

ปฏิบัติการวงจรดิจิทัลและพื้นฐานการทำงานของคอมพิวเตอร์

อ. ดร. วาธิส ลีลาภัทร

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

เนื้อหา

- หลักการพื้นฐานของคอมพิวเตอร์
- ระบบเลขฐาน
- ตรรกศาสตร์บูลีนเบื้องต้น
- วงจรดิจิทัลเบื้องต้น
- วงจรบวกและลบเลข

หลักการพื้นฐานของคอมพิวเตอร์

คอมพิวเตอร์ถูกออกแบบและสร้างขึ้นบนหลักการ 2 ข้อได้แก่

1. คำสั่ง (Instruction) และข้อมูล (Data) สำหรับคอมพิวเตอร์
จะต้องอยู่ในรูปของตัวเลขเท่านั้น
2. คำสั่งและข้อมูลจะเก็บในหน่วยความจำในรูปของตัวเลข
การเขียนและอ่านก็จะอยู่ในรูปของตัวเลข

ระบบเลขฐาน

เลขฐาน (Base Number) เป็นการเขียนแทนค่าของตัวเลขในรูปแบบที่ต่างกันตามฐาน (base) ดังนี้:

- เลขฐาน 2 แต่ละหลักจะมีได้ 2 ค่าได้แก่ 0, 1 (Binary number)
- เลขฐาน 3 แต่ละหลักจะมีได้ 3 ค่าได้แก่ 0, 1, 2
- เลขฐาน 8 แต่ละหลักจะมีได้ 8 ค่าได้แก่ 0, 1, 2, 3, 4, 5, 6, 7 (Octal number)
- เลขฐาน 10 แต่ละหลักจะมีได้ 10 ค่าได้แก่ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (Decimal number)
- เลขฐาน 16 แต่ละหลักจะมีได้ 16 ค่าได้แก่ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F (Hexadecimal number)

ระบบเลขฐาน 10

แต่ละหลักจะมีตัวเลขที่เป็นไปได้ 10 ค่า (0-9)

0	10	20		90	100
1	11	21		91	101
2	12	22		92	102
3	13	23		93	103
4	14	24	94	104
5	15	25		95	105
6	16	26		96	106
7	17	27		97	107
8	18	28		98	108
9	19	29		99	109

ระบบเลขฐาน 10

$$\begin{aligned} 641_{10} &= 600 + 40 + 1 = 6*100 + 4*10 + 1*1 \\ &= 6*(10)^2 + 4*(10)^1 + 1*(10)^0 \end{aligned}$$

Diagram illustrating the expansion of the decimal number 641 into its positional values. Arrows point from the coefficients (6, 4, 1) in the second equation to the text "ค่าประจำหลัก" (Place Value), and from the bases (10, 10, 10) to the text "ฐาน" (Base).

$$35607_{10} = 3*(10)^4 + 5*(10)^3 + 6*(10)^2 + 0*(10)^1 + 7*(10)^0$$

ระบบเลขฐาน 2

แต่ละหลักจะมีตัวเลขที่เป็นไปได้ 2 ค่า (0, 1)

0	10	100	110	1000	
1	11	101	111	1001

หลักของเลขฐาน 2 เรียกว่า binary digit (bit)

Bit ซ้ายสุดจะมีค่าสูงที่สุด หรือ มีนัยสำคัญ (Significant) สูงที่สุด

Bit ขวาสุดจะมีค่าต่ำที่สุด หรือ มีนัยสำคัญต่ำที่สุด

MSB	10101100	LSB
<u>M</u> ost <u>S</u> ignificant <u>B</u> it		<u>L</u> east <u>S</u> ignificant <u>B</u> it

ระบบเลขฐาน 2

แต่ละหลักจะมีตัวเลขที่เป็นไปได้ 2 ค่า (0, 1)

0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9

ระบบเลขฐาน 2

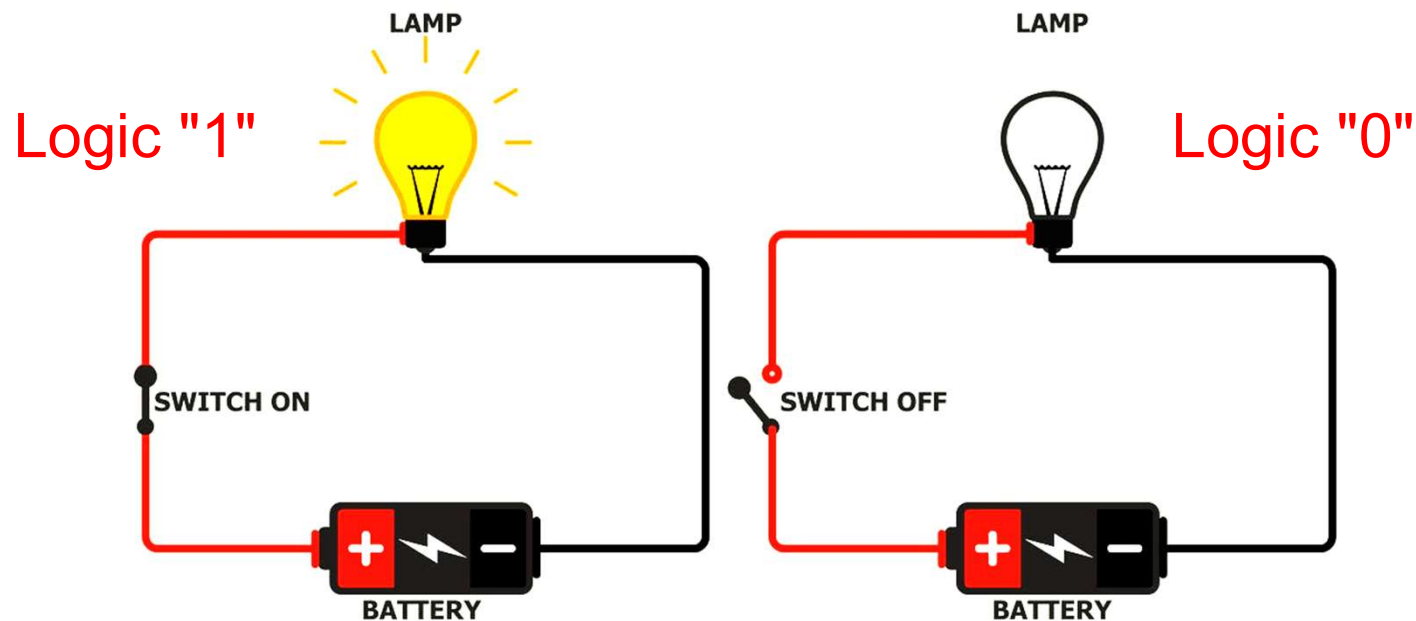
$$1101_2 = 1*(2)^3 + 1*(2)^2 + 0*(2)^1 + 1*(2)^0 = 8 + 4 + 0 + 1 = 13_{10}$$

