



DIGITAL DESIGN
School of Computer Engineering

Suranaree University
of Technology

Lecture 1 Review

Paramate Horkaew

School of Computer Engineering, Institute of Engineering
Suranaree University of Technology



Previous Lecture

- Discrete Systems - **Digitization**
- Symbol Representation - **Encoding**
- Conversion of Base-2, 8, 16 Numbers
- Numeric Quantization - **Rounding and Truncation**
- Simple Binary Arithmetic Operations
- Half and Full Adder Block Diagrams



Base-2 Numbers

Cipher คือตัวเลขที่เป็นสมาชิกของเซต ที่ประกอบขึ้นเป็นเลขฐานใด ๆ ตัวอย่างเช่น เลขฐาน 2 มี Cipher คือ {0, 1}

เลขฐาน เป็นการ เข้ารหัส (Encoding) แบบอ้างอิงตำแหน่ง (Positional Code) ของกลุ่มตัวเลข ซึ่งมีความสามารถแสดงได้ด้วยสมการรหัส โดยที่ P และ B แทนตัวเลขแต่ละหลัก และ ฐาน

$$D = \sum_{i=-I^-}^{I^+} P_i B^i$$

ตัวอย่าง เลข 101.01 ฐาน 2 มีค่าเท่ากับ

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = (4 + 1 + 0.25)_{10} = (5.25)_{10}$$



Conversion of Fractions

Repeated Subtraction

คล้ายกับการแปลงเลขจำนวนเต็ม

Repeated Multiplication

- คูณเลขเศษส่วนฐาน 10 ด้วยสอง ผลลัพธ์ที่ได้จะมีค่า $0 \leq R \leq 2$
- ถ้าผลลัพธ์ที่ได้มีค่า มากกว่าหรือเท่ากับ 1 ค่าทศนิยมฐานสอง ณ ตำแหน่งนั้น มีค่าเป็น 1 แล้วลบผลลัพธ์ด้วย 1 ทำซ้ำขั้นตอนแรก สำหรับตำแหน่งถัดไป (ทางขวามือ)
- แต่ถ้าผลลัพธ์ที่ได้มีค่า น้อยกว่า 1 ค่าทศนิยมฐานสอง ณ ตำแหน่งนั้น มีค่าเป็น 0 ทำซ้ำขั้นตอนแรก สำหรับตำแหน่งถัดไป

ถ้าผลลัพธ์ที่ได้ไม่เป็นศูนย์เราจะหยุด ณ ตำแหน่งที่มีความละเอียดตามที่กำหนดไว้ (ตำแหน่งที่ N) เรียกว่า การตัดเศษ (Truncation)

การปัดเศษ (Rounding) ถึงตำแหน่งที่ N ทำได้โดยคำนวณถึง ตำแหน่งที่ N+1 แล้วบวกผลลัพธ์ที่ได้ด้วยเลขเศษส่วนที่เป็นศูนย์ทั้งหมดยกเว้นตำแหน่ง N+1 แล้วทำ Truncation



Conversion of Fractions

ตัวอย่าง แปลงเลข $(1.68)_{10}$ เป็นฐาน 2 ให้ละเอียด ถึงตำแหน่งที่ 3 โดยการปัดเศษ Rounding

1. เลขหน้าจุดทศนิยมแปลงต่างหาก ได้ **1**.
2. คูณเลขหลังจุดทศนิยม ด้วย 2 ผลลัพธ์เป็น $0.68 \times 2 = 1.36$
3. ผลลัพธ์ที่ได้มากกว่า 1 ให้นำ 1 ไปลบออก ใส่เลข 1 ที่ทศนิยมหลักแรก **1.1**
4. คูณผลลัพธ์ข้อ 3 ด้วย 2 $= 0.36 \times 2 = 0.72$
5. ผลลัพธ์ที่ได้น้อยกว่า 1 ใส่เลข 0 ที่ทศนิยมตำแหน่งถัดไป **1.10**
6. คูณผลลัพธ์ข้อ 4 ด้วย 2 $= 0.72 \times 2 = 1.44$
7. ผลลัพธ์ที่ได้มากกว่า 1 ให้นำ 1 ไปลบออก ใส่เลข 1 ที่ทศนิยมตำแหน่งถัดไป **1.101**
8. คูณผลลัพธ์ข้อ 7 ด้วย 2 $= 0.44 \times 2 = 0.88$
9. ผลลัพธ์ที่ได้น้อยกว่า 1 ใส่เลข 0 ที่ทศนิยมตำแหน่งถัดไป **1.1010**
10. บวกผลลัพธ์ 9 ด้วย 0.0001 ได้ $= 1.1011$ แล้วตัดตำแหน่งที่ 4ทิ้ง **1.101**
11. ตรวจสอบคำตอบแปลงเป็นเลขฐาน 10 $= 1 + 0.5 + 0.125 = 1.625$



