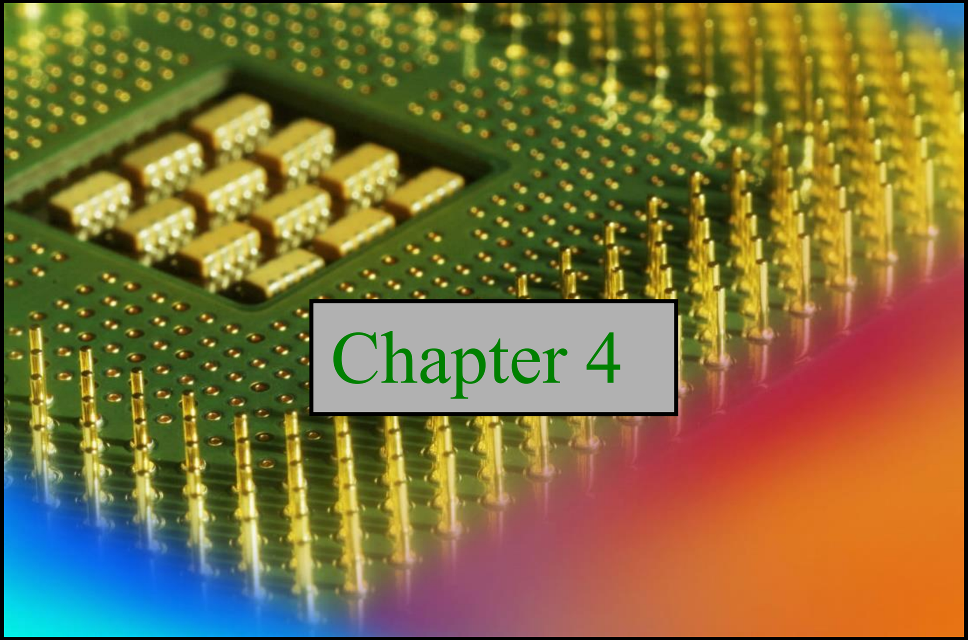
# Digital Fundamentals

Tenth Edition

Floyd



## Boolean Cebiri (Aritmetiği)

Boole cebirinde değişken, bir eylemi, koşulu veya verileri temsil etmek için kullanılan bir semboldür. Tek bir değişken yalnızca 1 veya 0 değerine sahip olabilir.

Tümleyen, bir değişkenin tersini temsil eder ve bir üst çubuk ile gösterilir. Böylece, $A$'nın tümleyeni $\overline{A}$'dır.

Toplama işareti OR işlemini gösterir. Ama bu işlem matematikteki toplama değildir. Bir veya daha fazla, değişken değeri 1 ise, sonuç terimi 1'dir. Sonuç terimi yalnızca her değer 0 ise sıfırdır.

**Örnek:** $A$, $B$, ve $C$ değişkenleri için $\overline{A} + B + \overline{C} = 0$ Yapan değerleri bulun.

**Çözüm:** Bu eşitlikteki her değişken = 0 olmalıdır; Dolayısıyla A = 1, B = 0 ve C = 1 olursa eşitlik sağlanır.

Boole cebirinde çarpma, AND işlemine eşdeğerdir. Değişkenlerin çarpımında tüm değişkenler 1 ise sonuç 1 olacaktır.

**Örnek:** $A.\overline{B}.\overline{C} = 1$ ise *A, B ve C* 'nin değerleri ne olmalıdır?

**Çözüm:** Çarpımdaki her değişken = 1 olmalıdır; bu nedenle A = 1, B = 0 ve C = 0 olmalı.

## Değişme Özelliği

Değişme özelliği toplama ve çarpmaya uygulanır. Toplama için değişme özelliğinde;

**Sonuç açısından, değişkenlerin OR ile verildiği sıra hiçbir fark yaratmaz.**

$$\boxed{A + B = B + A}$$

Çarpma için değişme özelliğinde;

**Sonuç açısından, değişkenlerin AND ile verildiği sıra bir fark yaratmaz.**

$$\boxed{AB = BA}$$

## Birleşme Özelliği

Birleşme özelliği aynı zamanda toplama ve çarpmaya da uygulanır. Toplama için birleşme özelliğinde;

**İkiden fazla değişkeni OR işlemi yaparken, değişkenlerin gruplandırılmasına bakılmaksızın sonuç aynıdır.**

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

Çarpma için birleşme özelliğinde;

