

# ระบบเลข และการแทนรหัสข้อมูล

หัวข้อ 1 ระบบเลข และการแทนรหัสข้อมูล

หัวข้อ 2 การคำนวณเลขฐาน

หัวข้อ 3 การแทนรหัสข้อมูลในหน่วยความจำ: การแทนรหัสข้อมูลที่เป็นอักขระ

หัวข้อ 4 การแทนรหัสข้อมูลในหน่วยความจำ: การแทนรหัสข้อมูลที่เป็นตัวเลข

# ระบบเลข และการแทนรหัสข้อมูล

- ระบบเลขฐานต่างๆ (เน้น ฐาน 2 ฐาน 8 และ ฐาน 16)
  - การแปลงเลขฐาน
  - ความสัมพันธ์ของเลขฐาน 2 ฐาน 8 และฐาน 16
- การคำนวณทางคณิตศาสตร์ในระบบเลขฐาน
- การแทนรหัสข้อมูลในระบบ BCD, EBCDIC, ASCII
- การแทนรหัสข้อมูลของจำนวนเลข จำนวนเต็ม จำนวนจริง

# ระบบเลขฐาน (Numeral System)

- แสดงโดย (b=base or radix)

$$(a_n a_{n-1} \cdots a_1 a_0 . c_1 c_2 c_3 \cdots)_b = \sum_{k=0}^n a_k b^k + \sum_{k=1}^{\infty} c_k b^{-k}.$$

- ค่าประจำหลัก คือ ค่าของเลขฐานนั้นๆ ยกกำลังตามตำแหน่งหลัก
- Least significant digit : คือเลขที่มีค่าประจำหลักน้อย
- Most significant digit : คือเลขที่มีค่าประจำหลักสูง
- การเขียนเลขฐานต้องมีค่าฐานกำกับ ยกเว้นฐาน 10

# ตัวเลขในฐานต่าง ๆ

- ฐาน 2 มีเลข 0,1
- ฐาน 8 มีเลข 0,1,2,3,4,5,6,7
- ฐาน 10 มีเลข 0,1,2,3,4,5,6,7,8,9
- ฐาน 16 มีเลข 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

# การแปลงเลขฐานที่ใช้บ่อย

- การแปลงเลขฐานใดๆ เป็น ฐาน 10
- การแปลงเลขฐาน 10 เป็น ฐานใดๆ
- การแปลงเลขฐาน 2 เป็น ฐาน 8 ฐาน 16
- การแปลงเลขฐาน 8 ฐาน 16 เป็น ฐาน 2

# การแปลงเลขฐานใด ๆ เป็นฐาน 10

- อาศัยค่าประจำหลักคูณตัวเลขแต่ละหลัก นำผลคูณที่ได้มารวมกัน

$(2542)_{10}$

ค่าประจำหลัก       $10^3$     $10^2$     $10^1$     $10^0$

ค่าเลข                      2      5      4      2

$$\begin{aligned}\text{ค่าฐาน 10} &= (2 \times 10^3) + (5 \times 10^2) + (4 \times 10^1) + (2 \times 10^0) \\ &= 2000 + 500 + 40 + 2 = 2542_{10}\end{aligned}$$

$(2542)_8$

ค่าประจำหลัก       $8^3$     $8^2$     $8^1$     $8^0$

ค่าเลข                      2      5      4      2

$$\begin{aligned}\text{ค่าฐาน 8} &= (2 \times 8^3) + (5 \times 8^2) + (4 \times 8^1) + (2 \times 8^0) \\ &= 1024 + 320 + 32 + 2 = 1370_{10}\end{aligned}$$

# การแปลงเลขฐานใด ๆ เป็นฐาน 10

- ตัวอย่าง  $1101_2 = (?)_{10}$

$2^3$   $2^2$   $2^1$   $2^0$  ← ค่าประจำหลัก

|   |   |   |   |
|---|---|---|---|
| 1 | 1 | 0 | 1 |
|---|---|---|---|

$$\begin{aligned} &= (1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) \\ &= 8+4+0+1 \\ &= 13 \end{aligned}$$

# การแปลงเลขฐาน 10 เป็นฐานใด ๆ

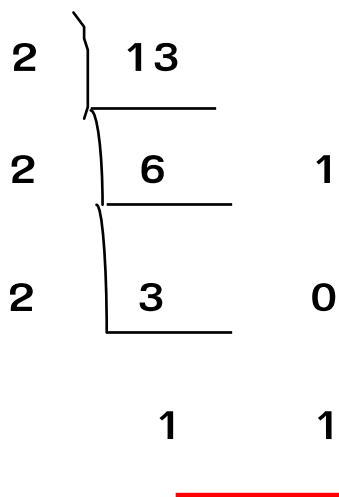
- กรณี เลขจำนวนเต็ม
- ใช้หลัก MODULO คือ
  - เลขฐาน 10 เป็นตัวตั้งหารด้วยเลขฐานที่กำลังจะแปลง
  - ให้เก็บเศษจากการหาร
  - หารเลขต่อไปจนกระทั่งไม่สามารถหารได้
  - นำเศษของการหารมาวางต่อกัน เศษตัวสุดท้ายเป็น Most significant



# ตัวอย่างการแปลงเลขฐาน 10 เป็นฐานใดๆ

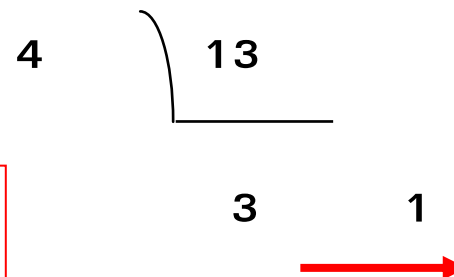
- เลขจำนวนเต็ม

$$13_{10} = ( \quad )_2$$



ตอบ  $1101_2$

$$13_{10} = ( \quad )_4$$



ตอบ  $31_4$

คำตอบอ่านจาก  
ล่างขึ้นบน

# การตรวจสอบคำตอบ

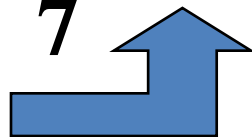
$$632_{10} = (?)_9$$

$$9 \overline{) 632}$$

$$9 \overline{) 70} \quad 2$$

7

7



**Ans.**  $772_9$

ลองตรวจคำตอบ

$772_9$

ค่าประจำหลัก  $9^2 \ 9^1 \ 9^0$

$81 \ 9 \ 1$

$$= 7 \times 81 + 7 \times 9 + 2$$

$$= 567 + 63 + 2$$

$$= 632_{10}$$

# การแปลงเลขฐาน 10 เป็นฐานใด ๆ

- กรณี เลขจำนวนจริง: การแปลงแบ่งเป็น 2 ส่วน
- ส่วนหน้าจุดทศนิยมใช้วิธี MODULO
- ส่วนเลขหลังจุดทศนิยม
  - คูณเลขหลังจุดทศนิยมด้วยฐานที่ต้องการ แล้วบันทึกเฉพาะเลขหน้าจุดทศนิยมของผลคูณ
  - ส่วนเลขหลังจุดนำมาคูณต่อ จนครบจำนวนตำแหน่งหลังจุดทศนิยมที่ต้องการ

