

W05-Lab1

Boolean Logic and Gates

Assembled for 204111
by Ratsameetip Wita

Boolean Logic (Algebra)

- การประมวลผลระดับวงจร ของคอมพิวเตอร์ อยู่บนพื้นฐานของ **Boolean Logic (Algebra) True (1) False (0)**

A	B	$A \wedge B$ (A AND B) (conjunction)	$A \vee B$ (A or B) (disjunction)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

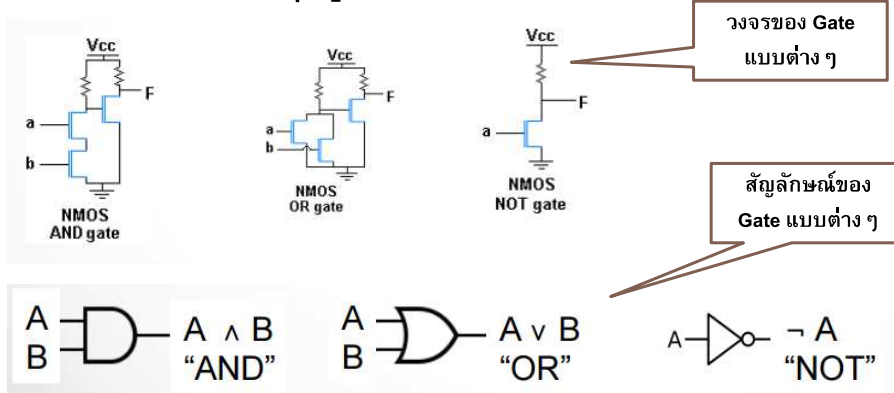
A	$\neg A$ (NOT A) (negation)
0	1
1	0

Computer Systems: A Programmer's Perspective, 2nd Edition

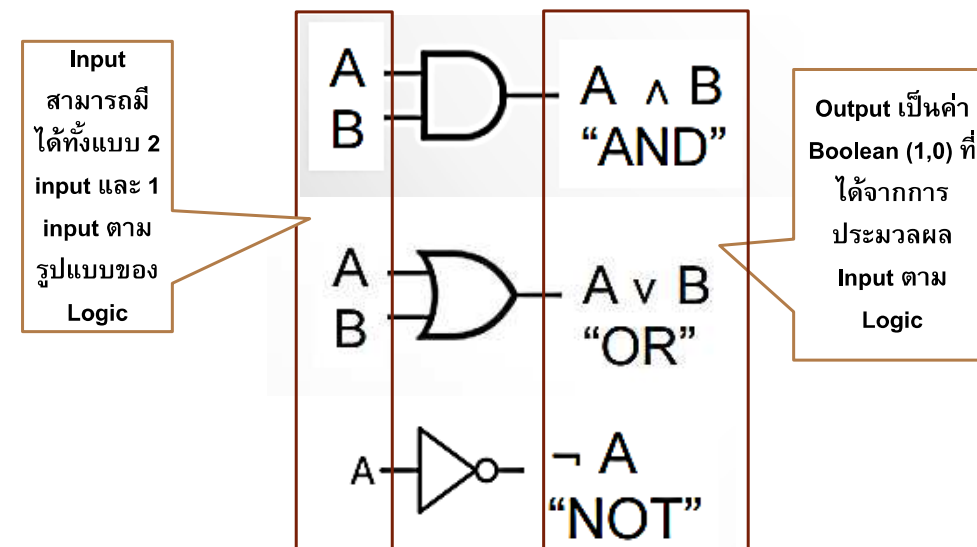
2

Logic Gates

- Gate เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ทำงานตาม **Boolean Logic** ในอุปกรณ์คอมพิวเตอร์ ปัจจุบัน Gate จำนวนมหาศาล จะบรรจุอยู่ใน Transistor 1 ตัว



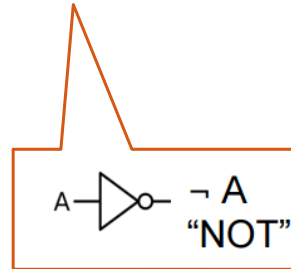
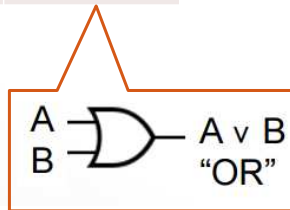
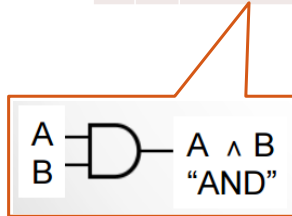
Logic Gates [2]



AND, OR, NOT Gates

A	B	$A \wedge B$ (A AND B) (conjunction)	$A \vee B$ (A or B) (disjunction)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

