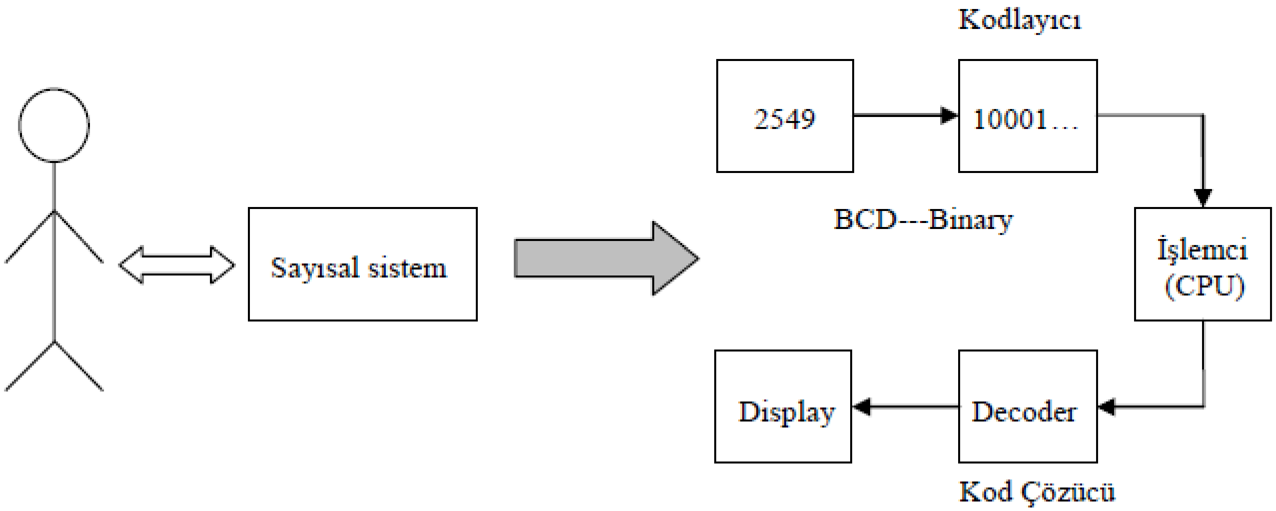
Introduction to Digital Logic

Ders 2

Kodlama, Lojik Devre Temelleri

Kodlama

- Bilginin veya verinin sayısal olarak gösterilmesi için kullanılan yöntemdir. Sonlu elemanlı bir kümenin her bir elemanına birer tane kod verilmesidir.



Sayısal Kodlama

- Sayısal kodlama, sayıların bellekte tutulma şeklini belirler, sayılar ya doğrudan ikili karşılıkları kullanılarak yada sayının her hanesindeki rakama bir kod atanarak temsil edilirler.

  Doğrudan temsil = 6 → 0110

  ASCII kodlama = A → 65

- BCD
- Üç Fazlalık
- Aiken
- Bitişik Kodlar (Gray)

BCD - Binary Coded Decimal

Ondalık sayının her hanesinin ikili olarak kodlanmasıdır. Bir taban dönüşümü değildir. Örneğin $(24)_{10} \rightarrow (11000)_2$ bu bir ikili taban karşılığıdır. Ancak BCD karşılığı $(0010\ 0100)_2$ olarak yazılır. BCD kodlamada 0-9 arası sayılar kullanılır. 10-15 arası kullanılmamaktadır. Dolayısıyla BCD kodlama artıklı bir kod olarak karşımıza çıkmaktadır.

$$(25)_{10} \rightarrow (0010\ 0101)_2$$

$$(63)_{10} \rightarrow (0110\ 0011)_2$$

Üç Fazlalık Kodu

İkili sistemin üç fazlası alınarak oluşturulan kodlama işlemidir. Ağırlığı olmayan simetrik bir koddur.

Sayı	3-Fazlalık Kodu
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

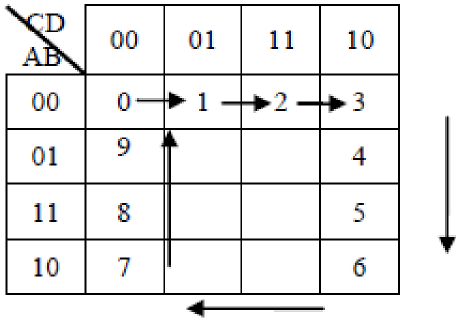
Aiken Kodu

Tabloda verildiği gibi, 0-9 arasındaki sayıların ilk beş ve son beş rakamlarının ikili karşılıklarından oluşur, simetrik bir koddur. Bu kodlama türünün özelliği (0-4) arasındaki ilk beş sayının bilinen ikili kodlamaya eşdeğer olduğu, (5-9) arasındaki ikinci beş sayının ise ilk beş sayının 1' e tümleyeni olduğu söylenebilir.

