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 compliant[1]
- High-speed fully pipelined architecture
- Variable latency from 2 to 49 clock cycles

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
reg_stages	in	Generic parameter fixes latency at compile time	N/A

Functional Specification

Operand v1	Operand v2	Result
Standard IEEE	Standard IEEE	v1 / v2 If |v1 / v2| > MaxFloat then result is: [sign(v1) xor sign(v2)] Inf If |v1 / v2| ≤ MinFloat then result is: [sign(v1) xor sign(v2)] 0
NaN	Anything	NaN
Anything	NaN	NaN
+/- Inf	+/- Inf	NaN
+/- 0	+/- 0	NaN
+/- Inf	Standard IEEE	[sign(v1) xor sign (v2)] Inf
Standard IEEE	+/- Inf	[sign(v1) xor sign (v2)] 0
+/- 0	Standard IEEE	[sign(v1) xor sign (v2)] 0
Standard IEEE	+/- 0	[sign(v1) xor sign (v2)] Inf

1 Some minor features diverge from the IEEE 754 specification

Block Diagram

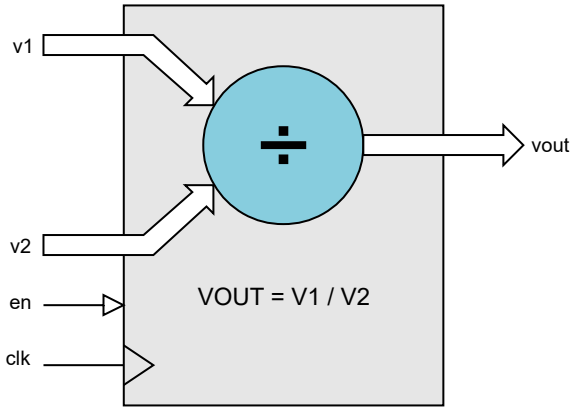
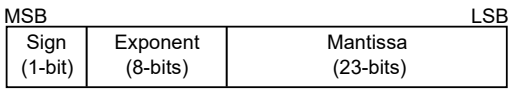


Figure 1: 32-bit Floating-point Divider

General Description

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



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

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

The divider is fully compliant with the IEEE 754 standard with the exception that denormalized (subnormal) numbers are treated as zero throughout the implementation. The maximum floating-point value that may be represented in hex 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:

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

Other points to note are that a NaN is always generated as the value 0xFFC00000. By default, the divider uses round towards zero, although other rounding methods are available on request.

All values are sampled on the rising clock-edge of *clk* when *en* is high. The latency of the divider pipeline is generic and may be fixed during synthesis.

Integer values of between 2 and 49 clock cycles are possible, with the overall latency given by:

$$Latency = (48 / reg\_stages) + 1$$

Functional Timing

Figure 2 demonstrates the division: 0x3FA00000 / 0x40333333 (or 1.25 / 2.8 = 0.44643 in real numbers). In this particular case, the generic parameter *reg_stages* has been set to 24 giving a result with a latency of 3 clock cycles (48/24+1).

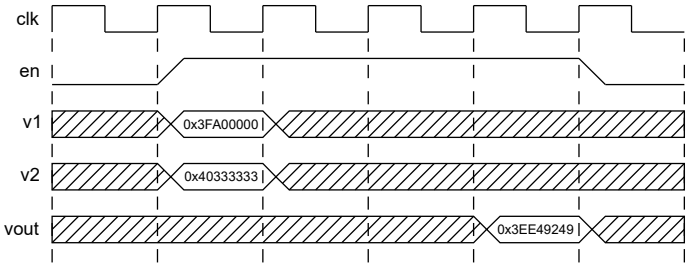


Figure 2: Division of two floating-point numbers with the pipeline latency fixed at 3 clock cycles

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_div_shiftsub.vhd	Pipelined divider shift-subtract module
ieee_div_pipe.vhd	Pipelined divider module
ieee_div.vhd	Top-level component
ieee_div_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_div_shiftsub.vhd
2. ieee_div_pipe.vhd
3. ieee_div.vhd
4. ieee_div_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 divider input.

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

Synthesis and Implementation

The source files required for synthesis and the design hierarchy is shown below:

- ieee_div.vhd
  - ieee_div_pipe.vhd
  - ieee_div_shiftsub.vhd

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.

By adding more pipeline stages (reducing the value of the *reg_stage* generic) will result in faster implementations. Conversely, reducing the number of pipeline stages will generally result in a smaller but slower design. Generally, using around 13-17 pipeline stages will give the optimal results.

Trial synthesis results are shown with a setting of *reg_stages* = 1 (maximum pipelining). Resource usage is specified after Place and Route.

XILINX® 7-SERIES FPGAS

Resource type	Artix-7	Kintex-7	Virtex-7
Slice Register	2352	2352	2351
Slice LUTs	1489	1489	1442
Block RAM	0	0	0
DSP48	0	0	0
Occupied Slices	451	451	449
Clock freq. (approx)	250 MHz	300 MHz	350 MHz

Revision History

Revision	Change description	Date
1.0	Initial revision	30/04/2008
1.1	Added *reg_stages* generic to allow flexible pipeline depths. Updated synthesis results.	16/09/2011
1.2	Cosmetic changes to the source code Updated results for Xilinx® 7-series	02/07/2018