Binary Toplama ve Çıkarma

0+0 → 0	111010
0+1 → 1	− 100010
1+0 → 1	11000
1+1 → 10	
1+1+1 → 11	

1'e Tümleyen

2'li sayılar sisteminde 1'e tümleyen için 0 → 1'a çevir.
1 → 0

Örnek 1011001 → 0100110

2'e Tümleyen

1'e tümlemeye 1 ekle

Örnek 1011001 → 0100110
+ 1
0100111

Örnek

1010100
− 1000011 → 0111100
+ 1

1'e tümleme

'ye göre

leme

0111101

1010100 + 0111101 = 10010001

➤ Örnek-2

$(1011)_2$ 1.Komplementeri 1111 ⟶ 0000

$-(1111)_2$ + 1

0001

2. Komplementer

$$\begin{array}{r} 1011 \\ +\ 0001 \\ \hline 1100 \end{array}$$

Eğer elde 1 oluşmamışsa sonuç negatiftir ve gerçek sonuç çıkan sonucun tersine "1" eklenmesi ile bulunur.

$$\begin{array}{r} 0011 \\ +\ \ \ 1 \\ \hline \end{array}$$

$(-0100)_2$ olur.

En anlamlı ve en az anlamlı bit

1001001 → En az anlamlı

↳ En anlamlı

En anlamlı bit 1 → negatif

0 → pozitif

ASCI GRAY BCD

BCD KODU ⇒

$(357)_{10} \Rightarrow \underbrace{0011}_{3}\ \underbrace{0101}_{5}\ \underbrace{0111}_{7}$

GRAY KODU ⇒

GRAY KODU ⇒

Gray Code'u Binary Code'a çevirmek için, XOR işlemi kullanılır. XOR işlem tablosu aşağıdaki gibidir;

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

ÖRN $(101110101)_2$ gray koduna çeviriniz,

0 1 0 1 1 1 0 1 0 1

1 1 1 0 0 1 1 1 1

$(111001111)_{gray}$

Parity (Eşitlik) Kodu

– Çift eşitlik yöntemi : Eşitlik bitinin değeri, kodlanacak bilgideki 1'ların toplam sayısı (eşitlik biti dahil) çift olacak şekilde seçilir.

Kodlanacak sayıdaki 1'ların sayısı tek ise, eşitlik biti olarak '1' eklenir. Kodlanacak bilgideki 1'ların sayısı çift olması durumunda ise, eşitlik biti olarak '0' eklenir.

ÖRN $(1000011)_2$ sayısına çift eşitlik yöntemi ile eşitlik biti ekleyelim

Kodlanacak bilgide 3 adet 1 olduğuna göre '1' eklenir

$(\underline{1}1000011)_2$

– Tek eşitlik yöntemi

Aynı mantığa göre düzenlenir. Tek fark kodlanan bilgideki 1'ların sayısı tek olmalıdır.

ÖRN $(1000001)_2$ sayısına tek eşitlik biti yöntemi uygulayalım

çift sayıda 1 olduğundan eşitlik biti '1' olur.

$(\underline{1}1000001)_2$

ASCII KODU

ASCII KODLARI

$b_4b_3b_2b_1$ \ $b_7b_6b_5$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	¦
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Lojik İşlemlerin Doğruluk Tabloları

VE ×
AND

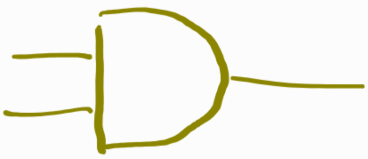
0 0 → 0
0 1 → 0
1 0 → 0
1 1 → 1

VEYA +
OR

0
1
1
1

DEĞİL
NOT

0 → 1
1 → 0



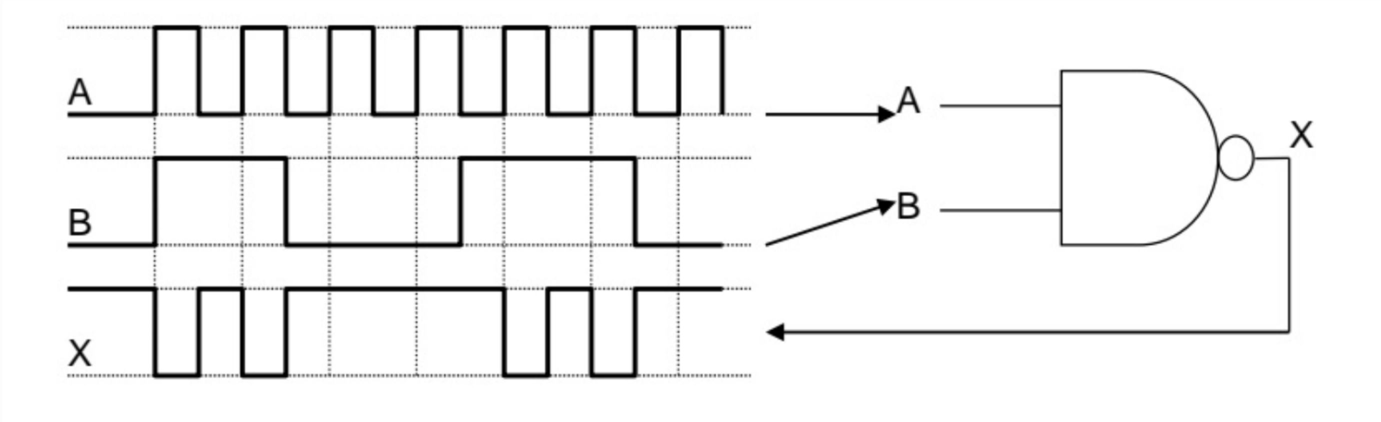
Özel Veya

0 0 → 0

$x \oplus y$

$\underline{\bar{x}y + x\bar{y}}$

0	1	→	1
1	0	→	1
1	1	→	0



Boolean Kanunları ve Demorgan Teoremi

→ Boolean Toplama

Veya işlemine eşittir.

