



Lojik Tasarım

Ders 4

Kaynak:

M.M. Mano, M.D. Ciletti, "Digital Design with An Introduction to Verilog HDL"



Kanonik Standart ve Biçimler

Minterm ve Maxterm

- Her fonksiyon
 - **Mintermlerin toplamı**
yada
 - **Maxtermlerin çarpımı**
şeklinde ifade edilebilir

Minterm ve Maxterm

► Üç değişkenli bir fonksiyon için minterm ve maxterm tablosu

			Minterms		Maxterms	
<i>x</i>	<i>y</i>	<i>z</i>	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

Minterm için fonksiyonun 1 olduğu satırlar işleme alınır

<i>x</i>	<i>y</i>	<i>z</i>	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

$$f_2 = x'yz + xy'z + xyz' + xyz = m_3 + m_5 + m_6 + m_7$$

Örnek : (Fonksiyonun tümleyeni)

x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

$$f_1' = x'y'z' + x'yz' + x'yz + xy'z + xyz'$$


$$f_1 = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$$

$$f_1 = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

Örnek : (Fonksiyonun tümleyeni)


x	y	z	Function f_1	Function f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

$$\begin{aligned}f_2 &= (x + y + z)(x + y + z')(x + y' + z)(x' + y + z) \\ &= M_0 M_1 M_2 M_4\end{aligned}$$



Örnek : Aşağıda verilen fonksiyonu mintermlerin toplamı şeklinde ifade ediniz

$$F = A + B'C$$


$$F = A + B'C$$

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

This function is still missing one variable, so

$$\begin{aligned} A &= AB(C + C') + AB'(C + C') \\ &= ABC + ABC' + AB'C + AB'C' \end{aligned}$$

The second term $B'C$ is missing one variable; hence,


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

Combining all terms, we have

$$\begin{aligned} F &= A + B'C \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

But $AB'C$ appears twice, and according to theorem 1 ($x + x = x$), it is possible to remove one of those occurrences. Rearranging the minterms in ascending order, we finally obtain

$$\begin{aligned} F &= A'B'C + AB'C' + AB'C + ABC' + ABC \\ &= m_1 + m_4 + m_5 + m_6 + m_7 \end{aligned}$$

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$


Alternatif çözüm: Doğruluk tablosu üzerinden çözüm

$$F = A + B'C$$

Truth Table for $F = A + B'C$

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

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

Örnek : Aşağıda verilen fonksiyonu makstermlerin çarpımı şeklinde ifade ediniz

$$F = xy + x'z$$

Terimlere dağılma özelliği uygulanabilir

$$\begin{aligned} F &= xy + x'z = (xy + x')(xy + z) \\ &= (x + x')(y + x')(x + z)(y + z) \\ &= (x' + y)(x + z)(y + z) \end{aligned}$$

Eksik terimler fonksiyona eklenir

$$\begin{aligned} x' + y &= x' + y + zz' = (x' + y + z)(x' + y + z') \\ x + z &= x + z + yy' = (x + y + z)(x + y' + z) \\ y + z &= y + z + xx' = (x + y + z)(x' + y + z) \end{aligned}$$

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0 M_2 M_4 M_5 \end{aligned}$$

$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Kanonik formlar arası dönüşüm

$$F(A, B, C) = \Sigma(1, 4, 5, 6, 7)$$

This function has a complement that can be expressed as

$$F'(A, B, C) = \Sigma(0, 2, 3) = m_0 + m_2 + m_3$$

Now, if we take the complement of F' by DeMorgan's theorem, we obtain F in a different form:

$$F = (m_0 + m_2 + m_3)' = m'_0 \cdot m'_2 \cdot m'_3 = M_0 M_2 M_3 = \Pi(0, 2, 3)$$

The last conversion follows from the definition of minterms and maxterms as shown in Table 2.3. From the table, it is clear that the following relation holds:

$$m'_j = M_j$$

$$F = xy + x'z$$

Truth Table for $F = xy + x'z$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