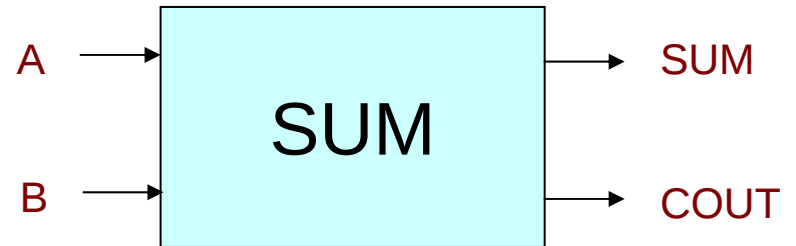
Half Adder

การบวกเลขฐาน 2 เหมือนกับเลขฐาน 10

CIN

$$\begin{array}{r} 1111\ 111 \\ 1010.1011 \\ + 1101.0101 \\ \hline 1\ 1000.0000 \end{array}$$

COUT

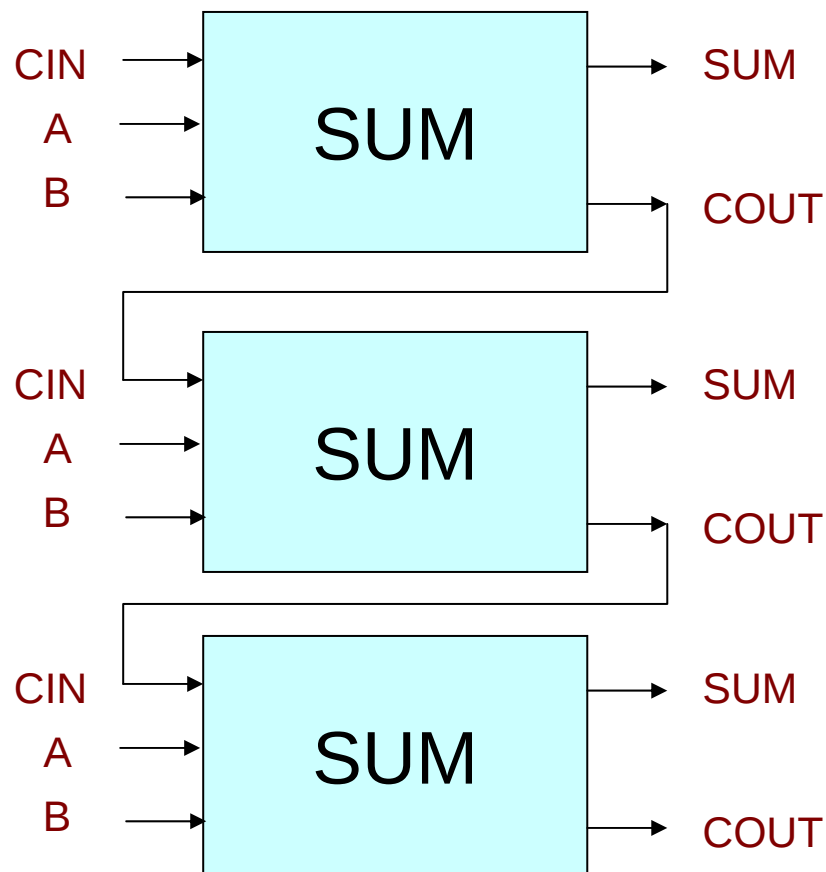


B	A	SUM	COUT
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder

Full Adder คือการบวกแบบมีตัวทดเข้า สามารถนำมาต่อกันได้

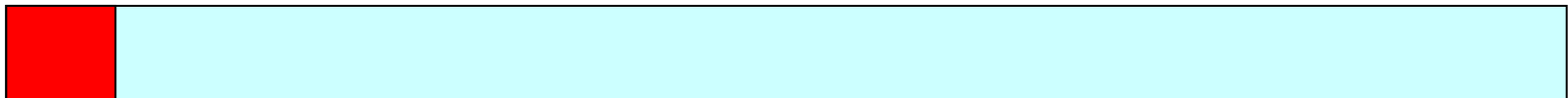
CIN	B	A	SUM	COUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1





Course Outline

- Introduction to Digital System
- Truth Table and Boolean Algebra
- Methods for Minimizing Boolean Expression
- Programmable Circuits
- Sequential Flip-Flops
- Synchronous Counters
- Synchronous Finite State Machine
- A/D and D/A Conversion
- Asynchronous Sequential Circuit Design
- Practical Digital Designs (Part I Arithmetic Operations)
- Practical Digital Designs (Part II Microprocessor)
- Practical Digital Designs (Part III Data Communication)





DIGITAL DESIGN
School of Computer Engineering

Suranaree University
of Technology

Lecture 2 Introduction (II)

Paramate Horkaew

School of Computer Engineering, Institute of Engineering
Suranaree University of Technology



Lecture Outline

- Negative Binary and Operations
- Overflow and Underflow
- Introduction to Codes and Encoding
- Switches and Basic Gates
- Simple Digital Circuit: A Multiplexer
- Introduction to Digital Systems



