
Chapter 2

Boolean Algebra and Logic Gates

พีชคณิตแบบบูล ตั้งชื่อตามนักคณิตศาสตร์ (George Boole)
ลอจิกเกต เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ประมวลผลข้อมูลที่เป็นดิจิทัล (0/1)

Binary Logic

Table 1.8
Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

ໃຜ 0 ແນ False ແລະໃຜ 1 ແນ TRUE

Binary operator: ໃຜ $\bullet$ ແນ AND ແລະໃຜ $+$ ແນ OR

Unary operator: ໃຜ $'$ ແນ NOT

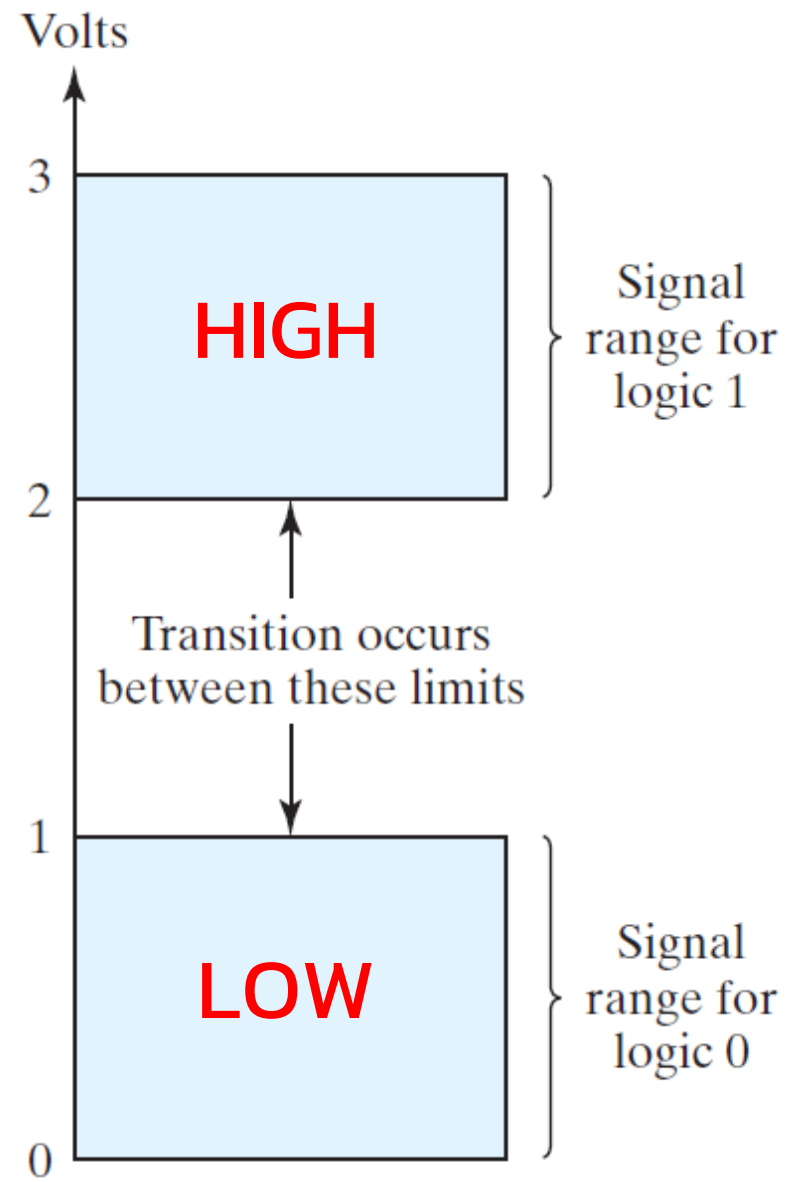
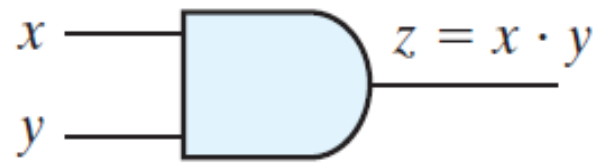
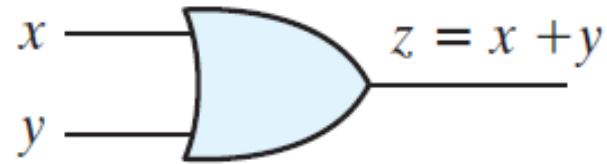


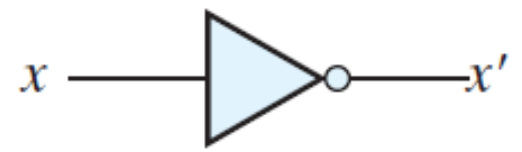
FIGURE 1.3
Signal levels for binary logic values



(a) Two-input AND gate



(b) Two-input OR gate



(c) NOT gate or inverter

FIGURE 1.4

Symbols for digital logic circuits

Timing Diagram แกนนอนคือเวลา

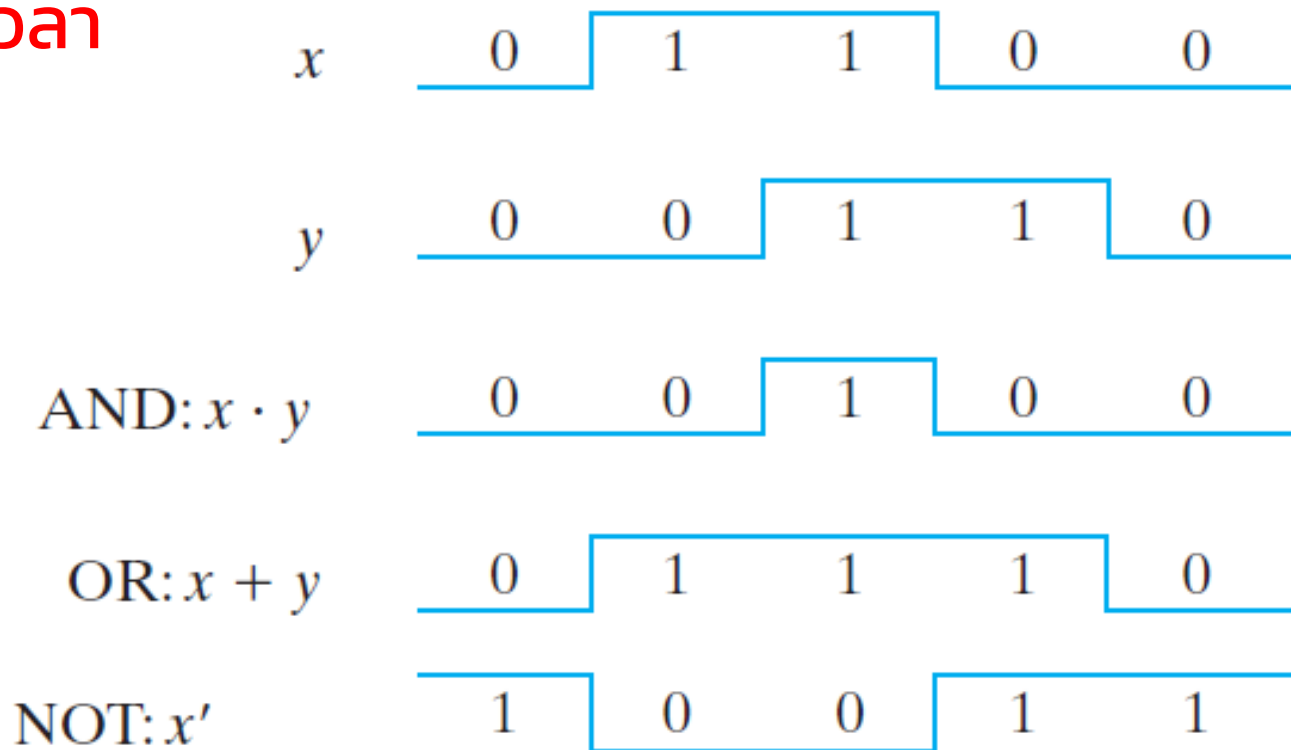
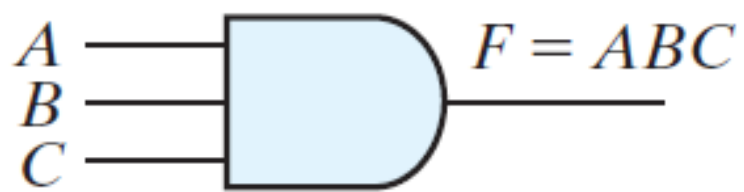
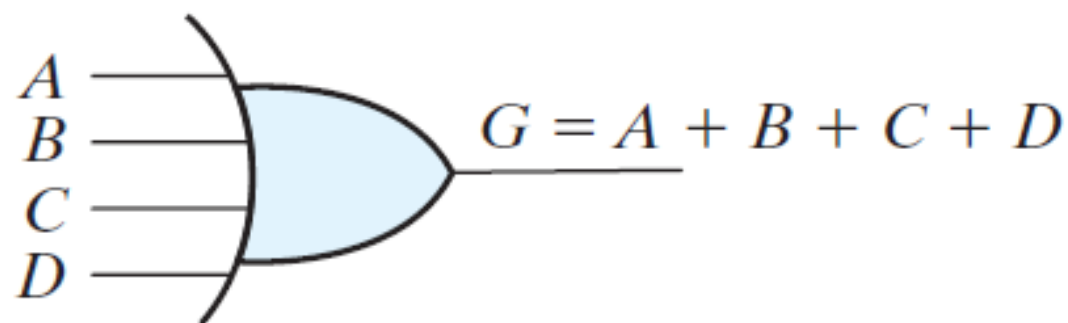


FIGURE 1.5

Input-output signals for gates



(a) Three-input AND gate



(b) Four-input OR gate

FIGURE 1.6

Gates with multiple inputs

Distributive Law

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

$y + z$	$x \cdot (y + z)$
0	0
1	0
1	0
1	0
0	0
1	1
1	1
1	1

$x \cdot y$	$x \cdot z$	$(x \cdot y) + (x \cdot z)$
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
1	0	1
1	1	1

Table 2.1*Postulates and Theorems of Boolean Algebra*

Postulate 2	(a)	$x + 0 = x$	(b)	$x \cdot 1 = x$
Postulate 5	(a)	$x + x' = 1$	(b)	$x \cdot x' = 0$
Theorem 1	(a)	$x + x = x$	(b)	$x \cdot x = x$
Theorem 2	(a)	$x + 1 = 1$	(b)	$x \cdot 0 = 0$
Theorem 3, involution		$(x')' = x$		
Postulate 3, commutative	(a)	$x + y = y + x$	(b)	$xy = yx$
Theorem 4, associative	(a)	$x + (y + z) = (x + y) + z$	(b)	$x(yz) = (xy)z$
Postulate 4, distributive	(a)	$x(y + z) = xy + xz$	(b)	$x + yz = (x + y)(x + z)$
Theorem 5, DeMorgan	(a)	$(x + y)' = x'y'$	(b)	$(xy)' = x' + y'$
Theorem 6, absorption	(a)	$x + xy = x$	(b)	$x(x + y) = x$