# ตัวอย่าง การแปลงเลขฐาน 10 เป็นฐานใด ๆ

## ● เลขจำนวนจริง

$$13.4_{10} = ( \quad )_2$$

$$\begin{array}{r} 2 \overline{) 13} \\ \underline{10} \phantom{0} \\ 3 \phantom{0} \\ 2 \overline{) 6} \\ \underline{6} \phantom{0} \\ 0 \phantom{0} \\ 2 \overline{) 3} \\ \underline{2} \phantom{0} \\ 1 \phantom{0} \\ 2 \overline{) 1} \\ \underline{0} \\ 1 \end{array}$$

$$\begin{array}{r} .4 * 2 = 0 .8 \\ .8 * 2 = 1 .6 \\ .6 * 2 = 1 .2 \\ .2 * 2 = 0 .4 \end{array}$$

ทศนิยมคำตอบ  
อ่านจากบนลง  
ล่าง

หน้าทศนิยม  
คำตอบอ่านจาก  
ล่างขึ้นบน

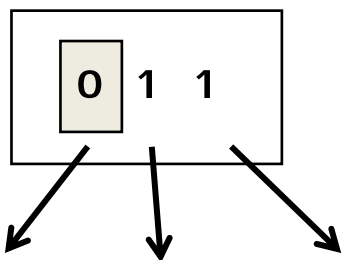
**Ans: 1101.0110<sub>2</sub>**

# การแปลงเลขฐาน 2 เป็น ฐาน 8 ฐาน 16

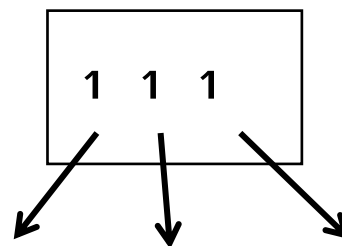
- หลักการ ใช้การจัดกลุ่มบิต
  - เลขฐาน 2 เป็น ฐาน 8 จัดกลุ่มละ 3 บิต
  - เลขฐาน 2 เป็น ฐาน 16 จัดกลุ่มละ 4 บิต
- โดยเริ่มจากบิตที่อยู่ใกล้จุดทศนิยม หากกลุ่มสุดท้ายไม่ครบเติม 0 ไปข้างหน้า
- ทำการหาค่าเป็นเลขฐาน 10 ของแต่ละกลุ่มที่จัดแล้ว
- นำมาต่อกันเพื่อเป็นคำตอบโดยเรียงตามหลักซ้ายไปขวา