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

Bitişik Kodlar ve Gray Kodu

Birbirini izleyen sayılara karşılık alınan ikili kod sözcükleri arasındaki uzaklık (Hamming uzaklığı) “1” ise bu tür kodlara “Bitişik Kodlar” adı verilir. Ayrıca kod sözcüklerinin birincisi ile sonuncusu arasındaki uzaklık yine “1” ise bu tür kodlamalara “Çevrimli Bitişik Kodlar” denir. Örneğin, 0’ dan 3’ e kadar sayılar kodlamada 00, 01, 11, 10 sözcükleri kullanılırsa bitişik kodlama yapılmış olur.



Sayı	Kod Sözcüğü (ABCD)
0	0000
1	0001
2	0011
3	0010
4	0110
5	1110
6	1010
7	1000
8	1100
9	0100

Gray Kodu

2^n elemanlı bir küme için 2 tabanında artıksız ve çevrimli bir kodlama yapılırsa yansımalı bir kod yani “Gray Kodu” elde edilir. Sayma işleminde ve sütun tarama işlemlerinde kullanılır. Gray kodu Karnaugh diyagramının geçişlerinde kullanılacaktır.

Sayı	İkili 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
13	1101	1011
14	1110	1001
15	1111	1000

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

LOJİK DEVRE TEMELLERİ

Lojik devreler ikili işaretler veya ikili kodlanmış veriler üzerinde çalışan ve temeli Boole cebrine dayanan düzeneklerdir. Lojik devrelerde biri "lojik 0" diğeri "lojik 1" olarak adlandırılan iki durum vardır. Bilgisayarlar dahil tüm sayısal sistemler bu iki lojik değerin farklı şekilde kombinasyonları yapılarak tasarlanır.

Lojik devre temellerine ait temel lojik işlemler (VE, VEYA, TÜMLEME vb.)
Venn Diyagramları
Boole Cebri aksiyom ve teoremleri
Boole Cebrinin kanomik biçimleri (Minimum Terimler (MinTerm), Maksimum Terimler (MaksTerm))
Lojik Kapı elemanları ile devre gerçekleştirme

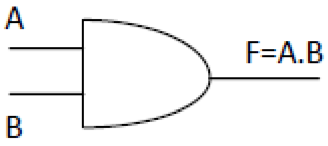
Lojik İşlemlerin Temeli

- Lojik işlemlerin temeli, çoğu geleneksel cebire benzeyen ama kendine has aksiyom ve teoremleri olan ***Boole Cebri*** üzerine kurulmuştur.
- Boole cebri işlemlerinin görsel olarak ifade edilmesi Venn Diagramları aracılığı ile yapılır.
- Boole cebri içinde VE, VEYA, TÜMLEME olmak üzere 3 temel lojik işlem tanımlanmıştır.
- Doğruluk Tablosu, Anahtar devresi, Lojik devre gösterilişleri, Zamanlama diyagramları Lojik Devre tasarımında kullanılacak en önemli araçlar olacaktır.

Ve İşlemi (AND)

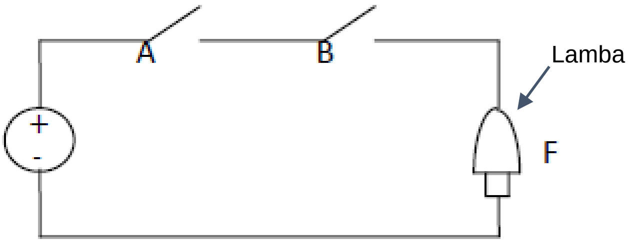
Birbirine VE işlemi ile bağlı iki önermeden oluşan bir birleşik önermenin doğru olması, her iki önermenin de doğru olmasına bağlıdır.

Lojik Devre Gösterilişi

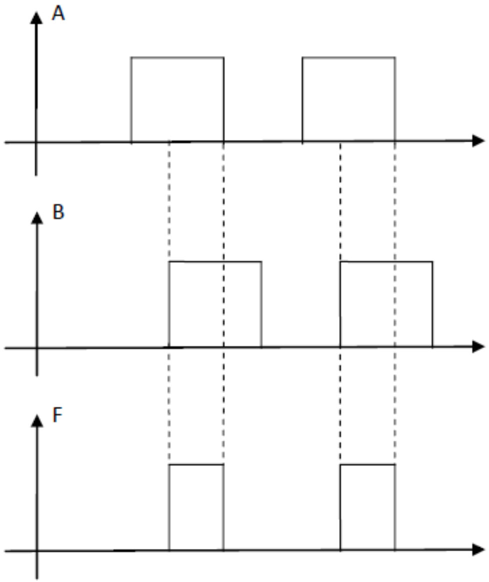


A	B	$F = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

Doğruluk Tablosu



Anahtar Devresi

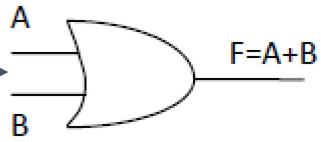


Zamanlama Diyagramı

Veya İşlemi (OR)