Digital Subtraction

จากหลักการพีชคณิตพื้นฐาน เราทราบว่า การลบเลข บวก จำนวน
หนึ่ง ออกจากอีกจำนวนหนึ่ง ก็คือ การ บวก เลขบวกตัวแรก ด้วย
ค่าติดลบ ของเลขตัวที่สอง นั่นคือ $A - B = A + (-B)$

พิจารณาการลบเลขฐานสิบ $8 - 3 = 8 + (-3) = 5$

ซึ่งสามารถคิดได้อีกวิธีหนึ่ง นำตัวเลข 10^N ลบด้วยเลข 3 โดยที่ N
คือ จำนวนหลัก (Det = Decimal Digit) ซึ่ง ในที่นี้คือ 1 แล้วนำ
ผลที่ได้ **ไปบวก** กับ 8 ได้เป็น $8 + (10 - 3) = 8 + 7 = 15$

สังเกตว่าเลขหลักซ้ายมือคือผลลบ ส่วนเลข 1 ที่เกินมาจะไม่คิด

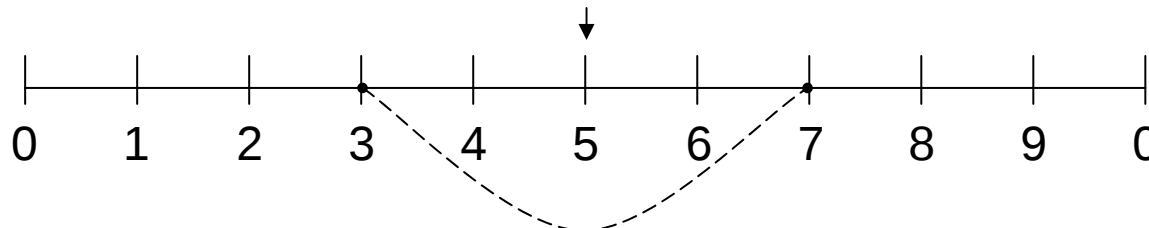


10's Complement (10C)

เราเรียกเลขจำนวนบวก ที่ไปแทน จำนวนลบ ในขั้นตอนการลบ เพื่อให้การคำนวณ เป็นการบวกเลขจำนวนบวกสองจำนวน ว่า (Base)-C

วิธีการหา Complement ดังกล่าว มีประโยชน์น้อย เนื่องจาก ยังคง ต้องใช้ การคำนวณด้วยเครื่องหมายลบ อยู่

จากการสังเกตพบว่า คู่ Complement ของ cipher ฐาน 10 ได้แก่ (1, 9), (2, 8), (3, 7), (4, 6), (5, 5) ดังนั้น เราอาจจะหาค่า 10C ของเลขจำนวนใดๆ โดยแทนด้วยค่าสะท้อนบนเส้นจำนวนแบบคาบ





2's Complement (10C)

เลขฐานสองมี cipher เพียง 2 ตัวตั้งนั้น 0 และ 1 จึงเป็น 2-complement กัน
อย่างไรก็ดี การหา 2C ด้วยวิธีสะท้อนค่าบนเส้นจำนวนแบบคานั้น ไม่สะดวก
สำหรับเลขที่มีหลายหลัก

ดังนั้น โดยอาศัย คุณลักษณะเฉพาะของเลขฐาน 2 เราสามารถหา
ค่า 2C ของเลขฐาน 2 ใดๆ ได้โดยการ เพิ่มบิตเครื่องหมายที่มีค่า
เป็น 0 ทางขวามือ (B_s) แล้วกลับ (invert) ตัวเลขในแต่ละหลัก
(บิต) จาก $1 \leftrightarrow 0$ แล้วบวกผลลัพธ์ที่ได้ด้วย 1

$$\text{if } +B = B_s B_2 B_1 B_0$$

$$-0010 = 1101 + 1$$

$$-B = \overline{B_s} \overline{B_2} \overline{B_1} \overline{B_0} + 1$$

$$= 1110$$



Examples

การลบเลขจำนวนเต็ม, **การทำ 2C ต้องระบุจำนวนบิตด้วยเสมอ**

ตัวอย่าง a) $4 - 1 = ?$ โดยใช้ 2-complement ขนาด 4 บิต

เพิ่มบิตเครื่องหมาย: $(4)_{10} = \mathbf{0}100$ และ $(1)_{10} = \mathbf{0}001$

4-bit 2-C ของ 0001 คือ $1110 + 1 = 1111$

ดังนั้น $4 + (-1) = 0100 + 1111 = \mathbf{1}0011$

เลือกตอบเฉพาะ 4 บิตทางขวามือ = 0011 หรือ 3

ตัวอย่าง b) $2 - 5 = ?$ โดยใช้ 2-complement ขนาด 4 บิต

เพิ่มบิตเครื่องหมาย: $(2)_{10} = \mathbf{0}010$ และ $(5)_{10} = \mathbf{0}101$

4-bit 2-C ของ 0101 คือ $1010 + 1 = 1011$

ดังนั้น $2 + (-5) = 0010 + 1011 = \mathbf{0}1101$

เลือกตอบเฉพาะ 4 บิตทางขวามือ = 1101 หรือ 13 (ทำไมคำตอบผิด??)



