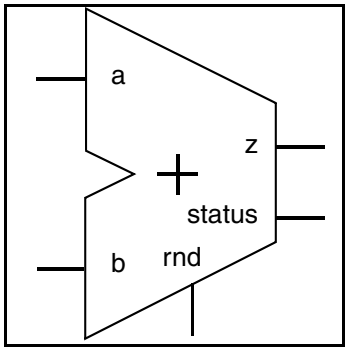
DW_fp_add

Floating-Point Adder

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
- Exponents can range from 3 to 31 bits
- Fractional part of the floating-point number can range from 2 to 253 bits
- 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_add is a floating-point component that adds two floating-point values, `a` and `b`, to produce a floating-point sum, `z`. The input `rnd` is a 3-bit rounding mode (see Rounding Modes in the *Datapath Floating-Point Overview*) and the output `status` is an 8-bit status value.

Table 1-1 Pin Description

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

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	When 1, the generated architecture is fully compliant with IEEE 754 standard, including the use of denormals and NaNs.

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_ADD_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW_fp_add_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_fp_add.v	Verilog simulation model source code

Table 1-5 Functional Description

a	b	status[a]	z[b]
a (floating-point)	b (floating-point)	*	a + b (floating-point)

a. The details of `status` flags, if used, can be found in Table 9 of the *Datapath Floating-Point Overview.*
b. The value for any a + b is defined by the rounding mode.

The parameter `ieee_compliance` controls the functionality of this component. When `ieee_compliance` = 0, the component is backward compatible with the DW_fp_add component (previously available in this library). With this setting, the DW_fp_add component does not work with NaNs and denormals. NaNs are considered infinities and denormals are considered zeros.

When `ieee_compliance` = 1, the component is fully compliant with the IEEE 754 standard and therefore operates with NaNs and denormals.

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_add_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_add_inst;


architecture inst of DW_fp_add_inst is

begin

    -- Instance of DW_fp_add
    U1 : DW_fp_add
    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_add_inst_cfg_inst of DW_fp_add_inst is
 for inst
 end for; -- inst
end DW_fp_add_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

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

parameter sig_width = 23;
parameter exp_width = 8;
parameter ieee_compliance = 0;


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_add
    DW_fp_add #(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