Combinational Circuits

Carmi Merimovich

The Academic College of Tel-Aviv

January 29, 2025

The Theme of this Chapter

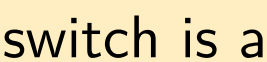
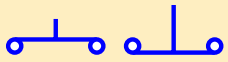
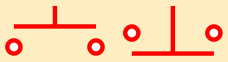
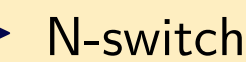
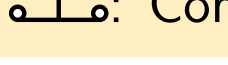
Boolean functions (Platonic world)

$$\Longleftrightarrow$$

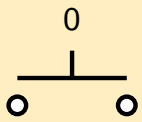
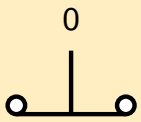
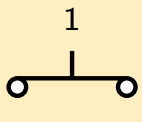
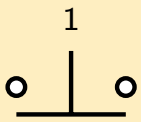
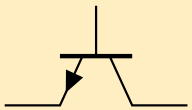
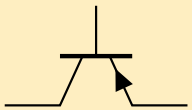
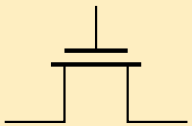
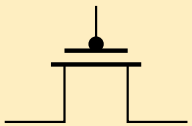
Combinational circuits (Real hardware)

Realizing the Boolean Operators

Switches

- A switch is a component having three tips 
  - One conrol pin
  - Two data pins
- A switch can be in one of two states
  - Unpressed  : The control is at logic 0
  - Pressed  : The control is at logic 1
- The data pins can be in one of two states
  - Connected 
  - Disconnected 
- Naming and drawing convention:
  - N-switch  : Connected $\iff$ Pressed
  - P-switch  : Connected $\iff$ Unpressed

Switch Drawings

Type	N	P
Unpressed switch		
Pressed switch		
Bipoloar transistor		
FET transistor		

Used in drawings

Switches $\equiv$ Transistors

Fame

The following implementation of logic gates is an abstraction of the CMOS technology. The CMOS technology was invented at Fairchild Semiconductor by Frank Wanlass and Chih-Tang Sah.

