

2. gaia:

Boole aljebraren
oinarriak eta ate logikoak

Algebra boolearra

- 1938an, Shannon-ek proposatu zuen zirkuitu digital diseinuari metodo matematiko bat aplikatzea, funtzio digitalak sortzeko
- Metodo hori, George Boole-k proposatu zuen XIX. mendean, Egia/Gezurra moduko proposaketa logikoak ikertzeko: Boole algebra

Funtzio logikoen adierazpena

- Metodo honen oinarriak bi balioko aldagai eta aldagai horren arteko hiru eragiketak dira
- Erlazioak, aldagai eta funtzioen arteko balio guztiak jasotzen dituzten taulen bidez definitzen dira → Egia-
taulak

AND		
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \bar{X}$
0	1
1	0

Boole aljebraren postulatuak

- Abiapuntua da egiaztatu beharrik ez dagoen zenbait baieztapen
→ Postulatuak
- Baieztapen hori (algebra honetako zenbaki eta eragiketaren definizioak direla) erabiliz, erlazio eta baieztapen berriak sortu ditzakegu

$$X \neq 1 \text{ denean } X = 0$$

$$X \neq 0 \text{ denean } X = 1$$

$$0 + 0 = 0$$

$$1 + 1 = 1$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 \cdot 1 = 1$$

$$0 \cdot 0 = 0$$

$$1 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$X = 1 \text{ denean } \bar{X} = 0$$

$$X = 0 \text{ denean } \bar{X} = 1$$

Boole aljebraren oinarritzko identitateak

1.	$X + 0 = X$	2.	$X \cdot 1 = X$	
3.	$X + 1 = 1$	4.	$X \cdot 0 = 0$	
5.	$X + X = X$	6.	$X \cdot X = X$	
7.	$X + \overline{X} = 1$	8.	$X \cdot \overline{X} = 0$	
9.	$\overline{\overline{X}} = X$			

10.	$X + Y = Y + X$	11.	$XY = YX$	Trukakorra
12.	$X + (Y + Z) = (X + Y) + Z$	13.	$X(YZ) = (XY)Z$	Elkartzea
14.	$X(Y + Z) = XY + XZ$	15.	$X + YZ = (X + Y)(X + Z)$	Banatzea
16.	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	17.	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	De Morgan

$$(X + Y) \cdot \left(X + \overline{Y} \right) = X \quad \text{Konbinazioa} \quad X \cdot Y + X \cdot \overline{Y} = X$$

Erlazio hauek egiazkoak dira, postulatuak betetzen badituzte

Boole aljebraren oinarrizko identitateak

- Identitateak frogatzeko, berdintasuneko bi aldeko funtzioaren egia taulak osatuko ditugu, eta berdinak direla egiaztatu
- Banatze (biderketa batuketan) propietatearen froga hemen daukazue:

X	Y	Z	$Y+Z$	$X \cdot (Y+Z)$	$X \cdot Y$	$X \cdot Z$	$X \cdot Y + X \cdot Z$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Boole aljebraren oinarrizko identitateak

- Oinarrizko eragiketa bat ez bada ere, badago definituta beste eragiketa bat: batuketa esklusiboa (XOR)

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

- Emaitza 1 da, eragigaiak ezberdinak badira

Boole aljebraren oinarrizko identitateak

- Batuketa eskusiboak ere bai elkartze propietatea dauka
- Horrela, hiru edo aldagai gehiago daukan funtzioa da, non irteera 1 da bere sarreran leko kopurua bakoitia denean → Funtzio bakoitia

X	Y	Z	$Y \oplus Z$	$X \oplus (Y \oplus Z)$	$X \oplus Y$	$(X \oplus Y) \oplus Z$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	1
0	1	1	0	0	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	0	0	0
1	1	1	0	1	0	1

Funtzio baten adierazpen kanonikoa

- Identitate eta postulatu bidez, aldagai logikoen arteko erlazio berriak asmatu ditzakegu
- Funtzio logiko baten egia tauletik atera dezakegu adierazpen aljebraikoa → Adierazpen kanonikoa
- Adierazpen kanonikoak funtzioaren aldagai guztiak ditu bere gai guztietan
- Adierazpen kanonikoaren gaiak, bidergaiak (minterminoak) edo batugaiak (maxterminoak) izan daitezke

Funtzio baten adierazpen kanonikoa

X	Y	Z	Biderkadura gaia	Ikurra	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7
0	0	0	$\overline{X}\overline{Y}\overline{Z}$	m_0	1	0	0	0	0	0	0	0
0	0	1	$\overline{X}\overline{Y}Z$	m_1	0	1	0	0	0	0	0	0
0	1	0	$\overline{X}Y\overline{Z}$	m_2	0	0	1	0	0	0	0	0
0	1	1	$\overline{X}YZ$	m_3	0	0	0	1	0	0	0	0
1	0	0	$X\overline{Y}\overline{Z}$	m_4	0	0	0	0	1	0	0	0
1	0	1	$X\overline{Y}Z$	m_5	0	0	0	0	0	1	0	0
1	1	0	$XY\overline{Z}$	m_6	0	0	0	0	0	0	1	0
1	1	1	XYZ	m_7	0	0	0	0	0	0	0	1

- Minterm-a biderketa bat da, emaitza 1 balio duena aldagaiko konbinazio bakar baterako eta besterako 0 ematen duena
- Minterm-ko adierazpenean, 1 ematen duen aldagaia 1 balioa badauka, ezeztatu gabe agertuko da, eta 0 baliokoa bada, ezeztatua

