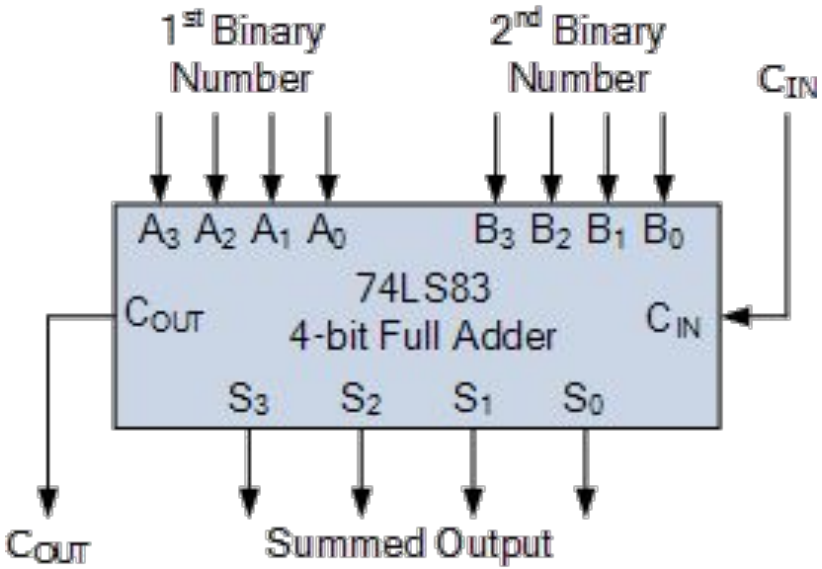
Binary (ikili) sayıyı 7-segment üzerinde göstermek için gerekli sinyalleri çıktılar.

Digital Comparator



Verilen 2 binary sayıyı karşılaştırır ve büyüklük, küçüklük ve eşitlik durumuna göre 3 çıktıdan birini 1 yapar.

Binary Adder



Binary adder binary 2 sayıyı toplar.

Negatif Sayılar

Negatif sayılar farklı şekillerde gösterilebilir. İki yöntemi inceleyeceğiz.

- Signed Magnitude Representation (SMR)
- 2’s complement

Programlama dillerinde signed integer ve unsigned integer kavramları buradan gelmektedir.

Signed Magnitude Representation (SMR)

Binary sayının en soldaki (most significant) biti işaret olarak kullanılır. Bu bit pozitif sayılarda 0, negatif sayılarda 1 olur. Diğer bitler büyüklüğü (magnitude) gösterir. 4 bit ile -7 ve +7 arasındaki sayılar gösterilebilir. Örnekler:

+5 → 0101, -5 → 1101 (kırmızı +, turuncu 5 anlamına gelir)

İnsanlar için anlaması kolay olsa da üzerinde toplama ve çıkarma işlemi yapmak basit değildir. Bu nedenle bilgisayarlar için işlemesi ek aşamalar gerektirir.

Bu gösterimin bir sonucu negatif ve pozitif 0 değerleri olmasıdır.

+0 → 0000, -0 → 1000

2’s complement

Bir sayının negatifini bulmak için tüm bitler ters çevrilir ve sonuç 1 artırılır. Örnek:

+5 → 0101, -5 → 1010 + 1 = 1011

En soldaki (most significant) bit yine işareti göstermiş olur. Bu gösterimin avantajı doğrudan toplama ve çıkarmaya izin vermesi ve bilgisayarlar tarafından hızlıca işlenebilmesidir. Arka arkaya gelen sayılar bu gösterimde bir önceki sayıya 1 eklenerek bulunabilir. Örnek:

-2 → (0010)’ + 1 = 1101 + 1 = 1110 (Yeşil kısım 2’s complement ile -2’yi hesaplıyor)
-1 → 1110 + 1 = 1111
 0 → 1111 + 1 = 0000 (en sola gelmesi gereken elde kaybolur)
 1 → 0000 + 1 = 0001

2’s complement

4 bit ile -8 ve +7 arasındaki 16 farklı değer gösterilebilir.