# ตัวอย่างการแปลงเลขฐาน 2 เป็นฐาน 8

- จำนวนเต็ม  $11111_2 = (37)_8$



$$(0 \cdot 4) + (1 \cdot 2) + (1 \cdot 1)$$



$$(1 \cdot 4) + (1 \cdot 2) + (1 \cdot 1)$$

แปลงค่าเป็นเลขฐาน10  
ของแต่ละกลุ่ม

3

นำมาต่อกันเรียง  
จากซ้ายไปขวา

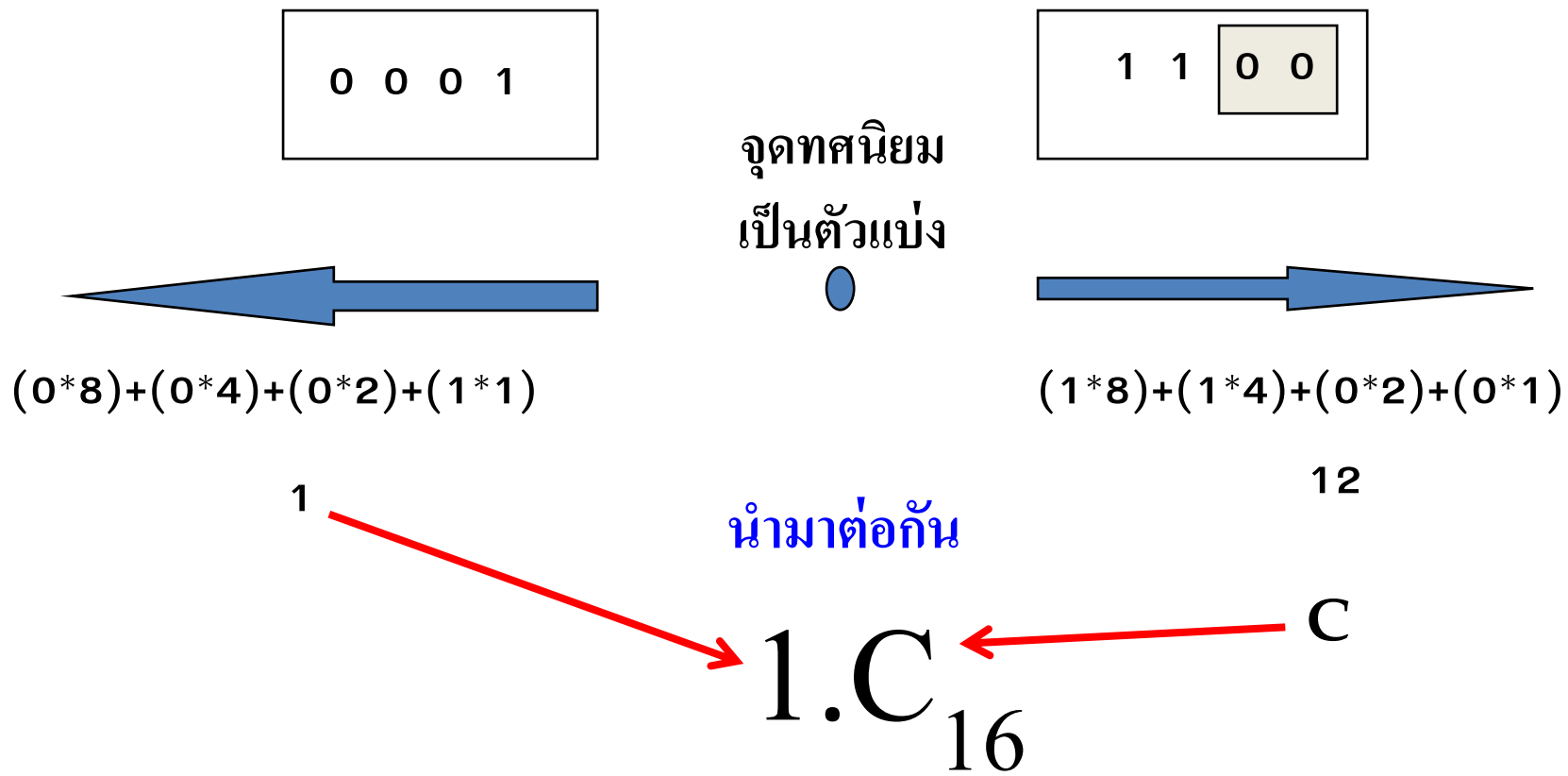
7

แปลงค่าเป็นเลขฐาน10  
ของแต่ละกลุ่ม

$37_8$

# ตัวอย่างการแปลงเลขฐาน 2 เป็นฐาน 16

- เลขจำนวนจริง  $0001.11_2 = (1.C)_{16}$



# การแปลงเลขฐาน 8 ฐาน 16 เป็น ฐาน 2

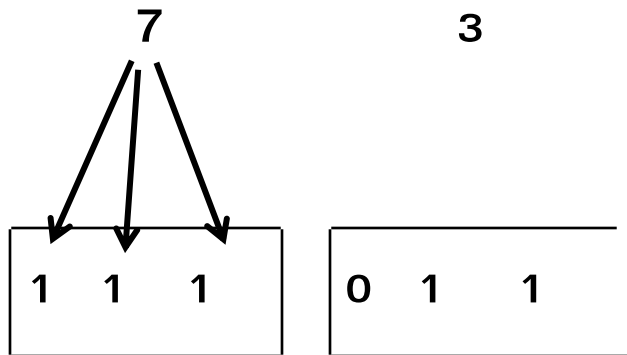
- ใช้หลักการกระจายเลขแต่ละหลักออกเป็น บิต ด้วยเลขฐาน 2
- แยกเลขฐาน 8 หนึ่งหลัก กระจายเป็นเลขฐาน 2 ได้ 3 บิต
- แยกเลขฐาน 16 หนึ่งหลัก กระจายเป็นเลขฐาน 2 ได้ 4 บิต



# ตัวอย่างการแปลงเลขฐาน 8 ฐาน 16 เป็น ฐาน 2

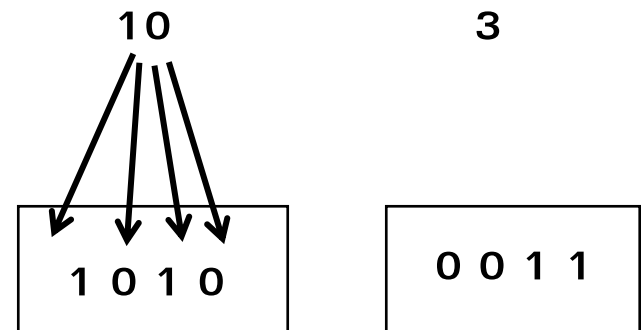
- กรณี เลขจำนวนเต็ม

$$73_8 = ( \quad )_2$$



**Ans:**  $111011_2$

$$A3_{16} = ( \quad )_2$$

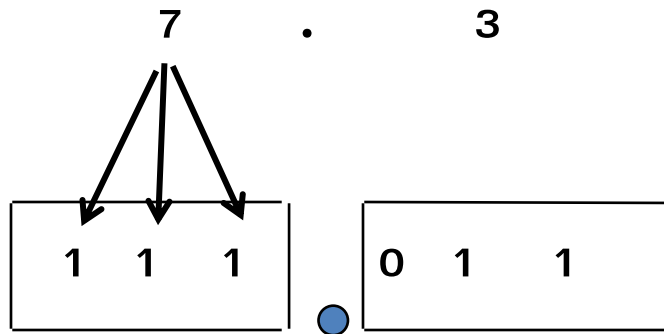


**Ans:**  $10100001_{16}$

# ตัวอย่างการแปลงเลขฐาน 8 ฐาน 16 เป็น ฐาน 2

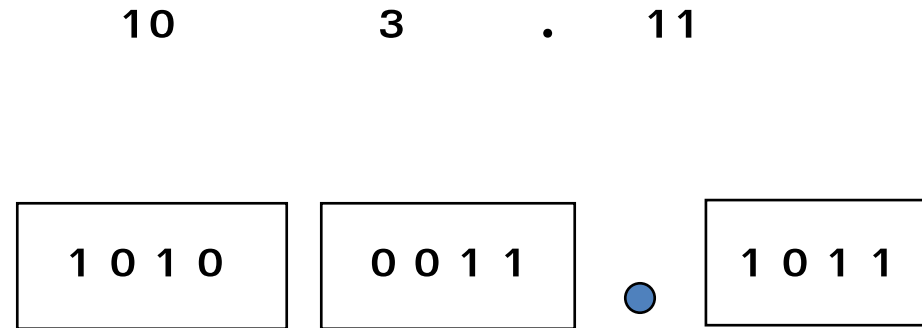
- กรณี เลขจำนวนจริง

$$7.3_8 = ( \quad )_2$$



**Ans:**  $111.011_2$

$$A3.B_{16} = ( \quad )_2$$



**Ans:**  $10100001.1011_{16}$

# แบบฝึกหัด 1 จงแสดงวิธีการแปลงเลขฐานต่อไปนี้เป็น

## ➤ ระหว่างเลขฐานใด ๆ และเลขฐาน 10

➤  $1001_2 = ( ? )_{10}$

➤  $1001_5 = ( ? )_{10}$

➤  $12 = ( ? )_2$

➤  $123 = ( ? )_5$

➤  $2550 = ( ? )_{16}$

## • ระหว่างเลขฐาน 2, 8, และ 16

•  $10101_2 = ( ? )_8$

•  $101011.0110_2 = ( ? )_{16}$

•  $21.8_{16} = ( ? )_2$

•  $235_8 = ( ? )_8$

# การคำนวณเลขฐาน

- การบวกเลขฐาน
- การลบเลขฐาน
- การลบเลขฐาน โดยการใช้ complement

## ● การบวก

$$\begin{array}{r} 1101.11_2 \\ + \\ 0111.01_2 \\ \hline 10101.00_2 \end{array}$$

$$\begin{array}{r} 534_5 \\ + \\ 123_5 \\ \hline \end{array}$$

คำนวณไม่ได้  
เพราะไม่มีเลข 5  
ในฐาน 5

$$\begin{array}{r} F31C_{16} \\ + \\ 2350_{16} \\ \hline 1166C_{16} \text{ คำตอบที่ถูกต้อง} \end{array}$$