ฐาน

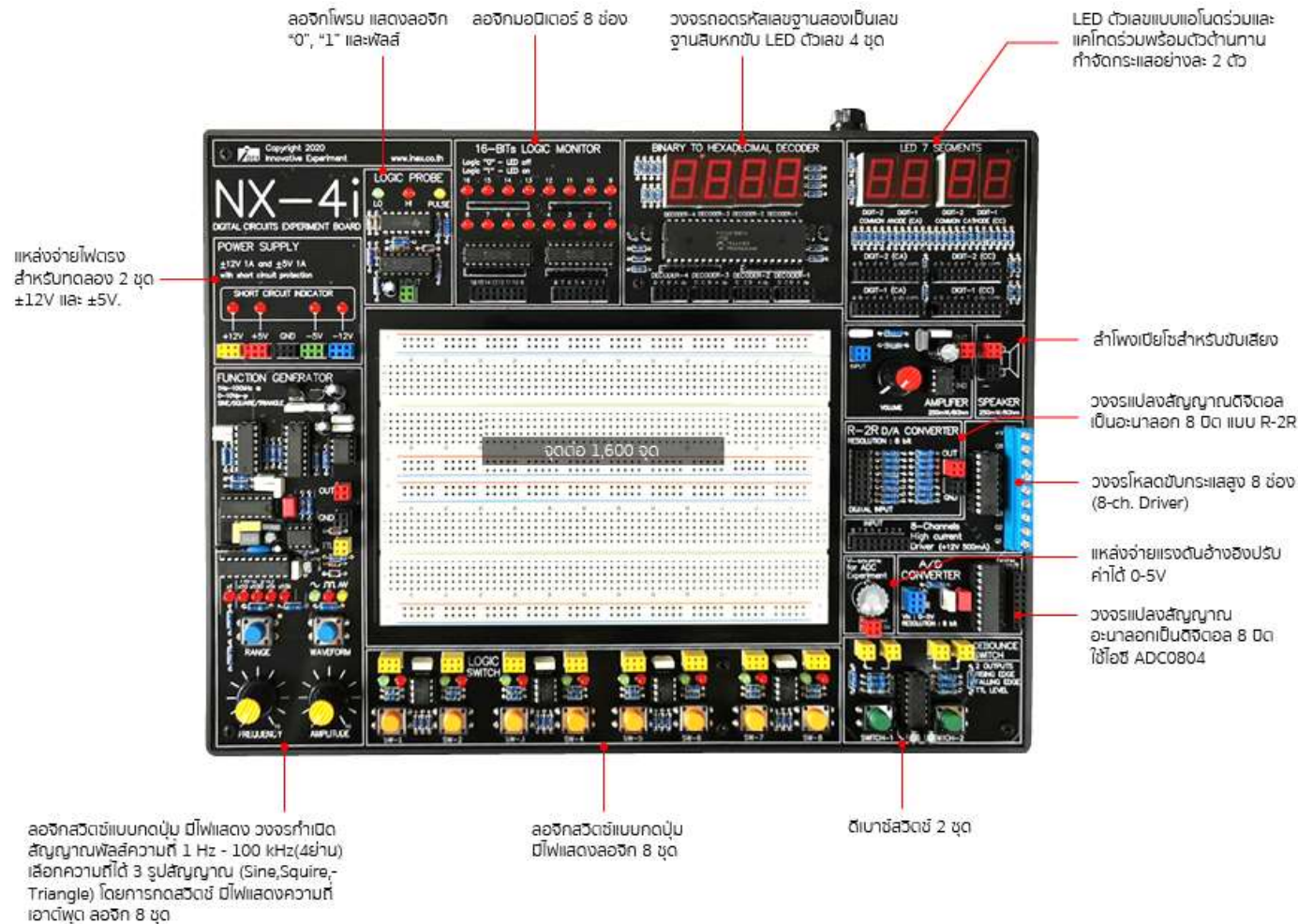
$$111_2 = 1*(2)^2 + 1*(2)^1 + 1*(2)^0 = 4 + 2 + 1 = 7$$

ระบบเลขฐาน 2

- ระบบเลขฐาน 2 สามารถสร้างเป็นวงจรไฟฟ้าได้ง่าย
- เลข 0 แทนด้วย การไม่มีแรงดันไฟฟ้า (0 Volt)
- เลข 1 แทนด้วย การมีแรงดันไฟฟ้า (5 Volt)



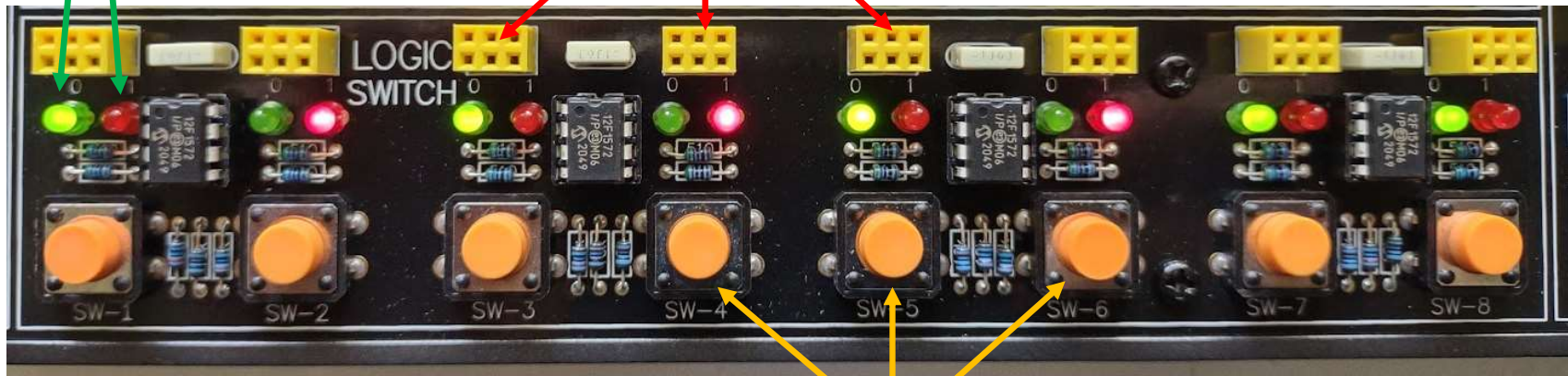
Lab 1: แนะนำชุดทดลองวงจรดิจิทัล



Lab 1: แนะนำชุดทดลองวงจรดิจิทัล

หลอดไฟแสดงสถานะ (● = 0, ● = 1)

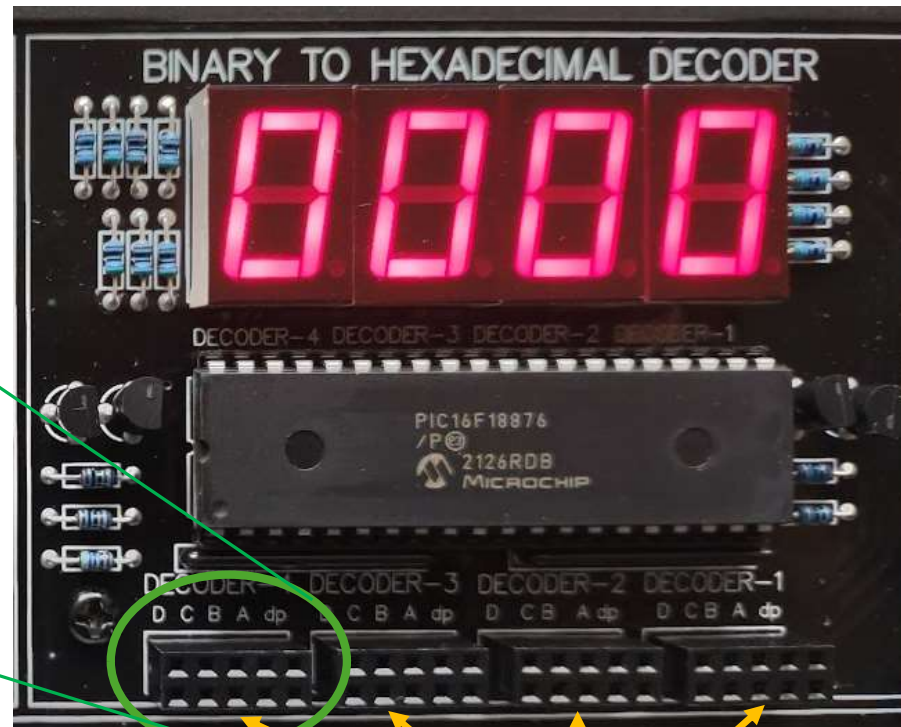
จุดต่อสายไฟ



ปุ่มกดเลือกสถานะ (8 ชุด แยกอิสระ)