x ต้องเป็นจริง หรือ
yz ต้องเป็นจริงทั้งคู่

สัจพจน์ (postulate) - ข้อความเบื้องต้นที่ยอมรับกันว่าจริงโดยไม่ต้องพิสูจน์

Boolean Functions

$$F_1 = x + y'z$$

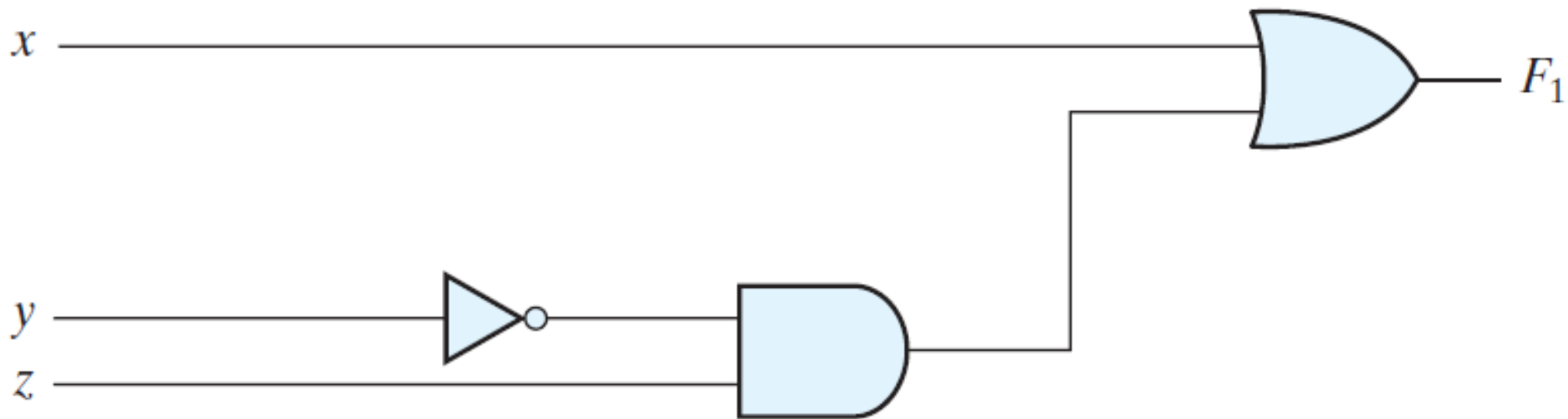
$$F_2 = x'y'z + x'yz + xy'$$

Table 2.2

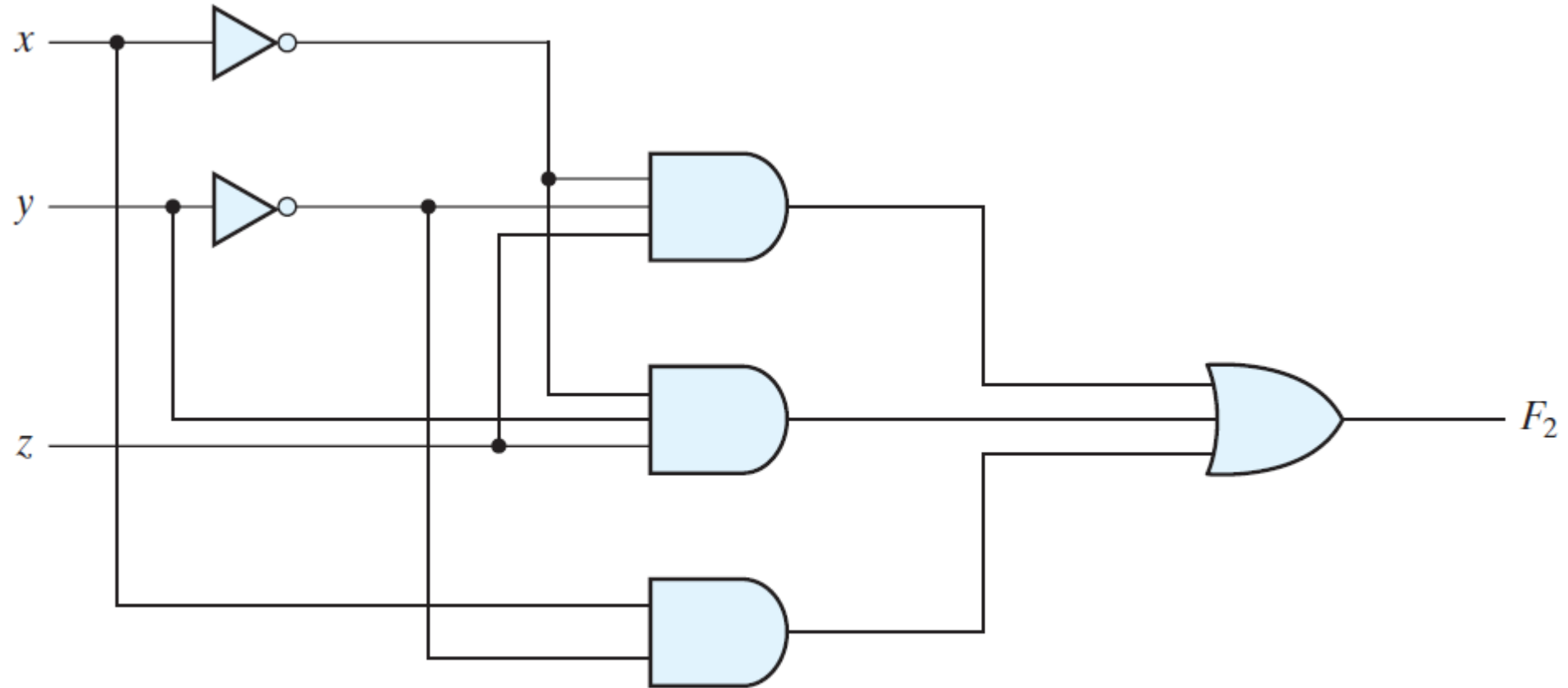
Truth Tables for F_1 and F_2

x	y	z	F_1	F_2
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

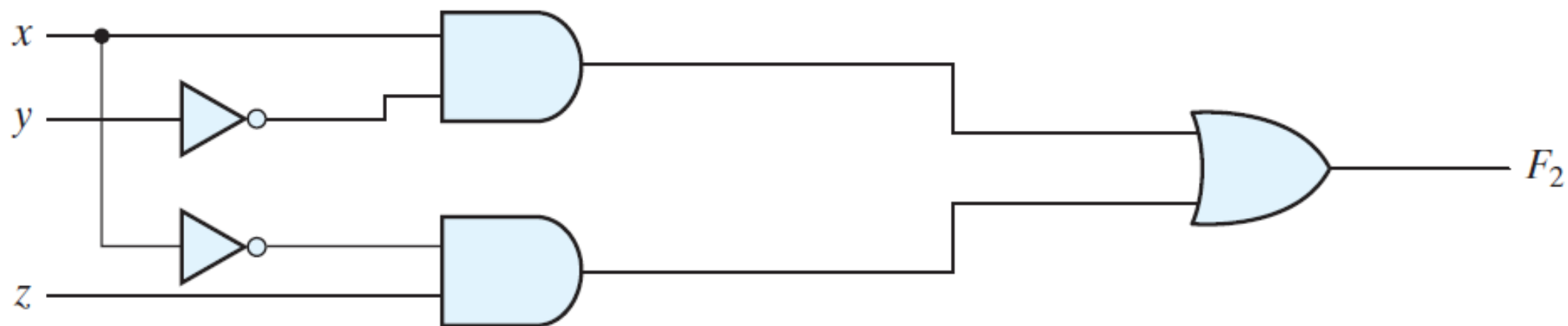
$$F_1 = x + y'z$$



$$F_2 = x'y'z + x'yz + xy'$$



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



(b) $F_2 = xy' + x'z$

Simplify the following Boolean functions to a minimum number of literals.

1. $x(x' + y) = xx' + xy = 0 + xy = xy.$

2. $x + x'y = (x + x')(x + y) = 1(x + y) = x + y.$

3. $(x + y)(x + y') = x + xy + xy' + yy' = x(1 + y + y') = x.$

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

5. $(x + y)(x' + z)(y + z) = (x + y)(x' + z),$ by duality from function 4.

พิสูจน์ยังงงว่าใช้ duality จากข้อ 4 (เปลี่ยน $\times$ เป็น $+$ และเปลี่ยน $+$ เป็น $\times$) แล้วจะได้คำตอบถูกต้อง ใส่ not ทั้งสองข้างของข้อ 4 แล้วใช้ DeMorgan Law เปลี่ยน x, y, z เป็น x', y', z' และเปลี่ยน x', y', z' เป็น x, y, z (ตัวแปร x, y, z ในข้อ 4 และ 5 เป็นคนละตัวแปรกัน) x กับ x' แค่มองว่าต้องมีค่าตรงข้ามกัน จะสลับ x กับ x' ก็ได้

Complement of a function

$$\begin{aligned}(A + B + C)' &= (A + x)' && \text{let } B + C = x \\ &= A'x' && \text{by theorem 5(a) (DeMorgan)} \\ &= A'(B + C)' && \text{substitute } B + C = x \\ &= A'(B'C') && \text{by theorem 5(a) (DeMorgan)} \\ &= A'B'C' && \text{by theorem 4(b) (associative)}\end{aligned}$$

$$(A + B + C + D + \cdots + F)' = A'B'C'D' \dots F'$$

$$(ABCD \dots F)' = A' + B' + C' + D' + \cdots + F'$$

EXAMPLE 2.2

Find the complement of the functions $F_1 = x'yz' + x'y'z$ and $F_2 = x(y'z' + yz)$. By applying DeMorgan's theorems as many times as necessary, the complements are obtained as follows:

$$\begin{aligned}F_1' &= (x'yz' + x'y'z)' = (x'yz')'(x'y'z)' = (x + y' + z)(x + y + z') \\F_2' &= [x(y'z' + yz)]' = x' + (y'z' + yz)' = x' + (y'z')'(yz)' \\&= x' + (y + z)(y' + z') \\&= x' + yz' + y'z\end{aligned}$$

A simpler procedure for deriving the complement of a function is to take the dual of the function and complement each literal. This method follows from the generalized forms of DeMorgan's theorems. Remember that the dual of a function is obtained from the interchange of AND and OR operators and 1's and 0's.

EXAMPLE 2.3

Find the complement of the functions F_1 and F_2 of Example 2.2 by taking their duals and complementing each literal.

เปลี่ยน AND เป็น OR
เปลี่ยน OR เป็น AND

1. $F_1 = x'yz' + x'y'z.$

The dual of F_1 is $(x' + y + z')(x' + y' + z).$

Complement each literal: $(x + y' + z)(x + y + z') = F_1'.$

2. $F_2 = x(y'z' + yz).$

The dual of F_2 is $x + (y' + z')(y + z).$

Complement each literal: $x' + (y + z)(y' + z') = F_2'.$

ใช้ DeMorgan ก็ได้ผลลัพธ์เหมือนกัน



Minterms and Maxterms

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

$$m'_j = M_j$$

Sum of Minterms

Table 2.4
Functions of Three Variables

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

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

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

Product of Maxterms

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

หา complement ก่อน ดูจากช่องที่เป็น 0 ใน Truth Table

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

↙ ตก $M_3 = (x + y' + z')$

$$\begin{aligned} f_1 &= (x + y + z)(x + y' + z)(x' + y + z')(x' + y' + z) \\ &= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6 \end{aligned}$$

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

Boolean functions វិធីសាស្ត្រប្រើប្រាស់ sum of minterms ໖

$$F = A + B'C$$

Table 2.5

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

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

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

$$= m_1 + m_4 + m_5 + m_6 + m_7$$

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

Boolean functions ព្រំព័ទ្ធរបស់ product of maxterms ផ្គុំ

EXAMPLE 2.5

Express the Boolean function $F = xy + x'z$ as a product of maxterms. First, convert the function into OR terms by using the distributive law:

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

The function has three variables: x , y , and z . Each OR term is missing one variable; therefore,

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

Combining all the terms and removing those which appear more than once, we finally obtain

$$\begin{aligned} F &= (x + y + z)(x + y' + z)(x' + y + z)(x' + y + z') \\ &= M_0 M_2 M_4 M_5 \end{aligned} \quad \boxed{F(x, y, z) = \Pi(0, 2, 4, 5)}$$

Sum of minterms
and
product of maxterms
are **"canonical forms."**
เป็นที่ยอมรับ ถูกต้องตามบัญญัติ (ศาสนา)

Boolean functions ใดใดสามารถเขียนให้อยู่ในรูปของ sum of minterms หรือ product of maxterms ได้

Conversion between Canonical Forms

The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function. This is because the original function is expressed by those minterms which make the function equal to 1, whereas its complement is a 1 for those minterms for which the function is a 0. As an example, consider the function

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

This function has a complement that can be expressed as

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

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

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

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

$$m'_j = M_j$$

That is, the **maxterm with subscript j is a complement of the minterm with the same subscript j and vice versa.**

Table 2.6

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

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

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

Inverse 2 ครั้ง

- เลือก $F = 0$ แทน $F = 1$
- ใช้ De Morgan Law

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

Standard Forms

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

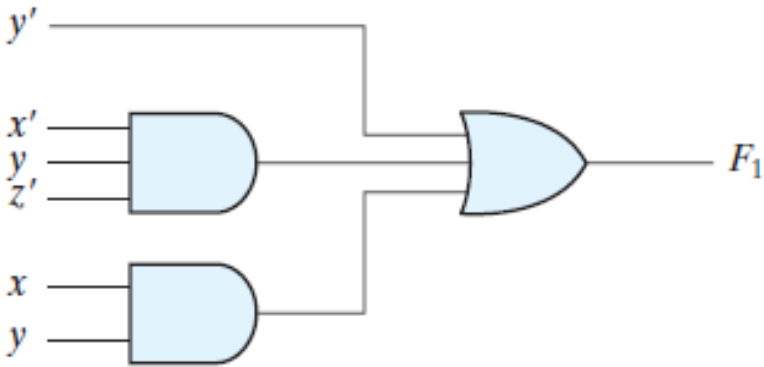
$$F_2 = x(y' + z)(x' + y + z')$$

ในแต่ละ term ไม่ต้องเขียนตัวแปรทั้งหมด

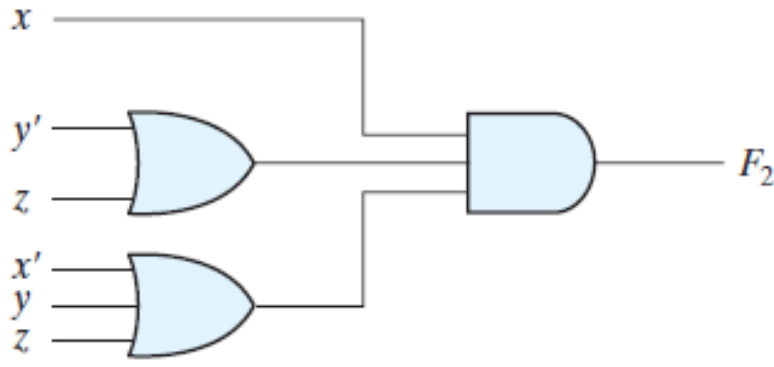
$$F_3 = AB + C(D + E)$$

Non-standard form

Two-level implementation is less delayed.

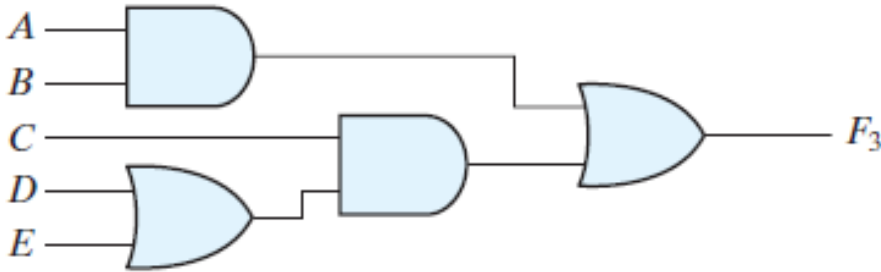


(a) Sum of Products Non-canonical form!

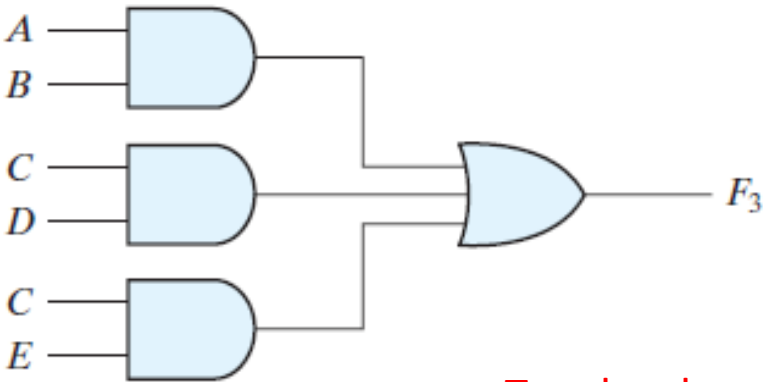


(b) Product of Sums Non-canonical form!

