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 24 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 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	***Result***
Standard IEEE	√ v1 If |v1| > MaxFloat then result is Inf If |v1| ≤ MinFloat then result is 0 (Inputs are assumed unsigned)
NaN	NaN
Inf	Inf
0	0

1 Some minor features diverge from the IEEE 754 specification

Block Diagram

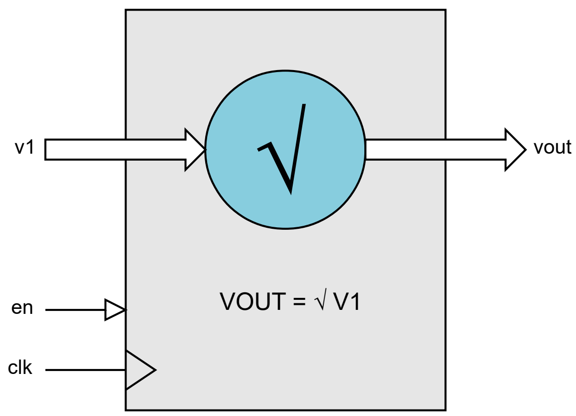

Figure 1: 32-bit Floating-point Square-root

General Description

The IEEE_SQRT IP Core (Figure 1) is a high-speed fully pipelined 32-bit bit floating-point square-root function 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:

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

The square-root is fully compliant with the IEEE 754 specification with the exception that denormalized (subnormal) numbers are treated as zero throughout the implementation. In addition, all input operands are assumed to be unsigned. 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:

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

Other points to note are that a NaN is always generated as the value 0xFFC00000. By default, the square-root result 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 square-root pipeline is generic and may be fixed during synthesis. Integer values of between 2 and 24 clock cycles are possible, The overall latency given by a round-up of the following calculation:

$$Latency = (23 / reg\_stages) + 1$$

Functional Timing

Figure 2 demonstrates the square-root of 0x3FA00000 (or √1.25 = 1.118034 in real numbers). In this particular case, the generic parameter *reg_stages* has been set to 12 giving a result with a latency of 3 clock cycles (23/12+1).

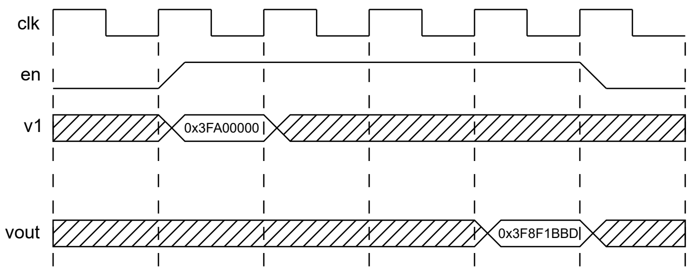

Figure 2: Square-root of a floating-point number with the pipeline latency fixed at 3 clock cycles

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_sqrt_subquare.vhd	Pipelined sqaure-root subtract-square module
ieee_sqrt_pipe.vhd	Pipelined square-root module
ieee_sqrt.vhd	Top-level component
ieee_sqrt_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_sqrt_subsquare.vhd
2. ieee_sqrt_pipe.vhd
3. ieee_sqrt.vhd
4. ieee_sqrt_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 module input.

The simulation generates two text files called: *ieee_sqrt_in.txt* and *ieee_sqrt_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_sqrt.vhd
  - ieee_sqrt_pipe.vhd
  - ieee_sqrt_subsquare.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.

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	861	861	861
Slice LUTs	1148	1146	1147
Block RAM	0	0	0
DSP48	44	44	44
Occupied Slices	460	435	440
Clock freq. (approx)	150 MHz	200 MHz	250 MHz

Revision History

Revision	***Change description***	***Date***
1.0	Initial revision	18/10/2010
1.1	Added *reg_stages* generic to allow flexible pipeline depths. Updated synthesis results.	25/11/2011
1.2	Cosmetic changes to the source code Updated results for Xilinx® 7-series	02/07/2018