$0+0=0$ $\quad$ $1+0=1$

$0+1=1$ $\quad$ $1+1=1$

→ Boolean Çarpma

Ve işlemine eşdeğerdir.

$0.0=0$ $\quad$ $1.0=0$

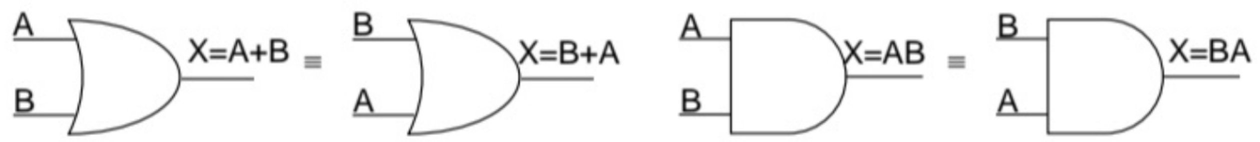
$0.1=0$ $\quad$ $1.1=1$

Boolean Aritmetiği Kanunları

→ Değişme Özelliği

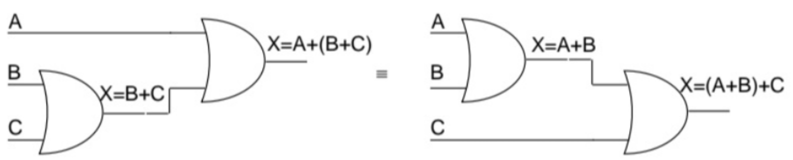
"Veya" ve "Ve" işlemlerinde sıra önemli değil.

$A+B = B+A$ $\qquad$ $A.B = B.A$

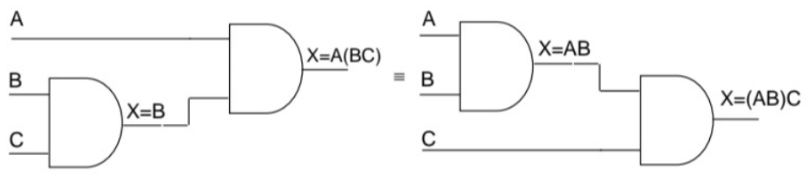


Şekil-4.1 Değişme özelliği.

→ Birleşme Özelliği



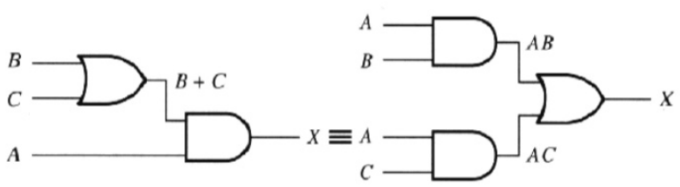
Şekil-4.2 Toplamada Birleşme özelliği.



Şekil-4.3 Çarpmada birleşme özelliği

2'den fazla değişkenin sırasının önemli olmadığını gösterir.

→ Dağılma Özelliği



Şekil-4.4 Dağılma özeliğinin uygulaması.

$A(B+C)$

$=$

$AB + AC$

Boolean Kuralları

1.	A+0=A
2.	A+1=1
3.	A•0=0
4.	A•1=A
5.	A+A=A
6.	A+A'=1
7.	A•A=A
8.	A•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

Tablo-4.1 Boolean kuralları.

Kural 10 = $A + AB = A$

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

Kural 11 = $A + \bar{A}B = A+B$

$(A+AB) + \bar{A}B$

$AA + AB + A\bar{A} + \bar{A}B$

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

$$\underbrace{(\ldots)}_{1}(\ldots) = A+B$$

Kural 12 = $(A+B)(A+C) = A+BC$

$AA + AC + AB + BC$

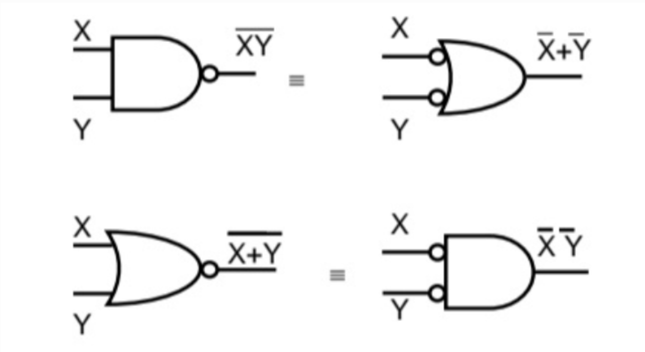
$AC(1+C) + AB + BC$

$A \cdot 1 + AB + BC$

$A(1+B) + BC$

$\underline{A + BC}$

De Morgan Teoremi



- $\overline{XY} = \overline{X} + \overline{Y}$
- $\overline{X+Y} = \overline{X} \cdot \overline{Y}$