Cautions (I)

การลบเลขแล้วผลลัพธ์ติดลบ

จากตัวอย่าง b) $(2 - 5)_{10} = 1101 = 13$ ซึ่งเป็นคำตอบที่ผิด

ทั้งนี้เนื่องจากว่า เลขฐาน 2 ขนาด N บิตสามารถแทนเลขได้ทั้งหมด 2^N ค่า

- ถ้าเป็นเลขจำนวนบวกทั้งหมด จะแทนเลขได้ จาก 0 ถึง 2^{N-1}
- ถ้าเป็นเลข 2C จะแทนเลขได้ จาก $-2^{N-1} - 1$ ถึง $2^{N-1} - 1$

ถ้า $N = 4$ เลข 2C แทนเลขได้ -7 ถึง 7 โดยที่เลข 0 และ -0 แทนได้ด้วย 0000 และ 1000

ดังนั้นคำตอบสุดท้ายจึงต้องพิจารณาบิตเครื่องหมายก่อน 0 (บวก) และ 1 (ลบ)



Cautions (II)

จากตัวอย่าง b) $(2 - 5)_{10} = 1101$ ซึ่งบิตเครื่องหมาย (บิตซ้ายสุด) เป็น 1 หมายความว่า ค่าที่ได้ติดลบ

ดังนั้นคำตอบสุดท้ายจึงต้องทำเป็น 2C ก่อน

2C ขนาด 4 บิตของ 1101 คือ $0010 + 1 = 3$

เนื่องจากการทำ 2C ดังนั้นคำตอบคือ -3

การบ้าน คำนวณ $2.75 - 0.50$ ด้วยวิธี 2C โดยมีคำแนะนำดังนี้

- 1) แปลงเลขทั้งสองตัวเป็นฐานสองตามปกติโดยวิธี repeated multiplication
- 2) ถ้าเลขที่ได้มีจำนวนหลักไม่เท่ากัน เพิ่ม 0 ซ้ายมือของตัวที่มีหลักน้อยกว่า
- 3) เพิ่มบิตเครื่องหมาย (0) ทางซ้ายมือ ของทั้งสองตัว 1 บิต
- 4) ทำการลบปกติเหมือนตัวอย่างโดยไม่สนใจทศนิยม
- 5) คำตอบสุดท้ายให้ใส่ทศนิยมกลับเข้าไปที่เดิม ตรวจสอบโดยแปลงเป็นฐาน 10



Cautions (III)

Overflow และ Underflow

ตัวอย่าง c) $3 + 7 = ?$ โดยใช้ 2-complement ขนาด 4 บิต

เพิ่มบิตเครื่องหมาย: $(3)_{10} = \mathbf{0}011$ และ $(7)_{10} = \mathbf{0}111$

ดังนั้น $3 + 7 = 0011 + 0111 = \mathbf{0}1010 = (10)_{10}$

กรณีนี้ คำตอบจะถูก ก็ต่อเมื่อ บิตเครื่องหมาย ไม่เปลี่ยน (ซึ่งในที่นี้ไม่ใช่)

ถ้าบิตเครื่องหมายเปลี่ยนจากการบวก เรียกว่า Overflow

ตัวอย่าง d) $-4 - 5 = ?$ โดยใช้ 2-complement ขนาด 4 บิต

เพิ่มบิตเครื่องหมาย: $(4)_{10} = \mathbf{0}100$ และ $(5)_{10} = \mathbf{0}101$

4-bit 2-C ของ 0100 คือ $1011 + 1 = 1100$

4-bit 2-C ของ 0101 คือ $1010 + 1 = 1011$

$-4 - 5 = 1100 + 1011 = \mathbf{1}0111$ บิตเครื่องหมายเปลี่ยน $\rightarrow$ Underflow



Code and Encoding

Digital Code หรือรหัสดิจิทัล คือ ตัวเลขที่ใช้ในการแทนค่าสัญลักษณ์ ที่สนใจ

Encoding คือ กระบวนการที่ทำการแปลงสัญลักษณ์ เป็นตัวเลข ด้วยระเบียบวิธีเฉพาะ ปกติมักจะพิจารณาจาก สัญลักษณ์ที่เป็นไปได้ทั้งหมด

Decoding คือ กระบวนการกลับของ Encoding หรือ การกู้คืนสัญลักษณ์เดิม เมื่อกำหนด รหัสตัวเลขที่สมมูลกัน

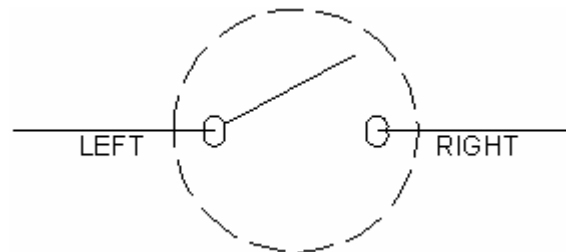
การบ้าน ศึกษารหัส ASCII และ รหัสแท่ง (Barcode)

