

FLOATING POINT ADD/SUB IP

Introduction:

Floating Point numbers are represented in IEEE 754 format in most of the DSP Processors. Floating point arithmetic is useful in applications where a large dynamic range is required or in rapid prototyping applications where the required number range has not been thoroughly investigated.

The Floating Point Adder/Subtractor IP helps designers to perform floating point addition / subtraction on FPGA represented in IEEE 754 format.

Functional Description:

A Floating point adder/ subtracter is more complicated than a floating point multiplier. Generally the addition/subtraction operation is performed in three steps namely alignment, addition and normalization. The major bottleneck is the normalization part.

In the alignment stage the smaller of the inputs is adjusted to match with the exponent of the larger mantissa. Then the inputs are added / subtracted in the addition stage and the result is adjusted to IEEE 754 standard in the normalization stage.

The present IP has additionally a comparator stage at the beginning, which swaps the numbers to avoid handling negative numbers at the later stages.

Features:

- Available for wide range of FPGA families
- Supports Floating Point addition/subtraction
- Compliance with IEEE 754 standard
- Optimized for Speed and Latency
- Fully Synchronous design with single clock
- Compatible, Flexible and integrable with other modules

Overview of Floating Point Numbers:

For representing real numbers most of the DSP Processors use Floating Point Numbers. The speed of floating point operations is an important measure of performance of these DSP processors.

Floating point representation:

The floating point numbers are represented using three fields sign, exponent and significant as shown in fig 1.

In general, binary floating point numbers are stored in a sign-magnitude form where the most significant bit is the sign bit, exponent is the biased exponent and 'fraction' is the significand without the most significant bit.

The exponent is biased by $(2^{e-1})-1$, where e is the number of bits used for exponent field. The exponents are biased so that there will be no negative exponents and the comparison of numbers will be easier.

So the value of a floating point number can be represented as

$$V = s \times 2^E \times m$$

where $s = +1$ (for positive numbers) ,
or sign bit = 1

$s = -1$ (for negative numbers) ,
or sign bit = 0

$E = \text{exponent} - \text{exponent bias}$

$m = 1.\text{fraction}$

or

$$V = (-1)^{\text{sign}} \times 2^{\text{exponent} - \text{exponent bias}} \times 1.\text{fraction}$$

IEEE 754 Format:

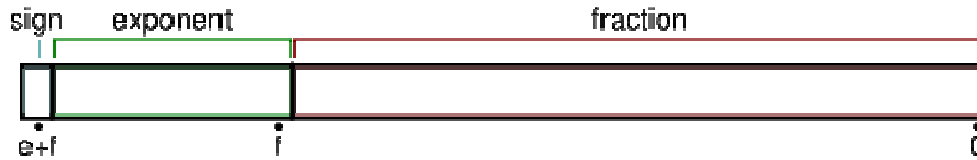


Figure 1: Floating Point Representation in IEEE 754 format

Implementation:

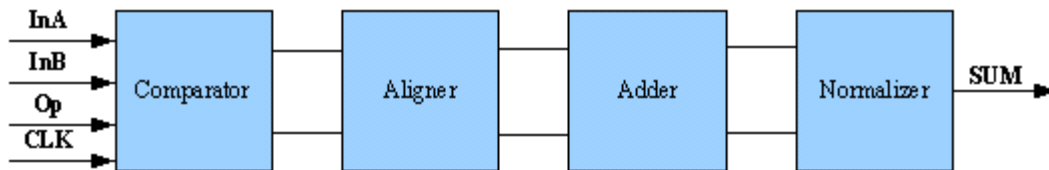


Figure2: Implementation Blocks of FPU

Signal Table:

Signal	Direction	Data Width	Description
InA	IN	32	This is the port through which the first input is fed through.
InB	IN	32	This is the port through which the second input is fed through.
CLK	IN	1	This is the input port through which the entire system is synchronous with
Op	IN	1	The input to this port decides the mode of arithmetic operation. 0 for addition and 1 for subtraction
SUM	OUT	32	This is the port through which the output is fed out.

Table 1: Signal Definition Table

Performance:

Device	Slice Count	Frequency (MHz)
Virtex-4 (XC4VLX15-SF363)	331	201
Virtex-5 (XC5VLX30-FF324)	361	294

Table 2: FPU add/sub IP Performance table.

Verification:

The Floating point adder/subtractor IP has been verified with the following approaches:

- Exhaustive Functional/Timing simulation
- Results are compared with the C-code generated results and Behavioral model results.
- Emulated on Xilinx FPGA

Deliverables:

- Verilog Behavioral, RTL source code
- Test Benches
- C- code for generation of test vectors
- Detailed user documentation