Tahmini Ders İçeriği
(Tentative Couse Schedule – Syllabus)

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

1. Hafta: Sayısal Sinyaller/Sistemler, İkilik Tabanda Sayılar, Taban Aritmetiği, İşaretli/Eksi Sayıların Gösterimi, Sayısal Tasarım Tarihçesi

2. Hafta: İkili Mantık Aritmetiği ve Kapıları, Bool Cebiri Teorisi ve Tanımları, Bool Fonksiyonları, Kapı-Seviyesinde Yalınlaştırma, Karnough Haritası, Önemsenmeyen Durumlar, NAND, NOR, XOR

3-4. Hafta: Birleşik (Combinational) Devreler, Aritmetik Modüller, Decoder, Encoder, Mux, Verilog HDL

Lab Sınavı

5-6. Hafta: Ardışık (Sequential) Devreler, Mandal (Latch), Flip-Flop, Zamanlama (Timing)

Proje Duyurusu

7. Hafta: Durum Makinaları, Örnek Tasarımlar

8. Hafta: Yazmaçlar (Registers), Sayaçlar (Counters)

Ara Sınav

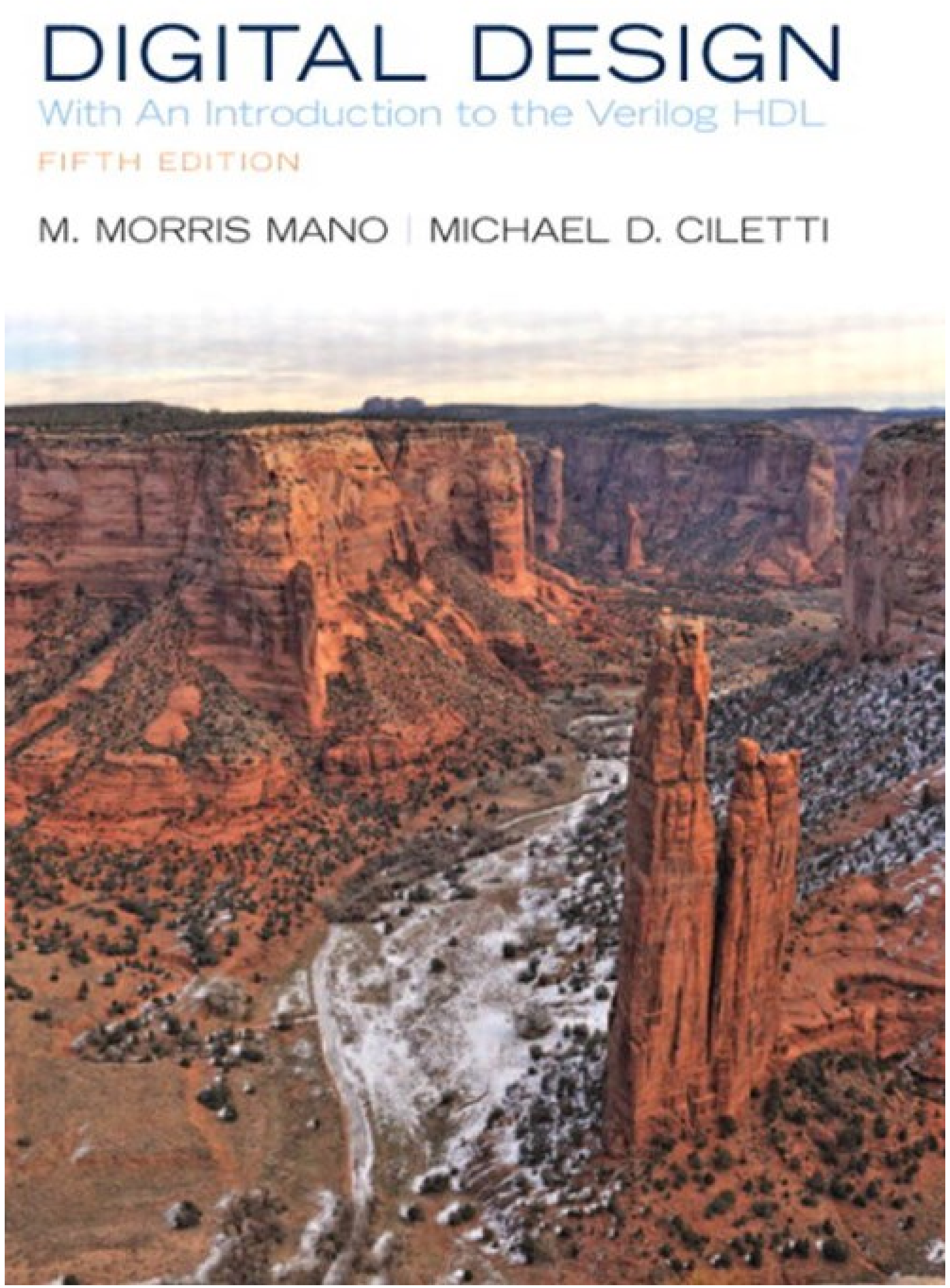
9. Hafta: Bellekler, FPGA'da Block RAM, OpenRAM

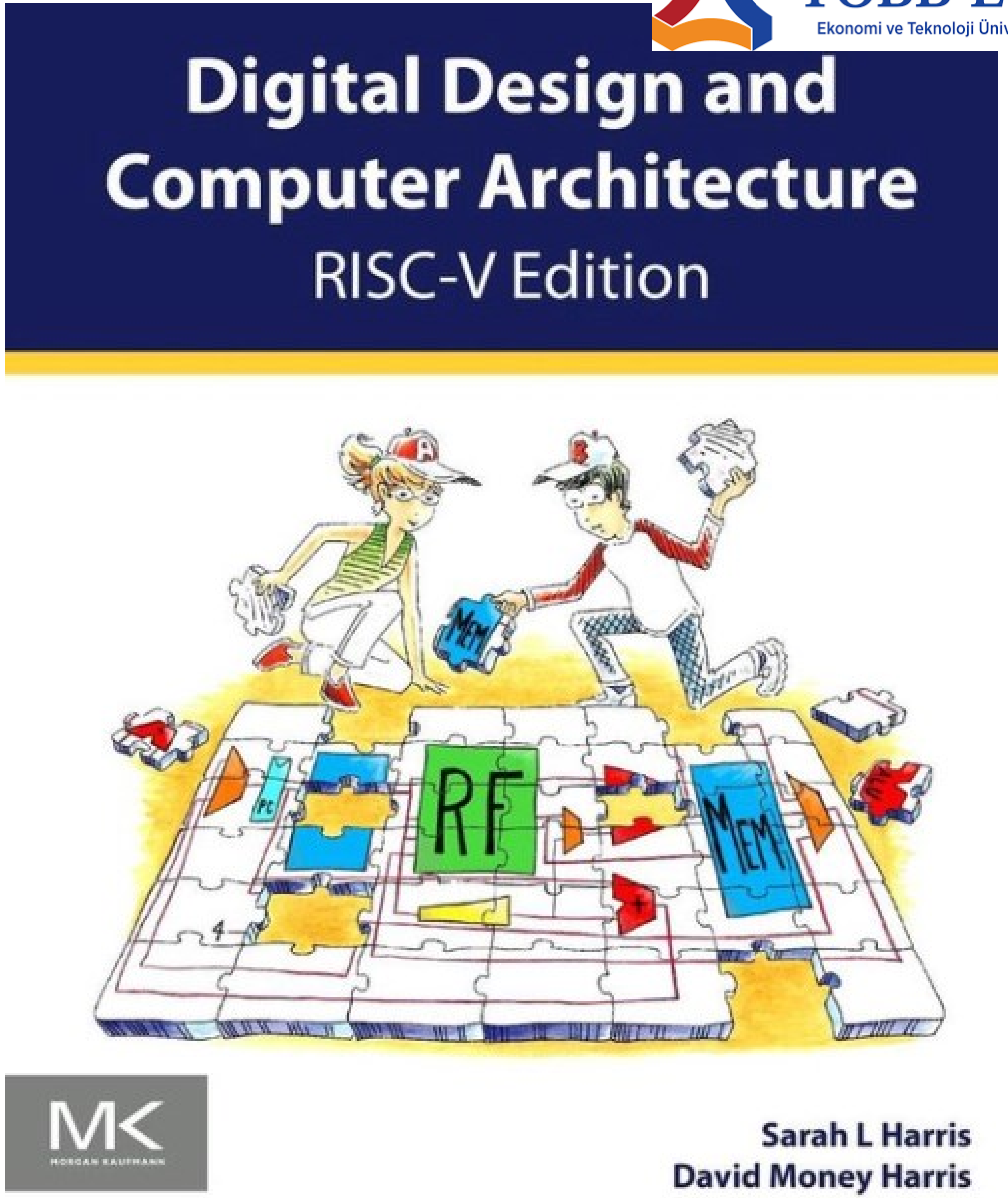
10. Hafta: RTL (Register Transfer Level) ASMD (Algorithmic State Machine and Datapath) Tasarımları

11-12. Hafta: Boru hattı, FPGA ve ASIC Tasarım Akışları

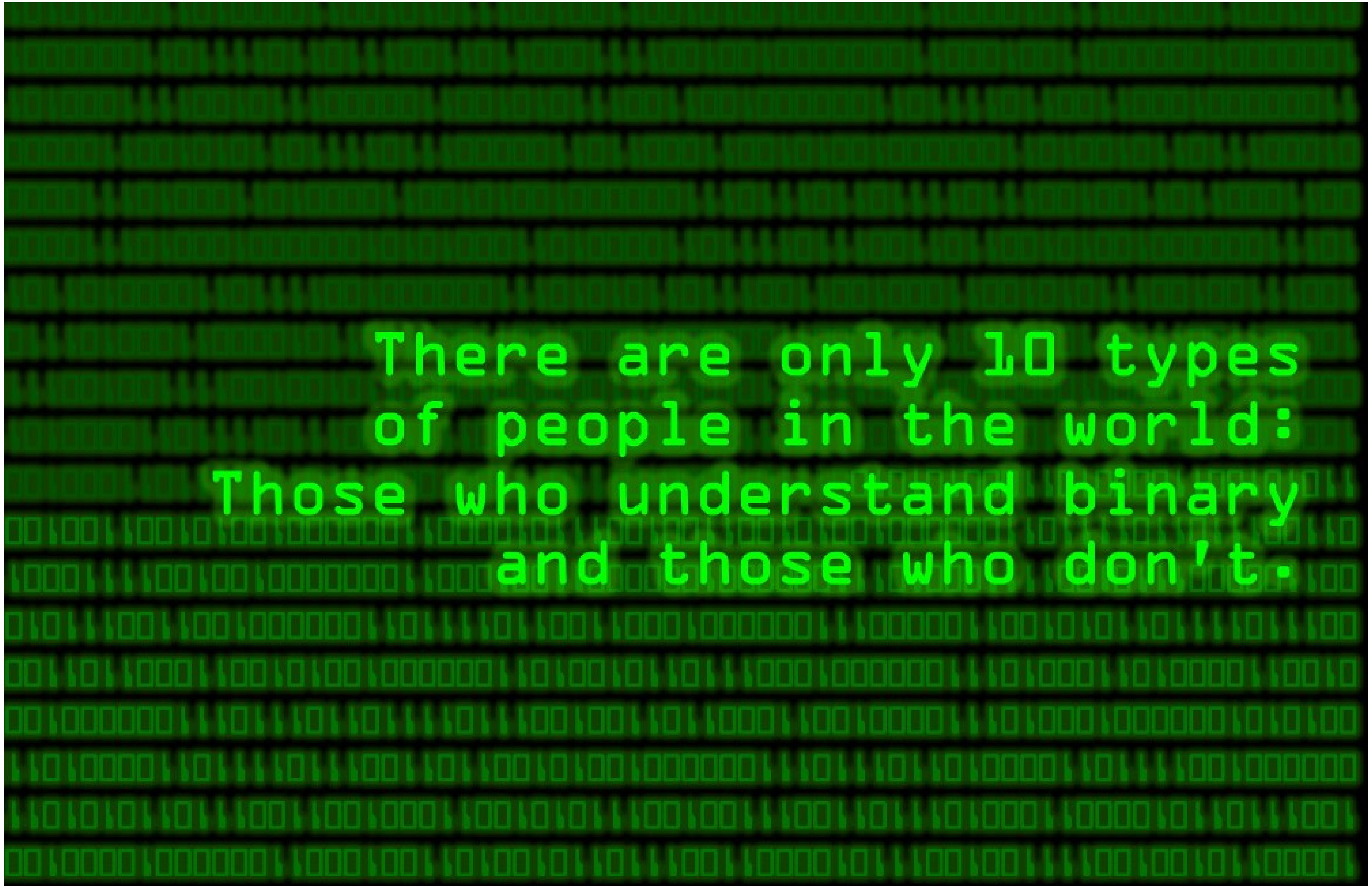
Final – Proje Teslimleri

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi





İlk 4 Chapter


There are only 10 types
of people in the world:
Those who understand binary
and those who don't.