Funtzio baten adierazpen kanonikoa

X	Y	Z	Batuketa gaia	Ikurra	M ₀	M ₁	M ₂	M ₃	M ₄	M ₅	M ₆	M ₇
0	0	0	$X + Y + Z$	M ₀	0	1	1	1	1	1	1	1
0	0	1	$X + Y + \overline{Z}$	M ₁	1	0	1	1	1	1	1	1
0	1	0	$X + \overline{Y} + Z$	M ₂	1	1	0	1	1	1	1	1
0	1	1	$X + \overline{Y} + \overline{Z}$	M ₃	1	1	1	0	1	1	1	1
1	0	0	$\overline{X} + Y + Z$	M ₄	1	1	1	1	0	1	1	1
1	0	1	$\overline{X} + Y + \overline{Z}$	M ₅	1	1	1	1	1	0	1	1
1	1	0	$\overline{X} + \overline{Y} + Z$	M ₆	1	1	1	1	1	1	0	1
1	1	1	$\overline{X} + \overline{Y} + \overline{Z}$	M ₇	1	1	1	1	1	1	1	0

- Maxterm-a batuketa bat da, emaitza 0 balio duena aldagaiko konbinazio bakar baterako eta besterako 1 ematen duena
- Maxterm-eko adierazpenan, 0 ematen duen aldagaia 1 balioa badauka, ezeztatua agertuko da, eta 0 baliokoa bada, ezeztatu gabe
- Maxterm-ak, bere minterm-eko ezeztapenak dira $\rightarrow M_i = m'_i$

Funtzio baten adierazpen kanonikoa

- Funtzioaren balioa 1 egiten dituzten aldagai balioen konbinazioei dagozkien **minterm-eko batuketa** da adierazpen kanonikoa
- Funtzioaren balioa 0 egiten dituzten aldagai balioen konbinazioei dagozkien **maxterm-eko biderketa** ere bai da adierazpen kanonikoa
- Funtzio bakoitzari bi adierazpen kanoniko dagokio → **Minterm-eko batuketa/Maxterm-eko biderketa**
- Bi adierazpen kanonikotik edozein funtzio betetzen dituzten zirkuitu digitalak atera daitezke, behin bere egia-taula ezagututa (baina oso luzeak izan daitezke...)

Funtzio baten adierazpen kanonikoa

X	Y	Z	F	\bar{F}
0	0	0	1	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

De Morgan teoremaren bidez ikus daiteke bi adierazpenak berdinak direla

$$F = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}Z + XYZ$$

$$= m_0 + m_2 + m_5 + m_7$$

$$\bar{F} = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}\bar{Z} + XY\bar{Z}$$

$$= m_1 + m_3 + m_4 + m_6$$

$$F = \overline{m_1 + m_3 + m_4 + m_6} = \bar{m}_1 \cdot \bar{m}_3 \cdot \bar{m}_4 \cdot \bar{m}_6$$

$$M_i = \bar{m}_i$$

$$F = M_1 \cdot M_3 \cdot M_4 \cdot M_6$$

Funtzio logikoen sinplifikazioa

- Adierazpen kanonikoen bidez edozein funtzio logikoen adierazpen aljebraikoa lortu dezakegu
- Adierazpen hauek oso luzeak izan daitezke
- Aljebra boolearraren bidez, gai gutxiago daukan eta, beraz, gauzatzeko errezagoa den, adierazpen baten bihur daiteke adierazpen kanonikoa ➔ Sinplifikazioa
- Karnaugh mapen metodoaren bidez, funtzioaren sinplifikazioa sistematiko bihurtuko dugu

Karnaugh mapen bidezko sinplifikazioa

- Karnaugh mapa, sarrera bikoitzeko taula bat gisa egia-
taularen adierazpen bat da
- Errenkada eta zutaberen gurutzagune (mapen laukia)
bakoitza, funtzioaren balio bat da
- Lauki bakoitzari, minterm bat dagokio, funtzioaren minterm-
eko laukietan, 1 idazten da

m_0	m_1
m_2	m_3

		y	
		0	1
x	0	$x'y'$	$x'y$
	1	xy'	xy

Karnaugh mapen bidezko sinplifikazioa

- Funtzioaren aldagai kopuruaren arabera, Karnaugh mapak 4 lauki (bi aldagai), 8 lauki (hiru aldagai), 16 lauki (lau aldagai),... daukate
- Lauki auzokideei dagozkien minterm-ak, gai bakarren bildu daitezke (konbinazioa)
- Gai (inplikatzailea) horren adierazpenean bakarrik agertzen dira lauki guztientzat balio amankomuneko aldagaiak

m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6

		y					
		yz		00	01	11	10
x	0	$x'y'z'$	$x'y'z$	$x'yz$	$x'yz'$		
x	1	$xy'z'$	$xy'z$	xyz	xyz'		
		z					

Karnaugh mapen bidezko sinplifikazioa

- m_1, m_3, m_5 eta m_7 implikatzaile bakarrean biltzen dira: C
- m_3 eta m_2 $A' \cdot B$ implikatzailean biltzen dira
- *Bi implikatzaile honen batuketa, funtzioko minterm guztiak ditu barne, beraz adierazpen kanonikoaren baliokidea da $\Rightarrow F = C + A' \cdot B$*

		BC		B	
A		0 0	0 1	1 1	1 0
A	0		1	1	1
	1		1	1	
				C	

$$F = m_1 + m_2 + m_3 + m_5 + m_7$$

$$F = C + A' \cdot B$$

Karnaugh mapen bidezko sinplifikazioa

- Gai gutxien eta gai bakoitzean aldagai kopuru txikien, baina funtzio baten minterm guztiak barne dauzkan batuketa, **funtzio baten batuketa minimoa** da
- Lauki kopuru (2ko berredurak) handien daukan inplikatzailerak, guztiak funtzioaren mintermak direnak, **inplikatzailerak lehenak** da

