2장. 정보의 표현과 처리

2.1 정보의 저장
2.2 정수의 표시
2.3 정수의 산술 연산

2.4 부동소수점

2.4 부동소수점

실수 표현

- 실수(Real numbers)

  $3.14159265...{}_{ten}$ (pi), $2.71828...{}_{ten}$ (e),

  0.000000001_{ten} , $0.1_{ten} \times 10^{-8}$ or $1.0_{ten} \times 10^{-9}$,

  $3{,}155{,}760{,}000_{ten}$, $0.00315576 \times 10^{12}$ or 3.15576×10^{9}

- 과학적 표기법(Scientific notations)

  $0.1_{ten} \times 10^{-8}$, $1.0_{ten} \times 10^{-9}$,

  $0.00315576 \times 10^{12}$, 3.15576×10^{9}

- 정규화된 수(Normalized numbers)

  $1.0_{ten} \times 10^{-9}$, 3.15576×10^{9}

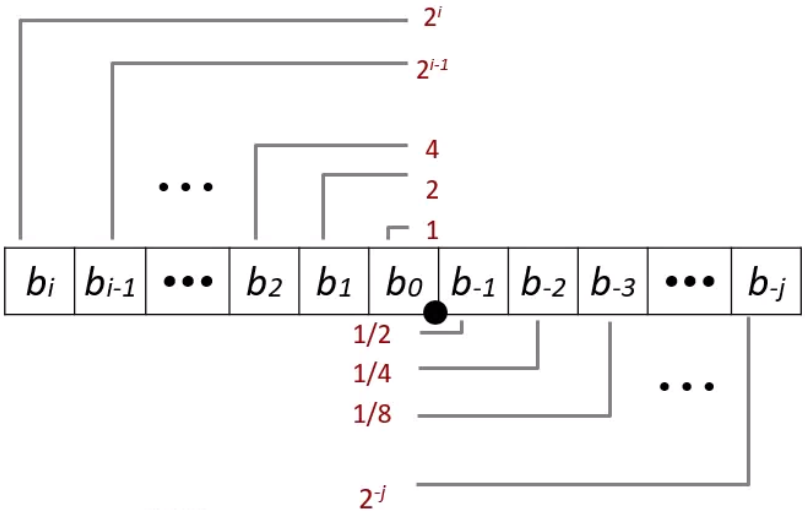
- -2.34×10^{56} ← normalized
- $+0.002 \times 10^{-4}$ ← not normalized
- $+987.02 \times 10^{9}$ ← not normalized

- 부동소수점(floating point)의 이진표현

  $\pm 1.xxxxxxxxx_{two} \times 2^{yyyy}$

실수의 이진수 표현

$$\sum_{k=-j}^{i} b_k \times 2^k$$



- 101.11_2 의 값은?
  - $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
    $= 4 + 0 + 1 + \frac{1}{2} + \frac{1}{4} = 5\frac{3}{4} = 5.75$

실수의 이진수 표현의 한계

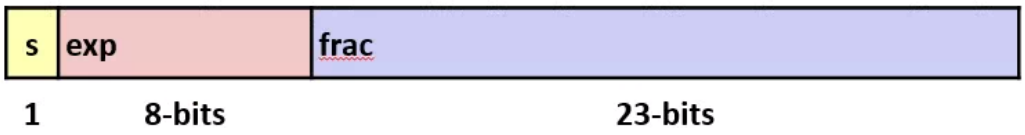
- $x/2^k$ 형식의 수만 정확하게 표현
  - 다른 값들은 근사화할 수 밖에 없음
  - 예
    - 1/3 $0.0101010101[01]\cdots_2$
    - 1/5 $0.001100110011[0011]\cdots_2$
    - 1/10 $0.0001100110011[0011]\cdots_2$
- 근사값의 표기
  - 0.11.....12의 형태의 수는 1.0 에 가깝지만 1보다는 작은 수의 표시 => $1.0 - \varepsilon$