**İkiden fazla değişken ile AND işlemi yapılmışsa, değişkenlerin gruplandırılmasına bakılmaksızın sonuç aynıdır.**

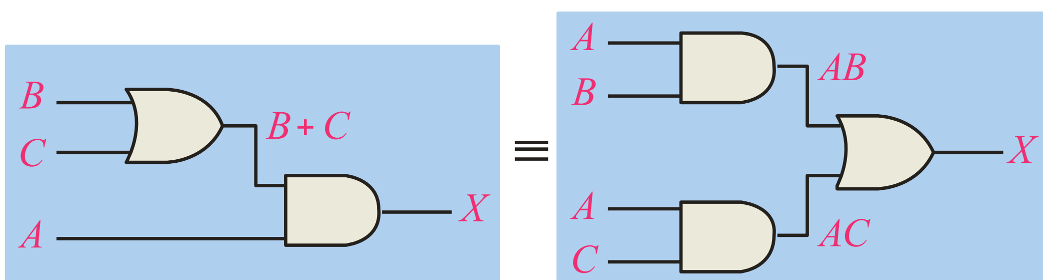
$$\boxed{A(BC) = (AB)C}$$

## Dağılma Özelliği

Bir ortak değişken, sıradan cebirde olduğu gibi bir ifadeden çarpanlarına ayrılabilir. Yani;

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

Dağılma özelliği, eşdeğer devrelerle gösterilebilir:




$A(B + C)$ $\qquad$ $AB + AC$

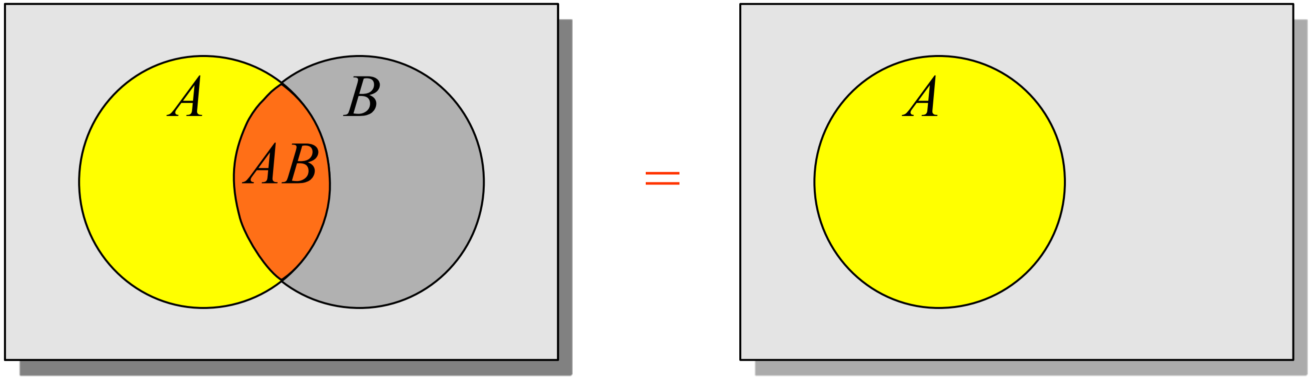
# Boolean Cebirinin kuralları

1. $A + 0 = A$
2. $A + 1 = 1$
3. $A \cdot 0 = 0$
4. $A \cdot 1 = 1$
5. $A + A = A$
6. $A + \overline{A} = 1$
7. $A \cdot A = A$
8. $A \cdot \overline{A} = 0$
9. $\overline{\overline{A}} = A$
10. $A + AB = A$
11. $A + \overline{A}B = A + B$
12. $(A + B)(A + C) = A + BC$

# Boolean Cebirinin kuralları

Boolean cebirinin kuralları *Venn* diagramlarıyla (kümelerle) gösterilebilir. *A* değişkeni bir küme göstersin.

$A + AB = A$ olduğu bu diyagramlarla kolayca gösterilebilir. *B* değişkeni ile kesişim alanı *AB* ile gösterilebilir. Dolayısı ile $A + AB = A$



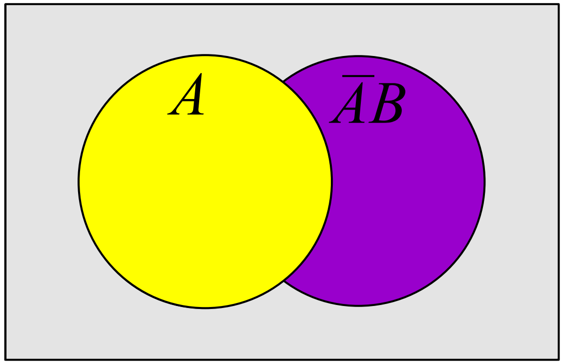
# Boolean Cebirinin kuralları

**Örnek:** $A + \overline{A}B = A + B$ olduğunu Venn diagramı ile gösterelim.

**Çözüm:**

Bu defa , $\overline{A}B$ yi eflatun renkle gösterelim.