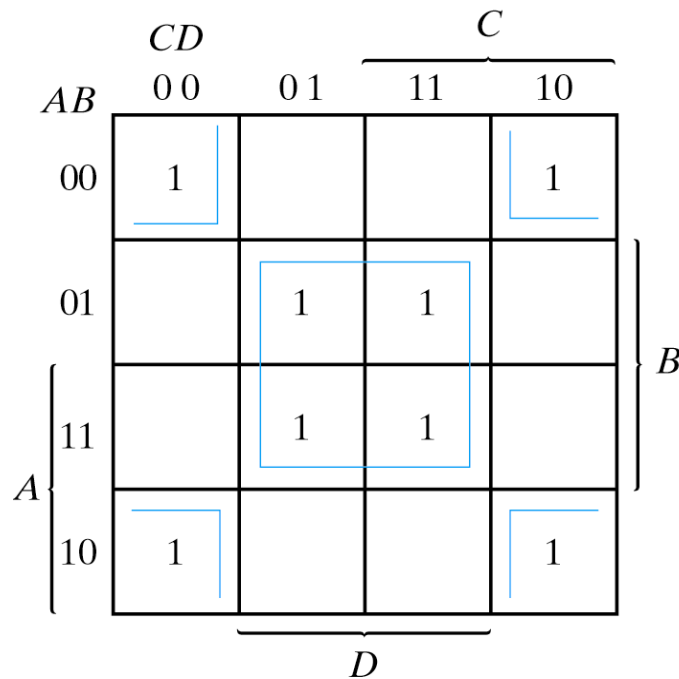
m_0	m_1	m_3	m_2
m_4	m_5	m_7	m_6
m_{12}	m_{13}	m_{15}	m_{14}
m_8	m_9	m_{11}	m_{10}

		y			
		yz		11	10
wx	00	$w'x'y'z'$	$w'x'y'z$	$w'x'yz$	$w'x'yz'$
	01	$w'xy'z'$	$w'xy'z$	$w'xyz$	$w'xyz'$
	11	$wxy'z'$	$wxy'z$	$wxyz$	$wxyz'$
	10	$wx'y'z'$	$wx'y'z$	$wx'yz$	$wx'yz'$

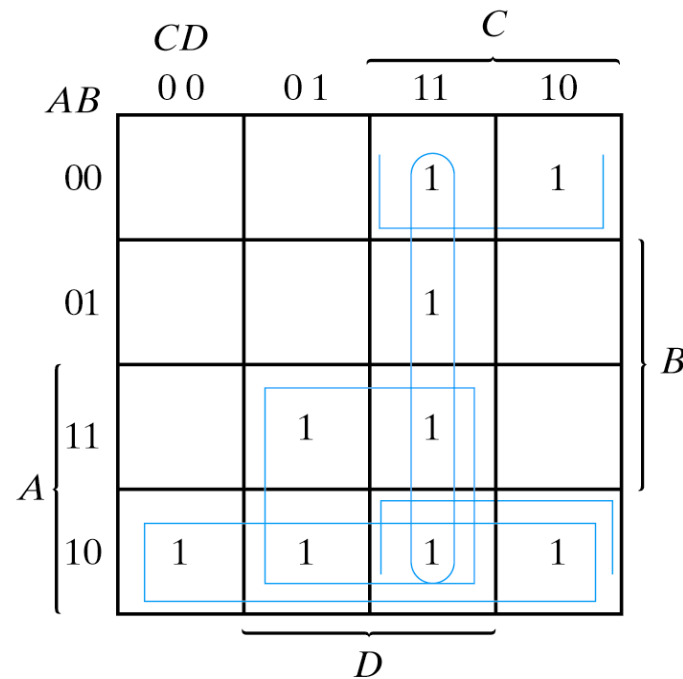
z

Karnaugh mapen bidezko sinplifikazioa

- Implikatzaile lehen bakar bati dagozkion laukiak, **lauki bereziak** dira
- Lauki bereziak dauzkaten implikatzaile lehenak, **funtsezko lehen implikatzaileak** dira
- Funtzio baten **batuketa minimoa**, funtzioaren implikatzaile leheneko batuketa bat da, eta barnean beti **funtsezko implikatzaile lehen guztiak**



Funtsezko lehen implikatzaileak:
 BD y $B'D'$



Funtsezko lehen implikatzaileak:
 CD , $B'C$, AD y AB'

Karnaugh mapen bidezko sinplifikazioa: zeroak erabiliz

- Hutsik diren laukiak (funtzioaren zeroak), maxterm-ak irudikatzen dituzte
- Zero lauki auzokideei dagozkien maxterm-ak, gai bakarren bildu daitezke (konbinazioa), inplikatzaile hori batuketa da
- Inplikatzaile horren adierazpenean bakarrik agertzen dira lauki guztientzat balio amankomuneko aldagaiak
- Zero inplikatzaileraren biderketa, funtzioaren maxterm guztiak daukana, adierazpen kanonikoaren baliokidea da
- Gai gutxien eta gai bakoitzean aldagai kopuru txikien, baina funtzio baten maxterm guztiak barne dauzkan biderketa, **funtzio baten biderketa minimoa** da

Karnaugh mapen bidezko sinplifikazioa: zeroak erabiliz

- M_4, M_6, M_{12} y M_{14} biltzen dira inplikatzaile bakar baten: $B'+D$
- Beste biak $A'+B'$ eta $C'+D'$ dira
- Inplikatzaile guztiak funtsezkoak direnez, biderketa minimoa da:
 $F=(B'+D) \cdot (A'+B') \cdot (C'+D')$

