

## Key Design Features

- Synthesizable, technology independent VHDL Core
- Function  $f(x) = \sqrt{x}$
- Input and output values as unsigned integers
- Configurable data width
- Unsigned n-bit input generates a result with n/2 integer bits and n/2 fraction bits
- High-speed fully pipelined architecture with pipeline optimization mode for area/speed trade off
- Modular design

## Applications

- Basic building block in digital processing functions
- Square root of integers and fixed-point numbers<sup>1</sup>
- Magnitude of a complex signals in digital communications systems

## Pin-out Description

Pin name	I/O	Description	Active state
clk	in	Synchronous clock	rising edge
en	in	Clock enable	high
a_in [dw-1:0]	in	Unsigned input number	data
sqrt [dw -1:0]	out	Unsigned output square root	data

## Generic Parameters

Generic name	Description	Type	Valid range
dw	Input data width	integer	$\geq 2$ (dw = even)
opt_mode	Pipeline optimization mode	integer	0 = Max no. of pipeline stages 1 = Half no. of pipeline stages

<sup>1</sup> For fixed-point numbers then inputs must be pre-scaled by a square of 2 and the result wire-shifted. E.g. the computation  $\sqrt{0.2}$  could be done as  $\sqrt{51}$  with a shift right of 4

## Block Diagram

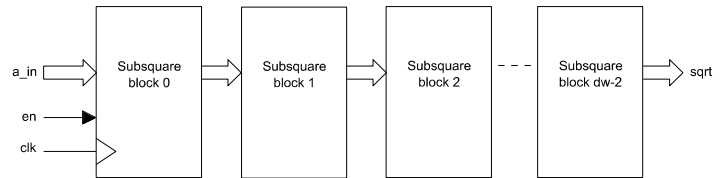


Figure 1: Pipelined square-root architecture

## General Description

PIPE\_SQRT (Figure 1) is a pipelined square-root with configurable data width. The design is fully scalable and modular permitting the user to specify large bit widths without compromising maximum attainable clock-speed.

The function accepts input values as unsigned integers whose width is specified by the parameter *dw*. The output result also contains *dw* bits and is separated into an integer part and fractional part each of *dw*/2 bits wide. Figure 2 below shows an example calculation using 8-bit arithmetic. The input  $\sqrt{00111101}$  ( $\sqrt{61}$ ) would result in an output of: 0111.1101 – i.e. an integer part of 7 and a fractional part of 0.8125.

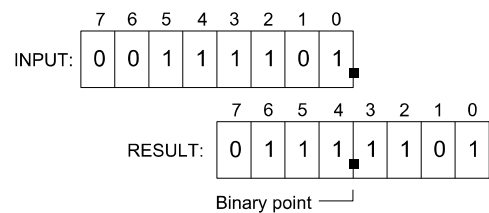


Figure 2: 8-bit square-root example

Values are sampled on the rising clock-edge of *clk* when *en* is high. When the generic parameter *opt\_mode* = 0, the function has a fixed cycle latency of *dw*-1. When *opt\_mode* = 1, then latency is equal to *dw*/2.

## Functional Timing

Figure 3 demonstrates the calculation of  $\sqrt{5}$ . In this example, the generic parameter *dw* has been set to 4 and *opt\_mode* is set to 0. The result has a latency of 4 clock cycles.

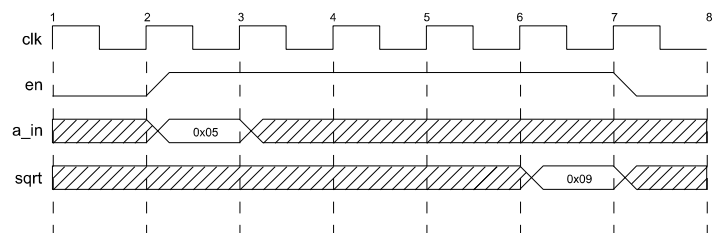


Figure 3: Calculation of  $\sqrt{5}$

## 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
pipe_sqrt_subsquare.vhd	Subtract square block
pipe_sqrt.vhd	Top-level block
pipe_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. pipe\_sqrt\_subsquare.vhd
2. pipe\_sqrt.vhd
3. pipe\_sqrt\_bench.vhd

The VHDL testbench instantiates the square-root component and the user may modify the generic parameters as required. The simulation must be run for at least 2 ms during which time the square root function will be driven with a randomized sequence of input values. The test terminates automatically.

The simulation generates two text files called: *pipe\_sqrt\_in.txt* and *pipe\_sqrt\_out.txt*. These files respectively contain the input and output data samples captured at the interfaces during the test.

Figure 4 shows the results of the square-root function for the first 1000 natural numbers with the generic parameter *dw* = 16.

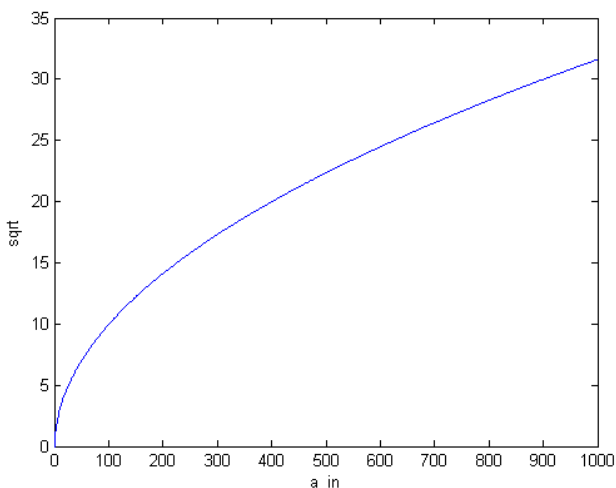


Figure 4: Plot of test results for function:  $f(x) = \sqrt{x}$

## Synthesis

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

- pipe\_sqrt.vhd
  - pipe\_sqrt\_subsquare.vhd

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx Virtex 5 and the Altera Stratix III series of FPGA devices. The lowest and highest speed grade devices have been chosen in both cases for comparison.

Note that the generic parameter *opt\_mode* will have a signification effect on the speed and area of the synthesized design. For the fastest designs, full pipelining should be used (*opt\_mode* = 0). For slower designs but with half the number of registers then half the number of pipeline stages should be used (*opt\_mode* = 1).

Trial synthesis results are shown with the generic parameters set to *dw* = 16 and *opt\_mode* = 0. Resource usage is specified after Place and Route.

### VIRTEX 5

Resource type	Quantity used
Slice register	329
Slice LUT	49
Block RAM	0
DSP48	15
Clock frequency (worst case)	161 MHz
Clock frequency (best case)	264 MHz

### STRATIX III

Resource type	Quantity used
Register	360
ALUT	303
Block Memory bit	0
DSP block 18	26
Clock frequency (worse case)	105 MHz
Clock frequency (best case)	158 MHz

## Revision History

Revision	Change description	Date
1.0	Initial revision	23/07/2008
1.1	Added new <i>opt_mode</i> generic parameter	06/10/2008