- การลบเลข

$432_5$

—

$143_5$

————

$234_5$

$1011.011_2$

—

$0111.101_2$

————

$0011.110_2$

# การบวก-ลบ เลขฐานสอง และการตรวจสอบคำตอบด้วยฐานสิบ

$$1001_2 + 1011_2 = ( ? )$$

$$\begin{array}{r} 1001_2 \\ + 1011_2 \\ \hline 10100_2 \\ \hline \hline \end{array}$$

ตรวจสอบคำตอบด้วยตนเอง

โดยเทียบกับเลขฐานสิบ คือ

$$\begin{aligned} 1001_2 + 1011_2 \\ 9_{10} + 11_{10} &= 20 \\ &= 10100_2 \end{aligned}$$

$$1101_2 - 110_2 = ( ? )$$

$$\begin{array}{r} 1101_2 \\ - 110_2 \\ \hline 0111_2 \\ \hline \hline \end{array}$$

เทียบกับเลขฐานสิบคือ

$$\begin{aligned} 1101_2 - 110_2 \\ 13_{10} - 6_{10} &= 7_{10} \\ &= 111_2 \end{aligned}$$

# การหาคอมพลีเมนต์ (Complement)

- Complement ของฐานใดๆ (ให้ R แทนฐาน) มี 2 ประเภท  
คือ R-1's complement และ R's complement

| <u>ฐาน</u> | <u>R-1's Comp.</u> | <u>R's Comp</u> |
|------------|--------------------|-----------------|
| 2          | 1's Comp           | 2's Comp        |
| 8          | 7's Comp           | 8's Comp        |
| 10         | 9's Comp           | 10's Comp       |



# การหาค่า R-1 คอมพลิเมนต์

- การหาคอมพลิเมนต์ ที่ R-1 ของเลขใดๆ
  - นำค่าสูงสุดที่เป็นไปได้ของเลขที่มีจำนวนหลักเท่ากับตัวเลขที่ต้องการ ลบด้วยตัวเลขที่ต้องการหาคอมพลิเมนต์
  - ผลที่ได้คือ คอมพลิเมนต์ของเลขจำนวนนั้น
  - เลข  $29_{10}$  มีสองหลัก ค่าสูงสุดของเลขสองหลัก คือ 99
    - 9's Comp :  $99 - 29 = 70$
  - เลข  $101.11_2$  มีค่าสูงสุดคือ 111.11
    - 1's Comp :  $111.11 - 101.11 = 010.00$

# การหาค่า R คอมพลีเมนต์

- การหาคอมพลีเมนต์ ที่ R ของเลขใดๆ
  - นำค่าสูงสุดของเลขที่มีหลักเท่ากับเลขนั้นบวกด้วยเลขที่ทำให้เกิดการเปลี่ยนหลัก แล้วจึงลบด้วยเลขจำนวนนั้นอีกที
  - เลข  $29_{10}$  มี R คอมพลีเมนต์ เป็น
    - 10's Comp :  $(99 + 1) - 29 = 71$
  - เลข  $101.11_2$  มี R คอมพลีเมนต์เป็น
    - 2's Comp :  $(111.11 + .01) - 101.11 = 0010.01$

# ข้อสังเกตเกี่ยวกับคอมพลีเมนต์

- การหา R-1 Complement คือ การนำเลขจำนวนนั้นลบออกจากเลขสูงสุด ที่มีจำนวนหลักเท่ากับเลขนั้น
- 1'S Complement ในเลขฐาน 2 คือ การเปลี่ยนค่าของแต่ละบิตให้ตรงกันข้าม
- ค่า R-1 Complement มีค่าน้อยกว่า R Complement อยู่ 1 เสมอ ณ หลักขวามือสุด
  - $2'S \text{ Comp} = 1'S \text{ Comp} + 1$  ที่หลักขวามือสุด

# การลบเลขแบบคอมพลิเมนต์(ลบเลขด้วยวิธีบวก)

- นำเลขตัวลบไปหาคอมพลิเมนต์
- นำคอมพลิเมนต์ที่หาได้ บวก กับเลขตัวตั้ง
- ผลลัพธ์ที่ได้ถ้ามีเลขเกินหลัก
  - กรณี R'S Comp. ให้ตัดทิ้ง
  - กรณี R-1'S Comp. ให้นำเลขที่เกินหลัก บวกกับผลลัพธ์
- ผลที่ได้คือคำตอบ

# ตัวอย่างการลบเลขแบบคอมพลีเมนต์ (R-1)

$$432_5 - 143_5 = ?$$

①  $444 - 143 = 301$

②  $432 + 301$

③ 
$$\begin{array}{r} 432 + \\ 301 \\ \hline 1233 + \\ 1 \\ \hline 234 \end{array}$$

$$432_5 - 143_5 = 234_5$$

เทียบได้กับฐาน 10  
 $(117-48)_{10} = 69_{10}$

$$1011.011_2 - 111.101_2 = ?$$

①  $1111.111 - 0111.101 = 1000.010$

②  $1011.011 + 1000.010$

③ 
$$\begin{array}{r} 1011.011 + \\ 1000.010 \\ \hline 10011.101 + \\ 1 \\ \hline 0011.110 \end{array}$$

$$1011.011_2 - 111.101_2 = 11.11_2$$

เทียบได้กับฐาน 10  
 $(11.375-7.625)_{10} = 3.75_{10}$

# ตัวอย่างการลบเลขแบบคอมพลีเมนต์ (R)

$$432_5 - 143_5 = ?$$



①  $1000 - 143 = 302$

② 
$$\begin{array}{r} 432 + \\ 302 \\ \hline 234 \end{array}$$

③  $\downarrow$   
ตัดทิ้ง

$$432_5 - 143_5 = 234_5$$

เทียบได้กับฐาน 10  
 $(117 - 48)_{10} = 69_{10}$

$$1011.011_2 - 111.101_2 = ?$$

①  $10000.000 - 0111.101 = 1000.011$

② 
$$\begin{array}{r} 1011.011 + \\ 1000.011 \\ \hline 0011.110 \end{array}$$

③  $\downarrow$   
ตัดทิ้ง

$$1011.011_2 - 111.101_2 = 11.11_2$$

เทียบได้กับฐาน 10  
 $(11.375 - 7.625)_{10} = 3.75_{10}$

# การแทนรหัสข้อมูลในหน่วยความจำ (Data Representation)

- การแทนรหัสข้อมูลที่เป็นอักขระ (Alphanumeric Data Representation)
- การแทนรหัสข้อมูลที่เป็นจำนวนเลข (Numeric Data Representation)
  - เลขจำนวนเต็ม (Integer Representation)
  - เลขที่มีจุดทศนิยม (Floating Point Representation)