Çarpımların Teoremi

! Değişkenin üzerindeki değil çizgileri birleşik çizilmez

$$\overline{A}\,\overline{B}\,\overline{C} \neq \overline{ABC}$$

Örnek $A(AB + CD) \Rightarrow AB + ACD$

Toplamların Çarpımı

Örnek $AB + A(B+C) + B(B+C)$ en basit hali

$$\underbrace{AB + AB}_{AB} + AC + \underbrace{BB}_{B} + BC$$

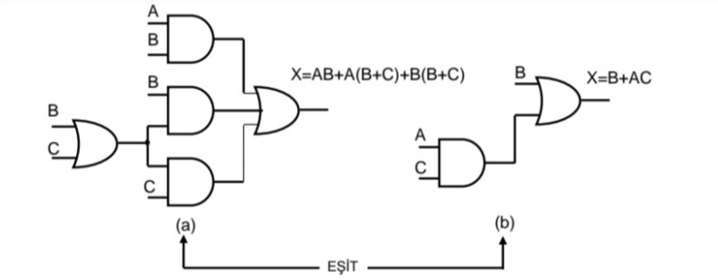
$$AB + AC + B + BC$$

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

$$\rightarrow AB + AC + B$$

$$AC + B(A+1)$$

$$AC + B$$



Soru $F = A.(\overline{B+C}) + \bar{A} + (\bar{B}.C) \Rightarrow \bar{F}$

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

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

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

Soru $F_1 \quad \bar{x}yz + \bar{x}\bar{y}z \qquad F_2 \quad x(\bar{y}\bar{z} + yz)$

$$\bar{F_1} = (x + \bar{y} + \bar{z}) . (x + y + \bar{z})$$

$$\bar{F_2} = \bar{x} + ((y+z) . (\bar{y} + \bar{z}))$$

Her fonk. mintermlerin toplamı ya da maxtermlerin çarpımına eşittir. Mintermler küçük "m" ile gösterilir. Maxtermler büyük "M" ile gösterilir.

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \bar{x}_1\bar{x}_2\bar{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \bar{x}_1\bar{x}_2x_3$	$M_1 = x_1 + x_2 + \bar{x}_3$
2	0	1	0	$m_2 = \bar{x}_1x_2\bar{x}_3$	$M_2 = x_1 + \bar{x}_2 + x_3$
3	0	1	1	$m_3 = \bar{x}_1x_2x_3$	$M_3 = x_1 + \bar{x}_2 + \bar{x}_3$
4	1	0	0	$m_4 = x_1\bar{x}_2\bar{x}_3$	$M_4 = \bar{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\bar{x}_2x_3$	$M_5 = \bar{x}_1 + x_2 + \bar{x}_3$
6	1	1	0	$m_6 = x_1x_2\bar{x}_3$	$M_6 = \bar{x}_1 + \bar{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$

- ↳ Mintermde değil varsa "0"
- ↳ Maxtermde değil varsa "1"

olur.

Fonk. yazmak için mintermlerin 1 maxtermlerin 0 olduğu yerler yazılır.

Örnek $A + B'C$ mintermlerin toplamı

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$f(A+B+C) = m_1 + m_4 + m_5 + m_6 + m_7$$

$$= A'B'C + AB'C' + AB'C + ABC' + ABC$$

Karnaugh Diyagramı ile Sadeleştirme

Örnek $f = x'y' + xy' + xy$ Sadeleştir.