2’li Mantık (Binary Logic)

Lise matematik’ten mantık konusunu hatırlayan var mı?

VE, VEYA, İSE, ANCAK VE ANCAK !?

İkili mantıksal işlemler **bit** adı verilen 2’lik taban sayılar üzerinde tanımlanır

1101 : 4-bit sayı

00001100 : 8-bit sayı

Temel 2’li mantıksal operasyonlar: VE – VEYA – DEĞİL (İng. AND – OR - NOT)

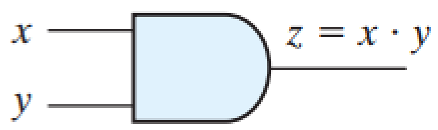
Doğruluk Tablosu (Truth Table)

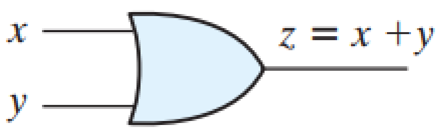
VE (AND)		
Giriş		Çıkış
x	y	x.y
0	0	0
0	1	0
1	0	0
1	1	1

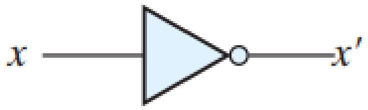
VEYA (OR)		
Giriş		Çıkış
x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

DEĞİL (NOT)	
Giriş	Çıkış
x	x'
0	1
1	0

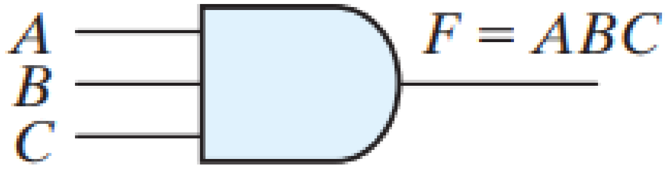
Kapı-Seviyesi Gösterimi (İng. Gate-Level)





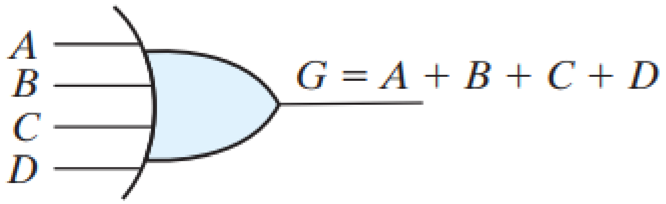


Doğruluk Tablosu (Truth Table)



VE (AND)			
Giriş			Çıkış
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

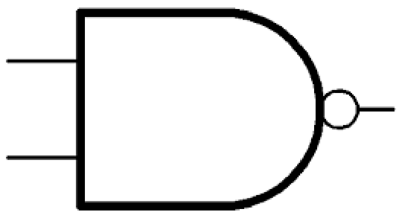
Doğruluk Tablosu (Truth Table)



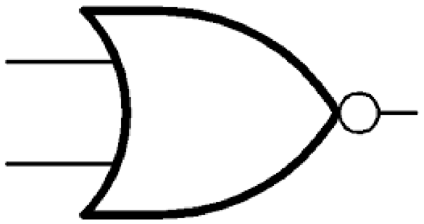
VEYA (OR)				
Giriş				Çıkış
A	B	C	D	G
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Doğruluk Tablosu (Truth Table)

VE DEĞİL (NAND)		
Giriş		Çıkış
x	y	(x.y)’
0	0	1
0	1	1
1	0	1
1	1	0

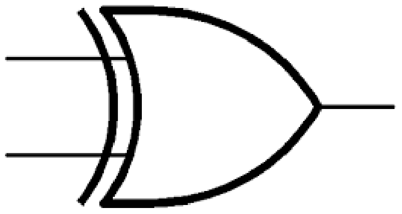


VEYA DEĞİL (NOR)		
Giriş		Çıkış
x	y	(x+y)’
0	0	1
0	1	0
1	0	0
1	1	0



ÖZEL VEYA (XOR)		
Giriş		Çıkış
x	y	x^y
0	0	0
0	1	1
1	0	1
1	1	0

x^y = x’y + xy’



Bool Cebiri (Boolean Algebra)

1854 Geogre Bool, cebirsel sistem: Bool Cebiri
1904 Huntington postulasyonları
1938 Shannon, 2 değerli Bool cebiri (switching algebra)

Bool cebirinde 6 adet postulasyon/axiom (esas, varsayım) mevcuttur:

“A postulate is a statement accepted to be true without proof”

Benzer bir örnek için vector uzayı (İng. Vector Space Axioms) araştırabilirsiniz

Bool Cebiri (Boolean Algebra)

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

Bool cebiri, küme elemanları, B, ve 2 adet operatör/işlem (+, .) ile aşağıdaki postulasyonları gerçekleştiren bir cebirsel yapıdır:

1) Bool cebiri, + ve . işlemlerine göre **kapalıdır** (**closed**)
2) ‘0’ elemanı + işlemine göre, ‘1’ elemanı . işlemine göre **etkisiz elemandır** (**identity**)
3) + ve . işlemleri **değişme** özelliğine sahiptir (**commutative**)
4) + işleminin . işlemi üzerinde ve . işleminin + işlemi üzerinde **dağılma** özelliği vardır (**distributive**)
5) B kümesindeki her bir x elemanı için, yine B kümesinde x + x’ = 1 ve x.x’ = 0 işlemlerini sağlayan bir x’ **tümleyen** elemanı vardır (**complement**)
6) B kümesinde x != y özelliğini sağlayan en az 2 eleman vardır

Bool Cebiri (Boolean Algebra)

Bool cebiri, küme elemanları, B, ve 2 adet operatör/işlem (+, .) ile aşağıdaki postulasyonları gerçekleştiren bir cebirsel yapıdır:

1) Bool cebiri, + ve . işlemlerine göre **kapalıdır** (**closed**)

S = {x,y}; x = 0, y = 1 : x.x, x.y, y.y, x+x, x+y, y+y yine S kümesinin elemanıdır

2) ‘0’ elemanı + işlemine göre, ‘1’ elemanı . işlemine göre **etkisiz elemandır** (**identity**)

$x + 0 = x$ $\quad$ $x.1 = x$

3) + ve . işlemleri **değişme** özelliğine sahiptir (**commutative**)

$x + y = y + x$ $\quad$ $x.y = y.x$

Bool Cebiri (Boolean Algebra)

Bool cebiri, küme elemanları, B, ve 2 adet operatör/işlem (+, .) ile aşağıdaki postulasyonları gerçekleştiren bir cebirsel yapıdır:

4) + işleminin . işlemi üzerinde ve . işleminin + işlemi üzerinde **dağılma** özelliği vardır (**distributive**)

$x + (y.z) = (x+y).(x+z)$ $x.(y+z) = (x.y) + (x.z)$

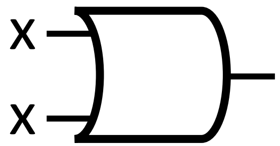
5) B kümesindeki her bir x elemanı için, yine B kümesinde x + x’ = 1 ve x.x’ = 0 işlemlerini sağlayan bir x’ **tümleyen** elemanı vardır (**complement**)

6) B kümesinde x != y özelliğini sağlayan en az 2 eleman vardır
→ 0 != 1   1 != 0

Bool Cebiri (Boolean Algebra)

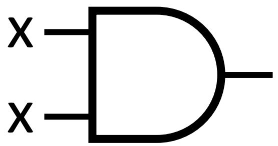
Ekstra çıkarımlar:

$x+x=x$ **İkilik (Duality)** $x.x = x$



=

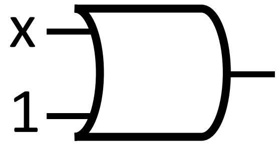
x—



=

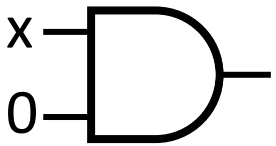
x—

$x+1=1$ **İkilik (Duality)** $x.0 = 0$



=

1—



=

0—

Bool Cebiri – İkilik (Duality)

0 ve 1 sembolleri ve . ve + operatörleri karşılıklı değiştirilirse kural yine geçerli olur

	Axiom		Dual	Name
A1	$B = 0$ if $B \neq 1$	A1′	$B = 1$ if $B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2′	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	A3′	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	A4′	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5′	$1 + 0 = 0 + 1 = 1$	AND/OR

$x+yz = (x+y)(x+z)$

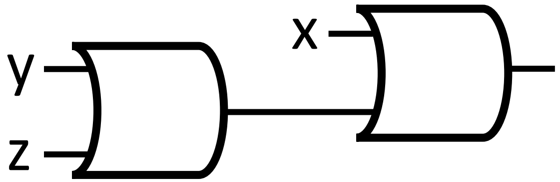
$x(y+z) = xy+xz$

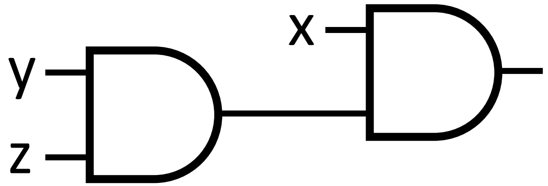
Bool Cebiri (Boolean Algebra)

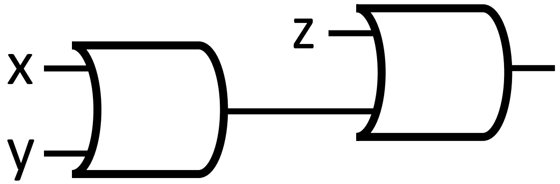
Ekstra çıkarımlar:

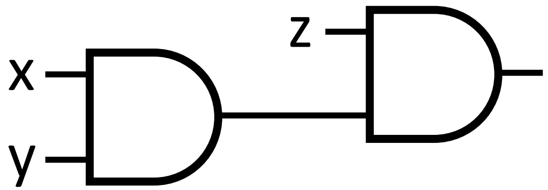
+ ve . işlemleri **ilişkiseldir** (**associativity**)