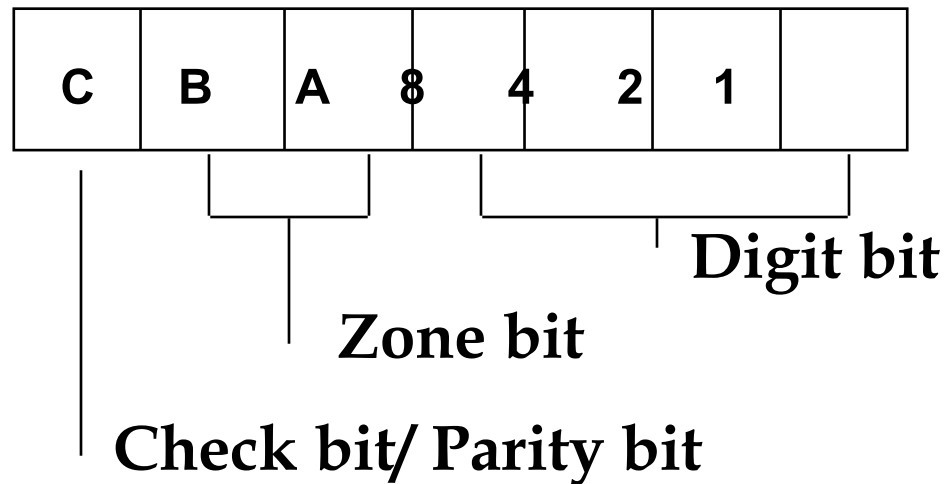
# การแทนข้อมูลที่เป็นอักขระ

- รหัส BCD : Binary Coded Decimal
- รหัส EBCDIC: Extended Binary Coded Decimal  
Interchange Code
- รหัส ASCII : American Standard Code for Information  
Interchange



# รหัส BCD (Binary-Coded Decimal)

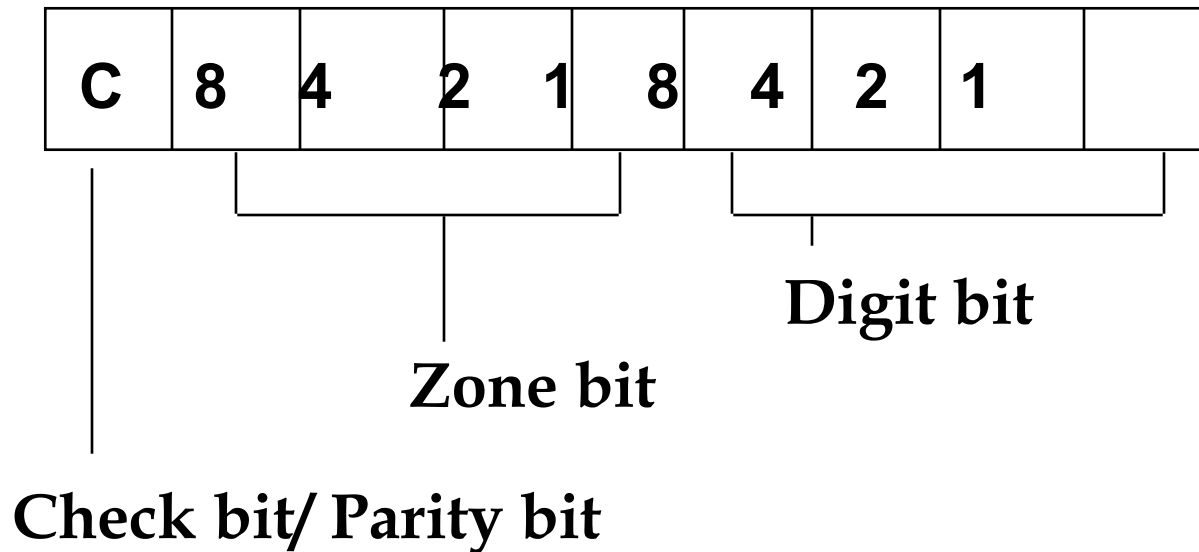
- ใช้ 6 บิตแทนอักขระ 1 ตัว
- ระบบนี้แทนอักขระได้ 64 ตัว ( $2^6$  รูปแบบ)



- การแทนรหัส BCD
  - อักขระแบบตัวเลข (0 - 9) Zone Bit จะเป็น 00
  - อักขระแบบตัวอักษร หรือ สัญลักษณ์พิเศษ Zone bit เป็น 11

# รหัส EBCDIC

- ใช้ 8 บิตแทนอักขระ 1 ตัว
- ระบบนี้แทนอักขระได้ 256 ตัว ( $2^8$  รูปแบบ)



# รหัส EBCDIC

- การบันทึกข้อมูลในระบบ EBCDIC มี 2 แบบ

- การบันทึกแบบ Zoned Decimal

- การบันทึกแบบ Packed Decimal

- การบันทึกแบบตัวเลข Zone bit มีค่าเป็น

1111 สำหรับเลขที่ไม่มีเครื่องหมายนำหน้า ( 12 ,F)

1100 สำหรับเลขที่มีเครื่องหมายบวก และ (+12 ,C)

1101 สำหรับเลขที่มีเครื่องหมายลบ (-12 ,D)