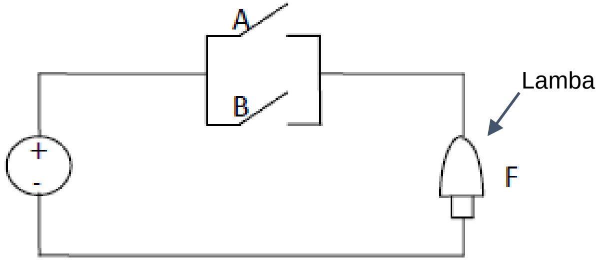
Birbirine VEYA işlemi ile bağlı iki önermeden oluşan bir birleşik önermenin doğru olması, birleşik önermeyi meydana getiren önermelerden en az birinin doğru olmasına bağlıdır.

Lojik Devre Gösterilişi

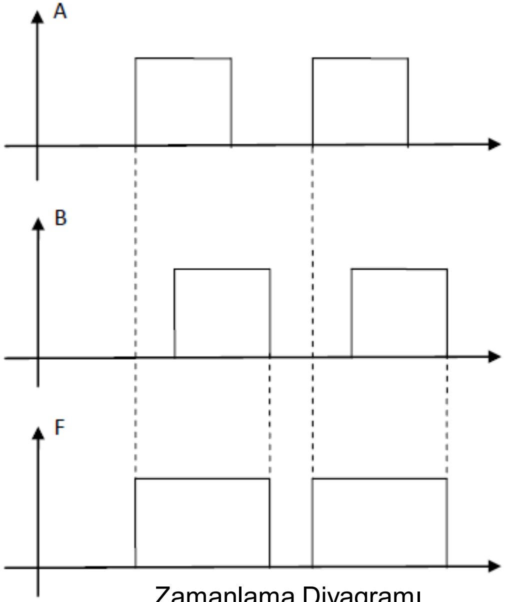


A	B	$F = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Doğruluk Tablosu



Anahtar Devresi

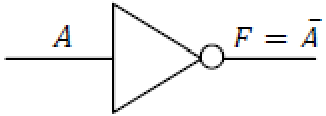


Zamanlama Diyagramı

Tümleme işlemi (NOT)

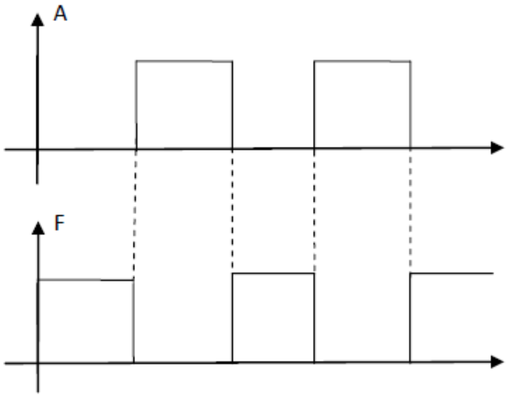
"NOT" işlemi uygulanan önerme, başlangıçta doğru ise yanlış, yanlış ise doğru olacaktır.

Lojik Devre Gösterilişi



A	$F = \bar{A}$
0	1
1	0

Doğruluk Tablosu



Zamanlama Diyagramı

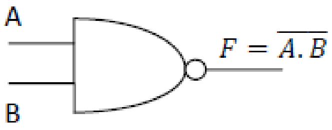
Anahtar Devresi

Tümleyen Ve işlemi (NAND)

Türetilen işlemler

Birbirine VE DEĞİL işlemi ile bağlı iki önermeden oluşan bir birleşik önermenin yanlış olması, her iki önermenin de doğru olmasına bağlıdır.

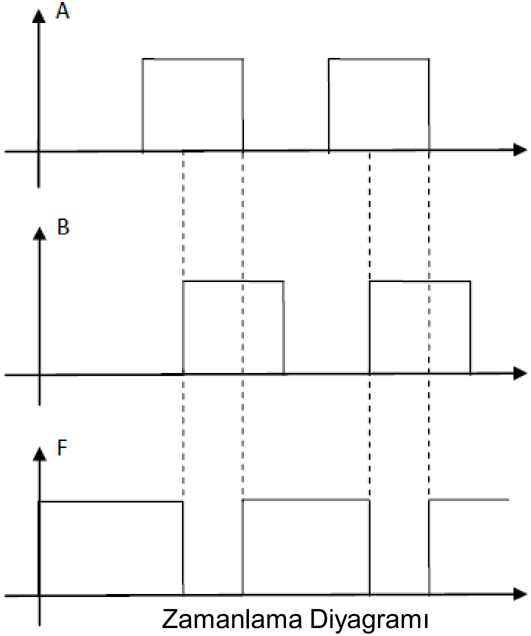
Lojik Devre Gösterilişi



A	B	$F = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Doğruluk Tablosu

Anahtar Devresi



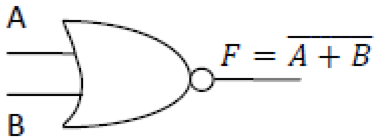
Zamanlama Diyagramı

Tümleyen Veya işlemi (NOR)