Lab 1: แนะนำชุดทดลองวงจรดิจิทัล

จอแสดงผลเลขฐาน 16



MSB D C B A LSB

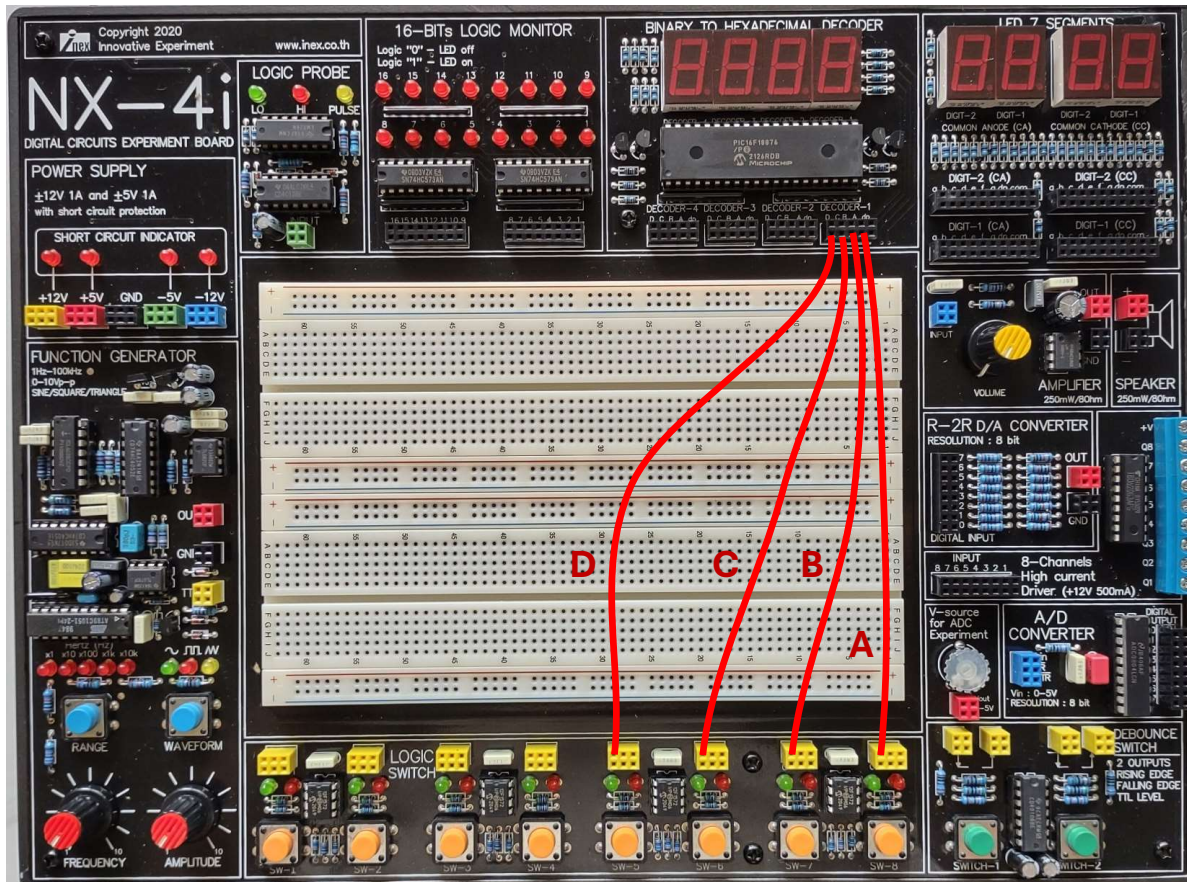
2^3 2^2 2^1 2^0

0 1 1 1 = 7

จุดต่อสัญญาณเลขฐาน 2

Lab 2: การนับเลขฐาน 2

ใช้สายไฟ 4 เส้น ต่อเข้ากับปุ่มกดและจอแสดงตัวเลข



● ● ● ●	→ 0000 = 0
● ● ● ●	→ 0001 = 1
● ● ● ●	→ 0010 = 2
● ● ● ●	→ 0011 = 3
● ● ● ●	→ 0100 = 4
● ● ● ●	→ 0101 = 5
● ● ● ●	→ 0110 = 6
● ● ● ●	→ 0111 = 7
● ● ● ●	→ 1000 = 8
● ● ● ●	→ 1001 = 9

การแปลงเลขฐาน 10 เป็นฐาน 2

การแปลงจากเลขฐาน 10 ให้เป็นฐานใดๆ จะใช้การหาร

ตัวอย่าง

แปลง 13_{10} ให้เป็นฐาน 2

13 หาร 2 ได้ 6 เศษ 1

6 หาร 2 ได้ 3 เศษ 0

3 หาร 2 ได้ 1 เศษ 1

นำผลลัพธ์สุดท้ายและเศษที่ได้ไล่จากบรรทัดสุดท้ายย้อนกลับไปบรรทัดแรกจะได้ 1101

ซึ่งเป็นเลขฐาน 2 ที่มีค่า 13_{10}

การบวกเลขฐาน 2

- ใช้หลักการเดียวกันกับการบวกเลขฐาน 10
- บวกจากหลักทางขวา ไปทางซ้าย
- ถ้าผลบวกเกินค่าสูงสุด ให้ทดไปหลักถัดไปทางขวา

$$0 + 0 = 0 \text{ ทด } 0$$

$$0 + 1 = 1 \text{ ทด } 0$$

$$1 + 0 = 1 \text{ ทด } 0$$

$$1 + 1 = 0 \text{ ทด } 1$$

$$\begin{array}{r} \textcolor{blue}{1} \textcolor{blue}{1} \\ 101 \\ 011 \\ \hline \mathbf{1000} \end{array} +$$

$$\begin{array}{r} \textcolor{blue}{1} \textcolor{blue}{1} \textcolor{blue}{1} \\ 1011 \\ 0111 \\ \hline \mathbf{10010} \end{array} +$$

ตรรกศาสตร์บูลีน (Boolean Logic)

- ค่าความจริง (Truth) ในเชิงตรรกศาสตร์นิยามไว้ 2 ค่าได้แก่ **จริง (True)** กับ **เท็จ (False)**
- ค่าความจริงอาจแทนด้วยตัวเลข (True = 1) และ (False = 0)
- ตัวกระทำ (operator) สำหรับค่าความจริง มีได้ดังนี้

NOT (\sim , \neg)

นิเสธ

$$\sim 0 = 1$$

$$\sim 1 = 0$$

AND (\wedge , \bullet)