y \ x	0	1
0	1	1
1	0	1

$f = \underline{y'} + x$ → y değiştiği için x aldık
→ 0 ise değili

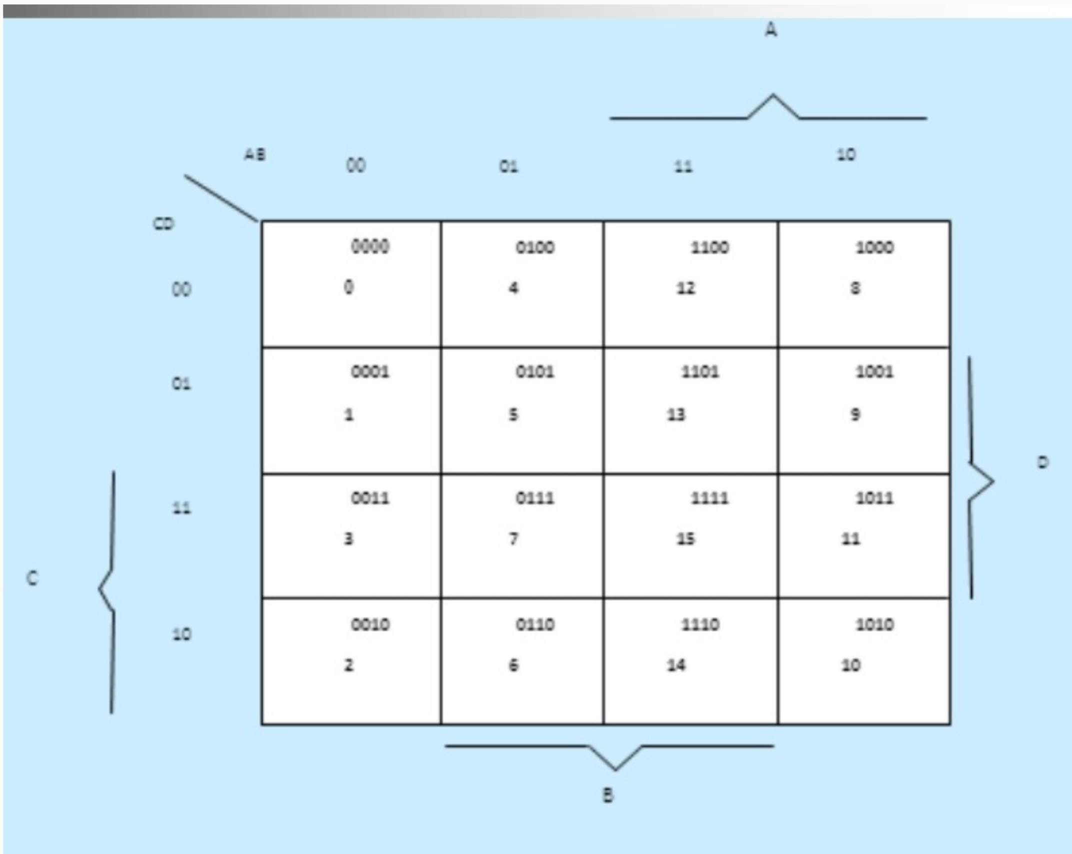
x değeri değiştiği için

Karnaugh Kanunları

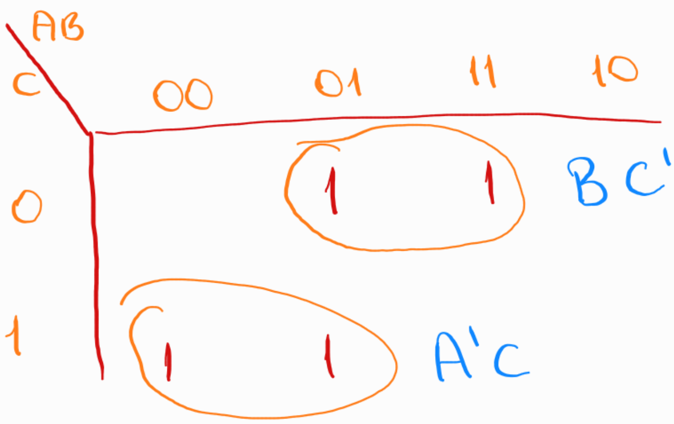
- Hedef en çok 1 i gruplamaktır.
- Gruplamalar 1, 2, 4, 8, 16 gibi 2^n üssü şeklinde olmalı
- Grup içlerinde AND (çarpım) yapılır.
- Gruplar arası OR (toplama) yapılır.
- Tüm kutular 1 ise 1 tüm kutular 0 ise 0 'dır.
- Çapraz gruplama yapılamaz.
- Gruplarda değişenler ele alınmaz
- 0 ise değil 1 ise kendisi alınır.

