01076001

วิศวกรรมคอมพิวเตอร์เบื้องต้น

Introduction to Computer Engineering

Number System

ข้อมูลที่เก็บในคอมพิวเตอร์มีประเภทใดบ้าง?

ข้อมูลที่เก็บในคอมพิวเตอร์มีประเภทใดบ้าง?

- คอมพิวเตอร์เป็นอุปกรณ์ มัลติมีเดีย
  - ตัวเลข
  - ตัวอักษร
  - เสียง
  - ภาพและกราฟิก
  - ภาพเคลื่อนไหว

เก็บค่าเหล่านี้

เหตุใดข้อมูลที่เก็บในคอมพิวเตอร์
จึงเป็นเลขฐาน 2 ?

Digital Information

ทำให้ไม่ถูกรบกวน

- คอมพิวเตอร์เป็นอุปกรณ์ Digital

ไม่ต่อเนื่อง

- Digital มีการทำงานเป็นแบบ Discrete

เปิด(1)
ปิด(0)

- ดังนั้น Computers are finite!

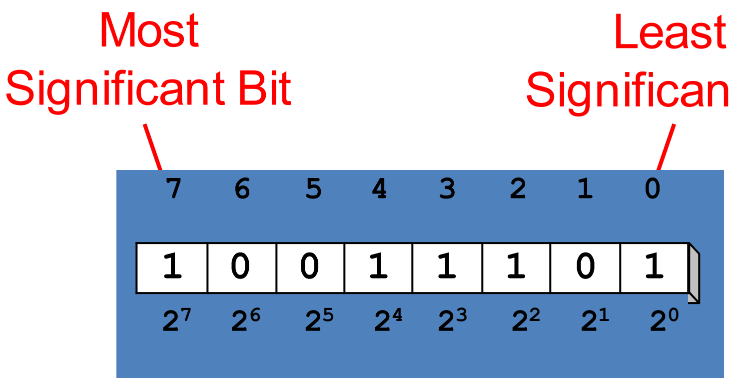
How do we represent an infinite world?

Your Turn

- ให้ยกตัวอย่างการเก็บข้อมูลที่ไม่จำกัด (Infinite) ในระบบคอมพิวเตอร์ที่จำกัด (finite)

Binary Numbers

- เลขฐาน 2 แต่ละหลัก (binary digit เรียกสั้นๆ ว่า bit) มีค่าเป็น 1 หรือ 0 เท่านั้น
- บิต ไม่มีความหมาย แต่เมื่อใช้แทนข้อมูลแล้วจึงมีความหมาย
  – เลขจำนวนเต็ม (Integer)
  – ตัวอักษร (Character)
  – เลขทศนิยม (Floating-point)
  – ภาพ เสียง ฯลฯ
- Bit Numbering



  – Least significant bit (LSB) คือ บิตขวาสุด (bit 0)
  – Most significant bit (MSB) คือ บิตซ้ายสุด (bit 7 in an 8-bit number)

Binary Representations

- 1 บิต สามารถมีค่าเป็น 0 หรือ 1
- 1 บิต สามารถแทนค่าได้ 2 ค่า
- 2 บิต สามารถแทนค่าได้ 4 ค่า

2^n ค่า

- 3 บิตสามารถแทนค่าได้กี่ค่า?
- 4 บิตสามารถแทนค่าได้กี่ค่า?
- 8 บิตสามารถแทนค่าได้กี่ค่า?

Binary Representations

1 Bit	2 Bits	3 Bits	4 Bits	5 Bits
0	00	000	0000	00000
1	01	001	0001	00001
	10	010	0010	00010
	11	011	0011	00011
		100	0100	00100
		101	0101	00101
		110	0110	00110
		111	0111	00111
			1000	01000
			1001	01001
			1010	01010
			1011	01011
			1100	01100
			1101	01101
			1110	01110
			1111	01111
				10000
				10001
				10010
				10011
				10100
				10101
				10110
				10111
				11000
				11001
				11010
				11011
				11100
				11101
				11110
				11111

Binary Representations

- N บิต สามารถแทนได้กี่ค่า

2^n

- เพราะอะไร

มีรูปแบบที่เป็นไปได้เท่านั้น

- ทุกครั้งที่เพิ่ม 1 บิต สิ่งที่เปลี่ยนแปลงคืออะไร

จำนวนค่าที่เก็บเปลี่ยน $\frac{\cancel{2^{n+1}}}{\cancel{2^n}}$ 2 (เท่า)

- 9 บิต, 10 บิต แทนได้กี่ค่า

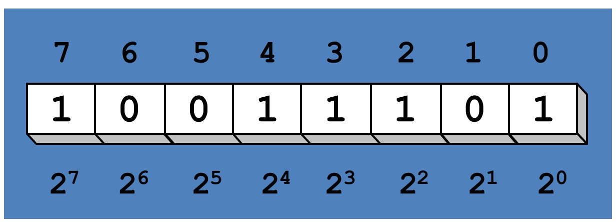
2^9 , 2^{10}

Binary Representations

- ข้อมูลที่อยู่ในคอมพิวเตอร์ เราบอกไม่ได้ว่า มันคืออะไร จนกว่าจะถูกนำไปใช้
- เช่น 65_{10} ยังไม่มีความหมาย
  - อาจจะเป็นตัวเลข 65
  - อาจจะเป็นตัวอักษร ‘A’
  - อาจจะเป็น “จุด” ในภาพอะไรสักภาพหนึ่ง
  - อาจจะเป็นค่าระดับเสียงเพลง
  - อื่นๆ อีกมากมาย