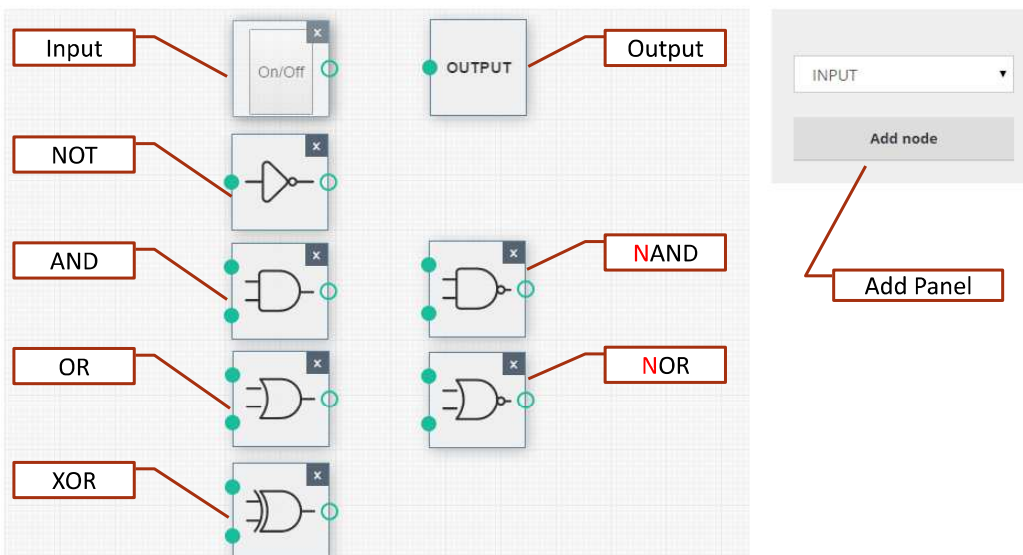
A	$\neg A$ (NOT A) (negation)
0	1
1	0



Computer Systems: A Programmer's Perspective, 2nd Edition

5

Academo.org

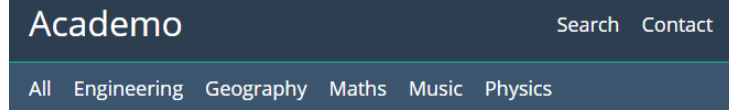


Computer Systems: A Programmer's Perspective, 2nd Edition

7

Online Logic Gate Simulator

- Academo.org <http://goo.gl/LMNg3G>



<http://academo.org/demos/logic-gate-simulator/>

Logic Gate Simulator

A free, simple, online logic gate simulator. Investigate the behaviour of AND, OR, NOT, NAND, NOR and XOR gates. Select gates from the dropdown list and click "add node" to add more gates. Drag from the hollow circles to the solid circles to make connections. Right click connections to delete them. See below for more detailed instructions.

Engineering Electronics Logic

6

Practice 0

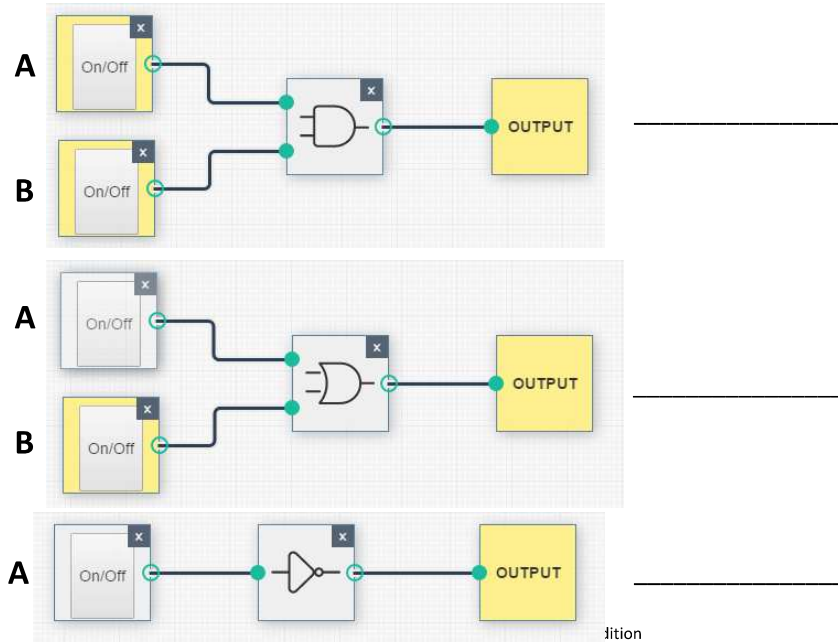
- จาก $A \wedge B$, $A \vee B$ และ $\neg A$ ให้ทำการวาดแผนวงจรเพื่อทดสอบว่า Truth table ด้านล่างเป็นจริง