Türetilen işlemler

Birbirine VEYA DEĞİL işlemi ile bağlı iki önermeden oluşan bir birleşik önermenin doğru olması, her iki önermenin de yanlış olmasına bağlıdır.

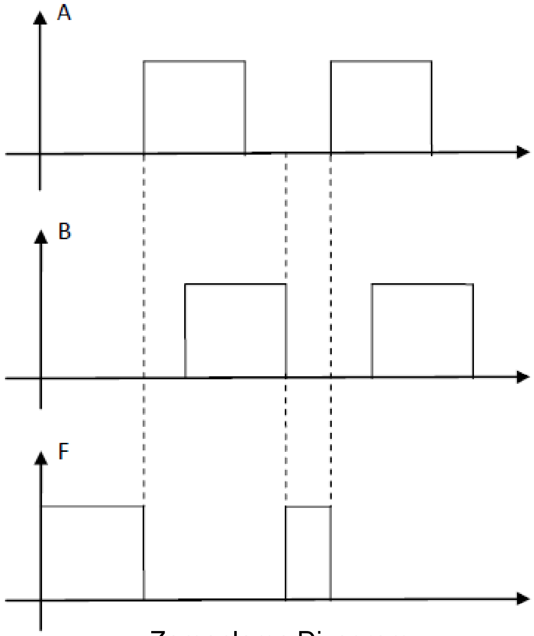
Lojik Devre Gösterilişi



A	B	$F = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Doğruluk Tablosu

Anahtar Devresi



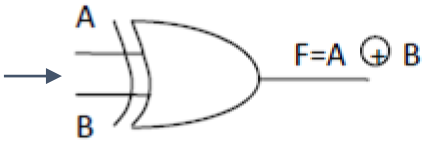
Zamanlama Diyagramı

YA DA - Özel VEYA işlemi (XOR)

Türetilen işlemler

Birbirine ÖZEL VEYA işlemi ile bağlı iki önermeden oluşan bir birleşik önermenin doğru olması, birleşik önermeyi meydana getiren önermelerden birinin doğru diğerinin yanlış olmasına bağlıdır.

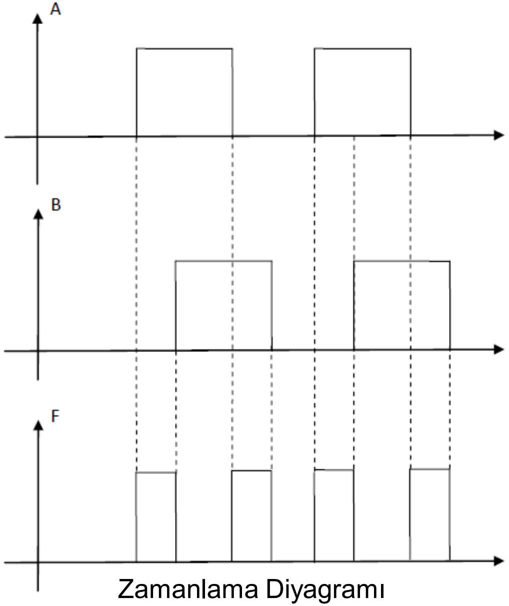
Lojik Devre Gösterilişi



A	B	$F = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Doğruluk Tablosu

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



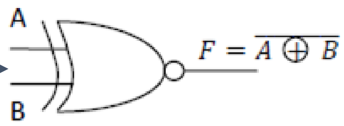
Zamanlama Diyagramı

TYA DA - Özel VEYA DEĞİL işlemi (XNOR)

Türetilen işlemler

Birbirine ÖZEL VEYA DEĞİL işlemi ile bağlı iki önermeden oluşan bir birleşik önermenin doğru olması, her iki önermenin de yanlış olmasına veya her iki önermenin de doğru olmasına bağlıdır.