$A + \overline{A}B = A + B$ olduğu kolayca görülebilir.



## Boolean Cebirinin kuralları

(A + B) (A + C) = A + BC olduğunu belirten Kural 12, aşağıdaki gibi daha önceki kurallar uygulanarak kanıtlanabilir:

$$\begin{aligned}(A + B)(A + C) &= AA + AC + AB + BC \\ &= A + AC + AB + BC \\ &= A(1 + C + B) + BC \\ &= A \cdot 1 + BC \\ &= A + BC\end{aligned}$$

Bu çözüm biraz karmaşıktır, ancak aşağıdaki slaytta verildiği gibi bir Venn diyagramı ile de gösterilebilir ...

Üç alan A, B ve C değişkenlerini temsil eder.

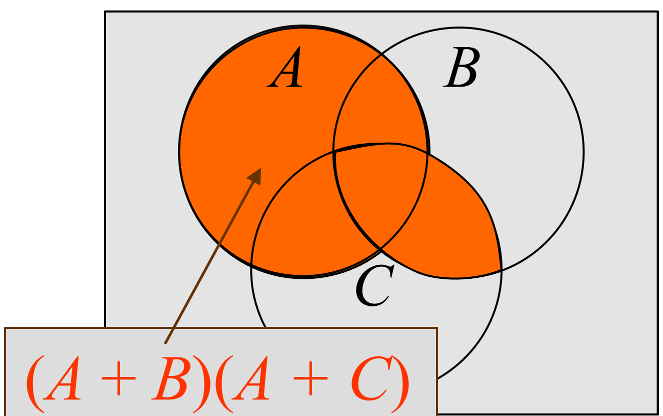
A + B'yi temsil eden alan sarı ile gösterilmiştir.

A + C'yi temsil eden alan kırmızıyla gösterilir.

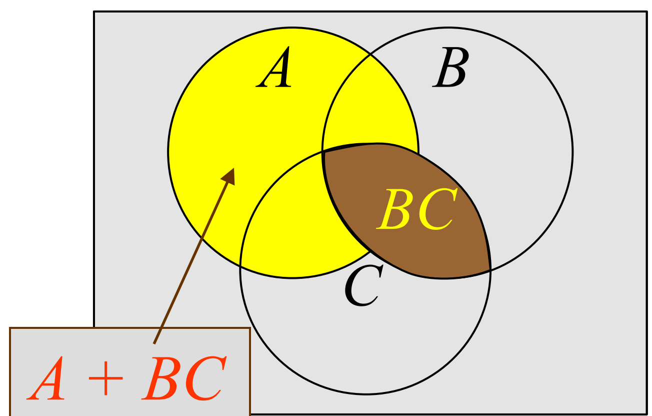
Kırmızı ve sarının örtüşmesi turuncu renkte gösterilir.

B ve C arasındaki örtüşen alan BC'yi temsil eder.

A ile OR işlemi öncekiyle aynı alanı verir.




=



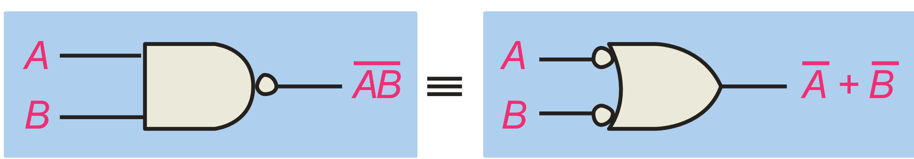
# DeMorgan Teoremi

## DeMorgan'ın 1nci Teoremi

**Değişkenlerin çarpımlarının tümleyeni (değili), değişkenlerin ayrı ayrı tümleyenlerinin (değillerinin) toplamına eşittir.**

$$\overline{AB} = \overline{A} + \overline{B}$$

DeMorgan'ın ilk teoremini kapılara uygularsak:




NAND Negative-OR

| Inputs | | Output | |
|---|---|---|---|
| $A$ | $B$ | $\overline{AB}$ | $\overline{A} + \overline{B}$ |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

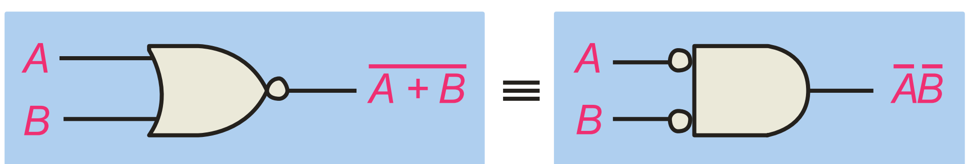
# DeMorgan Teoremi

DeMorgan'ın 2nci Teoremi

**Değişkenlerin toplamlarının tümleyeni (değili), değişkenlerin ayrı ayrı tümleyenlerinin (değillerinin) çarpımına eşittir.**