Converting Binary to Decimal

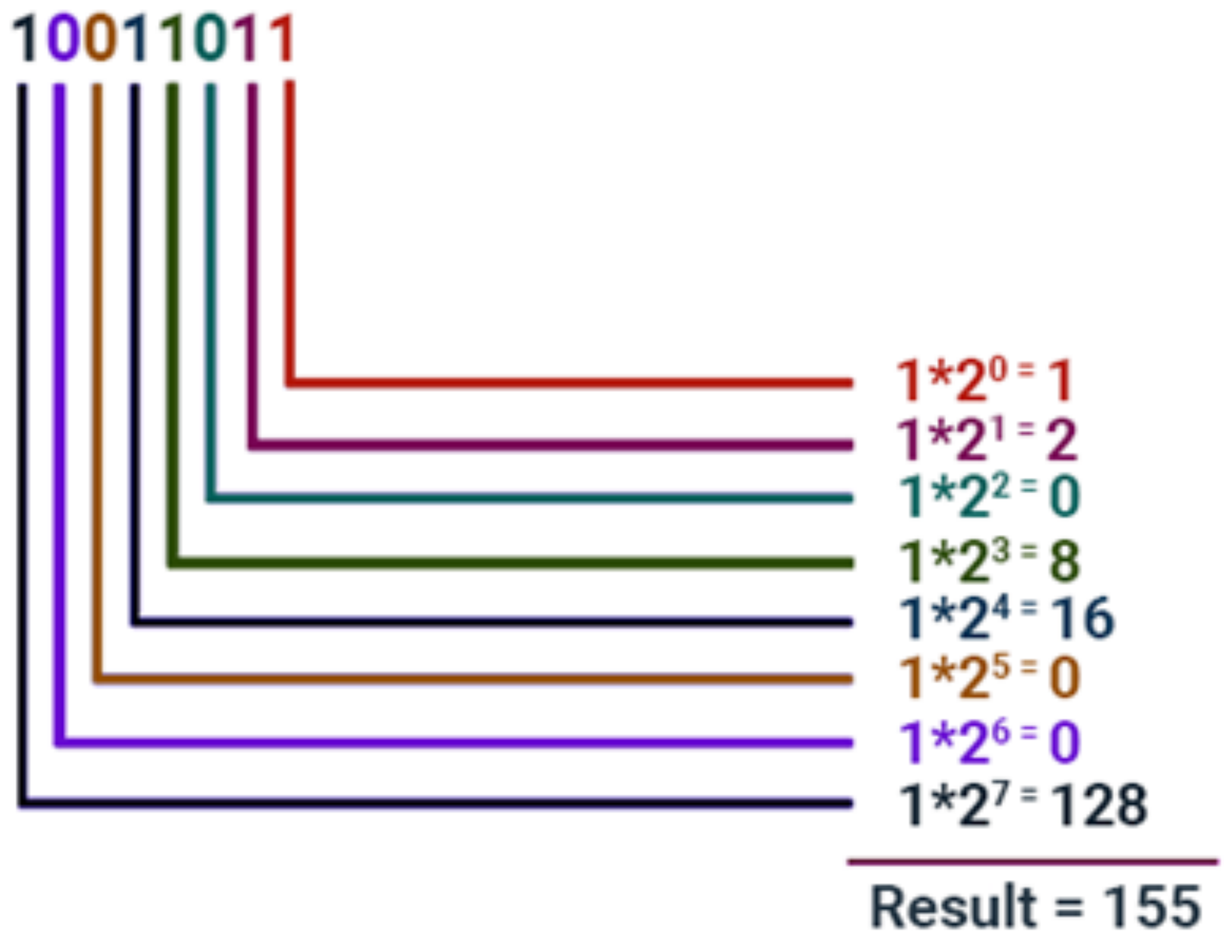
- บิตที่ 0 จะมีค่าน้ำหนักเป็น 1 บิตต่อไปจะมีค่าน้ำหนักเป็น 2 เท่าของบิตก่อนหน้า
- ผลรวมของของเลขฐาน 2 คือ ผลบวกของค่าประจำหลักคูณกับค่าน้ำหนัก
- Binary $(10011101)_2 = 2^7 + 2^4 + 2^3 + 2^2 + 1 = 157$



Some common powers of 2

2^n	Decimal Value	2^n	Decimal Value
2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768

Example #1


10011011
1*2^0 = 1
1*2^1 = 2
1*2^2 = 0
1*2^3 = 8
1*2^4 = 16
1*2^5 = 0
1*2^6 = 0
1*2^7 = 128
Result = 155

Example #2

$$1011001_2$$

$$0 \times 2 + 1 = 1$$
$$1 \times 2 + 0 = 2$$
$$2 \times 2 + 1 = 5$$
$$5 \times 2 + 1 = 11$$
$$11 \times 2 + 0 = 22$$
$$22 \times 2 + 0 = \underline{44}$$
$$\underline{44} \times 2 + 1 = 89$$

Exercise #1 : Binary to Decimal

- แปลงเลขฐาน 2 ต่อไปนี้ ให้เป็น Decimal
  - **1101_2**
  - **1010_2**
  - **$0010\ 1001_2$**
  - **$0110\ 1011_2$**
  - **$1000\ 1011_2$**

Exercise #1 : Binary to Decimal

- แปลงเลขฐาน 2 ต่อไปนี้ ให้เป็น Decimal
  - **1101_2** **->** **13**
  - **1010_2** **->** **10**
  - **$0010\ 1001_2$** **->** **41**
  - **$0110\ 1011_2$** **->** **107**
  - **$1000\ 1011_2$** **->** **139**

Tips

- ตัวเลขบางค่า เช่น 1111 1101 การแปลงโดยวิธีปกติ อาจจะช้าเพราะมีเลข 1 จำนวนมาก
- กรณีนี้ เราอาจใช้ shortcut เนื่องจากเราทราบว่า 1111 1111 คือ 255_{10} ดังนั้น 1111 1101 = 255 – 2 = 253 เป็นต้น
- แต่การจะใช้วิธีการนี้ เราต้องจำตัวเลขค่าประจำหลักได้
  - เช่น 255 คือ 256-1 = 1 0000 0000 – 1
- ตย. 1111 0011 = 255 – (8+4) = 243 เป็นต้น