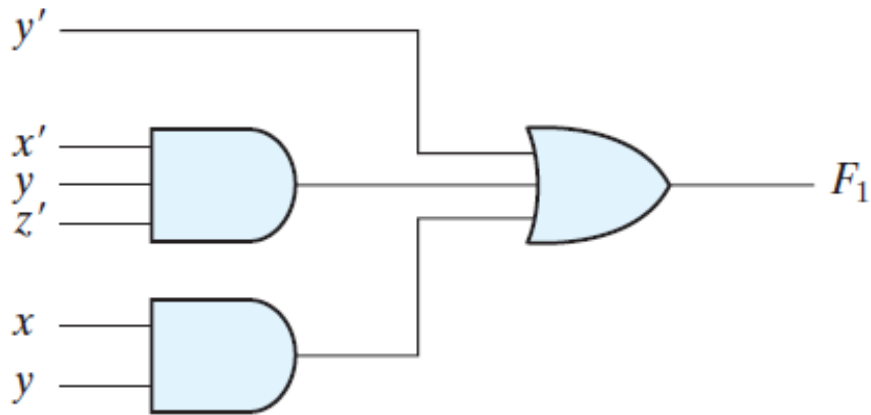
Minterms

Maxterms

$$F(x, y, z) = \Sigma(1, 3, 6, 7)$$

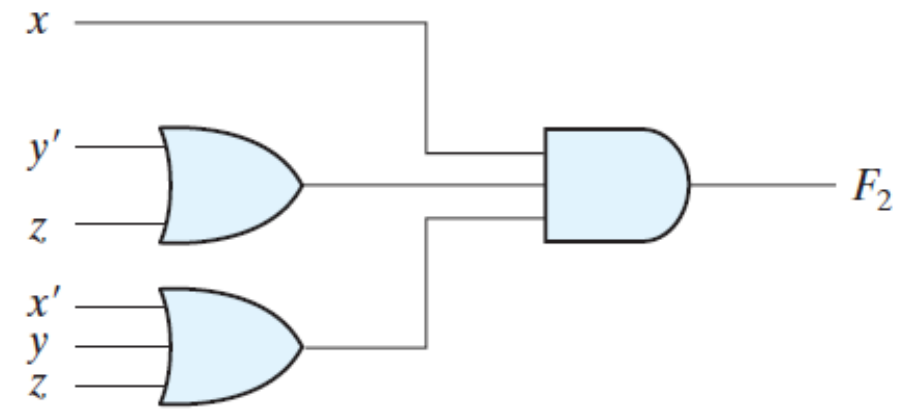
$$F(x, y, z) = \Pi(0, 2, 4, 5)$$

Çarpımların toplamı / Toplamların Çarpımı



(a) Sum of Products

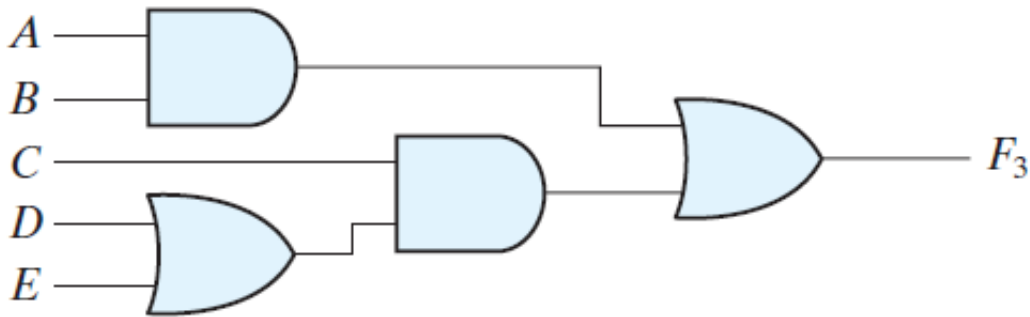
$$F_1 = y' + xy + x'yz'$$



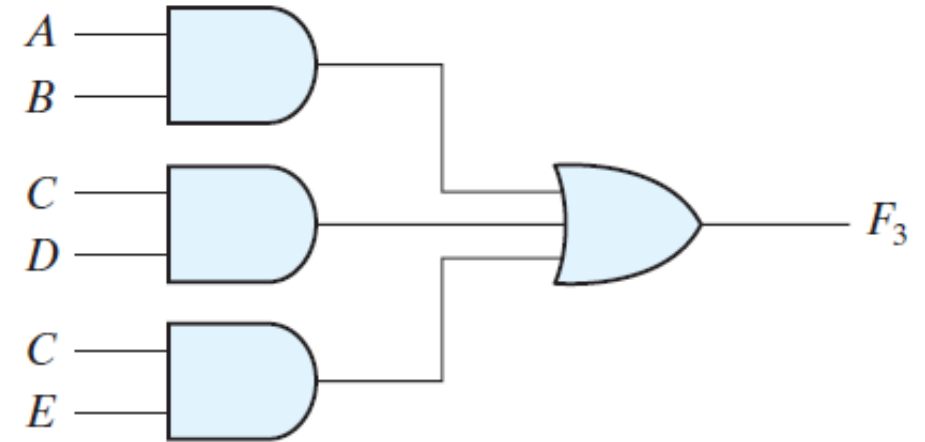
(b) Product of Sums

$$F_2 = x(y' + z)(x' + y + z')$$


Çarpımların toplamı / Toplamların Çarpımı



(a) $AB + C(D + E)$



(b) $AB + CD + CE$





[illegible][illegible]





İki değişkenden 16 farklı fonksiyon üretilebilir

Boolean Expressions for the 16 Functions of Two Variables

Boolean Functions	Operator Symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x , but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y , but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y , but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR
$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \supset y$	Implication	If y , then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x , then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

Sayısal Lojik Kapılar

Name	Graphic symbol	Algebraic function	Truth table															
AND		$F = x \cdot y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x + y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	1