และ

$$0 \text{ AND } 0 = 0$$

$$1 \text{ AND } 0 = 0$$

$$0 \text{ AND } 1 = 0$$

$$1 \text{ AND } 1 = 1$$

OR (\vee , $+$)

หรือ

$$0 \text{ OR } 0 = 0$$

$$1 \text{ OR } 0 = 1$$

$$0 \text{ OR } 1 = 1$$

$$1 \text{ OR } 1 = 1$$

EXCLUSIVE-OR (XOR: \oplus)

$$0 \text{ XOR } 0 = 0$$

$$1 \text{ XOR } 0 = 1$$

$$0 \text{ XOR } 1 = 1$$

$$1 \text{ XOR } 1 = 0$$

$$A = 01011001, \sim A = 10100110$$

อาจใช้ \overline{A} แทน $\sim A$ ได้

$$1011 \text{ AND } 1110 = 1010$$

$$1011 \text{ OR } 1110 = 1111$$

$$1011 \text{ XOR } 1110 = 0101$$

วงจรดิจิทัลเบื้องต้น

- วงจรดิจิทัลเป็นวงจรอิเล็กทรอนิกส์ที่ใช้สัญญาณไฟฟ้า 2 ระดับ เพื่อแทนสถานะ "0" และ "1"
- วงจรดิจิทัลประกอบด้วย เกต (Gate) เชื่อมต่อเข้าด้วยกัน

1. AND gate ($c = a \cdot b$)



a	b	$c = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

2. OR gate ($c = a + b$)



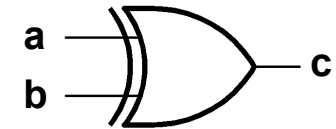
a	b	$c = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

3. Inverter ($c = \bar{a}$)



a	$c = \bar{a}$
0	1
1	0

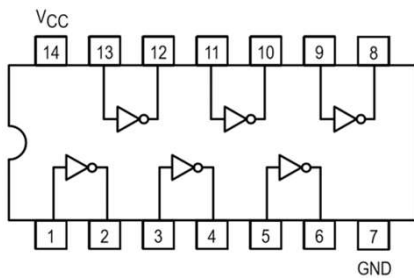
4. XOR gate ($c = a \oplus b$)



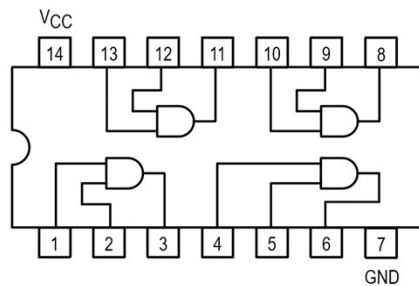
a	b	$c = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Lab 2: วงจรดิจิทัลพื้นฐาน

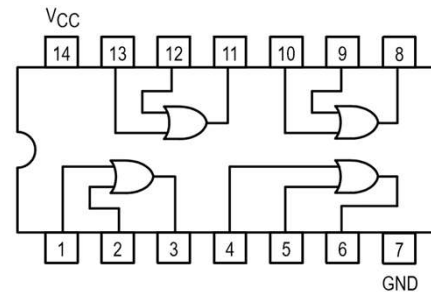
- เกตถูกสร้างและบรรจุในวงจรรวม (Integrated Circuit: IC)
- IC จะมีหมายเลขกำกับ เพื่อใช้บอกรายละเอียดวงจรที่อยู่ภายใน



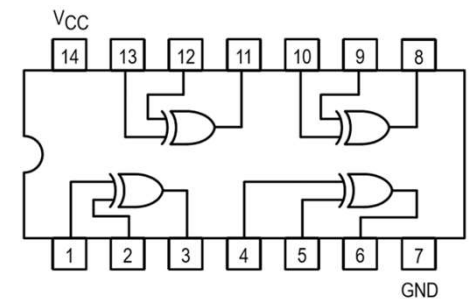
74LS04 Inverters



74LS08 AND Gates



74LS32 OR Gates



74LS86 XOR Gates



สมการบูลีนพื้นฐาน

- สมการบูลีน (Boolean Expression) คือสมการที่แสดงความสัมพันธ์ระหว่างตัวแปรและตัวกระทำทางตรรก (Operator) เช่น

$$A = BC + DE$$

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

- ผลลัพธ์ของสมการบูลีนหาได้จากการแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปรทางขวามือของสมการ
- สามารถใช้ตารางค่าความจริง (Truth table) เพื่ออธิบายสมการบูลีนได้

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

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

วงจรบวกเลข

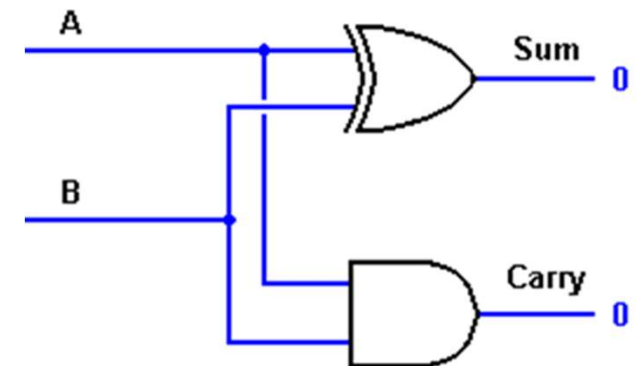
- วงจรดิจิทัลบวกเลขจะใช้หลักการพื้นฐานมาจากการบวกเลขฐาน 2 ที่ละหลัก
- สมการบูลีนของวงจรบวกเลขที่ละหลักได้จากตารางค่าความจริง
- กำหนดให้ **a** และ **b** คือตัวตั้งและตัวบวก, **Sum** คือผลบวก และ **Cout** คือตัวทด ทั้งหมดมีขนาด 1 bit