부동소수점수 표준

- IEEE 754 부동소점 표준
- 각각 다른 실수의 표현 방식에 따른 호환성 이슈의 문제를 해결하기 위해 개발
- 현재 널리 사용되는 표준
- 두 가지 실수 표현 방식
  - 32 비트 단일정밀도 (single precision)
  - 64 비트 이중정밀도 (double precision)
- C 언어
  - 단일 및 이중 정밀도에 대해 float, double 자료형 제공

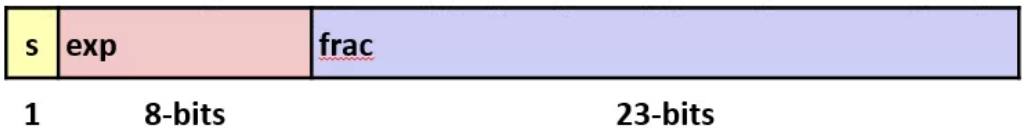
IEEE 부동소수점 수의 표현 – 단일정밀도

- IEEE 754 단일정밀도의 32비트 인코딩
  - 부호(Sign) 1비트, 지수(Exponent) 8비트, 비율(Fraction, 유효자리) 23 비트
    - 비율은 소수부분 또는 유효숫자를 의미



$$V = (-1)^s \times (1+frac) \times 2^{(exp-bias)}$$

IEEE 754 단일정밀도 인코딩 값



- $V = (-1)^s \times (1+frac) \times 2^{(exp-bias)}$
  - s: 부호 비트 (0: 양수, 1: 음수)
  - 정규화된 유효자리 (normalized significand)
    - 유효자리 = 1 + frac
    - 1.0과 2-ε 사이의 값 , $1.0 \le |significand| < 2.0$,
    - 숨겨진 1을 덧붙여서 정규화된 이진수 표현
  - exp: 지수 (exponent)
    - 음수의 표현을 제거하기 위해 바이어스(bias) 표현
    - exp = 실제값 + bias, 단일정밀도 bias = 127

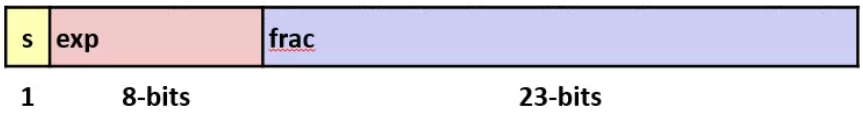
단일정밀도 부동소수점 – 인코딩 예

- 10진수 −0.75의 IEEE 754 단일정밀도 인코딩 예
  - $-0.75_{10} = -0.11_2 = -1.1_{two} \times 2^{-1}$
  - 음수이므로 부호비트 s = 1
  - 유효자리 = 1.1 = 1 + frac
  - 지수 exp = −1 + bias = −1 + 127 = 126

    $(-1)^1 \times (1 + .1000...00) \times 2^{(126-127)}$

    = 1 01111110 10000000000000000000000

$$V = (-1)^s \times (1+frac) \times 2^{(exp-bias)}$$



단일정밀도 부동소수점 – 디코딩 예

- 1 10000001 01000000000000000000000의 디코딩 예

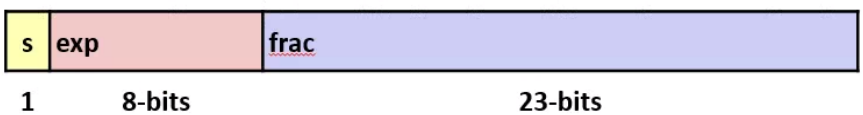
$$V = (-1)^s \times (1+frac) \times 2^{(exp-bias)}$$