CMOS

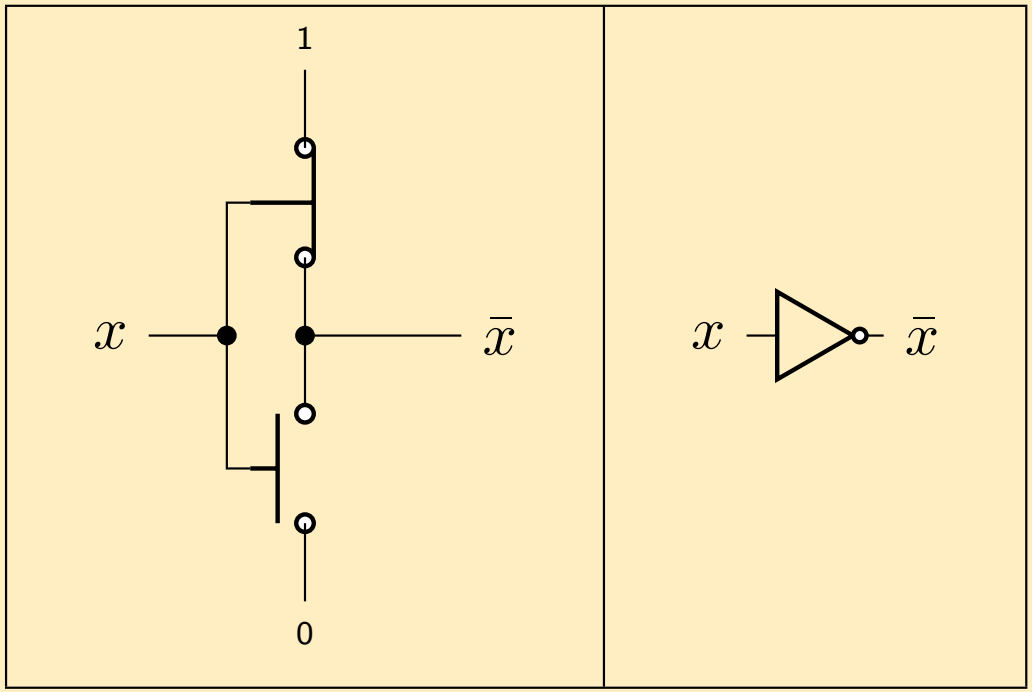
Complementary Metal-Oxide-Semiconductor

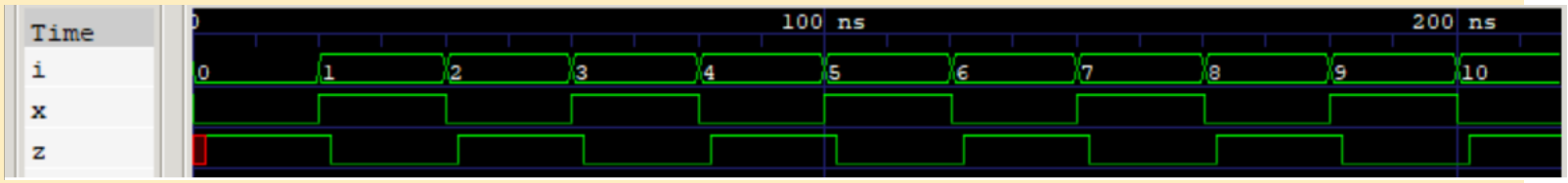
Logic Gates

Realizing the Operators $\cdot$, $+$ and $^-$

Later on we will see that there are more gates

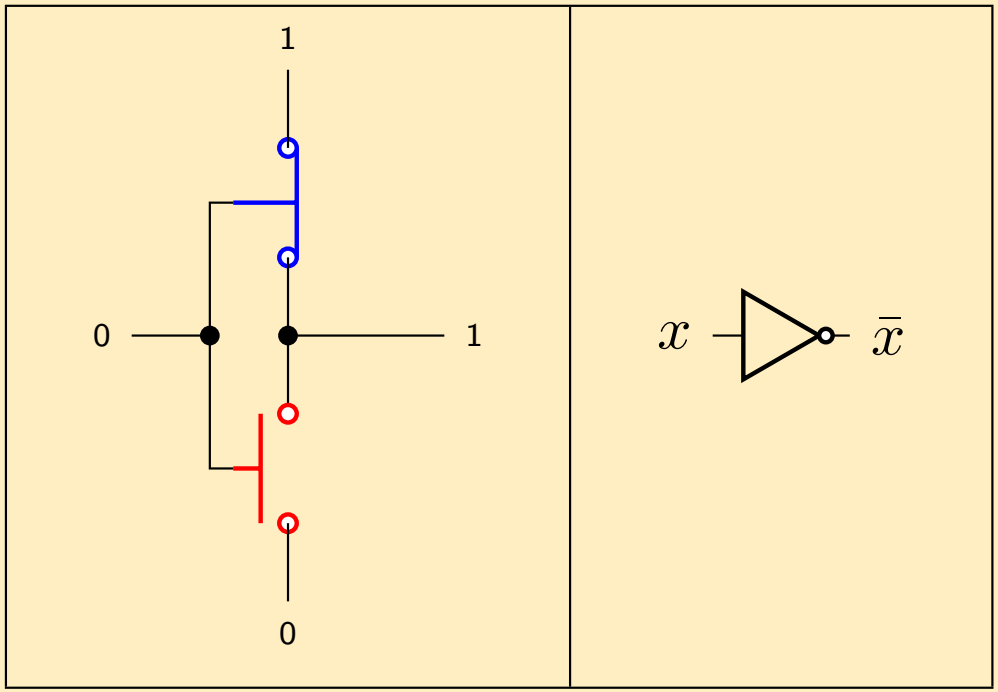
Not Gate

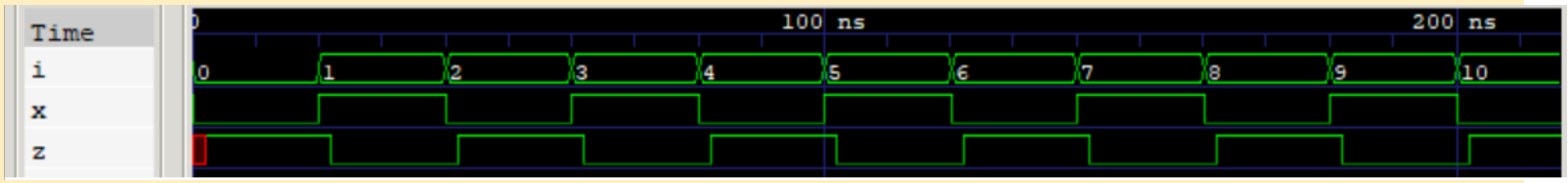




$$z = \bar{x}$$

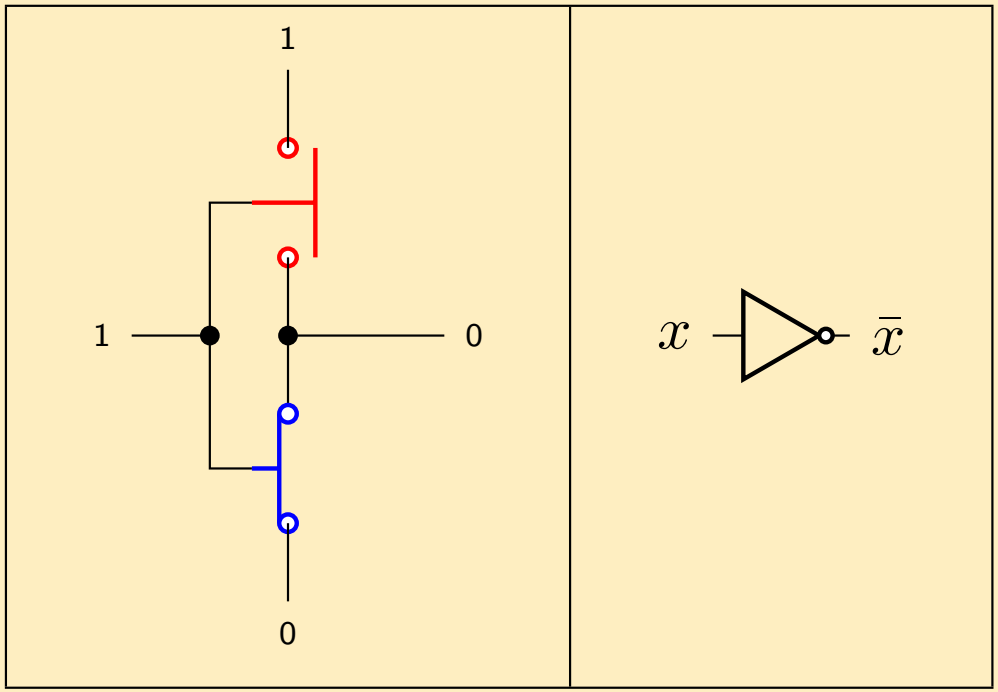
Not Gate $x = 0$

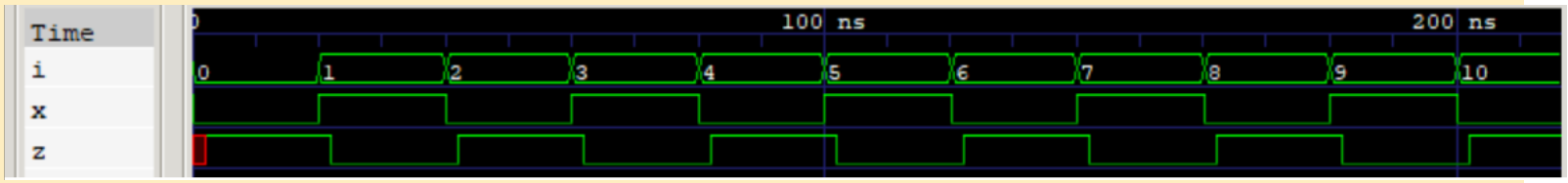




$$z = \bar{x}$$

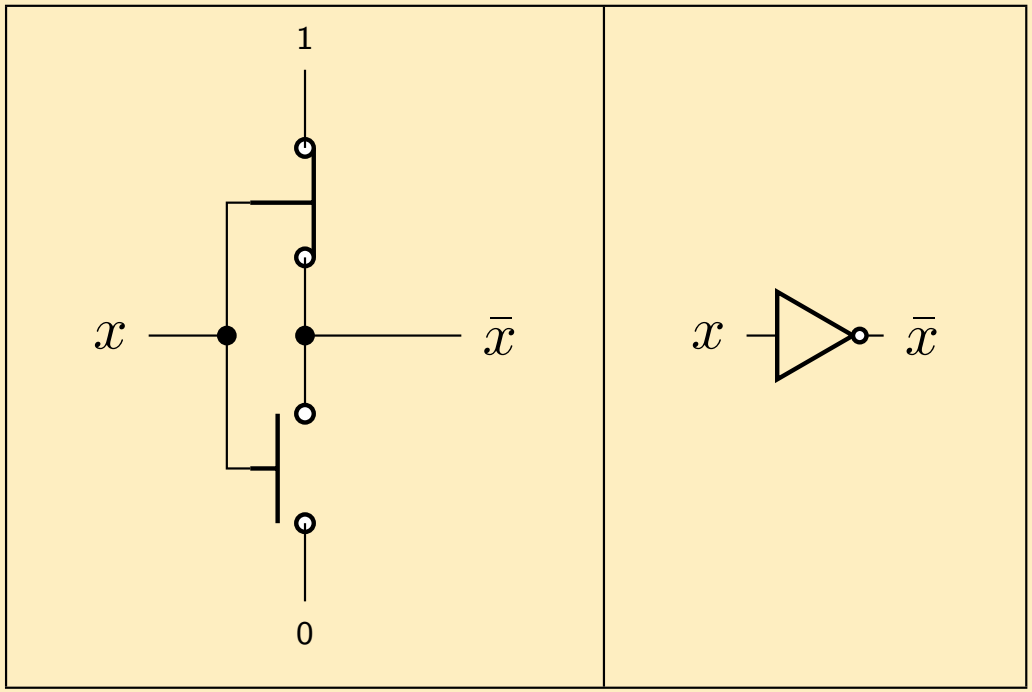
Not Gate $x = 1$

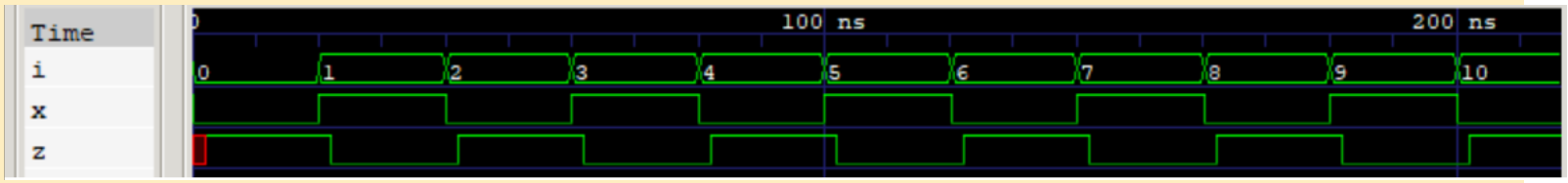




$$z = \bar{x}$$

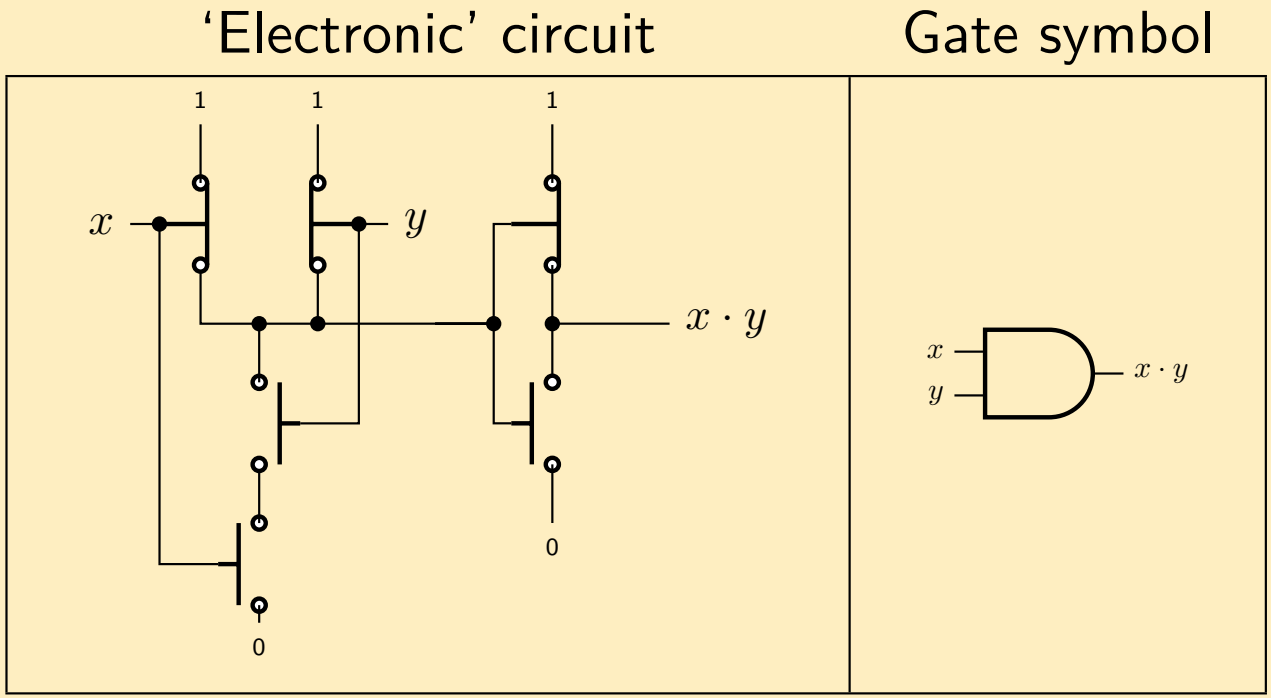
Not Gate

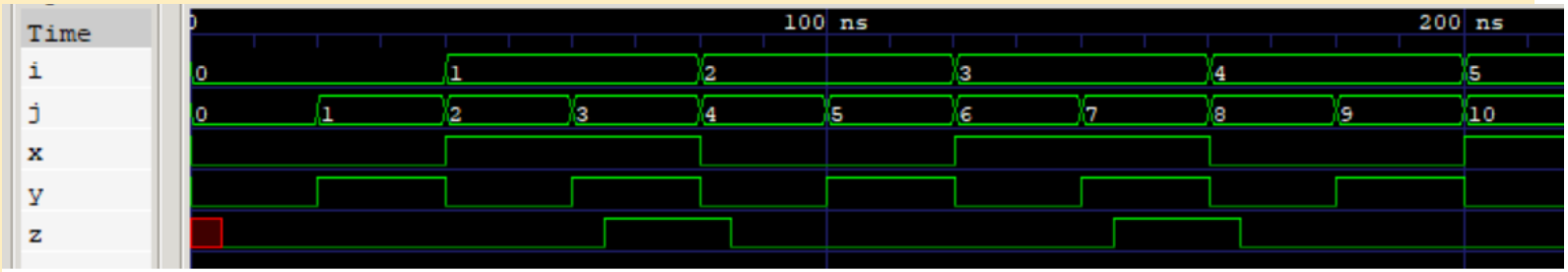




$$z = \bar{x}$$

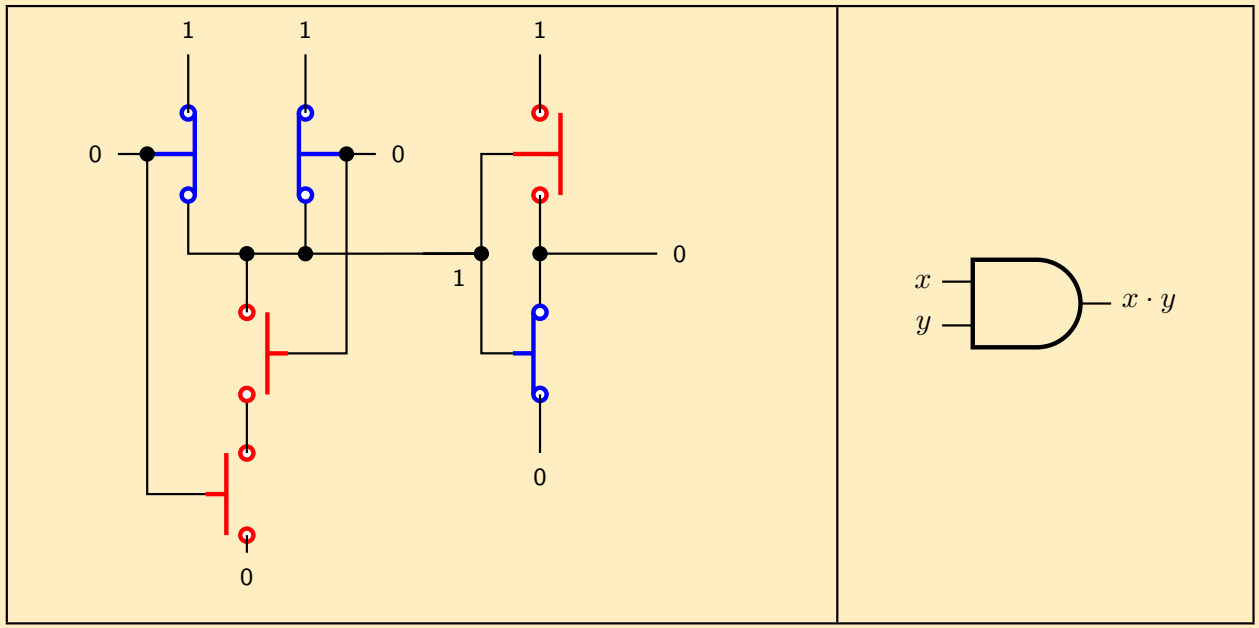
And Gate

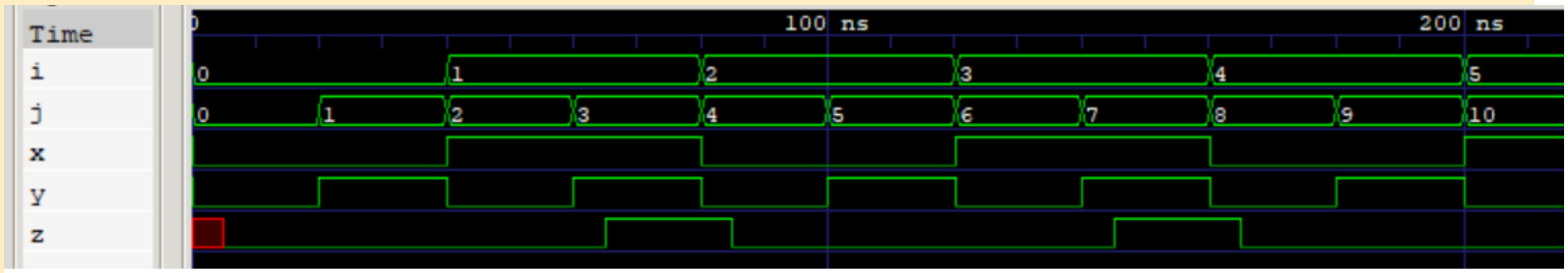




$$z = x \cdot y$$

And Gate $x = 0,\ \ y = 0$

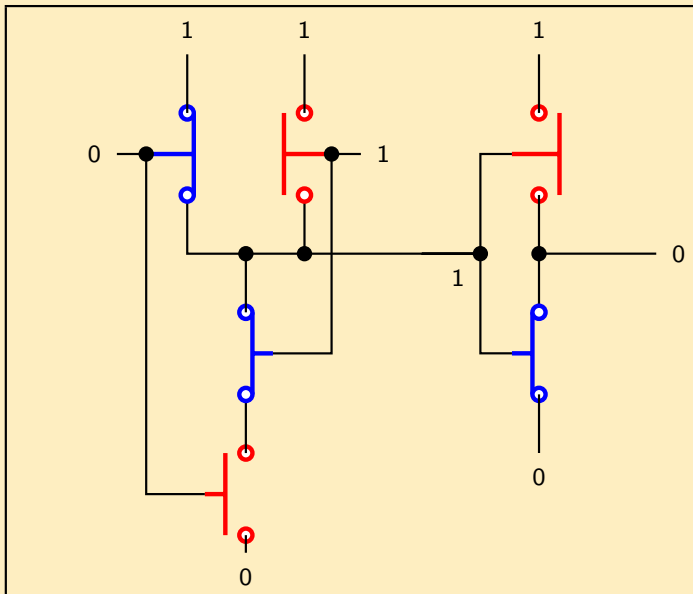




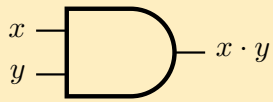
$$z = x \cdot y$$

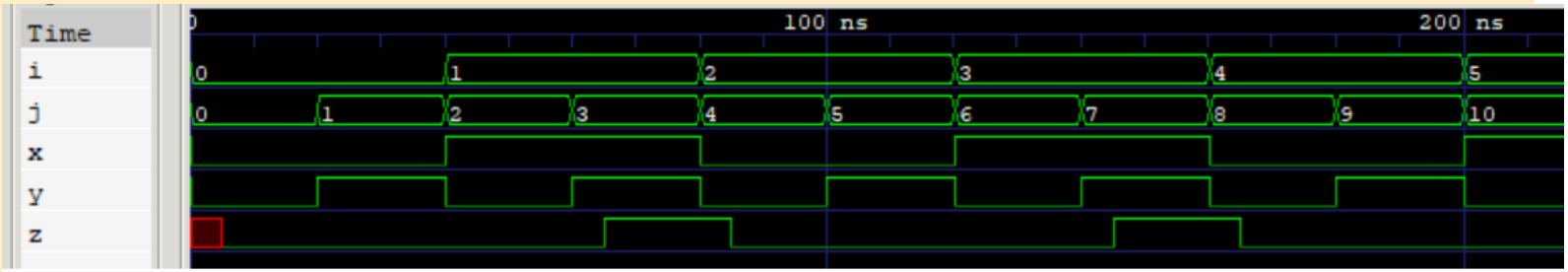
And Gate $x = 0, \quad y = 1$

‘Electronic’ circuit



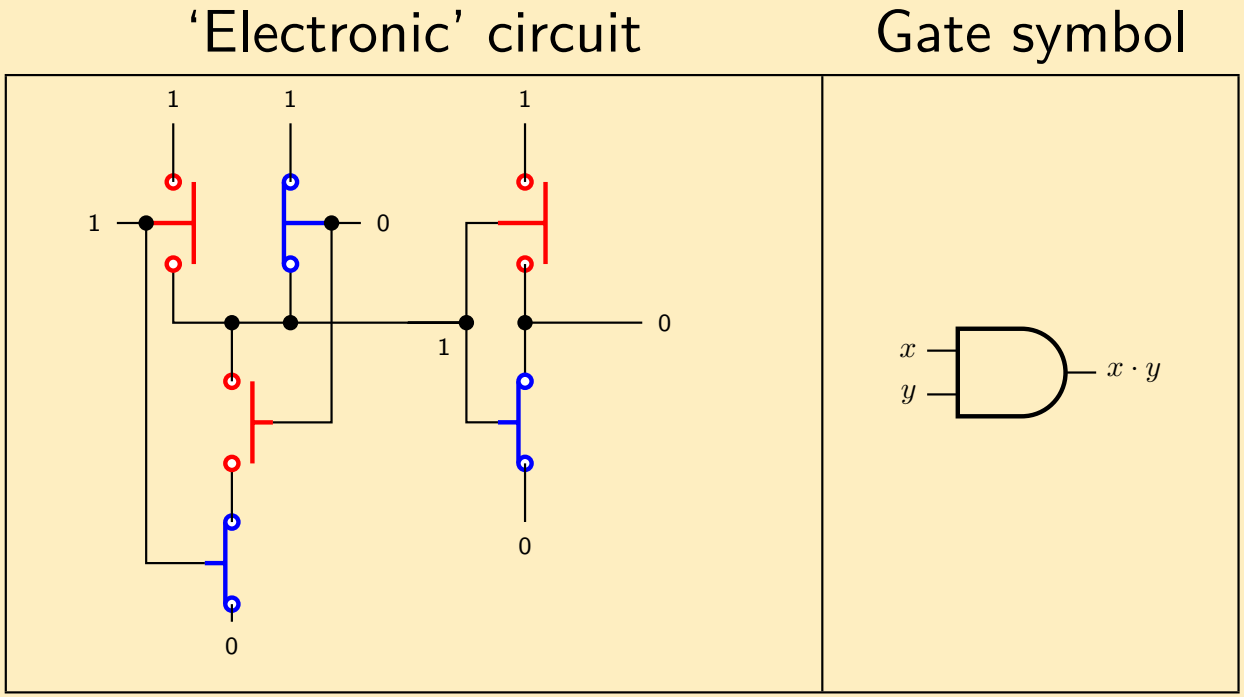
Gate symbol

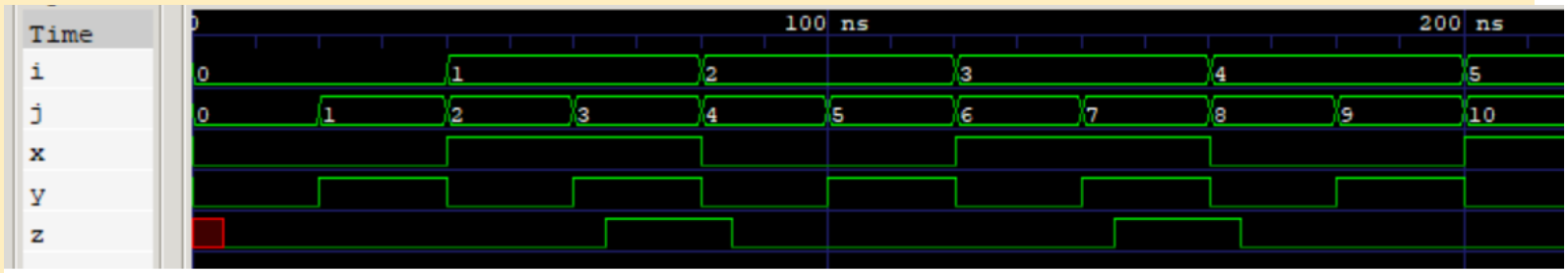




$$z = x \cdot y$$

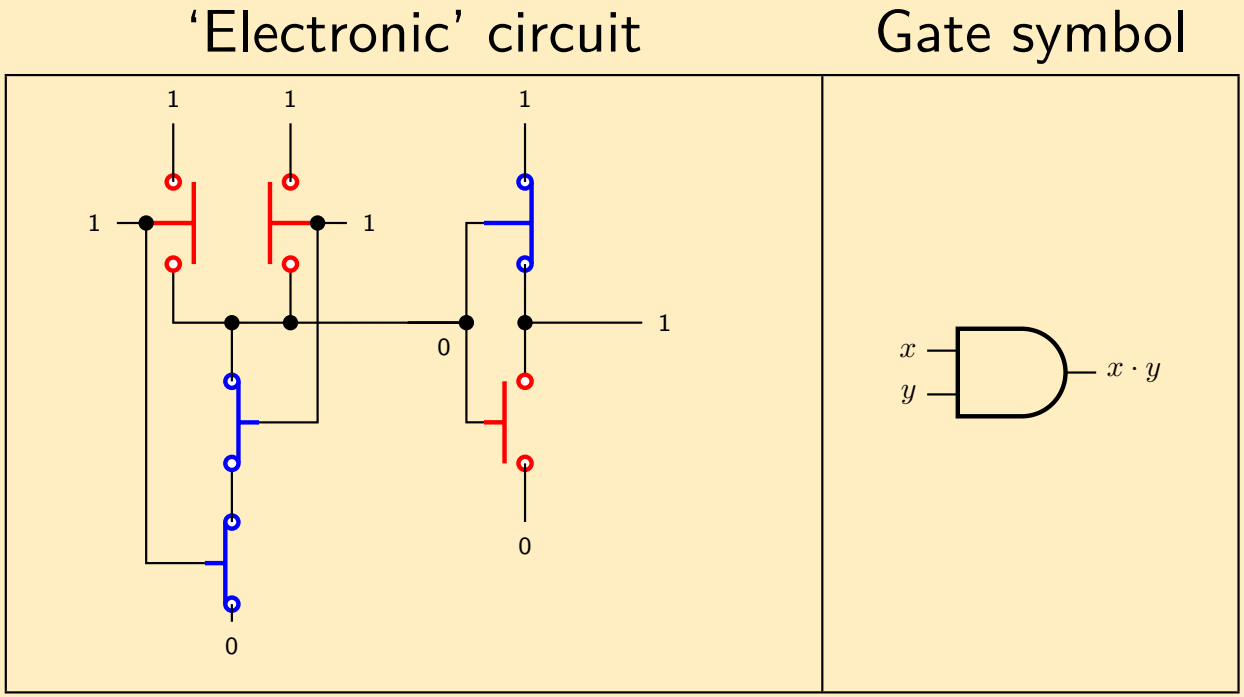
And Gate $x = 1, y = 0$

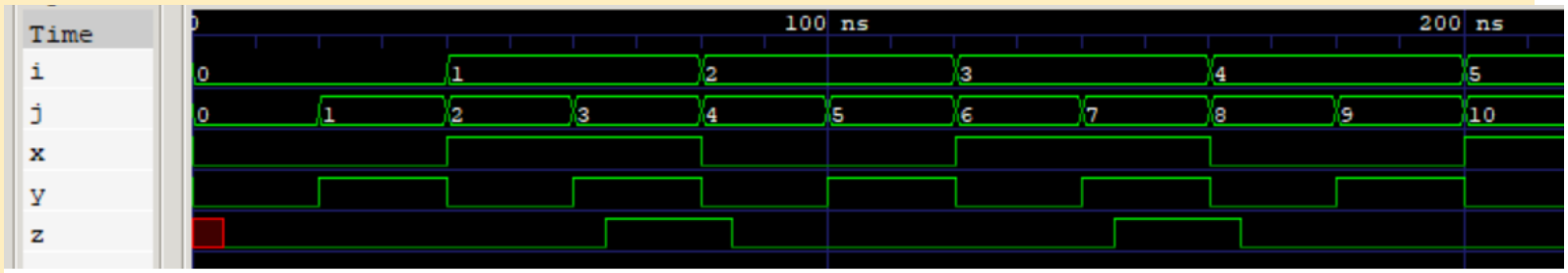




$$z = x \cdot y$$

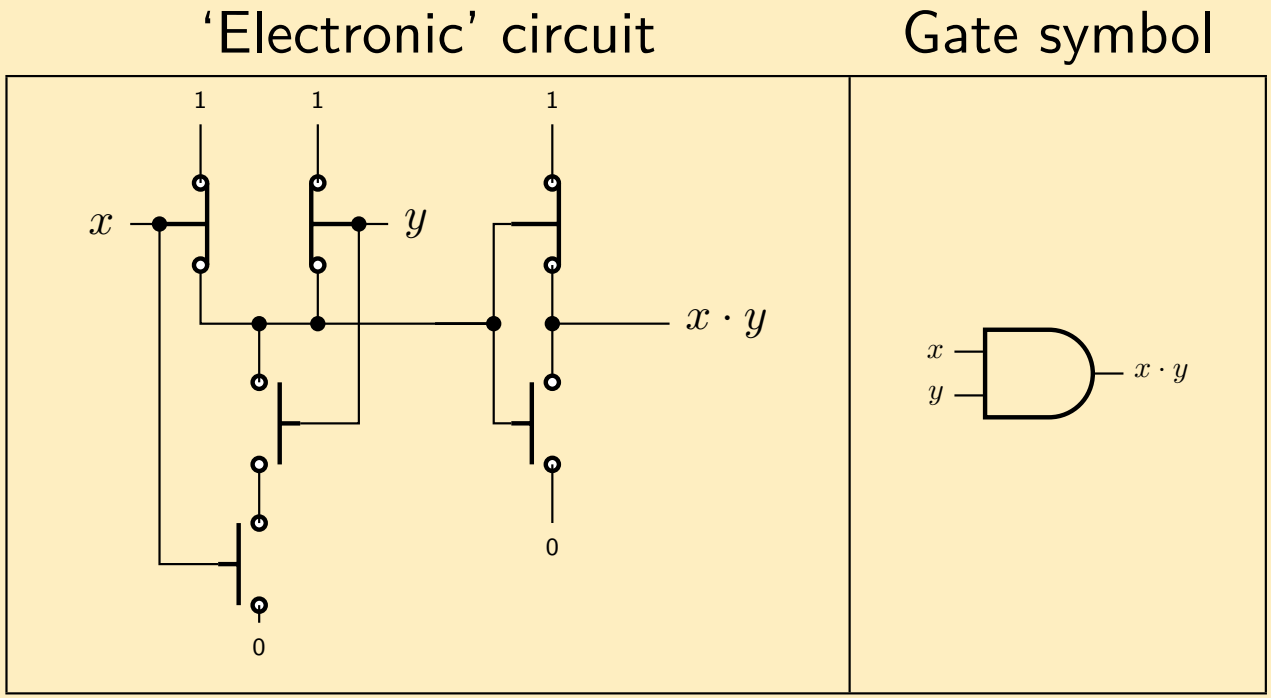
And Gate $x = 1,\ y = 1$

