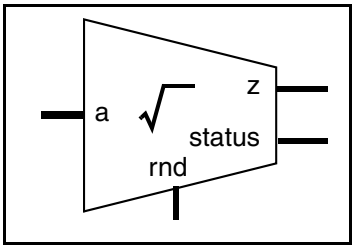
DW_fp_sqrt

Floating-Point Square Root

Version, STAR and Download Information: IP Directory

Features and Benefits



- The precision format is parameterizable for either IEEE single, double precision, or a user-defined custom format
- Hardware for denormal numbers of IEEE 754 standard is selectively provided.
- Accuracy conforms to IEEE 754 Floating-point standard[1]
- DesignWare datapath generator is employed for better timing and area

Description

DW_fp_sqrt computes the floating-point square root of a floating-point operand, a.

The input `rnd` is a 3-bit rounding mode value (see Rounding Modes in the *Datapath Floating-point Overview*). The output `status` has 8-bits of status flags.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
a	*exp_width* + *sig_width* + 1 bits	Input	Input data
rnd	3 bits	Input	Rounding mode
z	*exp_width* + *sig_width* + 1 bits	Output	Square root of `a`
status	8 bits	Output	See STATUS Flags in the *Datapath Floating-Point Overview*

Table 1-2 Parameter Description

Parameter	Values	Description
sig_width	2 to 253 bits	Word length of fraction field of floating-point numbers `a` and `z`
exp_width	3 to 31 bits	Word length of biased exponent of floating-point numbers `a` and `z`
ieee_compliance	0 or 1	When 1, the generated architecture is fully compliant with IEEE 754 standard, including the use of denormals and NaNs.

1. For more information, see IEEE 754 Compatibility in the *Datapath Floating-Point Overview*.

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
rtl	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW02.DW_FP_SQRT_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW_fp_sqrt_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_fp_sqrt.v	Verilog simulation model source code

DW_fp_sqrt provides the hardware for denormal numbers and NaNs of IEEE 754 standard. If the parameter `ieee_compliance` is turned off, denormal numbers are considered as zeros, and NaNs are considered as Infinity. Otherwise, denormal numbers and NaNs become effective and additional hardware for the square root of denormal numbers is integrated.

Alternative Implementation of Floating-point Square Root Using DW_lp_fp_multifunc

The floating-point square root operation can also be implemented by DW_lp_fp_multifunc component (a member of the minPower Library, licensed separately), which evaluates the value of floating-point square root with 1 ulp error bound. There will be 1 ulp difference between the value from DW_lp_fp_multifunc and the value from DW_fp_sqrt. Performance and area of the synthesis results are different between the DW_fp_sqrt and reciprocal implementation of the DW_lp_fp_multifunc, depending on synthesis constraints, library cells and synthesis environments. By comparing performance and area between the square root implementation of DW_lp_fp_multifunc and DW_fp_sqrt component, the DW_lp_fp_multifunc provides more choices for the better synthesis results. Below is an example of the Verilog description for the floating-point square root of the DW_lp_fp_multifunc. For more detailed information, see the DW_lp_fp_multifunc datasheet.

```
DW_lp_fp_multifunc #(sig_width, exp_width, ieee_compliance, 2) U1 (
    .A(A),
    .FUNC(16'h0002),
    .RND(3'h0),
    .Z(Z),
    .STATUS(STATUS)
);
```

For more information about the floating-point, including `status` flag bits, and integer and floating-point formats, refer to the *Datapath Floating-Point Overview*.

Related Topics

- Datapath Floating-Point Overview
- DesignWare Building Block IP Documentation Overview

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.dw_foundation_comp.all;

entity DW_fp_sqrt_inst is
  generic (
    inst_sig_width : POSITIVE := 23;
    inst_exp_width : POSITIVE := 8;
    inst_ieee_compliance : INTEGER := 0
  );
  port (
    inst_a : in std_logic_vector(inst_sig_width+inst_exp_width downto 0);
    inst_rnd : in std_logic_vector(2 downto 0);
    z_inst : out std_logic_vector(inst_sig_width+inst_exp_width downto 0);
    status_inst : out std_logic_vector(7 downto 0)
  );
end DW_fp_sqrt_inst;

architecture inst of DW_fp_sqrt_inst is
begin

  -- Instance of DW_fp_sqrt
  U1 : DW_fp_sqrt
    generic map (
      sig_width => inst_sig_width,
      exp_width => inst_exp_width,
      ieee_compliance => inst_ieee_compliance
    )
    port map (
      a => inst_a,
      rnd => inst_rnd,
      z => z_inst,
      status => status_inst
    );

end inst;

-- pragma translate_off
configuration DW_fp_sqrt_inst_cfg_inst of DW_fp_sqrt_inst is
  for inst
  end for;
end DW_fp_sqrt_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_fp_sqrt_inst( inst_a, inst_rnd, z_inst, status_inst );

parameter inst_sig_width = 23;
parameter inst_exp_width = 8;
parameter inst_ieee_compliance = 0;


input [inst_sig_width+inst_exp_width : 0] inst_a;
input [2 : 0] inst_rnd;
output [inst_sig_width+inst_exp_width : 0] z_inst;
output [7 : 0] status_inst;

  // Instance of DW_fp_sqrt
  DW_fp_sqrt #(inst_sig_width, inst_exp_width, inst_ieee_compliance) U1 (
                      .a(inst_a),
                      .rnd(inst_rnd),
                      .z(z_inst),
                      .status(status_inst) );

endmodule
```

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043

www.synopsys.com