a	b	Sum
0	0	0
0	1	1
1	0	1
1	1	0

$$\text{Sum} = a \oplus b \quad \text{หรือ} \quad a \text{ xor } b$$

a	b	Cout
0	0	0
0	1	0
1	0	0
1	1	1

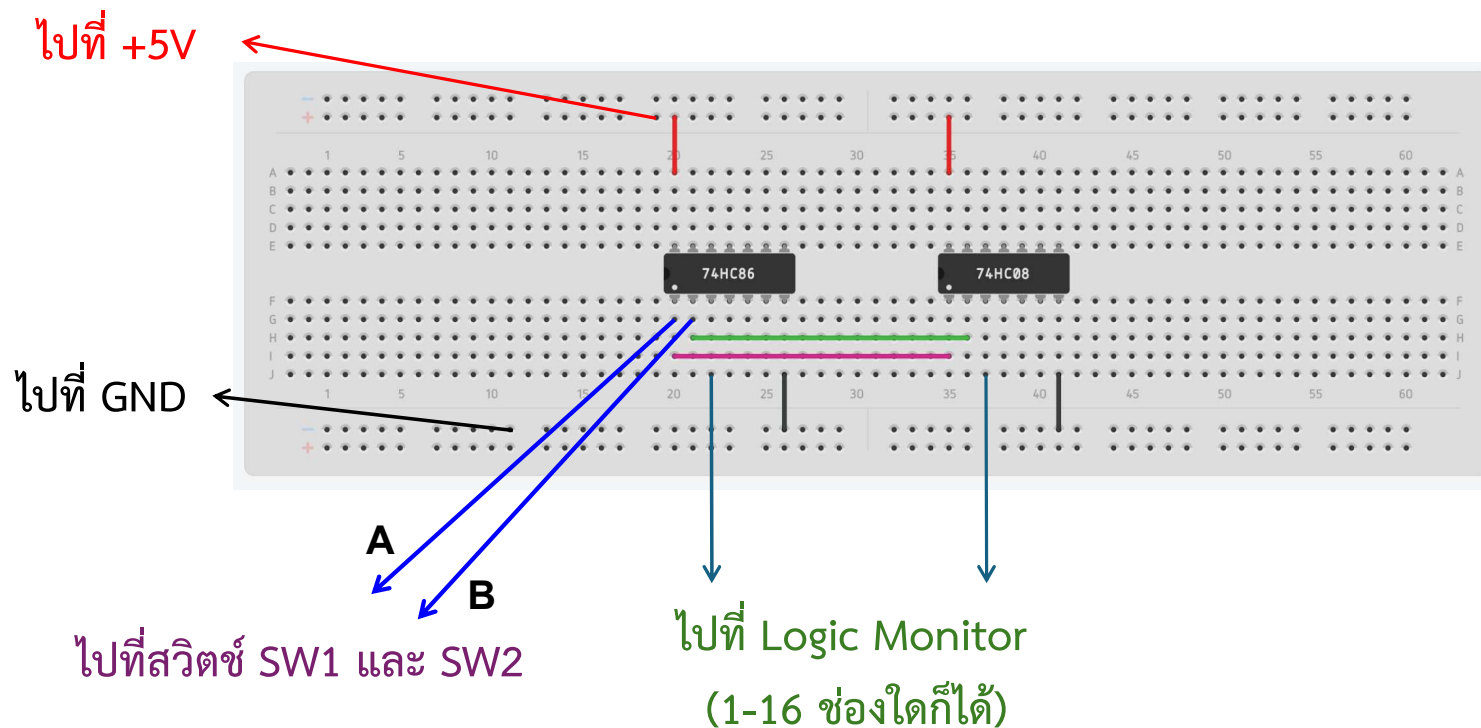
$$\text{Cout} = a \cdot b \quad \text{หรือ} \quad a \text{ and } b$$



วงจร Half-adder

Lab 3: วงจร Half-adder

- วงจร half-adder ประกอบด้วย XOR gate และ AND gate อย่างละ 1 ตัว
- IC จะมีหมายเลขกำกับ เพื่อใช้บอกรายละเอียดวงจรที่อยู่ภายใน



วงจรบวกเลขที่สมบูรณ์

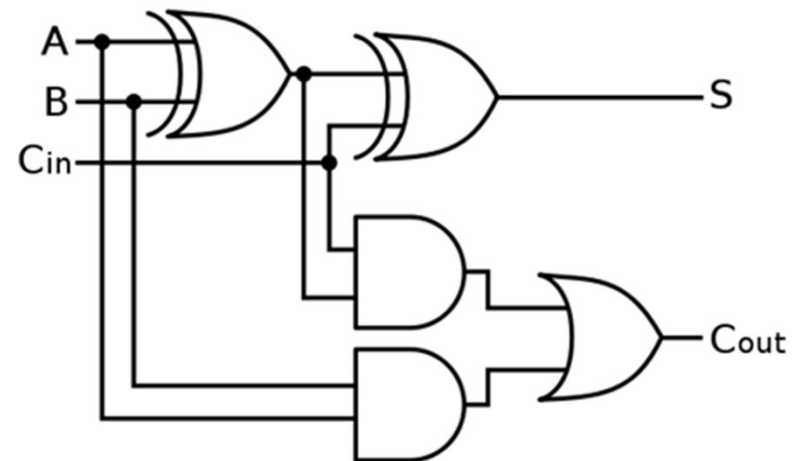
- วงจรดิจิทัลบวกเลขที่สมบูรณ์จะสามารถบวกเลขฐาน 2 ได้หลายหลัก เรียกว่าวงจร Full-adder
- กำหนดให้ **a** และ **b** คือตัวตั้งและตัวบวก, **Sum** คือผลบวก **Cout** คือตัวทด ทั้งหมดมีขนาด 1 bit

a	b	Cin	Sum
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\text{Sum} = a \oplus b \oplus \text{Cin}$$

a	b	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

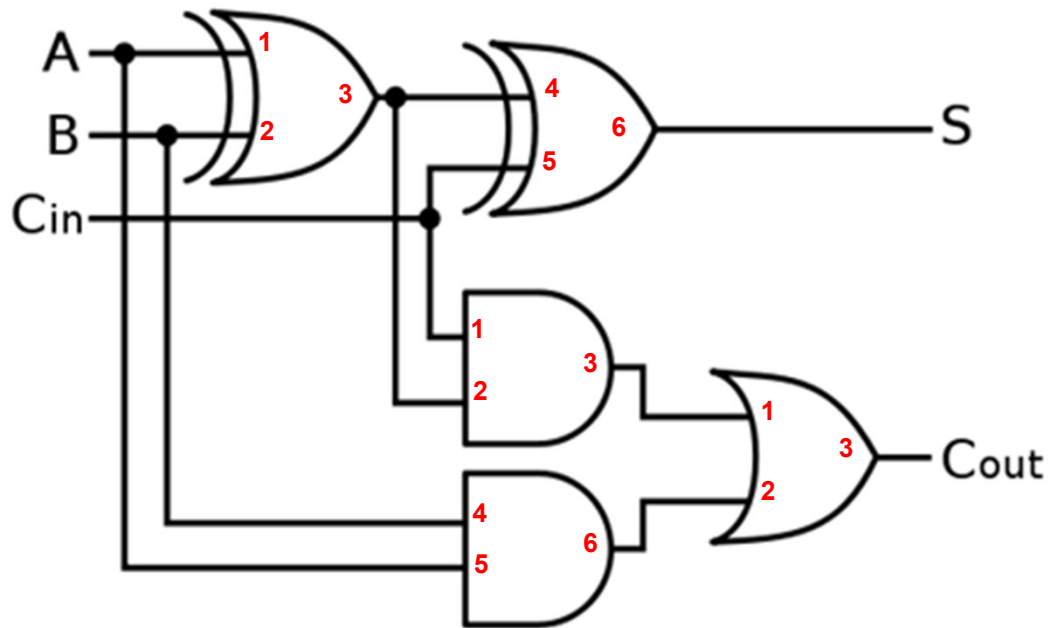
$$\text{Cout} = (a \oplus b) \cdot \text{Cin} + a \cdot b$$



วงจร Full-adder 1 บิต

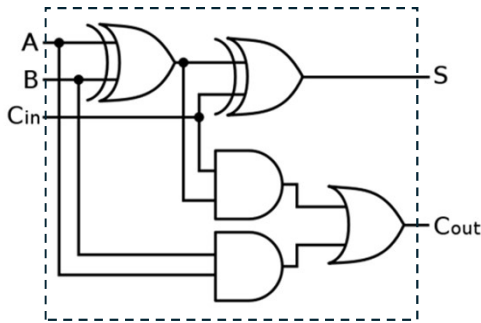
Lab 4: วงจร 1 Bit Full-adder

- วงจร half-adder ประกอบด้วย XOR gate และ AND gate อย่างละ 1 ตัว
- IC จะมีหมายเลขกำกับ เพื่อใช้บอกรายละเอียดวงจรที่อยู่ภายใน