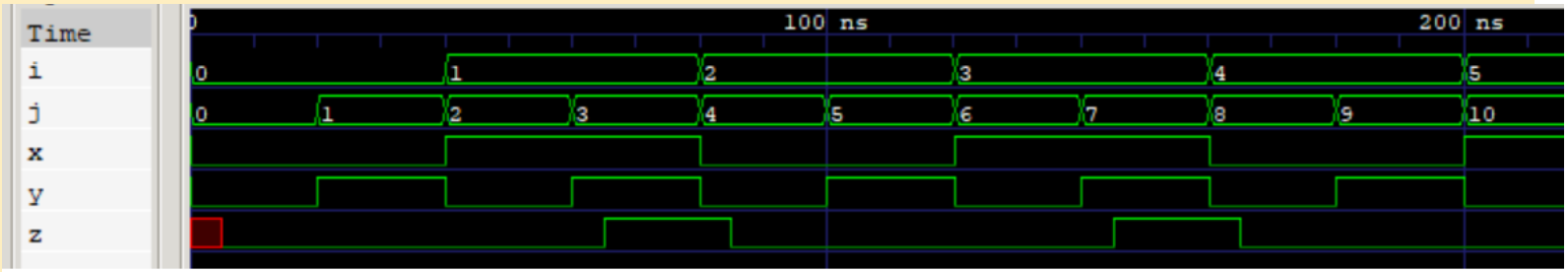




$$z = x \cdot y$$

And Gate

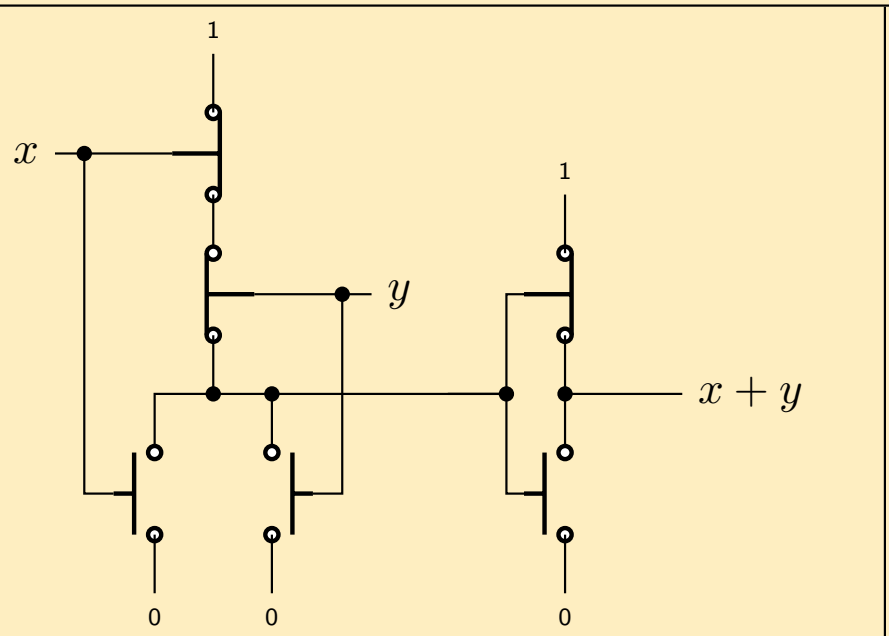




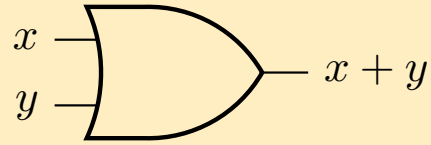
$$z = x \cdot y$$

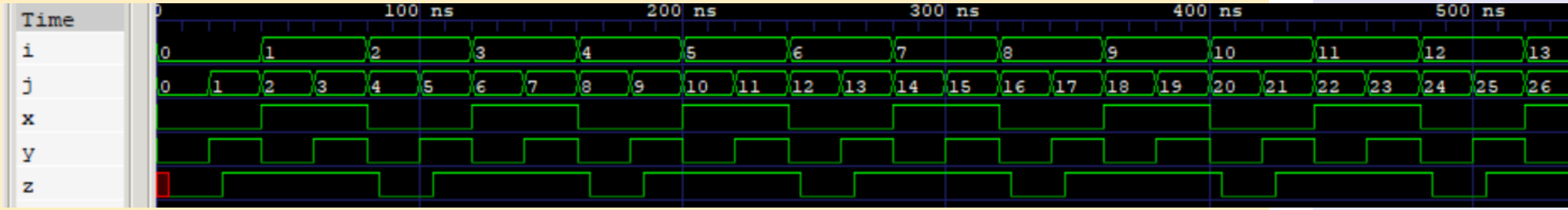
Or Gate

‘Electronic’ circuit



Gate Symbol

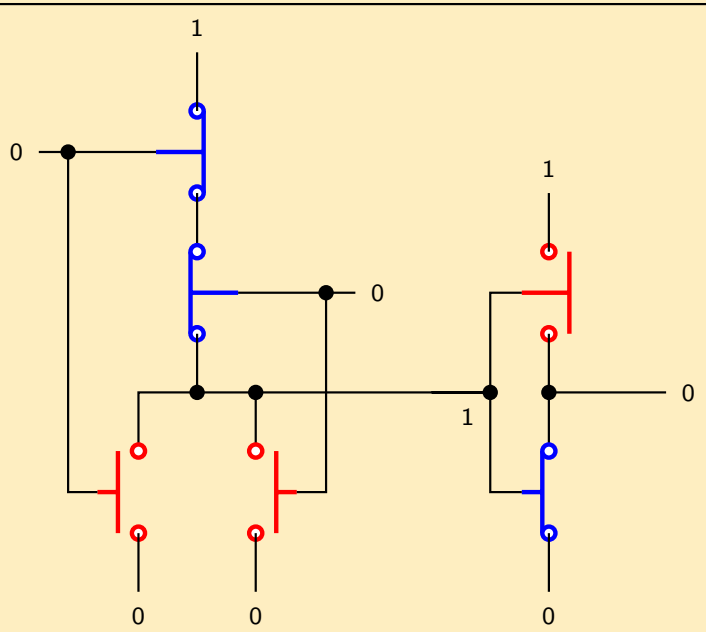




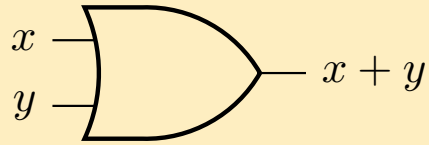
$$z = x + y$$

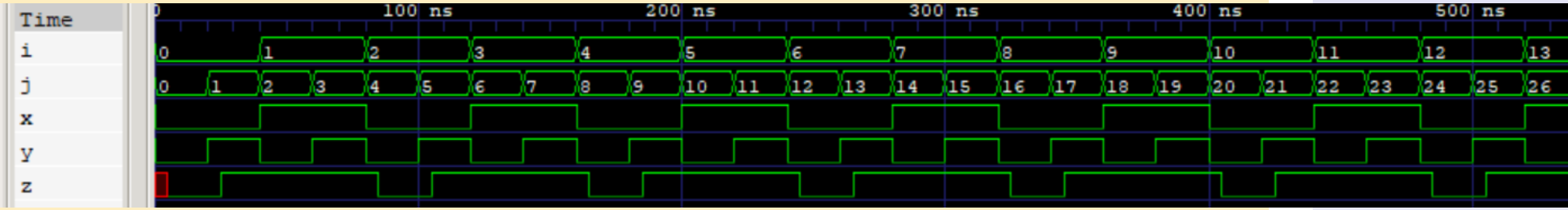
Or Gate $x = 0,\ \ y = 0$

'Electronic' circuit



Gate Symbol

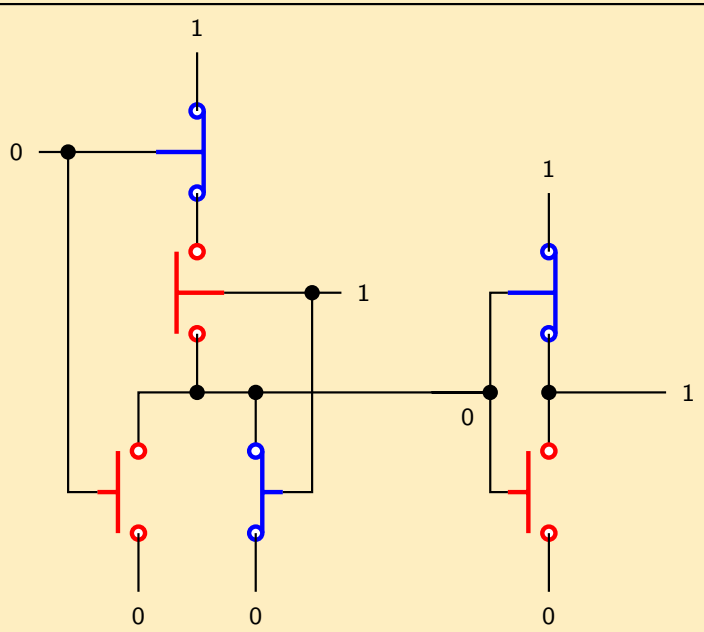




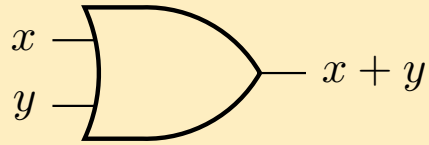
$$z = x + y$$

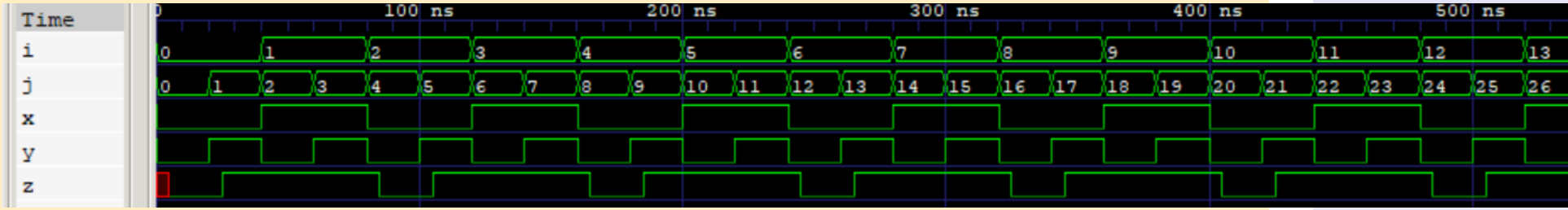
Or Gate $x = 0, \quad y = 1$

'Electronic' circuit



Gate Symbol

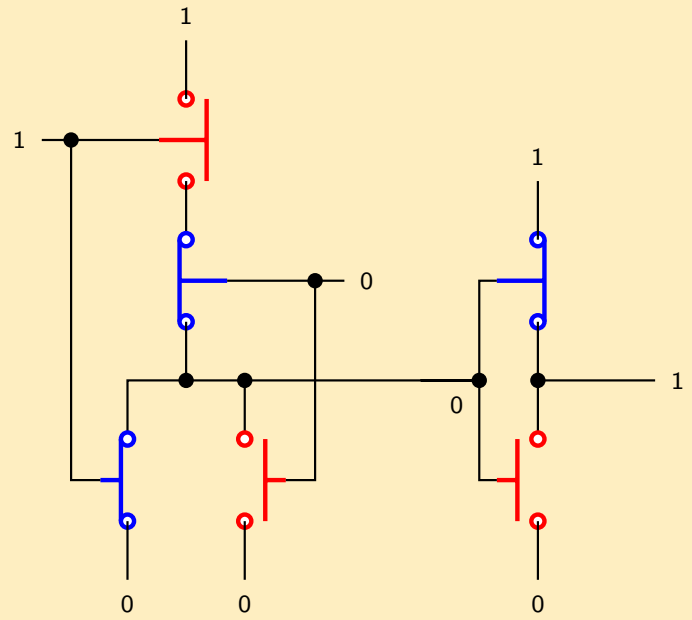




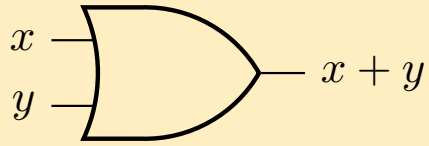
$z = x + y$

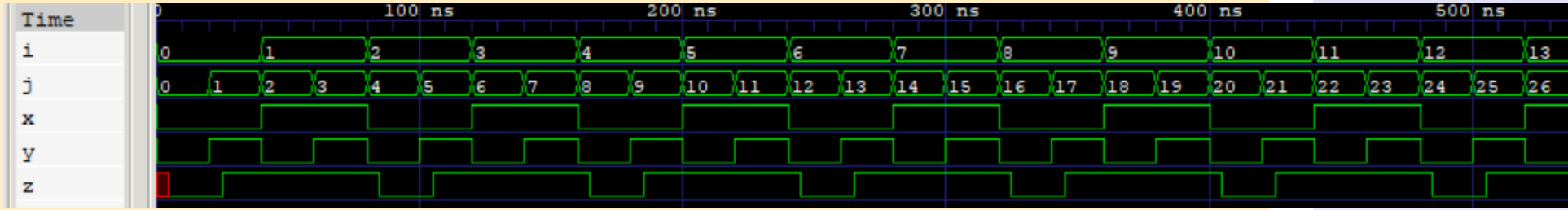
Or Gate $x = 1,\ y = 0$

'Electronic' circuit



Gate Symbol

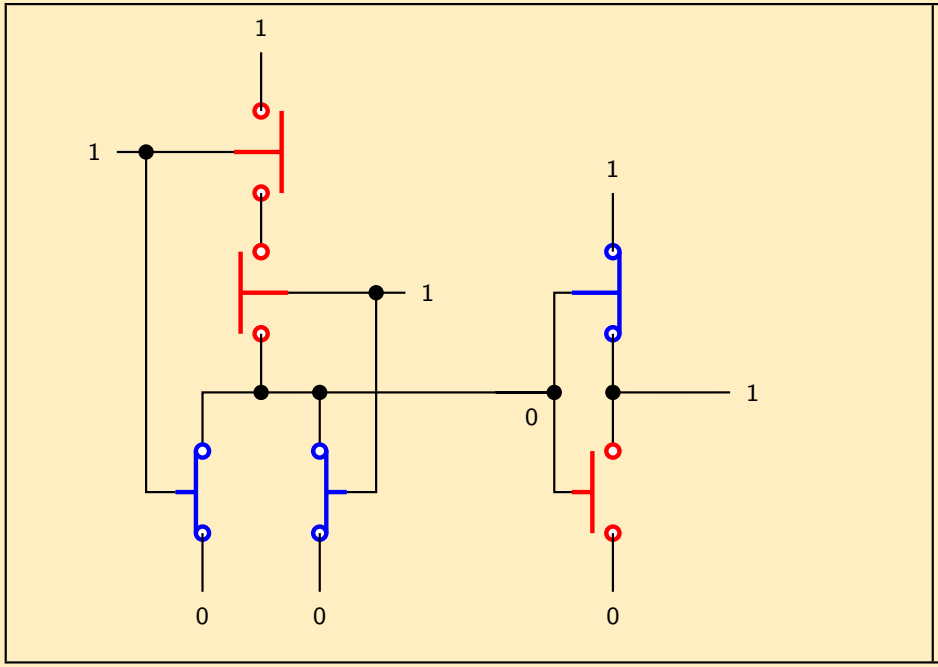




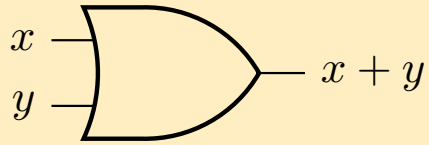
$$z = x + y$$

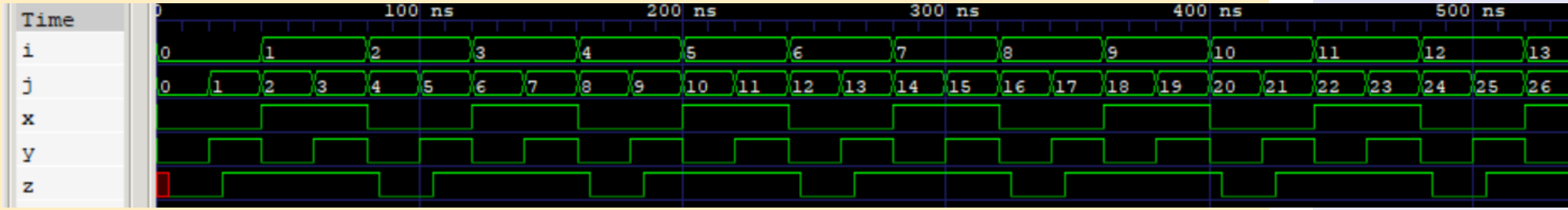
Or Gate $x = 1,\ y = 1$

'Electronic' circuit



Gate Symbol

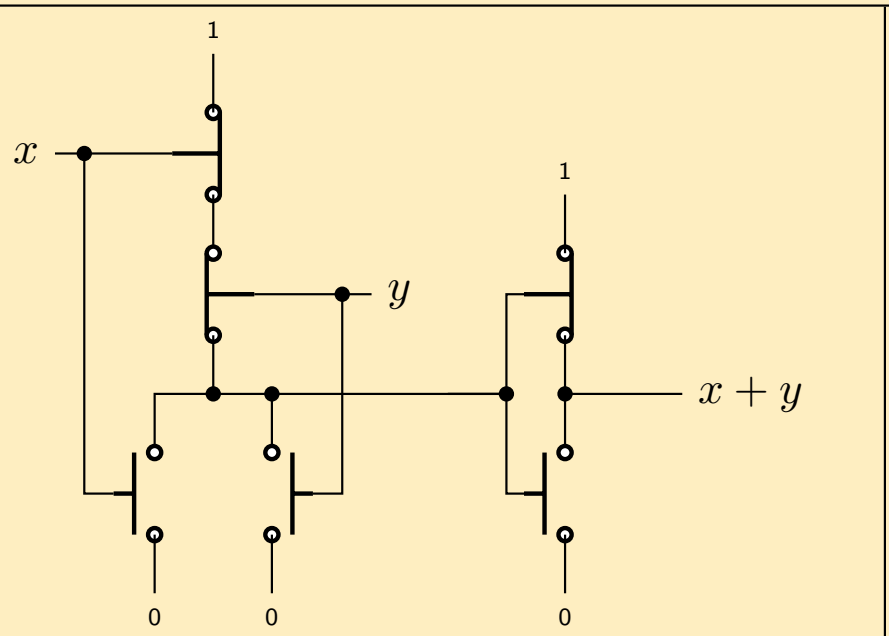




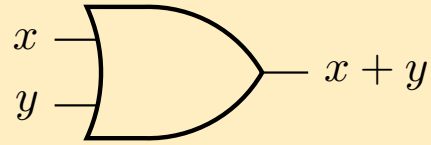
$$z = x + y$$

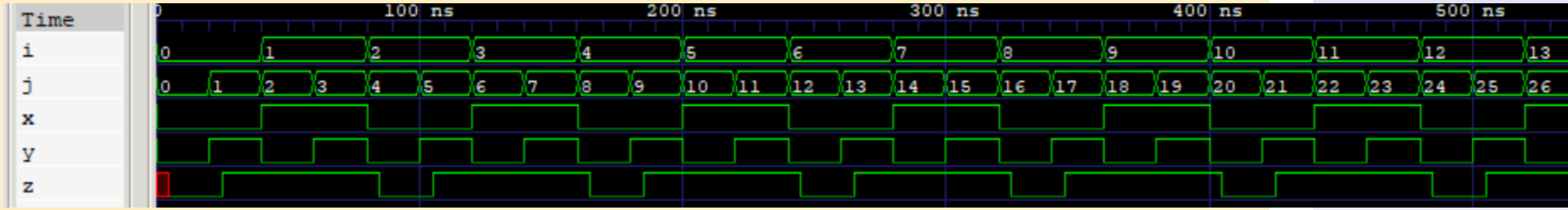
Or Gate

‘Electronic’ circuit



Gate Symbol

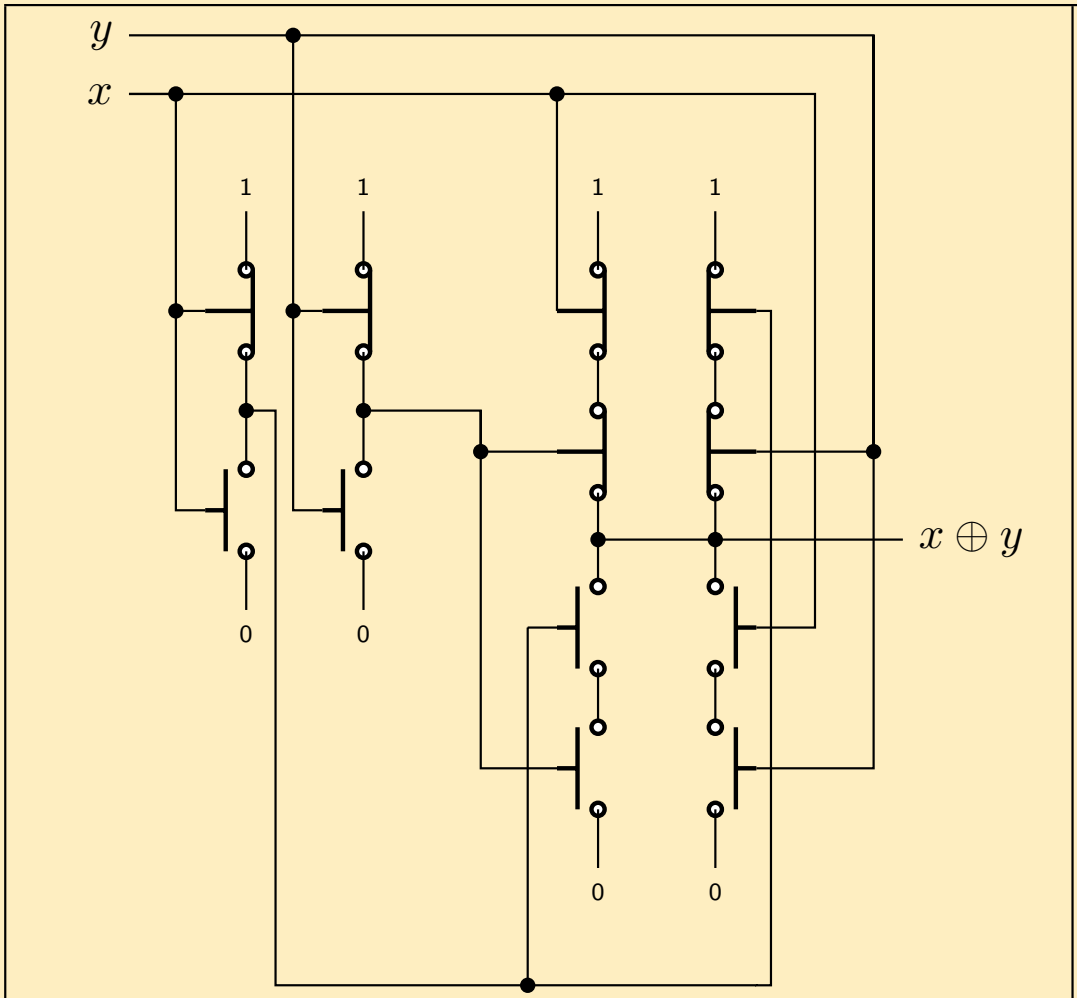




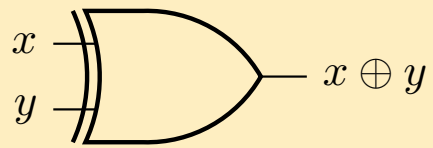
$$z = x + y$$

Xor gate (good place as any)

