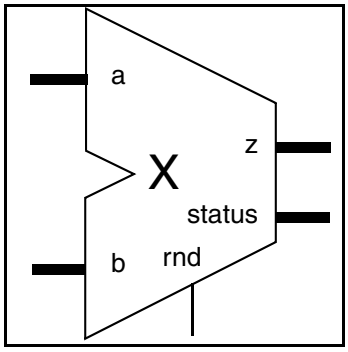
DW_fp_mult

Floating-Point Multiplier

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.
- Fully compatible with the IEEE 754 Floating-point standard[1] with proper set of parameters
- DesignWare datapath generator is employed for better timing and area

Description

DW_fp_mult is a floating-point multiplier that multiplies two floating-point values, `a` and `b`, to produce a floating-point product, `z`.

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

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
a	*exp_width* + *sig_width* + 1 bits	Input	Multiplier
b	*exp_width* + *sig_width* + 1 bits	Input	Multiplicand
rnd	3 bits	Input	Rounding mode
z	*exp_width* + *sig_width* + 1 bits	Output	Product of `a` X `b`
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 Default: 23	Word length of fraction field of floating-point numbers `a`, `b`, and `z`
exp_width	3 to 31 bits Default: 8	Word length of biased exponent of floating-point numbers `a`, `b`, and `z`

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

Table 1-2 Parameter Description (Continued)

Parameter	Values	Description
ieee_compliance	0 or 1 Default: 0	IEEE 754 standard support. 0: MC (Module Compiler) compatible 1: IEEE 754 standard compatible, including NaNs and denormal expressions

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
rtl	Synthesis model	DesignWare
str[a]	Synthesis model -- delay optimized for non-ieee_compliance	DesignWare

a. Only available when parameter ieee_compliance is 0.

Table 1-4 Simulation Models

Model	Function
DW02.DW_FP_MULT_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW_fp_mult_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_fp_mult.v	Verilog simulation model source code

DW_fp_mult 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 multiplication of denormal numbers is integrated.

For more information about the floating-point system defined for all the DW_fp components, 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;
-- If using numeric types from std_logic_arith package,
-- comment the preceding line and uncomment the following line:
-- use DWARE.DW_Foundation_comp_arith.all;

entity DW_fp_mult_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_b      : 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_mult_inst;


architecture inst of DW_fp_mult_inst is

begin

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

end inst;


-- pragma translate_off
configuration DW_fp_mult_inst_cfg_inst of DW_fp_mult_inst is
```

```
    for inst
    end for;
end DW_fp_mult_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_fp_mult_inst( inst_a, inst_b, inst_rnd, z_inst, status_inst );

parameter sig_width = 23;
parameter exp_width = 8;
parameter ieee_compliance = 1;


input [sig_width+exp_width : 0] inst_a;
input [sig_width+exp_width : 0] inst_b;
input [2 : 0] inst_rnd;
output [sig_width+exp_width : 0] z_inst;
output [7 : 0] status_inst;

    // Instance of DW_fp_mult
    DW_fp_mult #(sig_width, exp_width, ieee_compliance)
      U1 ( .a(inst_a), .b(inst_b), .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