Binary Sayılarda Tümleyen Aritmetiği

Bilgisayarlarda çıkarma işlemini gerçekleştirmek için kullanılan matematiksel yöntemdir. Negatif sayıların bilgisayarda işlenmesine imkan sağlayan "ikili tümleyen" yöntemi en önemlisidir.

1'li tümleyen

Herhangi bir binary sayıda 1'li tümleyen bulunurken sayıdaki tüm bitler terslenir.

0 → 1 1 → 0

Genel olarak "r" taban "n" basamak sayısı "N" sayı olmak üzere

r-1 tümleyeni

$r^n - N - 1$

formülü ile bulunur.

N = 10011010

2^8 - 10011010 - 1

2^8 → 100000000
N → 10011010

00110011 0
1
—
00110 0101

Tümleyeni bulduk

2'li tümleyen

Bir binary sayının ikili tümleyenini bulmak için 1'li tümleyenine 1 eklenir.

$(101100)_2$ 1'li tümleyeni
$(010011)_2$ +1 ekle
$(010100)_2$

Genel olarak r - taban
n - basamak sayısı
N - sayı olmak
üzere r tümleyeni
$r^n - N$ formülü
ile bulunur.

2'li tümleyen ile çıkarma

İki pozitif sayının M - N işlemi yapılırken M ve N'nin 2'li tümleyeni toplanır. Eğer toplama sonucunda en sol basamakta elde oluşuyorsa elde biti atılır kalan kısım pozitiftir.

Eğer elde biti yoksa bulunan değerin ikili tümleyeni alınıp önüne eksi konulur.

Örnek 1.1.1.1.1.1.1.1

M = 1010100
N = 1000100
M - N?

$N^{1'li}$ → 0111011
$N^{2'li}$ → 0111100

M-N → 1010100
+ 0111100
—
1 0010000

Bu sayı atılır ve sonuç pozitiftir

M - N = 0010000 (pozitif)

Örnek 1.1.1.1.1.1.1.2

N - M = ?

$M^{1'li}$ → 0101011
$M^{2'li}$ → 0101100

N 1000100
+ $M^{2'li}$ 0101100
—
0 1110000

elde oluşmadı
ikili tümleyen alınır
sonuç negatiftir

1'li 1110000
0001111

2'li 0010000

−0010000

İşaretli Sayılar

Bilgisayar gibi sayısal sistemler hem pozitif hem de negatif sayılarla işlem yapabilir. Bir işaretli sayı hem işaret hem büyüklük içerir. İşaretli binary sayılar, işaret - büyüklük, 1'li tümleyen ve ikili tümleyen biçiminde gösterilebilir.

En sık kullanılan ikili tümleyen gösterimidir. Bir işaretli binary sayıda, işaret biti MSB bitidir. MSB "0" ise pozitif "1" ise negatif bir sayıdır.

İşaret - Büyüklük Biçiminde Sayılar

Bu biçimdeki sayıda MSB biti işaret, kalan bitler sayı büyüklüğünü gösterir.

$+17 = \underbrace{0}_{\text{MSB işaret biti}}\underbrace{0010001}_{\text{büyüklük}}$

$-17 = \underbrace{1}_{\text{MSB işaret biti}}\underbrace{0010001}_{\text{büyüklük}}$

1'li Tümleyen Biçiminde Sayılar

Bu formatta pozitif sayılar işaret - büyüklük biçimiyle aynı, negatif sayılar ise pozitif sayının birli tümleyeni ile gösterilir.

$+17 = \underbrace{0}_{\text{işaret}}\underbrace{0010001}_{\text{büyüklük}}$

1'li tümleyeni alınır

$-17 = \underbrace{1}_{\text{işaret}}\underbrace{1101110}_{\text{1'li tümleyen}}$

İkili Tümleyen Biçiminde Sayılar

Bu formatta da pozitif sayılar işaret büyüklük biçimi ile aynı negatif sayılar ise o sayının ikili tümleyeni ile gösterilirler.