A	B	$A \wedge B$ (A AND B) (conjunction)	$A \vee B$ (A or B) (disjunction)	A	$\neg A$ (NOT A) (negation)
0	0	0	0	0	1
0	1	0	1	1	0
1	0	0	1		
1	1	1	1		

Computer Systems: A Programmer's Perspective, 2nd Edition

8

Practice 0 (Key)

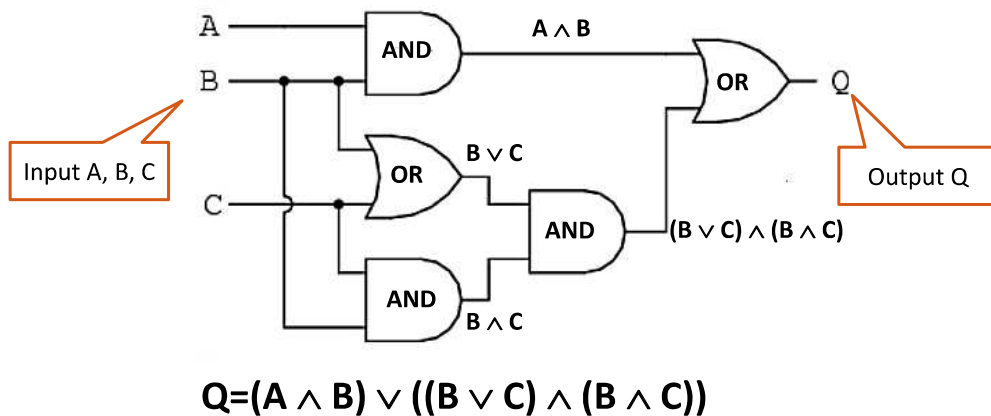


dition

9

Combinational Circuits

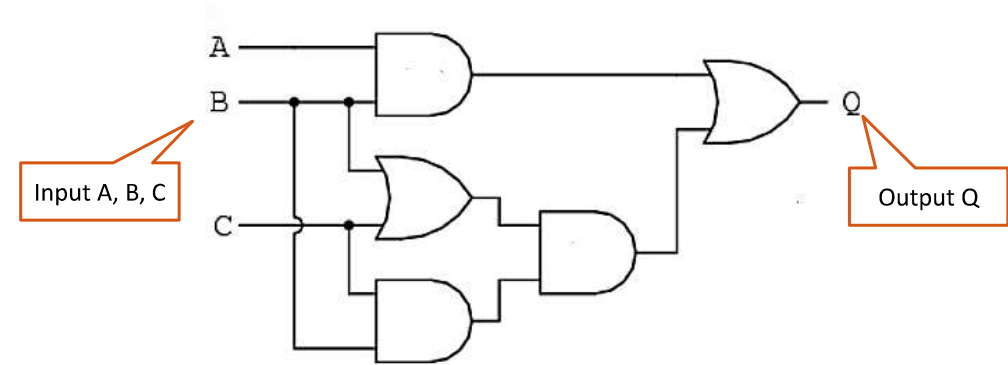
- ในการทำงานจริง จะมีการประกอบกันของ Gate หลายตัว เพื่อประมวลผลข้อมูลต่าง ๆ



$$Q = (A \wedge B) \vee ((B \vee C) \wedge (B \wedge C))$$

Combinational Circuits

- ในการทำงานจริง จะมีการประกอบกันของ Gate หลายตัว เพื่อประมวลผลข้อมูลต่าง ๆ

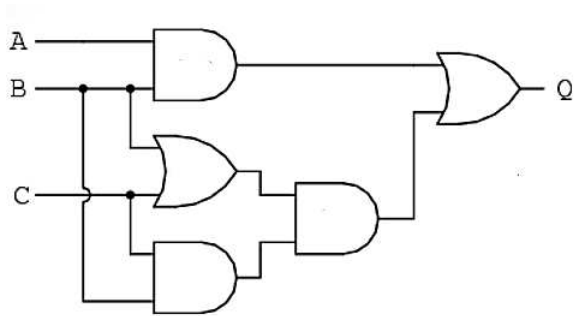


Manipulating circuits

- Boolean expressions
 - Circuit diagrams
 - Truth tables
- Equivalent notations
- ในการทำงานจริง อาจสามารถทำได้หลายรูปแบบ โดยที่ให้ผลลัพธ์เหมือนกัน
 - Boolean algebra ถูกใช้ในการออกแบบของแผงวงจรให้ทำงานได้ตามลอจิกที่ต้องการ และใช้จำนวน Gate น้อยที่สุด และมีความซับซ้อนน้อยที่สุด