Convert Unsigned Decimal to Binary

- นำตัวเลขฐาน 10 มาหารด้วย 2 จนกว่าจะหารไม่ได้
- นำเศษของการหารแต่ละครั้งมาใช้เป็นแต่ละหลักในเลขฐาน 2

Division	Quotient	Remainder
37 / 2	18	1
18 / 2	9	0
9 / 2	4	1
4 / 2	2	0
2 / 2	1	0
1 / 2	0	1

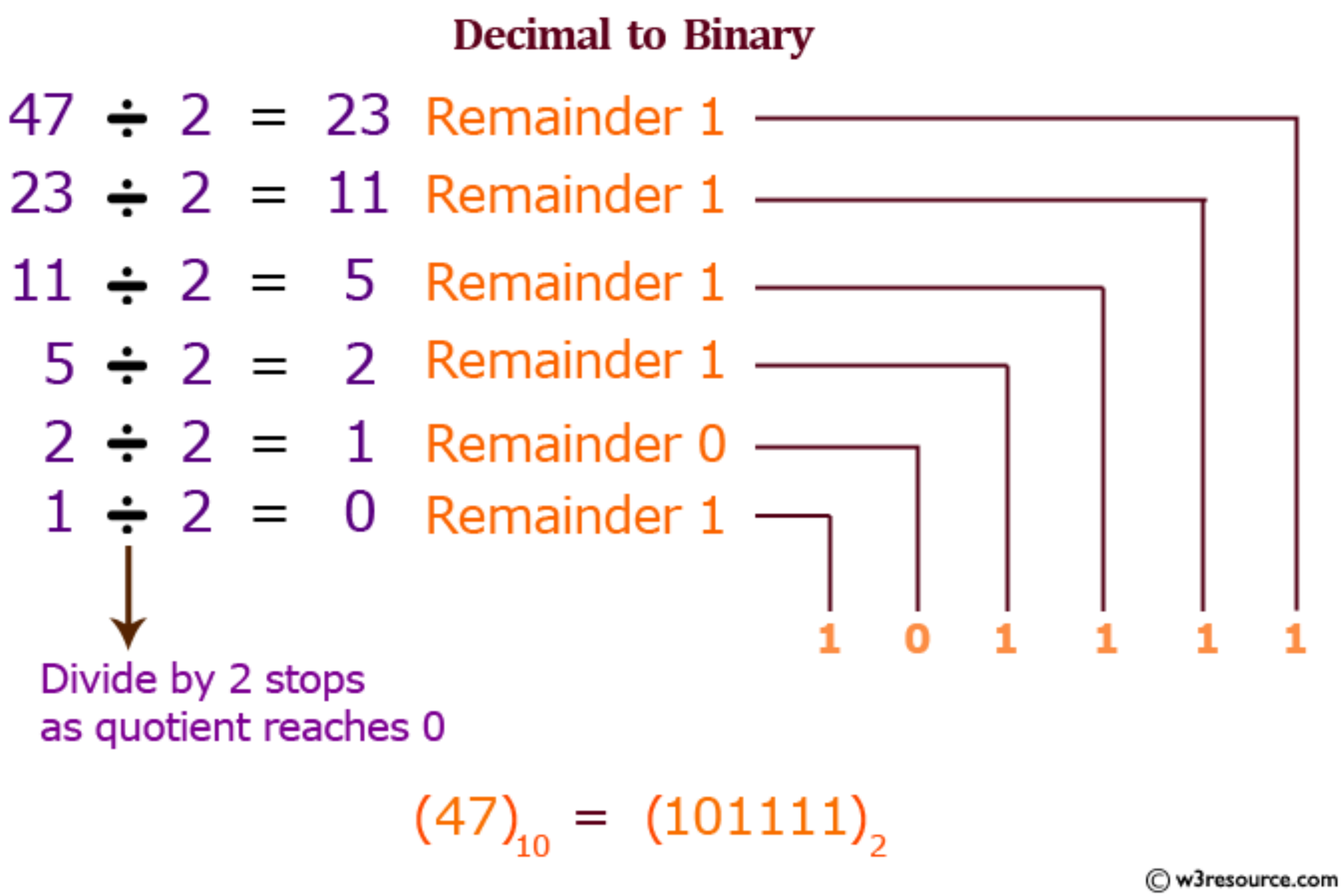
least significant bit

$37 = (100101)_2$

most significant bit

stop when quotient is zero

Example #3


Decimal to Binary
47 ÷ 2 = 23 Remainder 1
23 ÷ 2 = 11 Remainder 1
11 ÷ 2 = 5 Remainder 1
5 ÷ 2 = 2 Remainder 1
2 ÷ 2 = 1 Remainder 0
1 ÷ 2 = 0 Remainder 1
1 0 1 1 1 1
Divide by 2 stops
as quotient reaches 0
$(47)_{10} = (101111)_2$
© w3resource.com

Example #4

	Remainder:
2)156	
2)78	0
2)39	0
2)19	1
2)9	1
2)4	1
2)2	0
2)1	0
	1

$$\boxed{156_{10} = 10011100_2}$$

Exercise #2 : Decimal to Binary

- แปลงเลขฐาน 10 ต่อไปนี้ ให้เป็นเลขฐาน 2
  - **26** 11010
  - **250** 1111 1010
  - **150** 1001 0110
  - **68** 1000100
  - **137** 1000 1001

Exercise #2 : Decimal to Binary

- แปลงเลขฐาน 10 ต่อไปนี้ ให้เป็นเลขฐาน 2
  - **26** **->** 0001 1010
  - **250** **->** 1111 1010
  - **150** **->** 1001 0110
  - **68** **->** 0100 0100
  - **137** **->** 1000 1001


Kahoot!