$= (-1)^1 \times (1 + 0.25) \times 2^{(129-127)}$

$= -1 \times 1.25 \times 2^2$

$= -1.25 \times 4$

$= -5.0$

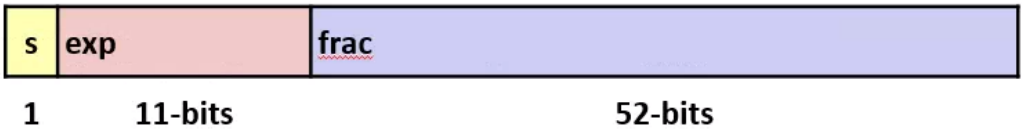


단일 정밀도 부동소수점 최소, 최대값

- 지수 (Exponent) = 실제값 + 바이어스 (127)
  - 지수 값 00000000 (0), 11111111 (255)는 예약된 값
- 가장 작은 실수
  - 지수: 00000001 => 실제 지수값 = 지수 – 바이어스 = 1 – 127 = –126
  - 소수부분: 0000 … 00000 => 유효자리 = 1.0
  - $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$
- 가장 큰 실수
  - 지수: 11111110 => 실제 지수값 = 254 – 127 = +127
  - 소수부분: 1111 … 1111111 => 유효자리 ≈ 2.0
  - $\pm 2.0 \times 2^{+127} \approx \pm 3.4 \times 10^{+38}$

IEEE 754 이중정밀도

- 이중정밀도(double precision) 부동소수점의 64비트 인코딩
  - 부호(Sign) 1 비트, 지수(Exponent) 11 비트,비율(Fraction) 52 비트
  - 바이어스 1023
  - 가장 작은 실수: $\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$
  - 가장 큰 실수: $\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$



$$V = (-1)^s \times (1+frac) \times 2^{(exp-bias)}$$

부동소수점의 IEEE 754 인코딩 요약

이중정밀도		이중정밀도		표현하는 값
지수	소수부분	지수	소수부분	
0	0	0	0	0
0	0 아닌 수	0	0 아닌 수	±비정규화 수 (denormalized number)
1~254	모든 수	1~2046	모든 수	±부동소수점 수
255	0	2047	0	±∞
255	0 아닌 수	2047	0 아닌 수	NaN (Not a Number)

그림 3.13

부동소수점 연산

- 부동소수점 연산 표현
  - $x +_f y = Round(x + y)$
  - $x \times_f y = Round(x \times y)$

- 먼저 정확한 값을 계산
- 계산 결과를 해당 정밀도에 맞춤 (round)
  - 지수의 값이 너무 크면 오버플로우 발생
  - 유효자리 frac에 자릿수에 맞춤(근사화, round)

부동소수점 덧셈: 10진수 예

- 4자리 10진 수 예
  - $9.999 \times 10^1 + 1.610 \times 10^{-1}$

단계 1. 작은 지수를 갖는 수의 지수를 큰 수에 일치 (align)

$9.999 \times 10^1 + 0.016 \times 10^1$

단계 2. 유효자리 더함 (add)

$9.999 \times 10^1 + 0.016 \times 10^1 = 10.015 \times 10^1$

단계 3. 오버플로와 언더플로우 검사하면서 합을 정규화 (normalize)

1.0015×10^2

단계 4. 결과를 유효자리 맞춤,
자리 4자리에 맞춤(필요하면 반올림)

1.002×10^2

부동소수점 덧셈 예: 0.5 + (-0.4375)

[Answer]

$0.5_{ten} = 0.1_{two} \times 2^0 = 1.000_{two} \times 2^{-1}$

$-0.4375_{ten} = -0.0111_{two} \times 2^0 = -1.110_{two} \times 2^{-2}$

(유효자리가 4비트로 가정)

단계 1. 작은 지수를 갖는 수의 지수를 큰 수에 일치

작은 수: $-1.110_{two} \times 2^{-2} = -0.111_{two} \times 2^{-1}$

단계 2. 유효자리 더함

$(1.0_{two}) + (-0.111_{two}) = 0.001_{two}$

단계 3. 오버플로와 언더플로우 검사하면서 합을 정규화

$0.001_{two} \times 2^{-1} = 0.010_{two} \times 2^{-2} = 0.100_{two} \times 2^{-3} = 1.000_{two} \times 2^{-4}$