$+17 = \underbrace{0}_{\text{işaret}}\underbrace{0010001}_{\text{sayı büyüklüğü}}$

2'li tümleyeni

$-17 = \underbrace{1}_{\text{işaret}}\underbrace{1101111}_{\text{2'li tümleyen}}$

SAYISAL KODLAR

Analog bilgiler dijital sistemlerde işlenirken bazı özel kodlar kullanılır. Bu kodlarla sayı harf ve semboller temsil edilir.

BCD Kodu

Onluk sistemdeki bir sayının herbir basamağının ikilik sistemdeki karşılığının 4 bit şeklinde yazılması yöntemidir.

$(263)_{10}$

$(0010 \quad 0110 \quad 0011)_{BCD}$

GRAY Kodu

İkilik sayı sisteminde bir artım adımında "r" basamak kadar bit değerinde değişim olabilmektedir. Bu değişimi minimum yapmak için kurulan sistemdir. Basamak değeri yoktur. Giriş çıkış birimleri ve ADC gibi elektronik aletlerde kullanılırlar. Yapılış yöntemi MSB bitinin soluna sıfır konulup sayının her biti solundaki bit ile toplanır. Bu işlem LSB bitine kadar devam ettirilir.

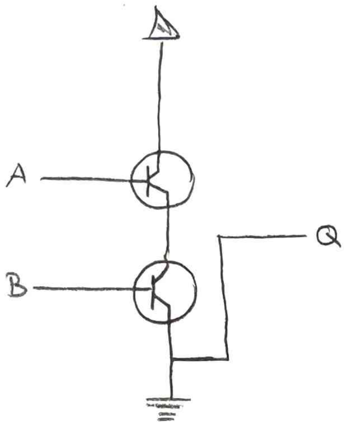
1101 sayısının gray kodu için başına 0 konulur.

01101
1011

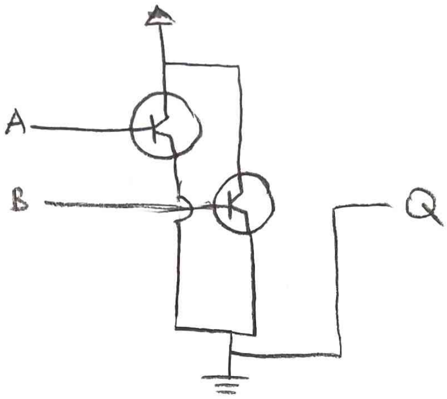
0111
1000 geçiş yaparken 3 bit değişiyor

