

Key Design Features

- Synthesizable, technology independent IP Core for FPGA, ASIC and SoC
- Supplied as human readable VHDL (or Verilog) source code
- 32-bit floating-point arithmetic
- IEEE 754 compliant¹
- High-speed fully pipelined architecture
- Only 4 clock-cycles of latency

Applications

- Floating-point pipelines and arithmetic units
- Floating-point processors

Pin-out Description

Pin name	I/O	Description	Active state
clk	in	Synchronous clock	rising edge
en	in	Clock enable	high
v1 [31:0]	in	Input operand 1 in IEEE 754 format	data
v2 [31:0]	in	Input operand 2 in IEEE 754 format	data
vout [31:0]	out	Output result in IEEE 754 format	data

Functional Specification

Operand v1	Operand v2	Result
Standard IEEE	Standard IEEE	$v1 * v2$ If $ v1 * v2 > \text{MaxFloat}$ then result is: $[\text{sign}(v1) \text{ xor } \text{sign}(v2)] \text{ Inf}$ If $ v1 * v2 \leq \text{MinFloat}$ then result is: $[\text{sign}(v1) \text{ xor } \text{sign}(v2)] 0$
NaN	Anything	NaN
Anything	NaN	NaN
+/- Inf	+/- 0	NaN
+/- 0	+/- Inf	NaN
+/- Inf	Standard IEEE	$[\text{sign}(v1) \text{ xor } \text{sign}(v2)] \text{ Inf}$
Standard IEEE	+/- Inf	$[\text{sign}(v1) \text{ xor } \text{sign}(v2)] \text{ Inf}$
+/- 0	Standard IEEE	$[\text{sign}(v1) \text{ xor } \text{sign}(v2)] 0$
Standard IEEE	+/- 0	$[\text{sign}(v1) \text{ xor } \text{sign}(v2)] 0$

1 Some minor features diverge from the IEEE 754 specification

Block Diagram

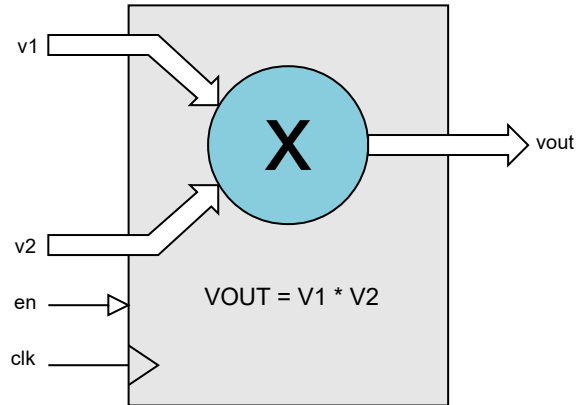


Figure 1: 32-bit Floating-point Multiplier

General Description

The IEEE_MULT IP Core (Figure 1) is a high-speed fully pipelined 32-bit floating-point multiplier based on the IEEE 754 standard. The arrangement of the 32-bit floating-point number is summarized below:

MSB		LSB
Sign (1-bit)	Exponent (8-bits)	Mantissa (23-bits)

All input and output values comply with the IEEE 754 specification. The real number representation is calculated according to the formula:

$$\text{Value} = -1(S) * 2^{(E-127)} * 1.M$$

There are two exceptions to the IEEE 754 specification. The first being that denormalized (subnormal) numbers are treated as zero throughout the implementation, and the second being that symmetric arithmetic rounding is employed (round half-up).

Other points to note are that NaN is always generated as the value 0xFFC00000. The maximum floating-point value that may be represented is 0x7F7FFFFF or 0xFF7FFFFF (+/- MaxFloat). Likewise, the minimum floating-point value that may be represented is 0x00800000 or 0x80800000 (+/- MinFloat). This means that a real number lies in the range:

$$2^{-126} \leq \text{Value} \leq 2^{127}(2-2^{-23})$$

All values are sampled on the rising clock-edge of *clk* when *en* is high. The function has a 4 clock-cycle latency².

2 The design may be optimized for higher-speed or lower latency on request. Please contact Zipcores for more information.

Functional Timing

Figure 2 demonstrates the multiplication: $0x40010000 * 0x40407000 = 0x40C0880E$ (or $2.0010 * 3.0068 = 6.0166$ in real numbers). The result has a latency of 4 clock cycles.

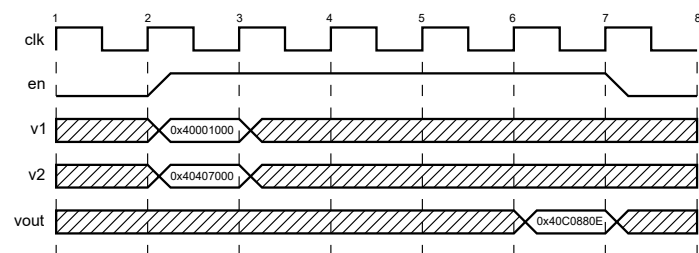


Figure 2: Multiplication of two floating-point numbers

Source File Description

All source files are provided as text files coded in VHDL. The following table gives a brief description of each file.

Source file	Description
ieee_mult.vhd	Top-level component
ieee_mult_bench.vhd	Top-level test bench

Functional Testing

An example VHDL testbench is provided for use in a suitable VHDL simulator. The compilation order of the source code is as follows:

1. ieee_mult.vhd
2. ieee_mult_bench.vhd

The simulation must be run for at least 2 ms during which time an input stimulus of randomized floating-point numbers will generated at the multiplier input.

The simulation generates two text files called: *ieee_mult_in.txt* and *ieee_mult_out.txt*. These files respectively capture the input and output floating-point numbers during the course of the test.

Synthesis and Implementation

The source file 'ieee_mult.vhd' is the only file required for synthesis. There are no sub-modules in the design.

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx® 7-series FPGAs. Synthesis results for other FPGAs and technologies can be provided on request.

Resource usage is specified after Place and Route.

XILINX® 7-SERIES FPGAS

Resource type	Artix-7	Kintex-7	Virtex-7
Slice Register	118	118	110
Slice LUTs	143	143	139
Block RAM	0	0	0
DSP48	4	4	4
Occupied Slices	60	60	54
Clock freq. (approx)	250 MHz	300 MHz	350 MHz

Revision History

Revision	Change description	Date
1.0	Initial revision	30/04/2008
1.1	Updated synthesis results	19/08/2009
1.2	Updated functional specification. Updated synthesis results in line with minor code changes	16/09/2011
1.3	Cosmetic changes to the source code Updated results for Xilinx® 7-series	02/07/2018

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Zipcores:](#)

[SKU12](#)