w10-Lab

Data Representation Part II

Rational Number and Floating Point

Assembled for 204111
by Kittipitch Kuptavanich
Ratsameetip Wita

Integer and Real Number

- Digital Data
- Integer Representation
- Binary Arithmetic
- Negative Integer Encoding
- **Real Number and Floating Point Representation**
- **Rounding**

Real Numbers in the Machine

- **Real Numbers** หรือจำนวนจริง เป็นค่าประเมินแบบต่อเนื่อง ดังแสดงได้ในระบบเส้นจำนวน
- ระบบตัวเลข จำเป็นต้องมีการแสดงผลค่าจำนวนจริงด้วยการประมาณค่า
- พิจารณาการแทนข้อมูลฐาน 10
  - เราไม่สามารถใช้เลขทศนิยมที่มีการจำกัดตำแหน่งเพื่อแทนค่าที่แท้จริงของ 1/3 หรือ 5/7
  - เนื่องจากเป็นทศนิยมไม่รู้จบ: $0.\dot{3}$ และ $0.\dot{7}1428\dot{5}$
  - 0.33333 ใกล้เคียงค่าจริงมากกว่า 0.33

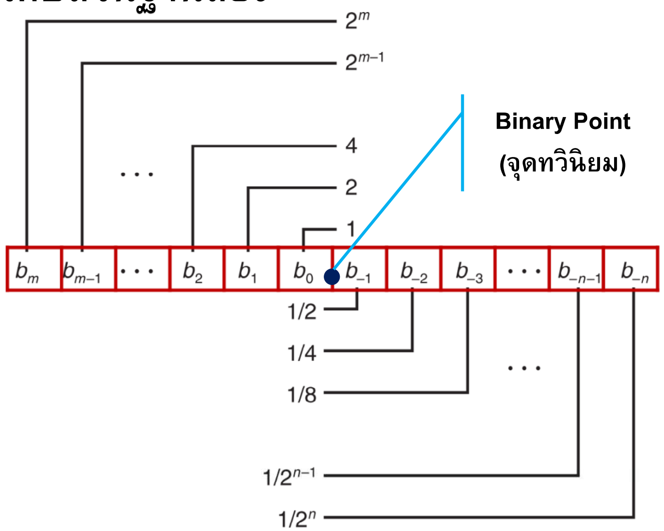
Fractional Binary Numbers

- ในระบบเลขฐาน 2 นั้นมีข้อจำกัดเช่นเดียวกัน
  - แทน<u>ค่าจริง</u>ของตัวเลขได้ เฉพาะตัวเลขที่สามารถเขียนในรูป $x \times 2^y$ เท่านั้น
  - นอกจากนั้นจะเป็น<u>ค่าประมาณ</u> – ความใกล้เคียงกับค่าจริง*<u>ขึ้นกับ</u>*จำนวนตำแหน่งที่ใช้แสดงค่า

ตำแหน่งหลังจุดทศนิยมมาก = ค่าใกล้เคียงจำนวนจริงมากขึ้น

Fractional Binary Numbers [2]

- เลขเศษส่วนฐานสอง



Fractional Binary Number

Representation	Value	Decimal
0.0_2	$\frac{0}{2}$	0.0_{10}
0.01_2	$\frac{1}{4}$	0.25_{10}
0.010_2	$\frac{2}{8}$	0.25_{10}
0.0011_2	$\frac{3}{16}$	0.1875_{10}
0.00110_2	$\frac{6}{32}$	0.1875_{10}
0.001101_2	$\frac{13}{64}$	0.203125_{10}
0.0011010_2	$\frac{26}{128}$	0.203125_{10}
0.00110011_2	$\frac{51}{256}$	0.19921875_{10}

