32-bit Floating-Point Multiplier Workflow

Input Breakdown:

- Inputs are 32-bit floating-point numbers: `A[31:0]` and `B[31:0]`.
- Split the inputs into components:
  - **Sign**: `A[31]` and `B[31]`.
  - **Exponent**: `A[30:23]` and `B[30:23]` (8-bit, biased by 127).
  - **Mantissa**: `A[22:0]` and `B[22:0]` (23 bits, add an implied leading `1`).

Step 1: Sign Calculation:

- Compute the result's sign using XOR: Signresult=A[31]⊕B[31]\text{Sign}_{\text{result}} = A[31] \oplus B[31]Signresult=A[31]⊕B[31]

Step 2: Exponent Addition:

- Add the exponents: Exponentsum=A[30:23]+B[30:23]\text{Exponent}_{\text{sum}} = A[30:23] + B[30:23]Exponentsum=A[30:23]+B[30:23]
- Subtract the bias (127) to get the adjusted exponent: Exponentresult=Exponentsum−127\text{Exponent}_{\text{result}} = \text{Exponent}_{\text{sum}} - 127Exponentresult=Exponentsum−127

Step 3: Mantissa Multiplication:

- Add the implied leading `1` to each mantissa: MantissaA={1′b1,A[22:0]}, MantissaB={1′b1,B[22:0]}\text{Mantissa}_A = \{1'b1, A[22:0]\}, \, \text{Mantissa}_B = \{1'b1, B[22:0]\}MantissaA={1′b1,A[22:0]},MantissaB={1′b1,B[22:0]}
- Multiply the 24-bit mantissas to get a 48-bit intermediate product: Mantissaproduct=MantissaA×MantissaB\text{Mantissa}_{\text{product}} = \text{Mantissa}_A \times \text{Mantissa}_BMantissaproduct=MantissaA×MantissaB

Step 4: Normalization:

- If the most significant bit of the product is `0`, shift left and decrement the exponent by 1.
- Keep the top 23 bits as the normalized mantissa.

Step 5: Rounding:

- Apply truncation (round toward zero) by taking the top 23 bits of the normalized mantissa.

Step 6: Output Assembly:

- Combine the computed components to form the 32-bit result: Result={Signresult,Exponentresult,Mantissaresult[22:0]}\text{Result} = \{\text{Sign}_{\text{result}}, \text{Exponent}_{\text{result}}, \text{Mantissa}_{\text{result}}[22:0]\}Result={Signresult,Exponentresult,Mantissaresult[22:0]}