Hexadecimal Integers

1 bit_{16} = 4 bit_2

- 16 Hexadecimal Digits: 0 – 9, A – F
- สะดวกต่อการใช้งานมากกว่าเลขฐาน 2 (เท่ากับ 4 หลักของฐาน 2)

0-9 ↔ A-F

Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

Converting Binary to Hexadecimal

- แต่ละหลักของเลขฐาน 16 จะเทียบเท่าเลขฐาน 2 จำนวน 4 บิต
- ดังนั้นเพียงแบ่งเป็นชุดละ 4 บิต และแปลงจากตารางเท่านั้น
- ตัวอย่าง
  - แปลงเลขฐาน 2 ขนาด 32 บิต ให้เป็นเลขฐาน 16
  - 1110 1011 0001 0110 1010 0111 1001 0100

E	B	1	6	A	7	9	4
1110	1011	0001	0110	1010	0111	1001	0100

Converting Hexadecimal to Decimal

- Multiply each digit by its corresponding power of 16

  Value = $(d_{n-1} \times 16^{n-1}) + (d_{n-2} \times 16^{n-2}) + \ldots + (d_1 \times 16) + d_0$

- Examples:

  หลักการเหมือนเลขฐาน 2 เลย

  $(1234)_{16} = (1 \times 16^3) + (2 \times 16^2) + (3 \times 16) + 4 =$

  Decimal Value 4660

  $(3BA4)_{16} = (3 \times 16^3) + (11 \times 16^2) + (10 \times 16) + 4 =$

  Decimal Value 15268

Converting Decimal to Hexadecimal

หลักการเลขฐานทั่วไป

- หารเลขฐาน 10 ด้วย 16 จนกว่าจะหารอีกไม่ได้
- นำเศษของการหารแต่ละครั้งมาใช้เป็นแต่ละหลักในเลขฐาน 16

Division	Quotient	Remainder
422 / 16	26	6
26 / 16	1	A
1 / 16	0	1

least significant digit

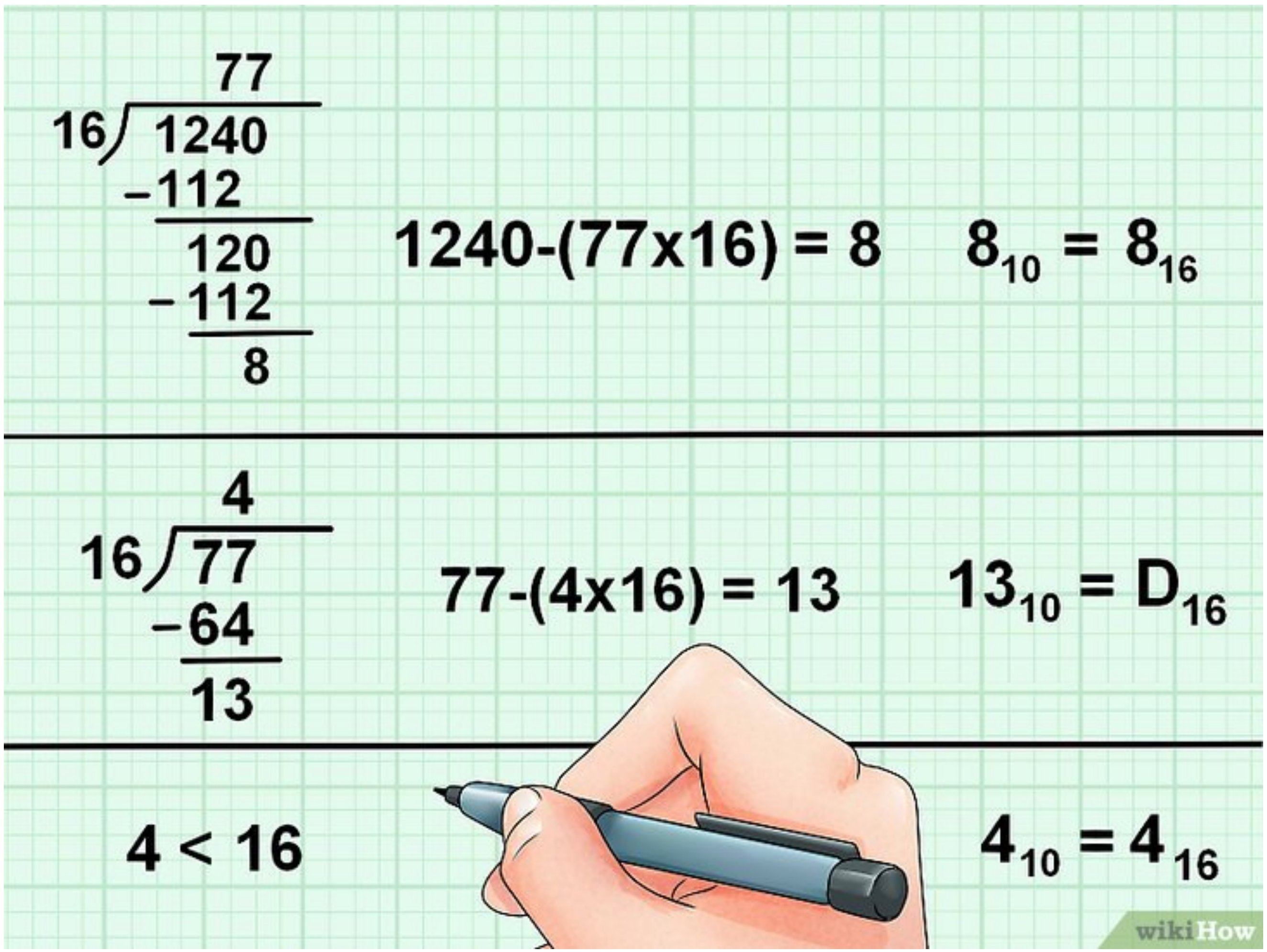
most significant digit

stop when quotient is zero

Decimal 422 = 1A6 hexadecimal

Example #5 Dec2Hex

ตั้งหารเหมือน binary



Example #6 Hex2Dec

$$10 \times 16^3 = 40960$$

$$5 \times 16^2 = 1280$$

$$9 \times 16^1 = 144$$

$$12 \times 16^0 = +\ 12$$

$$42{,}396$$

3	2	1	0
A	5	9	C

Exercise #3 : Binary to Hexadecimal

- แปลงเลขฐาน 2 ต่อไปนี้ ให้เป็นเลขฐาน 16

  1 A 8 9
  - 00011010 10001001 1A89

  B C 5 9
  - 10111100 01011001 BC59

  15 0 10 10
  - 11110000 10101010 F0AA

  1 2 3 4
  - 00010010 00110100 1234

Exercise #3 : Binary to Hexadecimal

- แปลงเลขฐาน 2 ต่อไปนี้ ให้เป็นเลขฐาน 16
  - 00011010 10001001 -> 1A89
  - 10111100 01011001 -> BC59
  - 11110000 10101010 -> F0AA
  - 00010010 00110100 -> 1234