‘Electronic’ circuit

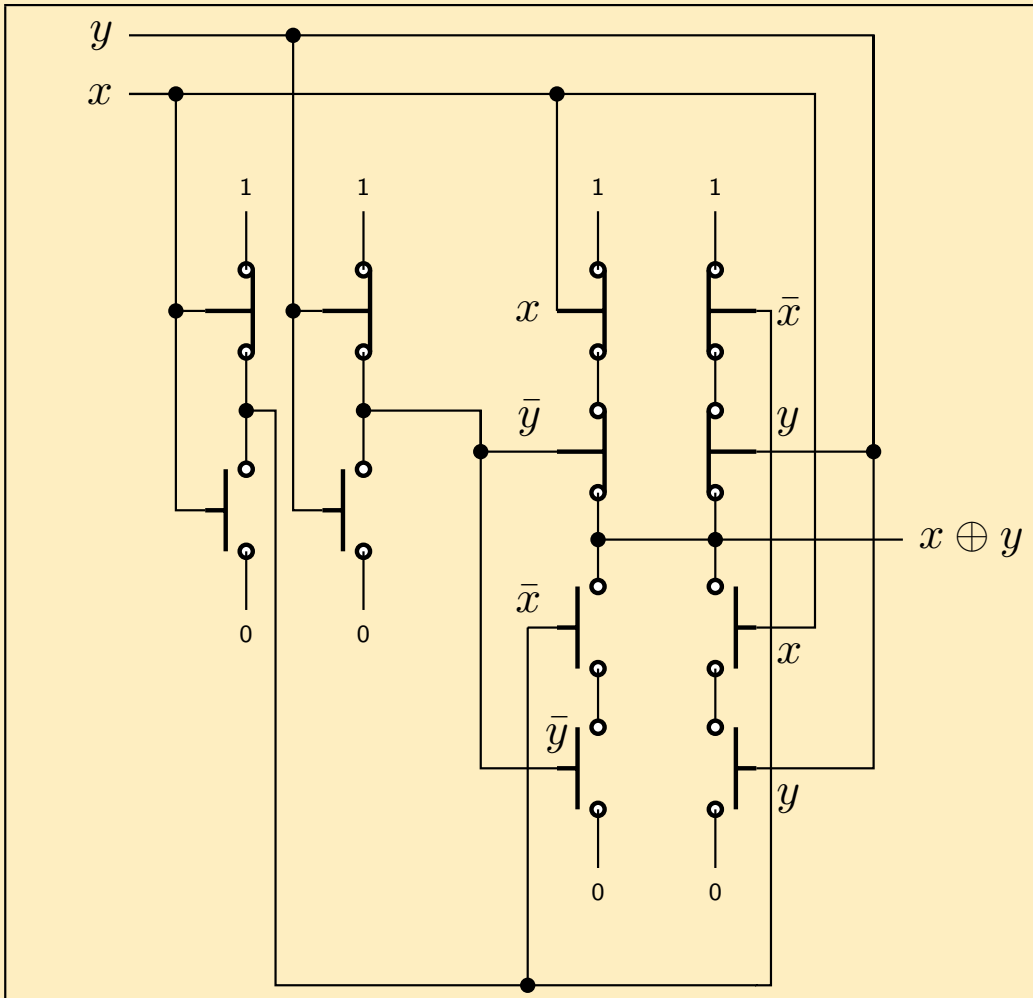


Gate symbol

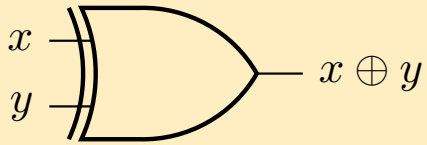


Xor gate

‘Electronic’ circuit

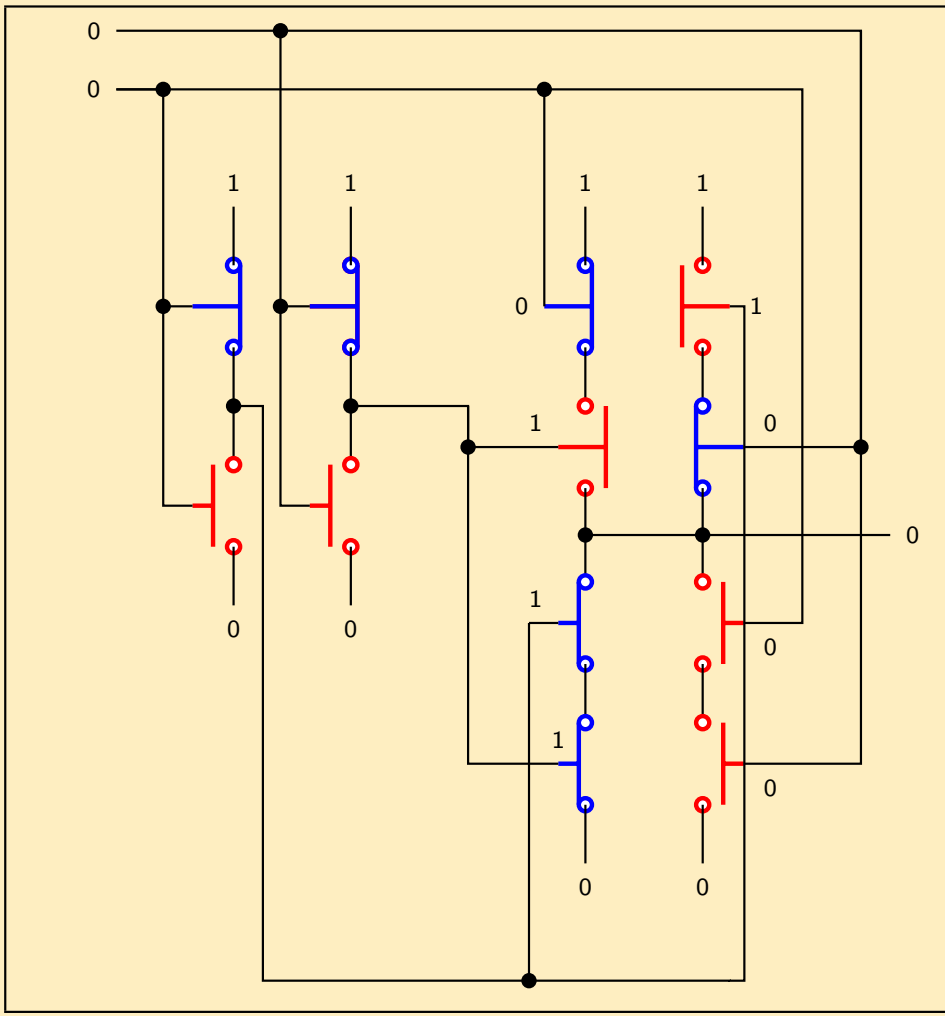


Gate symbol

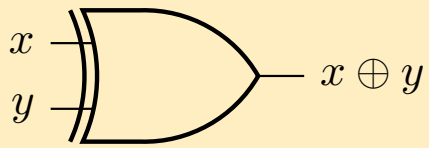


Xor gate $x = 0,\ y = 0$

‘Electronic’ circuit

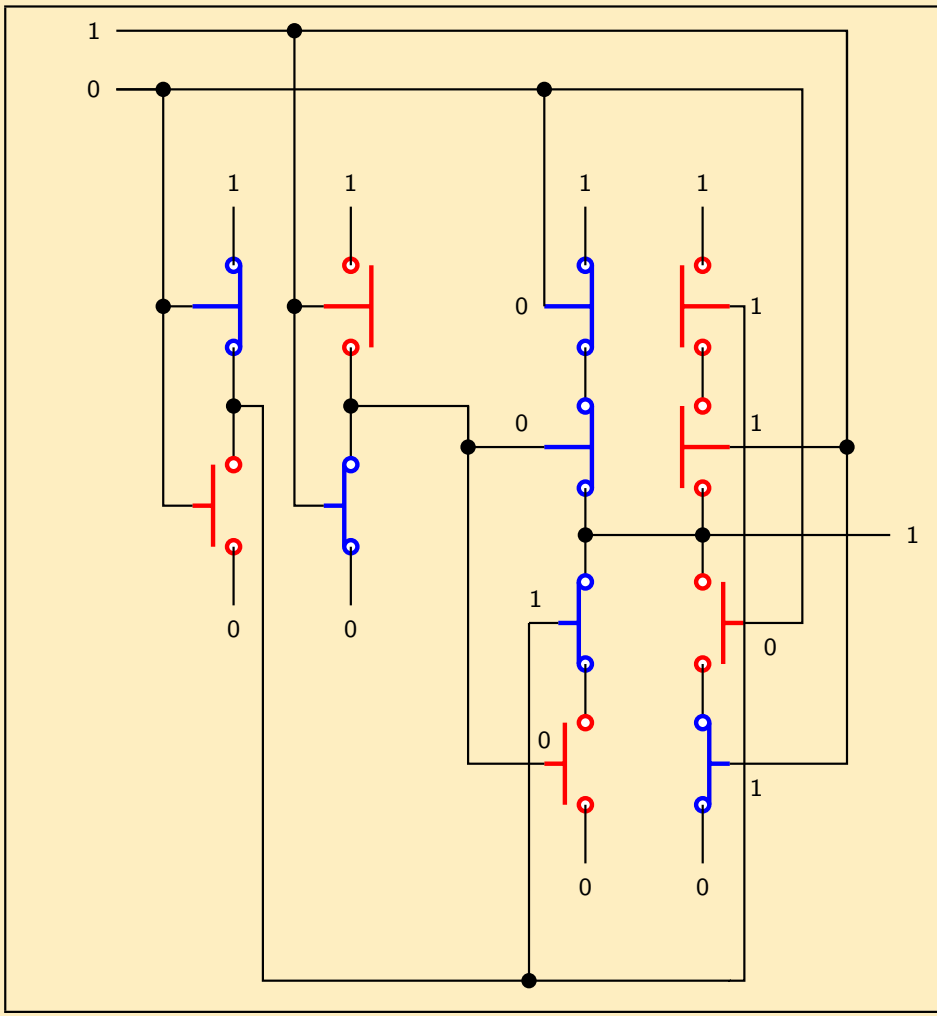


Gate symbol

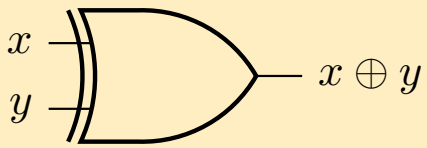


Xor gate $x = 0, \quad y = 1$

‘Electronic’ circuit

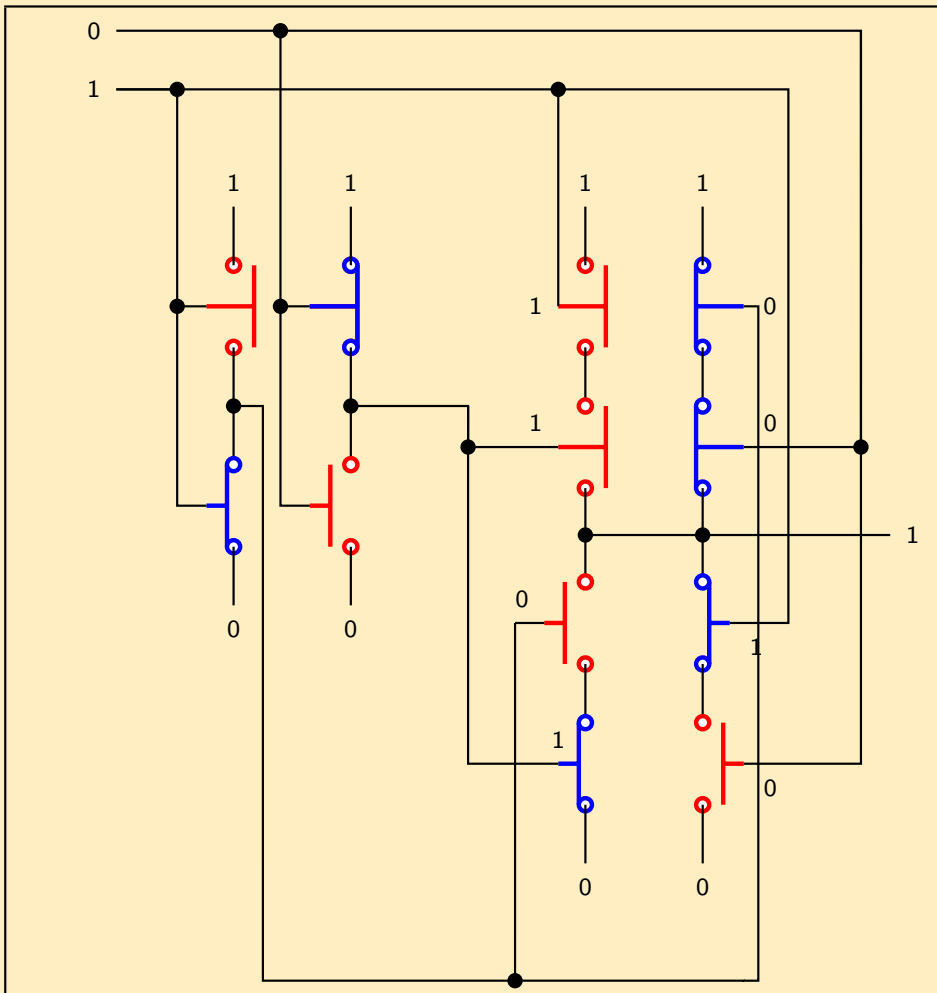


Gate symbol

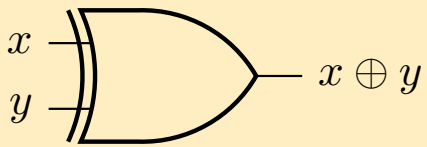


Xor gate $x = 1, y = 0$

‘Electronic’ circuit

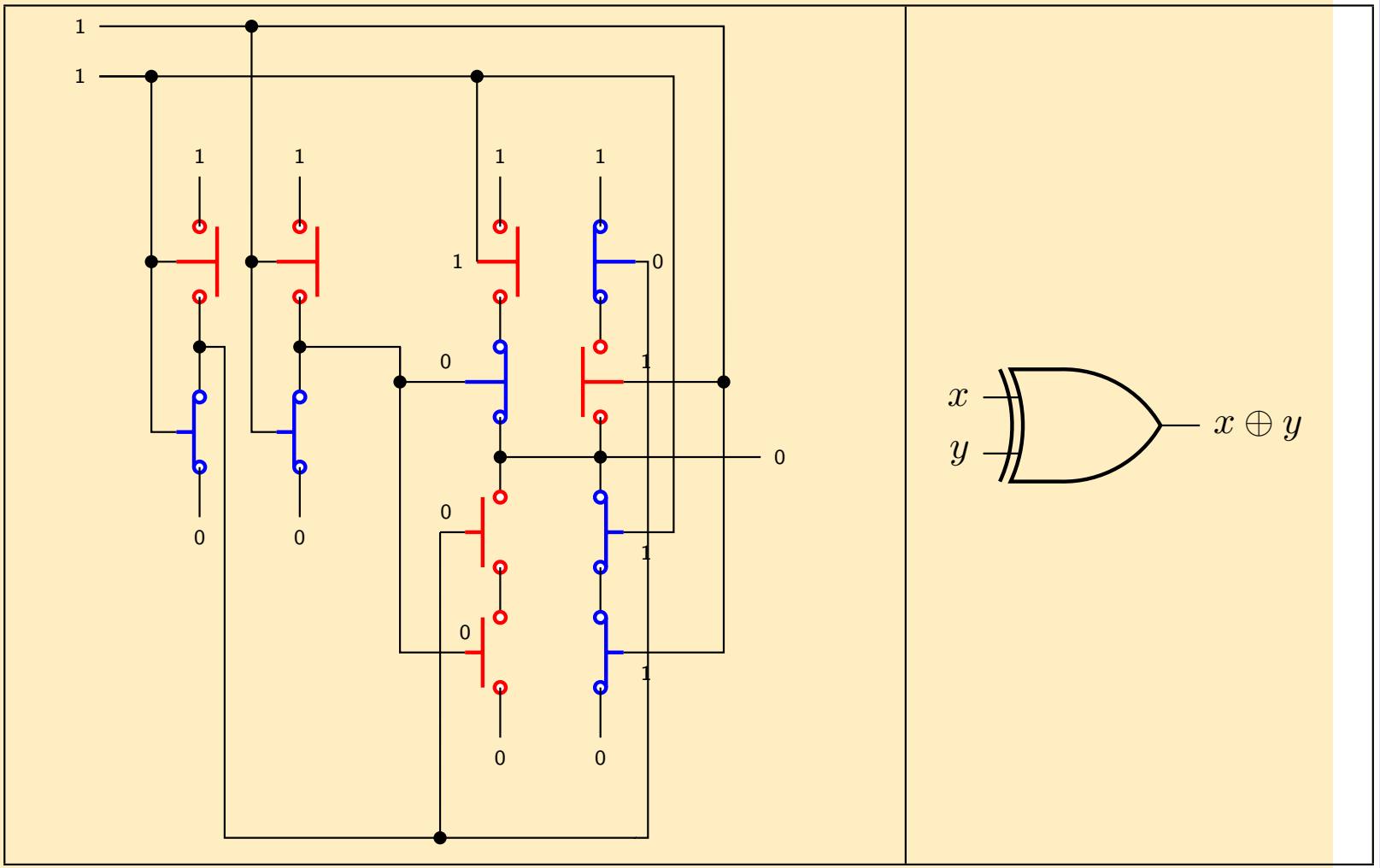


Gate symbol



Xor gate $x = 1,\ y = 1$

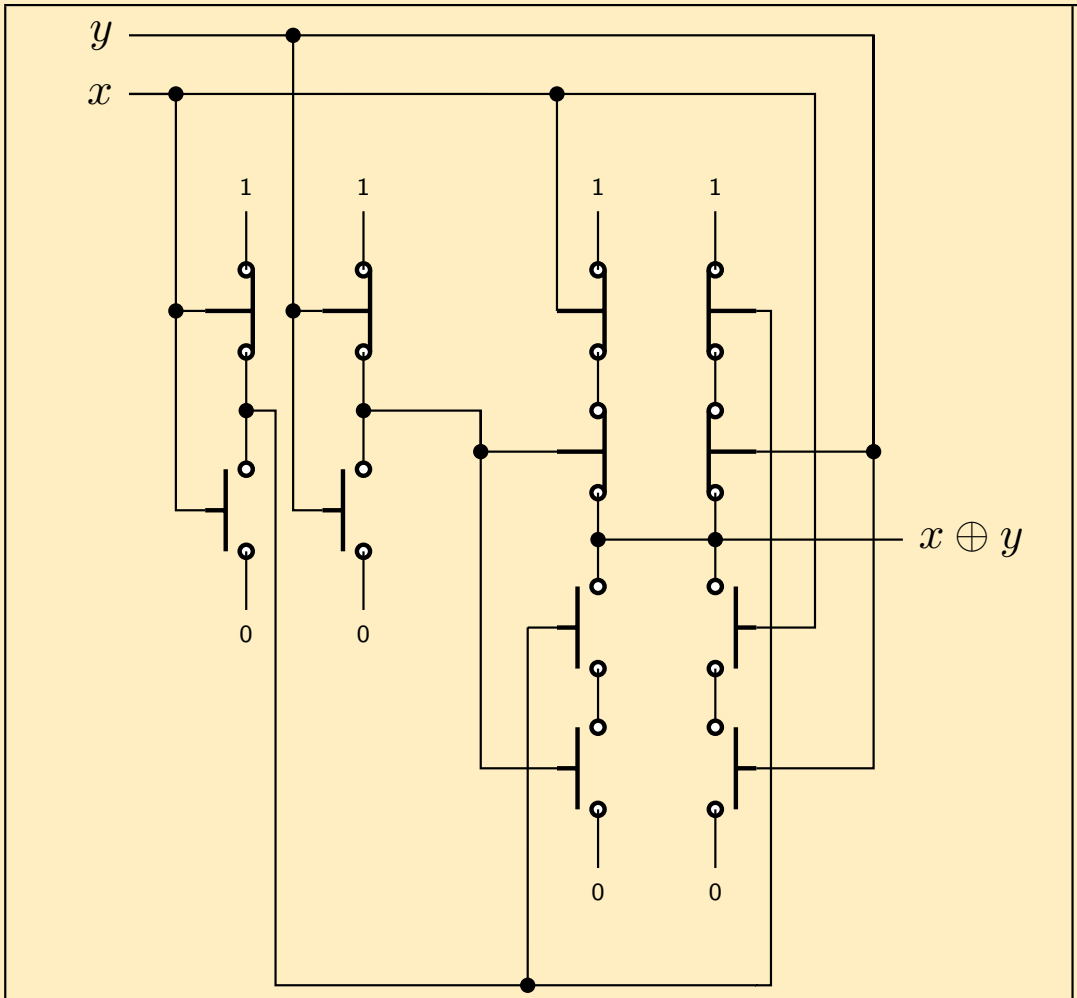
‘Electronic’ circuit



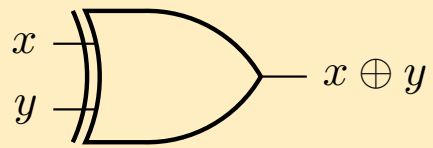
Gate symbol

Xor gate (good place as any)

‘Electronic’ circuit

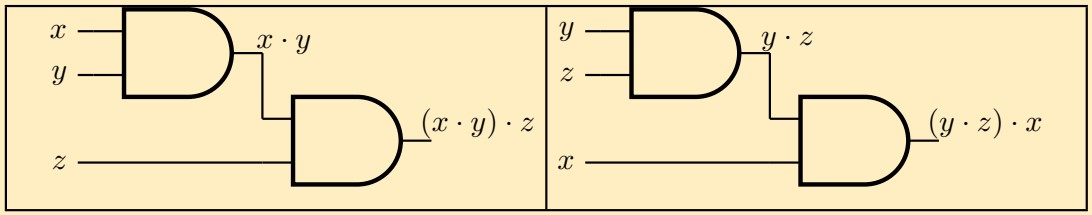


Gate symbol

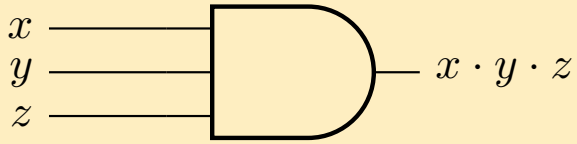


Commutativity and Associativity

- $\cdot$, $+$ and $\oplus$ are commutative: Gate inputs are symmetric
- $\cdot$, $+$ and $\oplus$ are associative: Gatewise we get, e.g.,

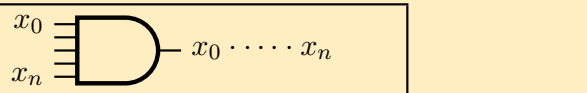
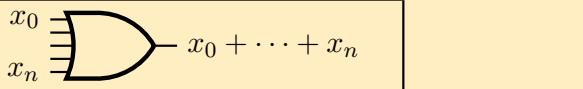
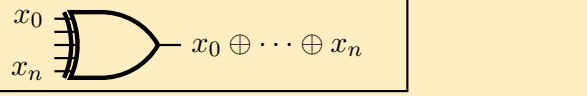


- Or we can use gate with three inputs, e.g.,



- There is a performance **hit** for each additional input pin
- Usually, two successive gate are slower

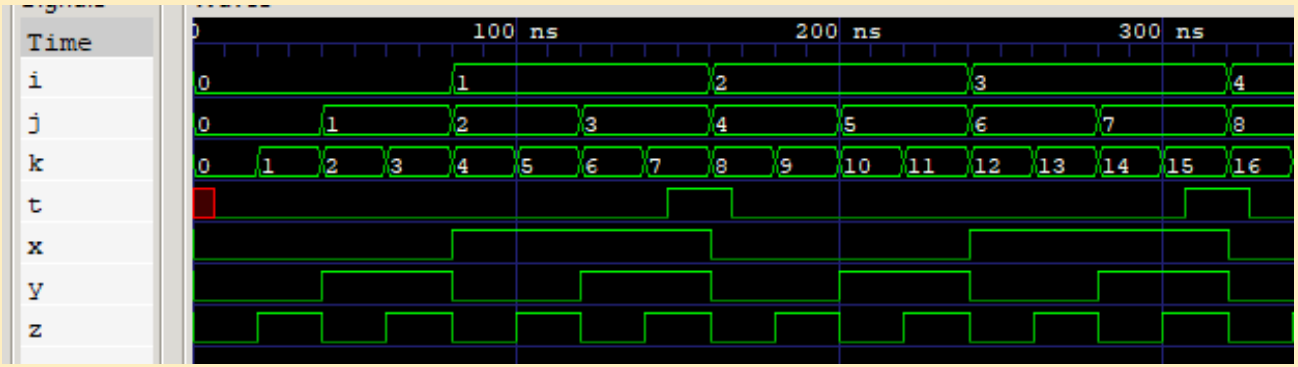
Multi Inputs Gates

And gate	
Or gate	
Xor gate	

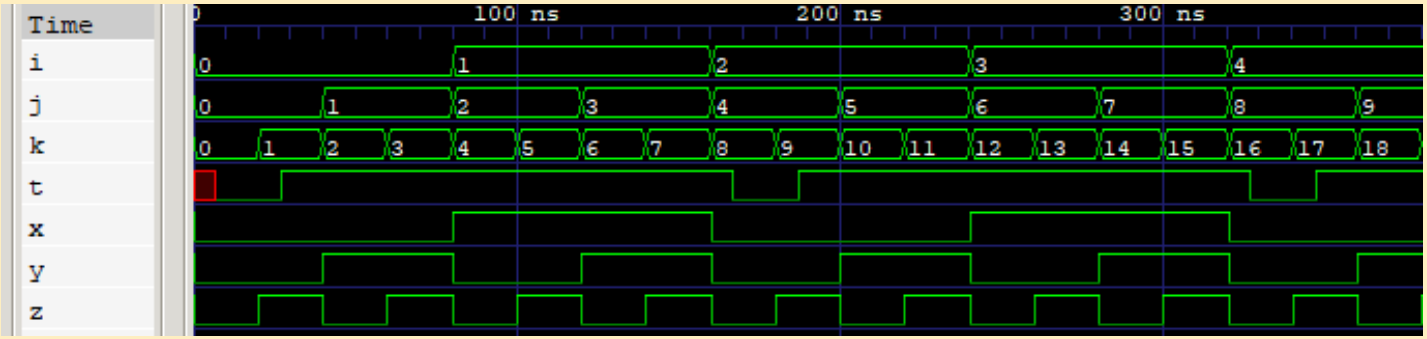
- Most technologies allow at most 5 input pins

3-Inputs Gates waveforms

$t = x \cdot y \cdot z$



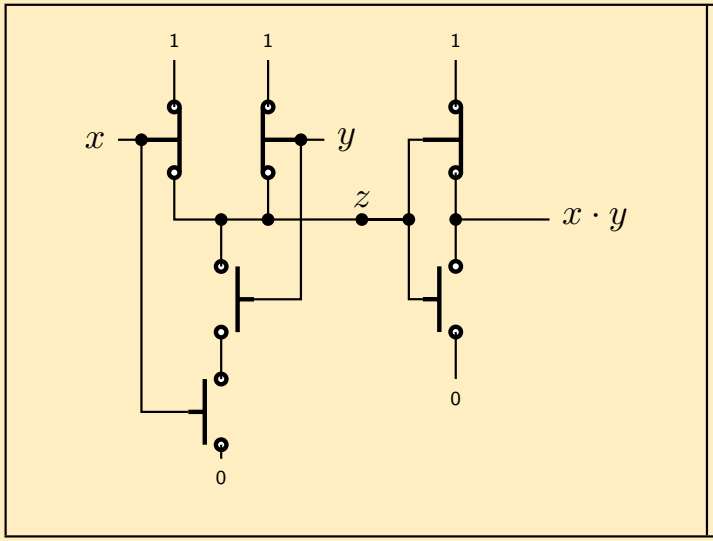
$t = x + y + z$



‘Electronics’ observation regarding gates

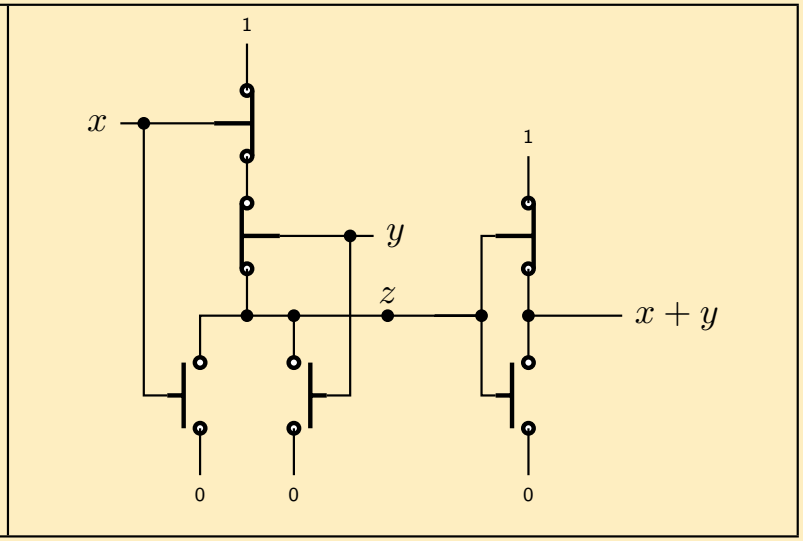
The realization of both ⊐ and ⊐ have ▷ at their end

And ‘Electronics’



$$z = \overline{x \cdot y}$$
$$= x \uparrow y$$

Or ‘Electronics’



$$z = \overline{x + y}$$
$$= x \downarrow y$$

Nand ($\uparrow$) and Nor ($\downarrow$) Properties

Nand	Nor
$\bar{x} = \overline{x \cdot x} = x \uparrow x$	$\bar{x} = \overline{x + x} = x \downarrow x$
$x \cdot y = \overline{\overline{x \cdot y}} = \overline{x \uparrow y} = (x \uparrow y) \uparrow (x \uparrow y)$	$x \cdot y = \overline{\overline{x \cdot y}} = \overline{\bar{x} + \bar{y}} = (x \downarrow x) \downarrow (y \downarrow y)$
$x + y = \overline{\overline{x + y}} = \overline{\bar{x} \cdot \bar{y}} = (x \uparrow x) \uparrow (y \uparrow y)$	$x + y = \overline{\overline{x + y}} = \overline{x \downarrow y} = (x \downarrow y) \downarrow (x \downarrow y)$

Complete System

Definition

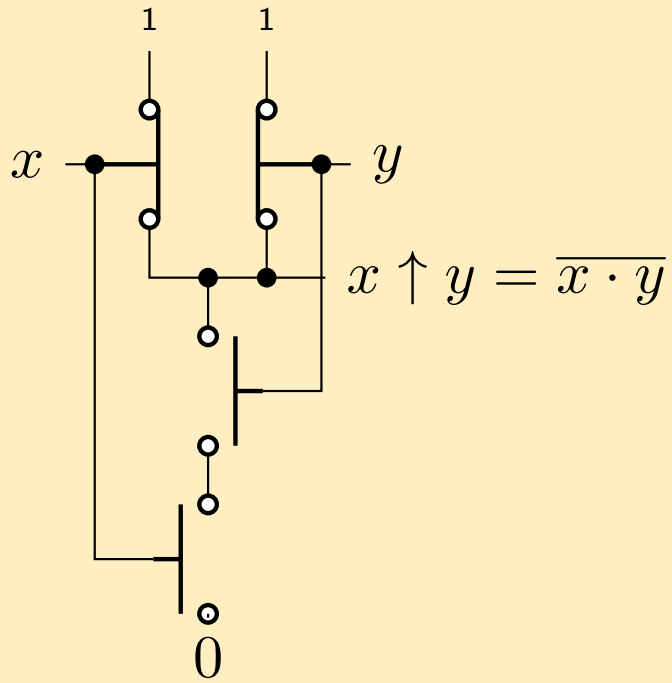
A set of functions and constants is a complete system if every boolean function can be defined with them

Corollary

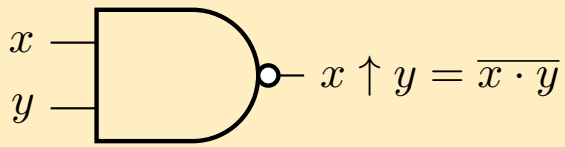
Both $\{\uparrow\}$ and $\{\downarrow\}$ are complete systems.