Ama gray kodu bunun önüne geçmeyi hedefler

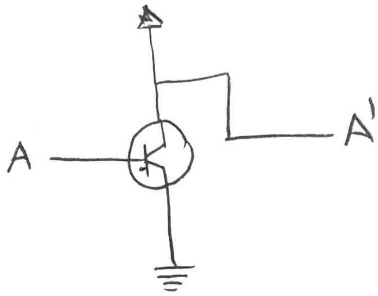
AND



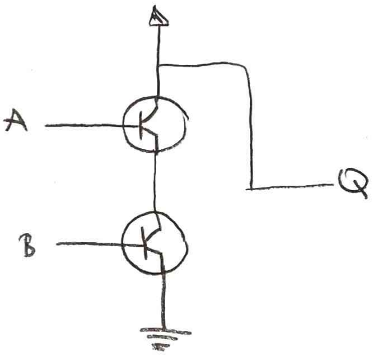
OR



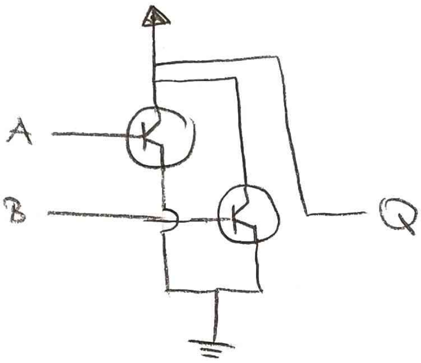
NOT



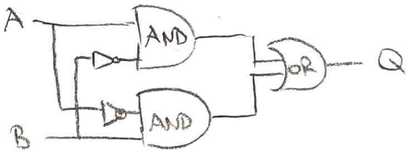
NAND



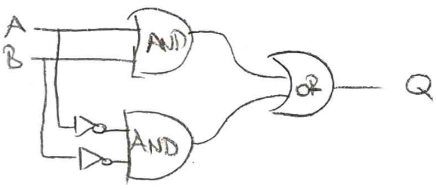
NOR



XOR



XNOR



Boolean Cebri

Tasarlanan bir lojik devrede sadeleştirme yaparak ilk tasarımın daha basit ve ucuz, indirgenmiş eşdeğer biçimini bulmaya yarayan matematiksel metod ve kurallar bütünüdür.

Boolean Kanunları

→ Yer değiştirme

$A+B \equiv B+A$

$A.B \equiv B.A$

→ Birleşme

$A+(B+C) \equiv (A+B)+C$

$A.(B.C) \equiv (A.B).C$

→ Dağılma

$A.(B+C) \equiv A.B + A.C$

$A+(B.C) \equiv (A+B).(A+C)$

Boolean Kuralları

$A+0 \equiv A$

$A+1 \equiv 1$

$A.0 \equiv 0$

$A.1 \equiv A$

$A+A \equiv A$

$A.A \equiv A$

$A+\bar{A} \equiv 1$

$A.\bar{A} \equiv 0$

$\bar{\bar{A}} \equiv A$

$A + A.B \equiv A$

$A.(A+B) \equiv A$

$A + \bar{A}B \equiv A+B$

DeMorgan Kuralları

$\overline{A+B} \equiv \bar{A}.\bar{B}$

$\overline{A.B} \equiv \bar{A}+\bar{B}$

Doğruluk tablosu

Lojik devrelerinde giriş değişkenlerinin alabileceği sayısal değerleri ve bunlara göre çıkışların durumunu gösteren tablolardır. N tane değişken için 2^N değişik durum oluşur.

Boolean Fonksiyonları

Binary değişkenler 0 ve 1 sabitleri lojik işlem sembolleri içeren cebirsel ifadelerdir.

MINTERM

Binary değişken A, ya normal formu A ile ya da tümleyeni olan $\bar{A}$ ile gösterilebilir. Bir AND lojik işlemini düşündüğümüzde A ve B binary değişkenleri vardır. İki formu ele alınarak 4 farklı olasılık vardır. Bu AND işleminin tanımlarından her birine MINTERM denir. Her bir MINTERM binary karşılığı sıfır olanların tümleyeni ve bir olanların normal formları alınarak AND işlemlerinden geçirilen n tane değişkenden oluşur.

$\bar{A}.\bar{B}$ / $\bar{A}.B$ / $A.\bar{B}$ / $A.B$

Boolean fonksiyonları da fonksiyondaki 1'leri sonuç veren, tüm mintermlerin OR'lanmasıyla ifade edilir. MINTERMLERİN Toplaması ya da çarpımların toplaması şeklinde de açıklanabilen bu fonksiyonlar kanonik formu oluşturur.

A	B	C	Minterm	Sembol
0	0	0	$\bar{A}.\bar{B}.\bar{C}$	m_0
0	0	1	$\bar{A}.\bar{B}.C$	m_1
0	1	0	$\bar{A}.B.\bar{C}$	m_2
1	0	0	$A.\bar{B}.\bar{C}$	m_3
0	1	1	$\bar{A}.B.C$	m_4
1	0	1	$A.\bar{B}.C$	m_5
1	1	0	$A.B.\bar{C}$	m_6
1	1	1	$A.B.C$	m_7

X	Y	Z	fonksiyon	
0	0	0	0	
0	0	1	1	← $\bar{X}.\bar{Y}.Z$
0	1	0	0	
1	0	0	1	← $X.\bar{Y}.\bar{Z}$
0	1	1	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	← $X.Y.Z$

$\bar{X}.\bar{Y}.Z + X.\bar{Y}.\bar{Z} + X.Y.Z$

MAXTERM

Binary değişkeni normal ve tümleyen formu düşünüldüğünde A ve B değişkenlerinin OR lojik işlemi yapılırsa 4 farklı olasılık verdi.