		CD		C		
		00	01	11	10	
AB	00	1	1	0	1	B
	01	0	1	0	0	
	11	0	0	0	0	
	10	1	1	0	1	
A		D				

F funtzioaren mapa: $F(A, B, C, D)=\Sigma (0, 1, 2, 5, 8, 9, 10)$

Karnaugh mapen bidezko sinplifikazioa

- Adierazpen kanonikoaren lekuan, adierazpen minimoa (**batuketa edo biderketa, bien arteko sinpleena**) erabiliz, sinplifikazioa lortzen da
- Funtzioaren aldagai kopurua lau baino handiagoa denean, sinplifikazioa egiteko ez dira erabiltzen Karnaugh mapak, konputagailuren bidezko metodo iteratiboak baizik

Zehazpen osagabeko funtzioaren sinplifikazioa

- Sarrerako aldagai konbinazio guztietarako, **funtzioaren balioa definituta ez denean, zehazpen osagabeko funtzioak** dira
- Sarrerako aldagaien zenbait konbinazio ezinezkoak direnean, ez da definitzen horretarako funtzio balioa, **X-ren bidez** aipatzen da hori
- Gai honi “berdin zaigu” deitzen da eta horrela adierazten da: d eta funtzio zehaztuta ez den sarrera konbinazio zenbakiaren azpiindizea
- Kasu honetan, sinplifikazioa egiteko **aukeratuko dugu inplikatzailerik handien lortzen den balioa: 0 edo 1**

Zehazpen osagabeko funtzioaren sinplifikazioa

$$F = \sum m(1, 3, 7, 11, 15) + \sum d(0, 2, 5)$$

		y			
		yz			
		00	01	11	10
w	x	X	1	1	X
	01	0	X	1	0
	11	0	0	1	0
	10	0	0	1	0

z

- 0, 2 eta 5 funtzioaren zeroak badira, inplikatzaile lehenaren barruan bakarrik daude 1 eta 3: $w' \cdot x' \cdot z$
- 0 eta 2 hartu dezakegu 1 bezala, inplikatzaile lehen berria osatzeko: $w' \cdot x'$
- 5 bada 0, a) batuketa minimoa da

a) $F = y \cdot z + w' \cdot x'$

Zehazpen osagabeko funtzioaren sinplifikazioa

$$F=\sum m(1, 3, 7, 11, 15)+\sum d(0, 2, 5)$$

		yz		y	
		00	01	11	10
w	00	X	1	1	X
	01	0	X	1	0
	11	0	0	1	0
	10	0	0	1	0
		z			

- 0 eta 2 zero eta 5 bat hartuta, inplikatzaile lehena da: $w' \cdot z$
- b) batuketa minimoa da orain
- Funtzio hau eta a) **ezberdinak dira, baina F osagabeko definizioarekin bat datoz biak**

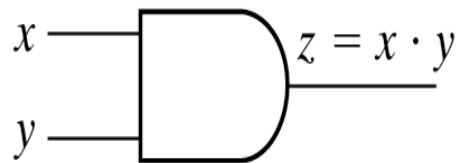
b) $F = y \cdot z + w' \cdot z$

Ate logikoak

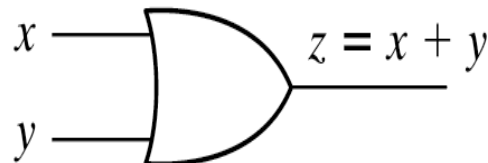
- Aljebra boolearraren bidez, funtzioak diseinatu ditzakegu
- Diseinu digitalean funtzio horiek erabiltzeko, eragiketa boolearrak betetzen dituzten zirkuituak behar ditugu
- Ate logikoak dira zirkuitu horiek

Ate logikoak

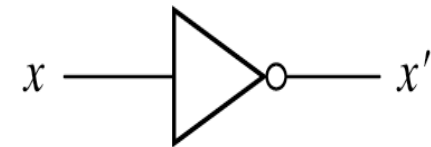
- Oinarrizko ate logikoei, aljebra boolearraren hiru eragiketak dagozkie



Funtzio logikoa ETA (AND)



Funtzio logikoa EDO (OR)