C \ AB	00	01	11	10
0	000 A'B'C' 0	010 A'BC' 2	110 ABC' 6	100 AB'C' 4
1	001 A'B'C 1	011 A'BC 3	111 ABC 7	101 AB'C 5

AB \ C	0	1
00	000 A'B'C' 0	001 A'B'C 1
01	010 A'BC' 2	011 A'BC 3
11	110 ABC' 6	111 ABC 7
10	100 AB'C' 4	101 AB'C 5

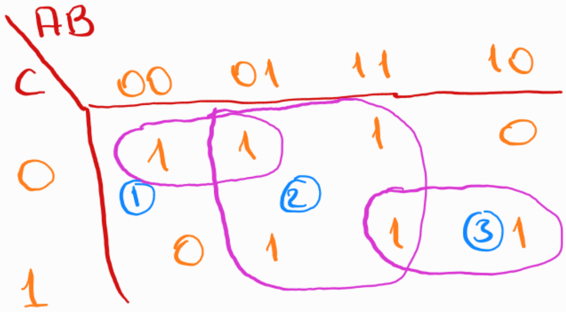


Örnek $F = A'BC + A'B'C + ABC' + A'BC'$



$F = A'C + BC'$

Örnek $A'B'C' + A'BC' + ABC' + A'BC + ABC + AB'C$



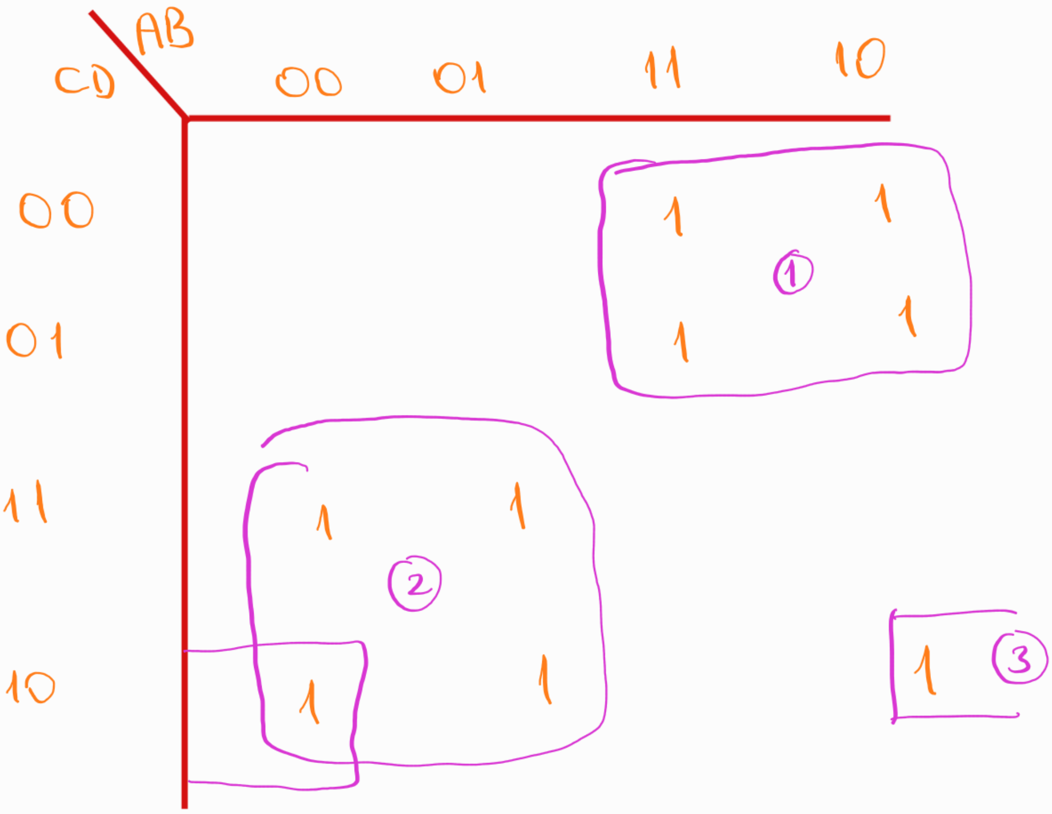
$1 \Rightarrow A'C'$
$2 \Rightarrow B$
$3 \Rightarrow AC$

$F = A'C' + B + AC$

Örnek Aşağıda verilen boolean eşitliğini Karnaugh haritası

Kullanarak çözünüz.