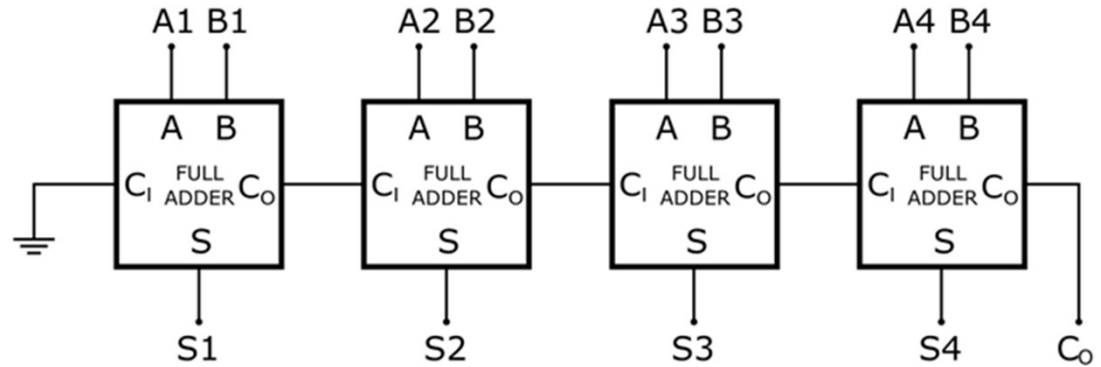


วงจรบวกเลขฐาน 2 หลายหลัก

- วงจรดิจิทัลบวกเลขที่สมบูรณ์จะสามารถบวกเลขฐาน 2 ได้หลายหลัก เรียกว่าวงจร Full-adder
- กำหนดให้ **a** และ **b** คือตัวตั้งและตัวบวก, **Sum** คือผลบวก และ **Cout** คือตัวทด ทั้งหมดมีขนาด 1 bit



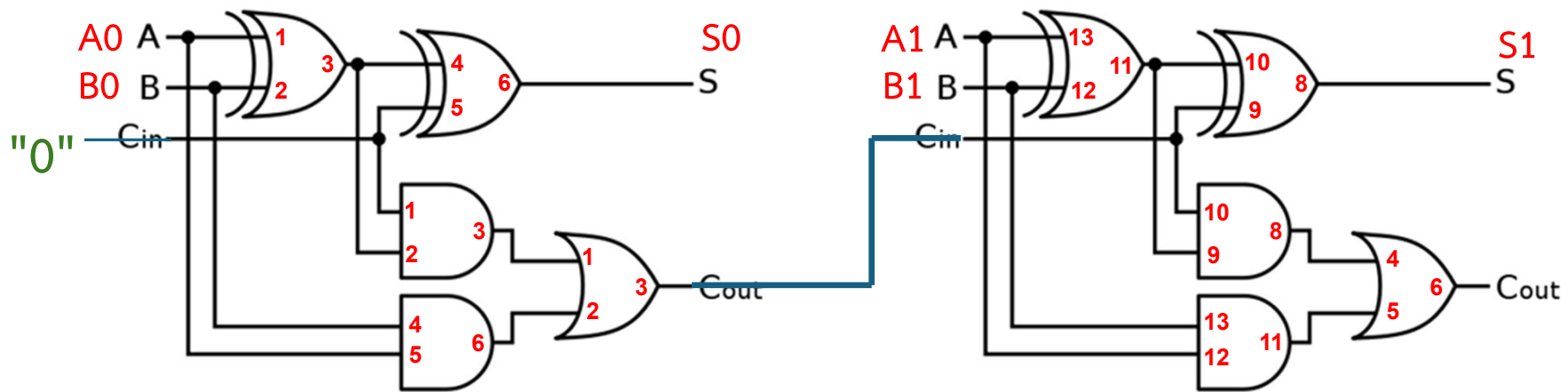
วงจร Full-adder 1 บิต



วงจร Full-adder 4 บิต

Lab 4: วงจร 2 Bit Full-adder

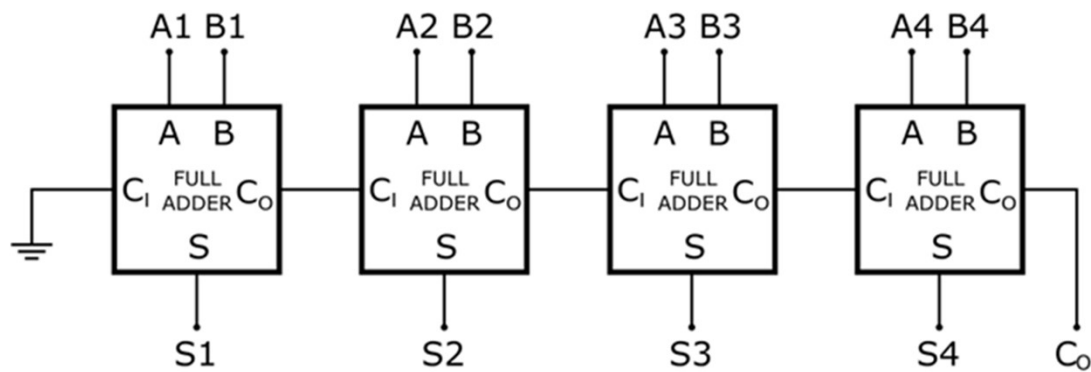
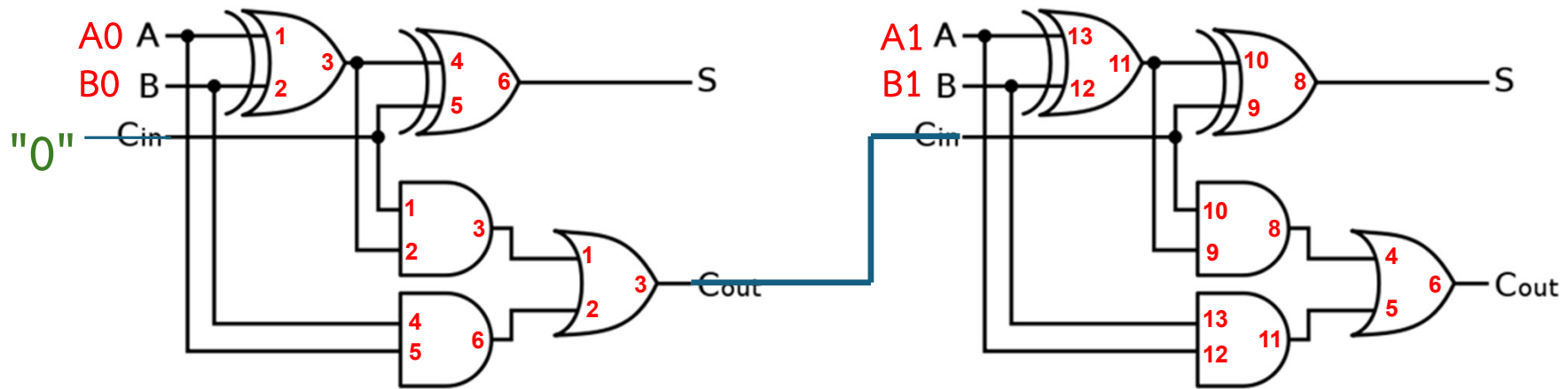
- วงจร 2 bit full-adder ประกอบด้วย full-adder 2 ชุด ต่อพ่วงกัน โดยต่อ Cout ของวงจรบวก LSB ไปเข้าที่ Cin ของวงจรบวก MSB