Switches

คือ อุปกรณ์ ที่ปรากฏขึ้นพร้อมๆ กับอารยธรรมมนุษย์ พบเห็นทั่วไปในรูปแบบต่างๆ กัน อาทิเช่น ลิ้น (Valve) ของเครื่องดนตรี ประเกทเป่า ฝาปิดถังเก็บน้ำ ก๊อกน้ำ สากตำข้าว หรือแม้กระทั่ง เซ็นเซอร์แบบไวแสง

หลักการของ Switch คือ เมื่อปิด จะมีการถ่ายทอดข่าวสารระหว่างจุด

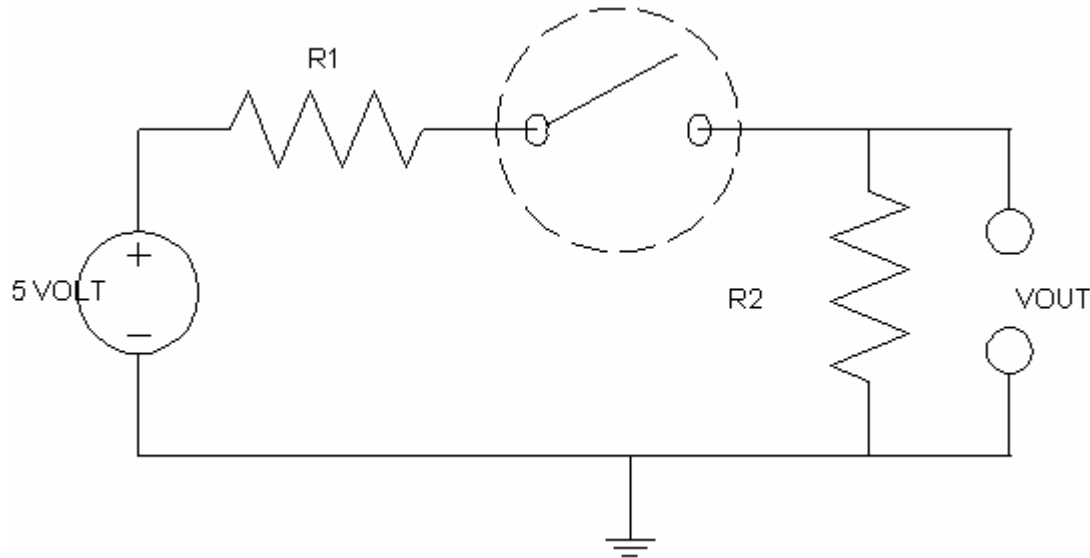
在这里เราจะพิจารณาเฉพาะ Switches แบบดิจิทัล ซึ่งมีเพียงสองสถานะ ได้แก่ เปิด (0) และ ปิด (1) ดังรูป





Electric Switches

ตัวอย่างของ Switch ไฟฟ้าควบคุมแรงดัน โดยการเปลี่ยนแปลง
กระแสที่ตกคร่อมความต้านทาน