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

DeMorgan'ın ikinci teoremini kapılara uygularsak:




NOR    Negative-AND

| Inputs | | Output | |
|---|---|---|---|
| $A$ | $B$ | $\overline{A + B}$ | $\overline{A}\overline{B}$ |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |

## DeMorgan Teoremi

**Ör:**

Aşağıdaki ifadeye DeMorgan teoremini uygulayalım.

$$X = \overline{\overline{C} + D}.$$

**Çözüm:**

Önce en dıştaki değile teoremi uygulayalım

$X = \overline{\overline{C}} \cdot \overline{D}.$

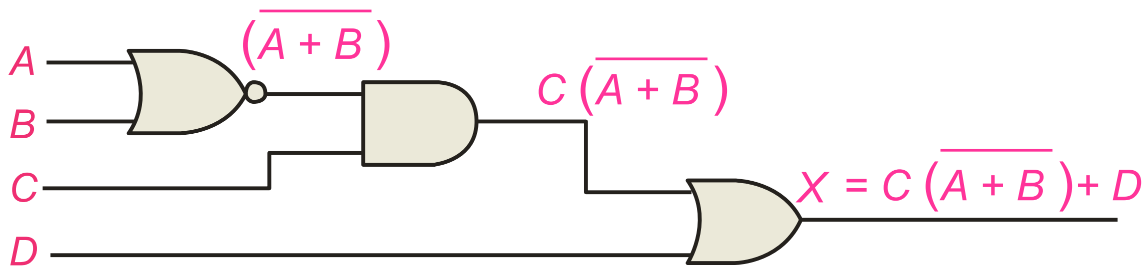
Değiller birbirini götürürse $X = C \cdot \overline{D}$ olur.

# Mantık devresinin Boolean Cebiri ile analizi

Mantık devreleri, her bir kapı için ifade yazarak ve ifadeleri Boolean cebri kurallarına göre birleştirerek analiz edilebilir.

**Örnek:** X çıkışını sadeleştirmek için Boolean cebirini uygulayın.

**Çözüm:** Her kapı için ifadeyi yazın:




DeMorgan teoremini ve dağılma kuralını uygularsak:

$$X = C\,(\overline{A}\;\overline{B}) + D = \overline{A}\,\overline{B}\,C + D$$

# SOP ve POS formları

Boole ifadeleri, Çarpımların Toplamı (**sum-of-products**/SOP) veya toplamların çarpımı (**product-of-sums** / POS) formunda yazılabilir. Bu formlar, özellikle PLD'ler ile kombinasyonel mantığın uygulanmasını basitleştirebilir. Her iki biçimde de, bir her değişken yada değişkenin değili çarpım yada toplam halinde ayrı ayrı yazılır. Yani parantez içindeki ifadenin toplam değili formunda bulunmaz.

İki veya daha fazla çarpım terimi aşağıdaki örneklerde olduğu gibi toplandığında bir ifade SOP formundadır:

$$\overline{A}\,\overline{B}\,\overline{C} + A\,B \qquad A\,B\,\overline{C} + \overline{C}\,\overline{D} \qquad C\,D + \overline{E}$$

Aşağıdaki örneklerde olduğu gibi iki veya daha fazla toplam terimi çarpıldığında bir ifade POS biçimindedir:

$$(A + B)(\overline{A} + C) \qquad (A + B + \overline{C})(B + D) \qquad (\overline{A} + B)C$$

## SOP Standard formu

SOP standart formunda, kullanılan tüm değişkenler her terimde görünmelidir. Bu form doğruluk tabloları oluşturmak veya PLD'lerde mantık devresi kurmak için daha kullanışlıdır.

Standart olmayan bir terimi, terimi eksik değişken ve onun tümleyeninin toplamından oluşan bir terimle çarparak standart biçime dönüştürebilirsiniz.

**Örnek:** $X = \overline{A}\,\overline{B} + A\,B\,C$'yi standart forma dönüştürün.

**Çözüm:** İlk terim C değişkenini içermiyor. Bu nedenle, ilk terimi $(C + \overline{C}) = 1$ terimi ile çarpalım. Böylece standart forma dönüşür.

$$X = \overline{A}\,\overline{B}\,(C + \overline{C}) + A\,B\,C$$
$$= \overline{A}\,\overline{B}\,C + \overline{A}\,\overline{B}\,\overline{C} + A\,B\,C$$

# POS Standard formu

**POS standart formunda**, kullanılan tüm değişkenler her terimde görünmelidir. Bu form doğruluk tabloları oluşturmak veya PLD'lerde mantık devresi kurmak için daha kullanışlıdır.