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	

NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = (x + y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	0
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
Exclusive-OR (XOR)		$F = xy' + x'y$ $= x \oplus y$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-NOR or equivalence		$F = xy + x'y'$ $= (x \oplus y)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Şekilde gösterilen kapılardan evirici ve buffer dışındaki kapılar ikiden fazla girişli olabilir.

Çok Girişli Devreler

- ▶ Kapıların giriş sayısı, temsil ettikleri ikili işlemin değişme ve birleşme özelliğine sahip olması şartıyla çok girişli hale getirilebilir.
- ▶ Boolean cebirinde tanımlanan VE ve VEYA işlemleri bu iki özelliğe sahiptir
- ▶ VEYA işlemi için;

$$x + y = y + x \quad (\text{commutative})$$

$$(x + y) + z = x + (y + z) = x + y + z \quad (\text{associative})$$

Çok Girişli Devreler

$x \downarrow y \downarrow z$	VEYADEĞİL işlemi
$x \uparrow y \uparrow z$	VEDEĞİL işlemi

- Değişme özelliğine sahip olan VEDEĞİL ve VEYADEĞİL kapılarının giriş sayısı, işlem tanımının değiştirilmemesi şartıyla ikiden fazla yapılabilir.
- Buradaki sorun, VEDEĞİL ve VEYADEĞİL işlemlerinin birleşme özelliğine sahip olmamasından kaynaklanır. Diğer bir deyişle;

