

8강_BFPN Representation/9강_BFPN Arithmetic

Single Precision 단정밀도 : float number / Double Precision 배정 밀도 : double number

Binary Floating Point # (BFPN) Representation: Single Precision BFPN


Single Precision
(단 정밀도) BFPN : 32bits

31 30 23 22 0

Sign	Exponent(8)	Mantissa(23)
------	-------------	--------------

기본형: $\pm 0.1M \times 2^E$
만일 1.1010×2^4 이라면 0.1101×2^5 로 Normalization(정규화)
• $S=0$
• $E=0000_0101$ (2's Complement 표현)
• $M=101_0000_0000_0000_0000$ (Unsigned 표현)
• 결국 S와 M을 합쳐서 Signed Magnitude로 표현

Handwritten notes: 0.11×2^{10} , 9×10 , 101



floating point number 부동소수점 (소수점 이동 가능)

sign / exponent(지수)(지수10) / mantissa(가수) (9)

9×10^{10}

Binary Floating Point # (BFPN) Representation: Double Precision BFPN

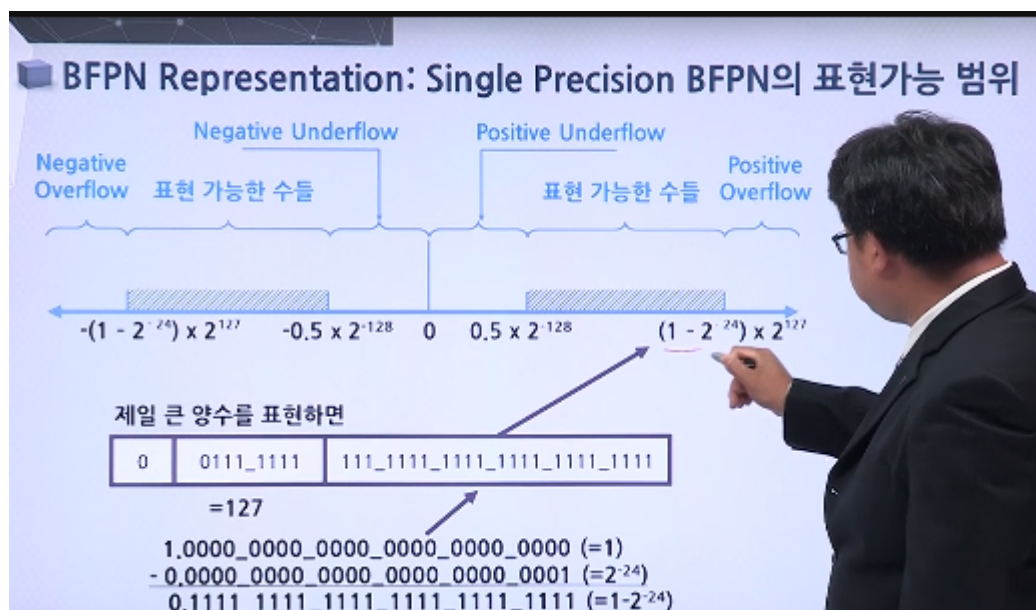
- Sign = 1 (음수), 0 (양수)
- Mantissa의 범위: $0.5 \leq \text{Mantissa} \leq 1 \rightarrow$ 정밀도 결정
- Exponent의 범위: $-2^7 < \text{Exponent} < 2^7 - 1 \rightarrow$ 표현 가능한 수의 범위 결정
- Mantissa와 Exponent간 길이조절 필요

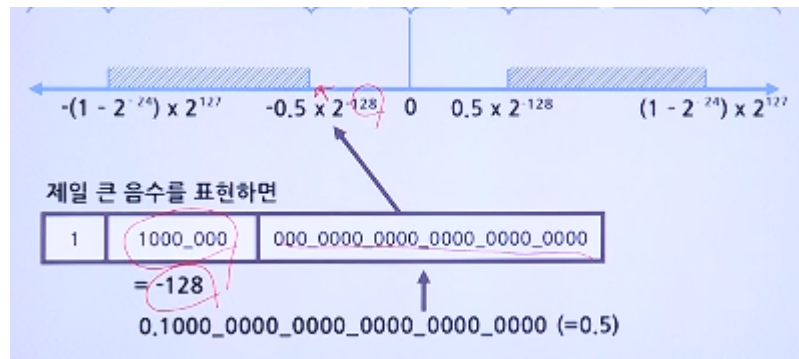
Double Precision (배 정밀도) BFPN: 64bits

63	62	52	51	0
Sign	Exponent(11)		Mantissa(52)	

표현 가능한 수의 범위를 결정하는데 exponent

Mantissa의 Bit를 많이 할당하면 Exponent의 Bit가 작아져야되고요. 그 반대의 경우가 생길수도 있겠죠.