Exercise #4 : Hexadecimal to Binary

- แปลงเลขฐาน 16 ต่อไปนี้ ให้เป็นเลขฐาน 2
  - F1BC 1111 0001 1011 1100
  - ED56 1110 1101 0101 0110
  - 1988 0001 1001 1000 1000
  - 2560 0010 0101 0110 0000

Exercise #4 : Hexadecimal to Binary

- แปลงเลขฐาน 16 ต่อไปนี้ ให้เป็นเลขฐาน 2
  - F1BC -> 1111 0001 1011 1100
  - ED56 -> 1110 1101 0101 0110
  - 1988 -> 0001 1001 1000 1000
  - 2560 -> 0010 0101 0110 0000

Exercise #5 : Hexadecimal to Decimal

- แปลงเลขฐาน 16 ต่อไปนี้ ให้เป็นเลขฐาน 10
  - F1 15 1 1111 0001 = 241
  - ED56 14 13 5 6 1110 1101 0101 0110 = เยอะ

Exercise #5 : Hexadecimal to Decimal

- แปลงเลขฐาน 16 ต่อไปนี้ ให้เป็นเลขฐาน 10
  - F1 -> 241
  - D56 -> 3,414

Exercise #6 : Decimal to Hexadecimal

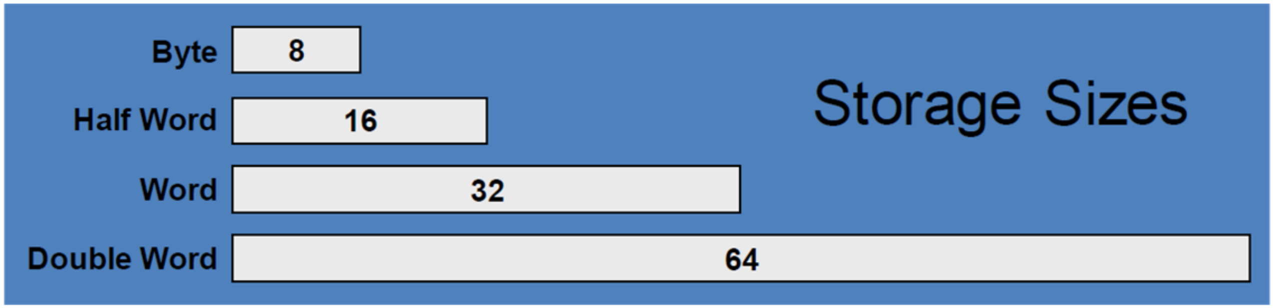
- แปลงเลขฐาน 10 ต่อไปนี้ ให้เป็นเลขฐาน 16
  - 123 78
  - 200 C8

Exercise #6 : Decimal to Hexadecimal

- แปลงเลขฐาน 10 ต่อไปนี้ ให้เป็นเลขฐาน 16
  - 123 -> 78
  - 200 -> C8


Kahoot!

Integer Storage Sizes



Storage Type	Unsigned Range	Powers of 2
Byte	0 to 255	0 to $(2^8 - 1)$
Half Word	0 to 65,535	0 to $(2^{16} - 1)$
Word	0 to 4,294,967,295	0 to $(2^{32} - 1)$
Double Word	0 to 18,446,744,073,709,551,615	0 to $(2^{64} - 1)$

What is the largest 20-bit unsigned integer?

↳ ไม่ติดเครื่องหมาย

Answer: $2^{20} - 1 = 1,048,575$

Binary Arithmetic Rules

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0 (carry 1) การทด

- 1+1+1 = 1 (carry 1) ทดมากสุดแค่ 1

Binary Addition

- Start with the least significant bit (rightmost bit)
- Add each pair of bits
- Include the carry in the addition, if present

carry		1	1	1	1				
	0	0	1	1	0	1	1	0	(54)
+	0	0	0	1	1	1	0	1	(29)
	0	1	0	1	0	0	1	1	(83)
bit position:	7	6	5	4	3	2	1	0	

Example #7 : Binary Addition

- ให้บวกเลขฐาน 2

Calculate the binary sum $0001\ 0011\ 1101 + 0000\ 1011\ 0111$.

$$
\begin{array}{rcccccccccccccl}
 & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & & \\
 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & \textit{Addend} \\
+ & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & \textit{Augend} \\
\hline
 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & \textit{Sum}
\end{array}
\quad (1)
$$