$$(x \downarrow y) \downarrow z \neq x \downarrow (y \downarrow z)$$

$$(x \downarrow y) \downarrow z = [(x + y)' + z]' = (x + y)z' = xz' + yz'$$

$$x \downarrow (y \downarrow z) = [x + (y + z)']' = x'(y + z) = x'y + x'z$$

Çok Girişli Devreler

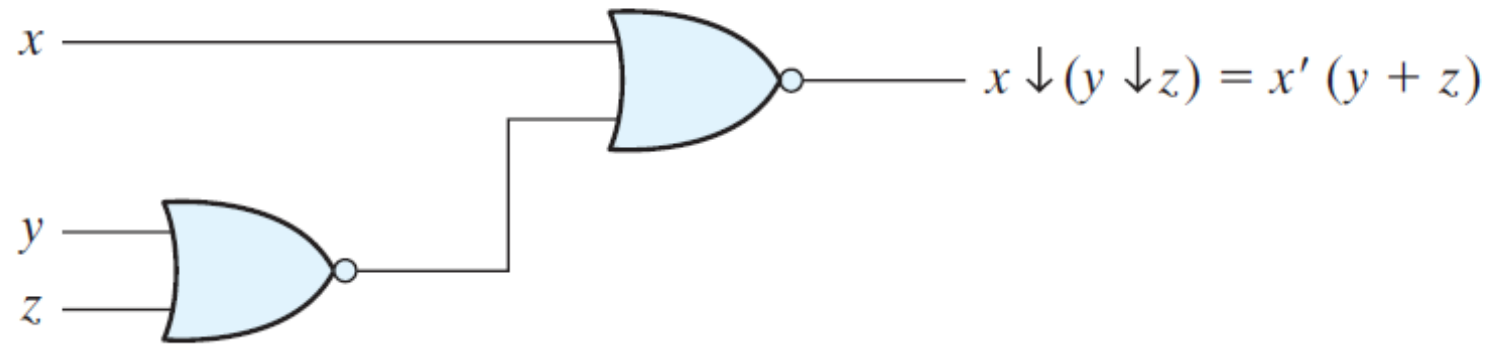
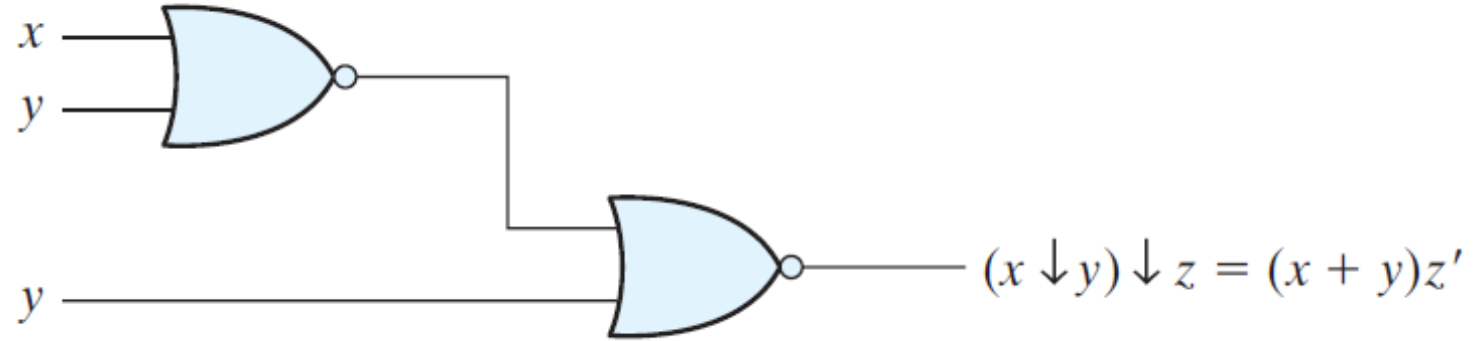
$$x \downarrow y \downarrow z = (x + y + z)'$$

$$x \uparrow y \uparrow z = (xyz)'$$

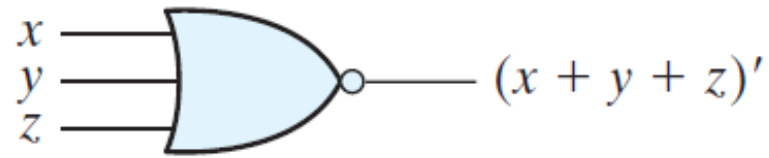
$$(x \downarrow y) \downarrow z = [(x + y)' + z]' = (x + y)z' = xz' + yz'$$