$F = ABC'D' + AB'C'D' + ABC'D + AB'C'D + A'B'CD + A'BCD +$
$A'B'CD' + A'BCD' + AB'CD'$



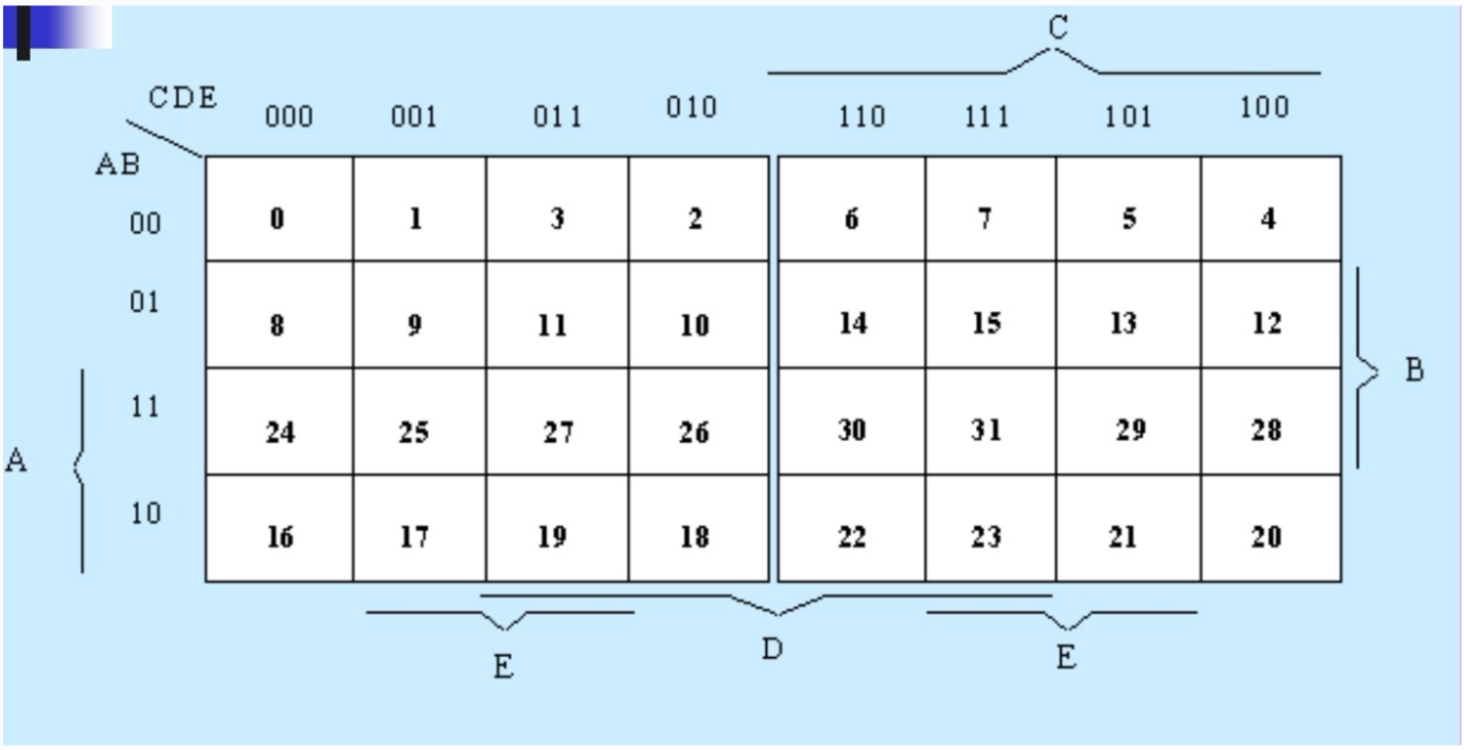
$1 \to AC'$

$2 \to A'C$

$3 \to B'CD'$

$F = AC' + A'C + B'CD'$

5'li Karnaugh Diyagramı



5 değişkenli karnough diyagramı

bc \ de	00	01	11	10
00		1	1	
01		1	1	
11				
10				

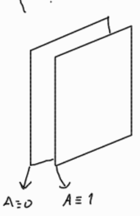
A = 0

bc \ de	00	01	11	10
00		1	1	
01		1	1	
11				
10				

A = 1

8'li grup

(3D olarak çözülür)



farketmez (etkisiz) koşullar (Don't care)

x	y	z	t	f	
0	0	0	0	0	
0	0	0	1	1	✓
0	0	1	0	0	
0	0	1	1	1	✓
0	1	0	0	1	✓
0	1	0	1	0	
0	1	1	0	1	✓
0	1	1	1	1	✓
1	0	0	0	0	

Devamı farketmez (yukarıdaki aralıkta doğru çalışsın yeter)

$\Rightarrow$

xy \ zt	00	01	11	10
00	0	1	1	0
01	1	0	∅	1
11	∅	∅	∅	∅
10	0	∅	∅	∅

$\Rightarrow$

xy \ zt	00	01	11	10
00		1	1	
01	1		∅	1
11	∅	∅	∅	∅
10		∅	∅	∅

- çözümde ∅ ile 1 aynı grupta olabilir ve ∅ elemanları en son açıkta kalabilir

∅, d, x | farketmez sembolleri

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1 —
0	0	1	1	1 —
0	1	0	0	0 —
0	1	0	1	X
0	1	1	0	1 —
0	1	1	1	1 —
1	0	0	0	1 —
1	0	0	1	1 —
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y

CD \ AB	00	01	11	10
00	1	0	X	1
01	0	X	X	1
11	1	1	X	X
10	1	1	X	X

Kombinezonal Lojik Bileşik Lojik Sistemleri

- Problem sözel olarak ifade edilir
- Giriş ve çıkış değerlerin sayısı belirlenir.
- Giriş ve çıkış sembollerine harf sembolleri atanır.
- Giriş ile çıkış arasında doğruluk tablosu oluşur.
- Her çıkış için sadeleşmiş fonk. elde edilir
- Devre çizilir.

Toplayıcı ve Çıkarıcılar

- 4 bitlik toplayıcı

Yarı Toplayıcı (Eldesiz)

Bir bitlik iki sayı toplayacak devre

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$S = x'y + xy'$

$C = xy$

Tam Toplayıcı (Eldeli)

Bir bitlik iki sayının toplamı

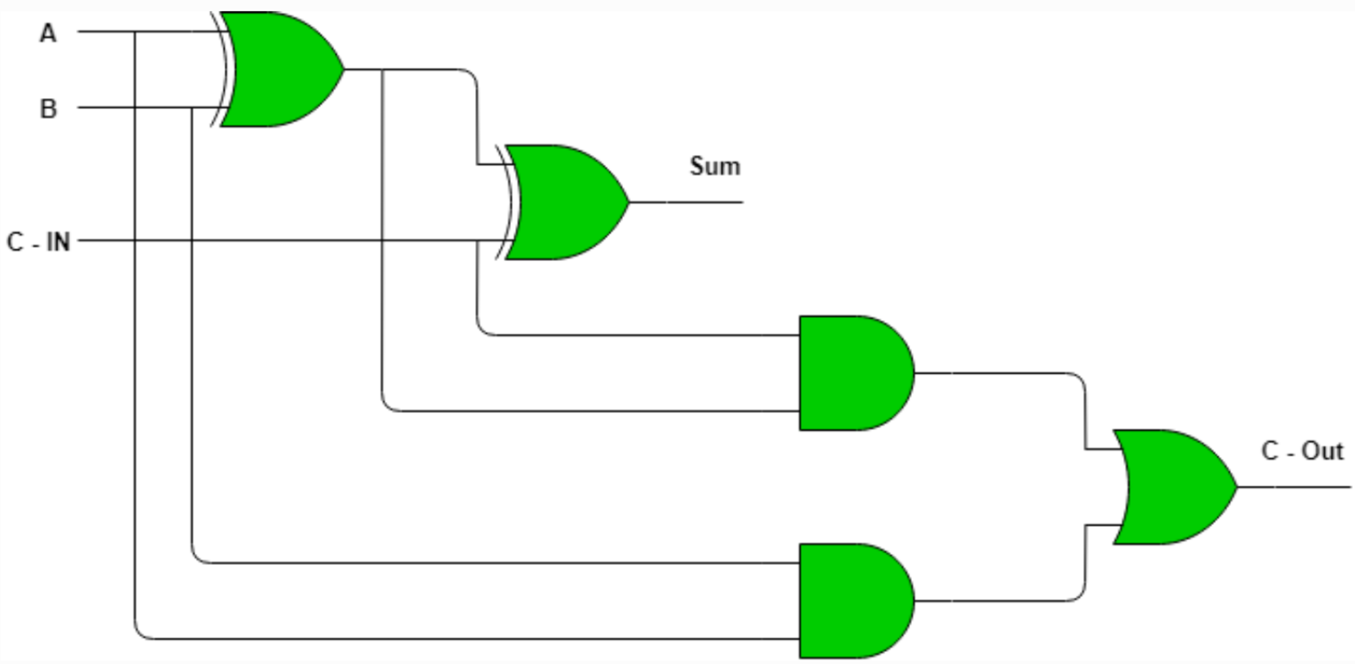
X	Y	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S