$x+(y+z) = (x+y)+z$ $\qquad$ $x.(y.z) = (x.y).z$





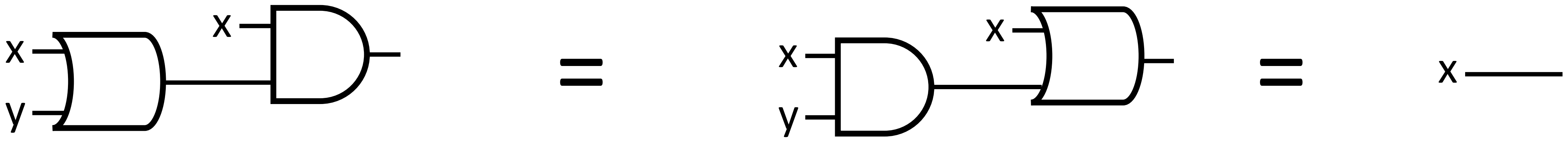




Bool Cebiri (Boolean Algebra)

Ekstra çıkarımlar:

$x.(x+y) = x+(x.y) = x$ **(covering/absorption)**

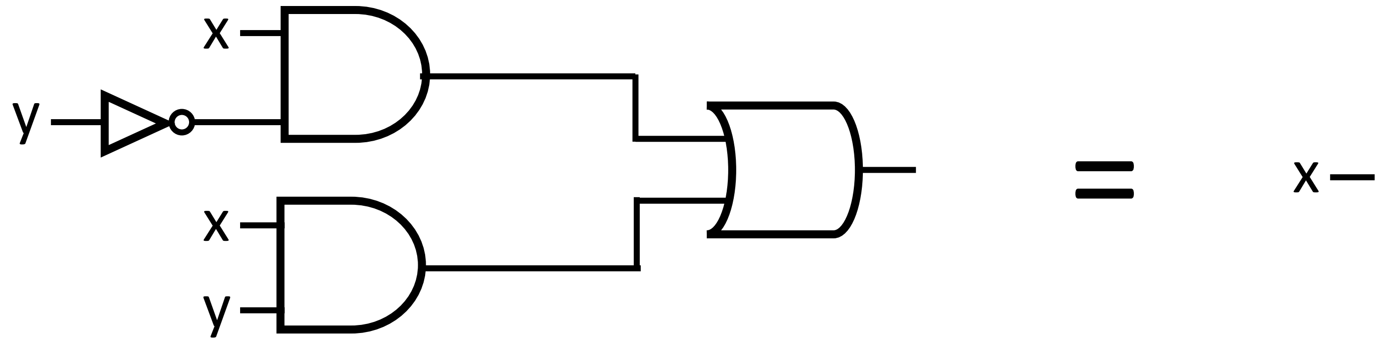


Bool Cebiri (Boolean Algebra)

Ekstra çıkarımlar:

$(x.y)+(x.y') = (x+y).(x+y') = x$ **(combining)**

$\rightarrow$ $(x.y+x).(x.y+y') = x.(y'+x.y) = x.(y'+x).(y'+y) = x.(x+y').1 = x.(x+y') = x$



Bool Cebiri (Boolean Algebra)

Ekstra çıkarımlar:

De Morgan Kuralı

(x.y)’ = x’.y’   (x+y)’ = x’+y’   (x.y.z)’ = x’+y’+z’

(x+y+z)’ = x’y’z’

x+x’=1   x.x’=0

(x’)’ = x

NAND: $Y = \overline{AB} = \overline{A} + \overline{B}$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR: $Y = \overline{A + B} = \overline{A}\,\overline{B}$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Figure 2.19 De Morgan equivalent gates

Bool Cebiri – İkilik (Duality)

	Theorem		Dual	Name
T6	$B \bullet C = C \bullet B$	T6′	$B + C = C + B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	T7′	$(B + C) + D = B + (C + D)$	Associativity
T8	$(B \bullet C) + (B \bullet D) = B \bullet (C + D)$	T8′	$(B + C) \bullet (B + D) = B + (C \bullet D)$	Distributivity
T9	$B \bullet (B + C) = B$	T9′	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	T10′	$(B + C) \bullet (B + \overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D)$ $= (B \bullet C) + (\overline{B} \bullet D)$	T11′	$(B + C) \bullet (\overline{B} + D) \bullet (C + D)$ $= (B + C) \bullet (\overline{B} + D)$	Consensus
T12	$\overline{B_0 \bullet B_1 \bullet B_2 \ldots}$ $= (\overline{B}_0 + \overline{B}_1 + \overline{B}_2 \ldots)$	T12′	$\overline{B_0 + B_1 + B_2 \ldots}$ $= (\overline{B}_0 \bullet \overline{B}_1 \bullet \overline{B}_2 \ldots)$	De Morgan's Theorem

İspatlar

THEOREM 1(a): $x + x = x$.

Statement	Justification
$x + x = (x + x) \cdot 1$	postulate 2(b)
$= (x + x)(x + x')$	5(a)
$= x + xx'$	4(b)
$= x + 0$	5(b)
$= x$	2(a)

Postülatlar

1. (a) The structure is closed with respect to the operator $+$.
(b) The structure is closed with respect to the operator $\cdot$.

2. (a) The element 0 is an identity element with respect to $+$; that is, $x + 0 = 0 + x = x$.
(b) The element 1 is an identity element with respect to $\cdot$; that is, $x \cdot 1 = 1 \cdot x = x$.

3. (a) The structure is commutative with respect to $+$; that is, $x + y = y + x$.
(b) The structure is commutative with respect to $\cdot$; that is, $x \cdot y = y \cdot x$.

4. (a) The operator $\cdot$ is distributive over $+$; that is, $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
(b) The operator $+$ is distributive over $\cdot$; that is, $x + (y \cdot z) = (x + y) \cdot (x + z)$.

5. For every element $x \in B$, there exists an element $x' \in B$ (called the *complement* of x) such that (a) $x + x' = 1$ and (b) $x \cdot x' = 0$.

6. There exist at least two elements $x, y \in B$ such that $x \neq y$.

İspatlar

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

THEOREM 6(a): $x + xy = x$.

Statement	Justification
$x + xy = x \cdot 1 + xy$	postulate 2(b)
$= x(1 + y)$	4(a)
$= x(y + 1)$	3(a)
$= x \cdot 1$	2(a)
$= x$	2(b)

Postülatlar

1. (a) The structure is closed with respect to the operator $+$.
(b) The structure is closed with respect to the operator $\cdot$.

2. (a) The element 0 is an identity element with respect to $+$; that is, $x + 0 = 0 + x = x$.
(b) The element 1 is an identity element with respect to $\cdot$; that is, $x \cdot 1 = 1 \cdot x = x$.

3. (a) The structure is commutative with respect to $+$; that is, $x + y = y + x$.
(b) The structure is commutative with respect to $\cdot$; that is, $x \cdot y = y \cdot x$.

4. (a) The operator $\cdot$ is distributive over $+$; that is, $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
(b) The operator $+$ is distributive over $\cdot$; that is, $x + (y \cdot z) = (x + y) \cdot (x + z)$.

5. For every element $x \in B$, there exists an element $x' \in B$ (called the *complement* of x) such that (a) $x + x' = 1$ and (b) $x \cdot x' = 0$.

6. There exist at least two elements $x, y \in B$ such that $x \neq y$.

Bool Cebiri (Boolean Algebra)

TOBB ETÜ Ekonomi ve Teknoloji Üniversitesi

Ekstra çıkarımlar: x.(x+y) = x+(x.y) = x **(covering/absorption)**

Giriş		Çıkış	Çıkış
x	y	x(x+y)	x+(xy)
0	0	0(0+0) = 0	0+(00) = 0
0	1	0(0+1) = 0	0+(01) = 0
1	0	1(1+0) = 1	1+(10) = 1
1	1	1(1+1) = 1	1+(11) = 1

THEOREM 6(a): $x + xy = x$.

Statement	Justification
$x + xy = x \cdot 1 + xy$	postulate 2(b)
$= x(1 + y)$	4(a)
$= x(y + 1)$	3(a)
$= x \cdot 1$	2(a)
$= x$	2(b)

DUALITY

THEOREM 6(b): $x(x + y) = x$ by duality.

1. (a) The structure is closed with respect to the operator $+$.
(b) The structure is closed with respect to the operator $\cdot$.

2. (a) The element 0 is an identity element with respect to $+$; that is, $x + 0 = 0 + x = x$.
(b) The element 1 is an identity element with respect to $\cdot$; that is, $x \cdot 1 = 1 \cdot x = x$.

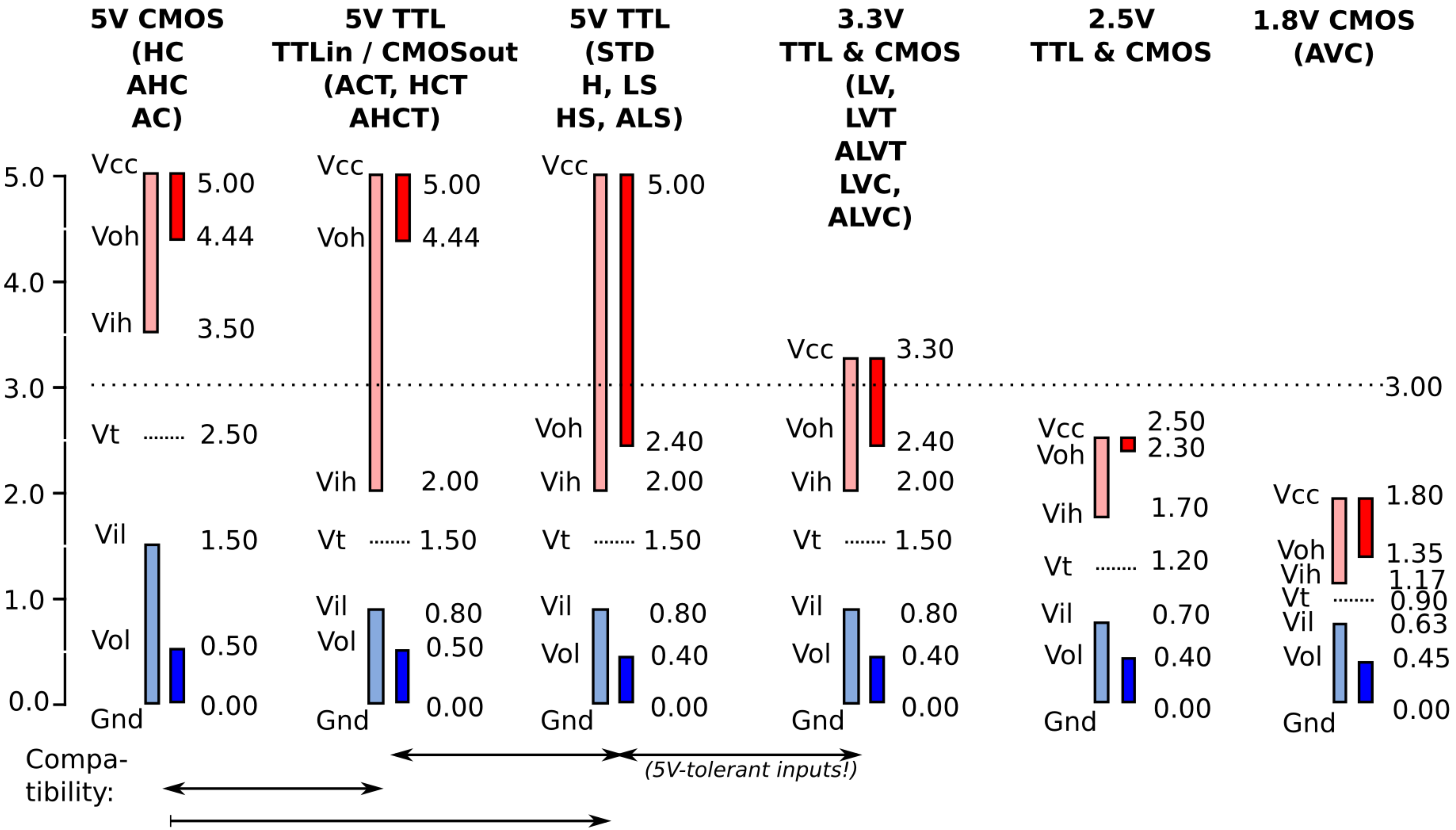
3. (a) The structure is commutative with respect to $+$; that is, $x + y = y + x$.
(b) The structure is commutative with respect to $\cdot$; that is, $x \cdot y = y \cdot x$.