$$x \downarrow (y \downarrow z) = [x + (y + z)']' = x'(y + z) = x'y + x'z$$

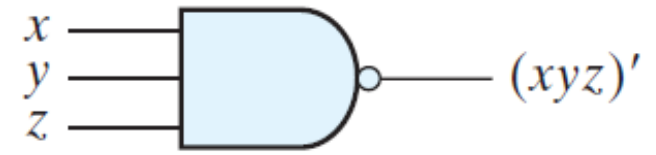
Çok Girişli Devreler



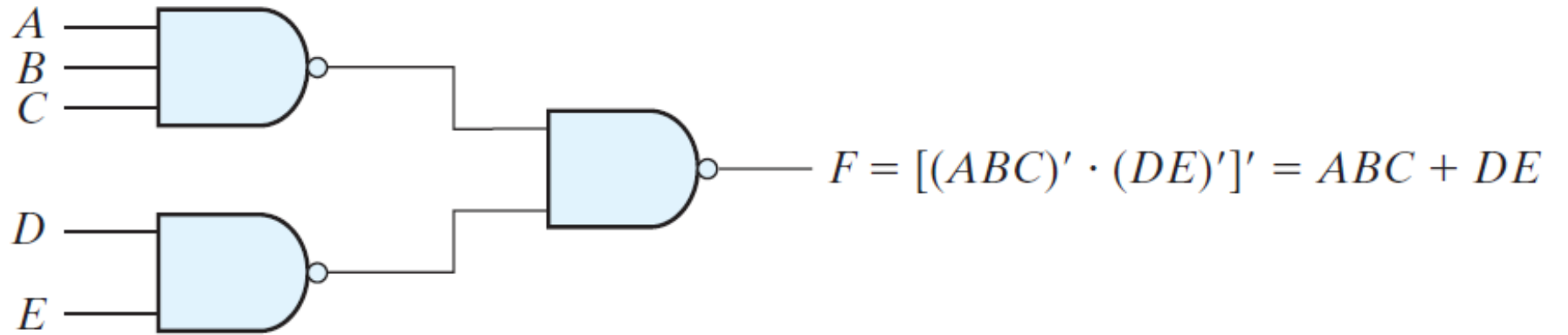
Çok Girişli Devreler



(a) 3-input NOR gate

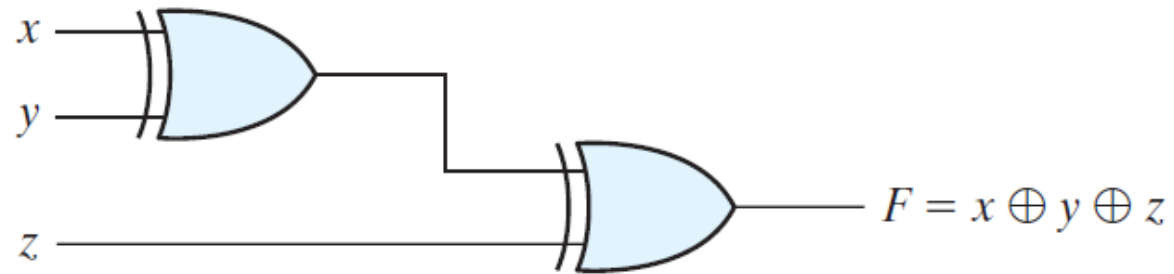


(b) 3-input NAND gate

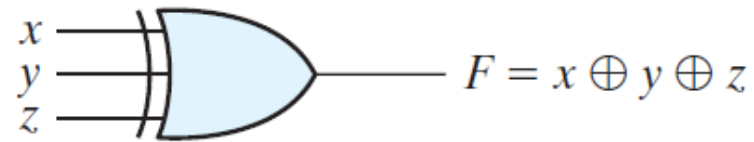


(c) Cascaded NAND gates

Üç girişli ÖZELVEYA kapısı



(a) Using 2-input gates



(b) 3-input gate

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

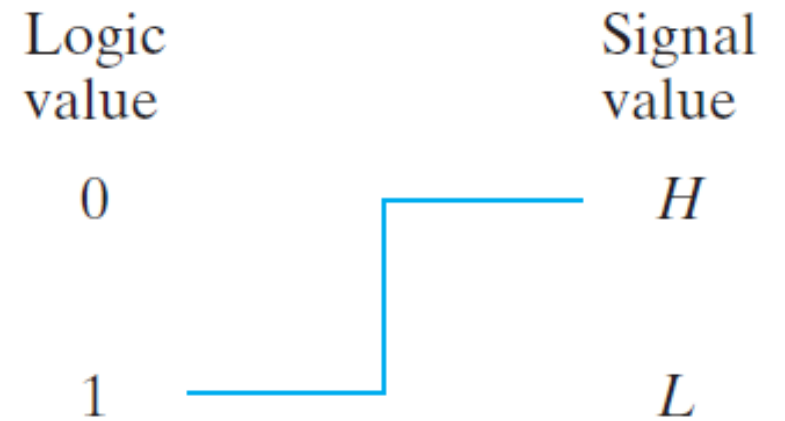
(c) Truth table

Tümleşiklik Seviyeleri

POZİTİF Lojik – NEGATİF Lojik



(a) Positive logic



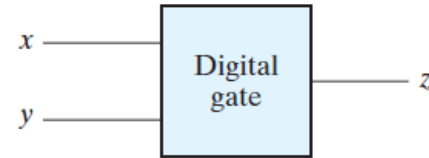
(b) Negative logic

Ders kapsamında POZİTİF lojik kullanılacaktır

Pozitif – Negatif Logik

x	y	z
L	L	L
L	H	L
H	L	L
H	H	H

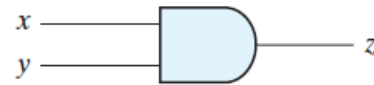
(a) Truth table with H and L



(b) Gate block diagram

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

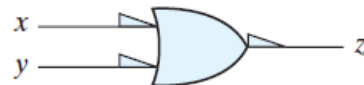
(c) Truth table for positive logic



(d) Positive logic AND gate

x	y	z
1	1	1
1	0	1
0	1	1
0	0	0

(e) Truth table for negative logic



(f) Negative logic OR gate



Sayısal Lojik Aileler

Sayısal tümdevreler üretim tekniği ve teknolojiye göre sınıflandırılırlar.

En yaygın olarak kullanılanları:

TTL	transistor–transistor logic;
ECL	emitter-coupled logic;
MOS	metal-oxide semiconductor;
CMOS	complementary metal-oxide semiconductor.



Sayısal Lojik Aileler

- TTL (Transistör Transistör Lojik):
Uzun yıllardır kullanılmakta olan yaygın bir standarttır.
- ECL (Emitör Kuplajlı Lojik):