Nand Gate

‘Electronic’ circuit

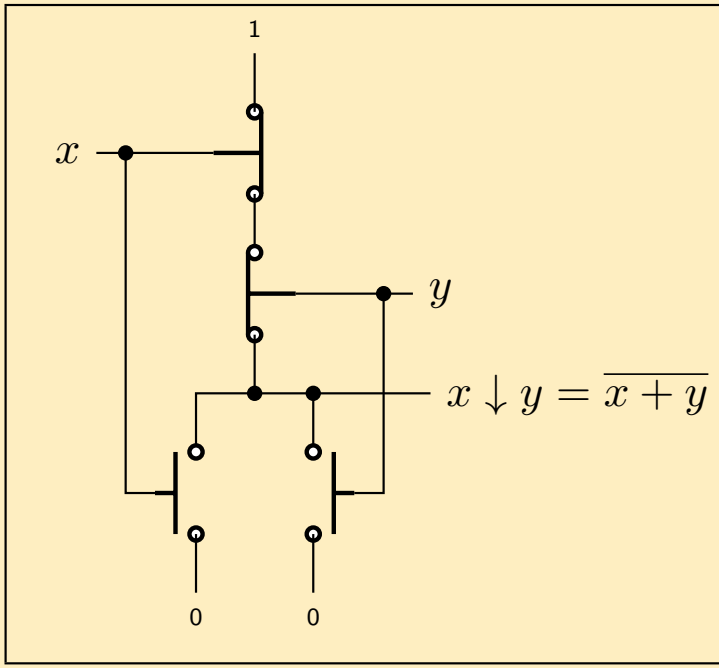


Logic symbol

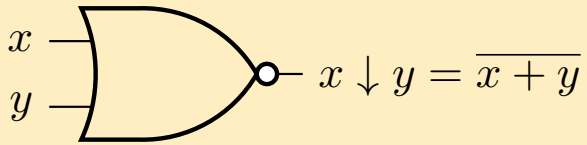


Nor Gate

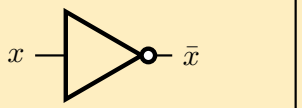
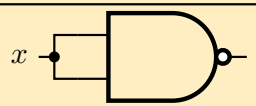
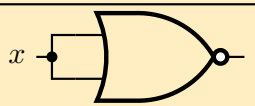
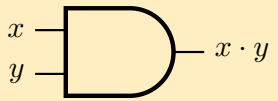
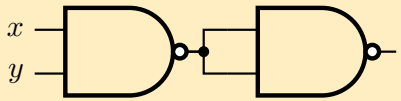
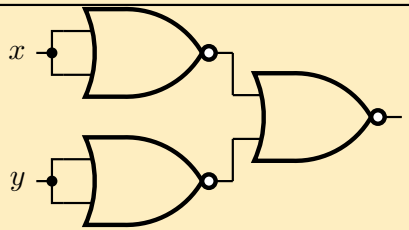
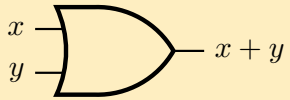
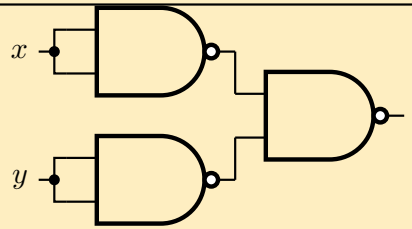
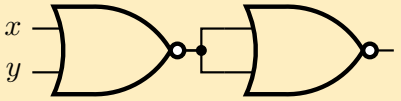
‘Electronic’ circuit



Logic symbol



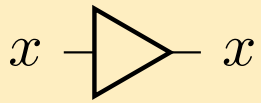
Realizing the operatros with Nand and Nor

Gate	Nand	Nor
	 $x \uparrow x = \bar{x}$	 $x \downarrow x = \bar{x}$
	 $(x \uparrow y) \uparrow (x \uparrow y) = x \cdot y$	 $(x \downarrow x) \downarrow (y \downarrow y) = x \cdot y$
	 $(x \uparrow x) \uparrow (y \uparrow y) = x + y$	 $(x \downarrow y) \downarrow (x \downarrow y) = x + y$

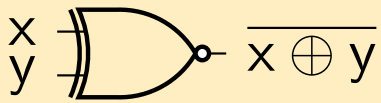
Realization of XOR should be here

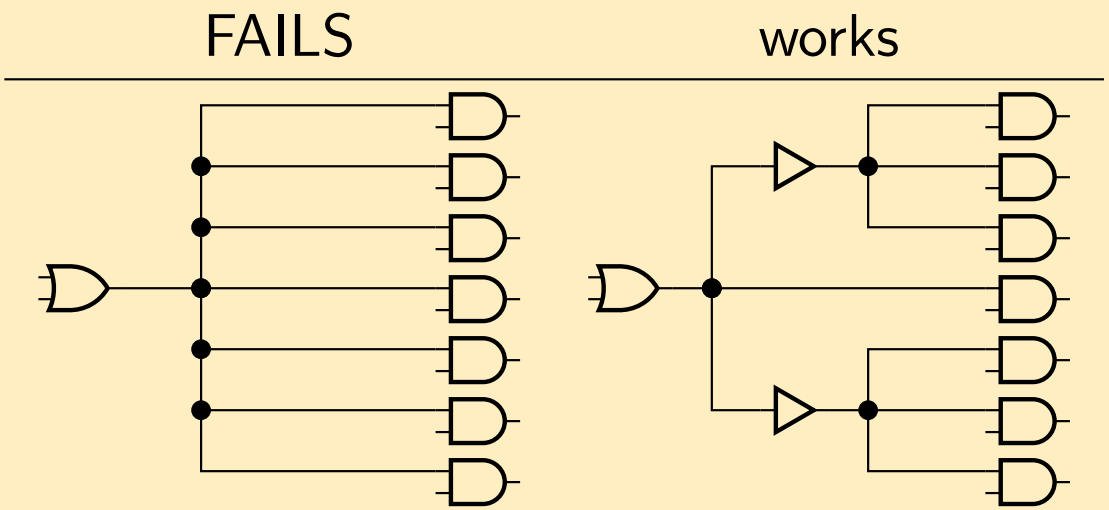
Two final gates

- The buffer gate (important in real circuits):



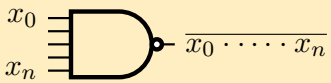
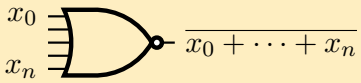
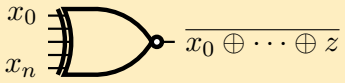
- The nxor gate:





Multi Inputs Variants

Associative complement

Nand gate	
Nor gate	
Nxor gate	

(Carmi) Lecture 6 reached here

Combinational Circuits

(Realization of boolean functions)

Recall that the booean operators are boolean functions

Functions to be Realized

- Seven Segment
- mod 3
- Half adder
- 2-bit binary adder
- Full adder
- Binary adder

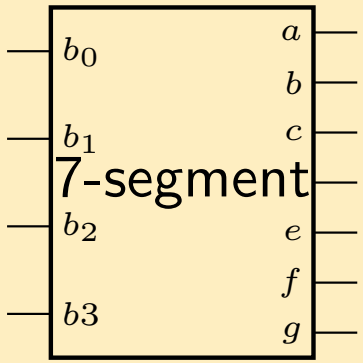
The boolean functions are from the previous chapter

Realizing the Seven-Segment

The formuale are from the previous lecture

1. Block diagram
2. Logic circuit

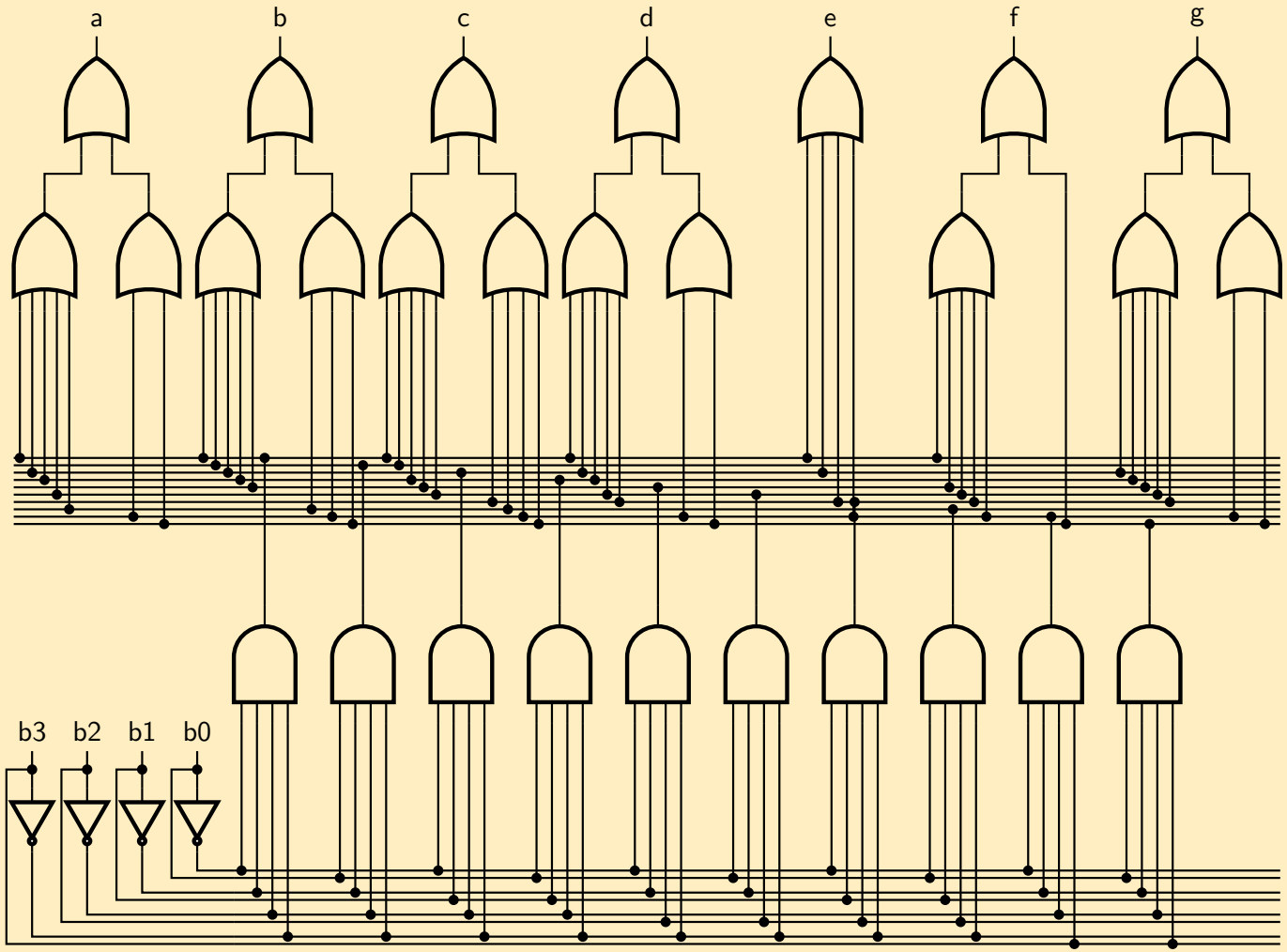
Seven-Segment (block diagram)



Seven-Segment (formulae)

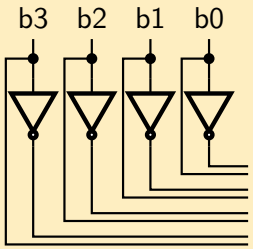
SOP	POS
$a = m_0 + m_2 + m_3 + m_5 + m_7 + m_8 + m_9$	$a = \bar{m}_1\bar{m}_4\bar{m}_6$
$b = m_0 + m_1 + m_2 + m_3 + m_4 + m_7 + m_8 + m_9$	$b = \bar{m}_5\bar{m}_6$
$c = m_0 + m_1 + m_3 + m_4 + m_5 + m_6 + m_7 + m_8 + m_9$	$c = \bar{m}_2$
$d = m_0 + m_2 + m_3 + m_5 + m_6 + m_8$	$d = \bar{m}_1\bar{m}_4\bar{m}_7$
$e = m_0 + m_2 + m_6 + m_8$	$e = \bar{m}_1\bar{m}_3\bar{m}_4\bar{m}_5\bar{m}_7\bar{m}_9$
$f = m_0 + m_4 + m_5 + m_6 + m_8 + m_9$	$f = \bar{m}_1\bar{m}_2\bar{m}_3\bar{m}_7$
$g = m_2 + m_3 + m_4 + m_5 + m_6 + m_8 + m_9$	$g = \bar{m}_0\bar{m}_1\bar{m}_7$

Seven Segment (SOP)

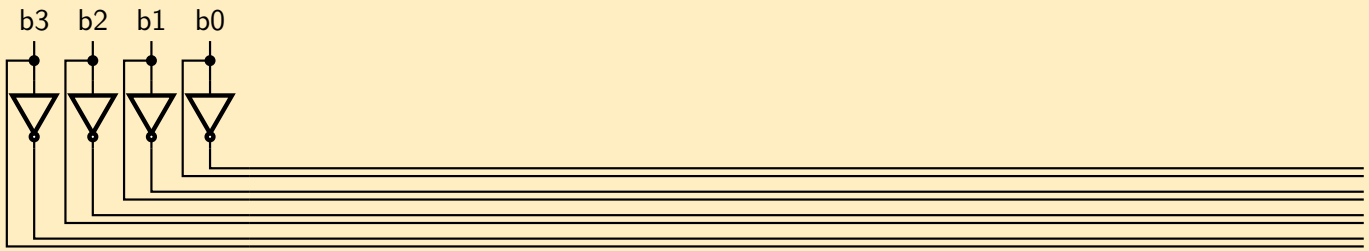


Seven Segment (SOP)

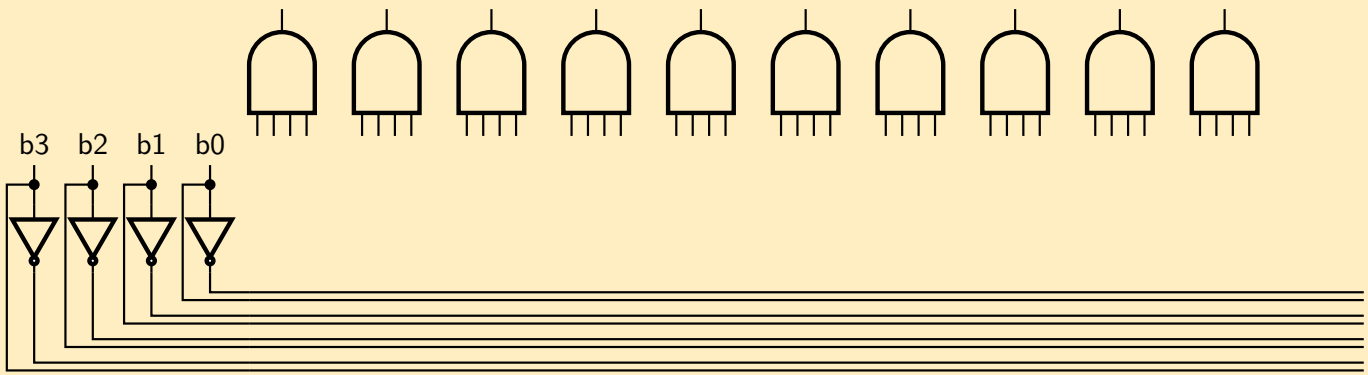
Seven Segment (SOP)



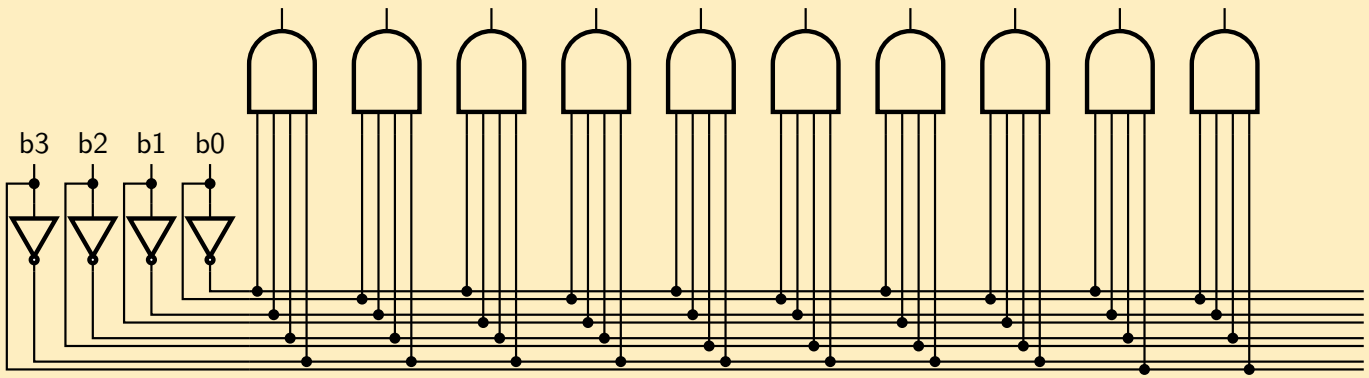
Seven Segment (SOP)



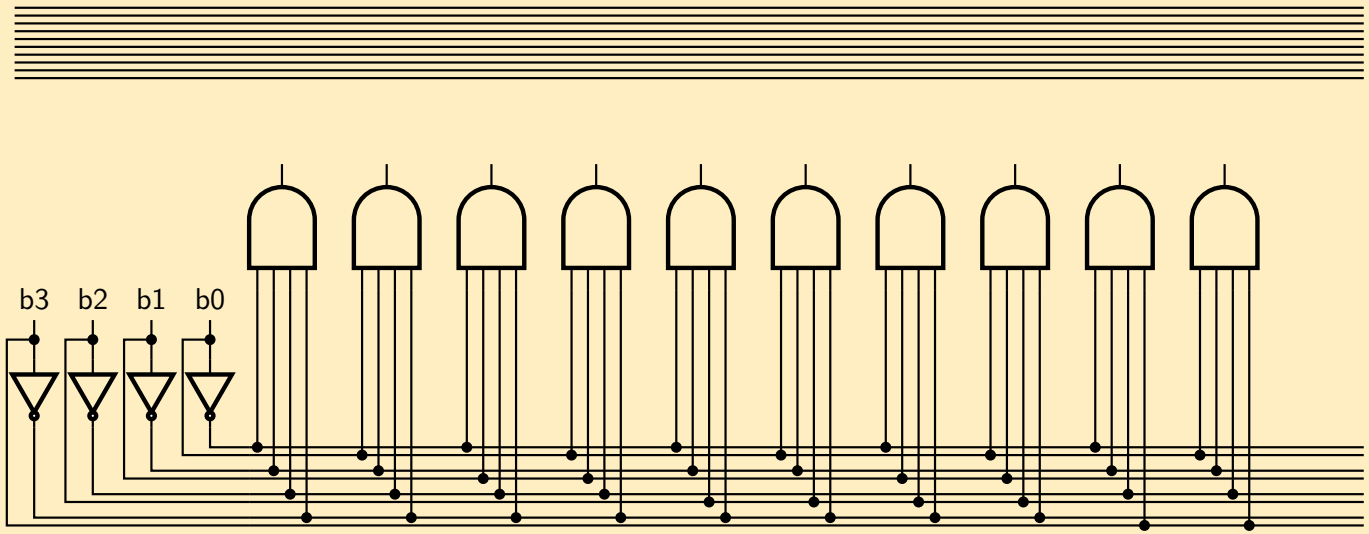
Seven Segment (SOP)