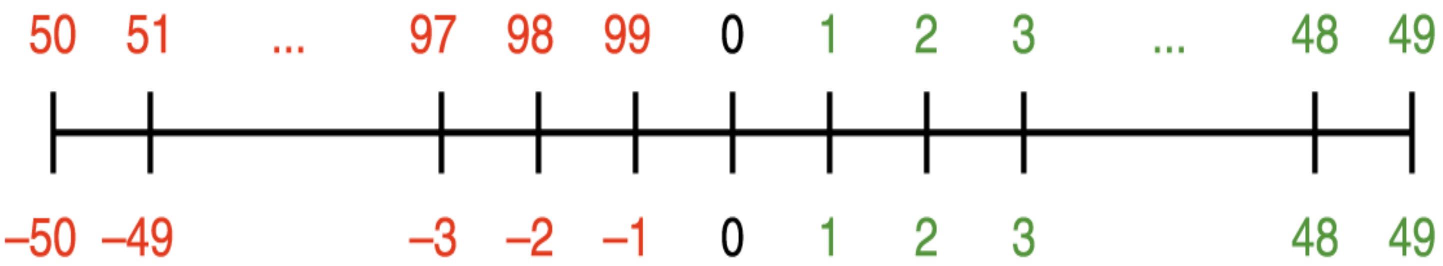
Exercise #7 Binary Addition

0	1	1	1	1	1	0	0
1	0	1	1	0	1	1	1

0	1	1	0	1	1	0	0
0	1	0	1	0	0	1	1

Representing Negative Values

- เนื่องจากโลกดิจิทัล จำกัด ดังนั้นหากมีเลข 8 บิต ซึ่งแทนค่าได้ 256 ค่า คือ เลขบวก 0-255 แต่หากนำจะต้องใช้กับเลขลบด้วย ก็ไม่สามารถใช้ 0-255 ได้ เพราะต้องนำส่วนหนึ่งไปแทนค่าเลขลบ
- รูปด้านล่างนี้ เป็นตัวอย่างของการแทนค่าลบแบบหนึ่ง โดยแบ่ง 0-99 เป็นจำนวน -50 -> 49



Representing Negative Values

- จากหลักการ a – b = a + (-b) ดังนั้นจะใช้เลขลบมาบวกแทน

Signed-Magnitude	New Scheme
5 + – 6 – 1	5 + 94 99
– 4 + 6 2	96 + 6 2
– 2 + – 4 – 6	98 + 96 94

Now you try it

48
- 1
47

How does it work in the new scheme?

Exercise #8 : Binary Subtraction

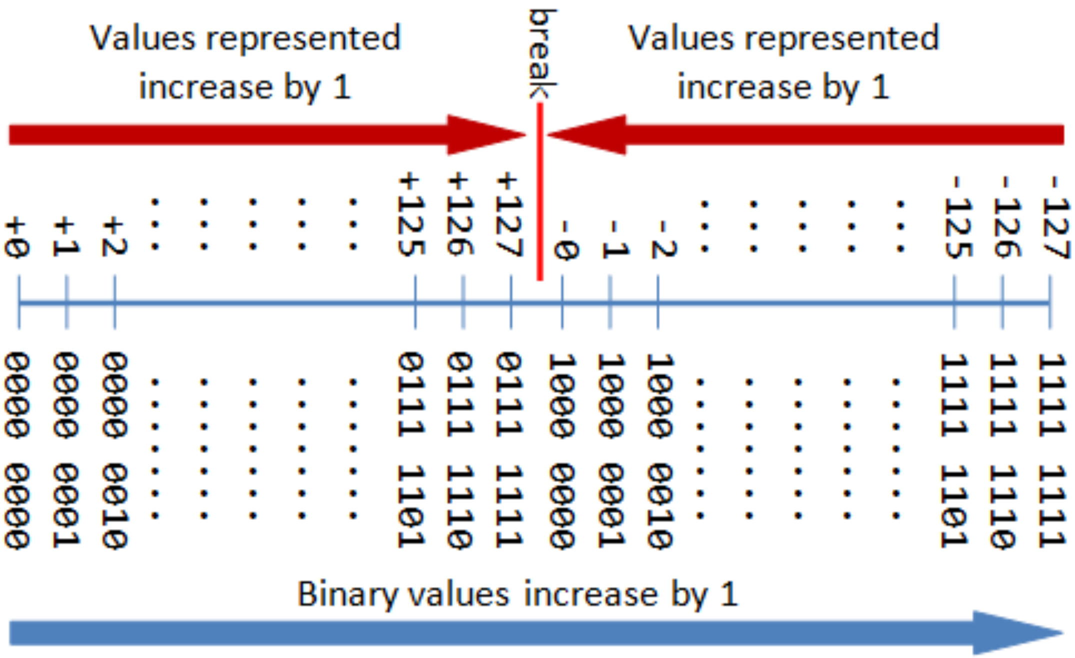
Signed-Magnitude	New Scheme	Add Negative
–5	95	95
– 3	– 3	+ 97
–8		92

Try

4	- 4	-4
- 3	+3	+ -3

Representing Negative Values

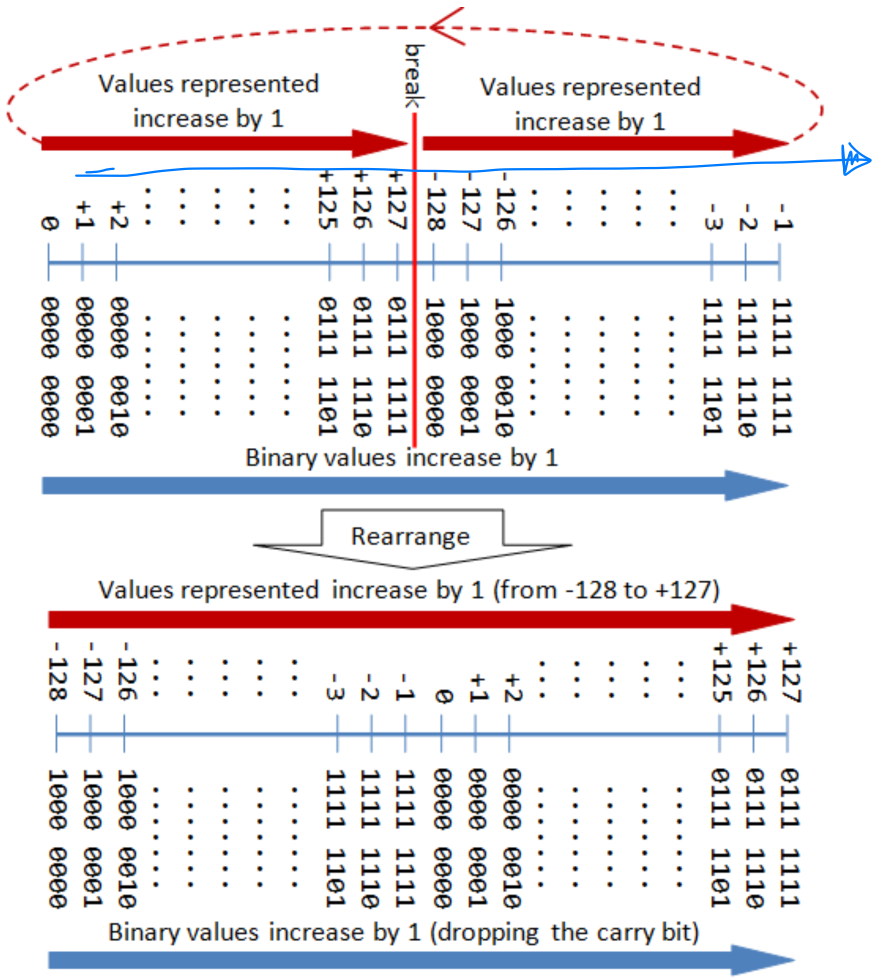
- กรณีเลข 8 บิต ซึ่งแทนค่าได้ 256 ค่า คือ เลขบวก 0-255 แต่หากนำจะต้องใช้กับเลขลบด้วย ก็ไม่สามารถใช้ 0-255 ได้ เพราะต้องนำส่วนหนึ่งไปแทนค่าเลขลบ