Yüksek Hızlı işlem gerektiren sistemlerde kullanışlıdır.
- MOS (Metal Oksit Yarıiletken):
Yüksek eleman yoğunluğu olan devreler için uygundur.
- CMOS (Tümleyici Metaloksit Yarıiletken):
Düşük güç tüketimi gerektiren sistemlerde tercih edilir.

TTL ailesi

- Kapıların oluşturulmasında transistör, direnç ve diyotlar kullanılır.
- Çalışma gerilimi $5V \pm \%25$
- Yüksek hızda çalışırlar
- CMOS'lara göre daha fazla güç harcarlar
- Türleri:
 - Standart TTL
 - Düşük güçlü TTL
 - Yüksek hızlı TTL
 - Schotky tipi TTL
 - Düşük güçlü Schotky tipi TTL

CMOS ailesi

- Kapıların oluşturulmasında sadece transistörler kullanılır. Diyot ve dirençlerin görevini transistörler yapar.
- Çalışma gerilimi 12 – 30V arasındadır.
- Güç harcamaları azdır. (düşük güç kaybı)
- Hızları düşüktür
- Gürültü toleransı yüksektir
- Yüksek frekanslı işlemler için pek tercih edilmezler
- Üretim yöntemi TTL'e göre daha basittir ve bu nedenle ucuzdur
- Türleri:
 - NMOS tipi
 - PMOS tipi

Lojik Entegre Parametreleri

- **Yayılım Gecikmesi (Propagation Delay):** Girişe uygulanan işaretin çıkışta görülmesi arasında geçen zamandır. Nanosaniyelerle ölçülür. Çalışma hızı bununla ters orantılıdır.
- **Güç Kaybı (Power Dissipation):** Kapıda harcanan güç miktarıdır. Mikrovat seviyesindedir.
- **Çıkış Yelpazesi (Fan Out):** Kapının çıkışına bağlanacak yada çalışmasını etkilemeyecek standart yük sayısını belirtir.
- **Gürültü Seviyesi (Noise Margin):** Devrenin çıkışında istenmeyen bir değişikliğe neden olmayan, sayısal devrenin giriş işaretine eklenen maksimum gürültü gerilimidir. mV seviyesindedir.