- ใช้ 2 XOR gate, 2 AND gate และ 1 OR gate

Lab 5: วงจร 4 Bit Full-adder

- ใช้ Tinker CAD จำลองการทำงาน เพื่อความสะดวก



การแทนจำนวนเต็มลบด้วยเลขฐาน 2

เลขฐาน 2 มี 2 ชนิด

- ไม่มีเครื่องหมาย (Unsigned binary number)
- มีเครื่องหมาย (Signed binary number)
 - 1's Complement
 - 2's Complement
- เลขฐาน 2 ที่ไม่มีเครื่องหมายแทนได้เฉพาะจำนวนบวก เลขฐาน 2 ที่มีเครื่องหมายแทนได้ทั้งจำนวนบวกและลบดังนี้

ฐาน 10	Unsigned	1's Complement	2's Complement
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
0	0000	0000	0000
-1	-	1110	1111
-2	-	1101	1110
-3	-	1100	1101

การแทนจำนวนเต็มลบด้วยเลขฐาน 2

การแปลงจำนวนเต็มลบ (ฐาน 2) ให้อยู่ในรูป 1's complement และ 2's complement

กำหนดให้ X เป็นจำนวนเต็มลบ การหา 1's complement ของ X มีขั้นตอนดังนี้

1. กำหนดจำนวนบิตของตัวเลขที่ต้องการ
2. หาเลขฐาน 2 ของ $|X|$
3. หานิเสธของแต่ละบิต ผลลัพธ์ที่ได้จะเป็น 1's complement ของ X
4. นำผลลัพธ์ในข้อ 3 บวกด้วย 1 จะได้เป็น 2's complement ของ X

การแทนจำนวนเต็มลบด้วยเลขฐาน 2

ตัวอย่าง จงหา 1's complement และ 2's complement ของ -9_{10} , และ -13_{10}

$|-9| = 9 = 1001_2$ กำหนดให้ใช้ตัวเลขฐาน 2 จำนวน **8 บิต** ดังนั้น $1001_2 = 00001001_2$

หาบิตของแต่ละบิต จะได้ $00001001 \rightarrow 11110110$ ดังนั้น $-9 = 11110110_2$ (1's complement)

$11110110_2 + 1 = \mathbf{11110111_2}$ (2's complement) ซึ่งเลขฐาน 2 ทั้งสอง มีค่าเท่ากับ -9

$|-13| = 13 = 1101$ กำหนดให้ใช้ตัวเลขฐาน 2 จำนวน 8 บิต ดังนั้น $1101 = 00001101$

จะได้ $-13 = 11110010$ (1's complement) และ $\mathbf{11110011}$ (2's complement)

การลบในระบบเลขฐาน 2

- การลบเลข อาจเขียนให้อยู่ในรูปการบวกได้ เช่น $Z = X - Y \rightarrow Z = X + (-Y)$
- หลักการคือ บวก X ด้วยค่า 2's Complement ของ Y จะได้ผลลัพธ์เท่ากับ $X - Y$

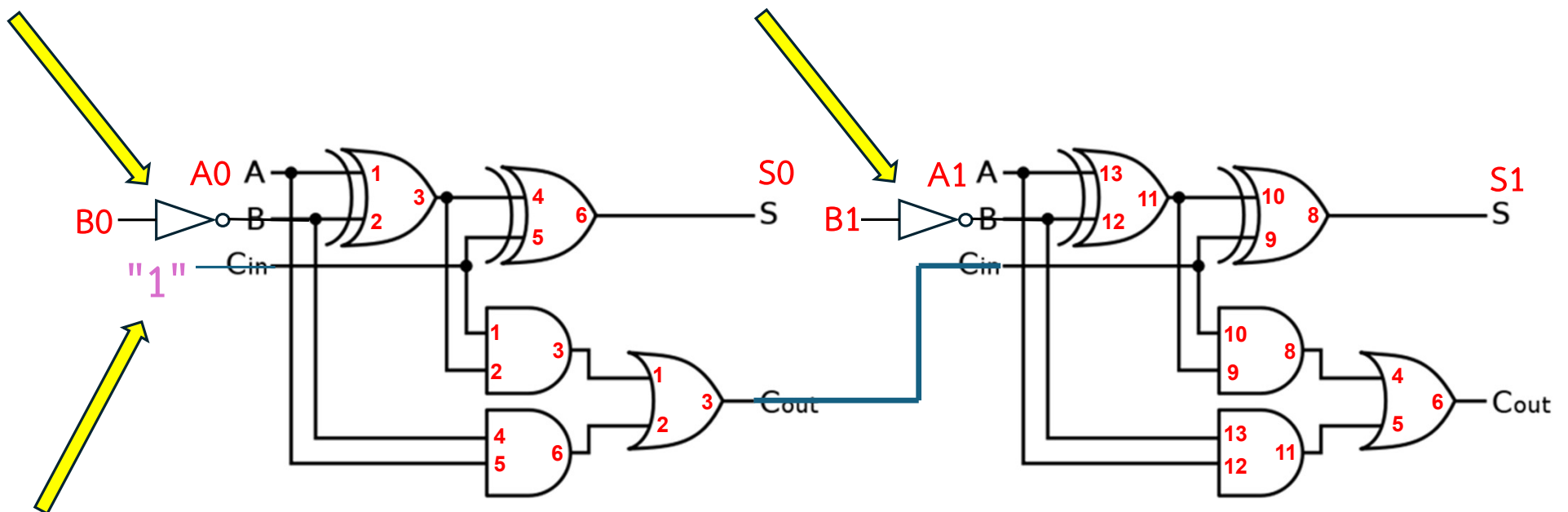
ตัวอย่าง $13 - 9$ ในระบบเลขฐาน 2 ขนาด 4 บิต

$13 = 1101$, $-9 \rightarrow 9 = 1001$, 1's complement = 0110,
2's complement = 0111