4. (a) The operator $\cdot$ is distributive over $+$; that is, $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
(b) The operator $+$ is distributive over $\cdot$; that is, $x + (y \cdot z) = (x + y) \cdot (x + z)$.

5. For every element $x \in B$, there exists an element $x' \in B$ (called the *complement* of x) such that (a) $x + x' = 1$ and (b) $x \cdot x' = 0$.

6. There exist at least two elements $x, y \in B$ such that $x \neq y$.

Logic 0&1 Gerilim (Voltaj) Değerleri



Data source: EETimes, A brief recap of popular logic standards (Mark Pearson, Maxim).

Logic 0&1 Gerilim (Voltaj) Değerleri

Data Sheet **AD4111**

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
LOGIC INPUTS					
Input Voltage[4]					
High, V_{INH}	2 V ≤ IOVDD < 2.3 V	0.65 × IOVDD			V
	2.3 V ≤ IOVDD ≤ 5.5 V	0.7 × IOVDD			V
Low, V_{INL}	2 V ≤ IOVDD < 2.3 V			0.35 × IOVDD	V
	2.3 V ≤ IOVDD ≤ 5.5 V			0.7	V
Hysteresis[4]	IOVDD ≥ 2.7 V	0.08		0.25	V
	IOVDD < 2.7 V	0.04		0.2	V
Leakage Current		−10		+10	μA
LOGIC OUTPUT (DOUT/$\overline{RDY}$)					
Output Voltage					
High, V_{OH}	IOVDD ≥ 4.5 V, I_{SOURCE} = 1 mA	0.8 × IOVDD			V
	2.7 V ≤ IOVDD < 4.5 V, I_{SOURCE} = 500 μA	0.8 × IOVDD			V
	IOVDD < 2.7 V, I_{SOURCE} = 200 μA	0.8 × IOVDD			V
Low, V_{OL}	IOVDD ≥ 4.5 V, I_{SINK} = 2 mA			0.4	V
	2.7 V ≤ IOVDD < 4.5 V, I_{SINK} = 1 mA			0.4	V
	IOVDD < 2.7 V, I_{SINK} = 400 μA			0.4	V
Leakage Current	Floating state	−10		+10	μA
Output Capacitance	Floating state		10		pF

Ders Puanlaması

- **BİL-264**

10 proje (fpga flow) + 35 Ara Sınav (264/265) + 55 Final (Verilog) (264/264L/265)

Extralar (sınav ort > 50): 15 proje (caravel asic flow) + 2 Ödev (10)

- **BİL-265**

10 proje (fpga flow) + 15 1. Ara Sınav (264L/265) + 20 2. Ara Sınav (264/265) + 55 Final (Verilog) (264/264L/265)

Extralar (sınav ort > 50): 15 proje (caravel asic flow) + 4 Ödev (10)

- **BİL-264L**

45 Ara Sınav (264L/265) + 55 Final (264/264L/265)

Extralar (sınav ort > 50): 2 Ödev (20)

Proje Nasıl Olacak

Proje takımları 3-4 kişiden oluşabilecek

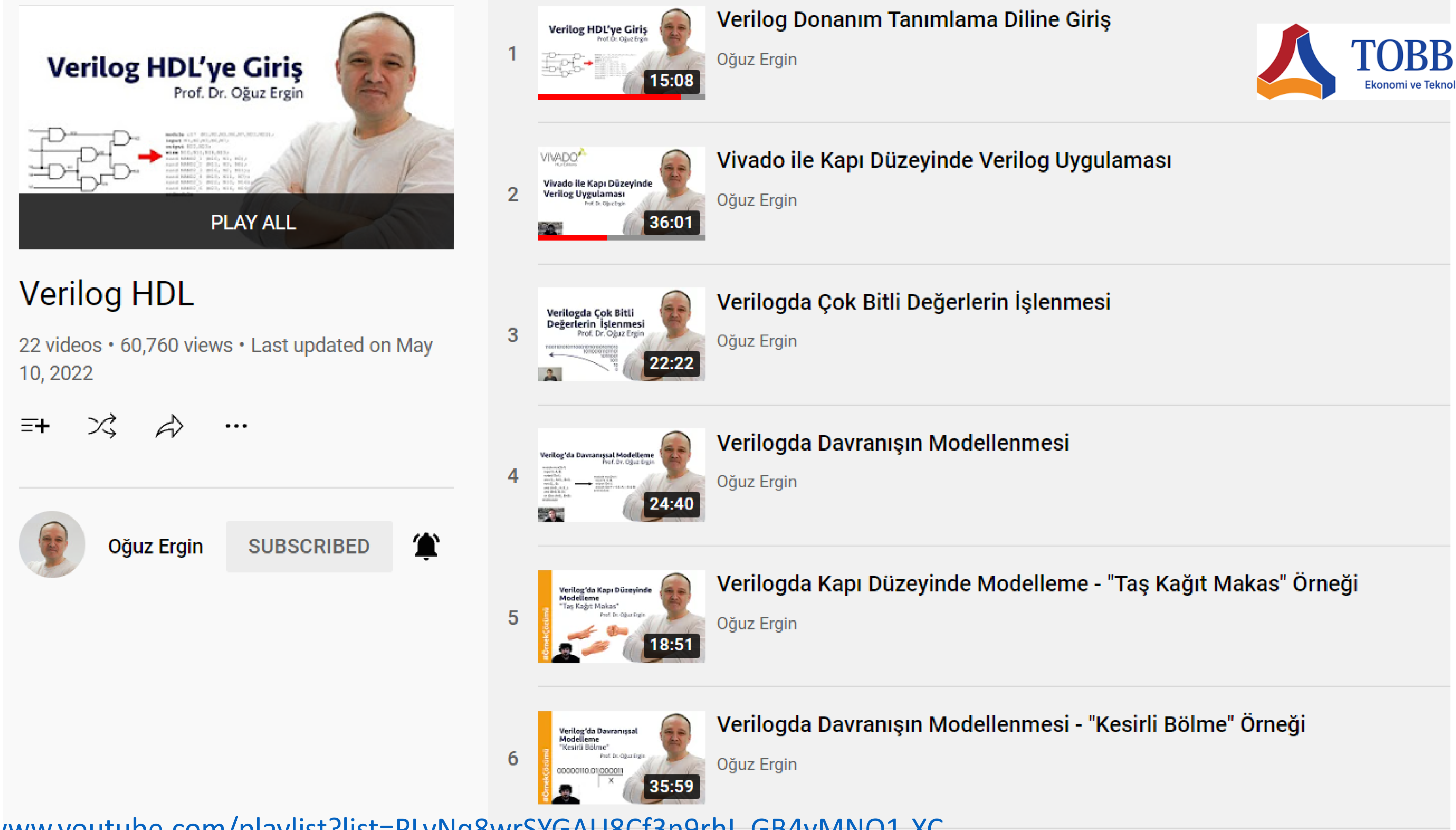
Her takımda en az bir elektronik ve bilgisayar öğrencisi olacak

Proje kapsamında verilecek olan tasarım 4 aşamadan oluşacaktır:

1) Verilog dili ile RTL tasarım (3 Puan)
2) Verilog dili ile simülasyon ve doğrulama (3 Puan)
3) Tasarımın FPGA üzerinde çalıştırması (4 Puan)
4) (Extra) OpenLane ile Caravel ortamında tape-out yapılması (15 Puan)

Proje yakında duyurulacak !!!

Lab Dersleri Kaydı Paylaşılmayacak – Çünkü Zaten Var



TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

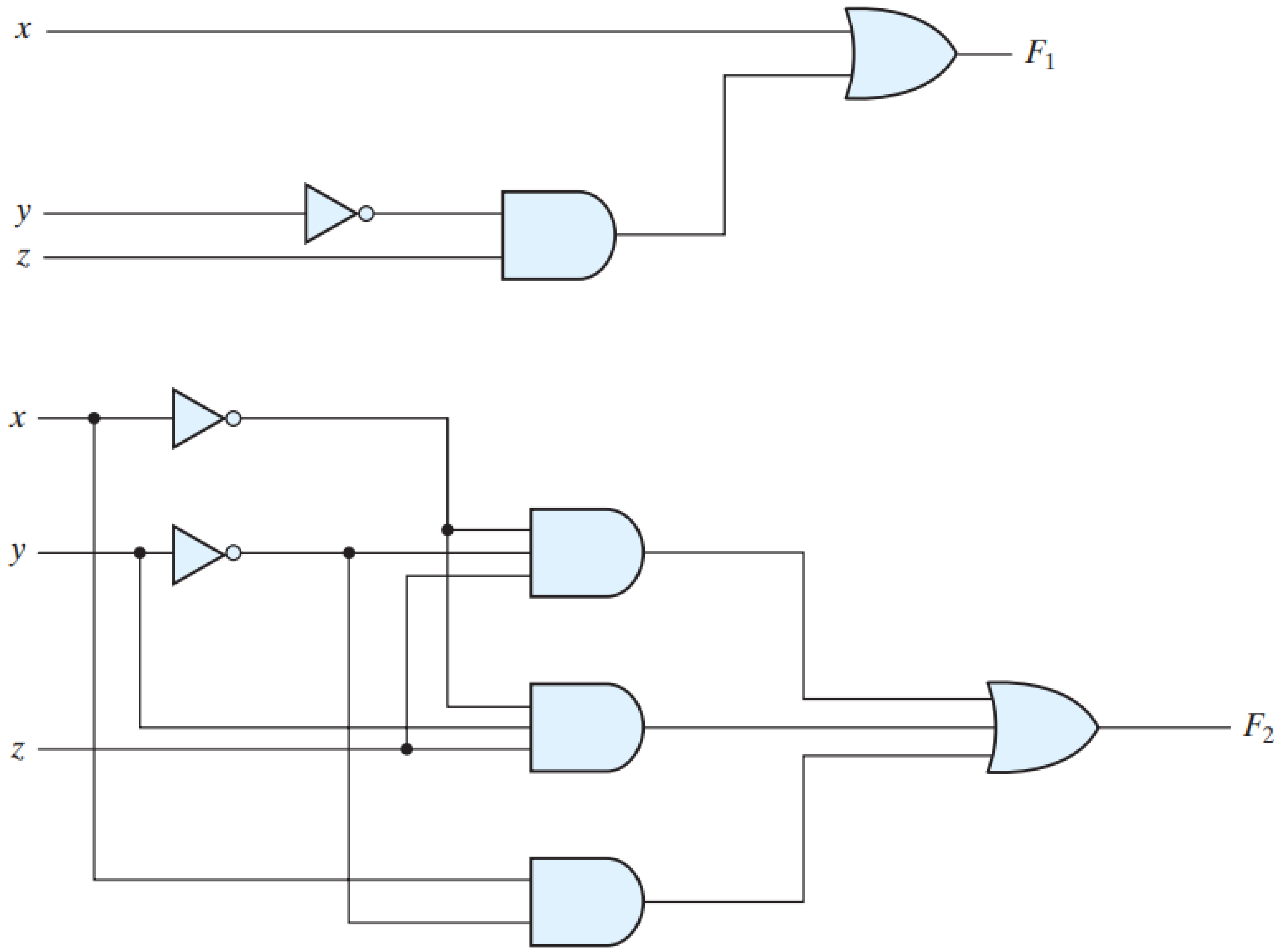
https://www.youtube.com/playlist?list=PLvNq8wrSYGAU8Cf3n9rhL-GB4yMNO1-XC