Funtzio logikoa EZ (NOT) †

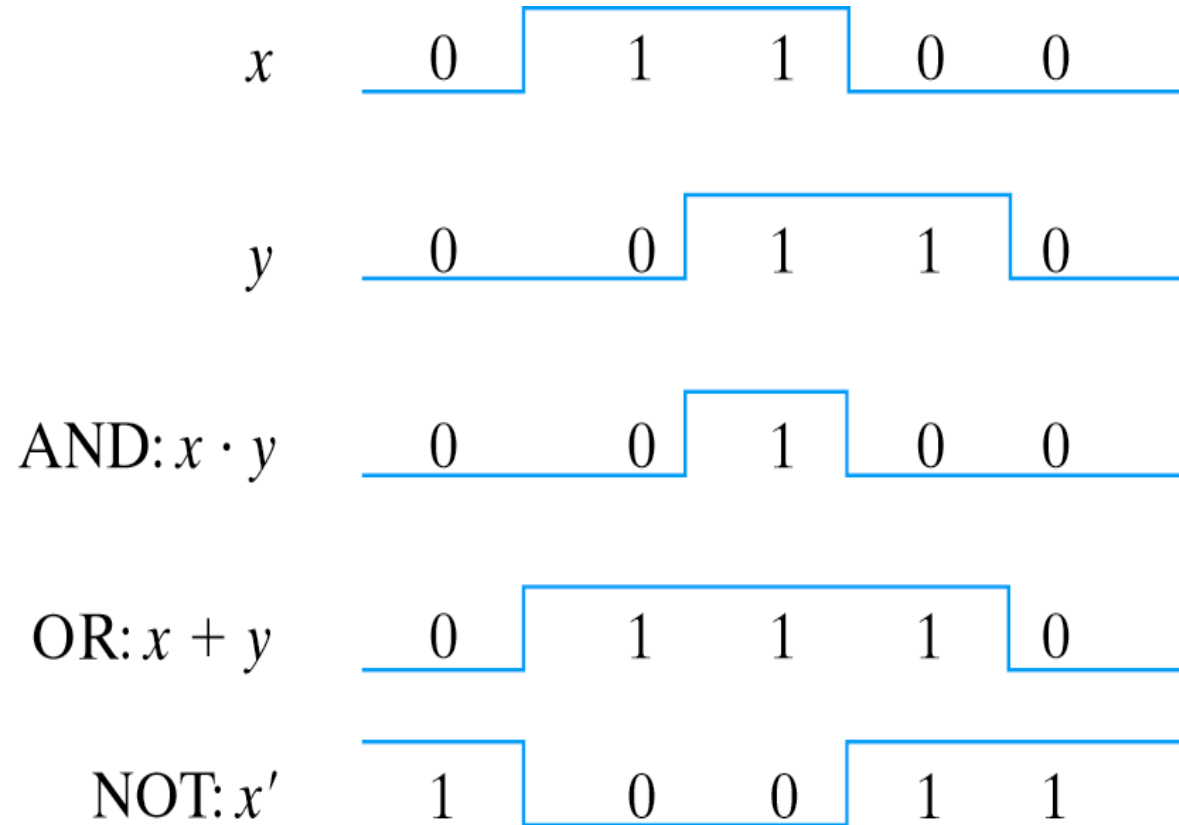
AND		
X	Y	$Z = X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	$Z = X + Y$
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \bar{X}$
0	1
1	0

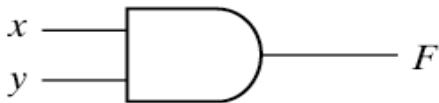

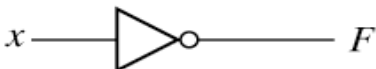
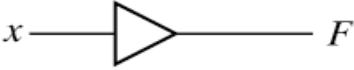
Ate logikoak

- Aldagai boolearraren lekuan, tentsio seinaleak jarriko ditugu
- 0-ri tentsio baxua (L) dagokio, eta tentsio altua (H) 1-i



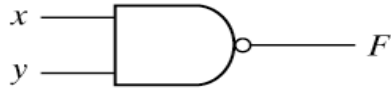



Zirkuitu digitalen sarrera eta irteerako seinaleak

Ate logikoak

Izena	Ikurra	Adierazpen algebraikoa	Egia-taula															
AND		$F = xy$	<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																
Inbertsorea (NOT)		$F = x'$	<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																	

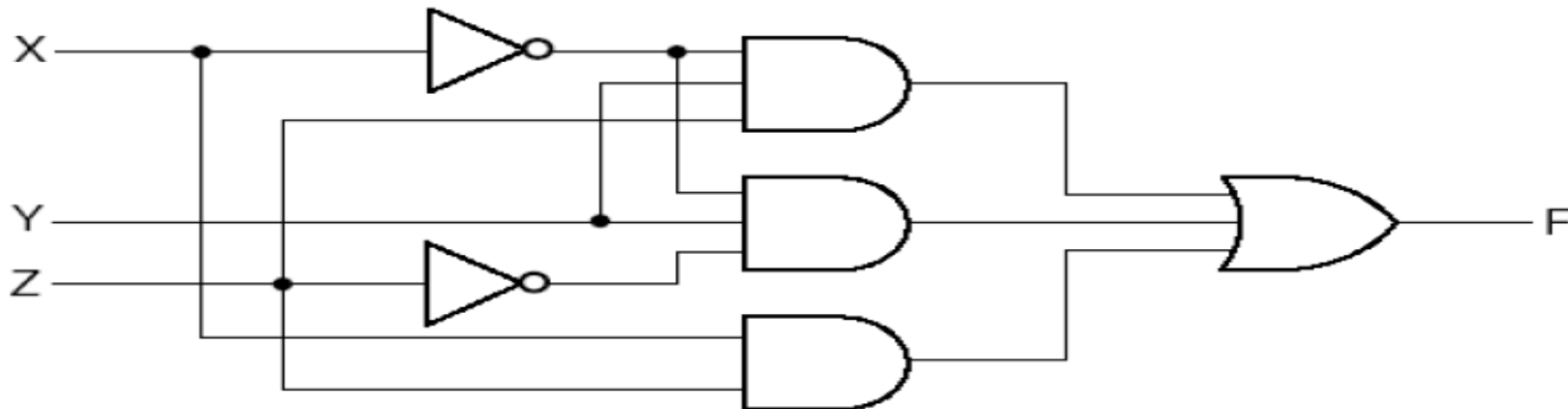
Tentsio eta korrante balioak mantentzeko erabiltzen da buffer-a