FIGURE 2.3
Two-level implementation



(a) $AB + C(D + E)$ Three-level implementation



(b) $AB + CD + CE$ Two-level implementation

FIGURE 2.4
Three- and two-level implementation

Logic Operations

Table 2.7

Truth Tables for the 16 Functions of Two Binary Variables

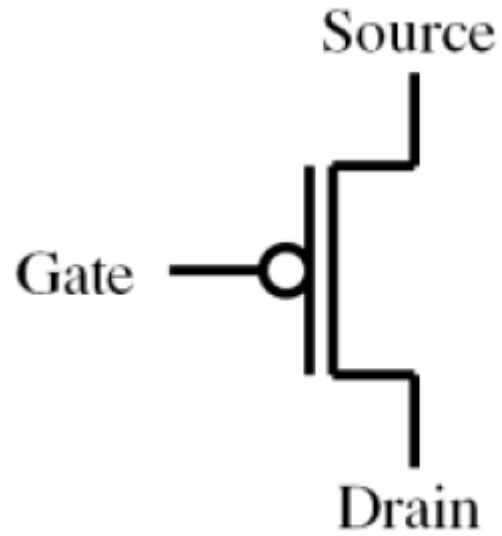
[illegible]

Table 2.8*Boolean Expressions for the 16 Functions of Two Variables*

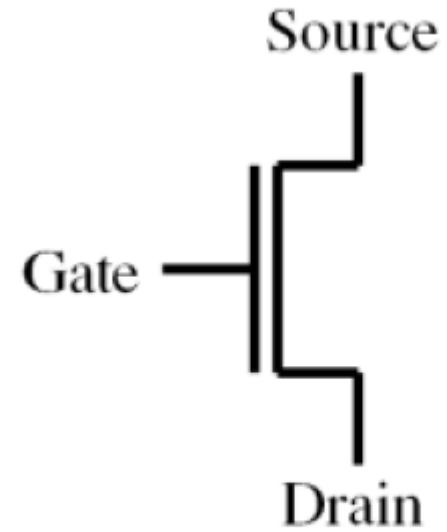
Boolean Functions	Operator Symbol	Name	Comments
$F_0 = 0$		Null	Binary constant 0
$F_1 = xy$	$x \cdot y$	AND	x and y
$F_2 = xy'$	x/y	Inhibition	x , but not y
$F_3 = x$		Transfer	x
$F_4 = x'y$	y/x	Inhibition	y , but not x
$F_5 = y$		Transfer	y
$F_6 = xy' + x'y$	$x \oplus y$	Exclusive-OR	x or y , but not both
$F_7 = x + y$	$x + y$	OR	x or y
$F_8 = (x + y)'$	$x \downarrow y$	NOR	Not-OR

$F_9 = xy + x'y'$	$(x \oplus y)'$	Equivalence	x equals y
$F_{10} = y'$	y'	Complement	Not y
$F_{11} = x + y'$	$x \subset y$	Implication	If y, then x
$F_{12} = x'$	x'	Complement	Not x
$F_{13} = x' + y$	$x \supset y$	Implication	If x, then y
$F_{14} = (xy)'$	$x \uparrow y$	NAND	Not-AND
$F_{15} = 1$		Identity	Binary constant 1

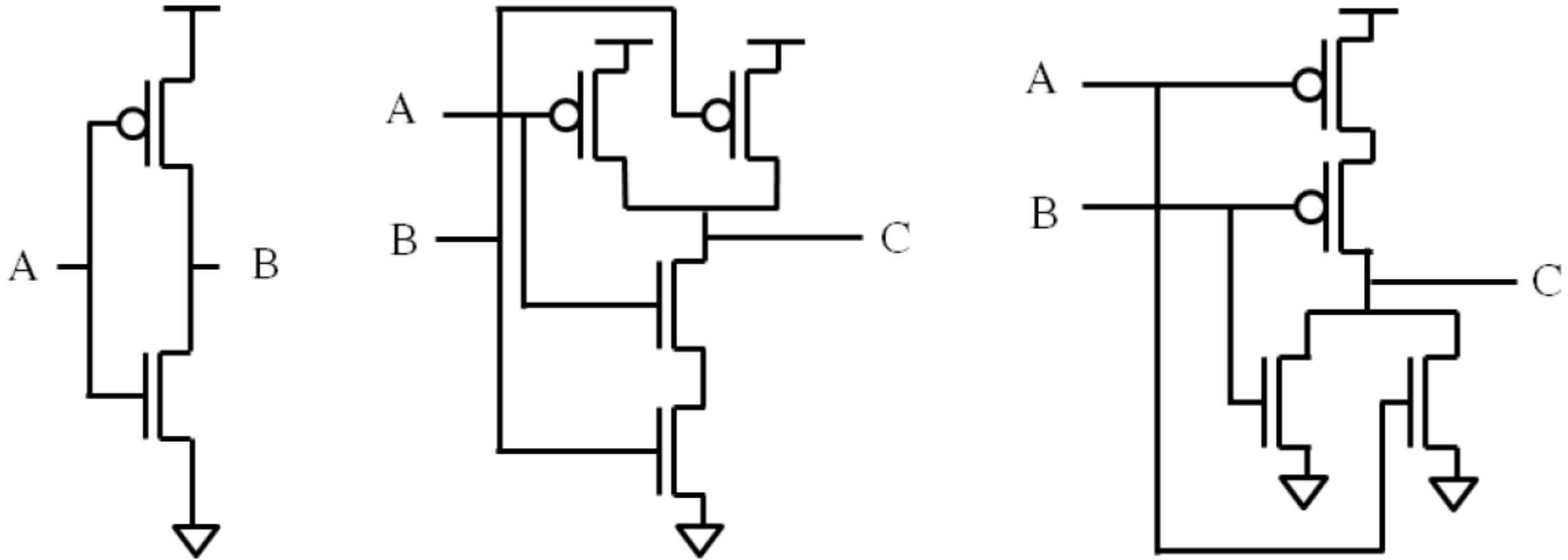
Transistors



(n) P-channel MOSFET



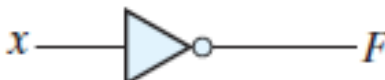
(u) N-channel MOSFET



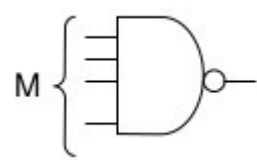
รูปที่ 3.3 เกต not, nand และ nor ที่ออกแบบด้วยเทคโนโลยี CMOS
(เรียงจากซ้ายไปขวาตามลำดับ)

ในการผลิตไมโครโพรเซสเซอร์โดยปกติจะใช้เทคโนโลยีการออกแบบที่เรียกว่า complementary metal-oxide-semiconductor หรือ CMOS หลักการออกแบบคือ P-channel MOSFET 1 ตัวจะคู่กับ N-channel MOSFET 1 ตัวเสมอ และในขณะที่ตัวใดตัวหนึ่ง on อีกตัวหนึ่งจะต้อง off จะกินไฟเฉพาะช่วงที่มีการเปลี่ยนแปลงระหว่าง on และ off เท่านั้น ทำให้ CMOS กินไฟน้อยมาก และไม่เกิดความร้อนมากนัก

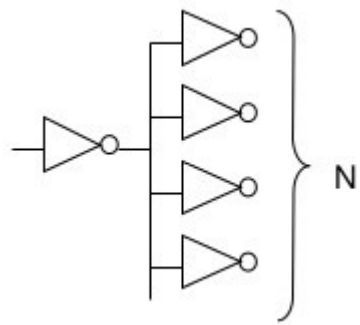
Digital Logic Gates

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

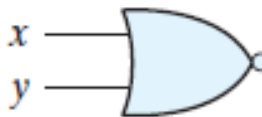
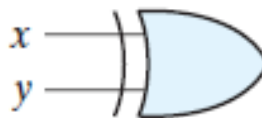
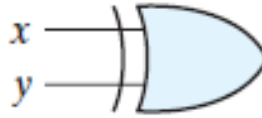
fan-in



fan-out



เกตจะทำงานได้ช้าลง

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

ใช้เวลาเดินสายไฟไกล ๆ
แรงดันจะตก ต้องใช้ buffer ช่วย

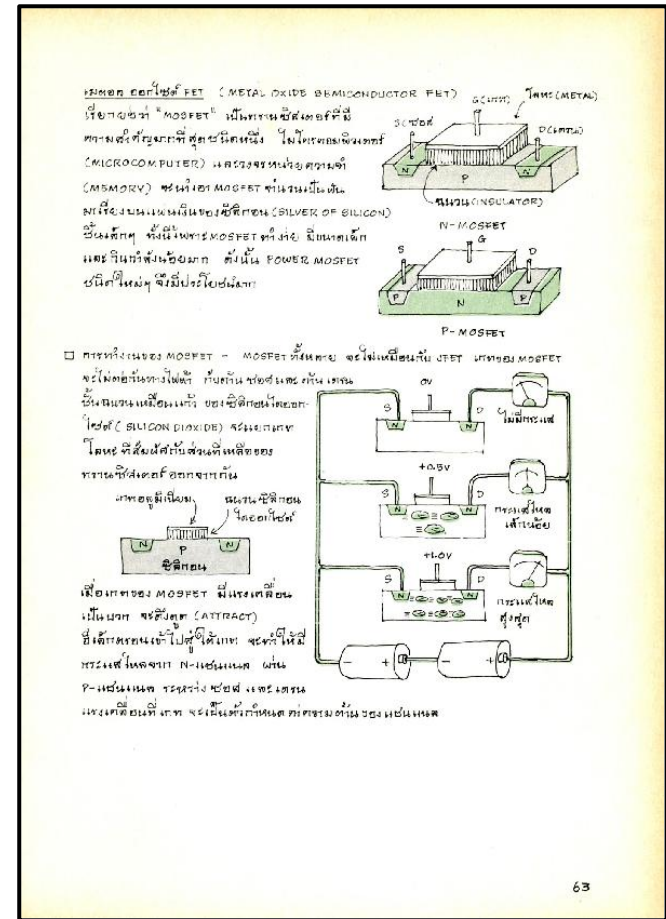
Universality

“Only NAND gates are needed to build all other gates.”

Exercise

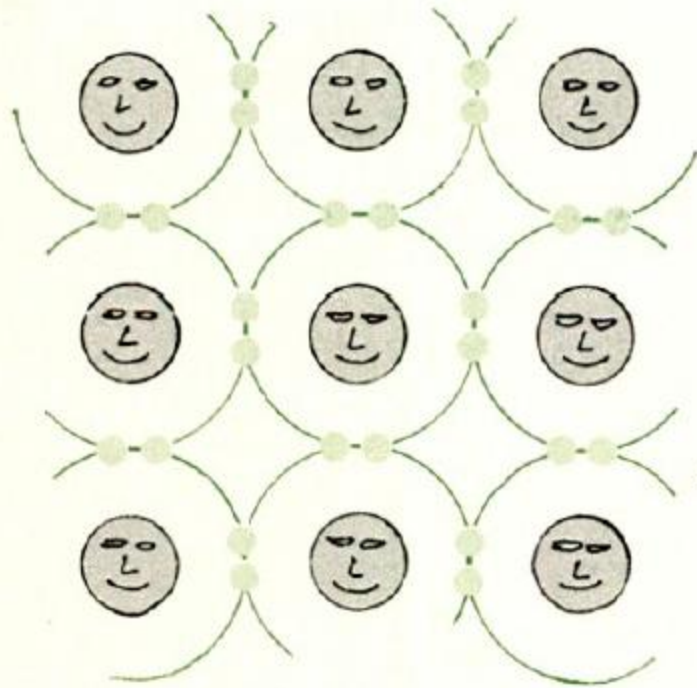
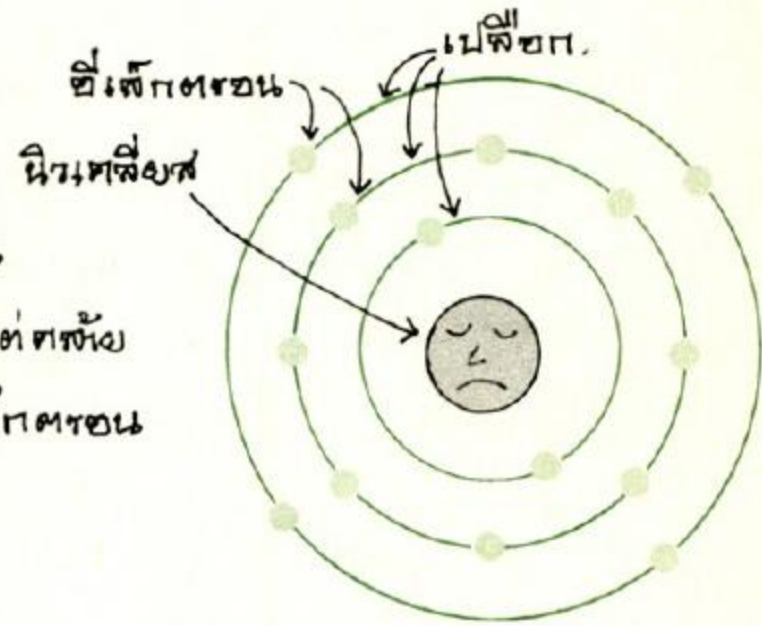
- สร้าง Inverter จาก NAND
- สร้าง AND จาก NAND
- สร้าง OR จาก NAND
- สร้าง NOR จาก NAND

อ่านเรื่องสารกึ่งตัวนำ และทรานซิสเตอร์



Double click เพื่อเปิดเอกสาร

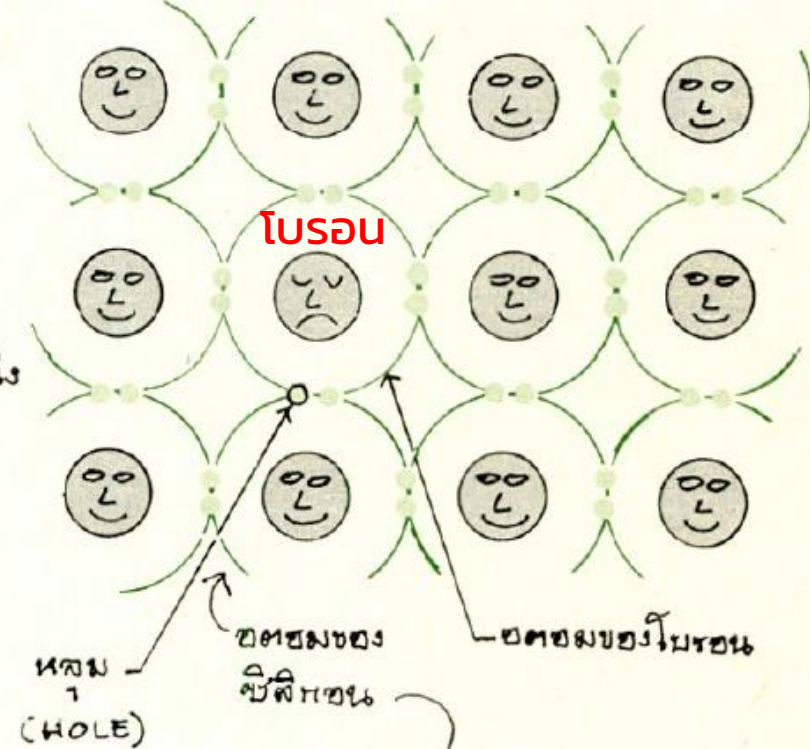
□ ซิลิกอน (SILICON) - มีสารกึ่งตัวนำแยกต่างหาก
 หลายชนิด แต่ซิลิกอนซึ่งเป็นส่วนประกอบของทราย
 เป็นที่รู้จักกันแพร่หลายมากที่สุด คุณสมบัติของซิลิกอน
 ใต้เปลือก (SHELL) นอกสุด จะมี 4 อิเล็กตรอน (แต่ที่หัว
 กันมี 8 อิเล็กตรอน) สามารถเชื่อมต่อกับอีก 4 อิเล็กตรอน
 ของอะตอมข้างเคียง เรียกว่า อิเล็กตรอนร่วม
 (SHARE ELECTRONS)



