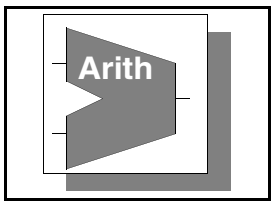
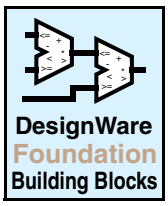




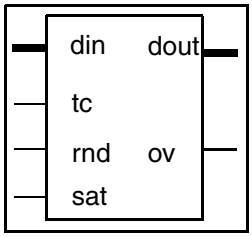
DW01_satrnd

Arithmetic Saturation and Rounding Logic

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterized word length
- Dynamically or statically configurable
- Arithmetic saturation (clipping) or wrap-around for MSB truncation
- Round to nearest logic for LSB truncation
- Signed and unsigned data operation



Applications

- Digital Signal Processing (DSP)
- Graphics

Description

DW01_satrnd performs arithmetic, precision-handling rounding and saturation functions on its input bus `din`.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
din	*width* bit(s)	Input	Input data
tc	1 bit	Input	Two's complement control 0 = unsigned 1 = signed
sat	1 bit	Input	Saturation enable 0 = no saturation 1 = enable saturation
rnd	1 bit	Input	Rounding enable 0 = no rounding 1 = enable rounding
ov	1 bit	Output	Overflow status
dout	*msb_out* – *lsb_out* + *1* bit(s)	Output	Output data

Table 1-2 Parameter Description

Parameter	Values	Description
width	≥ 2 Default: 16	Word length of din
msb_out	*width*–1 ≥ *msb_out* > *lsb_out* Default: 15	dout MSB position after truncation of din MSBs
lsb_out	*msb_out* > *lsb_out* ≥ 0 Default: 0	dout LSB position after truncation of din LSBs

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW01.DW01_SATRND_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW01_satrnd_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW01_satrnd.v	Verilog simulation model source code

The width of the `din` bus is set with the width parameter. The output bus, `dout`, is a subset of `din`. The width of `dout` is determined by the `msb_out` and `lsb_out` parameters such that `dout` equals `din(msb_out:lsb_out)`.

When the least significant bits `din(lsb_out-1:0)` are truncated, the round-to-nearest operation is performed if rounding is enabled (`rnd`=1) and overflow has not occurred (`ov`=0).

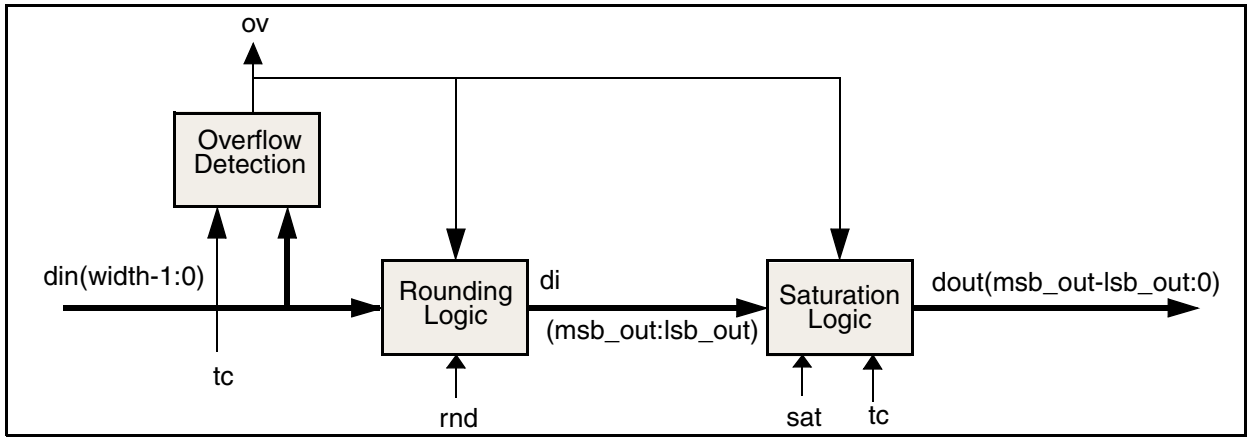
When the most significant bits `din(width-1:msb_out+1)` are truncated or rounding of a maximally positive value has occurred, data wrap-around may occur as a consequence of the finite data word length.

If saturation is enabled (`sat`=1) and overflow has occurred (`ov` 1), the output `dout` is saturated (or clipped) according to whether:

- The input data `din` is signed (`tc`=1) or unsigned (`tc`=0), and
- The sign of `din`.

Saturation always takes precedence over rounding. If overflow occurs and both saturation and rounding modes are enabled, the output `dout` is the saturation value.

Figure 1-1 Block Diagram



Related Topics

- Math – Arithmetic Overview
- DesignWare Building Block IP Documentation Overview

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW01_satrnd_inst is
  generic ( inst_width   : POSITIVE := 8;
            inst_msb_out : NATURAL  := 7;
            inst_lsb_out : NATURAL  := 1 );
  port ( inst_din  : in std_logic_vector(inst_width-1 downto 0);
         inst_tc   : in std_logic;
         inst_sat  : in std_logic;
         inst_rnd  : in std_logic;
         ov_inst   : out std_logic;
         dout_inst : out std_logic_vector(inst_msb_out-inst_lsb_out downto 0)
       );
end DW01_satrnd_inst;

architecture inst of DW01_satrnd_inst is
begin

  -- Instance of DW01_satrnd
  U1 : DW01_satrnd
    generic map ( width => inst_width,   msb_out => inst_msb_out,
                  lsb_out => inst_lsb_out )
    port map ( din => inst_din,   tc => inst_tc,   sat => inst_sat,
               rnd => inst_rnd,   ov => ov_inst,   dout => dout_inst );
end inst;

-- pragma translate_off
configuration DW01_satrnd_inst_cfg_inst of DW01_satrnd_inst is
  for inst
  end for; -- inst
end DW01_satrnd_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW01_satrnd_inst( inst_din, inst_tc, inst_sat, inst_rnd,
                         ov_inst, dout_inst );

  parameter width = 8;
  parameter msb_out = 7;
  parameter lsb_out = 1;

  input [width-1 : 0] inst_din;
  input inst_tc;
  input inst_sat;
  input inst_rnd;
  output ov_inst;
  output [msb_out-lsb_out : 0] dout_inst;

  // Instance of DW01_satrnd
  DW01_satrnd #(width, msb_out, lsb_out)
    U1 ( .din(inst_din),   .tc(inst_tc),   .sat(inst_sat),
         .rnd(inst_rnd),   .ov(ov_inst),   .dout(dout_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