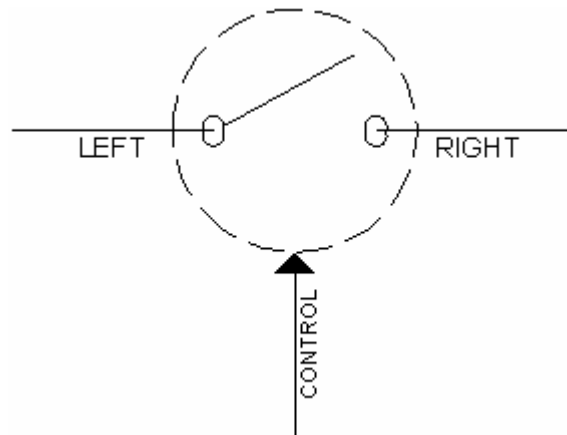


$$V_{out} = \frac{R2}{R1 + R2} \cdot 5V \quad R1 \ll R2$$



Control Switches

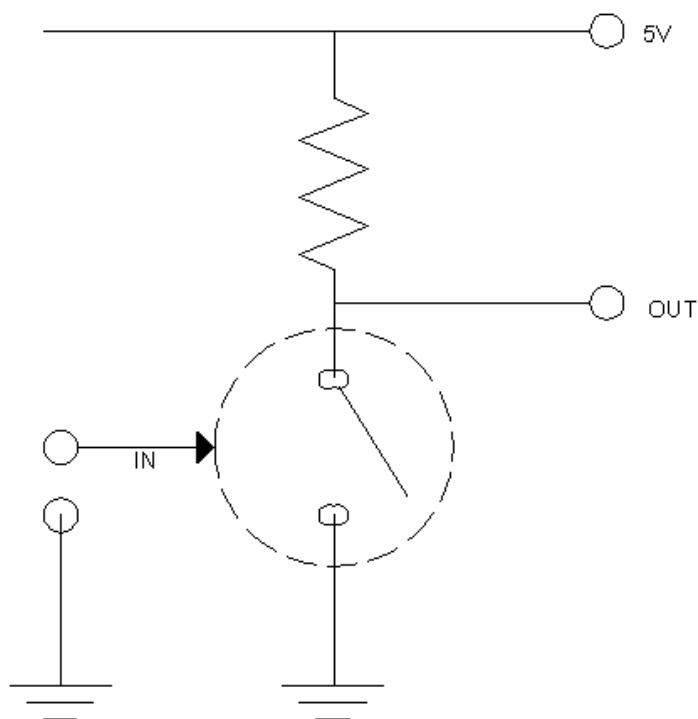
คุณสมบัติสำคัญ ของ Switches ในทางดิจิทัลคือ ความสามารถ
ที่ Switch ตัวหนึ่งจะควบคุม Switch (s) ตัวอื่นได้



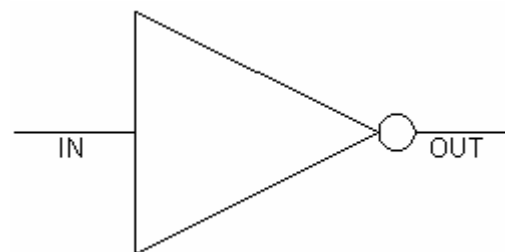
กฎทั่วไปคือ CONTROL กำหนดสถานะของ Switch ตามสถานะ
แรงดันที่ CONTROL เช่น 5V ปิด (ON) 0V เปิด (OFF) เป็นต้น

An Inverting Gate

ตัวอย่างของ Gate ที่ง่ายที่สุด ประกอบด้วย INPUT หนึ่งสาย และ OUTPUT หนึ่งสาย คือ Inverting Gate หรือ Not Gate

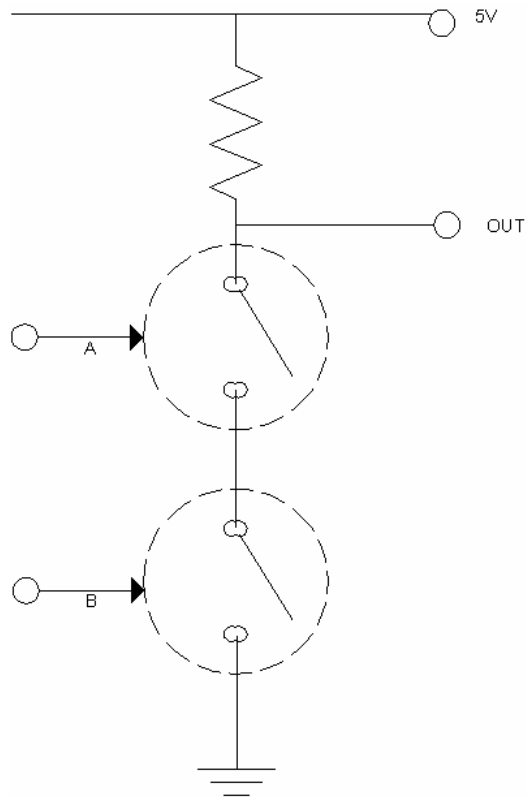


IN	OUT	Switch
0	1	OFF
1	0	ON

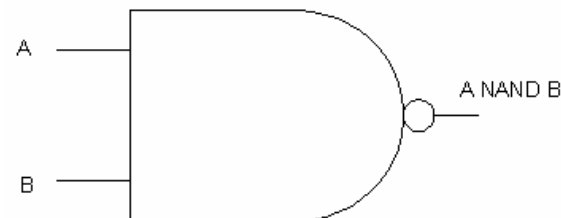


A NAND Gate

NAND Gate หรือ NOT-AND Gate ประกอบด้วย สอง input และหนึ่ง output สามารถสร้างได้จาก Switch สองตัว