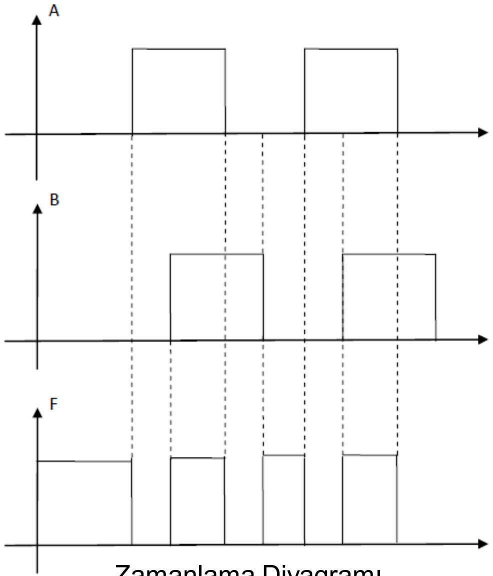
Lojik Devre Gösterilişi



A	B	$F = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

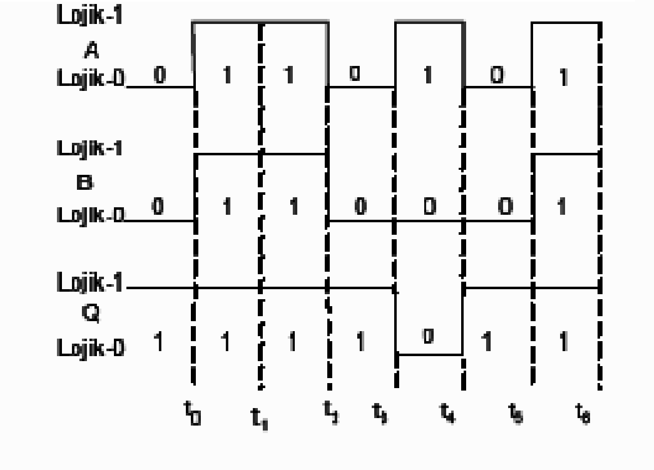
Doğruluk Tablosu

$$F = A \odot B = AB + \bar{A}\bar{B}$$

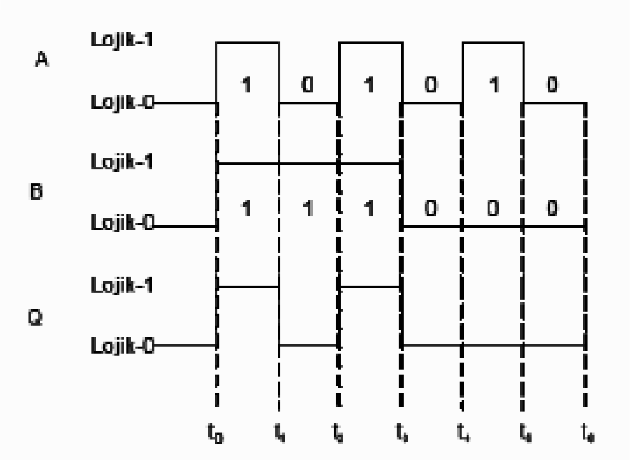


Zamanlama Diyagramı

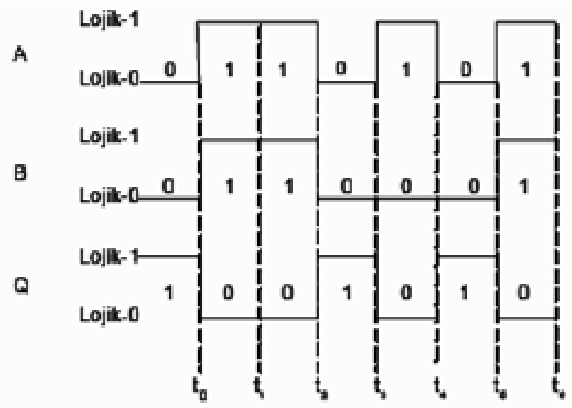
Sorular



ÖZEL VEYA DEĞİL



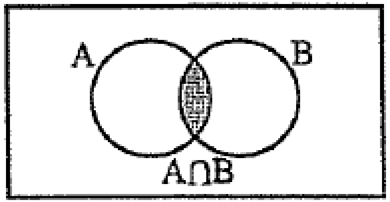
VE



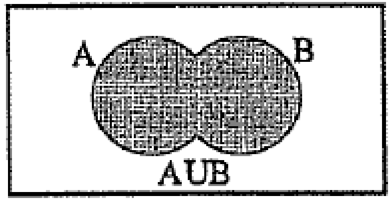
VEYA DEĞİL

Venn Diyagramları

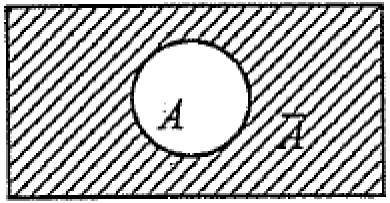
Kesişim



Birleşim



Tümleme



Diğer Kavramlar

$$A + \overline{A} = 1$$
$$A \cdot \overline{A} = 0$$
$$1 + A = 1$$
$$1 \cdot A = A$$
$$0 + A = A$$
$$0 \cdot A = 0$$

Lojik Diyagramların Tasarımları

$Q = A.B + B.\overline{C}.D$ ifadesinin lojik diyagramını çiziniz.