กลุ่มของ อะตอมซิลิกอน จะรวมกลุ่ม (SHARE)
 อิเล็กตรอน ของเปลือกนอกสุด จะเกิด
 (FORM) กันเป็นรูปของ ผลึก (CRYSTAL)

← ภาพขยายของ ผลึกซิลิกอน เพื่อให้เห็น
 ง่าย จึงแสดงเฉพาะ อิเล็กตรอนวงนอกสุด

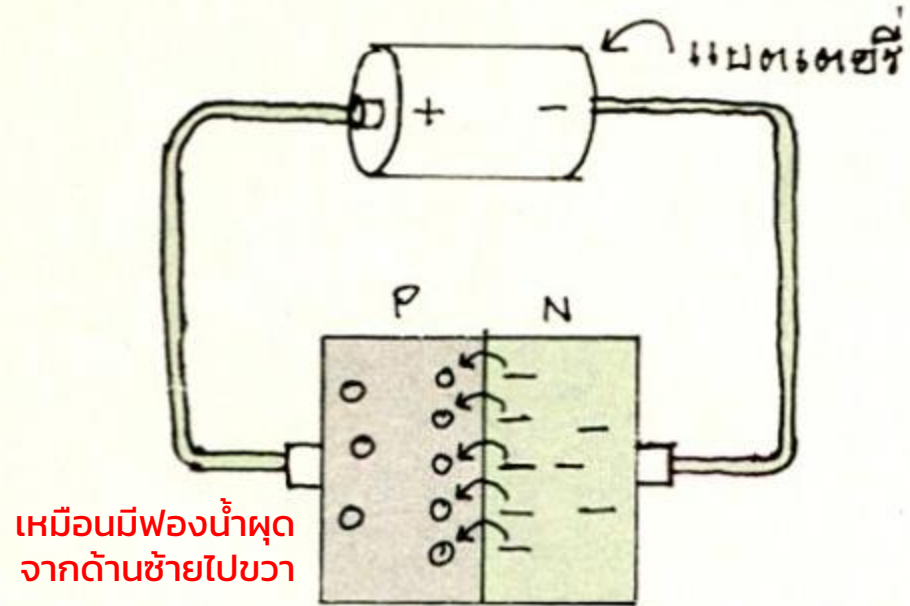
□ ชนิดกึ่งตัวนำแบบ P (P-TYPE SILICON) - วัสดุชนิดนี้ประกอบด้วยกลุ่มของซิลิคอนที่มีอิเล็กตรอนวาเลนซ์อยู่ เรียกว่า หลุม (HOLE) จะมีโอกาสให้ อิเล็กตรอนข้างเคียงมาทดแทนไป ในหลุม ดังนั้นหลุมจะเคลื่อนที่ไปอย่างช้าๆ หนึ่งหลุมได้ หลุมสามารถเคลื่อนที่ผ่านของซิลิคอนได้ ตลอดเวลาในลักษณะ



□ ชนิดกึ่งตัวนำแบบ N (N-TYPE SILICON) - วัสดุชนิดนี้ประกอบด้วยฟอสฟอรัสในกลุ่มของซิลิคอน จะมีการให้ อิเล็กตรอนพิเศษ ซึ่งจะมีอิเล็กตรอนที่เพิ่ม กลุ่มของซิลิคอน อย่างช้าๆ ตลอดเวลา หรืออีกนัยหนึ่ง ชนิดกึ่งตัวนำแบบ N นำกระแสไฟฟ้า เมื่อหลุมของซิลิคอนแบบ P ก็นำกระแสไฟฟ้าได้เช่นกัน

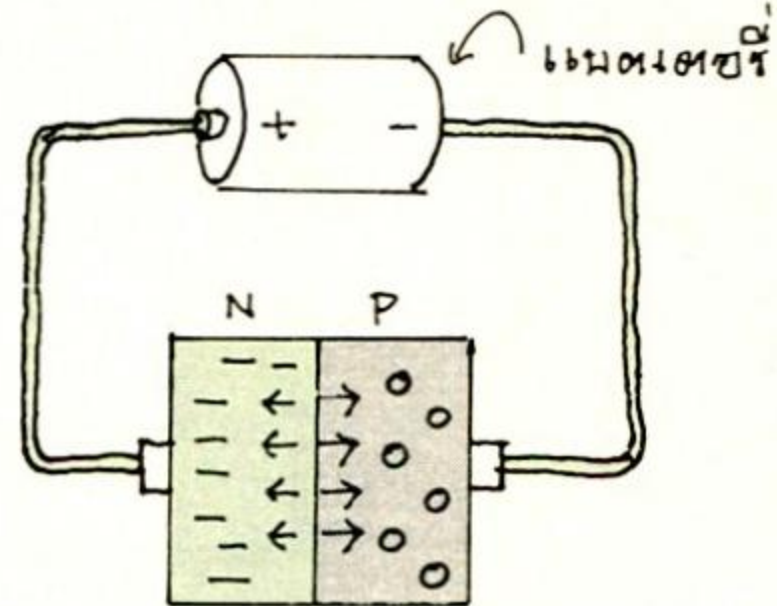


□ การทำงานของไดโอด - เพื่ออธิบายให้เข้าใจยิ่งขึ้น ขงดูภาพประกอบ แสดงไดโอดนำกระแส
ไปในทิศทางหนึ่ง ไดโอดจะเปรียบเสมือนกำแพง (BLOCK) กระแสจึงไหลไปในทิศทางตรงกันข้าม



←
การไหลของอิเล็กตรอน
→
การไหลของหลุม

พิกัดไบแอส (FORWARD BIAS)

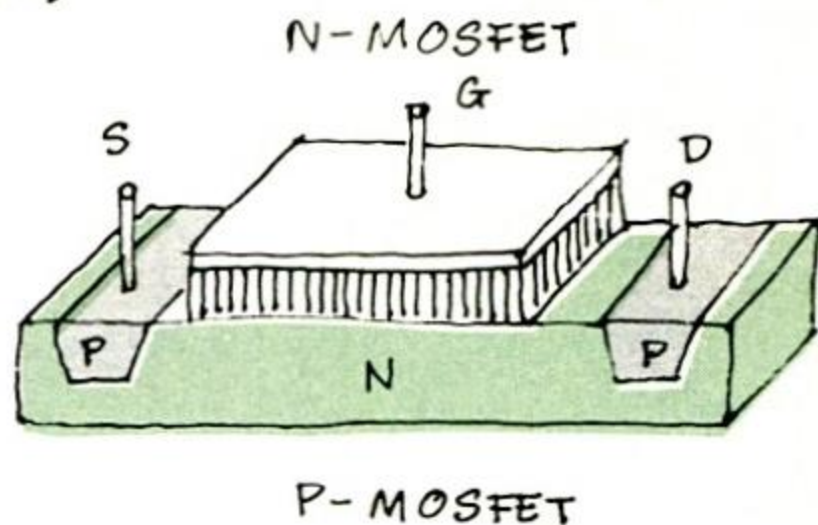
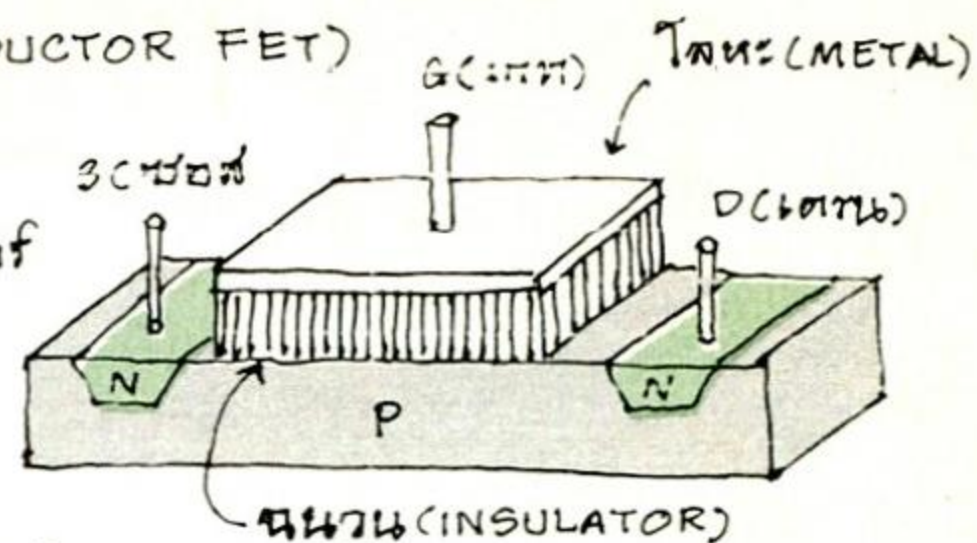


ไม่มีกระแสไหล
(NO CURRENT FLOW)

รีเวิร์สไบแอส (REVERSE BIAS)

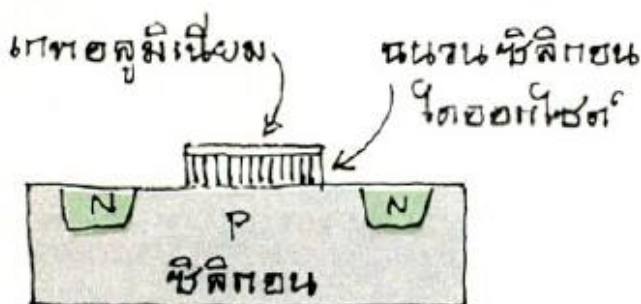
เมทัล ออกไซด์ ฟิลด์ เอฟอีที (METAL OXIDE SEMICONDUCTOR FET)

เรียกย่อว่า "MOSFET" เป็นทรานซิสเตอร์ที่มี
ตามหลักทฤษฎีที่สุตบดินทร์ ในไมโครคอมพิวเตอร์
(MICROCOMPUTER) และวงจรหน่วยความจำ
(MEMORY) จะนำเอา MOSFET จำนวนเป็นพัน
มาเรียงบนแผ่นเงินของซิลิคอน (SILVER OF SILICON)
ซึ่งมีลักษณะทั้งนี้เพราะ MOSFET ทำง่าย มีขนาดเล็ก
และกินกำลังน้อยมาก ดังนั้น POWER MOSFET
ชนิดใหม่ ๆ จึงมีประโยชน์มาก

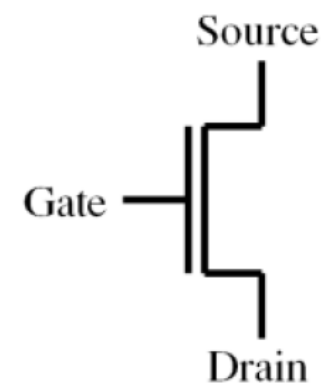
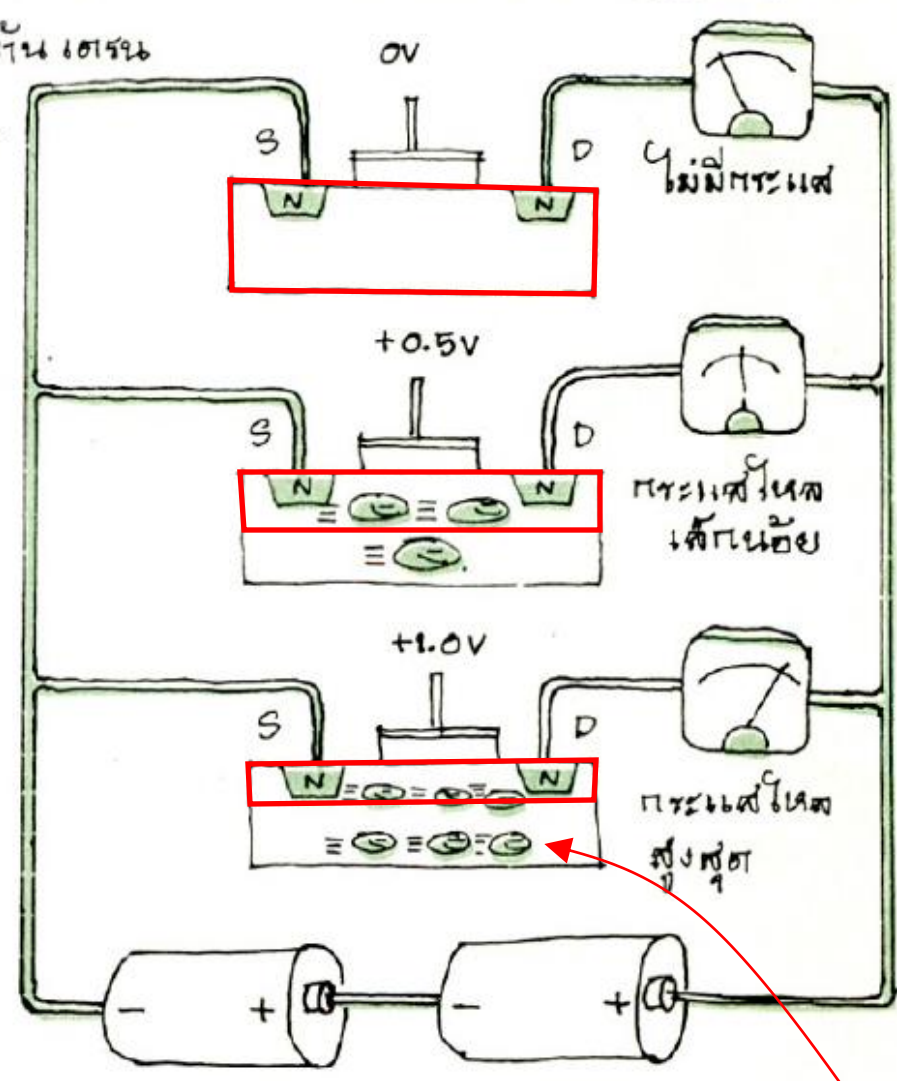