Standart olmayan bir POS ifadesini, eksik değişkenin ve tümleyeninin çarpımını ekleyerek ve (A + B) (A + C) = A + BC olduğunu belirten kural 12'yi uygulayarak standart forma genişletebilirsiniz.

**Örnek:** $X = (\overline{A} + \overline{B})(A + B + C)$ ifadesini standart forma dönüştürelim .

**Çözüm:** İlk toplam ifadesi $C$ terimini içermiyor. Dolayısı ile bu toplama, $C\overline{C}$ eklersek sonuç değişmez.

$$X = (\overline{A} + \overline{B} + C\overline{C})(A + B + C)$$

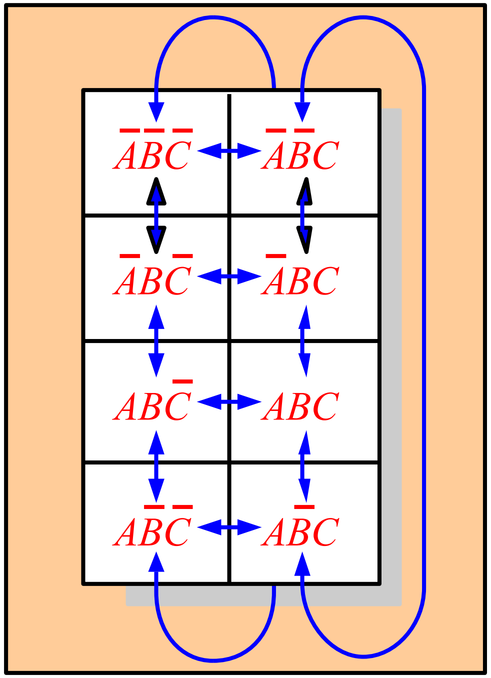
$$= (\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C})(A + B + C)$$

## Karnaugh Haritaları

Karnaugh haritası (K-map), 3 veya 4 değişkenli mantık devrelerini basitleştirmek için bir araçtır. 3 değişken için 8 hücre gereklidir ($2^3$).

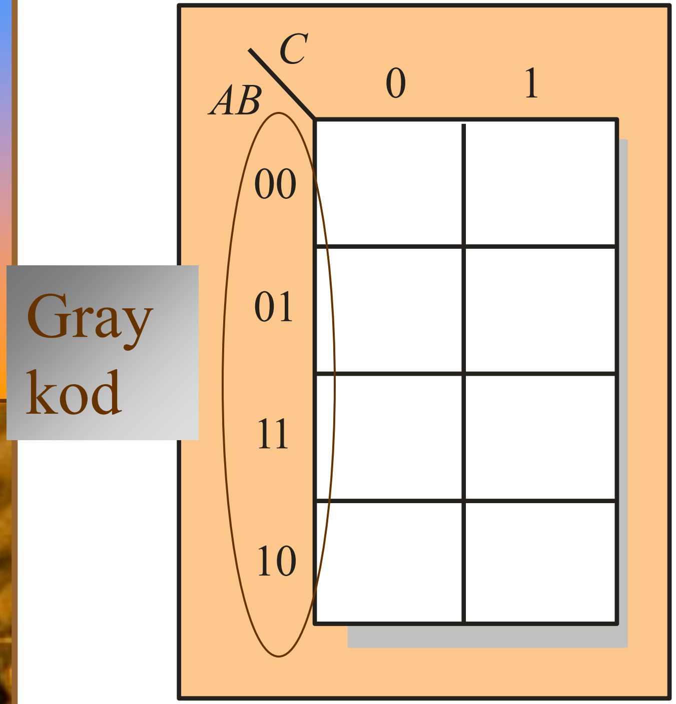
Gösterilen harita, A, B ve C olarak etiketlenmiş üç değişken içindir. Her hücre bir olası çarpanı terimini temsil eder.

Her hücre, bitişik bir hücreden yalnızca bir değişkenle farklılık gösterir.



# Karnaugh Haritaları

Hücreler, değişkeni ve onun tümleyenini temsil etmek için genellikle 1'ler ve 0'lar kullanılarak etiketlenir.



