FLOATING POINT

<---->

FIXED POINT

CONVERSION
Floating Point → Fixed Point Conversion

- If the exp is unsigned, the shifter shifts only to the left
- If the exp is signed, the shifter must shift to the left and right
- Example:

  \[ 01011. \times 2^2 \]

  \[ 01011. \ll 2 \]

  \[ 000101100. \]
Fixed Point $\rightarrow$ Floating Point Conversion

- Leading 0s/1s detector finds the optimum place to begin selecting bits for the mantissa.
- Common pitfall: If the mantissa is signed, its sign bit(s) must be maintained!

```
fixed point

leading 0s/1s detector

± offset

shifter

± offset

exp

mantissa
```
Fixed Point → Floating Point Conversion

- Fixed-to-float conversion example (*positive* input)
  - Input: 8-bit 2’s complement (signed) integer
  - Output: 4-bit 2’s complement (signed) mantissa

  a) integer mantissa

  \[
  \begin{array}{c|cccccc}
  S & 0 & 0 & 0 & 0 & 1 & 1 \\
  \hline
  & 0 & 1 & 1 & 0 & . & \rightarrow \\
  & 12 & = & 6 & \times & 2^1 \\
  \end{array}
  \]

  b) fractional “0.4 format” mantissa

  \[
  \begin{array}{c|cccccc}
  S & 0 & 0 & 0 & 0 & 1 & 1 \\
  \hline
  & .0 & 1 & 1 & 0 & \rightarrow \\
  & 12 & = & 0.375 & \times & 2^5 \\
  \end{array}
  \]
Fixed Point → Floating Point Conversion

• Fixed-to-float conversion example (negative input)
  – Input: 8-bit 2’s complement (signed) integer
  Output: 4-bit 2’s complement (signed) mantissa

  a) integer mantissa

  \[
  \begin{align*}
  \text{s} & \quad 4 \\
  1 & 1 & 0 & 1 & 0 & 0 & 0 & 1. \quad \rightarrow \quad 1 & 0 & 1 & 0. \times 2^{(011)} & \quad \% 2^3 \\
  -47 & \quad = & \quad -6 & \times 2^3
  \end{align*}
  \]

  b) fractional “2.2 format” mantissa

  \[
  \begin{align*}
  \text{s} & \quad 4 \\
  1 & 1 & 0 & 1 & 0 & 0 & 0 & 1. \quad \rightarrow \quad 1 & 0.1 & 0 \times 2^{(101)} & \quad \% 2^5 \\
  -47 & \quad = & \quad -1.5 & \times 2^5
  \end{align*}
  \]
Fixed Point $\rightarrow$ Floating Point Conversion

Special Cases

- **Example 1:** converting a fixed-point zero
  
  \[
  00000000
  \]
  
  - Clearly, the selection of mantissa bits does not matter $\rightarrow$ it will be all zeros
  
  - But then what should the exponent be?
    
    - In absolute terms it does not matter
    
    - Choose whatever makes the hardware **more regular** and simpler

- **Example 2:** converting a string of 1’s to FloatPt with a 4-bit mantissa
  
  \[
  11111111
  \]
  
  - We have at least two main approaches to selecting the mantissa bits which in general do not affect accuracy
    
    - 1) Choose mantissa after removing the max number of redundant sign bits
      
      \[
      1000. \times 2^{(-3)} = -8 \times (1/8) = -1
      \]
    
    - 2) Choose mantissa to preserve as many bits as possible while removing the max number of redundant sign bits
      
      \[
      1111. \times 2^0 = -1 \times 1 = -1
      \]
    
    - Choose whichever method makes the hardware **more regular** and simpler