Ate logikoak

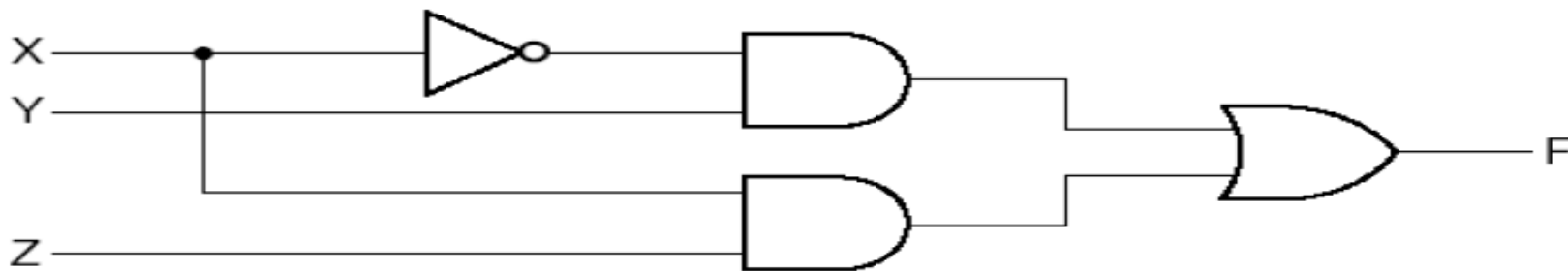
Izena	Ikurra	Adierazpen algebraikoa	Egia-taula															
NAND		$F = (xy)'$	<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	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																
OR eskusiboa (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																
NOR eskusiboa (XNOR)		$F = xy + x'y'$ $= (x \oplus y)'$	<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																

Erabilgarritasunagatik edo ekoizpen erraztasunagatik, beste ate logikoak diseinatu dituzte, nahiz eta ez dira Boole aljebraiko eragiketak

Zirkuitu logikoen sintesia



(a) $F = \bar{X}YZ + \bar{X}Y\bar{Z} + XZ$

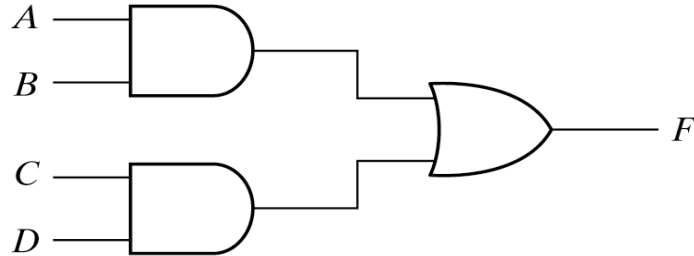


(b) $F = \bar{X}Y + XZ$

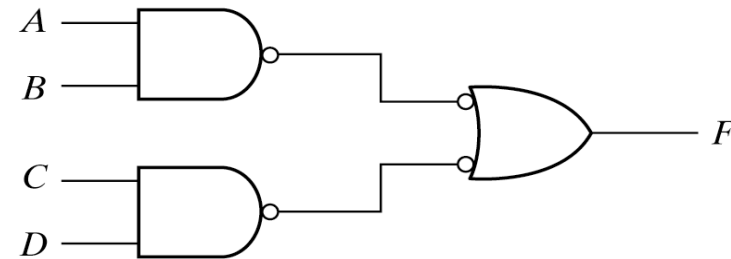
- Boole aljebra adierazpen guztiak zirkuituan bihurtu daitezke, eragiketa bakoitzaren lekuan dagokion ate logikoa kokatuz
- Gero eta adierazpen sinpleago, zirkuituaren osagarrien kopurua txikiago → Zirkuitua azkarragoa eta ekonomikoagoa

2 mailetako zirkuituak

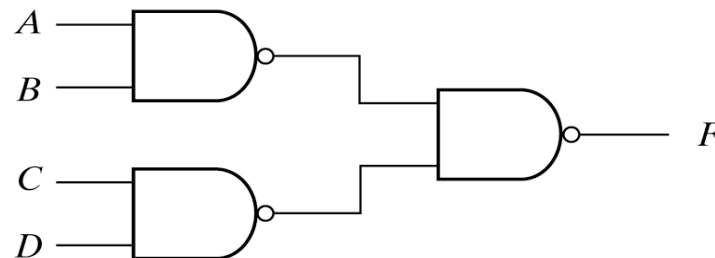
- Adierazpen aljebraikoa biderketako batuketa edo batuketako biderketa bada, zirkuituaren sarrera eta irteeraren artean bi ate logiko baino ez dago → 2 mailetako zirkuituak
- Kasu honetan, zirkuitu osoa gauzatu daiteke bakarrik NAND ate edo bakarrik NOR ate erabiliz



(a)



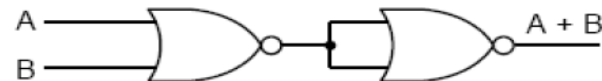
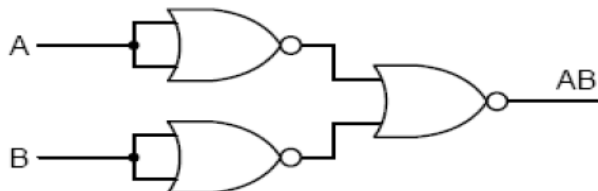
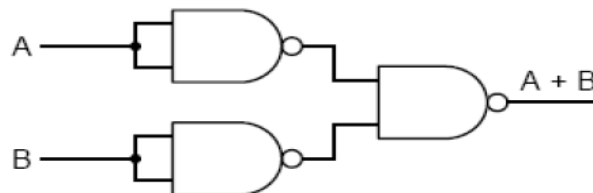
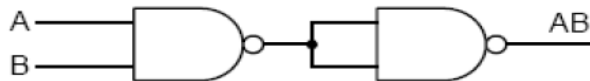
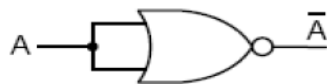
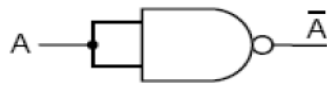
(b)



(c)