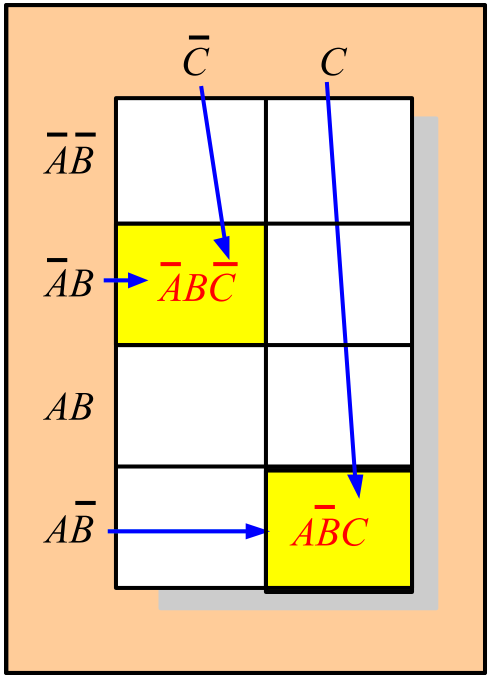

Sayılar, bitişik hücrelerin yalnızca bir değişkenle farklı olmasını sağlamak için gray kodla girilir.

Birler gerçek değişken olarak ve sıfırlar tümleyen değişken olarak okunur.

## Karnaugh Haritaları

Alternatif olarak, hücreler değişkenlerle etiketlenebilir. Bu, okumayı kolaylaştırır, ancak haritayı hazırlamak daha fazla zaman alır.

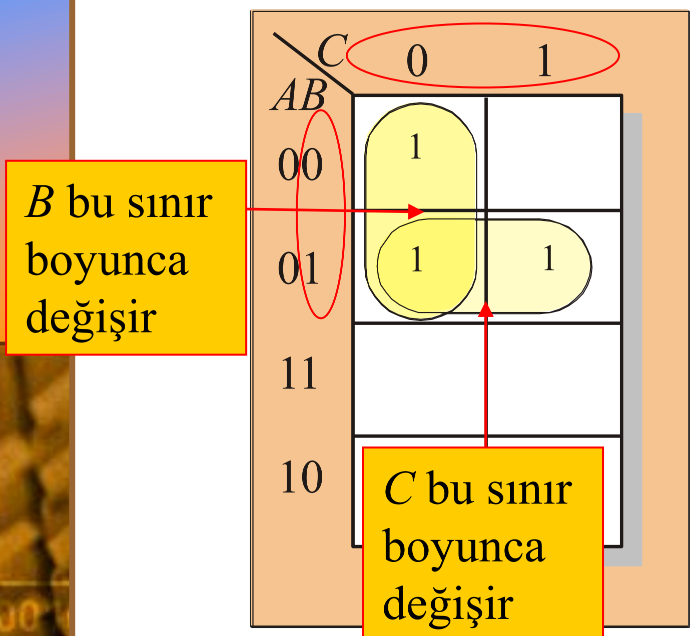
Sarı hücreler için terimler $\overline{A}B\overline{C}$ ve $A\overline{B}C$



# Karnaugh karitaları

K-haritaları, hücreleri gruplayarak ve değişen değişkenleri ortadan kaldırarak mantık devresini basitleştirebilir.

**Örnek:** 1'leri haritada gruplayın ve en sade fonksiyonu elde edin.



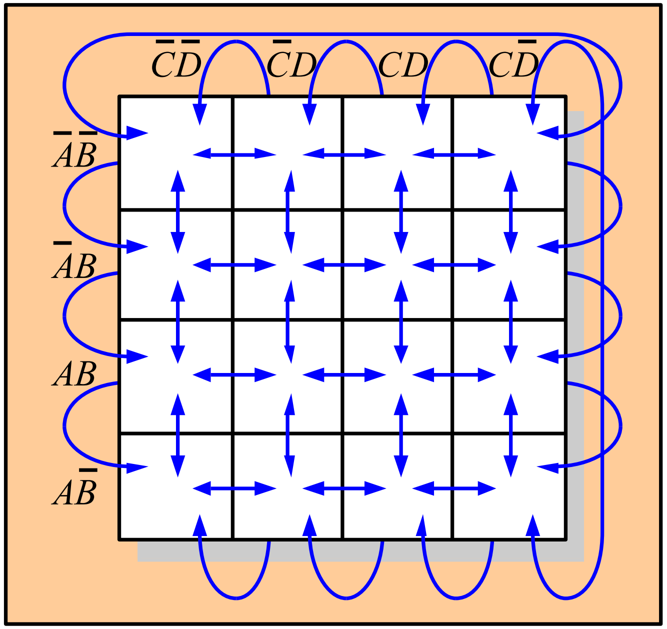

**Çözüm:**

1. 1'leri belirtildiği gibi örtüşen iki grup halinde gruplayın.
2. Bir sınır boyunca değişen herhangi bir değişkeni eleyerek her grubu okuyun.
3. Dikey grup şöyle okunur $\overline{A}\overline{C}$.
4. Yatay grup ise şöyle okunur $\overline{A}B$.

$$X = \overline{A}\overline{C} + \overline{A}B$$

## Karnaugh Haritaları

4 değişkenli bir harita, gösterildiği gibi dört sınırının her birinde bitişik bir hücreye sahiptir.



