# 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 $1.1010 \times 2^{4}$이라면 $0.1101 \times 2^{5}$로 Normalization(정규화)
  - S=0
  - E=0000_0101(2's Complement 표현)
  - M=101_0000_0000_0000_0000_0000(Unsigned 표현)
- 결국 S와 M을 합쳐서 Signed Magnitude로 표현

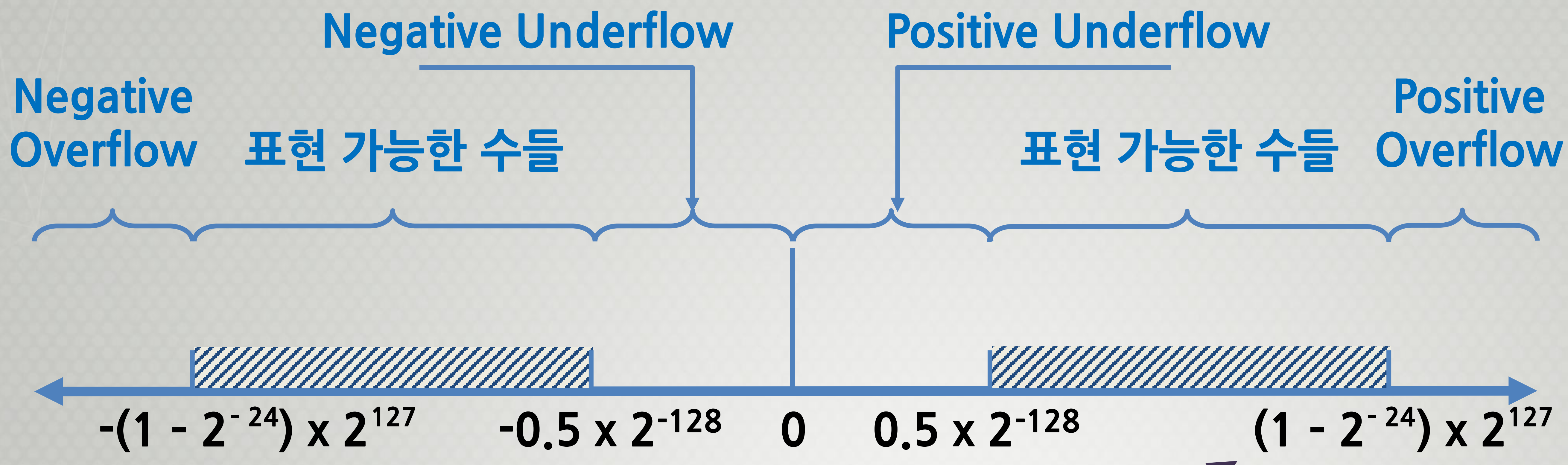
# Binary Floating Point #(BFPN) Representation: **Double Precision BFPN**

- Sign = 1 (음수), 0 (양수)
- Mantissa의 범위 : $0.5 \le Mantissa \le 1$ → 정밀도 결정
- Exponent 의 범위 : $-2^7 < Exponent < 2^7-1$ → 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) |

# BFPN Representation: Single Precision BFPN의 표현가능 범위



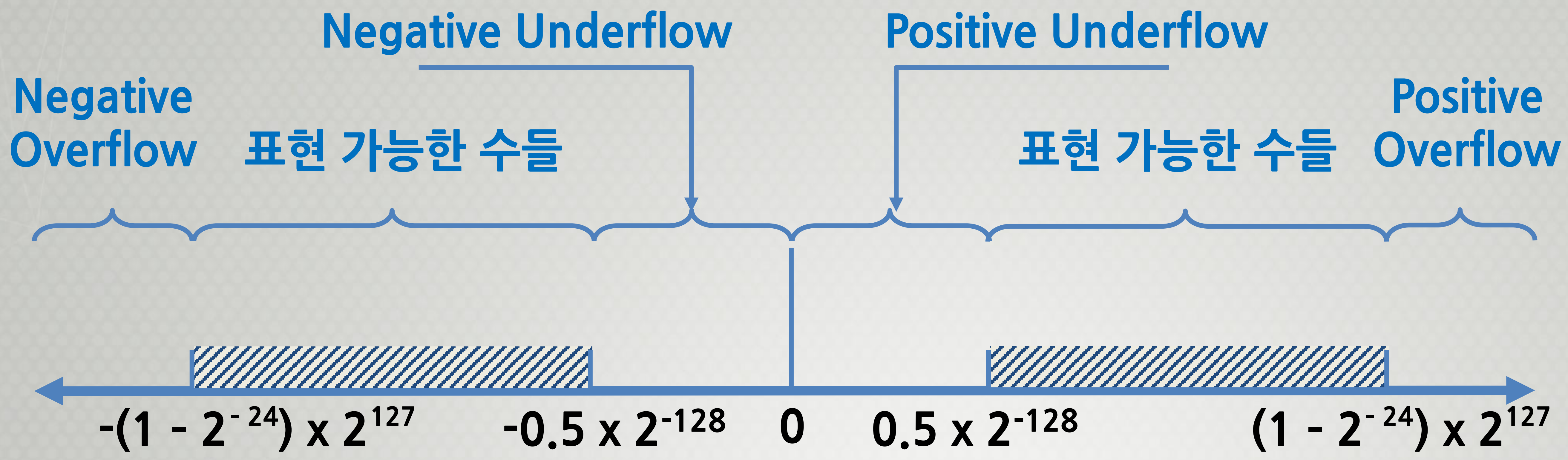

제일 큰 양수를 표현하면

| | | |
|---|---|---|
| 0 | 0111_1111 | 111_1111_1111_1111_1111_1111 |

=127

$$
\begin{array}{rl}
 & 1.0000\_0000\_0000\_0000\_0000\_0000\ (=1) \\
- & \underline{0.0000\_0000\_0000\_0000\_0000\_0001\ (=2^{-24})} \\
 & 0.1111\_1111\_1111\_1111\_1111\_1111\ (=1-2^{-24})
\end{array}
$$

# BFPN Representation: Single Precision BFPN의 표현가능 범위




제일 큰 음수를 표현하면

| | | |
|---|---|---|
| 1 | 1000_000 | 000_0000_0000_0000_0000_0000 |

= -128

0.1000_0000_0000_0000_0000_0000 (=0.5)

# BFPN Representation: Single Precision BFPN with Biased Exponent

## Bias=128일 때, N=-13.625에 대한 BFPN 표현

- $13.625_{10} = 1101.101_2 = 0.1101101 \times 2^4$
  - S = 1
  - M = 10110100000000000000000(소수점 우측의 첫 번째 1 제외)
  - E = 00000100 + 10000000 = 10000100 (Bias 128을 더함)

| S(1) | E(8) | M(23) |
|---|---|---|
| 1 | 1000_0100 | 101_1010_0000_0000_0000_0000 |

# 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

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

## Example

- $N = -13.625_{10} = -1101.101_2 = -1.101101 \times 2^3$

| | | |
|---|---|---|
| 1 | 10000010 | 10110100000000000000000 |

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

## Exceptions

| | E | M | Representation |
|---|---|---|---|
| NaN | E = 255/2047 | M ≠ 0 | N = 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 ≠ 0 | N = $(-1)^s \times 0 \times$(1.M) |
| Zero | E = 0 | M = 0 | N = $(-1)^s \times 0$ |