# Extended Binary Coded Decimal Interchange Code

| Bit Positions |   |   |   | 4   | 0               | 0               | 0               | 0   | 0  | 0   | 0   | 0   | 1  | 1   | 1  | 1               | 1   | 1   | 1   | 1 |   |
|---------------|---|---|---|-----|-----------------|-----------------|-----------------|-----|----|-----|-----|-----|----|-----|----|-----------------|-----|-----|-----|---|---|
|               |   |   |   | 3   | 0               | 0               | 0               | 0   | 1  | 1   | 1   | 1   | 0  | 0   | 0  | 0               | 1   | 1   | 1   | 1 | 1 |
|               |   |   |   | 2   | 0               | 0               | 1               | 1   | 0  | 0   | 1   | 1   | 0  | 0   | 1  | 1               | 0   | 0   | 1   | 1 | 1 |
|               |   |   |   | 1   | 0               | 1               | 0               | 1   | 0  | 1   | 0   | 1   | 0  | 1   | 0  | 1               | 0   | 1   | 0   | 1 | 0 |
| 8             | 7 | 6 | 5 |     |                 |                 |                 |     |    |     |     |     |    |     |    |                 |     |     |     |   |   |
| 0             | 0 | 0 | 0 | NUL | SUH             | STX             | ETX             | PF  | HT | LC  | DEL |     |    | SRM | VT | FF              | CR  | SU  | SI  |   |   |
| 0             | 0 | 0 | 1 | DLE | DC <sub>1</sub> | DC <sub>2</sub> | DC <sub>3</sub> | RES | NL | BS  | IL  | CAN | EM | CC  |    | IFS             | IGS | IRS | IUS |   |   |
| 0             | 0 | 1 | 0 | DS  | SOS             | FS              |                 | BYP | LF | EOB | PRE |     |    | SM  |    |                 | ENQ | ACK | BEL |   |   |
| 0             | 0 | 1 | 1 |     |                 | SYN             |                 | PN  | RS | UC  | EOT |     |    |     |    | DC <sub>4</sub> | NAK |     | SUB |   |   |
| 0             | 1 | 0 | 0 | SP  |                 |                 |                 |     |    |     |     |     |    | ¢   | .  | <               | (   | +   |     |   |   |
| 0             | 1 | 0 | 1 | &   |                 |                 |                 |     |    |     |     |     |    | !   | \$ | *               | )   | ;   | ~   |   |   |
| 0             | 1 | 1 | 0 | -   | /               |                 |                 |     |    |     |     |     |    |     | '  | %               | -   | >   | ?   |   |   |
| 0             | 1 | 1 | 1 |     |                 |                 |                 |     |    |     |     |     |    | :   | #  | @               | ,   | =   | ''  |   |   |
| 1             | 0 | 0 | 0 |     | a               | b               | c               | d   | e  | f   | g   | h   | i  |     |    |                 |     |     |     |   |   |
| 1             | 0 | 0 | 1 |     | j               | k               | l               | m   | n  | o   | p   | q   | r  |     |    |                 |     |     |     |   |   |
| 1             | 0 | 1 | 0 |     |                 | s               | t               | u   | v  | w   | x   | y   | z  |     |    |                 |     |     |     |   |   |
| 1             | 0 | 1 | 1 |     |                 |                 |                 |     |    |     |     |     |    |     |    |                 |     |     |     |   |   |
| 1             | 1 | 0 | 0 |     | A               | B               | C               | D   | E  | F   | G   | H   | I  |     |    |                 |     |     |     |   |   |
| 1             | 1 | 0 | 1 |     | J               | K               | L               | M   | N  | O   | P   | Q   | R  |     |    |                 |     |     |     |   |   |
| 1             | 1 | 1 | 0 |     |                 | S               | T               | U   | V  | W   | X   | Y   | Z  |     |    |                 |     |     |     |   |   |
| 1             | 1 | 1 | 1 | 0   | 1               | 2               | 3               | 4   | 5  | 6   | 7   | 8   | 9  |     |    |                 |     |     | □   |   |   |

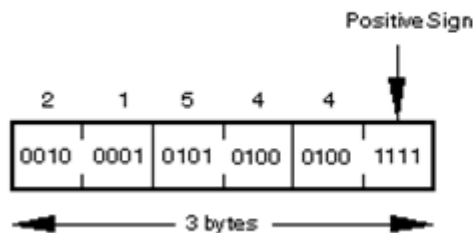
# การแทนข้อมูลแบบ Packed Decimal

- เป็นการเปลี่ยนลักษณะการเก็บรหัส EBCDIC ให้ใช้ในการคำนวณ
- การเปลี่ยนนี้จำนวนหลักสามารถยืดหยุ่นได้
- ไม่ใช้กับตัวเลขที่เป็นจุดทศนิยม

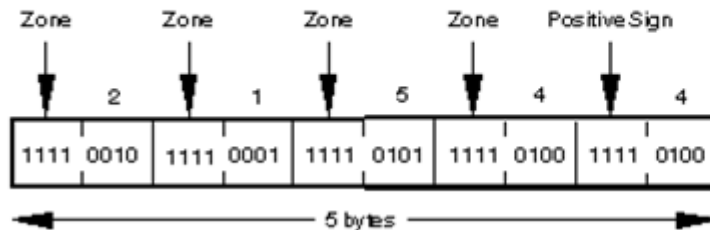
# วิธีการ Packed Decimal

- สลับส่วน Zone bit และ Digit bit ของไบต์ขวาสุด
- ตัด Zone bit ของไบต์ ที่เหลือ
- บีบข้อมูลซึ่งเป็น Digit Bit เข้ามา

Packed Decimal Format:



Zoned Decimal Format:



# รหัส ASCII

- มี 2 ชนิด คือ 7 บิต กับ 8 บิต กำหนดให้ตัวเลขมีค่าน้อยกว่าตัวอักษรเป็นรหัสที่นิยมในปัจจุบัน
- ลักษณะคล้าย EBCDIC มี Zone bit
  - 0101 และ 011 สำหรับตัวเลข
  - 1010 และ 100 สำหรับตัวอักษร



# Parity bit หรือ Check bit

- เป็นบิตที่ใช้ตรวจสอบการแทนรหัส มี 2 ระบบ
- Even Parity ระบบจำนวนคู่ ระบบนี้ต้องมีบิตที่เป็นเลข 1 ทั้งหมดมีจำนวนเป็นเลขคู่
- Odd Parity ระบบจำนวนคี่ ระบบนี้ต้องมีบิตที่เป็นเลข 1 ทั้งหมดมีจำนวนเป็นเลขคี่

1 1 1 0 0 0 1

0 1 1 0 1 0 1

0 1 1 0 0 0 1

1 1 1 0 1 0 1

# การแทนรหัสข้อมูลที่เป็นตัวเลข

- การแทนข้อมูลแบบนี้ กำหนดเนื้อที่ในหน่วยความจำมีขนาดตายตัว (Fixed length word) สำหรับแทนตัวเลข 1 จำนวน
- Half-word ใช้เนื้อที่ 2 ไบต์
- Full-word ใช้เนื้อที่ 4 ไบต์
- Double-word ใช้เนื้อที่ 8 ไบต์

# การแทนเลขจำนวนเต็ม

- Sign Magnitude / Pure binary code
- 2's Complement
- 1's Complement

\*\*หมายเหตุ โดยทั่วไปแล้วการแทนเลขด้วยระบบ 1's Complements, 2's Complements จะใช้สำหรับ**การแทนเลขจำนวนเต็มลบ** ดังนั้นถ้าเลขที่ต้องการแทนด้วยระบบ 1's Complements, 2's Complements เป็นเลขจำนวนเต็มบวก ก็จะแสดงคำตอบคล้ายกับใช้วิธี Sign Magnitude ที่ต่างคือวิธี Complements ไม่มีการกั้น sign bit จำนวน 1 bit