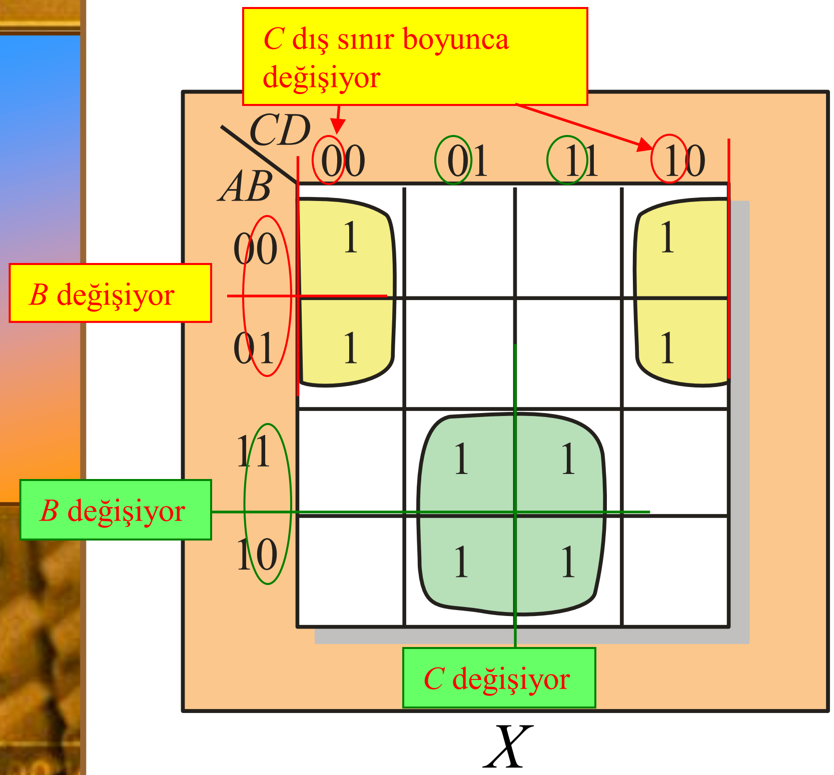

Her hücre, bitişik bir hücreden yalnızca bir değişkenle farklıdır.

Gruplama, metinde verilen kuralları izler.

Aşağıdaki slayt, değişkenler için ikili sayılar kullanarak dört değişkenli bir haritayı okumanın bir örneğini göstermektedir ...

## Karnaugh Haritaları

**Örnek:** 1'leri haritada gruplayın ve en sade fonksiyonu okuyun.




**Çözüm:**

1. Belirtildiği gibi 1'leri iki ayrı grupta gruplayın.
2. Bir sınır boyunca değişen herhangi bir değişkeni kaldırarak her grubu okuyun.
3. Yukardaki (sarı) grup şöyle okunur: $\overline{A}\overline{D}$.
4. Aşağıdaki (yeşil) grup şöyle okunur: $AD$.

$$X = \overline{A}\overline{D} + AD$$

## Donanım Tabanlı Dilleri (HDLs)

Donanım Tabanlı Dilleri (Hardware Description Language/ HDL)'ler, bir PLD'de mantık tasarımı uygulamak için bir araçtır. Önemli bir dil VHDL olarak adlandırılır. VHDL'de mantığı tanımlamaya yönelik üç yaklaşım vardır:

| | |
|---|---|
| 1. Yapısal | Donanım dili bir şematik gösterimlidir. (bileşenler ve blok diyagramlar). |
| 2. Veri akışı | Donanım dili  Boolean cebiri gibi eşitliklerle ifade edilir. |
| 3. Davranışsal | Donanım dili  zamanlayıcı kullanma özelliği vardır. (durum makineleri, vs.). |