Practice Problem 1

เติมตารางต่อไปนี้ให้สมบูรณ์

Fractional value	Binary representation	Decimal representation
$\frac{1}{8}$	0.001	0.125
$\frac{3}{4}$	______	______
$\frac{25}{16}$	______	______
______	10.1011	______
______	1.001	______
______	______	5.875
______	______	3.1875

Floating Point Representation

- ในการแสดงค่าประมาณของเลขจำนวนจริงในระบบคอมพิวเตอร์ จะมีการแทนค่าของข้อมูลในรูปแบบของเลขทวินิยม* (Fractional Binary)
- IEEE standard 754
  - ตั้งขึ้นในปี 1985
  - มาตรฐานกลางสำหรับ Floating Point
  - รองรับโดย CPU ส่วนมาก

*แนวความคิดเหมือนกับเลขทศนิยมในเลขฐานสิบ ทวิ แปลว่า สอง ทศ แปลว่า สิบ

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

Floating Point Representation

- การแทนค่า **Floating Point** อยู่ในรูปแบบ

Sign Bit　　Exponent

$$(-1)^{s} \times M \times 2^{E}$$

Significand or Mantissa

ลักษณะเดียวกันกับการเขียน $204111 = 2.04111 \times 10^{5}$ ในฐาน 10

- ***Sign Bit*** s เป็น bit ที่บอกว่าเป็นจำนวนบวกหรือลบ
- ***Significand*** M เป็นจำนวนในช่วง 1.0 ถึง 2.0 (**M: mantissa**)
- ***Exponent*** E กำหนดขนาดของจำนวนที่ต้องการแทนค่าในรูปกำลังที่ E ของ 2 (E สามารถมีค่าเป็น **+ 0 หรือ - ได้**)

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

Floating Point Representation

s	exp	frac
s	E	M

- **Base 10**

$$(-1)^{s} \times M \times 10^{E}$$

- **Example**

อายุของจักรวาล

13700000000 ปี

สามารถเขียนอยู่ในรูป floating point ดังนี้

$+1.37 \times 10^{10}$ ปี

s **= 0,** M **= 1.37,** E **= 10**

- **Base 2**

$$(-1)^{s} \times M \times 2^{E}$$

- **Example**

$.75 = \frac{1}{2} + \frac{1}{4} = 2^{-1} + 2^{-2}$

$5.75_{10} = 5 + 0.75$

$= 101_2 + 0.11_2$

$= 101.11_2$

$= 1.0111_2 \times 10^{10}$

binary ($= 2^2$)

s **= 0,** M **= 1.0111,** E **= 10**

`float` ใน Python (range ฐาน10 = $10^{-308} - 10^{308}$, ทศนิยม 16-17 ตำแหน่ง)

s	exp	frac
1	8-bits	23-bits

s	exp	frac
1	11-bits	52-bits

`32 bit (single-precision) vs 64 bit (double-precision)`

Floating Point Rounding

- เนื่องจากข้อมูลแบบ **Floating Point** จะเป็นค่าแบบประมาณ เช่น $\frac{1}{3}$, π สิ่งที่อาจเกิดขึ้นได้คือ **Error** จากการคำนวณ เช่น

```
>>> x = 1/10
>>> x
0.1
>>> y = 2/10
>>> y
0.2
>>> x + y
0.30000000000000004
>>> from decimal import Decimal
>>> Decimal(x)
Decimal('0.1000000000000000055511151231257827021181583404541015625')
>>> Decimal(y)
Decimal('0.200000000000000011102230246251565404236316680908203125')
>>> Decimal(x+y)
Decimal('0.3000000000000000444089209850062616169452667236328125')
>>>
```

python แสดงค่าที่ผ่านการ rounding แล้ว

ค่าที่เกิดจาก Accumulation Error

ค่าจริงของข้อมูล x, y (Decimal)

Rounding in Binary

- **พิจารณาการหาค่า $\frac{1}{10}$ โดยคำนวณด้วยการหารเลขฐาน 2**

```
          .000110011...
1010 ) 1.000000000...
         1010
          1100
          1010
          10000
           1010
            1100
            1010
              10...
```