# Sign Magnitude

- ระบบนี้บิตซ้ายสุดแทนเครื่องหมายเรียกว่า Sign bit ที่เหลือแทนขนาดของจำนวนเลข เรียกว่า Magnitude



↓  
1 แทน ลบ  
0 แทน บวก

1 Word = 4 Byte = 32 Bit



# แสดงการแทนค่า แบบ 1's Complement

- การแสดงจำนวนเลขในระบบ 1's Complements ดังนี้
  - เลขจำนวนเต็มบวก แทน ด้วย true binary โดยไม่ต้องกัน sign bit เช่น 32, 109 , 2008 เป็นต้น
  - เลขจำนวนเต็มลบ จะใช้ค่า 1's Complements ของเลขจำนวนบวกนั้น แทนเป็นเลขลบของเลขบวกจำนวนนั้น เช่น -32, -109 , -2008 เป็นต้น

# ตัวอย่าง การแทนค่า 1' Complements

- เช่น ต้องการแทนค่า 28 ด้วยระบบ 1's Comp

แปลง 28 จำนวนเต็มเป็นฐาน 2 ในรูปแบบ 32 บิต

$$28 = 11100_2$$

MSB

LSB

000000.....0000000000 1 1 1 0 0

ค่า true binary ของเลข 28

จะได้คำตอบคือ

0 0000000 00000000 00000000 00011100

แทนเลขจำนวนเต็มบวก 28 ด้วยวิธี 1's Complements

# ตัวอย่าง การแทนค่า 1's Complements

- เช่น ต้องการแทนค่า -28 ด้วยระบบ 1's Comp

true binary ของเลขบวก

$$28 = 11100_2$$

28 จำนวนเต็มเป็นเลขฐาน 2  
ในรูปแบบ 32 บิต

MSB

LSB

00000000 00000000 00000000 00011100

หาค่า 1's Comp ของ 28 ที่แปลงเป็นฐาน 2 โดยทำการกลับบิตข้อมูลของทุกบิต ในรูปแบบ 32 บิต นั่นคือจะใช้ 1's comp ของเลขบวกแทนเลขลบของเลขบวกนั้น

11111111 11111111 11111111 11100011

ดังนั้นจะแทน -28 ด้วยวิธี 1's Comp ในรูปแบบ 32 บิต ได้ดังนี้

1 11111111 11111111 11111111 11100011



# แสดงการแทนค่าแบบ 2's Complements

- แบบ 2's Complements
  - เป็นระบบที่นิยมใช้ในเครื่องคอมพิวเตอร์
- การแสดงจำนวนเลขในระบบ 2's Complements ดังนี้
  - **เลขบวก** แทนด้วยค่า true binary ของเลขบวกนั้น โดยไม่ต้องกั้ sign bit
  - **เลขลบ** จะใช้ค่า 2's Complements ของเลขบวกจำนวนนั้น แทนเป็นเลขลบของเลขบวกจำนวนนั้น

# ตัวอย่าง การแทนค่า 2's Complements

- เช่น ต้องการแทนค่า -28 ด้วยระบบ 2's Comp

True binary ของ 28

$$28 = 11100_2$$

แปลง 28 จำนวนเต็มเป็นฐาน 2  
ในรูปแบบ 32 บิต

MSB

LSB

00000000 00000000 0..0 00011100

หาค่า 1's Comp ของ 28 ที่แปลงเป็นฐาน 2

11111111 11111111 1..1 11100011

1<sup>+</sup>

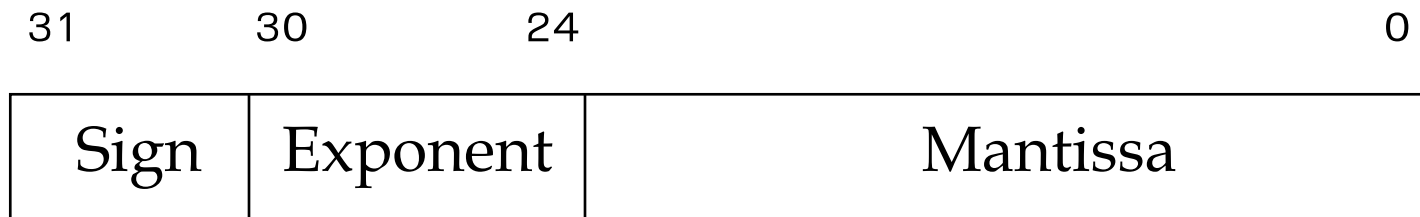
ได้คำตอบ คือ

11111111 11111111 1..1 11100100

หาค่า 2's Comp ของ 28 ที่แปลงเป็นฐาน 2 โดยนำ 1 มาบวกกับหลักขวาสุด

# Floating Point Representation

- $R = \pm .M * B^{\pm E}$
- วิธี Excess 64 (แสดง Exponent ด้วยจำนวน 7 บิต)



S Sign แทนเครื่องหมาย บวก ลบ ของจำนวนเลข 1 bit

E Exponent ส่วนที่ยกกำลัง จะใช้ bit ที่ 24-30 = 7 bits

M Mantissa เลขที่อยู่หลังจุด จะใช้ bit ที่ 0-23 = 24 bits

# ขั้นตอนการทำ Floating point

- เปลี่ยนเลข ไปเป็นเลขฐาน 16
- Normalisation เลขฐาน 16 (มีเลขหลังจุด และ ยกกำลัง)  
ในรูป  $R = \pm .M * B^{\pm E}$
- เปลี่ยน Sign, Exponent, Mantissa เป็นเลขฐาน 2
  - Sign : 1 แทนค่าลบ 0 แทนค่าบวก
  - Mantissa : เปลี่ยนเป็นฐาน 2 เติมเลขโดยต่อจาก Exponent จากซ้าย มา ขวาที่เหลือเติมศูนย์เพื่อให้ mantissa ครบ 24 bit

# ขั้นตอนการทำ Floating point (1/2)

- ใช้ Excess notation
  - ใช้เลข  $n$  บิตแสดง Exponent
  - เนื่องจาก Exponent มีได้ทั้งบวกและลบ จึงต้องแบ่งช่วงค่าที่เป็นไปได้ของเลข  $n$  บิตเป็นสองช่วง
  - จุดแบ่งของสองช่วงอยู่ที่จุดที่บิตสูงสุดเป็น 1 และบิตที่เหลือเป็น 0
    - สำหรับกรณี Excess 64 ( $= 40_{16}$ ) จุดแบ่งจะอยู่ที่ 1000000