Bool Fonksiyonları

$F1 = x + y'z$

$F2 = x'y'z + x'yz + xy'$

F1 & F2				
Giriş			Çıkış	
x	y	z	F1	F2
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0



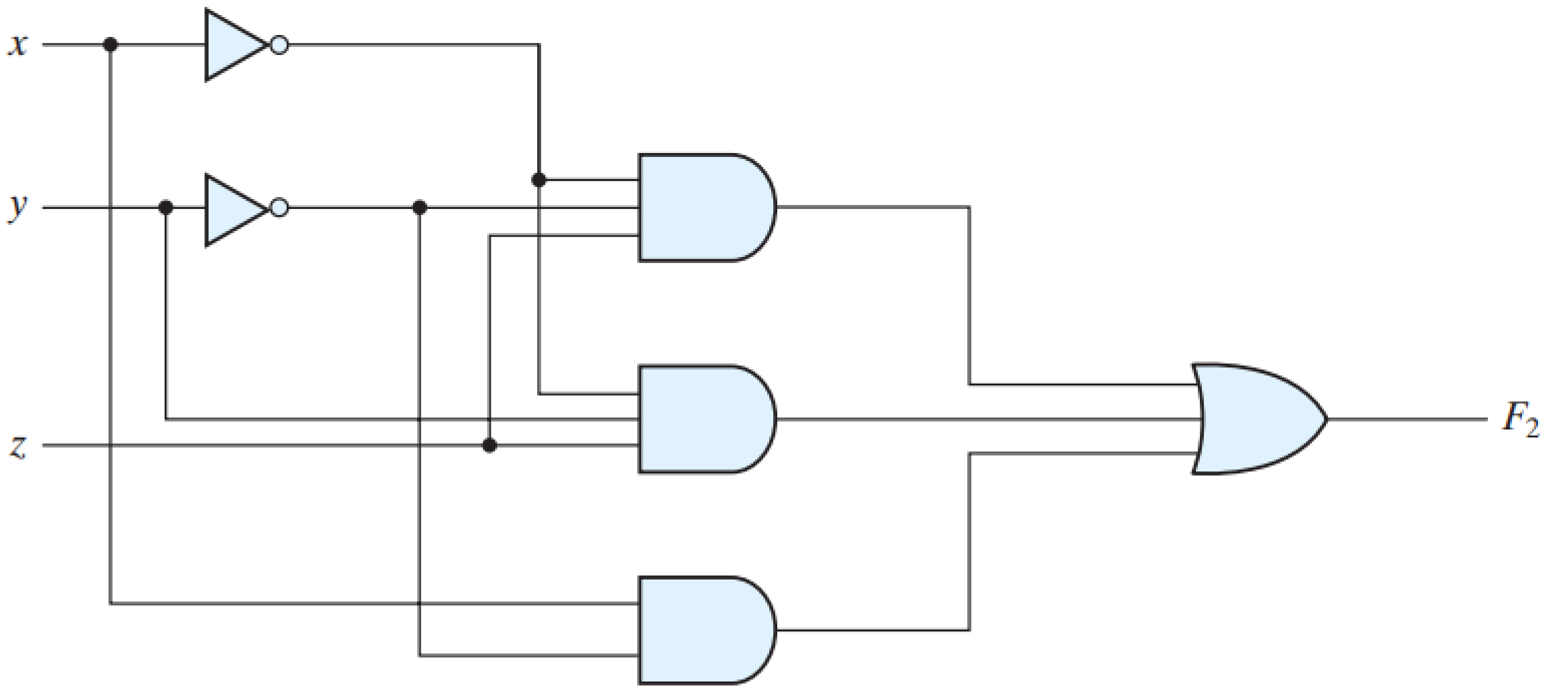
Bool Fonksiyonları - Sadeleştirme

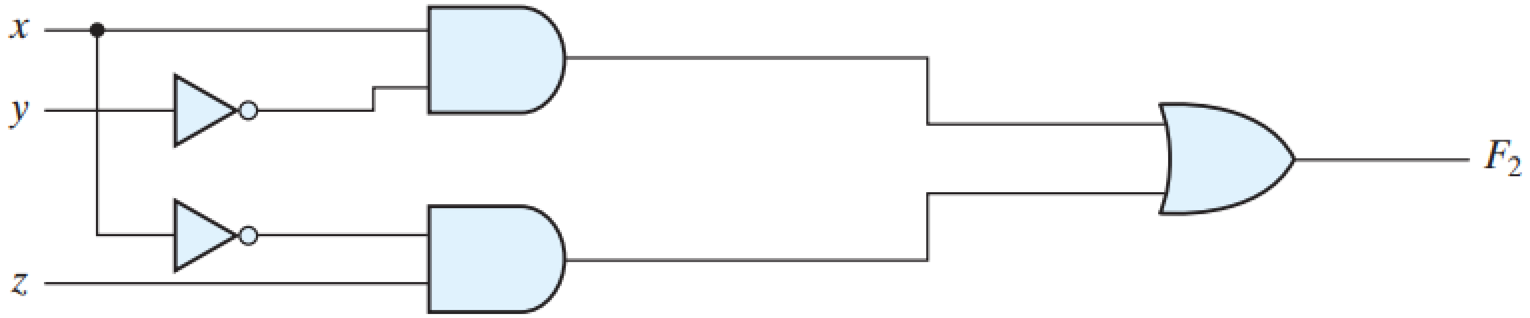
$F2 = x'y'z + x'yz + xy'$

$F2 = x'z(y'+y) + xy'$

$F2 = x'z + xy'$

Fonksiyonda sadeleştirme daha az kapı kullanımına, yani daha hızlı, daha az alana sahip, daha ucuz ve daha az güç tüketen devre demektir





Bool Fonksiyonları - Sadeleştirme

$F1 = x(x'+y) \rightarrow xx' + xy = 0 + xy = xy$

$F2 = x + x'y \rightarrow (x+x')(x+y) = 1(x+y) = x+y$

$F3 = (x+y)(x+y') \rightarrow x$ (combining) $\rightarrow xx+xy'+xy+yy' = x + x(y'+y) + 0 = x+x1 = x+x = x$

$F4 = xy + x'z + yz = xy+x'z+yz(x+x') = xy+x'z+xyz+x'yz = xy(1+z)+x'z(1+y) = xy+x'z$

$F5 = (x+y)(x'+z)(y+z) = (x+y)(x'+z)$ "F4'ün ikilisi (duality)"

Sadeleştirme yerine daha matematiksel ve analitik bir yöntem olan Karnough haritası sonraki derslerde anlatılacaktır!

Bool Fonksiyonları – Tümleyen (Complement)

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

F fonksiyonunun tümleyeni (complement): F’

$F1 = x(x'+y) \rightarrow xx' + xy = 0 + xy = xy$ $F1' = (xy)' = x'+y'$

$F2 = x + x'y \rightarrow (x+x')(x+y) = 1(x+y) = x+y$ $F2' = (x+y)' = x'y'$

$F3 = (x+y)(x+y') \rightarrow x$ (combining) $\rightarrow xx+xy'+xy+yy' = x + x(y'+y) + 0 = x+x1 = x+x = x$ $F3' = x'$

$F4 = xy + x'z + yz = xy+x'z+yz(x+x') = xy+x'z+xyz+x'yz = xy(1+z)+x'z(1+y) = xy+x'z$

$F4' = (xy+x'z)' = (xy)'(x'z)' = (x'+y')(x+z') = x'x + x'z' + y'x + y'z' = x'z'+xy'+y'z'$

$F5 = (x+y)(x'+z)(y+z) = (x+y)(x'+z)$ “F4’ün ikilisi (duality)”

$F5' = ((x+y)(x'+z))' = (x+y)'+(x'+z)' = x'y'+xz'$

Bool Fonksiyonları – Tümleyen (Complement)

$F1 = x'yz'+x'y'z \rightarrow F1' = (x'yz'+x'y'z)' = (x'yz')'(x'y'z)' = (x+y'+z)(x+y+z')$

$= xx+xy+xz'+y'x+y'y+y'z'+zx+zy+zz'=x+xy+xz'+xy'+y'z'=x+(xy+xy')+xz'+y'z'$

$= x+x+xz'+y'z' = x+xz'+y'z' = x(1+z')+y'z' = x+y'z'$

$F2 = x(y'z'+yz) \rightarrow F2' = (x(y'z'+yz))' = x'+(y'z'+yz)' = x'+(y'z')'(yz)' = x'+(y+z)(y'+z')$

$= x'+yy'+yz'+zy'+zz' = x'+yz'+y'z$

Duality Yöntemi ile Tümleyen (Complement)

Tümleyeni alınmak istenilen fonksiyona duality prensibi uygulanır. Sonuç olarak kalan her bir literalin (eleman) tümleyeni alınır.

Duality: 0 ve 1 sembolleri ve . ve + operatörleri karşılıklı değiştirilir

$F1 = x'yz'+x'y'z$ → $F1'$ için duality prensibi uygula → $(x'+y+z')(x'+y'+z)$
Her bir literalin tümleyenini al → $(x+y'+z)(x+y+z') = F1'$

$F2 = x(y'z'+yz)$ → $F2'$ için duality prensibi uygula → $x+(y'+z')(y+z)$
Her bir literalin tümleyenini al → $x'+(y+z)(y'+z') = F2'$

Maxterm – Minterm – Canonical Form

:) **Mamta Saini** · Follow

MBA from Indian Institute of Management Ranchi (Graduated 2020) · 4y

First thing first, they are called ***terms*** because they are used as the building-blocks of various canonical representations of arbitrary boolean functions.

Minterms are the product of literals which correspond to "1" in the K-maps. For example xy, x'yz'w

Maxterms are the sum of literals which correspond to "0" in the K-map. For example (x'+y'), (x+y'+z+w')

Clearly visible, the size of expression signifies which is minterm or maxterm. Maxterms involves more number of characters. But this is not the actual reason for maxterms and minterms being named so. The main reason is of the satisfiability being maximum or minimum as explained below.

$$G(w,x,y,z) = \Pi(0,2,5,8,10,13,15)$$
$$d(w,x,y,z) = \Sigma(4,11,14)$$

Sum of minterms is in the Sum of Products (SOP) form. So, there is *OR* operation between the minterms. Note here that OR has "**minimum**" **satisfiability**. Even if one minterm is true, the SOP will be true(1) irrespective of the value of other minterms.