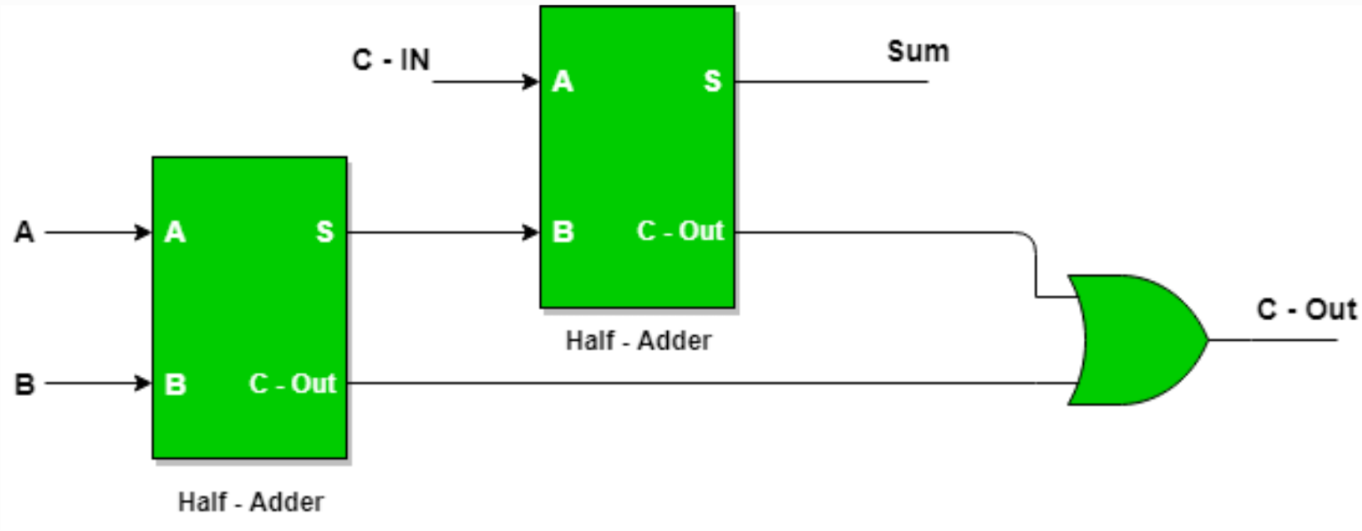
x \ y,cin	00	01	11	10
0		1		1
1	1		1	

Cout

x \ y,cin	00	01	11	10
0			1	
1		1	1	1



! 2 Yarım toplayıcı tam toplayıcıya eşittir.



Yarı Çıkarıcı

Bir bitlik iki sayının çıkarımı

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$B = x'y$

$D = x \oplus y$

Subtrahend

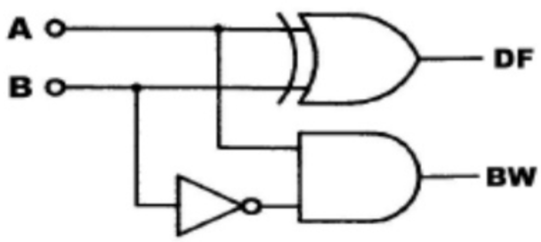
Minuend

Difference

Borrow

A	B	DF	BW
0	1	0	0
0	0	1	1
1	1	1	0
1	0	0	0

(a) Doğruluk Tablosu



(b) Lojik diyagramı

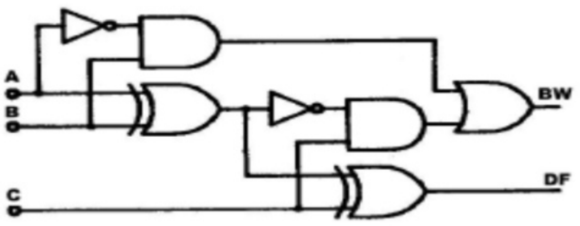
Şekil 5-1 Yarım Çıkarıcı

Tam Çıkarıcı

Previous borrow
Minuend
Subtrahend
Difference
Borrow

C	A	B	DF	BW
0	0	1	0	0
0	0	0	1	1
0	1	1	1	0
0	1	0	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