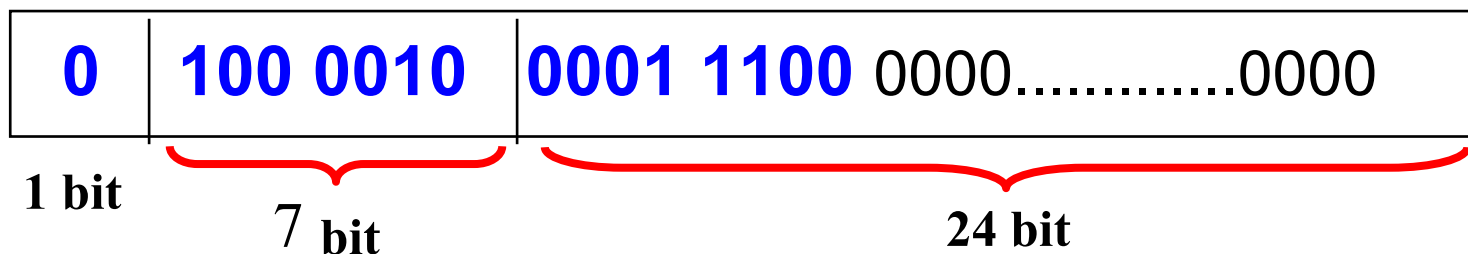
# ขั้นตอนการทำ Floating point (2/2)

- การบันทึกค่า Exponent
  - หาก Exponent เท่ากับ 0 จะแทนด้วย 1000000
  - หาก Exponent มากกว่า 0 ให้นำ True exponent มาบวกกับ  $40_{16}$  นั่นคือ
    - $1000000 + \text{True exponent}$  (ในฐานสอง)
    - หรือ  $40_{16} + \text{True exponent}$  (ในฐาน 16)
  - หาก Exponent น้อยกว่า 0 ให้นำ True exponent มาลบจาก  $40_{16}$  แทน

# ตัวอย่าง ทำ Floating point ที่เป็นเลขจำนวนเต็ม

- $28 \rightarrow 1C_{16}$
- $1C_{16} = (+.1C * 16^2)_{16}$
- Sign  $\rightarrow 0$
- Mantissa  $\rightarrow 0001\ 1100_2$
- Exponent  $\rightarrow 40_{16} + 2_{16} = 42_{16} \rightarrow 100\ 0010_2$

ถอยไปหลังจุดที่ตัวก็จะได้เลขยกกำลังของ  $(16_{16})$  เท่ากับจำนวนการถอย นั่นคือจะได้ true exponent คือ  $2_{16}$



# ตัวอย่าง ทำ Floating point ที่เป็นเลขทศนิยม

- $-28.5 \rightarrow -1C.8_{16}$

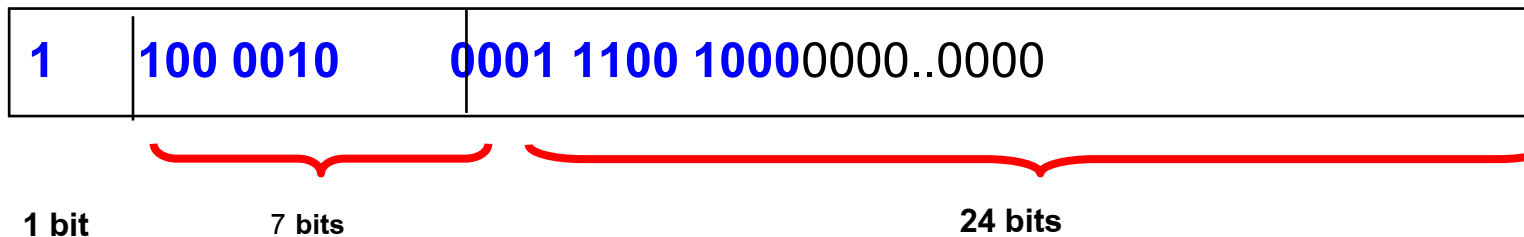
- $-1C.8_{16} = (-.1C8 * 16^2)_{16}$

\*\* สังเกตว่า ถอยไปหลังจุด 2  
ตำแหน่งจะได้ true exponent คือ  $2_{16}$

- Sign  $\rightarrow 1$

- Mantissa  $\rightarrow 0001\ 1100\ 1000_2$

- Exponent  $\rightarrow 40_{16} + 2_{16} = 42_{16} \rightarrow 100\ 0010_2$





# แบบฝึกหัดที่ 2

1. จงแสดงวิธีการลบเลขแบบธรรมดา , R complement และ R-1

Complement ของ  $(5624-567)_8 = (?)_8$

2. จงแสดง Floating Point Representation ของเลขฐานต่อไปนี้อย่างน้อยด้วยวิธี

Excess 64

- $75.25_{10}$

- $10011.10_2$

# แบบฝึกหัด

## 1. จงแสดงวิธีการแปลงเลขฐานต่อไปนี้

$$- 100.1_5 = ( ? )_{10}$$

$$- 123 = ( ? )_5$$

$$- 101011.011_2 = ( ? )_{16}$$

$$- 235_8 = ( ? )_{16}$$

## 2. การแทนค่าเลขจำนวนเต็มด้วยวิธี 2's complement ของ -33

## 3. การแสดง Floating Point Representation ของเลขฐานต่อไปนี้ด้วยวิธี Excess 64

a) 75.25

b) -100011011.102

## 4. แสดงการลบเลขฐานต่อไปนี้ แบบ 1's complement และแบบ 2's complement

$$1101_2 - 100_2$$

# 1. จงแสดงวิธีการแปลงเลขฐานต่อไปนี้

$$\square 100.1_5 = ( ? )_{10}$$

$$\square 123 = ( ? )_5$$

$$\square 101011.011_2 = ( ? )_{16}$$

$$\square 235_8 = ( ? )_{16}$$

$$\bullet 25.02_{10}$$

$$\bullet 503_5$$

$$\bullet 2B.6_{16}$$

$$\bullet 9D_{16}$$

# 2. การแทนค่าเลขจำนวนเต็มด้วยวิธี 2's complement ของ -33

**111111111011111<sub>2</sub>** กรณีใช้ 2 ไบต์

# 3. การแสดง Floating Point Representation ของเลขฐานต่อไปนี้ด้วยวิธี Excess 64

a) 75.25

b) -100011011.10<sub>2</sub>

**01000010010010110100000000000000**

**11000011000100011011100000000000**

# 4. แสดงการลบเลขฐานต่อไปนี้ แบบ 1's complement และแบบ 2's complement

**1101<sub>2</sub> - 100<sub>2</sub>** คำตอบที่ได้คือ **1001<sub>2</sub>**