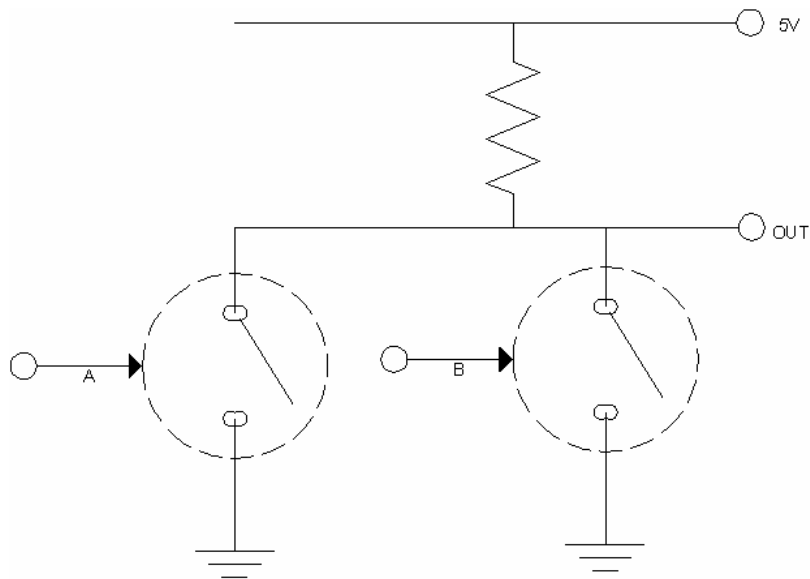
IN		
B	A	NAND
0	0	1
0	1	1
1	0	1
1	1	0



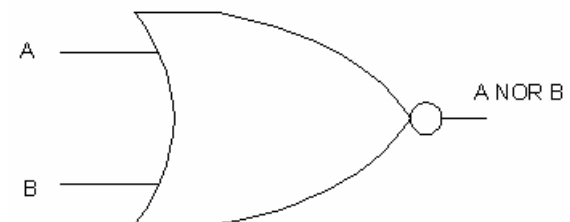


A NOR Gate

NOR Gate หรือ NOT-OR Gate ประกอบด้วย สอง input และ
หนึ่ง output สามารถสร้างได้จาก Switch สองตัว

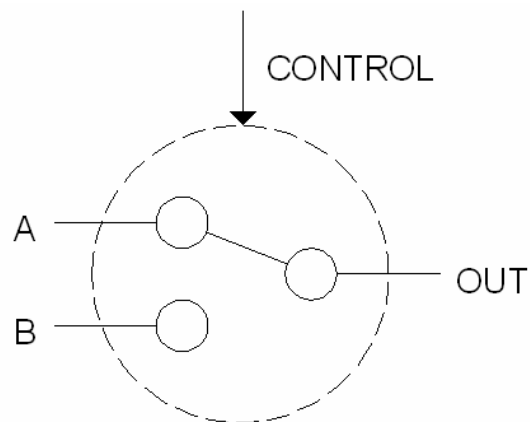


IN		
B	A	NOR
0	0	1
0	1	0
1	0	0
1	1	0

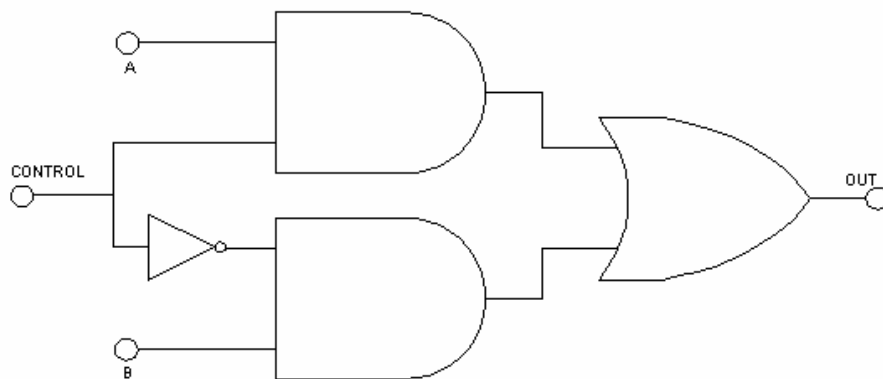


Simple Digital Circuit

ตัวอย่าง วงจรดิจิทัลอย่างง่าย วงจรหนึ่งได้แก่ Multiplexer คือ วงจรที่ output จะมีค่าเท่ากับ input ตัวที่เลือกโดย CONTROL



CONTROL	OUT
1	A
0	B





Digital Systems

ระบบ (Systems) คือ การประกอบกัน ขึ้นของอุปกรณ์ (Devices) ย่อยๆ เพื่อใช้ในการปรับเปลี่ยนข้อมูลข่าวสาร (Digital)

- ข้อมูลข่าวสาร ปรากฏที่ input ของระบบในรูปของสัญญาณที่เปลี่ยนแปลงตามเวลา (Time Varying Signals)
- หาก input ของระบบเปลี่ยนจะเกิดการเปลี่ยนแปลงที่ output ด้วยหลังจาก หน่วงเวลาไประยะหนึ่ง (Delay)
- ระบบอาจจะประกอบด้วยหน่วยความจำ สำหรับคงข้อมูลไว้ใช้ได้
- ข้อมูล ที่ปรากฏที่ output ของระบบจะถูกกำหนดโดย สัญญาณ ที่ input และ ข้อมูลที่จัดเก็บในหน่วยความจำ
- ข้อมูล ที่ปรากฏที่ output สามารถสร้างได้จากกฎหรืออัลกอริทึม ที่สามารถสร้างได้จากอุปกรณ์จริง (Physical Devices)



System Models

แบบจำลองระบบ (System Models) คือ การอธิบาย คุณสมบัติของระบบ อย่างง่าย โดยไม่ต้องสร้างขึ้นจริง