$127 >= -4 >= -126$ 는 언더플로/오버플로 아님

단계 4. 결과를 자리맞춤

$1.000_{two} \times 2^{-4} = 0.00625$

⇒ 유효자리 4비트가 적합한 비트이므로 자리 맞춤할 필요 없음

부동소수점 곱셈: 10진수 예

- 4자리 10진 수 예
  - $(1.110 \times 10^{10}) \times (9.200 \times 10^{-5})$

단계 1. 지수 덧셈 (add)

$10 + (-5) = 5$

단계 2. 유효자리 곱함 (multiply)

$1.110 \times 9.200 = 10.212 \Rightarrow 10.212 \times 10^{5}$

단계 3. 정규화 (normalize)와 언더플로우/오버플로 검사

1.0212×10^{6}

단계 4. 결과를 유효자리 맞춤, 유효자리 4자리에 맞춤(필요하면 반올림)

1.021×10^{6}

단계 5. 부호 결정

$+1.021 \times 10^{6}$

부동소수점 곱셈 예: 0.5 x (-0.4375)

[Answer]

$0.5_{ten} = 1.000_{two} \times 2^{-1}$

$-0.4375_{ten} = -1.110_{two} \times 2^{-2}$

(유효자리 수는 4비트 가정)

단계 1. 지수 덧셈

$(-1 + 127) + (-2 + 127) - 127 = (-3 + 127)$

단계 2. 유효자리 곱함

$1.000_{two} \times 1.110_{two} = 1.110000_{two} = 1.110_{two}$

단계 3. 정규화와 언더플로우/오버플로 검사

이미 정규화되었고 언더플로우/오버플로 없음

단계 4. 곱셈결과 자리맞춤

변화 없음. $1.110_{two} \times 2^{-3}$

단계 5. 부호 결정

Product = $-1.110_{two} \times 2^{-3} = -0.21875_{ten}$

과제 2-5: 부동소수점 오차의 영향

- 연습문제 2.46 (p.108)
  - 미국 패트리엇트 미사일 오차
    - 1991년 걸프 전쟁에서 스커드 미사일 요격 실패로 28명 사망
  - 내부 클럭
    - 0.1(1/10)초 마다 증가하는 카운터
  - 시간 계산
    - 0.1초 = 카운터 값 * 1/10 <- 시간 계산을 위해 1/10 상수 곱셈이 필요
  - $1/10 = 0.000110011[0011].....{}_2$ (실제값)
    - 23비트로 표현 (근사값)
    - x = 1/10 = 0.00011001100110011001100
  - A. 오차는 0.1 – x (실제값 – 근사값)
  - B. 0.1초 마다 발생하는 오차 값
  - C. 100시간 동작 시 오차
  - D. 초속 2000 미터로 날아오는 스커드 미사일의 위치 계산 오차

요 약

요약

- 부동소수점 표시는 숫자를 $x * 2^y$로 근사화
- IEEE 표준 754는 단일 (32비트) 및 이중 (64비트) 정밀도를 표현
- 부동소수점 연산은 계산 결과를 해당 정밀도에 맞춤 (round)

과 제

과제 2-5: 부동소수점 오차의 영향

❑ 연습문제 2.46 (p.108)

- 미국 패트리엇트 미사일 오차
  - 1991년 걸프 전쟁에서 스커드 미사일 요격 실패로 28명 사망
- 내부 클럭
  - 0.1(1/10)초 마다 증가하는 카운터
- 시간 계산
  - 0.1초 = 카운터 값 * 1/10 <- 시간 계산을 위해 1/10 상수 곱셈이 필요
- $1/10 = 0.000110011[0011].....{}_2$ (실제값)
  - 23비트로 표현 (근사값)
  - x = 1/10 = 0.00011001100110011001100
- A. 오차는 0.1 – x (실제값 – 근사값)
- B. 0.1초 마다 발생하는 오차 값
- C. 100시간 동작 시 오차
- D. 초속 2000 미터로 날아오는 스커드 미사일의 위치 계산 오차