□ การทำงานของ MOSFET - MOSFET ทั้งหลาย จะใช้เหมือนกับ JFET แต่ทว่า MOSFET

จะไม่ต่อทั้งทางไฟฟ้า กับทั้ง ขั้วต่อ และ ขั้วต่อ
ชั้นฉนวนเหมือนแก้ว ของซิลิคอนไดออกไซด์ (SILICON DIOXIDE) จะแยกเกต
โหนด ที่สัมผัสกับส่วนที่เหลือของ
ทรานซิสเตอร์ออกจากกัน



เมื่อเกตของ MOSFET มีแรงดันเป็นบวก จะดึงดูด (ATTRACT) อิเล็กตรอนเข้าไปสู่ใต้เกต จะทำให้มีกระแสไหลจาก N-แชนแนล ผ่าน P-แชนแนล ระหว่าง ขั้วต่อ และ ขั้วต่อ แรงดันที่เกต จะเป็นตัวกำหนดค่ากระแสที่ไหลของ แชนแนล

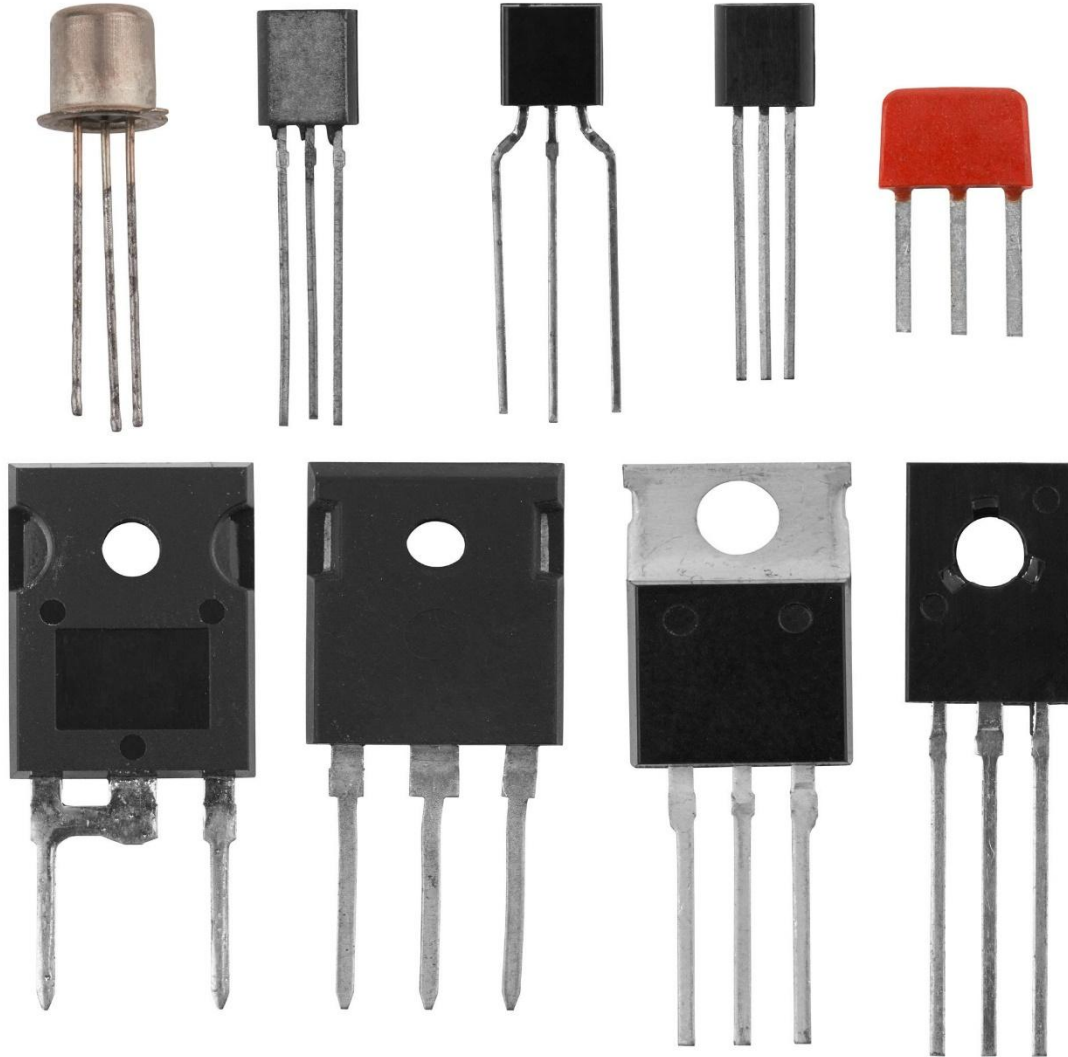


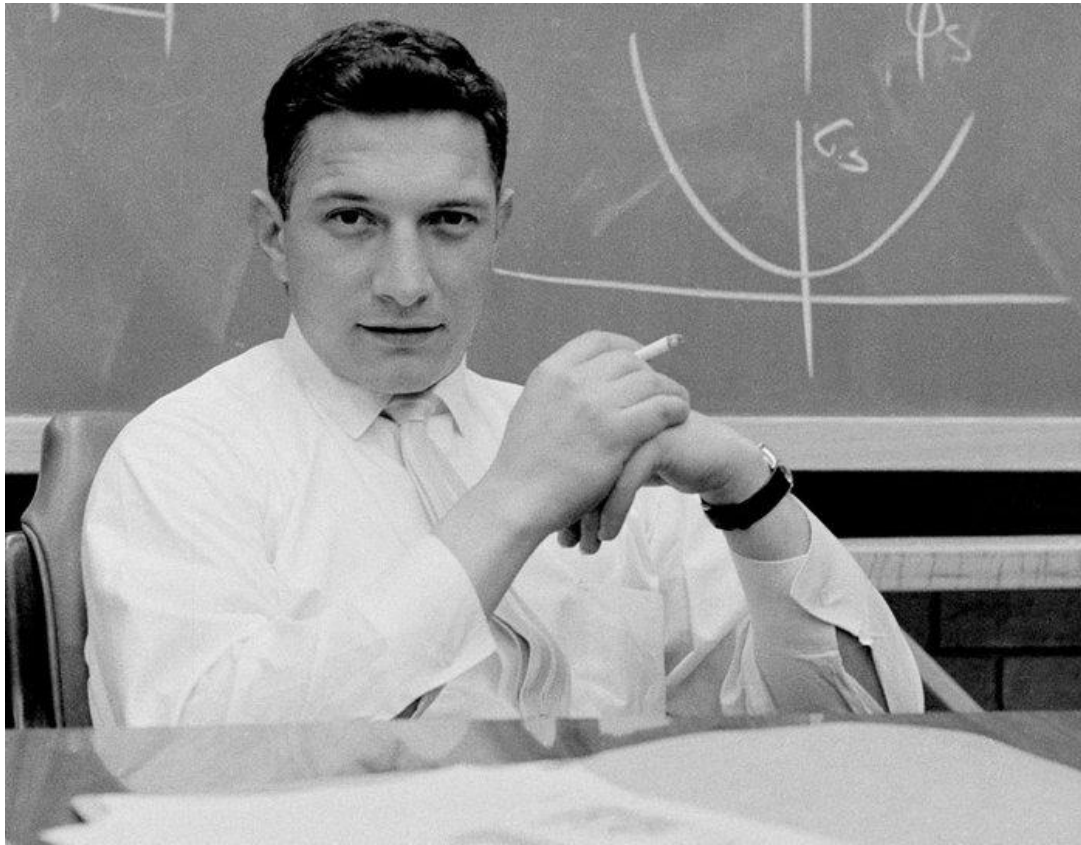
(ข) N-channel MOSFET

P-channel MOSFET ก็อธิบายในทำนองเดียวกัน แต่กลับกันคือ ให้แรงดันต่ำ (0V) ที่ Gate กระแสจะไหลจาก Source ไป Drain

แรงดันที่ gate จะดึง electron ไปใต้ gate ส่วน hole จะถูกผลักลงไปยังด้านล่าง ทำให้บริเวณใต้ gate มีสภาพเป็น N-type silicon
<https://www.youtube.com/watch?v=rkbjHNEKcRw>

Transistors





Robert Noyce (the Founder of Intel)

เป็นผู้คิดค้น integrated circuit (IC) ไม่ต้องผลิต transistor เป็นตัว ๆ แล้วนำมาประกอบกันเป็นวงจร แต่เชื่อม transistor ให้เป็นวงจรในขั้นตอนการผลิตเลย

April 25, 1961

R. N. NOYCE

2,981,877

End 1978

SEMICONDUCTOR DEVICE-AND-LEAD STRUCTURE

Filed July 30, 1959

3 Sheets-Sheet 2

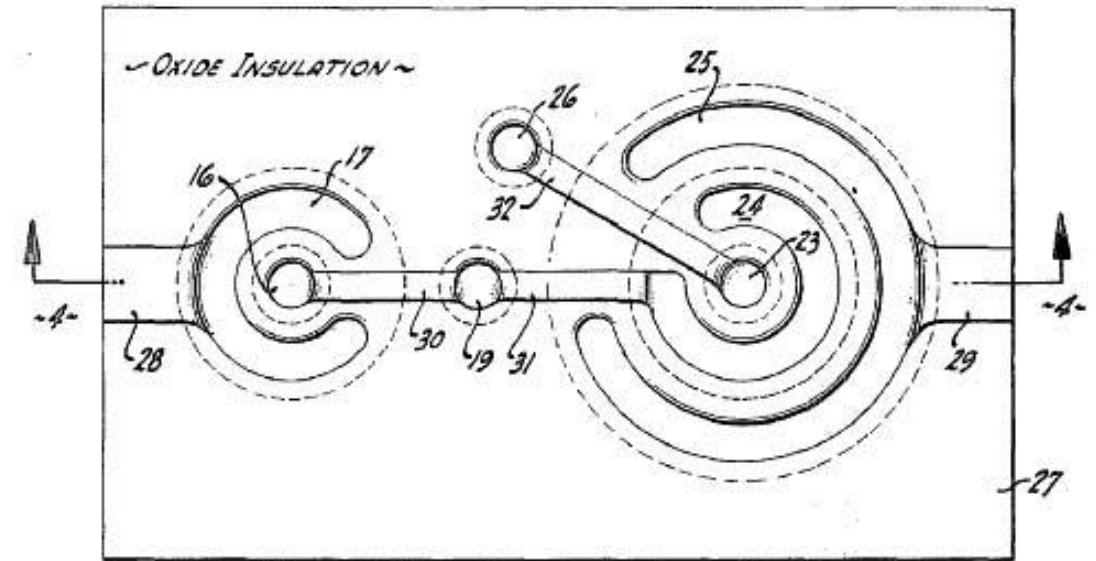


FIG. 3

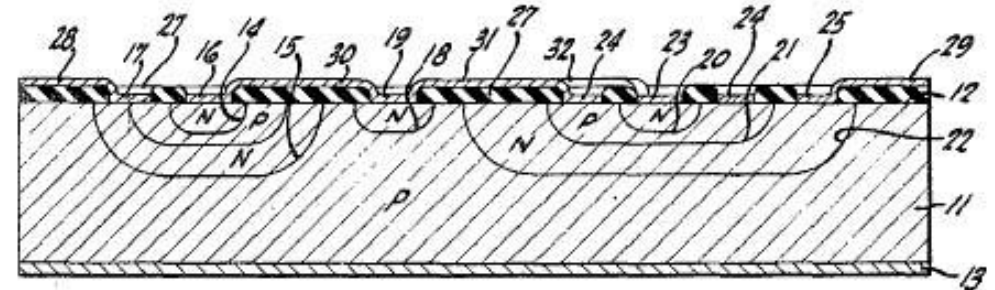
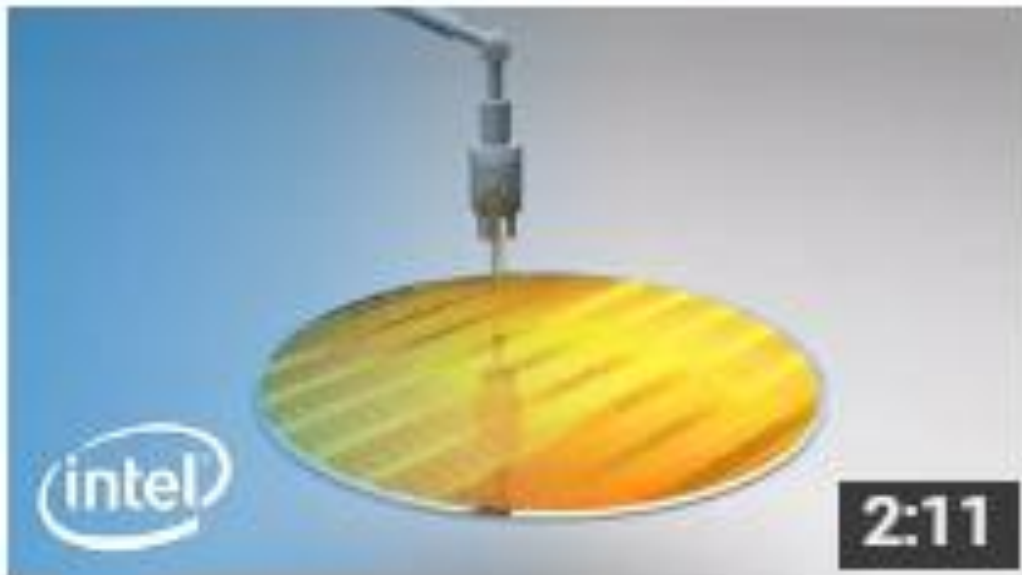


FIG. 4

สิทธิบัตรร่วมกับ Jack Kilby (Texas Instrument)