Product of maxterms is in the Product of Sums (POS) form. So, there is *AND* operation between the maxterms. Here, AND has "**maximum**" **satisfiability** because the output will be true if and only if all the maxterms gives true(1).

Maxterm – Minterm – Canonical Form

A formula is valid if it is true for all values of its terms. Satisfiability refers to the existence of a combination of values to make the expression true. So in short, a proposition is satisfiable if there is at least one `true` result in its truth table, valid if all values it returns in the truth table are `true`.

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	y	z	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

Maxterm – Minterm – Canonical Form

TOBB ETÜ Ekonomi ve Teknoloji Üniversitesi

Table 2.4
Functions of Three Variables

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

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

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

Sum of minterms

Maxterm – Minterm – Canonical Form

Table 2.4
Functions of Three Variables

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

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

$$\Downarrow$$

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

$$\Downarrow$$

Product of maxterms

$$f_1 = (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z) \quad = M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

Canonical Form

Sum of minterms ya da product of maxterms şeklinde gösterilmiş olan Bool fonksiyonu canonical form (kanunsal biçim) olarak tanımlanır.

Canon: kanōn (Greek), rule or measuring stick (İngilizce), قانون (Arapça) → Kanun

Canonical: Kanunsal, yasal, hukuki → Canonical Law (Dini Hükümler)

$F1 = A+BC+AC$ → SoM (sum of minterms) canonical formda göster

$F1 = A(B+B')+BC(A+A')+AC(B+B') = AB+AB'+ABC+A'BC+ABC+AB'C = AB(C+C')+AB'(C+C')+ABC+A'BC+AB'C$

$= ABC+ABC'+AB'C+AB'C'+A'BC = m(111)+m(110)+m(101)+m(100)+m(011) = m7+m6+m5+m4+m3$

$F1 = m3+m4+m5+m6+m7$

Canonical Form

F1 = A+BC+AC → PoM (product of maxterms) canonical formda göster

F1 = m3+m4+m5+m6+m7 (bir önceki örnekten çıkardık)

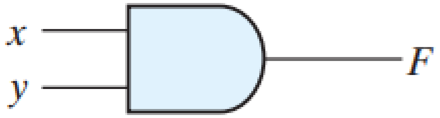
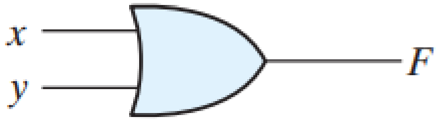
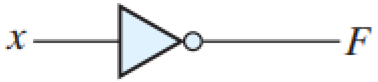
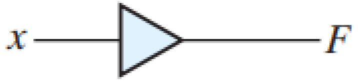
F1' = m0+m1+m2 → (F1')' = (m0+m1+m2)' → F1 = M0M1M2

Table 2.3
Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	y	z	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

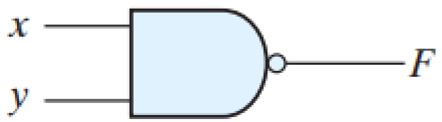
Giriş			Çıkış	
A	B	C	F1	F1'
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

MANTIKSAL KAPILAR

Name	Graphic symbol	Algebraic function	Truth table
AND		$F = x \cdot y$	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1
OR		$F = x + y$	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1
Inverter		$F = x'$	x | F 0 | 1 1 | 0
Buffer		$F = x$	x | F 0 | 0 1 | 1

MANTIKSAL KAPILAR

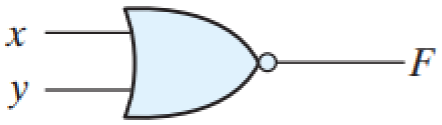
NAND



$F = (xy)'$

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

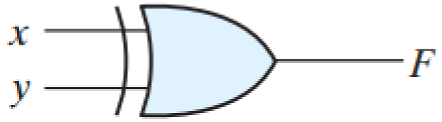
NOR



$F = (x + y)'$

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

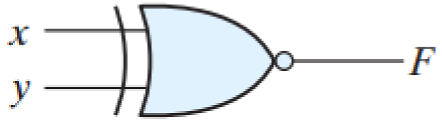
Exclusive-OR
(XOR)



$$F = xy' + x'y$$
$$= x \oplus y$$

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

Exclusive-NOR
or
equivalence



$$F = xy + x'y'$$
$$= (x \oplus y)'$$

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

MANTIKSAL KAPILAR (SKY130nm)

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

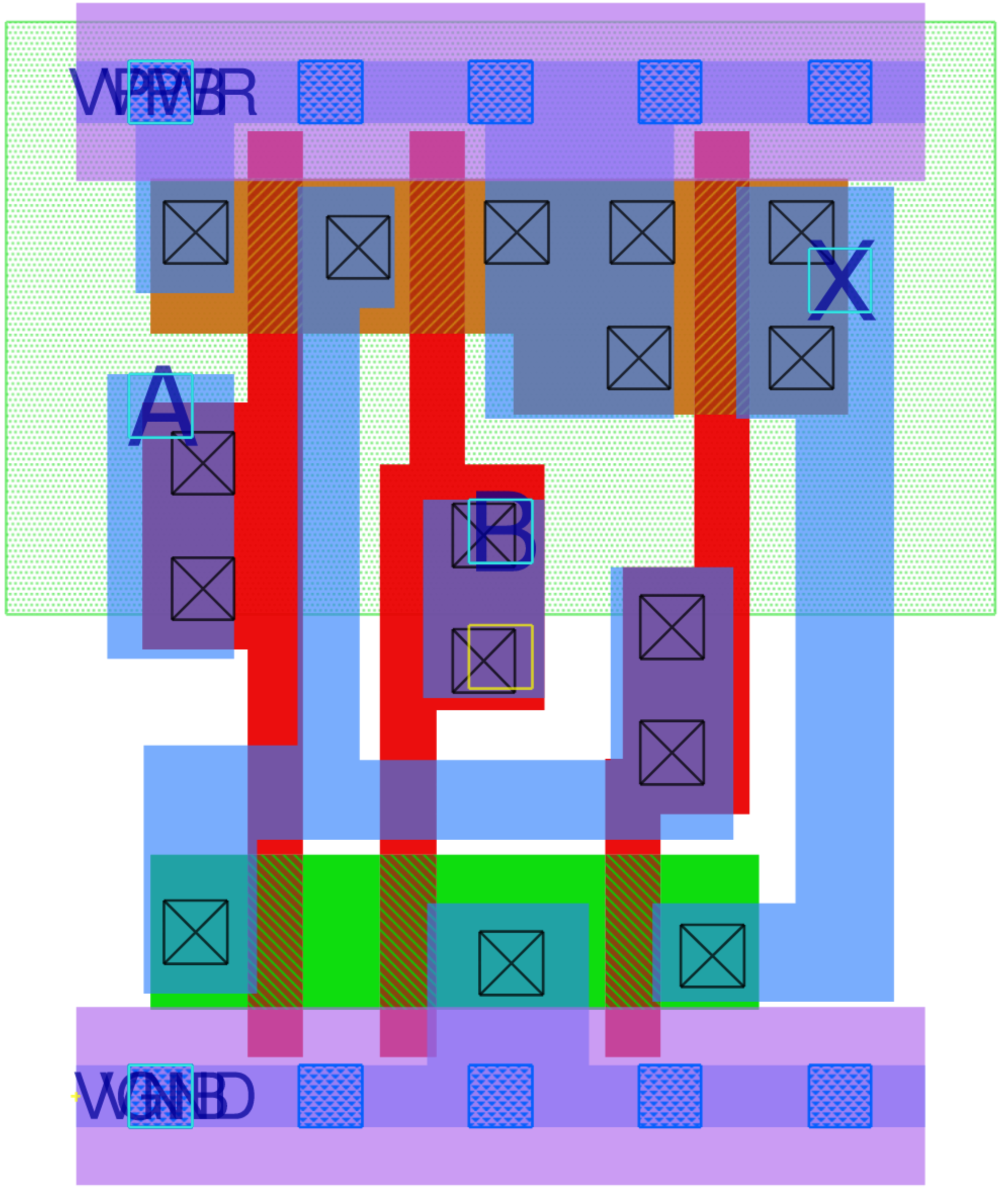
sky130_fd_sc_hd - SKY130 High Density Digital Standard Cells (SkyWater Provided)

Initial release of version (0, 0, 2).

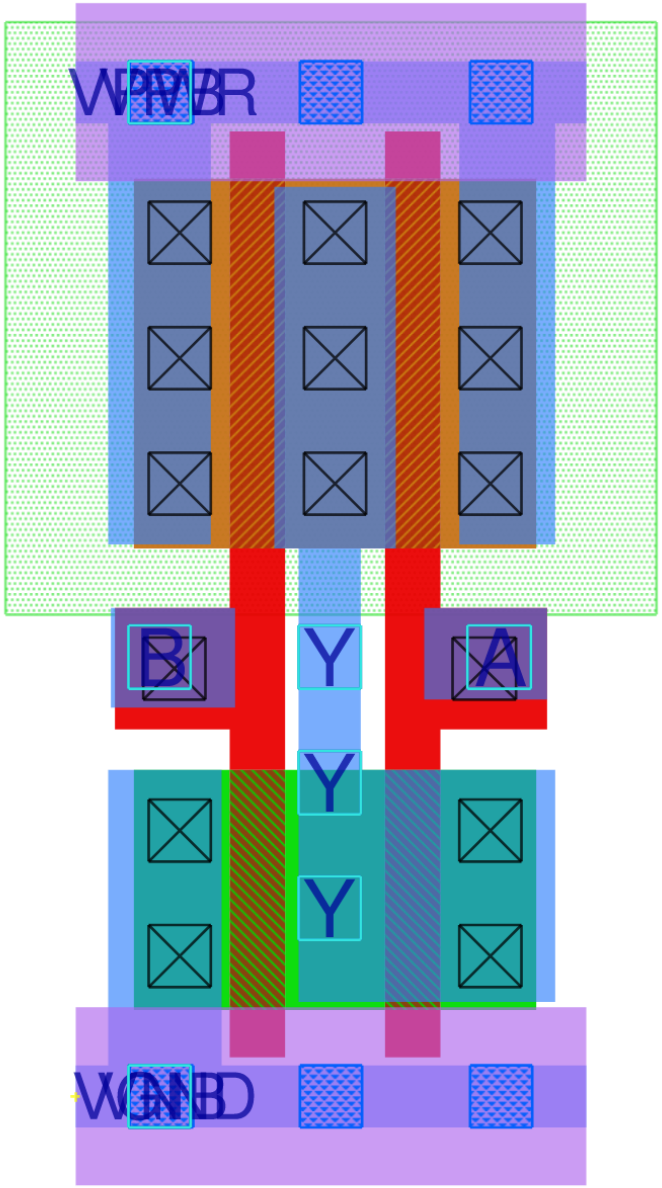
List of cells in sky130_fd_sc_hd

https://antmicro-skywater-pdk-docs.readthedocs.io/en/test-submodules-in-rtd/contents/libraries/sky130_fd_sc_hd/README.html

NAND – NOR KAPILARI (SKY130nm)