## Donanım Tabanlı Dilleri (HDLs)

VHDL için veri akışı yöntemi, Boole türü ifadeler kullanır. Temel bir veri akışı programının iki bölümü vardır: varlık ve mimari. Varlık bölümü, Giriş/Çıkış'ı açıklar. Mimari kısım mantığı açıklar. Aşağıdaki örnek, iki bölümü gösteren bir VHDL programıdır. Program geçersiz bir BCD kodunu tespit etmek için kullanılır.

```
entity BCDInv is
    port (B,C,D: in bit; X: out bit);
end entity BCDInv
```

```
architecture Invalid of BCDInv
    begin
        X <= (B or C) and D;
end architecture Invalid;
```

# Donanım Tabanlı Dilleri (HDLs)

Diğer bir standart HDL, Verilog'dur. Verilog'da, G / Ç ve mantık, modül adı verilen tek bir birimde açıklanır. Verilog, Boolean mantıksal operatörleri temsil etmek için belirli semboller kullanır.

Aşağıdaki, Verilog için yazılmış bir önceki slaytta olduğu gibi aynı programdır:

```
module BCDInv (X, B, C, D);
input B, C, D;
output X;
    assign X = (B | C)&D;
endmodule
```

# Anahtar Kelimeler

***Variable (Değişken)*** Genellikle italik bir harfle gösterilen, 1 veya 0 değerine sahip olabilen mantıksal bir değeri temsil etmek için kullanılan bir sembol.

***Complement (Tümleyen)*** Bir sayının tersi veya tümleyeni. Boole cebirinde, ters fonksiyon, değişken üzerinde bir çizgi ile ifade edilir.

***Sum term (Toplam terimi)*** OR işlemine eşdeğer iki veya daha fazla değişkenin Boolean toplamı.

***Product term (Çarpım terimi)*** AND işlemine eşdeğer iki veya daha fazla değişmezin Boolean çarpımı.

***Sum-of-products (SOP)*** Temelde AND'li terimlerin OR'lanması olan bir Boolean ifadesi biçimi.

***Product of sums (POS)*** Temelde OR'lu terimlerinin AND'lenmesi olan bir Boolean ifadesi biçimi.

***Karnaugh map (Haritası)*** Bir Boole ifadesindeki değişmez değerlerin kombinasyonlarını temsil eden ve ifadenin sistematik basitleştirilmesi için kullanılan bir hücre düzenlemesi.

***VHDL*** Standart bir donanım tabanlı dil. IEEE Std. 1076-1993.

# Quiz

1. The associative law for addition is normally written as

   a. $A + B = B + A$

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

   c. $AB = BA$

   d. $A + AB = A$

# Quiz

2. The Boolean equation $AB + AC = A(B + C)$ illustrates

   a. the distribution law

   b. the commutative law

   c. the associative law

   d. DeMorgan's theorem

# Quiz

3. The Boolean expression $A \cdot 1$ is equal to

    a. $A$

    b. $B$

    c. 0

    d. 1

# Quiz

4. The Boolean expression $A + 1$ is equal to

   a. $A$

   b. $B$

   c. 0

   d. 1

# Quiz

5. The Boolean equation $AB + AC = A(B + C)$ illustrates

   a. the distribution law

   b. the commutative law

   c. the associative law

   d. DeMorgan's theorem

# Quiz

6. A Boolean expression that is in standard SOP form is

   a. the minimum logic expression

   b. contains only one product term

   c. has every variable in the domain in every term

   d. none of the above

# Quiz

7. Adjacent cells on a Karnaugh map differ from each other by

    a. one variable

    b. two variables

    c. three variables

    d. answer depends on the size of the map

# Quiz

8. The minimum expression that can be read from the Karnaugh map shown is

a. $X = A$

b. $X = \overline{A}$

c. $X = B$

d. $X = \overline{B}$

| | $\overline{C}$ | $C$ |
|---|---|---|
| $\overline{A}\,\overline{B}$ | | |
| $\overline{A}B$ | | |
| $AB$ | 1 | 1 |
| $A\overline{B}$ | 1 | 1 |

# Quiz

9. The minimum expression that can be read from the Karnaugh map shown is

| | $\overline{C}$ | $C$ |
|---|---|---|
| $\overline{A}\overline{B}$ | 1 | 1 |
| $\overline{A}B$ | | |
| $AB$ | | |
| $A\overline{B}$ | 1 | 1 |

a. $X = A$

b. $X = \overline{A}$

c. $X = B$

d. $X = \overline{B}$

# Quiz

10. In VHDL code, the two main parts are called the

    a. I/O and the module

    b. entity and the architecture

    c. port and the module

    d. port and the architecture

# Quiz

Answers:

1. b
2. c
3. a
4. d
5. a
6. c
7. a
8. a
9. d
10. b