- แบบจำลอง ทำหน้าที่ทำนาย พฤติกรรม หรือ การทำงาน ของระบบ ซึ่งเปลี่ยนแปลงไปตาม input หรือค่า ในหน่วยความจำ
- แบบจำลองเดี่ยวๆ ไม่สามารถครอบคลุม ลักษณะทางกายภาพของระบบได้ทุกอย่าง ดังนั้น จึง ไม่ อาจใช้ทำนายพฤติกรรมของระบบได้อย่างสมบูรณ์แบบ
- แบบจำลองที่สร้างขึ้น อย่างมาก ไม่ควรจะซับซ้อนไปกว่าระบบที่พิจารณา (Occam Razor's Principal)
- เราอาจเลือกใช้ หลายๆ แบบจำลอง ตามต้องการ เพื่ออธิบายการทำงานของระบบในแง่มุมต่างๆ ที่สนใจได้



Structural and Behavioral Models

แบบจำลองโครงสร้าง (Structural Models) คือแบบจำลองที่อธิบายถึง **วิธีการสร้างระบบ**

มักจะอยู่ในรูปของ แผนผัง หรือ รายการของอุปกรณ์ และ ลักษณะการเชื่อมต่อ แบบจำลองชนิดนี้ไม่เหมาะสำหรับทำนายพฤติกรรมของระบบ และ พึ่งระวังว่าแบบจำลองชนิดนี้ อาจจะไม่ระบุรายละเอียดปลีกย่อยบางอย่าง เช่น ตำแหน่งการวางอุปกรณ์

แบบจำลองพฤติกรรม (Behavioral Models) คือแบบจำลองที่อธิบายถึง **การทำงานของระบบ** ในแง่ของการเปลี่ยนแปลงที่ output ในรูปของฟังก์ชันของ input

แบบจำลองชนิดนี้ มักจะอยู่ในรูปของแบบจำลองทางคณิตศาสตร์ หรือ คำสั่งคอมพิวเตอร์ ระบบต่างกัน อาจมีแบบจำลองเดียวกันได้



System Engineering

การวิเคราะห์และสร้างระบบ อย่างครบวงจร ประกอบด้วย 3 ขั้นตอนหลัก

- **Analysis** คือ การวิเคราะห์หาคุณสมบัติ และ พฤติกรรมของระบบ จาก คำอธิบายทางโครงสร้าง
- **Synthesis** คือ การสังเคราะห์ (สร้าง) ระบบขึ้นจริง ที่มีพฤติกรรมตามที่กำหนด ซึ่งต้องคำนึงถึง ความเร็ว การใช้พลังงาน ค่าใช้จ่าย และ ความมั่นใจเชื่อถือได้ ของระบบ
- **Verification** คือ การตรวจสอบความถูกต้องของระบบ โดยการสังเกตการเปลี่ยนแปลงของ output เมื่อมีการเปลี่ยนแปลงที่ input ซึ่งทำได้ โดยจำลองการทำงานด้วยคอมพิวเตอร์ หรือทดสอบจากระบบที่สร้างขึ้นก็ได้



Digital System Characteristics

ระบบดิจิทัล มีคุณสมบัติที่สำคัญดังต่อไปนี้

- สัญญาณที่เกี่ยวข้อง เป็นสัญญาณไม่ต่อเนื่อง (Discrete) ซึ่งปกติมักจะมีเพียงสองระดับ ซึ่งในทางไฟฟ้า จะแทนด้วยความต่างศักย์ สูง (H) และ ต่ำ (L)
- ระบบมีเสถียรภาพสูง ทนทานต่อสัญญาณรบกวน สามารถสร้างให้มีความแม่นยำ ได้ไม่จำกัด (ใช้สัญญาณหลายๆ ตัว) สร้างได้ง่าย มีค่าใช้จ่ายต่ำ (รวมถึงด้านพลังงาน)
- แบบจำลองของระบบ มักจะอยู่ในรูปของ คณิตศาสตร์ที่ไม่ต่อเนื่อง (Discrete Mathematics) หรือขั้นตอนวิธี (Algorithm) เช่น พีชคณิต Boolean กราฟสถานะ หรือ โปรแกรมคอมพิวเตอร์



Types of Digital Systems

Combinational Systems (Midterm)

คือระบบที่ปราศจากหน่วยความจำ ดังนั้น สัญญาณ input เพียงอย่างเดียว จะเป็นตัวกำหนด สัญญาณ output

Sequential Systems (Final)

คือระบบที่มีหน่วยความจำ ดังนั้น สัญญาณ output จึงขึ้นอยู่กับทั้ง สัญญาณ input ปัจจุบัน และ ก่อนหน้า ดังนั้น สัญญาณ input เดียวกัน อาจสร้าง output ที่ต่างกันได้ใน เวลาต่างกัน



Conclusion

- Review on Previous Lecture
- Negative Binary and Operations
- Overflow and Underflow
- Introduction to Codes and Encoding
- Switches and Basic Gates
- Simple Digital Circuit: A Multiplexer
- Introduction to Digital Systems