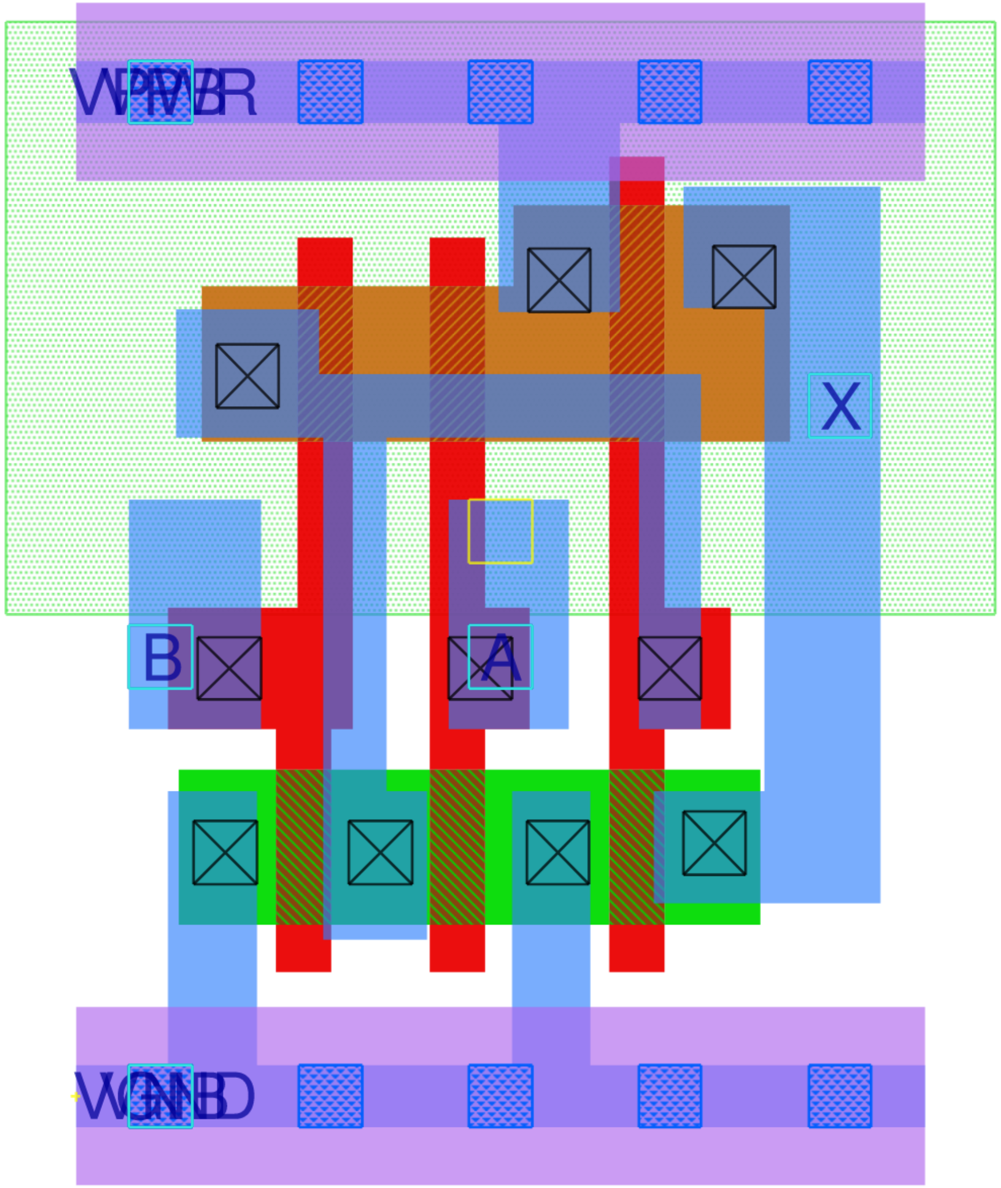


AND2

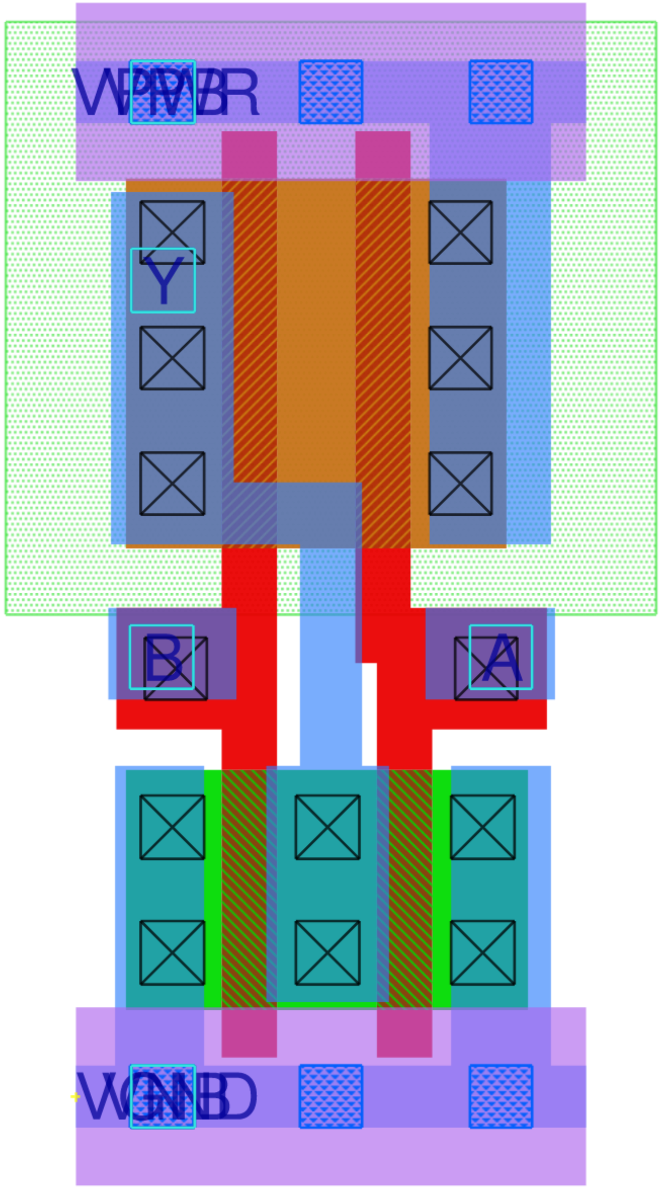


NAND2

NAND – NOR KAPILARI (SKY130nm)

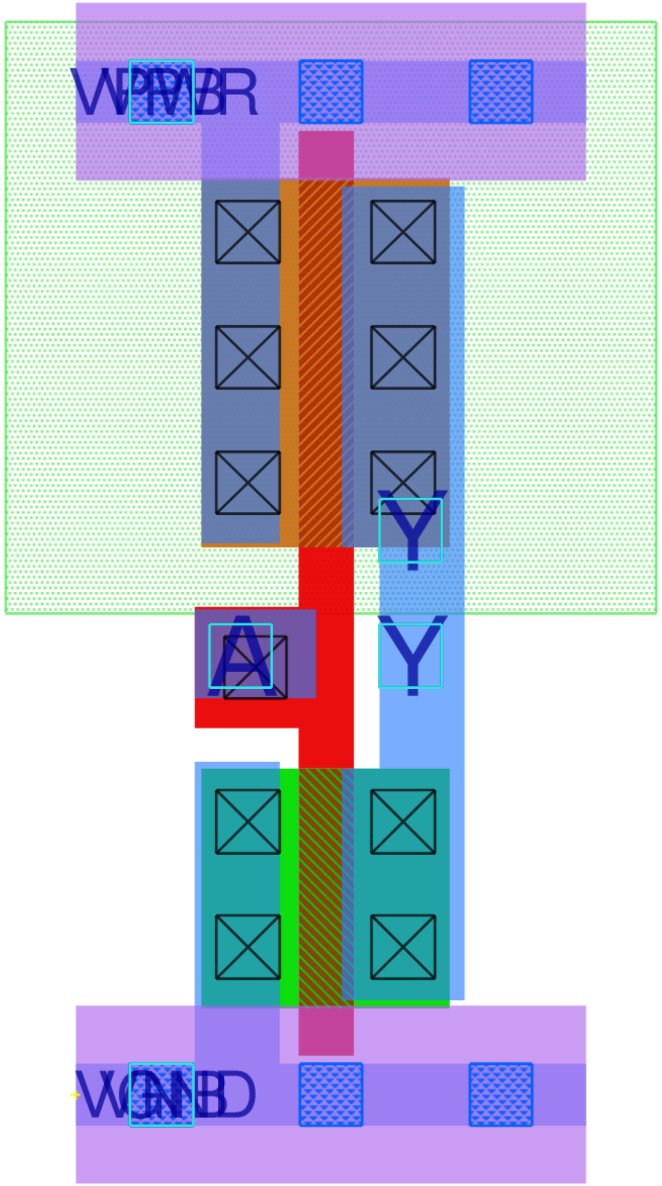


OR2

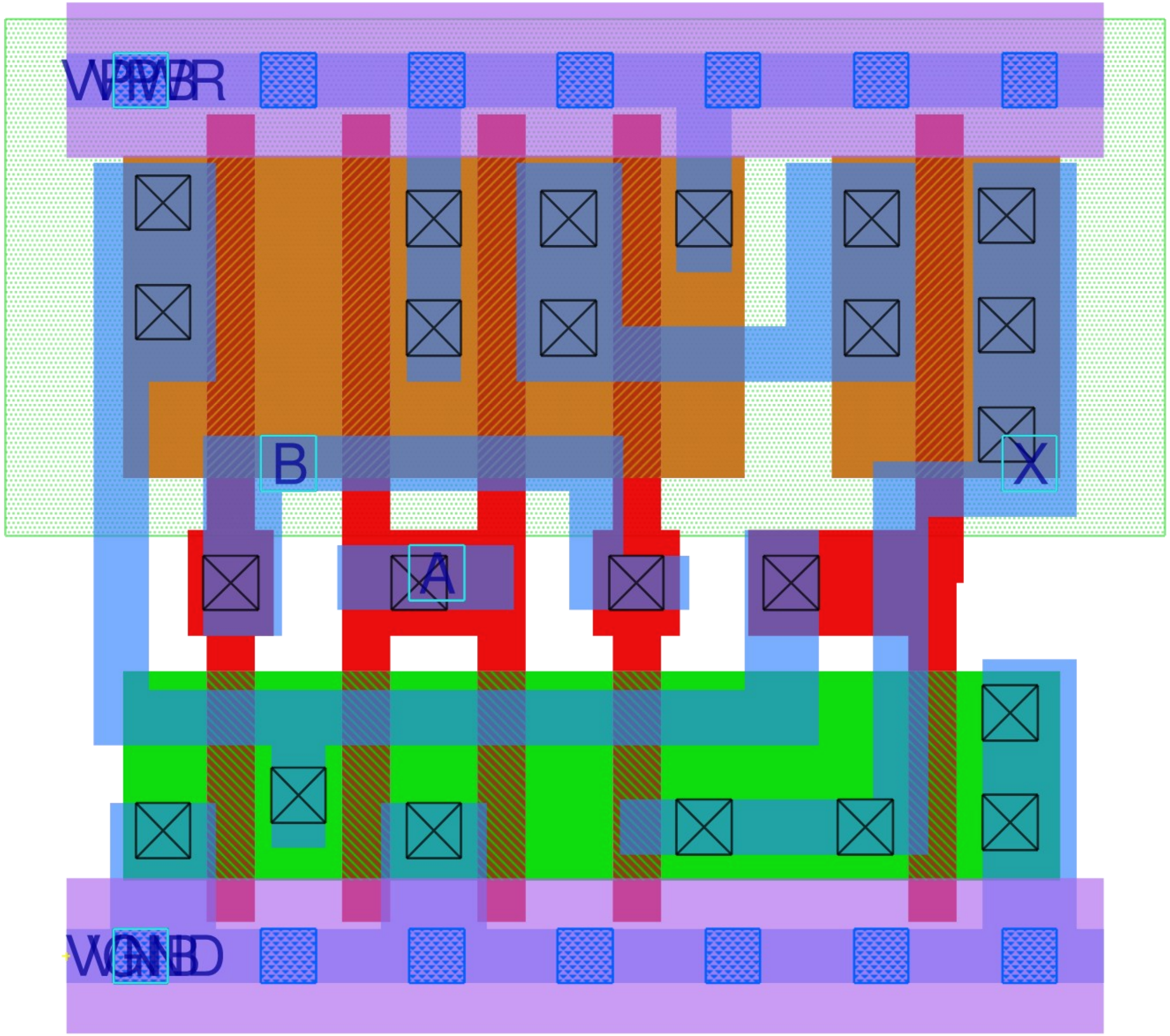


NOR2

NOT - XOR KAPILARI (SKY130nm)



INV



XOR2

KARNOUGH HARİTASI (K-MAP)

Bool fonksiyonlarında cebirsel sadeleştirme yöntemi kural ve her bir adım için algoritmik bir akış içermemekte.