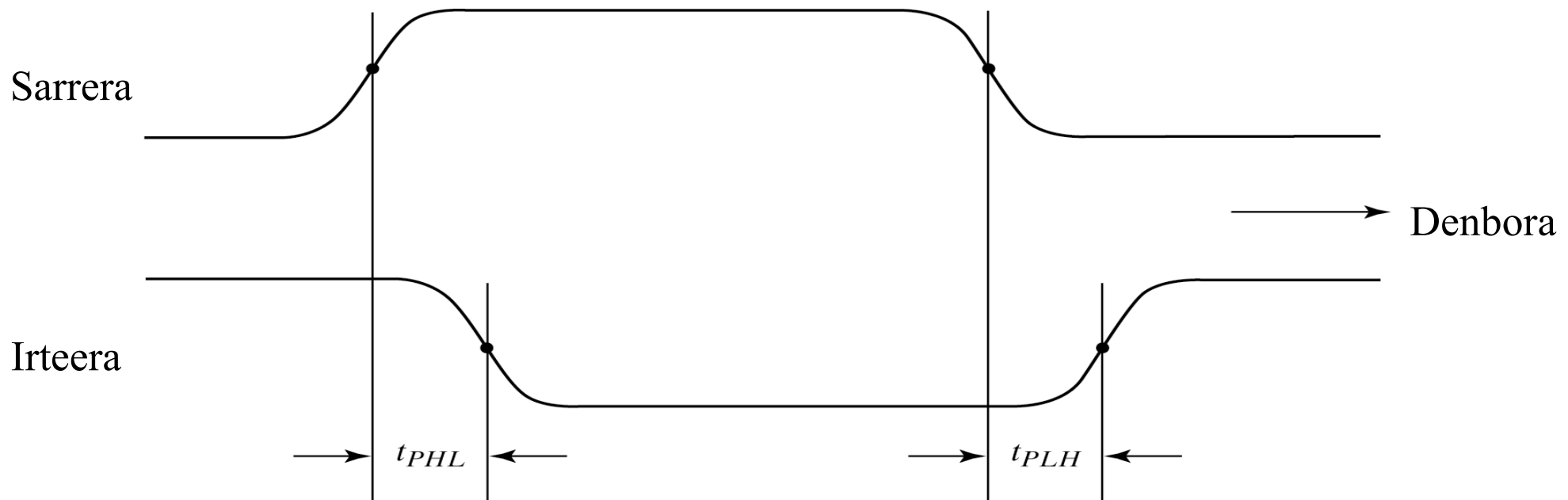
2 mailetako zirkuituak

- Aurrekoa ere bai aplikatu daiteke batuketako biderketa batera, NOR funtzioak sortzeko
- NAND eta NOR-en bidez, Boole aljebraکو eragiketa guztiak gauzatu daitezke



Atzerapen denbora

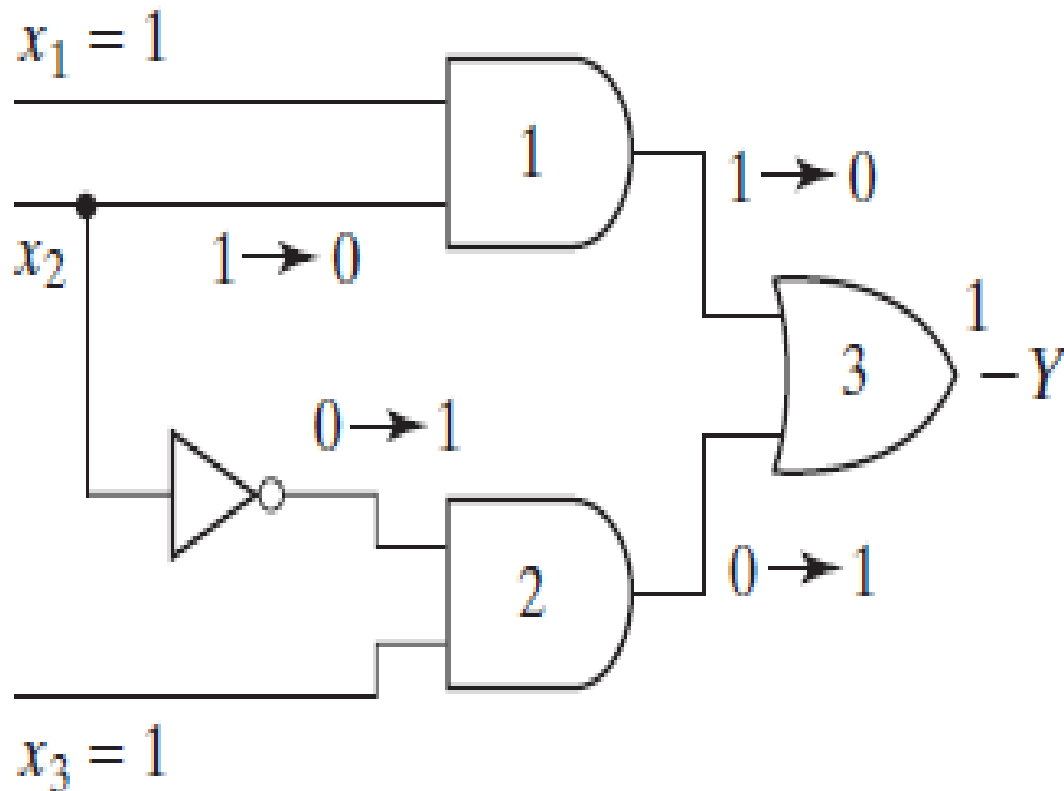
- Benetako zirkuituetan, sarrera aldatzen denetik denbora-tarte bat pasatuko da irteera aldatu arte
- Sarrera aldatu denetik irteera aldatu arte pasatutako denbora, atzerapen denbora t_p deitzen da



Atzerapen denbora: Arriskuak

- Zirkutu baten irteera, besteen sarrera denean, bi zirkuituen atzerapen denborak batutzen dira
- Zenbait zirkuituko irteeretan balio okerrak agertu daitezke denbora-tarte txikien bitartean
- Halako balio iragankorrak **arriskuak** deitzen dira, eta sarreretarik irteera heltzeko ate kopuru ezberdineko bideak daudelako gertatzen dira → bide ezberdina denean, atzerapen denbora ezberdina izango da