Lojik Diyagramların Tasarımları

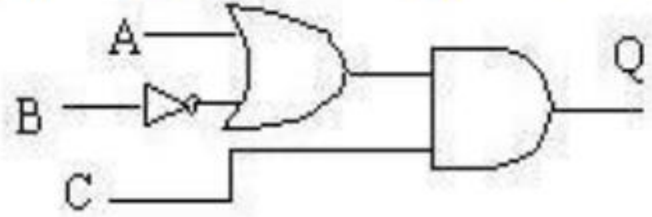
$Q = [A.(\overline{B} + \overline{C})] + \overline{A.B}$ ifadesinin lojik diyagramını çiziniz.

Lojik Diyagramların Tasarımları

$Q = (A + \overline{B}) \cdot (\overline{C + D}) + \overline{A \cdot B}$ ifadesinin lojik diyagramını çiziniz.

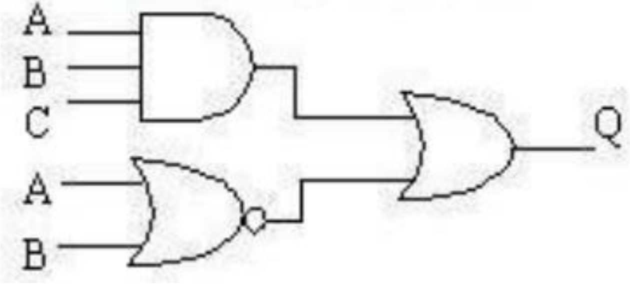
Lojik Diyagramlardan Lojik İfadelerin Bulunması

Örnek: Aşağıdaki lojik diyagramın lojik ifadesini yazınız.



Lojik Diyagramlardan Lojik İfadelerin Bulunması

Örnek: Aşağıdaki lojik diyagramın lojik ifadesini yazınız.



Boole Cebri Aksiyom ve Teoremleri

Her disiplinde olduğu gibi lojik devre tasarımı da belirli kurallar çerçevesinde yapılır. Lojik devrelerde bu kurallar kümesinin dayanağı Boole Cebri' dir. Bütün cebirsel yapılarda olduğu gibi Boole Cebri' nde de doğru olarak kabul edilen ve doğruluğu ispatlanabilen önermeler olmak üzere iki temel kurallar dizisi vardır. Doğru olarak kabul edilen önermelere "*aksiyom*", doğruluğu ispatlanabilen önermelere ise "*teorem*" adı verilir.

Aksiyomlar

"0" ve "1" ikilisinden oluşan bir B kümesine "+" ve "·" işlemleri uygulanmış olsun.

1. Her bir değişken "0" veya "1" değerinden sadece birini alabilir. Değişken "1" değerini almıyor ise değeri "0" dır.

$$a \neq 0 \Longrightarrow a = 1$$

$$a \neq 1 \Longrightarrow a = 0$$

2. a) $1 + 1 = 1$ Birbirine VEYA ile bağlı iki önermenin ikisi de doğru ise birleşik önerme de doğrudur.

   b) $0 \cdot 0 = 0$ Birbirine VE ile bağlı iki önermenin ikisi de yanlış ise birleşik önerme de yanlıştır.

3. a) $0 + 0 = 0$ Birbirine VEYA ile bağlı iki önermenin ikisi de yanlış ise birleşik önerme de yanlıştır.

   b) $1 \cdot 1 = 1$ Birbirine VE ile bağlı iki önermenin ikisi de doğru ise birleşik önerme de doğrudur.

4. a) $1 + 0 = 1$ Birbirine VEYA ile bağlı iki önermeden biri doğru ise birleşik önerme de doğrudur.

   b) $0 \cdot 1 = 0$ Birbirine VE ile bağlı iki önermeden birisi yanlış ise birleşik önerme de yanlıştır.

Teoremler

1. a) $a + b = b + a$ — Değişme Özelliği

   $b)\ a \cdot b = b \cdot a$

2. a) $a + b + c = (a + b) + c = a + (b + c)$ — Birleşme Özelliği

   $b)\ a \cdot b \cdot c = (a \cdot b) \cdot c = a \cdot (b \cdot c)$

3. a) $a + b \cdot c = (a + b) \cdot (a + c)$ — Dağılma Özelliği

   $b)\ a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

4. a) $a + a = a$ — Değişkende Fazlalık Özelliği

   $b)\ a \cdot a = a$

5. a) $a + a.b = a$ — Yutma Özelliği

   $b)\ a \cdot (a + b) = a$

6. a) $\overline{(a)} = \bar{a}$ — İşlemde Fazlalık Özelliği

   $b)\ \overline{(\bar{a})} = a$

Teoremler

7. a) $\overline{(a + b + c + \cdots)} = \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \cdots$ De Morgan Kuralı