Daha algoritmik, tahmin edilebilir ve uygulanabilir bir yönteme ihtiyaç var

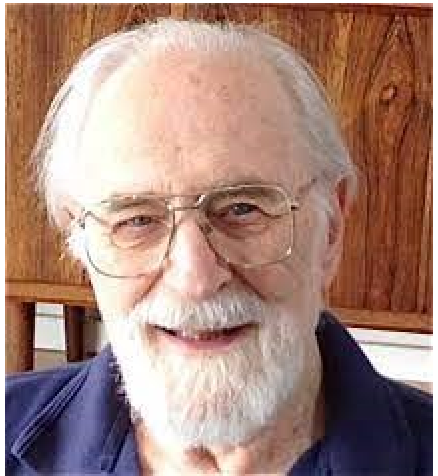
Sonuç: Karnough haritası (K-map) yöntemi

Yöntem: Fonksiyon giriş sinyallerine göre bütün Sum of Minterms elemanlarını içeren bir harita oluşturulur. Görsel olarak bazı paternler aranır ve buna göre sadeleştirme gerçekleştirilir.

Maurice Karnaugh

Fizikçi

Doğum tarihi: 4 Ekim 1924 (97 yıl yaşında), New York, New York, ABD



2 Girişli KARNOUGH HARİTASI (K-MAP)

m_0	m_1
m_2	m_3

$x \backslash y$	0	1
0	m_0 $x'y'$	m_1 $x'y$
1	m_2 xy'	m_3 xy

Giriş		Çıkış
x	y	F2
0	0	0
0	1	0
1	0	1
1	1	0

$x \backslash y$		1
0	0	0
1	1	0

F2 = xy’

Giriş		Çıkış
x	y	F1
0	0	0
0	1	0
1	0	0
1	1	1

$x \backslash y$		1
0	0	0
1	0	1

F1 = xy

Giriş		Çıkış
x	y	F3
0	0	0
0	1	1
1	0	1
1	1	0

$x \backslash y$		1
0	0	1
1	1	0

F3 = xy’ + x’y

2 Girişli KARNOUGH HARİTASI (K-MAP)

m_0	m_1
m_2	m_3

$x \backslash y$	0	1 (y)
0	m_0 $x'y'$	m_1 $x'y$
x 1	m_2 xy'	m_3 xy

Giriş		Çıkış
x	y	F5
0	0	1
0	1	0
1	0	1
1	1	0

		y 1
0	1	0
x { 1	1	0

F5 = y’

Giriş		Çıkış
x	y	F4
0	0	0
0	1	0
1	0	1
1	1	1

		y 1
0	0	0
x { 1	1	1

F4 = x

Giriş		Çıkış
x	y	F6
0	0	1
0	1	1
1	0	1
1	1	0

		y 1
0	1	1
x { 1	1	0

F6 = x’y’ + x’y + xy’

F6 = x’+ y’

2 Girişli KARNOUGH HARİTASI (K-MAP)

m_0	m_1
m_2	m_3

$x \backslash y$	0	$\overbrace{1}^{y}$
0	m_0 $x'y'$	m_1 $x'y$
$x\{\ 1$	m_2 xy'	m_3 xy

Giriş		Çıkış
x	y	F8
0	0	1
0	1	0
1	0	1
1	1	0

		$\overbrace{1}^{y}$
0	0	0
x { 1	0	0

F8 = 0

Giriş		Çıkış
x	y	F7
0	0	1
0	1	1
1	0	1
1	1	1

		$\overbrace{1}^{y}$
0	1	1
x { 1	1	1

F7 = 1

3 Girişli KARNOUGH HARİTASI (K-MAP)

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

Giriş			Çıkış
x	y	z	F1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

x \ yz	00	01	11	10
0	0	0	1	1
1	1	1	1	1

F1 = x + y

En fazla elemanı gruplamasak ne olur?

x \ yz	00	01	11	10
0	0	0	1	1
1	1	1	1	1

F1 = xy’ + y

x \ yz	00	01	11	10
0	0	0	1	1
1	1	1	1	1

F1 = xy’ + yz + yz’

3 Girişli KARNOUGH HARİTASI (K-MAP)

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
x { 1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

y: columns 11, 10; z: columns 01, 11

Giriş			Çıkış
x	y	z	F2
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

	00	01	11	10
0	1	0	1	0
x { 1	1	0	1	0

y: columns 11, 10; z: columns 01, 11

$$F2 = y'z' + yz$$
$$F2 = \sim(y \wedge z)$$

Giriş			Çıkış
x	y	z	F3
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

	00	01	11	10
0	0	1	0	1
x { 1	0	1	0	1

y: columns 11, 10; z: columns 01, 11

$$F3 = y'z + yz'$$
$$F3 = y \wedge z$$

3 Girişli KARNOUGH HARİTASI (K-MAP)

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

(y: columns 11, 10; z: columns 01, 11; x: row 1)

Giriş			Çıkış
x	y	z	F4
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

x \ yz	00	01	11	10
0	1	0	0	1
1	1	0	0	1

(y: columns 11, 10; z: columns 01, 11)

F4 = y'z' + yz'
F4 = z'(y'+y) = z'

x \ yz	00	01	11	10
0	1	0	0	1
1	1	0	0	1

(y: columns 11, 10; z: columns 01, 11)

F4 = z'

3 Girişli KARNOUGH HARİTASI (K-MAP)

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

y

$x \backslash yz$	00	01	11	10
0	m_0 $x'y'z'$	m_1 $x'y'z$	m_3 $x'yz$	m_2 $x'yz'$
x { 1	m_4 $xy'z'$	m_5 $xy'z$	m_7 xyz	m_6 xyz'

z

Giriş			Çıkış
x	y	z	F5
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

y

	00	01	11	10
0	0	0	0	0
x { 1	1	0	0	1

z

$$\boxed{F5 = xz'}$$

4 Girişli KARNOUGH HARİTASI (K-MAP)

TOBB ETÜ
Ekonomi ve Teknoloji Üniversitesi

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

$wx \backslash yz$	00	01	11	10
00	m_0 $w'x'y'z'$	m_1 $w'x'y'z$	m_3 $w'x'yz$	m_2 $w'x'yz'$
01	m_4 $w'xy'z'$	m_5 $w'xy'z$	m_7 $w'xyz$	m_6 $w'xyz'$
11	m_{12} $wxy'z'$	m_{13} $wxy'z$	m_{15} $wxyz$	m_{14} $wxyz'$
10	m_8 $wx'y'z'$	m_9 $wx'y'z$	m_{11} $wx'yz$	m_{10} $wx'yz'$

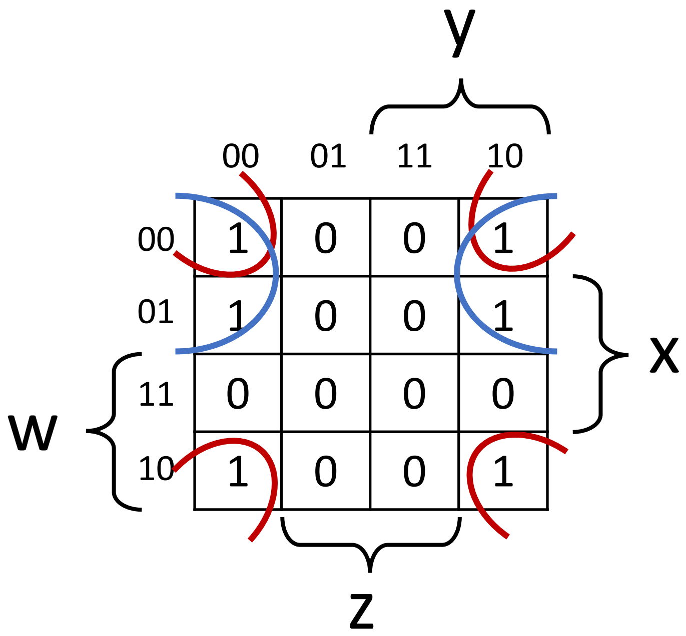
y x w z

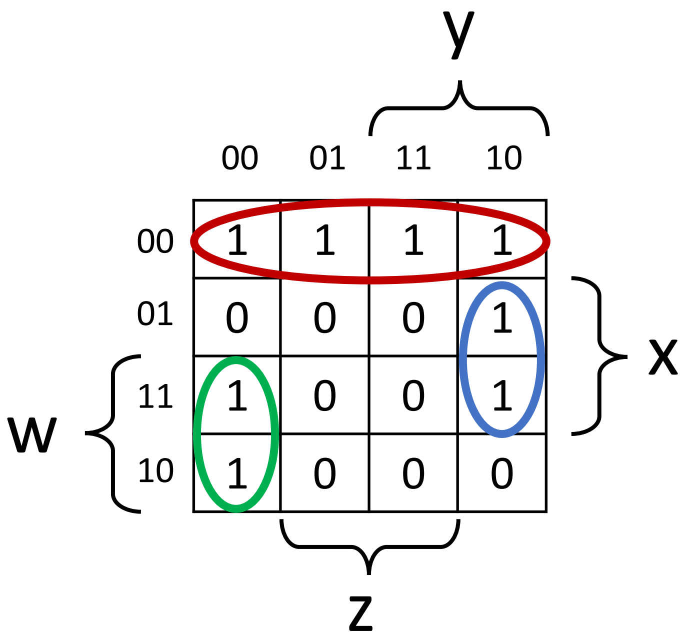
4 Girişli KARNOUGH HARİTASI (K-MAP)

$F1 = x'z' + z'w'$

Giriş				Çıkış
w	x	y	z	F1
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$F2 = w'x' + xyz' + wy'z'$





X – Don’t Care – Önemsenmeyen Durumlar

TOBB ETÜ Ekonomi ve Teknoloji Üniversitesi

$$F2 = w'x' + xyz' + wy'z'$$

	00	01	11	10
00	1	1	1	1
01	0	0	0	1
11	1	0	0	1
10	1	0	0	0

Giriş				Çıkış
w	x	y	z	F2
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	x
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

$$F2 = w'x' + xz' + y'z'$$

	00	01	11	10
00	1	1	1	1
01	X	0	0	1
11	1	0	0	1
10	1	0	0	0