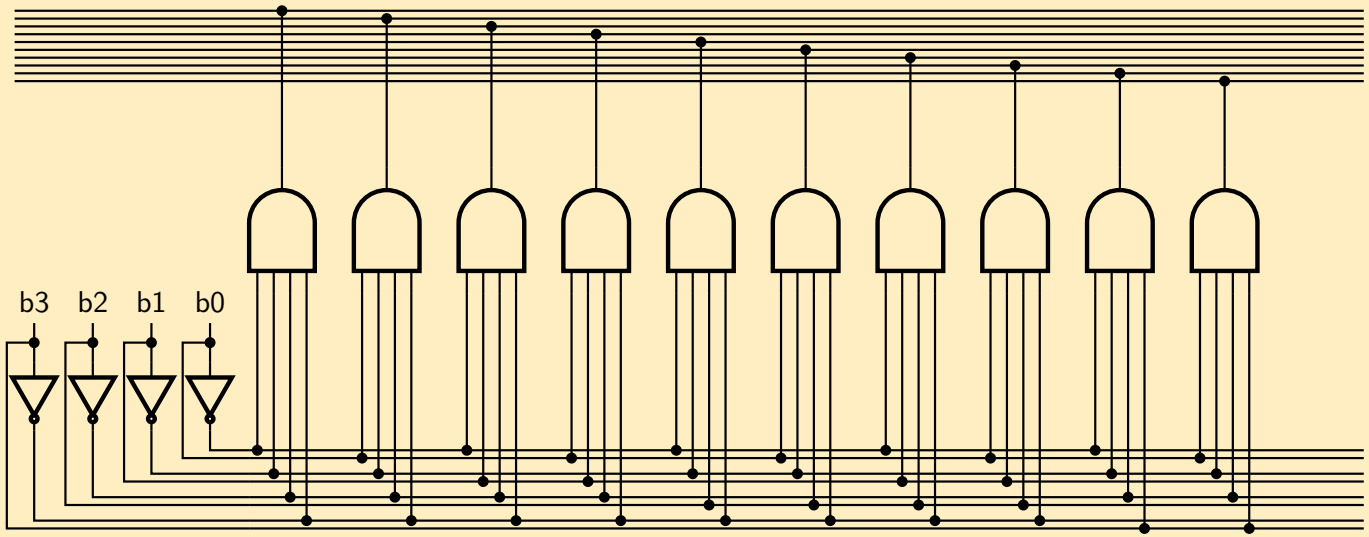
Seven Segment (SOP)



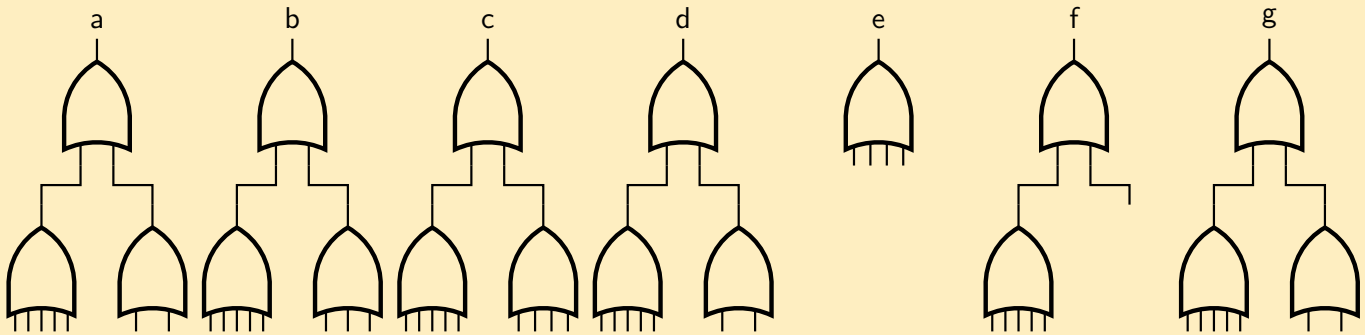
Seven Segment (SOP)

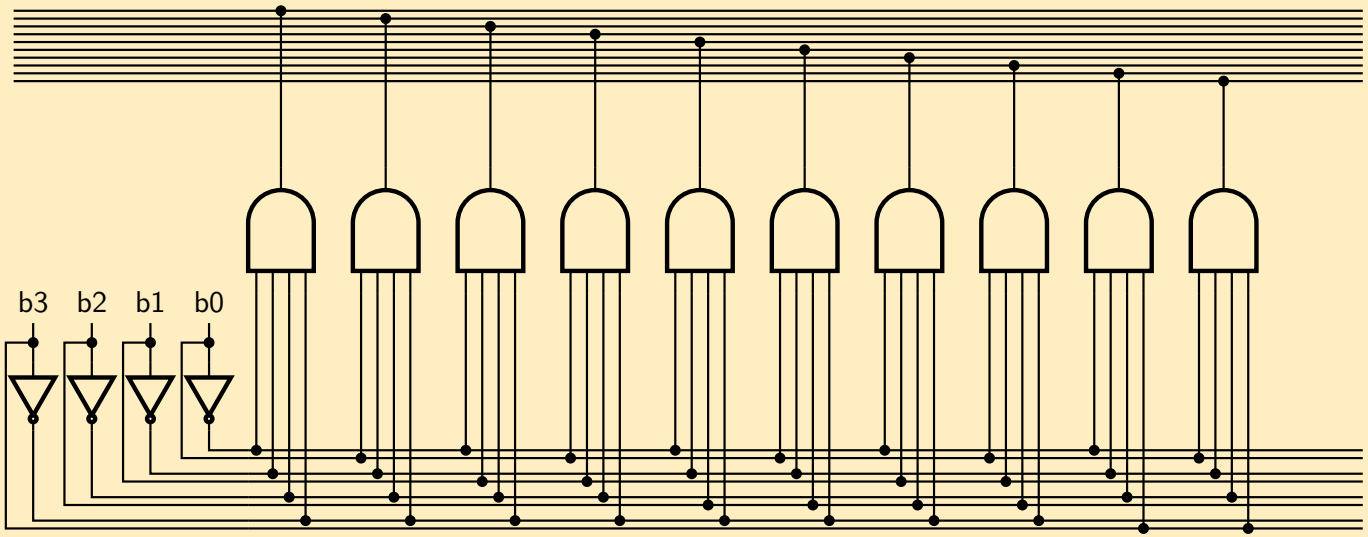


Seven Segment (SOP)

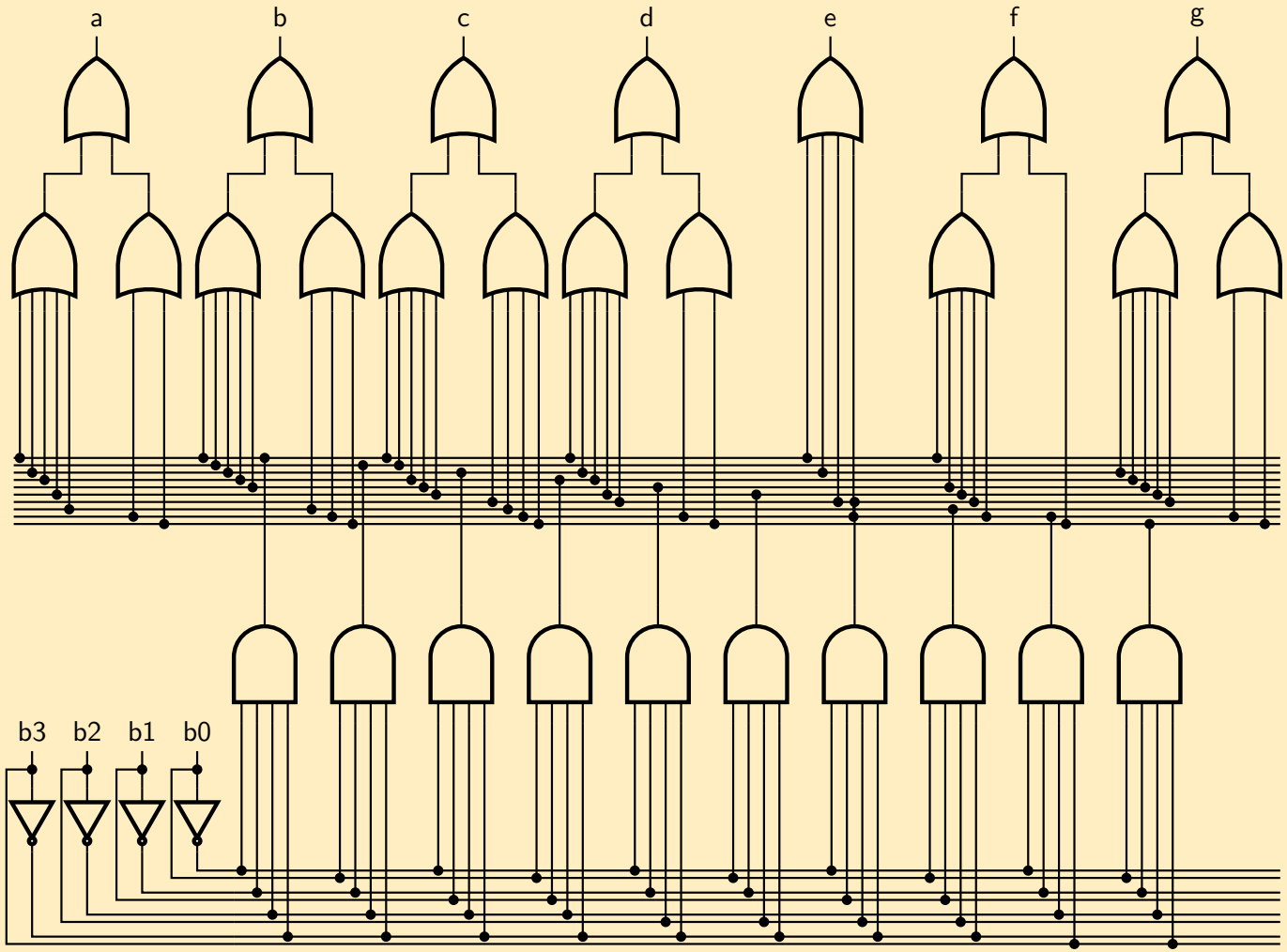


Seven Segment (SOP)

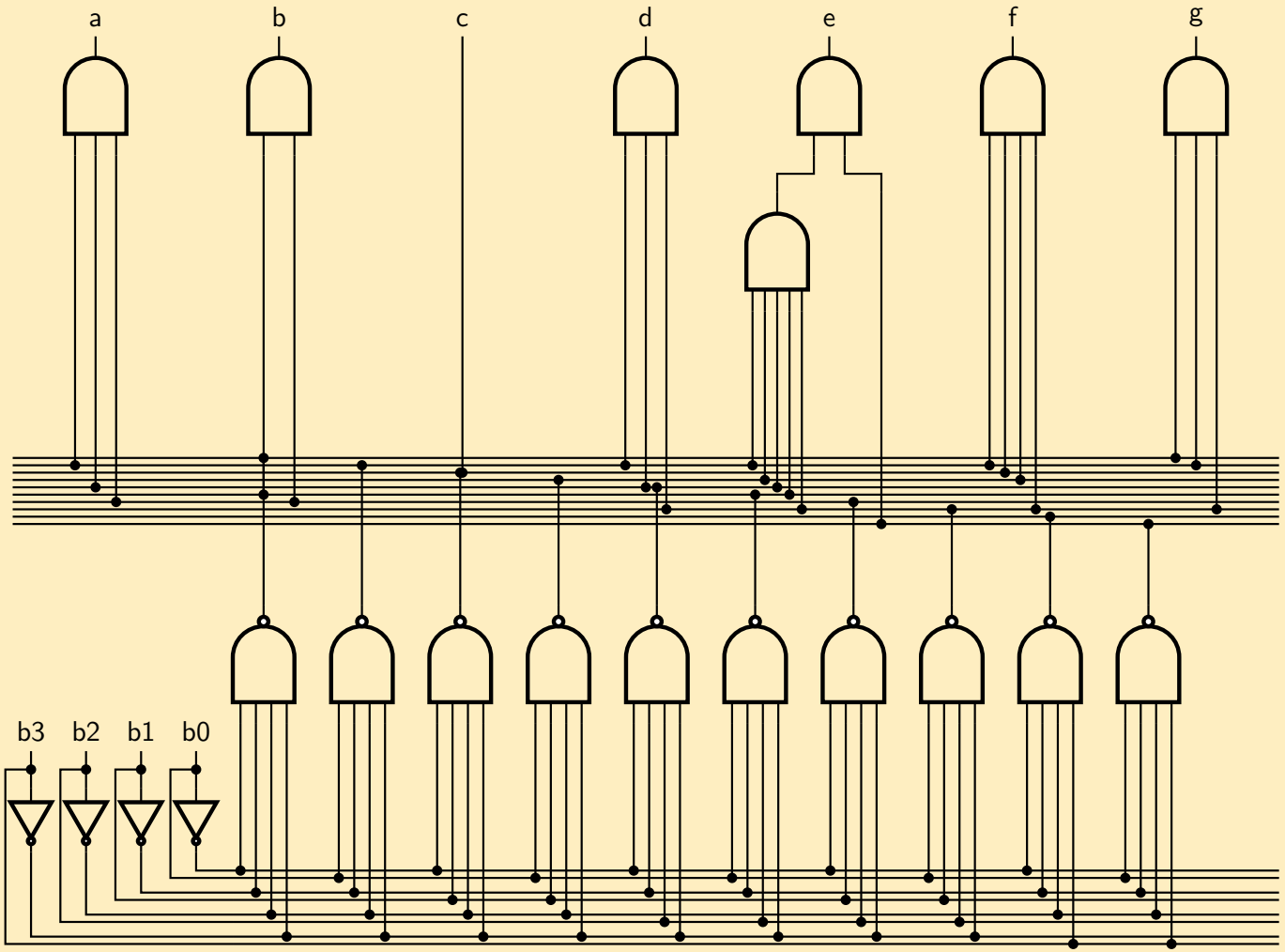




Seven Segment (SOP)

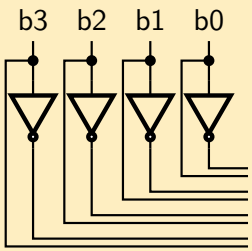


Seven Segment (POS)

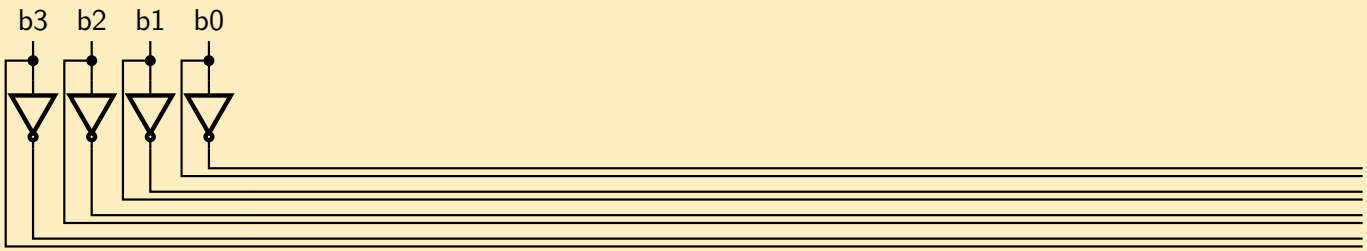


Seven Segment (POS)

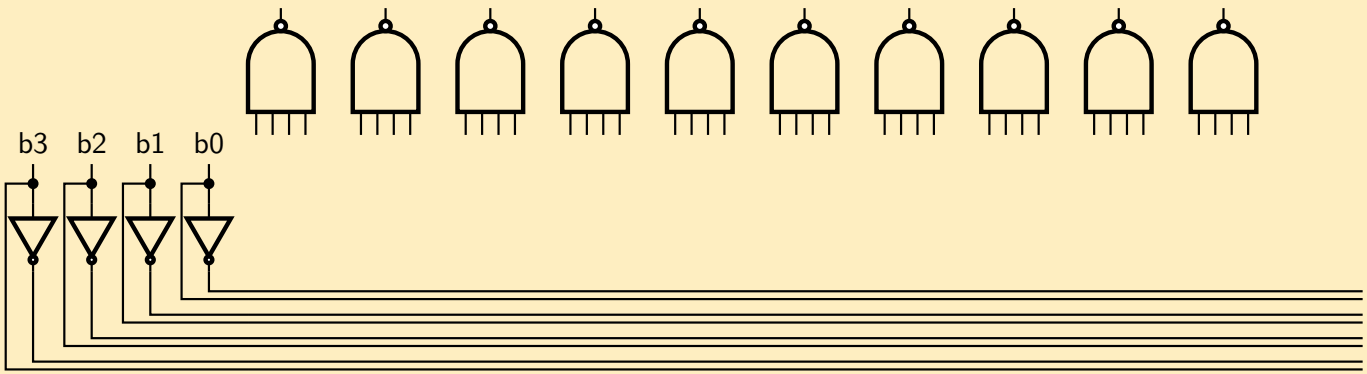
Seven Segment (POS)



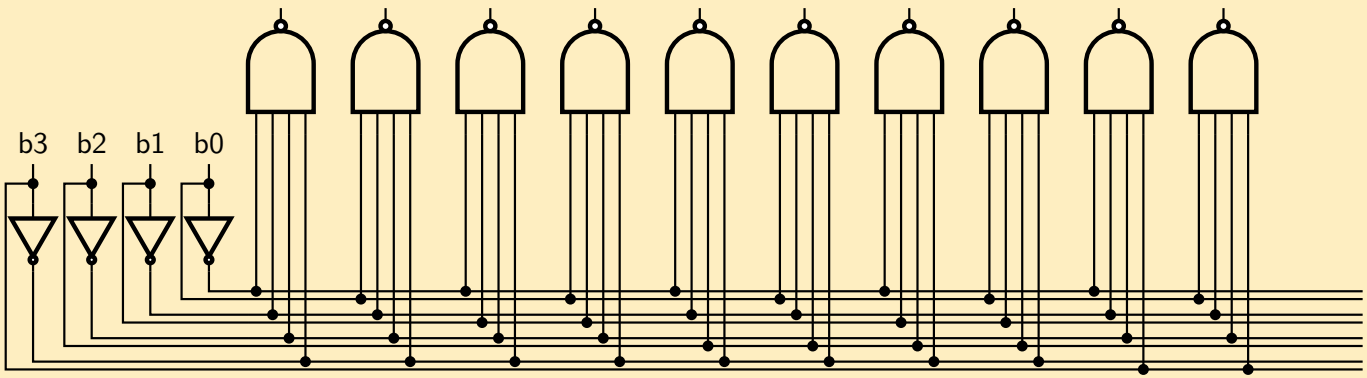
Seven Segment (POS)



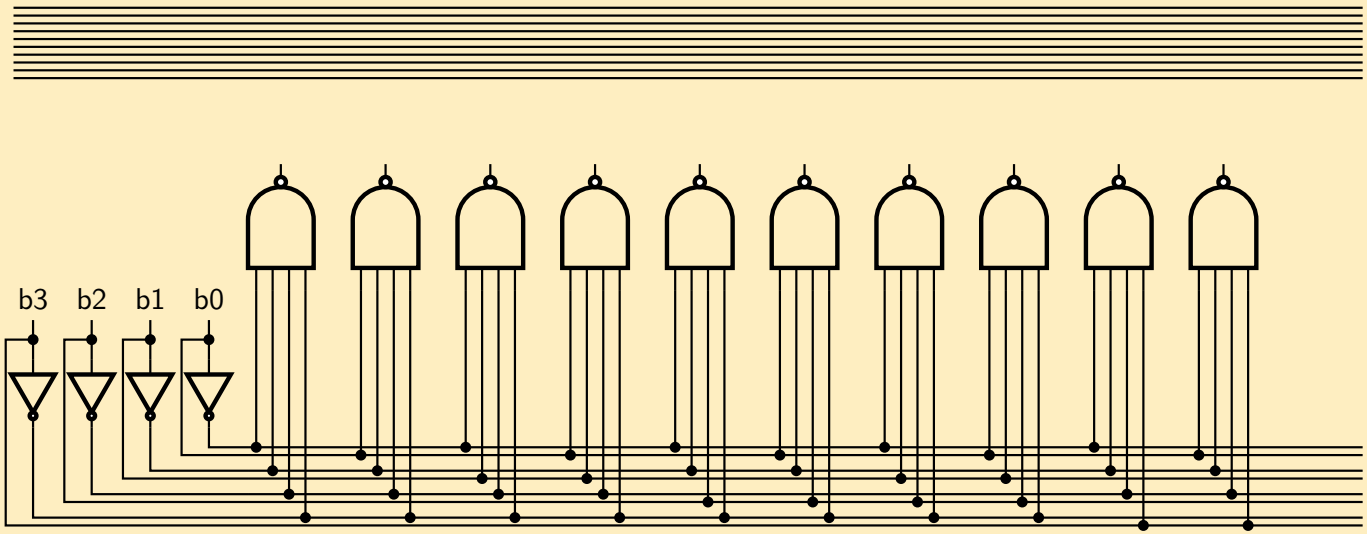
Seven Segment (POS)