   $b)$ $\overline{(a \cdot b \cdot c \cdot \cdots)} = \bar{a} + \bar{b} + \bar{c} + \cdots$

8. a) $a + \bar{a} = 1$ Sabit Özelliği

   $b)$ $a \cdot \bar{a} = 0$

9. a) $0 + a = a$ Etkisizlik Özelliği

   $b)$ $1 \cdot a = a$

10. a) $1 + a = 1$ Yutan Sabit Özelliği

    $b)$ $0 \cdot a = 0$

11. a) $(a + \bar{b}) \cdot b = a \cdot b$

    $b)$ $a \cdot \bar{b} + b = a + b$

12. a) $(a + b) \cdot (\bar{a} + c) \cdot (b + c) = (a + b) \cdot (\bar{a} + c)$

    $b)$ $a \cdot b + \bar{a} \cdot c + b \cdot c = a \cdot b + \bar{a} \cdot c$

13. a) $(a + b) \cdot (\bar{a} + c) = a \cdot c + \bar{a} \cdot b$

    $b)$ $a \cdot b + \bar{a} \cdot c = (a + c) \cdot (\bar{a} + b)$

Örnek

Örnek 5.1. $F = A.(\overline{B+C}) + \bar{A} + (\bar{B} \cdot C) \Longrightarrow \bar{F} = ?$

$$\bar{F} = \overline{A \cdot (\overline{B+C}) + \bar{A} + (\bar{B} \cdot C)}$$

$$\bar{F} = \bar{A} + \overline{(\overline{B+C})} \cdot \overline{(\bar{A})} \cdot \overline{(\bar{B} \cdot C)}$$

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

$$\bar{F} = (\bar{A} + B + C) \cdot (AB + A\bar{C})$$

Örnek 5.2. $F_1 = \bar{x}yz + \bar{x}\bar{y}z \;\; F_2 = x(\bar{y}\bar{z} + yz)$

Verilen fonksiyonların terslerini bulunuz.

$$\overline{F_1} = \overline{\bar{x}yz + \bar{x}\bar{y}z} = \overline{(\bar{x}yz)} \cdot \overline{(\bar{x}\bar{y}z)} = (x + \bar{y} + \bar{z}) \cdot (x + y + \bar{z})$$

$$\overline{F_2} = \overline{x(\bar{y}\bar{z} + yz)} = \bar{x} + \big((y + z) \cdot (\bar{y} + \bar{z})\big)$$

Örnek

Örnek: $Q = B.C+B.(C+A)+C.(B+A)$ ifadesini Boolean teoremleri yardımı ile sadeleştiriniz

Dualite Prensibi

Bir Boole ifadesinin duali mantıksal çarpım ve toplamların ve 1 ile 0' ların yer değiştirmesiyle bulunur.

$x(y+0)$ ifadesinin duali $x+(y\cdot 1)$ olarak bulunur.

$\bar{x}\cdot 1+(\bar{y}\cdot z)$ ifadesinin duali $(\bar{x}+0)\cdot(\bar{y}+z)$ olarak bulunur.

Circuit Optimization

- **Goal: To obtain the simplest implementation for a given function**
- **Optimization is a more formal approach to simplification that is performed using a specific procedure or algorithm**
- **Optimization requires a cost criterion to measure the simplicity of a circuit**
- **Two distinct cost criteria we will use:**
  - **Literal cost (L)**
  - **Gate input cost (G)**
  - **Gate input cost with NOTs (GN)**

Literal Cost

- **Literal – a variable or its complement**
- **Literal cost – the number of literal appearances in a Boolean expression corresponding to the logic circuit diagram**
- **Examples:**
  - $F = BD + A\overline{B}C + A\overline{C}\overline{D}$ $\quad L = 8$
  - $F = BD + A\overline{B}C + A\overline{B}\overline{D} + AB\overline{C}$ $\quad L =$
  - $F = (A + B)(A + D)(B + C + \overline{D})(\overline{B} + \overline{C} + D)$ $\quad L =$
  - **Which solution is best?**

Gate Input Cost

- **Gate input costs - the number of inputs to the gates in the implementation corresponding exactly to the given equation or equations.** **(G - inverters not counted, GN - inverters counted)**
- **For SOP and POS equations, it can be found from the equation(s) by finding the sum of:**
  - **all literal appearances**
  - **the number of terms excluding terms consisting only of a single literal,(G) and**
  - **optionally, the number of distinct complemented single literals (GN).**
- **Example:**
  - $F = BD + A\bar{B}C + A\bar{C}\bar{D}$ **G = 11, GN = 14**
  - $F = BD + A\bar{B}C + A\bar{B}\bar{D} + AB\bar{C}$ **G = , GN =**
  - $F = (A + \bar{B})(A + D)(B + \bar{C} + D)(\bar{B} + \bar{C} + D)$ **G = , GN =**
  - **Which solution is best?**