BFPN Representation: Single Precision BFPN with Biased Exponent

Why Biased Exponent?

➤ E의 값이 아주 작은 음수라면 전체 숫자는 거의 0에 가까워 짐

- 0에 대한 표현에서 모든 Bit들이 0이 되게 하여, Zero-Test(ZT)가 정수에서와 같은 방법으로 가능하게 하기 위함
- If M = 000_0000_0000_0000_0000_0000 then BFPN=0
 ∴ 일반적인 정수와 동일한 방법으로 ZT 가능
- If E = 1000_0000(BFPN에서 가장 작은 음수) then BFPN=0
 ∴ 일반적인 정수와 동일한 방법으로 ZT 불가능
- If E = 0000_0000(BFPN with Biased 128에서 가장 작은 음수) then BFPN=0
 ∴ 일반적인 정수와 동일한 방법으로 ZT 가능



➤ E의 값이 아주 작은 음수라면 전체 숫자는 거의 0에 가까워 짐

- 0에 대한 표현에서 모든 Bit들이 0이 되게 하여, Zero-Test(ZT)가 정수에서와 같은 방법으로 가능하게 하기 위함
- If M = 000_0000_0000_0000_0000_0000 then BFPN=0
 ∴ 일반적인 정수와 동일한 방법으로 ZT 가능
- If E = 1000_0000(BFPN에서 가장 작은 음수) then BFPN=0
 ∴ 일반적인 정수와 동일한 방법으로 ZT 불가능
- If E = 0000_0000(BFPN with Biased 128에서 가장 작은 음수) then BFPN=0
 ∴ 일반적인 정수와 동일한 방법으로 ZT 가능

Exponent 패턴	원래값	실제 Exponent 값	Bias=127	Bias=128
11111111	255	+128	+127	+127
11111110	254	+127	+126	+126
...
10000001	129	+2	+1	+1
10000000	128	+1	0	0
01111111	127	0	-1	-1
01111110	126	-1	-2	-2
...
00000001	1	-126	-127	-127
00000000	0	-127	-128	-128

BFPN Representation: IEEE 754 Standard - Format, Example, and Exceptions

Format

- Single Precision : $N = (-1)^s \times 2^{E-127} \times (1.M)$ → 1은 Hidden Bits
- Double Precision : $N = (-1)^s \times 2^{E-1023} \times (1.M)$
- Signed Magnitude Representation(Sign + Mantissa),
Biased-127/1023 Exponent

BFPN Representation: IEEE 754 Standard - Format, Example, and Exceptions

Exceptions

	E	M	Representation
NaN	$E = 255/2047$	$M \neq 0$	$N = \text{NaN}$ (0 나누기)
Overflow	$E = 255/2047$	$M = 0$	$N = (-1)^s \times \infty \times (1.0)$
일반식	$0 < E < 255/2047$		$N = (-1)^s \times 2^{E-127/1023} \times (1.M)$
Underflow	$E = 0$	$M \neq 0$	$N = (-1)^s \times 0 \times (1.M)$
Zero	$E = 0$	$M = 0$	$N = (-1)^s \times 0$

9강

BFPN Arithmetic Operation

덧셈/뺄셈

- Exponent들이 일치되도록 조정

$$\begin{array}{rcl}
 0.110100 \times 2^6 & \rightarrow & 0.001101 \times 2^5 \\
 + 0.111100 \times 2^5 & + & 0.111100 \times 2^5 \\
 \hline
 & & 1.001001 \times 2^5 \rightarrow 0.1001001 \times 2^6
 \end{array}$$
- Mantissa들을 연산
- Normalization을 통해 최종 결과 획득

BFPN Arithmetic Operation

곱셈/나눗셈

주어진 예

$$: (0.1011 \times 2^3) \times (0.1001 \times 2^5) = 0.1100011 \times 2^7$$

Mantissa Multiplication/Division

$$: 1011 \times 1001 = 01100011$$

Exponent Addition/Subtraction

$$: 3 + 5 = 8$$

Normalization

$$: 0.01100011 \times 2^8 = 0.1100011 \times 2^7$$

산술연산에서 발생 가능한 문제들

- Exponent Overflow : $+\infty$, $-\infty$ 로 Set
- Exponent Underflow : 0으로 Set
- Mantissa Overflow : Normalization
- Mantissa Underflow : Rounding(내림)