$\bar{A}+\bar{B}$ / $\bar{A}+B$ / $A+\bar{B}$ / $A+B$

Bu değişkenlerden her birine MAXTERM denir.

Her bir MAXTERM binary karşılığı "0" sıfır olanların normal formu ve "1" olanların tümleyeni alınarak OR işlemlerinden geçirilen N adet değişkenden oluşur.

Boolean fonksiyonları da fonksiyondaki "0" ları sonuç veren tüm MAXTERMLERİN AND'lenmesiyle elde edilir.

Bu ifadelere MAXTERMLERİN çarpımı ya da toplamların çarpımı da denir.

A	B	C	MAXTERM	Sembol
0	0	0	$A+B+C$	M_0
0	0	1	$A+B+\bar{C}$	M_1
0	1	0	$A+\bar{B}+C$	M_2
1	0	0	$\bar{A}+B+C$	M_3
0	1	1	$A+\bar{B}+\bar{C}$	M_4
1	0	1	$\bar{A}+B+\bar{C}$	M_5
1	1	0	$\bar{A}+\bar{B}+C$	M_6
1	1	1	$\bar{A}+\bar{B}+\bar{C}$	M_7

X	Y	Z	fonksiyon
0	0	0	0
0	0	1	1
0	1	0	0
1	0	0	1
0	1	1	0
1	0	1	0
1	1	0	0
1	1	1	1

$\text{fonksiyon} = (X+Y+Z).(X+\bar{Y}+Z).(X+\bar{Y}+\bar{Z}).(\bar{X}+Y+\bar{Z}).(\bar{X}+\bar{Y}+Z)$

MINTERMLERİN toplamı ve MAXTERMLERİN çarpımı ile Boolean fonksiyonlarının tam ifade edilmesi

---

Mintermlerin toplamı

Bir boolean fonksiyon Mintermler ile ifade edilmemiş ve sadeleştirilmemiş bir formda ise daha kullanışlı olduğu durumlarda, mintermlerin toplamı şekline dönüştürülebilir. Bunun için fonksiyon önce AND'lerin toplamına çevrilir daha sonra fonksiyonun bağımlı olduğu tüm değişkenler her bir AND ifadesine yerleştirilir. Eğer AND terimi örneğin X ifadesi içermiyorsa $X+X'$ ile AND'lenir.

$F = A + B'C$ mintermlerin toplamı?

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

$$(AB + AB').(C+C') =$$

$$\boxed{ABC + ABC' + AB'C + AB'C'}$$

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

$$\boxed{ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C}$$

$$\boxed{ABC + ABC' + AB'C + AB'C' + A'B'C}$$

$m_7 \quad m_6 \quad m_5 \quad m_4 \quad m_1$

$$\sum(1,4,5,6,7)$$

Maxtermlerin Çarpımı

Herhangi bir boolean fonksiyon benzer bir durumda Maxtermlerin çarpımı şekline dönüştürülebilir. Bunun için fonksiyon öncelikle dağılma yada birleşme özellikleri kullanılarak OR'ların çarpımı biçimine getirilir. Daha sonra eksik değişkenler Her bir OR ifadesine dahil edilir. OR terimi X ifadesi içermiyorsa $(X.X')$ ile OR'lanır.

$F = AB + A'C$ Maxterm çarpımı gösterimi

önce toplam çarpım şekline dönüştürülür

$$(AB + A').(AB + C)$$

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

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

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

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

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

20 μs

0 x 1023

0 2,7 SU

1023 x 2,7 = 5.x

$\frac{1023.2,7}{5}$