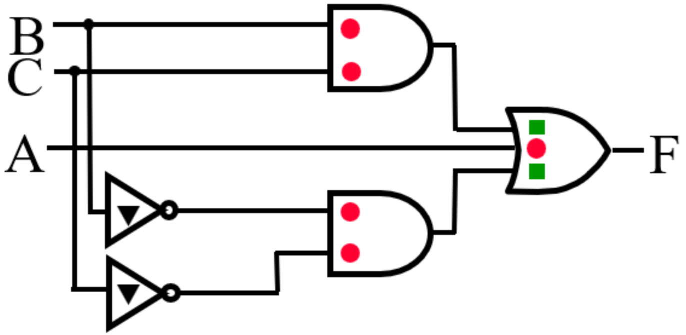
Cost Criteria (continued)

- **Example 1:**
- $\mathbf{F = A + B\,C + \overline{B}\,\overline{C}}$

$\mathbf{GN = G + 2 = 9}$

$\mathbf{L = 5}$

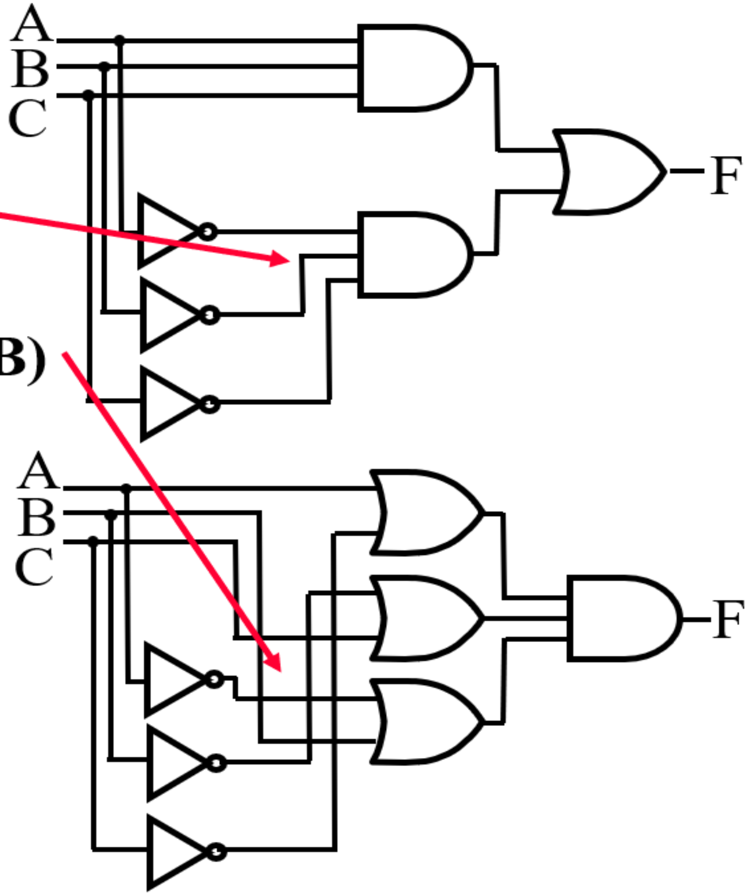
$\mathbf{G = L + 2 = 7}$



- **L (literal count) counts the AND inputs and the single literal OR input.**
- **G (gate input count) adds the remaining OR gate inputs**
- **GN(gate input count with NOTs) adds the inverter inputs**

Cost Criteria (continued)

- **Example 2:**
- $F = A\,B\,C + \bar{A}\,\bar{B}\,\bar{C}$
- $L = 6 \quad G = 8 \quad GN = 11$
- $F = (A + \bar{C})(\bar{B} + C)(\bar{A} + B)$
- $L = 6 \quad G = 9 \quad GN = 12$
- **<u>Same</u> function and <u>same</u> literal cost**
- **But first circuit has <u>better</u> gate input count and <u>better</u> gate input count with NOTs**
- **Select it!**



Boolean Function Optimization

- **Minimizing the gate input (or literal) cost of a (a set of) Boolean equation(s) reduces circuit cost.**
- **We choose gate input cost.**
- **Boolean Algebra and graphical techniques are tools to minimize cost criteria values.**
- **Some important questions:**
  - **When do we stop trying to reduce the cost?**
  - **Do we know when we have a minimum cost?**
- **Treat optimum or near-optimum cost functions for two-level (SOP and POS) circuits first.**
- **Introduce a graphical technique using Karnaugh maps (K-maps, for short)**

Minimum - Maksimum Terimler