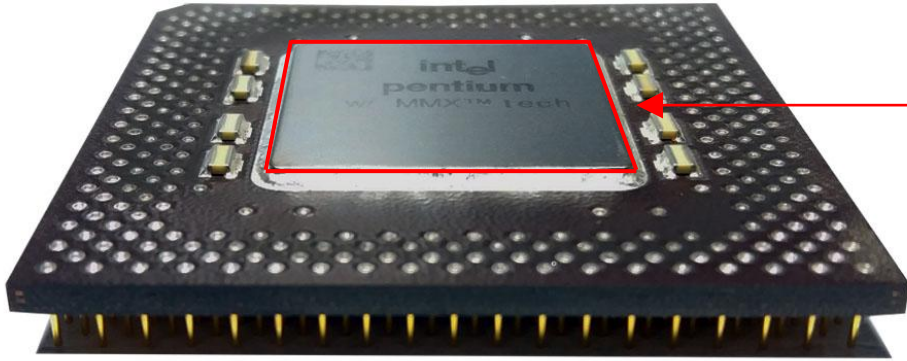


From Sand to Silicon: the Making of a Chip | Intel

Intel

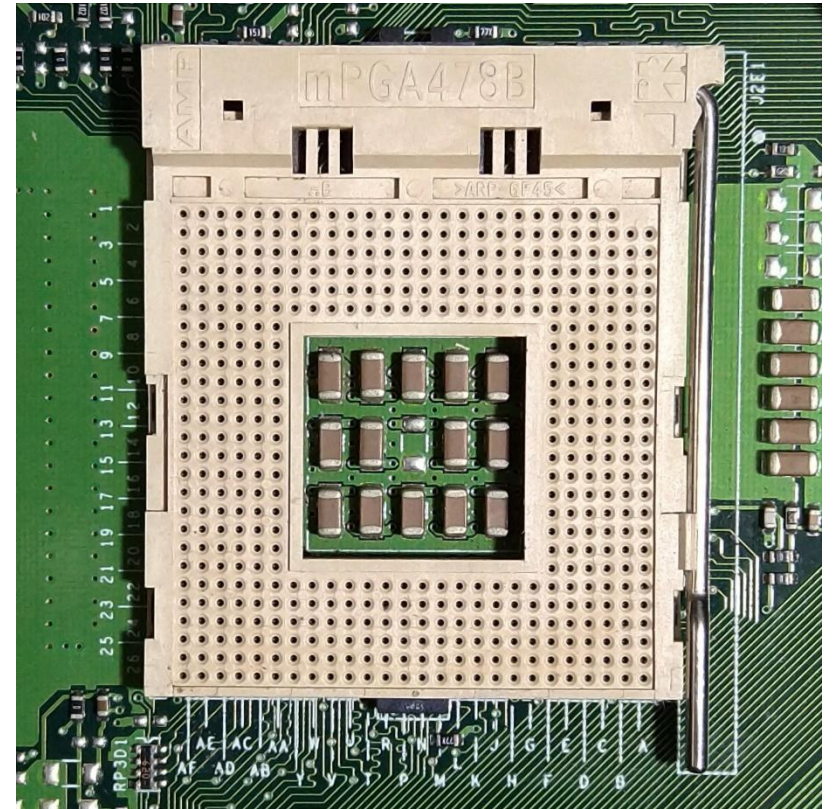
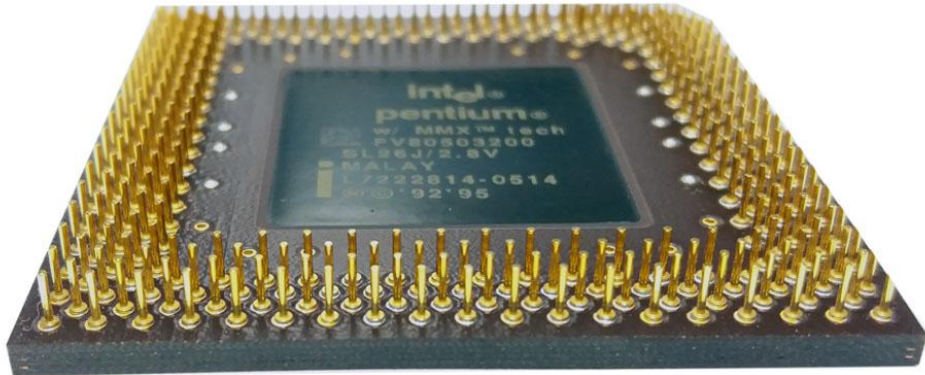
ดู 399,659 ครั้ง

<https://www.youtube.com/watch?v=Q5paWn7bFg4>



(CPU) Die

อยู่ไหน



(CPU) Socket

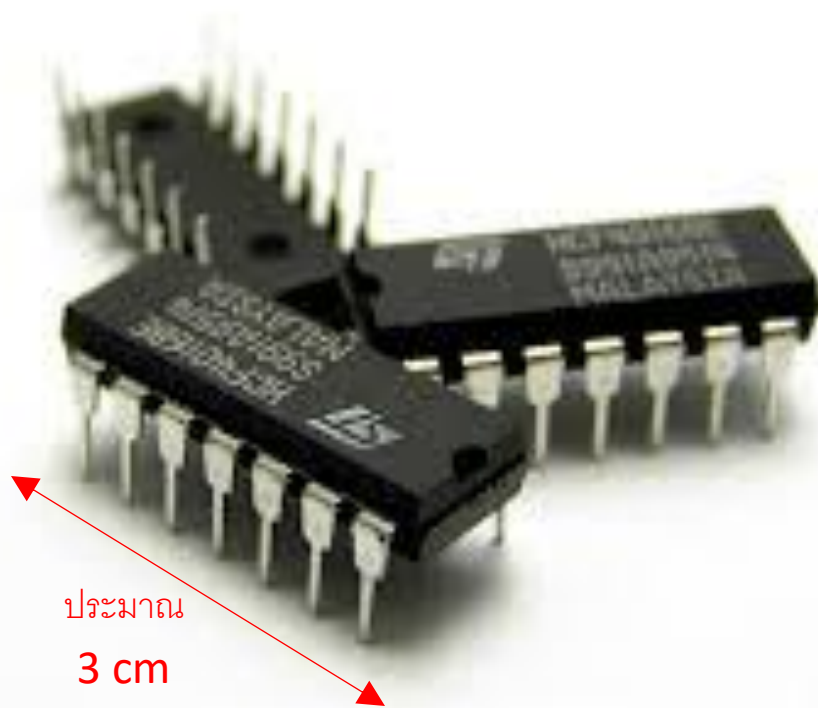
Level of Integration

- Small-scale integration (SSI)
- Medium-scale integration (MSI)
- Large-scale integration (LSI)
- Very-large scale integration (VLSI)

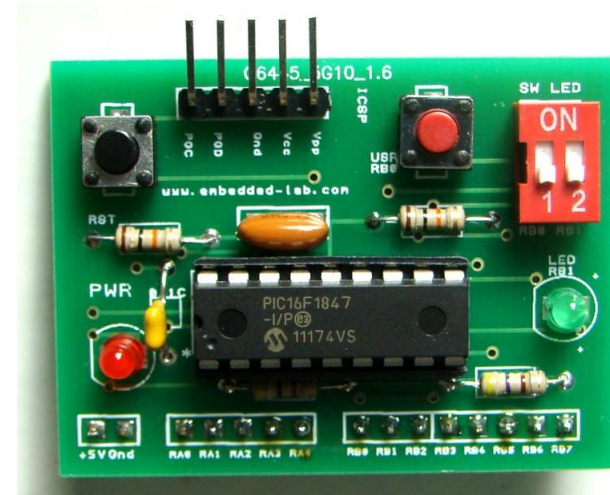
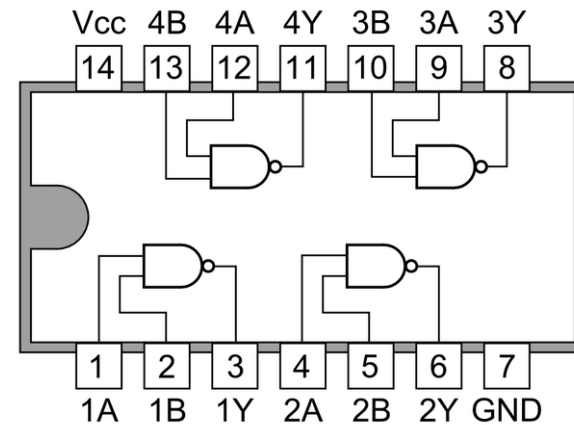
Name ◆	Signification ◆	Year ◆	Transistors number ^[47] ◆	Logic gates number ^[48] ◆
SSI	<i>small-scale integration</i>	1964	1 to 10	1 to 12
MSI	<i>medium-scale integration</i>	1968	10 to 500	13 to 99
LSI	<i>large-scale integration</i>	1971	500 to 20,000	100 to 9,999
VLSI	<i>very large-scale integration</i>	1980	20,000 to 1,000,000	10,000 to 99,999
ULSI	<i>ultra-large-scale integration</i>	1984	1,000,000 and more	100,000 and more

Small-Scale Integration (SSI)

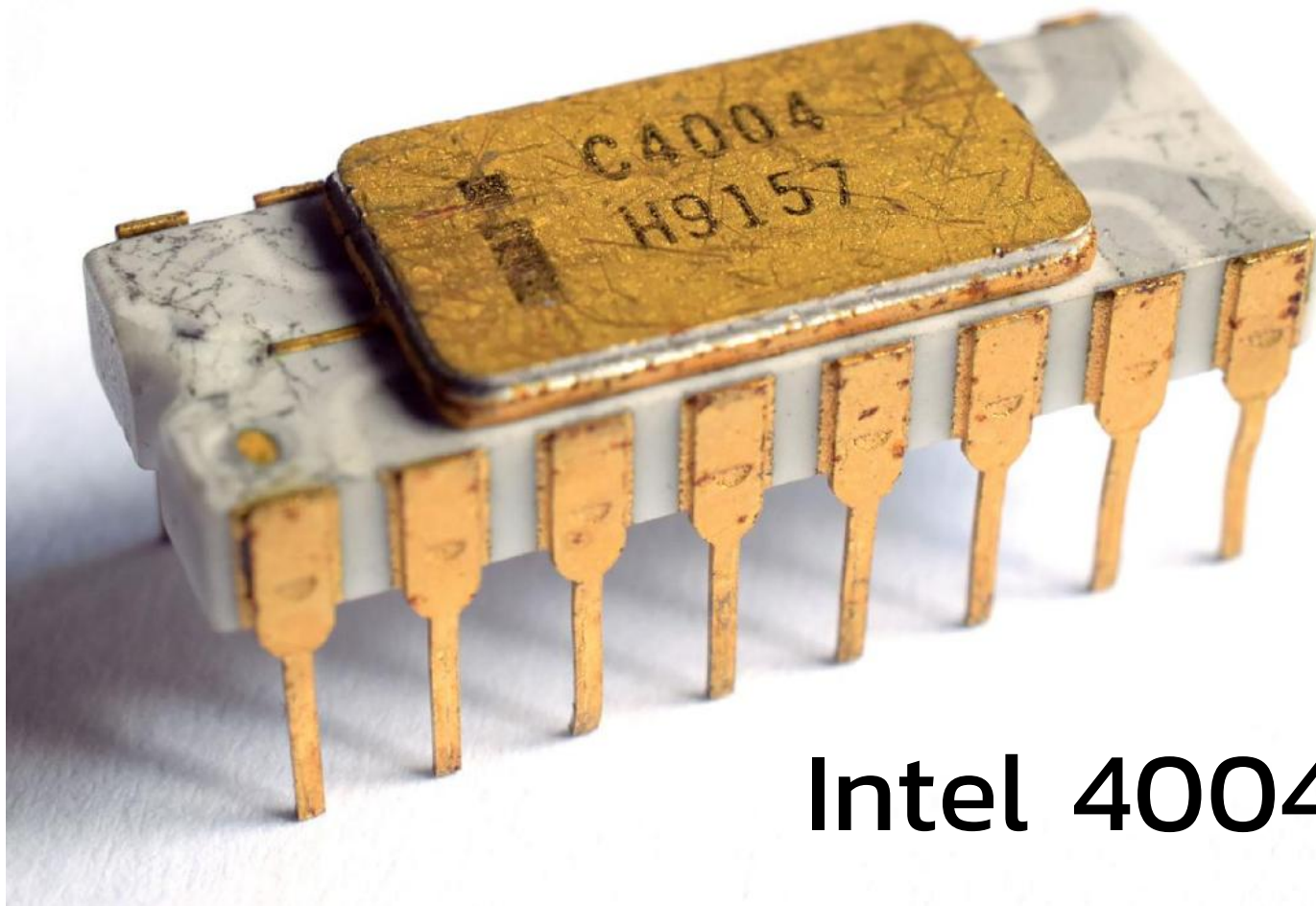
Medium-Scale Integration (MSI)



7400 Quad 2-input NAND Gates

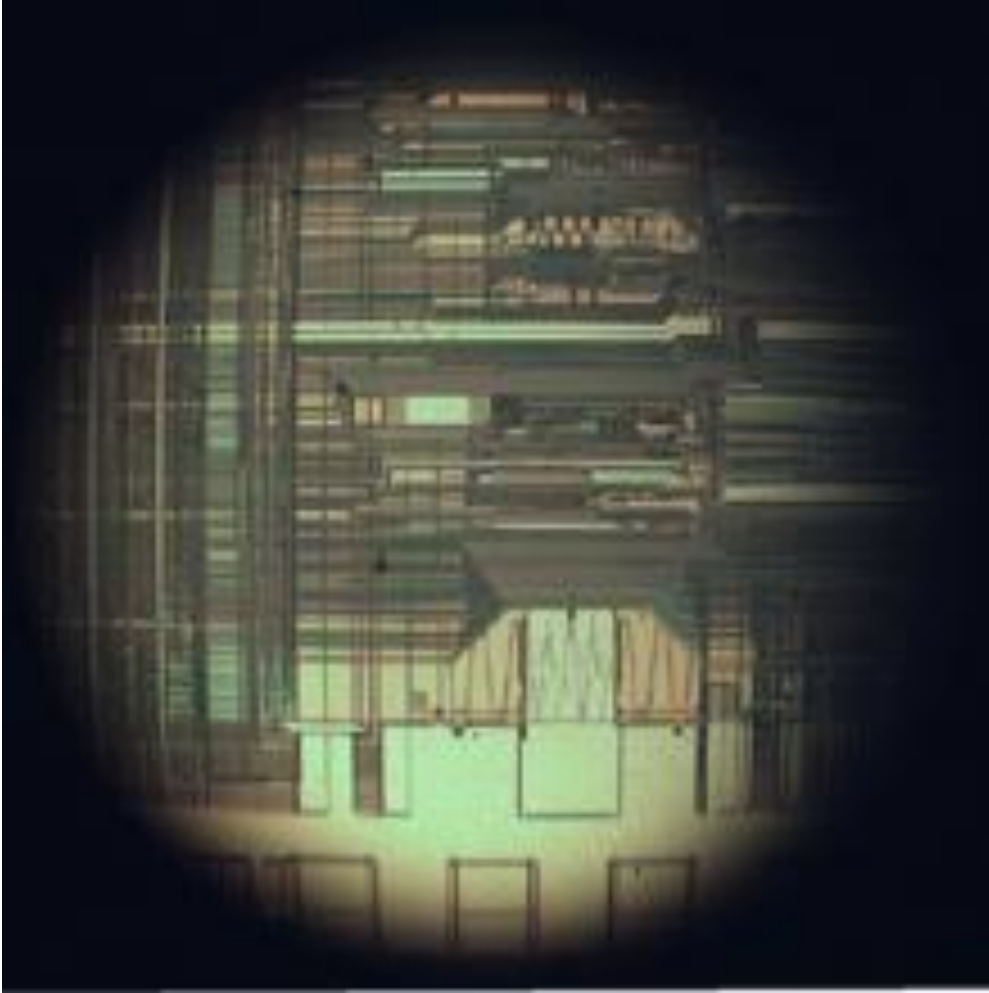


Large-Scale Integration (LSI)