Atzerapen denbora: Arriskuak

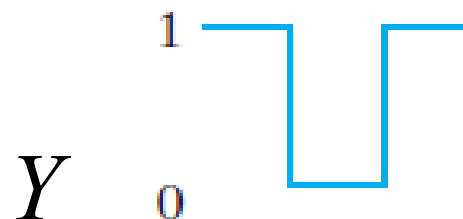


- x_2 1-tik 0-era aldatzen denean, OR ateko bi sarrerak atzerapen ezberdinak dituzte
- Denbora-tarte baten bitartean bi seinaleak dira $0 \rightarrow Y = 0$

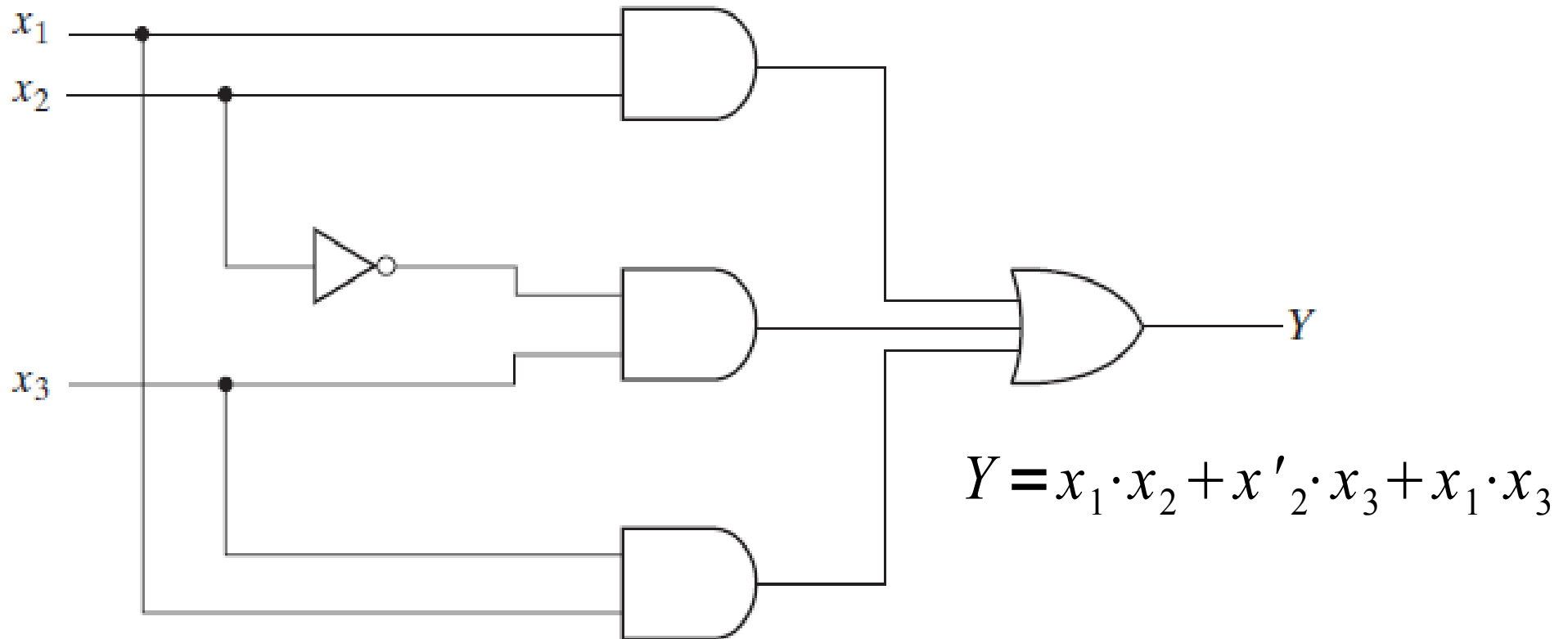
$$Y = x_1 \cdot x_2 + x'_2 \cdot x_3$$

$$x_1 x_2 x_3 = 111 \Rightarrow Y = 1$$

$$x_1 x_2 x_3 = 101 \Rightarrow Y = 1$$



Atzerapen denbora: Arriskuak



- Arazoa konpontzeko batuketan beste gai bat sartzen da
- Gai berriaren balioa 1 da x_2 aldatzen den bitartean $\rightarrow Y = 1$ denbora-tarte osoan

Hardware adierazpeneko lengoaia: VHDL

- Badago beste metodo bat zirkuitu digitalak irudikatzeko: hardware adierazpeneko lengoaia
- Testu bat da, non zenbait funtzio logikoak definitzen direla
- Testu fitxategia definitutako funtzioak betetzen dituen zirkuitu integratu baten programazio edo ekoizpenerako erabiltzen da
- Egun, hardware adierazpeneko lengoai erabiliena VHDL da (Verilog Hardware Description Language)

Hardware adierazpeneko lengoaia: VHDL

- VHDL testuko egitura bi atal du: entitatea eta arkitektura
- Entitatean bakarrik definitzen dira erabiliko ditugun aldagaiak, bere mota eta sarrera edo irteerakoak badira
- Arkitekturan entitatean definitutako aldagaien arteko funtzioak aipatzen dira

```
ENTITY example1 IS
PORT (   x1, x2, x3  : IN    BIT;
        f             : OUT BIT );
END example1 ;

ARCHITECTURE LogicFunc OF example1 IS
BEGIN
f <= (x1 AND x2) OR (NOT x2 AND x3) ;
END LogicFunc ;
```