$$13 - 9 = 13 + (-9) = 1101 + 0111 = (1)0100 = 0100 = 4_{10}$$

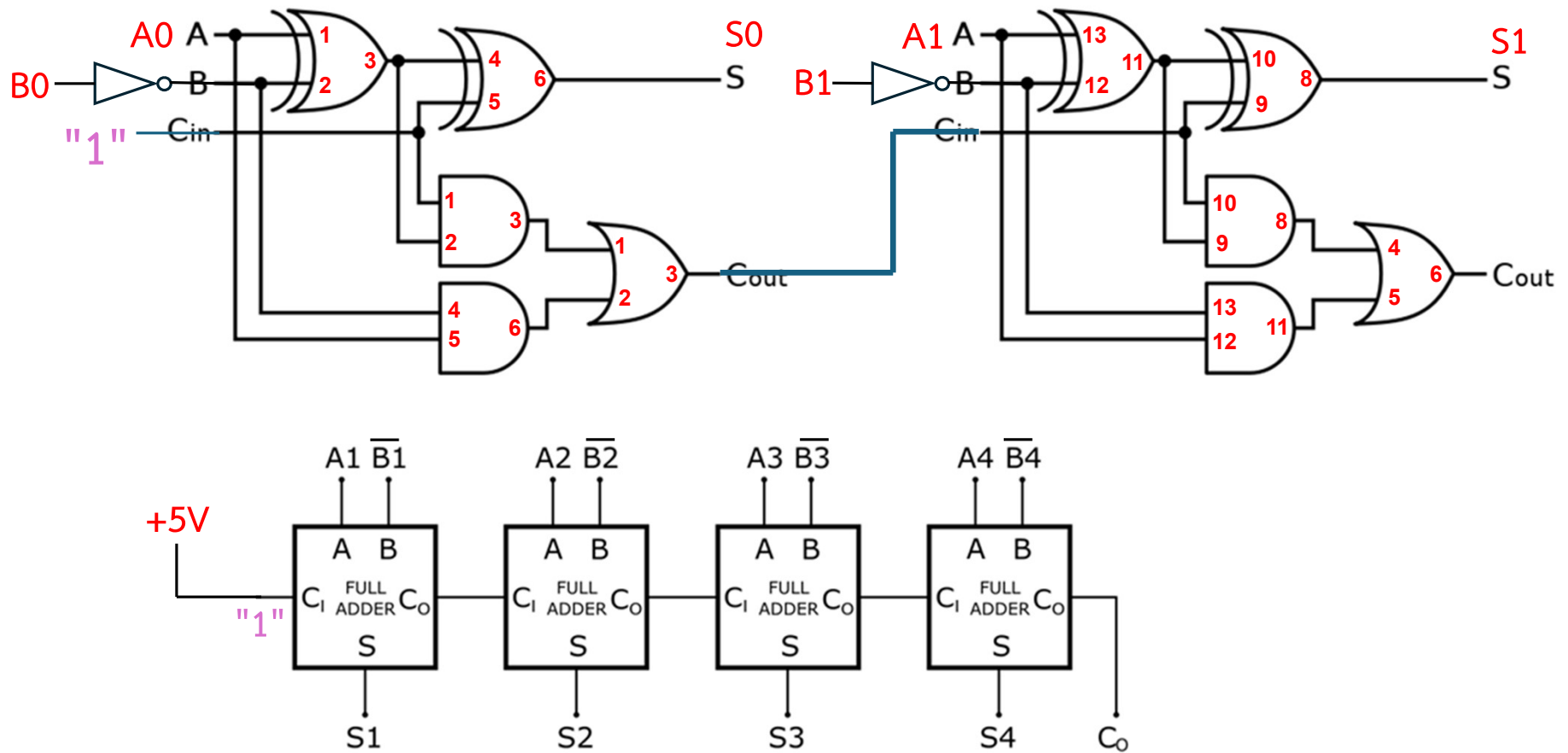
วงจรดิจิทัลลบเลขฐาน 2

- การลบเลขคือการบวกตัวตั้งด้วยค่า 2's Complement ของตัวลบ
- 2's Complement ได้จาก การหาอินเวอร์สของทุกบิต จากนั้นบวกด้วย 1
- การหาอินเวอร์สคือการเพิ่ม NOT gate และการบวก 1 ทำได้โดยกำหนดให้ Cin ของ LSB มีค่าเป็น 1



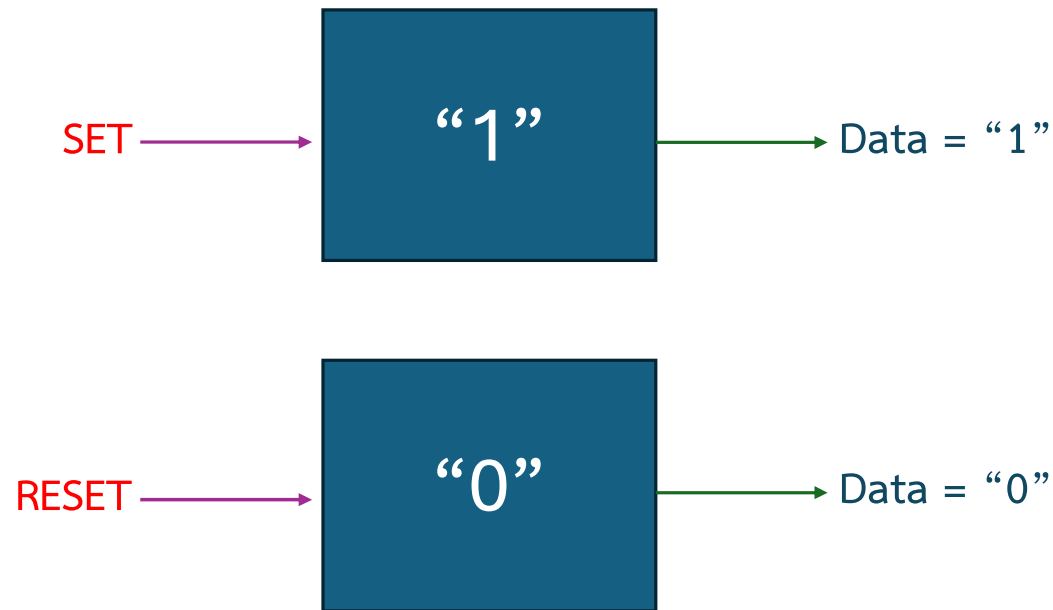
Lab 6: วงจร 4 Bit Subtractor

- ใช้ Tinker CAD จำลองการทำงาน เพื่อความสะดวก



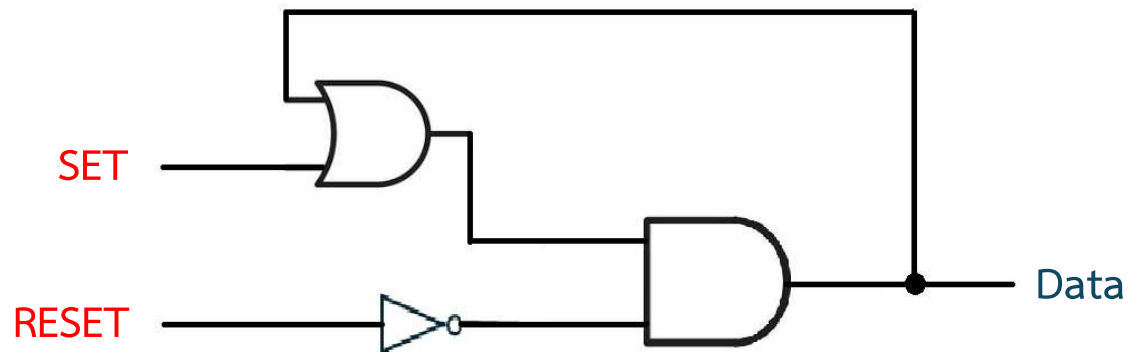
หน่วยความจำ

- หน่วยความจำ ทำหน้าที่คงค่าข้อมูลที่กำหนด จนกว่าจะมีการเปลี่ยนค่าข้อมูลใหม่
- การกำหนดข้อมูลในหน่วยความจำให้เป็น “1” เรียกว่า SET
- การกำหนดข้อมูลในหน่วยความจำให้เป็น “0” เรียกว่า RESET



Lab 7: วงจรหน่วยความจำ 1 บิต

- หน่วยความจำ ทำหน้าที่คงค่าข้อมูลที่กำหนด จนกว่าจะมีการเปลี่ยนค่าข้อมูลใหม่



	SET	RESET	Data	
↷	1	0	1	กำหนดข้อมูลเป็น 1
↘	0	0	1	จำ (Latch)
↷	0	1	0	กำหนดข้อมูลเป็น 0
↘	0	0	0	จำ (Latch)