(a) Doğruluk Tablosu



(b) Lojik diyagramı

Şekil 5-2 Tam Çıkarıcı

Karşılaştırma Devresi

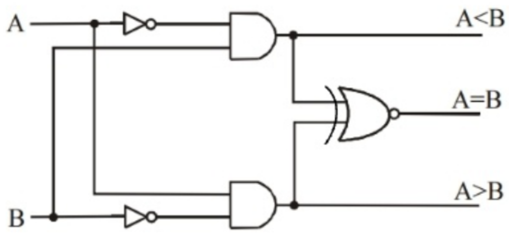
GİRİŞLER		ÇIKIŞLAR		
A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

(A<B) çıkışı = $\overline{A}.B$

(A=B) çıkışı = $\overline{A}.\overline{B} + A.B = \overline{A \oplus B} = A \otimes B$

(A>B) çıkışı = $A.\overline{B}$

Lojik Devresi



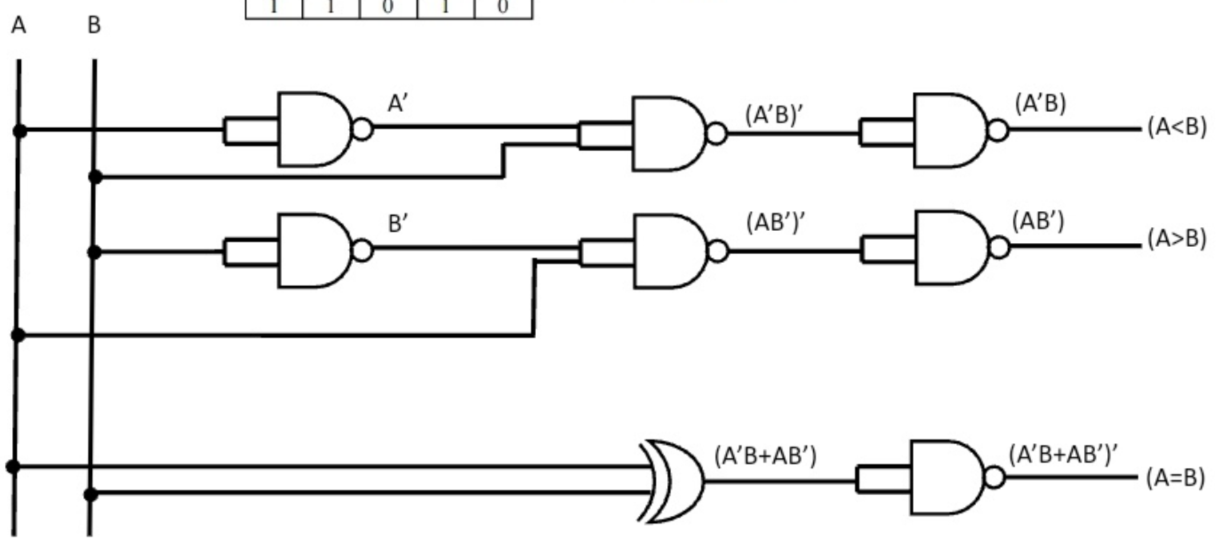
Soru: 2 girişli VE-DEĞİL ile 2 girişli XOR kapısı kullanarak en basit karşılaştırıcı tasarlayınız

GİRİŞLER		ÇIKIŞLAR		
A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

(A<B) çıkışı = $\overline{A}.B$

(A=B) çıkışı = $\overline{A}.\overline{B} + A.B = \overline{A \oplus B} = A \otimes B$

(A>B) çıkışı = $A.\overline{B}$



Kodlayıcı (Encoder)

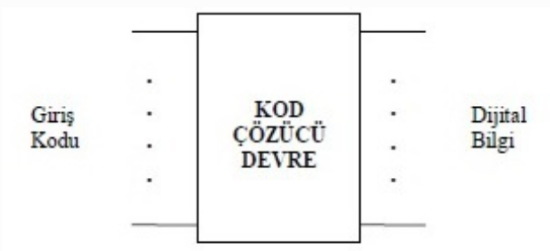
K_3	K_2	K_1	K_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$Y_0 = K_1 + K_3$

$Y_1 = K_2 + K_3$

Decoder

Kodlayıcının tersi işlemi yapar.



Giriş		Çıkış			
A	B	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

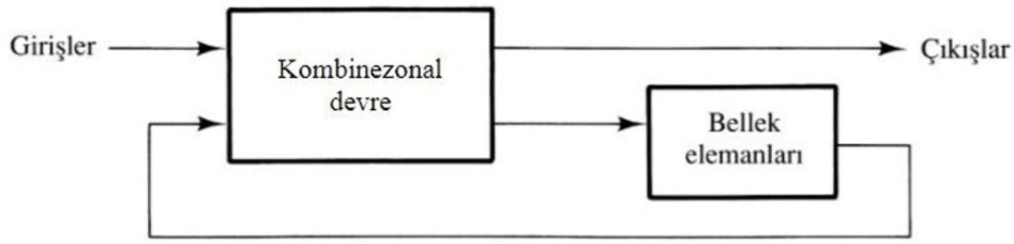
$Y_0 = A'B'$

$Y_1 = A'B$

$Y_2 = AB'$

$Y_3 = AB$

Ardışıl Lojik Sistemler



ŞEKİL 6-1
Ardışıl devrenin blok diyagramı

Flip Flop

Girişlerine verilen değerlerle senkron işaretlerine göre sonraki değerleri önceden hesap edebilen devre elemanı.

Saklama Elemanı

→ Latch

→ Flip Flop