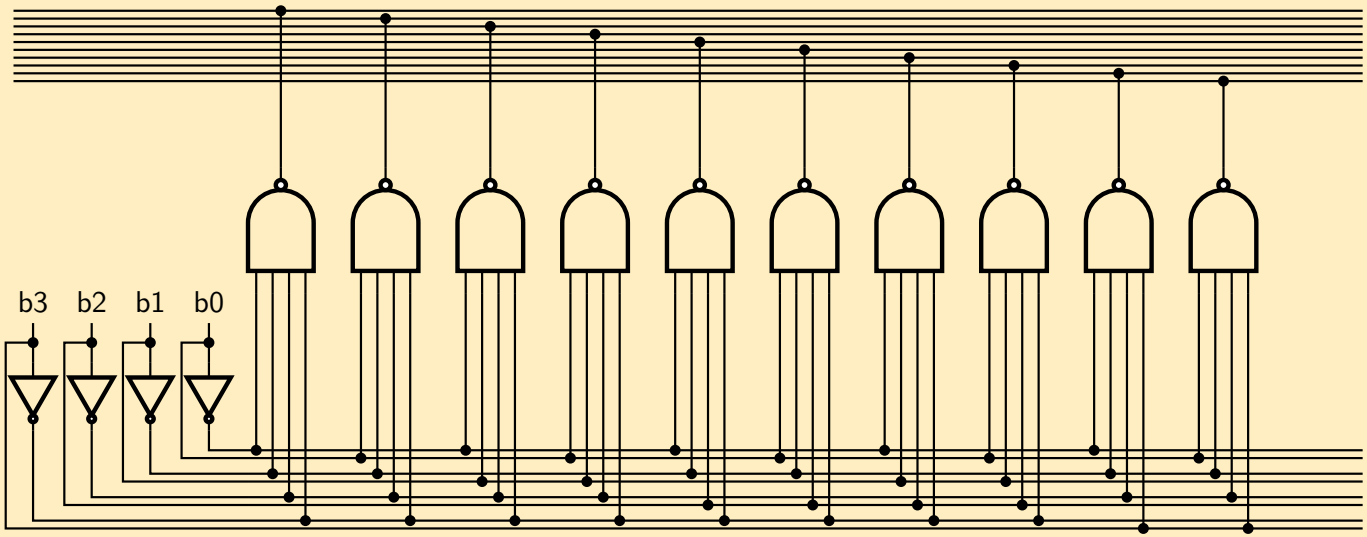
Seven Segment (POS)



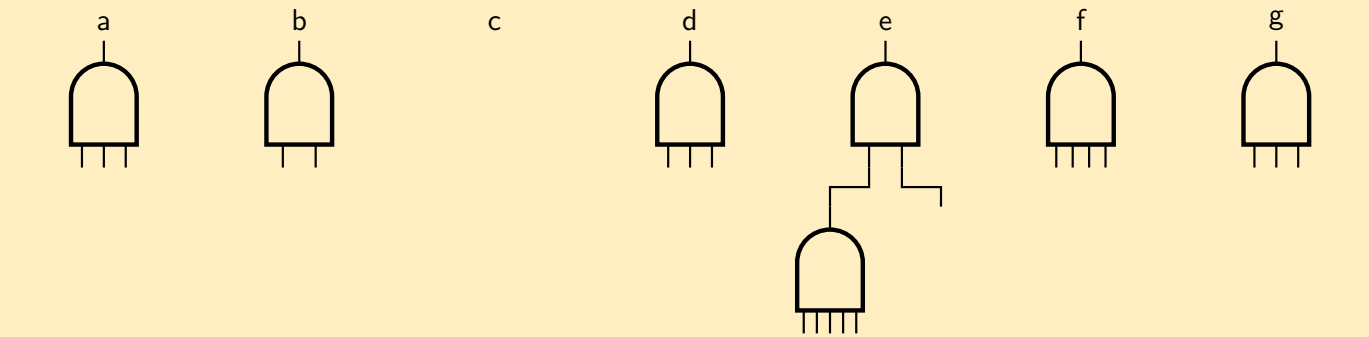
Seven Segment (POS)

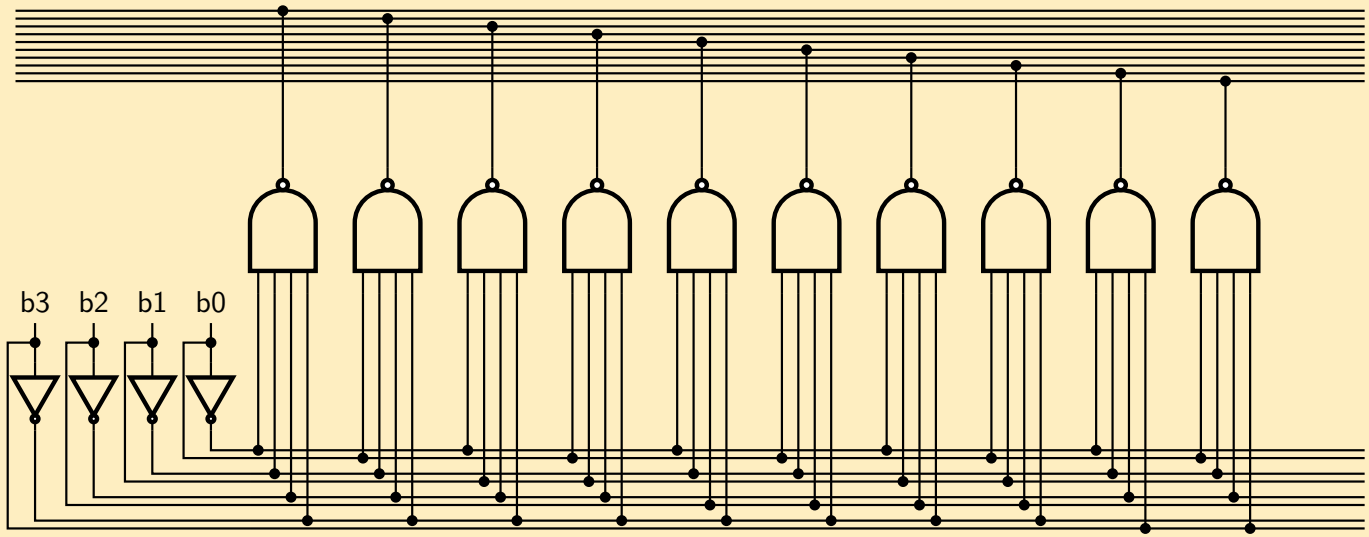


Seven Segment (POS)

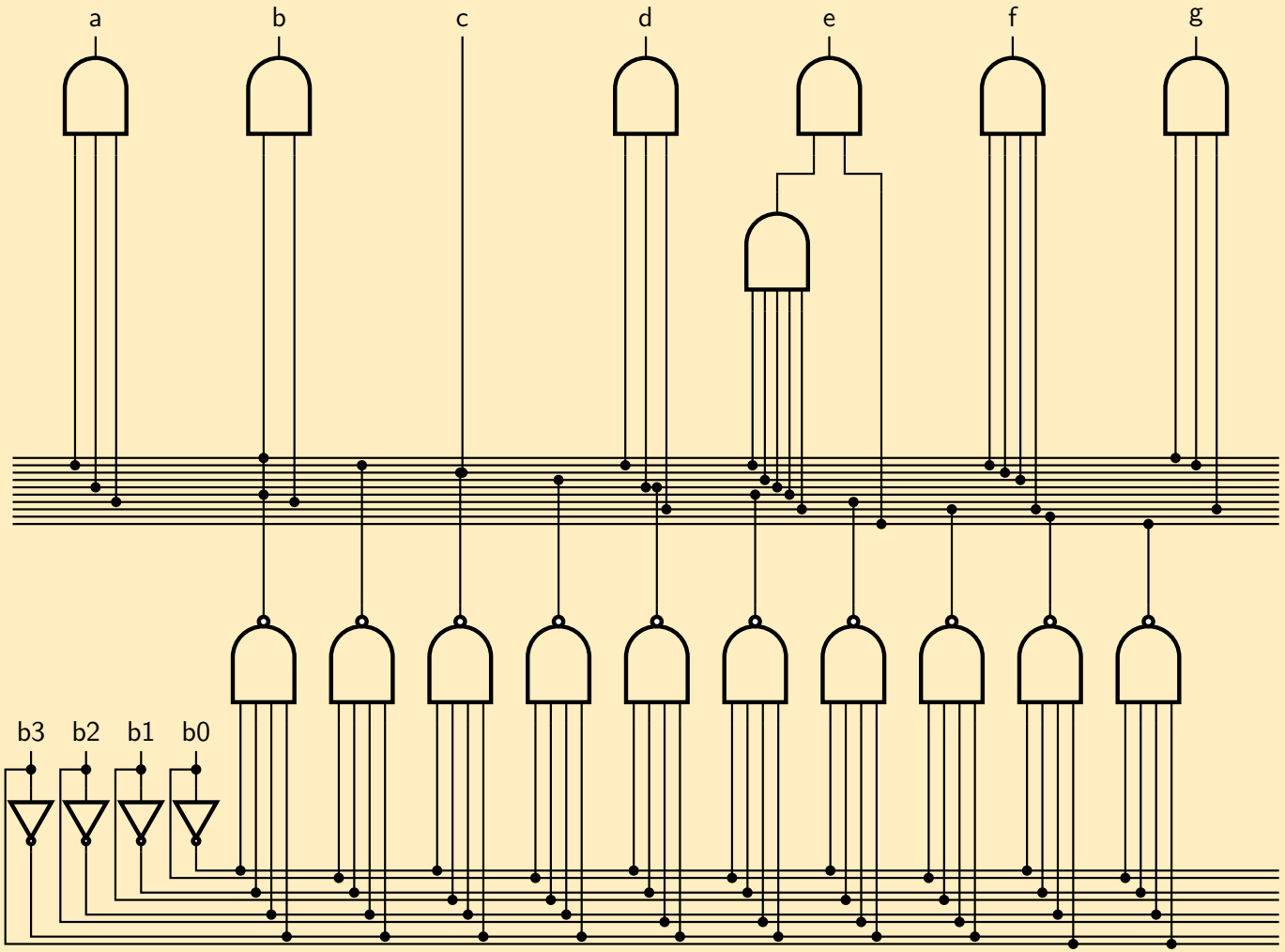


Seven Segment (POS)





Seven Segment (POS)

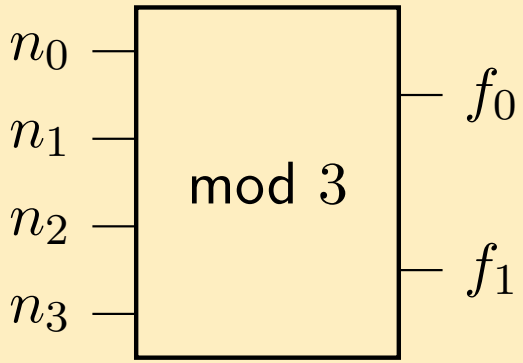


Realizing mod 3

The formuale are from the previous lecture

1. Block diagram
2. Logic circuit

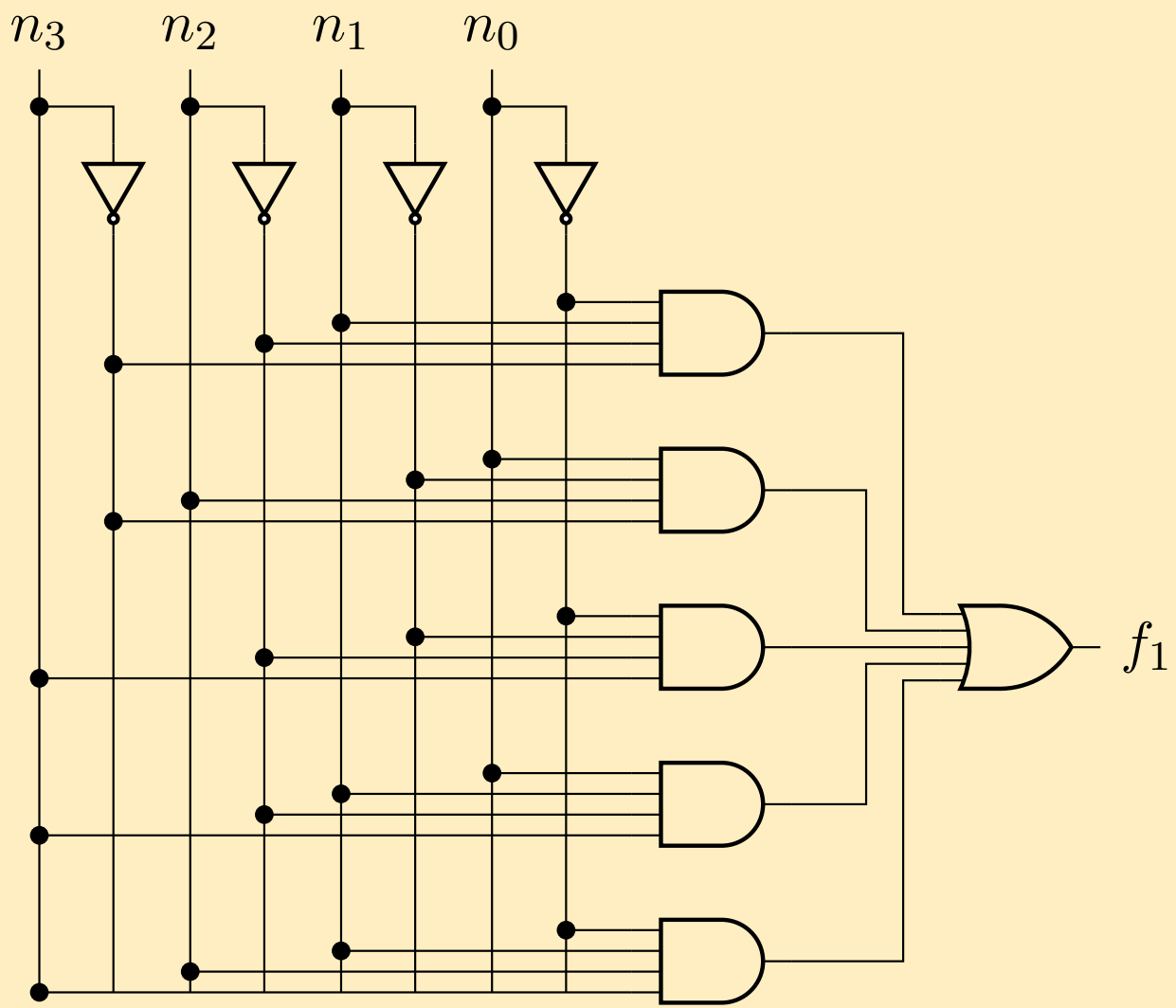
mod 3 Block Diagram



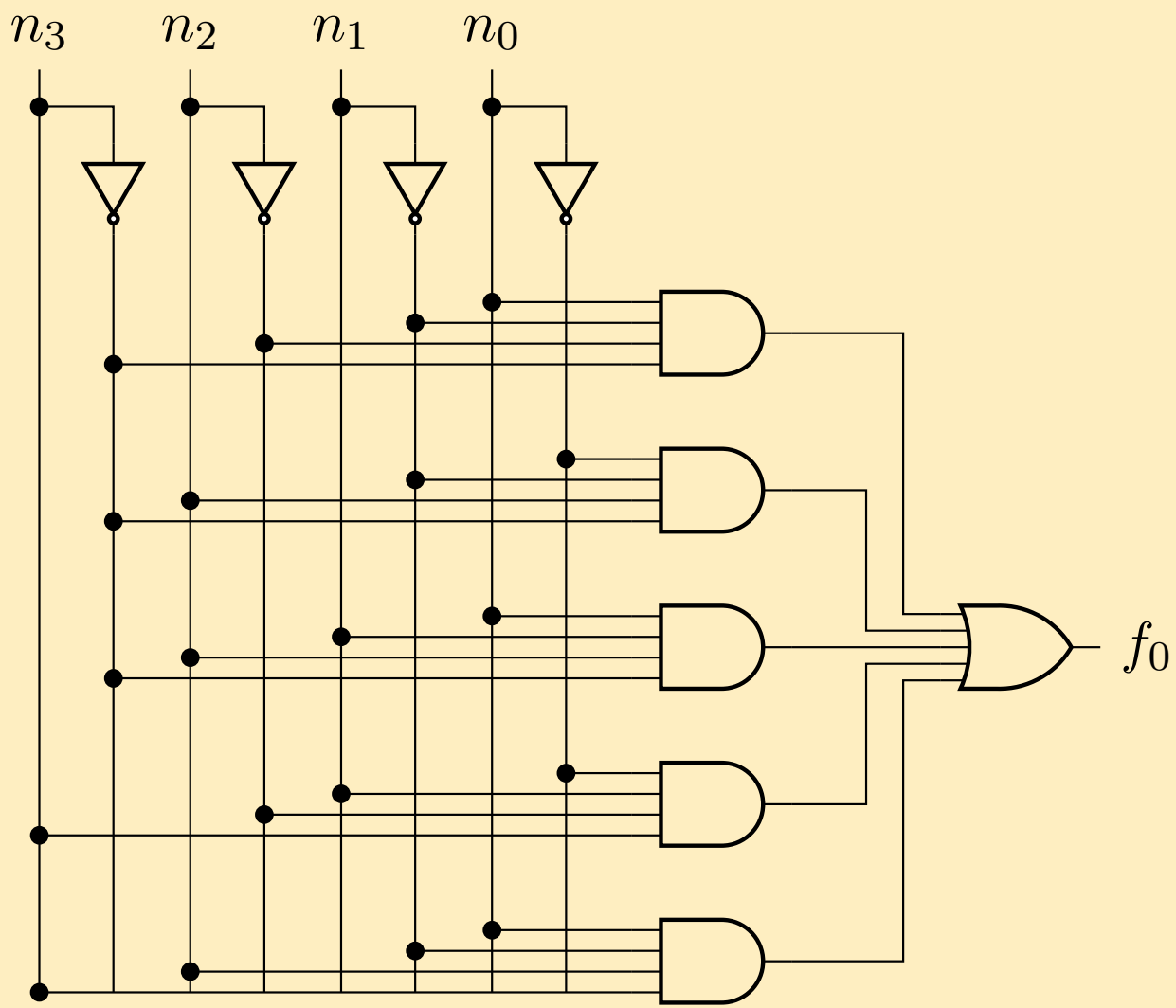
mod 3 Functions

$$
\begin{aligned}
f_1 =& \bar{n}_3\bar{n}_2 n_1\bar{n}_0 + \bar{n}_3 n_2\bar{n}_1 n_0 + n_3\bar{n}_2\bar{n}_1\bar{n}_0 + n_3\bar{n}_2 n_1 n_0 + \\
& n_3 n_2 n_1\bar{n}_0 \\
f_0 =& \bar{n}_3\bar{n}_2\bar{n}_1 n_0 + \bar{n}_3 n_2\bar{n}_1\bar{n}_0 + \bar{n}_3 n_2 n_1 n_0 + n_3\bar{n}_2 n_1\bar{n}_0 + \\
& n_3 n_2\bar{n}_1 n_0
\end{aligned}
$$

mod 3 f_1-Circuit



mod 3 f_0-Circuit



Realizing the Half Adder

The formuale are from the Boolean Algebra lecture

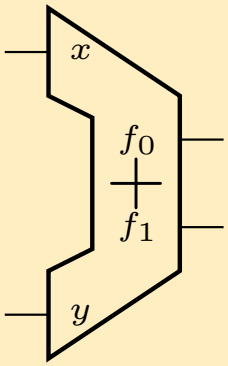
1. Block diagram
2. Logic circuit

Half Adder (block diagram)

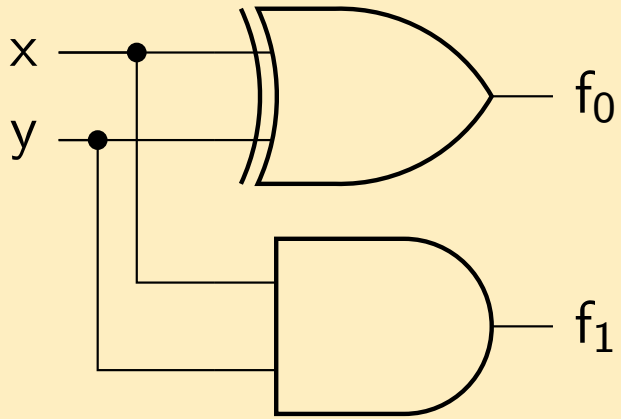
Half Adder (previous lecture)

$$f_1 = x \cdot y$$
$$f_0 = x \oplus y$$

Two bits in, two bits out



Half Adder (circuit)



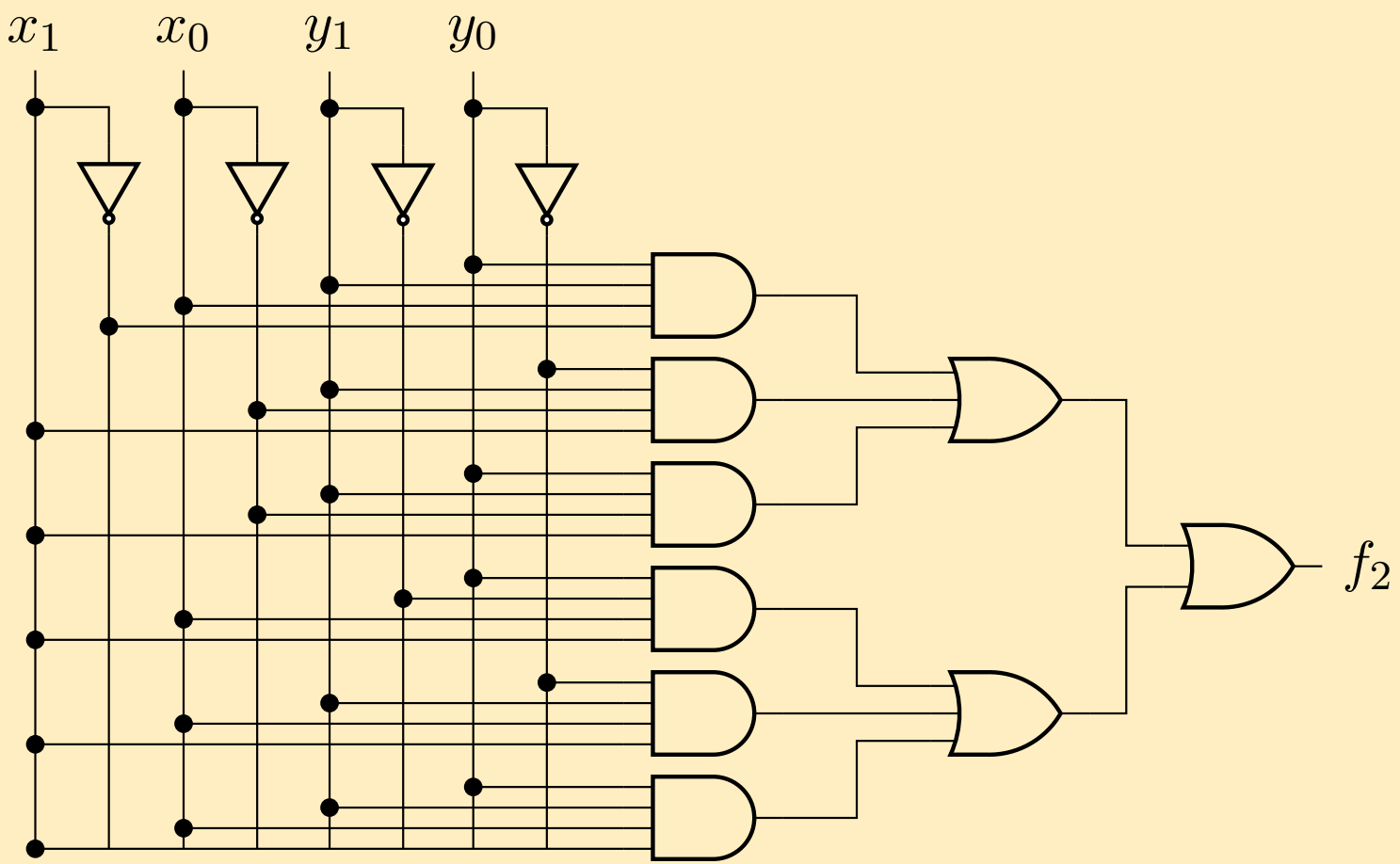
Realizing 2-Bits Adder

(The formulae are from the Boolean Algebra chapter)