Practice I

- จาก **Boolean Algebra** ให้เติมค่าที่เป็นไปได้ของ **Q** ใน **Truth Table** ตามเงื่อนไขของ **A, B, C** ตามที่กำหนด จากนั้นให้ใช้โปรแกรม **Logic Gate Simulator** เพื่อตรวจสอบคำตอบ



$$Q = (A \wedge B) \vee ((B \vee C) \wedge (B \wedge C))$$

A	B	C	Q
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Practice II

1. ชื่อ _____ รหัส นศ _____ ลำดับที่ _____
2. ชื่อ _____ รหัส นศ _____ ลำดับที่ _____
3. ชื่อ _____ รหัส นศ _____ ลำดับที่ _____

Lab04 204111 Sec _____

- $Q = (A \wedge B) \vee ((B \vee C) \wedge (B \wedge C))$ ให้ใช้กฎของ **Boolean Algebra** ในการลดรูป จากนั้นให้ใช้โปรแกรม **Logic Gate Simulator** เพื่อตรวจสอบคำตอบ

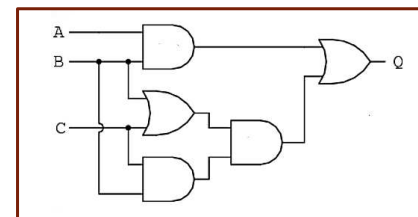
[illegible]

Laws for the Logical Operators \wedge and \vee

Rule	AND	OR
Commutative	$A \wedge B = B \wedge A$	$A \vee B = B \vee A$
Associative	$A \Delta B \Delta C = (A \wedge B) \wedge C = A \wedge (B \wedge C)$ $A \underline{\vee} B \underline{\vee} C = (A \vee B) \vee C = A \vee (B \vee C)$	
Distributive	$A \Delta (B \underline{\vee} C) = (A \wedge B) \vee (A \wedge C)$ $A \underline{\vee} (B \Delta C) = (A \vee B) \wedge (A \vee C)$	
Identity	$A \wedge 1 = A$	$A \vee 0 = A$
Dominance	$A \wedge 0 = 0$	$A \vee 1 = 1$
Idempotence	$A \wedge A = A$	$A \vee A = A$
Complementation	$A \wedge \neg A = 0$	$A \vee \neg A = 1$
Double Negation	$\neg \neg A = A$	

Practice II

- **Original**
- **Simplified**



A	B	C	$(A \wedge B) \vee ((B \vee C) \wedge (B \wedge C))$	
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

More gates (NAND, NOR, XOR)

- **NAND** (“not and”):

- $A \text{ nand } B = \text{not } (A \text{ and } B)$

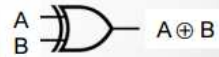
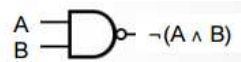
- **NOR** (“not or”):

- $A \text{ nor } B = \text{not } (A \text{ or } B)$

- **XOR** (“exclusive or”):

- $A \text{ xor } B = (A \text{ and not } B) \text{ or } (B \text{ and not } A)$

A	B	A NAND B	A NOR B	A XOR B
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	0



Computer Systems: A Programmer's Perspective, 2nd Edition

17

Practice III

- (Lab05_1_5XXXXXXXXX.py) ยูนเป็นผีเสื้อที่เกิดจากการโหมงานหนัก หมออิมจึงแนะนำยูนว่าควรหยุดพักผ่อนในช่วงสงกรานต์ที่จะถึง ให้เขียนฟังก์ชัน `count_down_to_songkran(d, m, y)` เพื่อช่วยยูนคำนวณว่าวันที่กำหนดห่างจากวันสงกรานต์ (13 เมษายน) ครั้จะไปถึงกี่วัน (แน่นอนว่าฟังก์ชันนี้ต้องใช้ได้ในป้อธิกสุรทินด้วย)

Hint: เทียบคำตอบได้จากเว็บ <http://www.timeanddate.com/date/duration.html>

Input	Output
1 2 2016	72
13 4 2016	0

Computer Systems: A Programmer's Perspective, 2nd Edition

19

Conclusions

- Identify basic gates.
- Describe the behavior of a gate or circuit using Boolean expressions, truth tables, and logic diagrams.
- Transform one Boolean expression into another given the laws of Boolean algebra.

Computer Systems: A Programmer's Perspective, 2nd Edition

18

Practice IV

- (Lab05_2_5XXXXXXXXX.py) ทันไดนั้นเองเจ้โทรศัพท์ท่่วนเข้ามาขอให้ยูนรับงานใหม่ มีกำหนดส่งภายใน x วัน ให้เขียนฟังก์ชัน `Boolean take_this_job(d, m, y, x)` เพื่อช่วยยูนตัดสินใจว่าจะรับงานจากเจ้หรือไม่

Input	Output
1 2 2016 75	False
1 2 2016 45	True

Computer Systems: A Programmer's Perspective, 2nd Edition

20