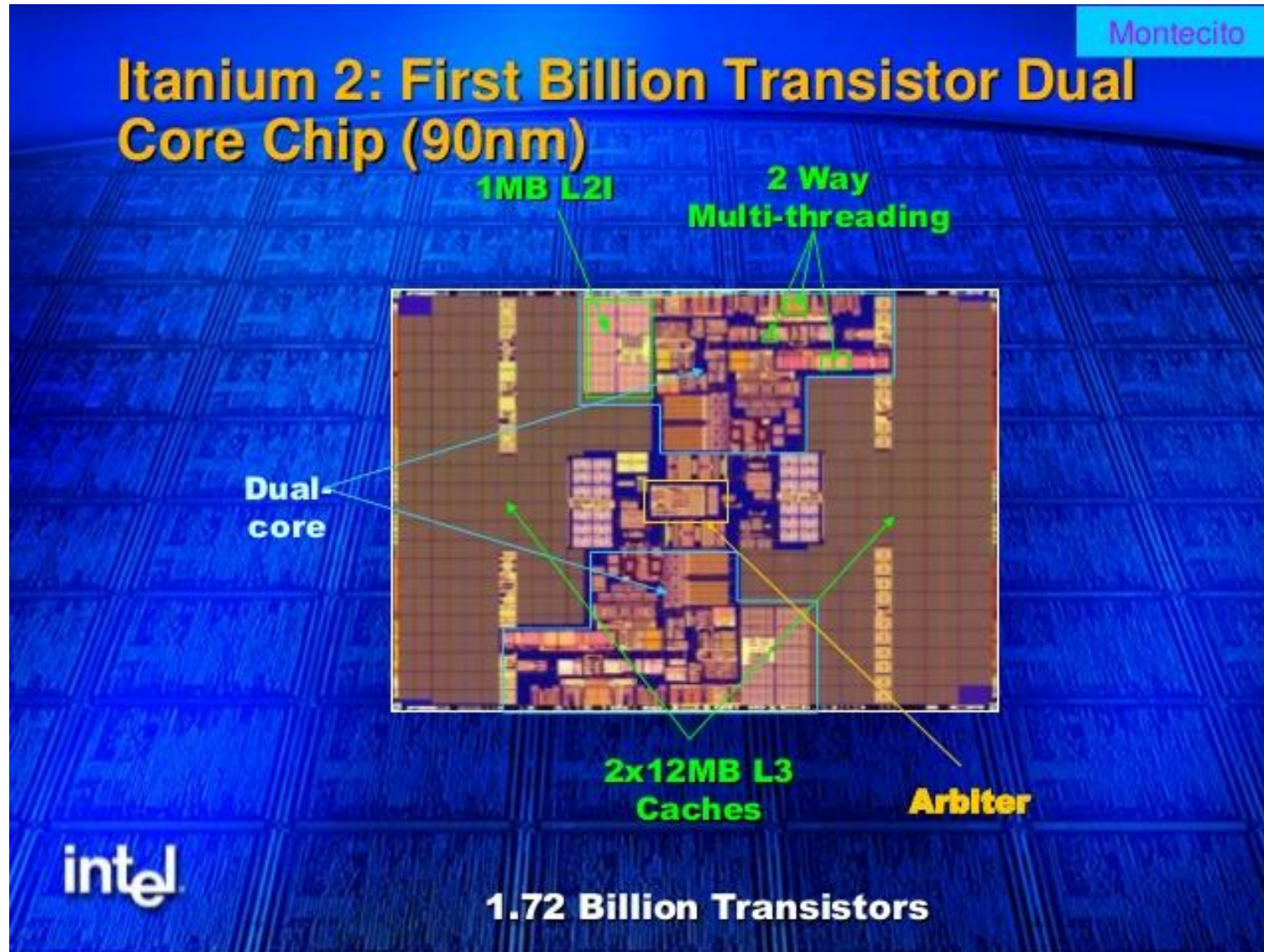


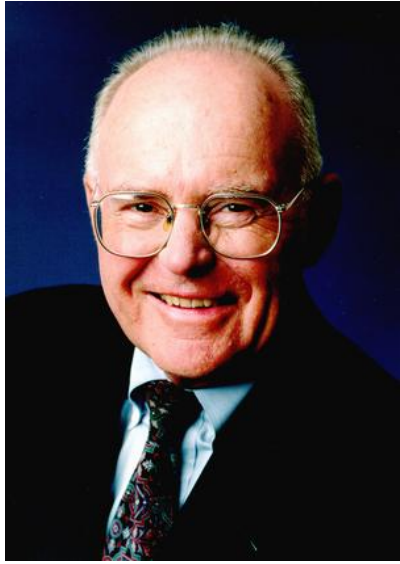
Intel 4004

Very Large-Scale Integration (VLSI)

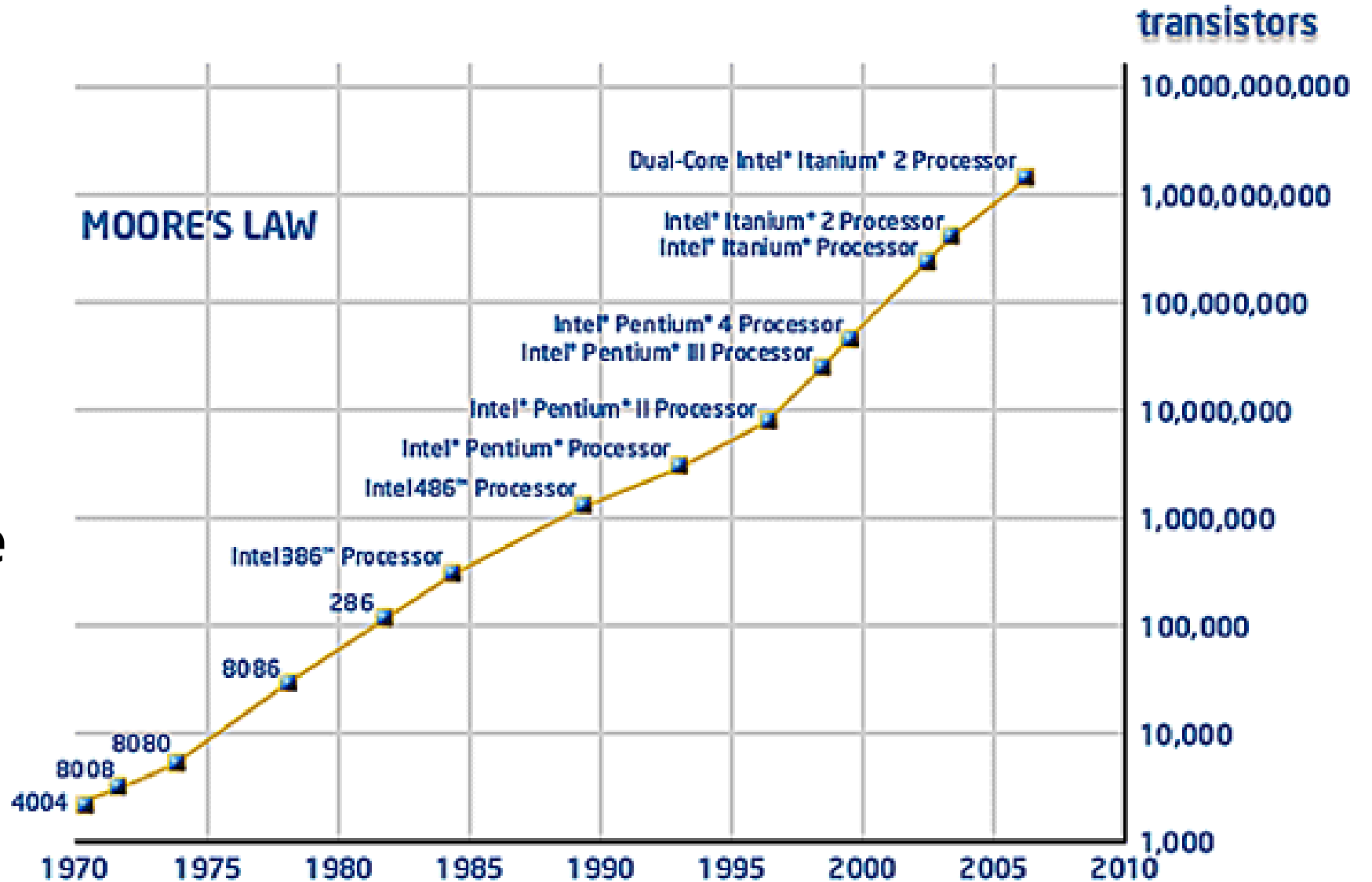


Ultra Large-Scale Integration



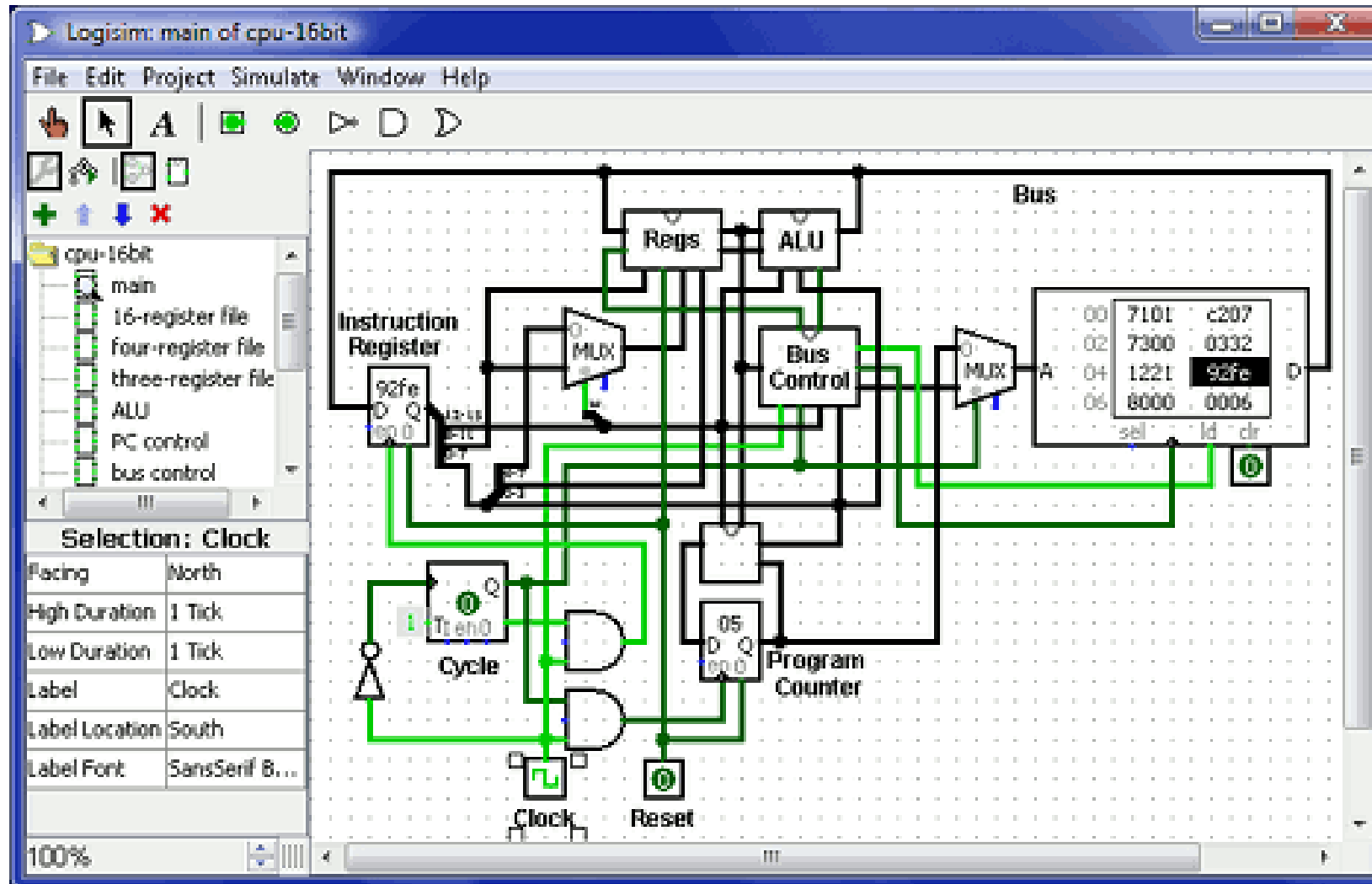


**Gordon Moore
(Intel)**



The M1 Max is equipped with **57 billion transistors** and it is built on a **5-nanometer process**. Intel doesn't disclose transistor count anymore. A typical atom is anywhere from 0.1 to 0.5 nanometers in diameter. DNA molecules are about 2.5 nanometers wide.