Sign-Magnitude Representation

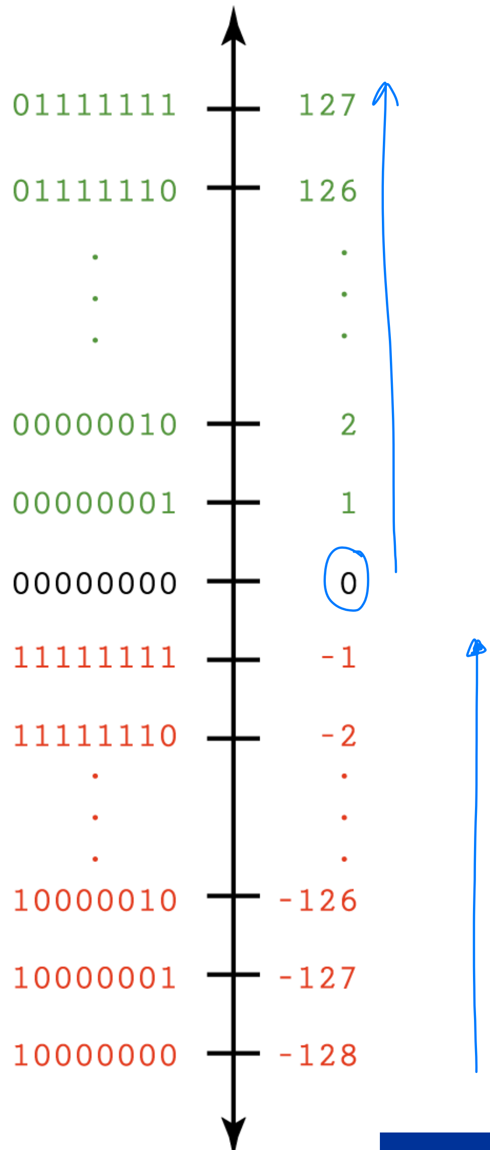
Representing Negative Values

- การแทนค่าจำนวนลบตามรูปที่แล้ว มีปัญหาที่มีเลข 0 จำนวน 2 ค่า จึงได้มีการปรับระบบใหม่ โดยการสลับด้านตามรูป ซึ่งจะทำให้ตัวเลขเรียงกันเพิ่มขึ้นทีละ 1
- ระบบที่ออกแบบใหม่นี้เรียกว่า 2’s Complement



2’s Complement Representation

Representing Negative Values



Two's Complement

(Vertical line is easier to read)

1	0	1	1	0	1	0	0
-128	64	32	16	8	4	2	1

$$= -128 + 32 + 16 + 4 = -76$$

Forming the Two's Complement

starting value	00100100 = +36
step1: reverse the bits (1's complement)	11011011
step 2: add 1 to the value from step 1	+ 1
sum = 2's complement representation	11011100 = -36

ผลบวกของตัวเลขและ 2’s complement ของมันจะต้องได้เป็น 0

00100100 + 11011100 = 00000000 (8-bit sum) (ไม่คิดตัวทด)

Another way to obtain the 2's complement:

Start at the least significant 1

Leave all the 0s to its right unchanged

Complement all the bits to its left

Binary Value

= 00100$\boxed{1}$00 least significant 1

2's Complement

= 11011$\boxed{1}$00

Exercise #9 : Two’s Complement

- 10001011 (คิดเครื่องหมาย) เมื่อแปลงเป็นฐาน 10 จะได้เท่ากับเท่าไร

+2 +1

$-128 + 8 = -117$

- 11111011 (คิดเครื่องหมาย) เมื่อแปลงเป็นฐาน 10 จะได้เท่ากับเท่าไร

$-128 + 64 + 32 + 16 + 8 + 2 + 1 = -5$

Tip

~~1111~~1011

$-8 + 2 + 1$

$= -5$

the same

Exercise #9 : Two’s Complement

- 10001011 (คิดเครื่องหมาย) เมื่อแปลงเป็นฐาน 10 จะได้เท่ากับเท่าไร

  -117

- 11111011 (คิดเครื่องหมาย) เมื่อแปลงเป็นฐาน 10 จะได้เท่ากับเท่าไร

  -5

Advantages of Two's Complement Notation

- ง่ายในการบวกเลข 2 จำนวน

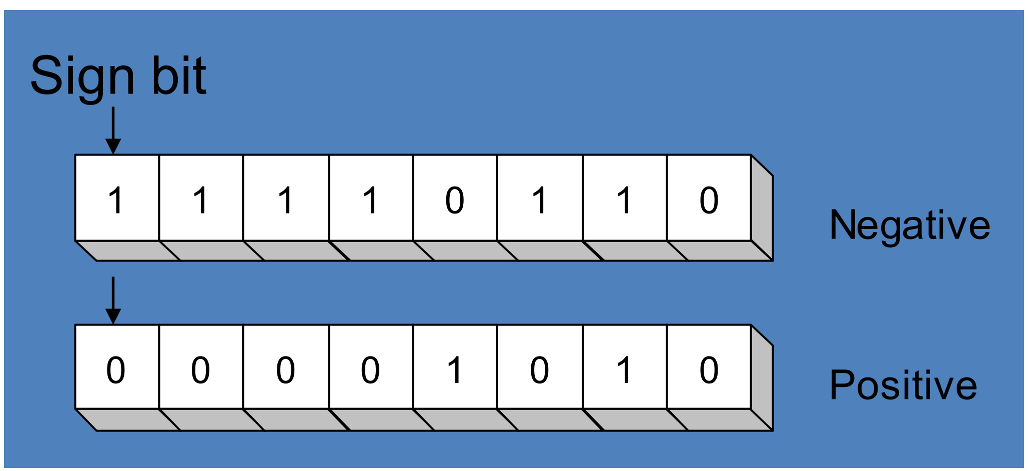
	0001	+1
+	0101	+5
	0110	**+6**

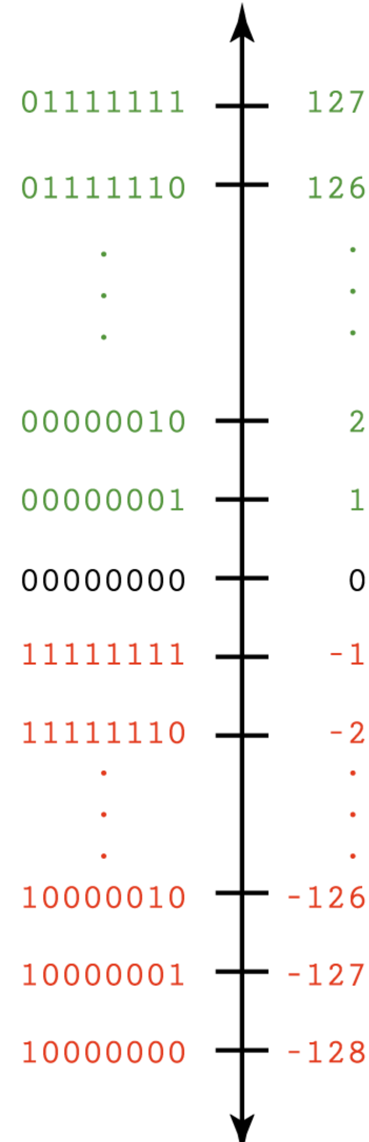
	1000	-8
+	0101	+5
	1101	**-3**

- การลบเลขก็ทำได้ง่าย
- การคูณก็เป็นการบวกหลายครั้ง
- การหารก็เป็นการลบหลายครั้ง
- ปัจจุบัน Two's Complement จึงมีการใช้งานในระบบคอมพิวเตอร์มาก

Sign Bit

- ในระบบเลขแบบคิดเครื่องหมาย จะใช้บิตซ้ายสุดในการบอกว่าเป็นเลข บวก หรือ ลบ
- 1 = ลบ 0 = บวก
- สำหรับเลขฐาน 16 จะเป็นบิตซ้ายสุด เช่น 7F จะเป็นบวก แต่ 80 จะเป็นลบ





Binary Subtraction

- เมื่อจะทำ A – B ขั้นแรกจะต้องแปลง B ให้อยู่ในรูป two’s Complement
- แล้ว บวก A + (-B)

borrow:	1 1 1
	0 1 0 0 1 1 0 1
–	0 0 1 1 1 0 1 0
	0 0 0 1 0 0 1 1

carry:	1 1 1 1	
	0 1 0 0 1 1 0 1	
+	1 1 0 0 0 1 1 0	(2's complement)
	0 0 0 1 0 0 1 1	(same result)

- สำหรับตัวทดสุดท้ายที่เป็น 1 จะไม่นำมาคิด

Example #9 Binary Subtraction

-

0	1	1	1	1	1	0	0
0	0	1	1	0	1	1	1

+

0	1	1	1	1	1	0	0
1	1	0	0	1	0	0	1
0	1	0	0	0	1	0	1

Example #8 Binary Subtraction

-

0	1	1	1	1	1	0	0
1	0	1	1	0	1	1	1

+

0	1	1	1	1	1	0	0
0	1	0	0	1	0	0	1
1	1	0	0	0	1	0	1

Exercise #10 : Binary Subtraction

- ให้แสดงการลบโดยใช้ Two's Complement
  - 55-33
  - 33-55

Ranges of Signed Integers

- For n-bit signed integers: Range is -2^{n-1} to $(2^{n-1} - 1)$
- Positive range: 0 to $2^{n-1} - 1$
- Negative range: -2^{n-1} to -1

Storage Type	Unsigned Range	Powers of 2
Byte	–128 to +127	-2^7 to $(2^7 - 1)$
Half Word	–32,768 to +32,767	-2^{15} to $(2^{15} - 1)$
Word	–2,147,483,648 to +2,147,483,647	-2^{31} to $(2^{31} - 1)$
Double Word	–9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	-2^{63} to $(2^{63} - 1)$

Practice: What is the range of signed values that may be stored in 20 bits?

Homework #1

- ให้ทำโจทย์ Homework #1 ในระบบ (คะแนน 2 เปอร์เซนต์)
- โหลด App Socrative หรือ เข้าเว็บ https://www.socrative.com/
- เลือก student login
- ในช่อง Room Name ป้อน ITC
- ในช่อง Enter Your Name ให้ใส่รหัสนักศึกษา
- ป้อนคำตอบจนครบทุกข้อ

For your attention