Sayıları göstermek için sınırlı sayıda bit kullanıldığı için mümkün olan aralığın dışına çıkıldığında sayı aralığın diğer tarafın yeniden başlar. Örneğin:

7 → 0111. Eğer bu sayıyı 1 artırırsak 1000 buluruz. Bu değer -8’dir. Benzer şekilde -8’i 1 azalttığınızda +7 bulursunuz. Bu olaya overflow denir. Overflow’u engellemek için daha fazla bit kullanabilirsiniz. Örneğin 8 bit gibi. Günümüz işlemcileri 64 bit veriyi tek seferde işleyebiliyor. Ancak daha hassas işlemler için 128 veya 256 bitlik sayıları kullanabilirsiniz. Tek farkı işlemci bu sayıları tek adımda değil, birkaç adımda toplar.

Overflow

32-bit signed integer
−2,147,483,648 (− 2^31)
ile
2,147,483,647 (2^31 − 1)
arasındaki sayıları gösterebilir.

Youtube izlenme sayısındaki overflow'dan sonra 64-bit sayılara geçti. Aralık:
-9,223,372,036,854,775,808
ile
9,223,372,036,854,775,807

Signed vs Unsigned Integer

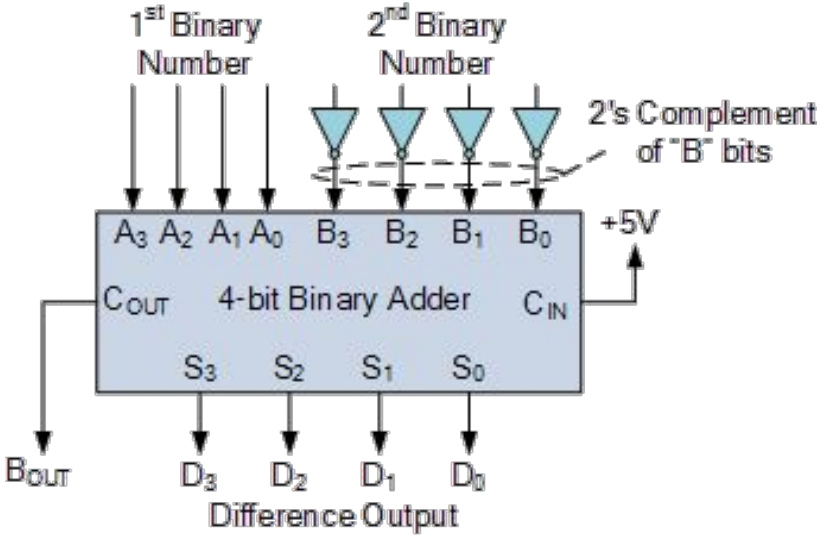
4 bit 2^4 = 16 farklı değer tutabilir. Negatif sayılara ihtiyacınız varsa ilk biti işaret (sign) biti olarak kullanıp -8 ve +7 arasındaki değerleri tutabilirsiniz.

Eğer negatif sayılara ihtiyacınız yoksa sign bitini ihmal edip 0 ve 15 arasındaki sayıları tutabilirsiniz.

Yani sayıyı işaretli veya işaretsiz (signed veya unsigned) kabul etmenize göre aynı gösterim farklı değerlere karşılık gelmektedir. Örnek:
Sign biti varsa: 0000 → 0, 0111 → 7, 1000 → -8, 1111 → -1
Sign biti yoksa: 0000 → 0, 0111 → 7, 1000 → 8, 1111 → 15

C ve C++ gibi bazı dillerde interger değişkenler uint32 (unsigned 32-bit int) veya int8 (signed 8-bit int) gibi sayının işaret içerip içermediğini gösterecek şekilde tanımlanır.

Binary Subtractor



Binary subtractor binary 2 sayıyı birbirinden çıkarır. Bu işlem için ayrı bir devre tasarlamak yerine binary adder'a ikinci sayının negatifi verilerek toplama işlemi yaptırılır.

Multiplexer kullanılarak toplama ve çıkarma işlemleri arasında geçiş yapılabilir.

Teşekkürler