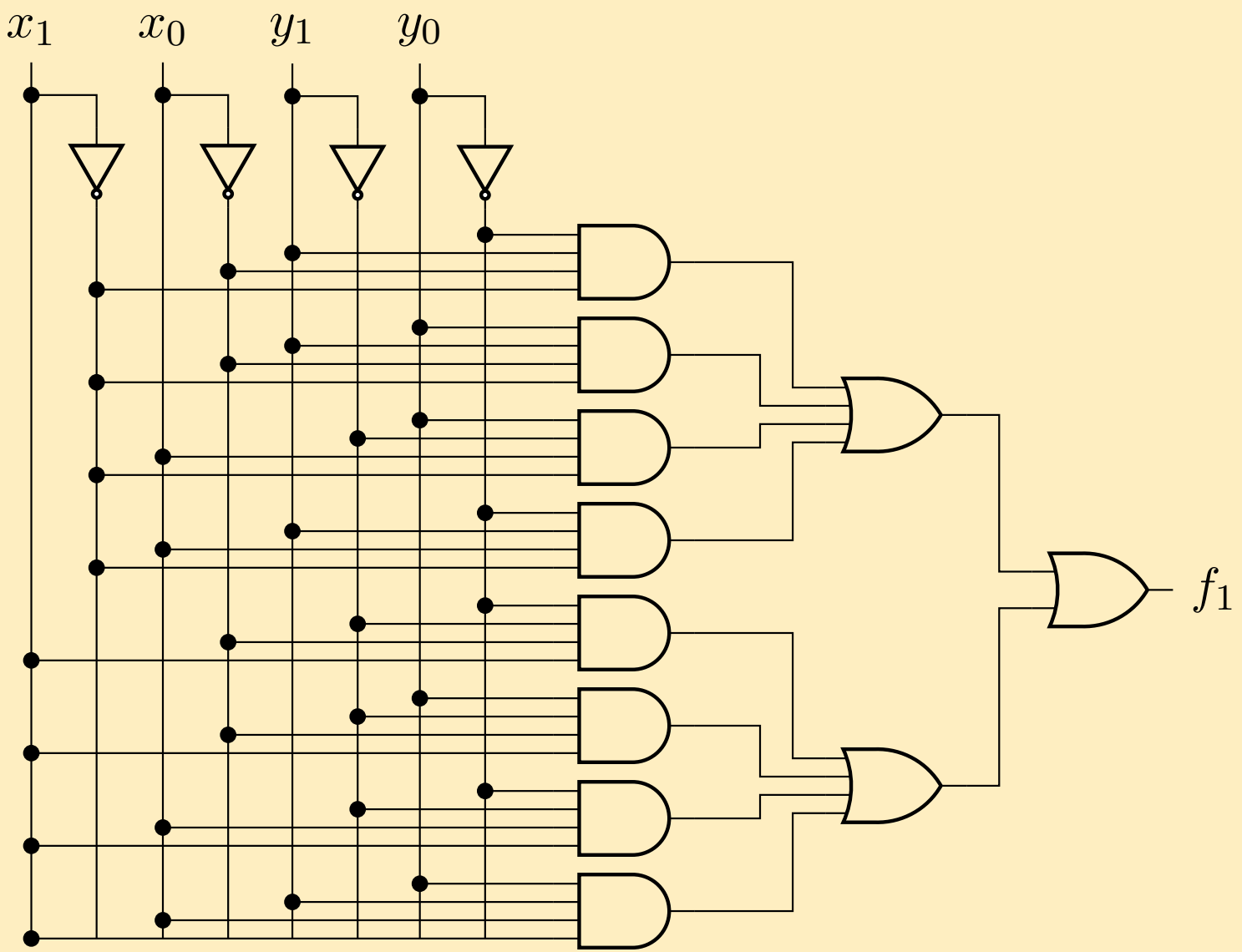
2-Bits Binary Adder (Function)

$$
\begin{aligned}
f_2 =& \bar{x}_1 x_0 y_1 y_0 + x_1 \bar{x}_0 y_1 \bar{y}_0 + x_1 \bar{x}_0 y_1 y_0 + x_1 x_0 \bar{y}_1 y_0 + \\
& x_1 x_0 y_1 \bar{y}_0 + x_1 x_0 y_1 y_0 \\
f_1 =& \bar{x}_1 \bar{x}_0 y_1 \bar{y}_0 + \bar{x}_1 \bar{x}_0 y_1 y_0 + \bar{x}_1 x_0 \bar{y}_1 y_0 + \bar{x}_1 x_0 y_1 \bar{y}_0 + \\
& x_1 \bar{x}_0 \bar{y}_1 \bar{y}_0 + x_1 \bar{x}_0 \bar{y}_1 y_0 + x_1 x_0 \bar{y}_1 \bar{y}_0 + x_1 x_0 y_1 y_0 \\
f_0 =& \bar{x}_1 \bar{x}_0 y_1 \bar{y}_0 + \bar{x}_1 \bar{x}_0 y_1 y_0 + \bar{x}_1 x_0 \bar{y}_1 \bar{y}_0 + \bar{x}_1 x_0 y_1 \bar{y}_0 + \\
& x_1 \bar{x}_0 \bar{y}_1 y_0 + x_1 \bar{x}_0 y_1 y_0 + x_1 x_0 \bar{y}_1 \bar{y}_0 + x_1 x_0 y_1 \bar{y}_0
\end{aligned}
$$

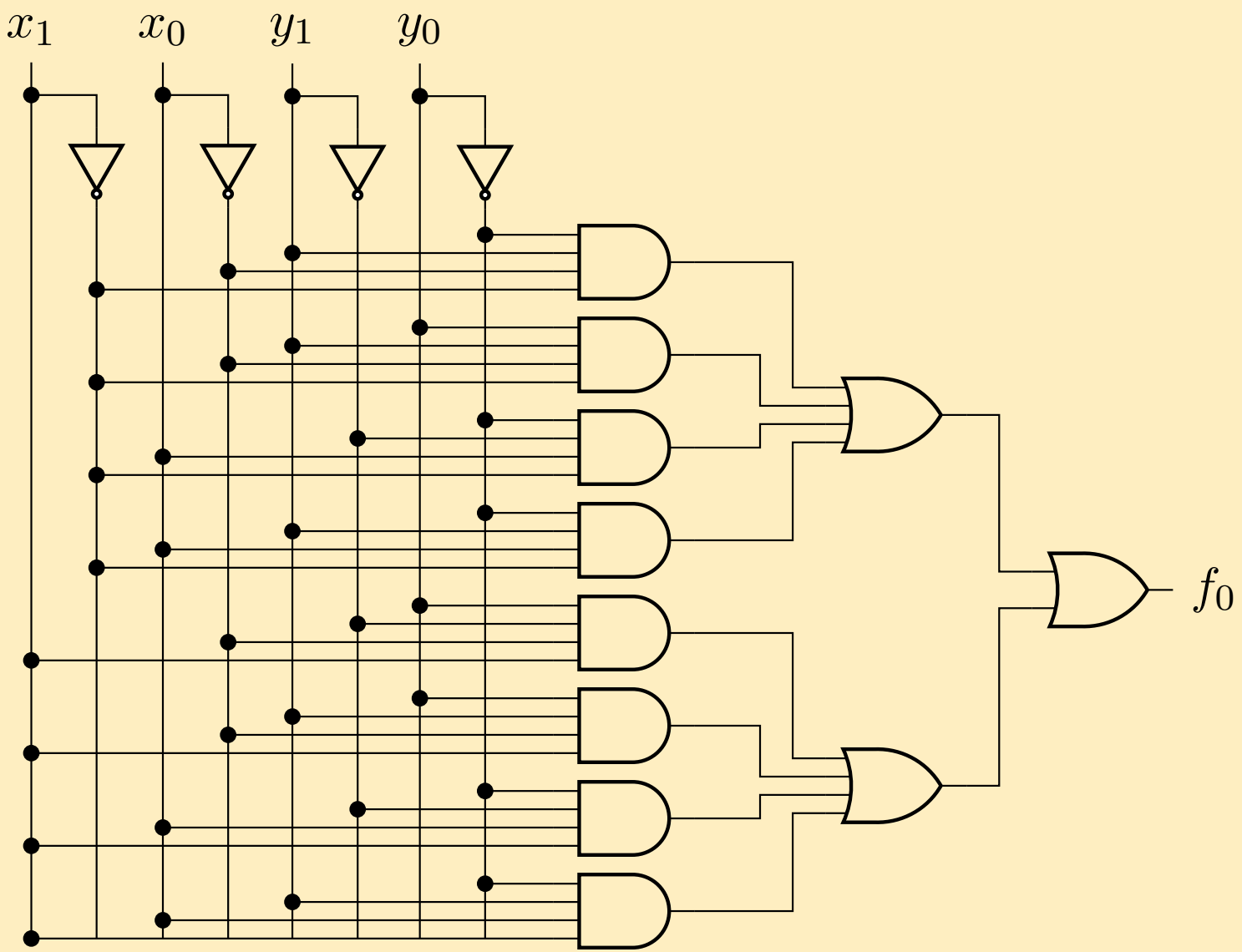
2-Bits Adder f_2-Circuit



2-Bits Adder f_1-Circuit



2-Bits Adder f_0-Circuit

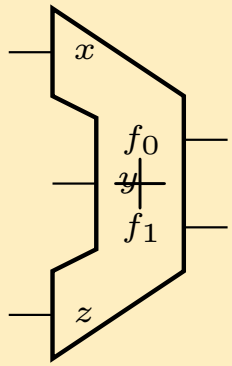


Realizing the Full Adder

Formuale are from the boolean algebras lecture

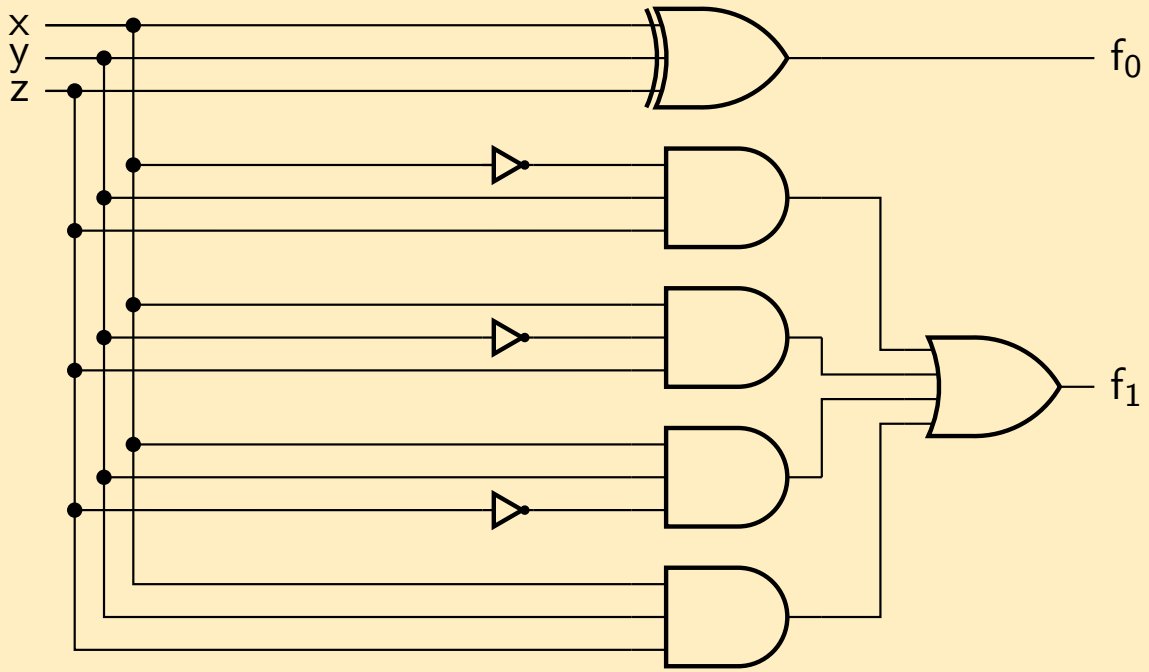
1. Block diagram
2. Logic circuit

Full Adder (block diagram)



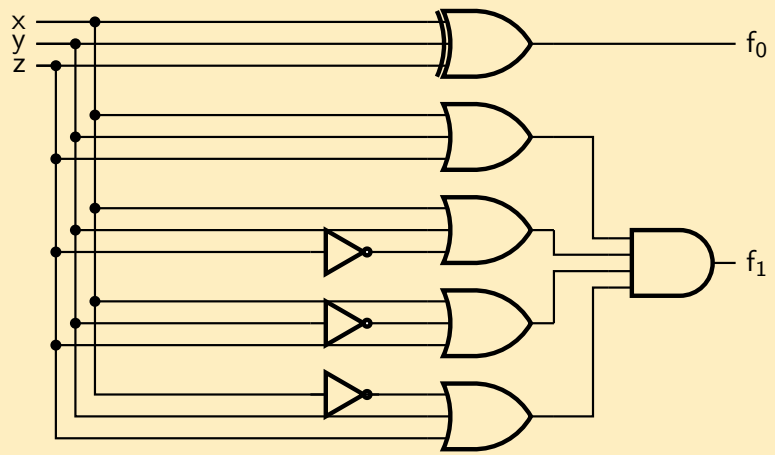
$$
\begin{aligned}
f_0(x,y,z) =& x \oplus y \oplus z \\
f_1(x,y,z) =& \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz = \\
=& (x+y+z)(x+y+\bar{z})(x+\bar{y}+z)(\bar{x}+y+z) = \\
=& yz + xz + xy
\end{aligned}
$$

Full adder (f_1 SOP)



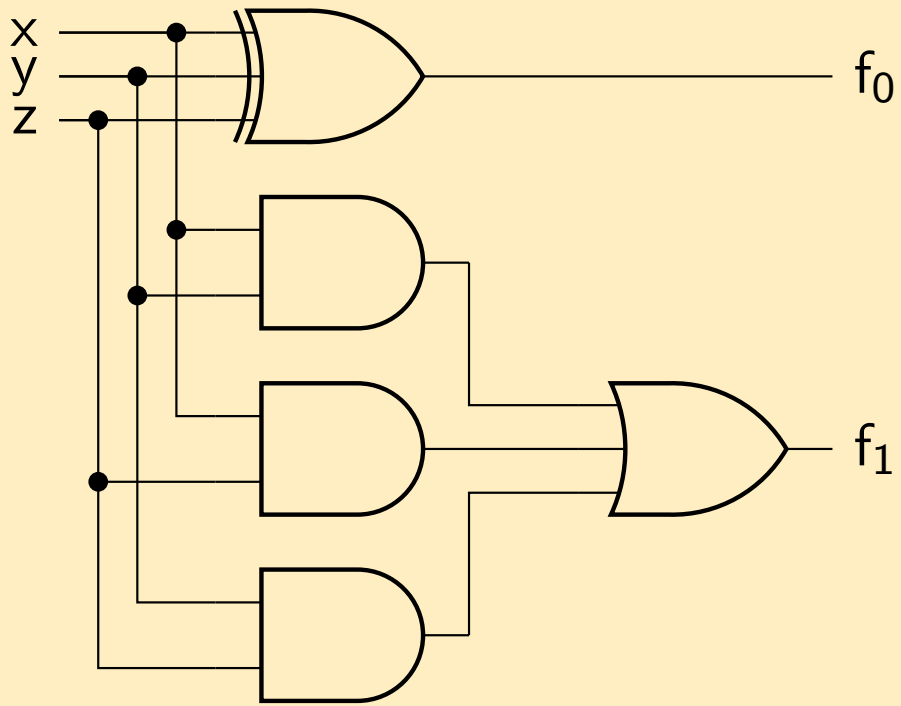
$$f_0(x,y,z) = x \oplus y \oplus z$$
$$f_1(x,y,z) = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz$$

Full adder (f_1 POS circuits)



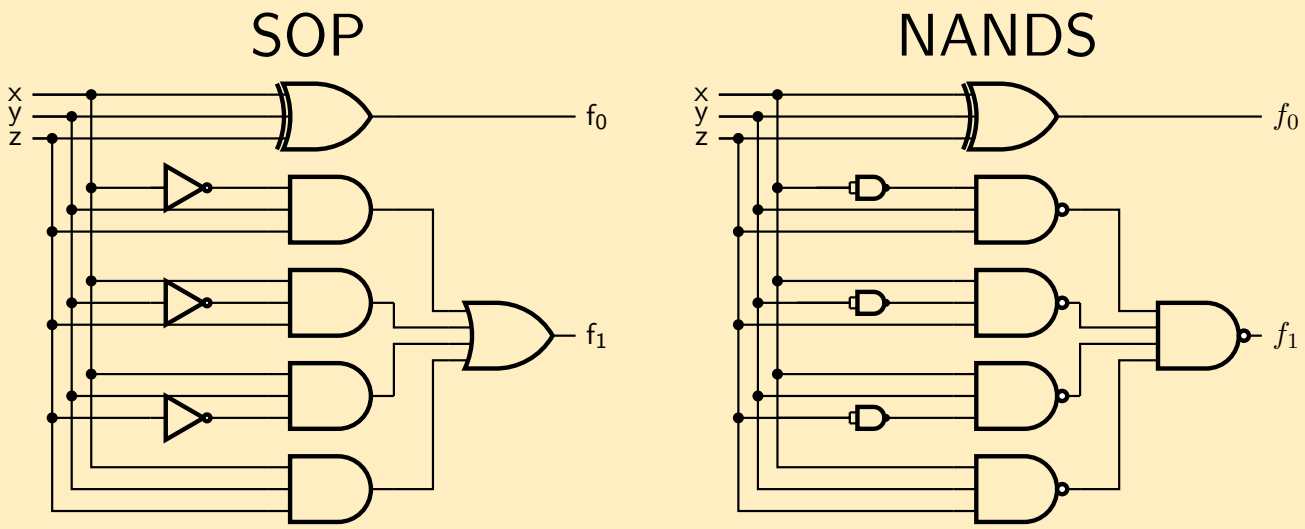
$$f_0(x,y,z) = x \oplus y \oplus z$$
$$f_1(x,y,z) = (x+y+z)(x+y+\bar{z})(x+\bar{y}+z)(\bar{x}+y+z)$$

Full adder (simplified circuit)

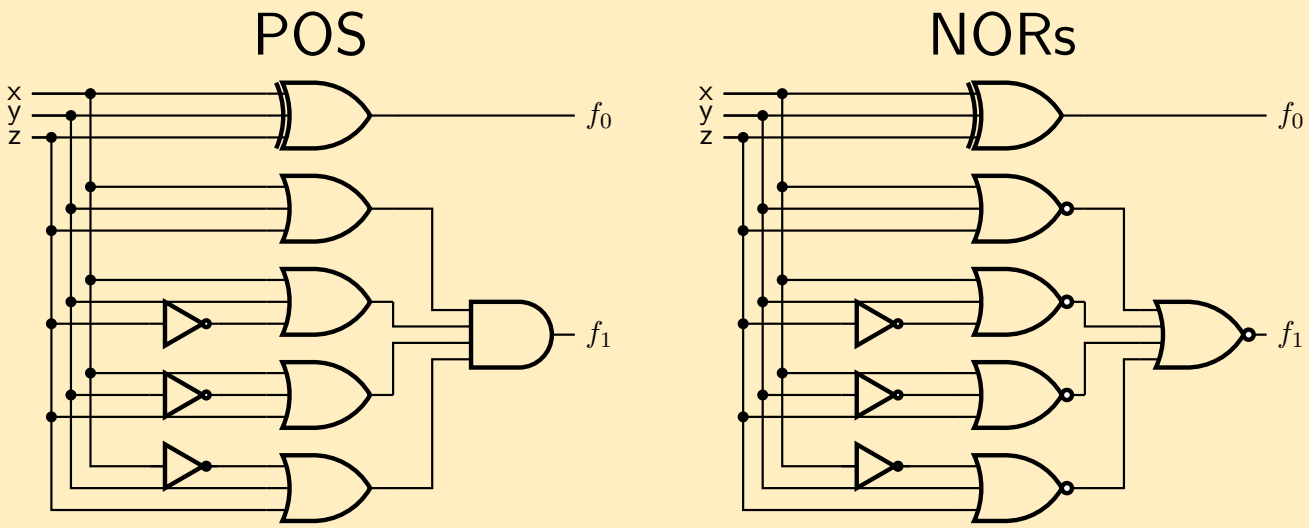


$$f_0(x,y,z) = x \oplus y \oplus z$$
$$f_1(x,y,z) = yz + xz + xy$$

Full adder (nands)



Full adder (nors)



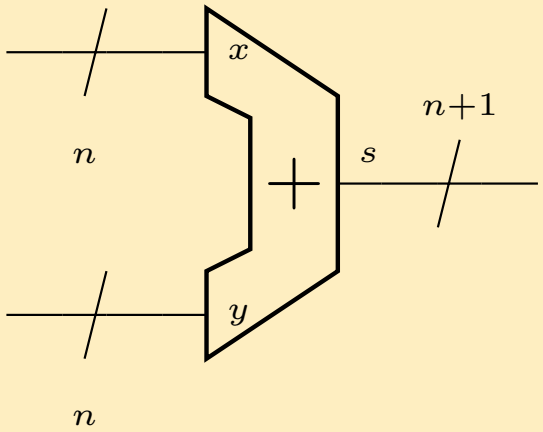
$$f_0(x,y,z) = x \oplus y \oplus z$$
$$f_1(x,y,z) = (x+y+z)(x+y+\bar{z})(x+\bar{y}+z)(\bar{x}+y+z)$$

Realizing the Binary Adder

The formuale are from the previous lecture

1. Block diagram
2. Logic circuit

Binary adder (block diagram)



Binary Adder (circuit)