LogicSim



Windows

logisim-win-2.7.1.exe

Mac

logisim-macosx-2.7.1.tar.gz

Java

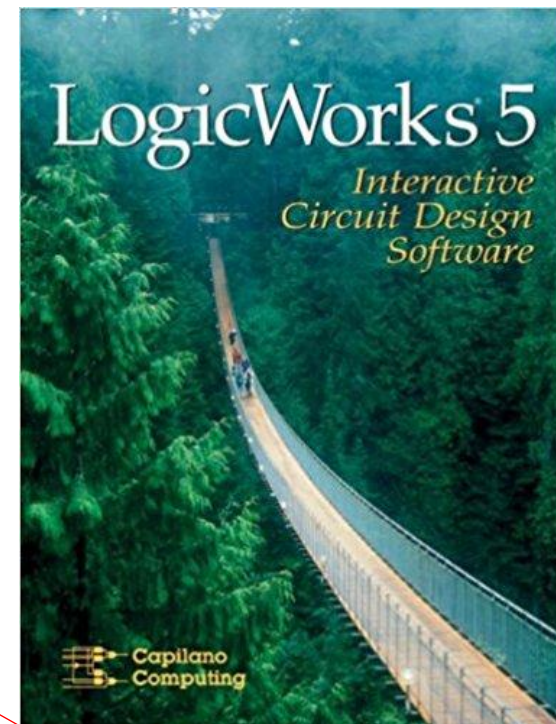
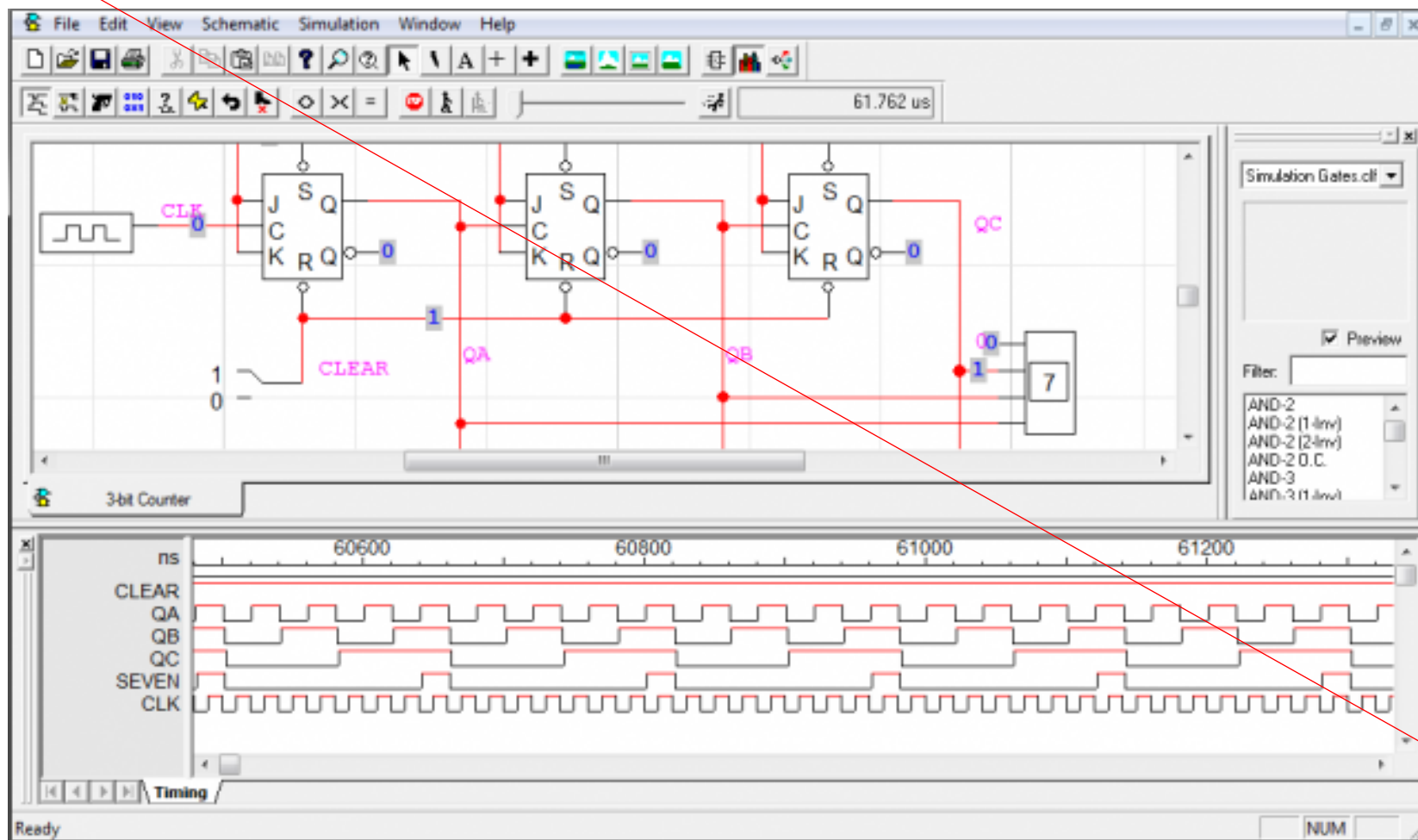
logisim-generic-2.7.1.jar

ใช้กับ Java SE Runtime Environment 8u351

<http://www.cburch.com/logisim>

<https://sourceforge.net/projects/circuit/>

LogicWorks อ่านคู่มือการใช้งานภาษาอังกฤษ

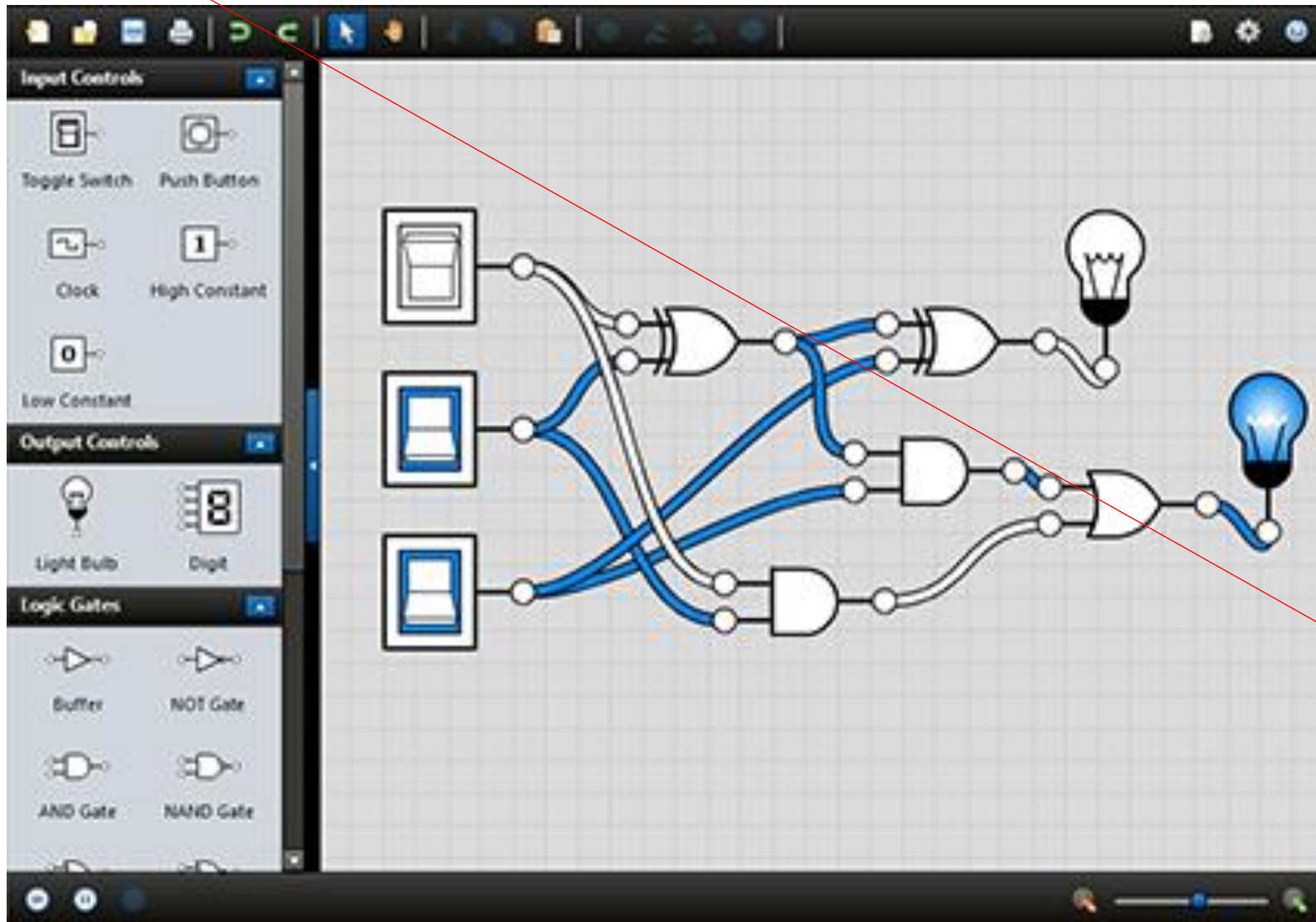


Windows: <https://cache111.com/me/2301274/LogicWorks5.zip>
Mac: <https://mac.softpedia.com/get/Math-Scientific/LogicWorks.shtml>

Contents

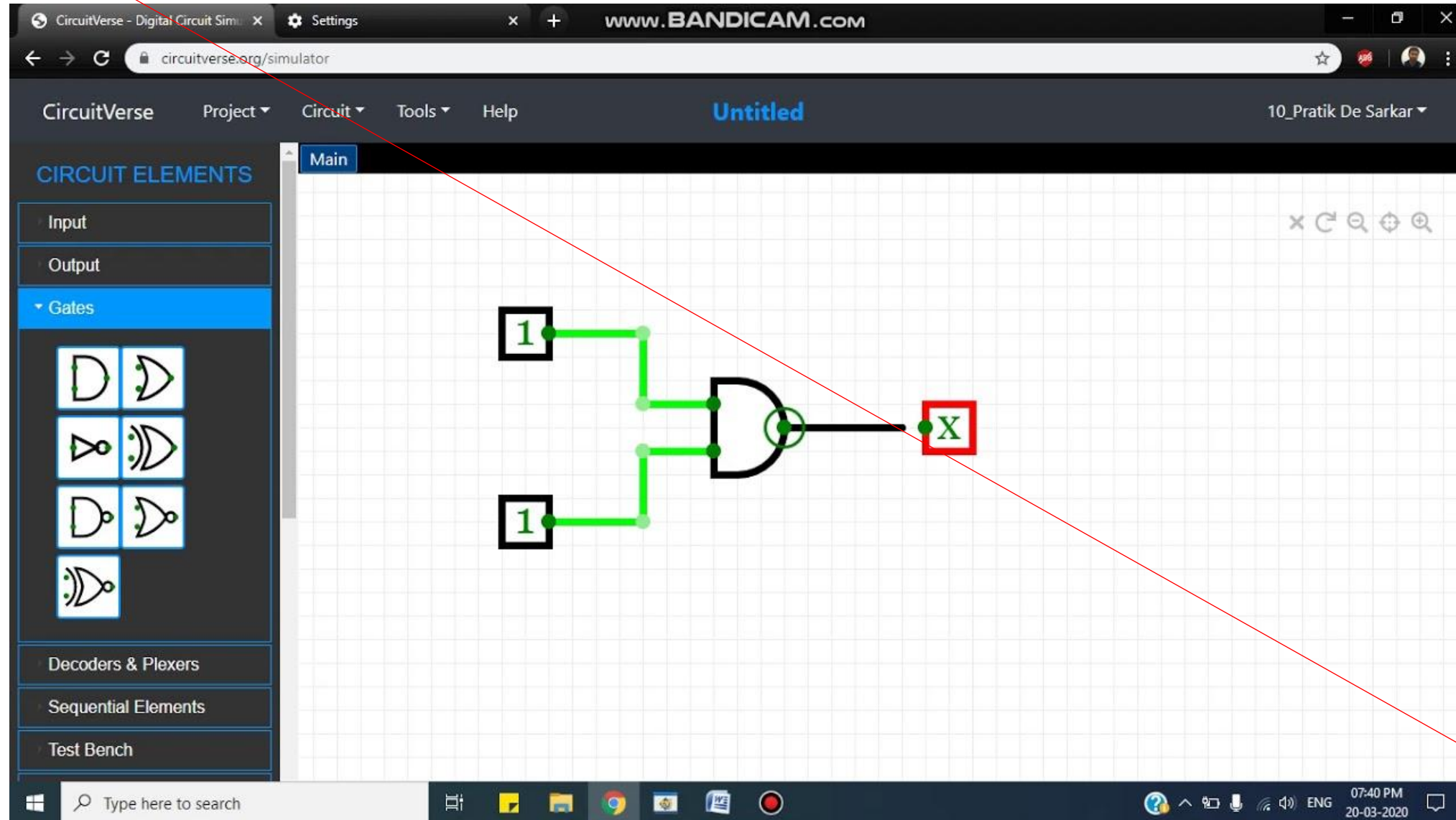
Course Syllabus (ดูใน CU-CAS)
[Download LogicWorks 5](#)
[Download Karnaugh Minimizer 2.0](#)

LogicLy



<https://logic.ly>

CircuitVerse



<https://circuitverse.org>

PROBLEMS

(Answers to problems marked with * appear at the end of the text.)

2.1 Demonstrate the validity of the following identities by means of truth tables:

- (a) DeMorgan's theorem for three variables: $(x + y + z)' = x'y'z'$ and $(xyz)' = x' + y' + z'$
- (b) The distributive law: $x + yz = (x + y)(x + z)$
- (c) The distributive law: $x(y + z) = xy + xz$
- (d) The associative law: $x + (y + z) = (x + y) + z$
- (e) The associative law and $x(yz) = (xy)z$

2.2 Simplify the following Boolean expressions to a minimum number of literals:

- (a)* $xy + xy'$
- (b)* $(x + y)(x + y')$
- (c)* $xyz + x'y + xyz'$
- (d)* $(A + B)'(A' + B')'$
- (e) $(a + b + c')(a'b' + c)$
- (f) $a'bc + abc' + abc + a'bc'$

2.3 Simplify the following Boolean expressions to a minimum number of literals:

- (a)* $ABC + A'B + ABC'$
- (b)* $x'yz + xz$
- (c)* $(x + y)'(x' + y')$
- (d)* $xy + x(wz + wz')$
- (e)* $(BC' + A'D)(AB' + CD')$
- (f) $(a' + c')(a + b' + c')$

2.4 Reduce the following Boolean expressions to the indicated number of literals:

- (a)* $A'C' + ABC + AC'$ to three literals
- (b)* $(x'y' + z)' + z + xy + wz$ to three literals
- (c)* $A'B(D' + C'D) + B(A + A'CD)$ to one literal
- (d)* $(A' + C)(A' + C')(A + B + C'D)$ to four literals
- (e) $ABC'D + A'BD + ABCD$ to two literals