ได้คำตอบเป็นเลขซ้ำของ **11001100...**

ชุดตัวเลขซ้ำ

Rounding

- เนื่องจากการแสดงค่าแบบ **Floating Point** สามารถแทนค่าได้ในช่วงความละเอียดที่จำกัด ทำให้จำเป็นต้องมีการประมาณค่าโดยการปัดเศษทิ้ง (Rounding) เช่น ปัด 1.4 ⟶ 1 หรือ ปัด 1.6 ⟶ 2
- ต้องพิจารณาตัดสินใจ กรณีค่าที่ต้องการปัดเศษอยู่กึ่งกลางระหว่างผลลัพธ์ที่เป็นไปได้ 2 จำนวน
  - เช่น 1.5 (1 vs 2)
  - ปัดไปทางไหน? เพราะอะไร?
  - 1.5 + 2.5 + 3.5 + 4.5 vs 2 + 3 + 4 + 5

Rounding [2]

- ใน IEEE floating-point format มีวิธีการปัดเศษ ที่ต่างกันถึง 4 วิธี

IEEE Default Mode

Mode	\$1.40	\$1.60	\$1.50	\$2.50	\$-1.50
Round-to-even	\$1	\$2	\$2	\$2	\$-2
Round-toward-zero	\$1	\$1	\$1	\$2	\$-1
Round-down	\$1	\$1	\$1	\$2	\$-2
Round-up	\$2	\$2	\$2	\$3	\$-1

Round to Even

- การปัดเศษเลขคู่ เป็น **Default Mode**
  - แก้ปัญหากรณีค่าความคลาดเคลื่อนจากการปัดเศษของเลขหลาย ๆ จำนวน (ลงเสมอ หรือ ขึ้นเสมอ) สะสมรวมกัน
  - ใช้ในกรณีตัวเลขที่ต้องการปัดอยู่กึ่งกลางเท่านั้น
    - เช่น กรณีปัดให้เป็นทศนิยม 2 ตำแหน่ง

1.2349999	1.23	(Less than half way)
1.2350001	1.24	(Greater than half way)
1.2350000	1.24	(Half way—round up)
1.2450000	1.24	(Half way—round down)

กรณีอื่น ๆ ปัดตามหลักคณิตศาสตร์ปกติ

from Carnegie Mellon University's 15213 course slide by Greg Kesden

Rounding Binary Numbers

- **Binary Fractional Numbers**
  - เป็นเลขคู่ก็ต่อเมื่อบิทขวาสุดหลังจากการปัดเศษ มีค่าเป็น 0
  - เศษที่ต้องการปัดจะมีค่ากึ่งกลางก็ต่อเมื่ออยู่ในรูป = $100...0_2$
- **Examples**
  - ปัดเหลือแค่ 2 ตำแหน่งหลัง binary point

Value	Binary	Rounded	Action	Rounded Value
2 3/32	10.00011_2	10.00_2	(<1/2—down)	2
2 3/16	10.00110_2	10.01_2	(>1/2—up)	2 1/4
2 7/8	10.11100_2	11.00_2	(1/2—up)	3
2 5/8	10.10100_2	10.10_2	(1/2—down)	2 1/2

Conclusions

- **Real Number and Fractional Binary Numbers**
- **Floating Point Representation**
- **Rounding in Binary System**
  - **Rounding Standard**

Practice Problem 1: KEY

Fractional value	Binary representation	Decimal representation
$\frac{1}{8}$	0.001	0.125
$\frac{3}{4}$	0.11	0.75
$\frac{25}{16}$	1.1001	1.5625
$\frac{43}{16}$	10.1011	2.6875
$\frac{9}{8}$	1.001	1.125
$\frac{47}{8}$	101.111	5.875
$\frac{51}{16}$	11.0011	3.1875