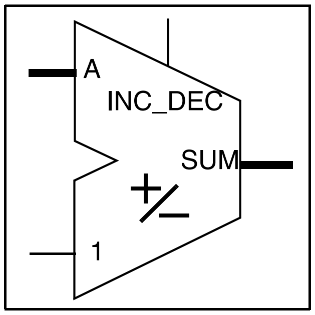
DW01_incdec

Incrementer-Decrementer

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterized word length



Description

DW01_incdec is an incrementer-decrementer. DW01_incdec either adds `1` to (INC_DEC=0) or subtracts `1` (INC_DEC=1) from an input number `A` to produce the output `SUM`.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*width* bit(s)	Input	Input data
INC_DEC	1 bit	Input	Increment control 0 = increment (A + 1) 1 = decrement (A – 1)
SUM	*width* bit(s)	Output	Increment (A + 1) or decrement (A – 1)

Table 1-2 Parameter Description

Parameter	Values	Function
width	≥ 1	Word length of A and SUM

Table 1-3 Synthesis Implementations[a]

Implementation Name	Function	License Feature Required
rpl	Ripple carry synthesis model	none
cla	Carry look-ahead synthesis model	none
pparch	Delay-optimized flexible parallel-prefix	DesignWare
apparch	Area-optimized flexible architecture that can be optimized for area, for speed, or for area, speed	DesignWare

a. During synthesis, Design Compiler will select the appropriate architecture for your constraints. However, you may force Design Compiler to use any architectures described in this table. For more, see *DesignWare Building Block IP User Guide*

Table 1-4 Obsolete Synthesis Implementations[a]

Implementation	Function	Replacement Implementation
clf	Fast carry-look-ahead synthesis model	pparch

a. DC versions and DesignWare EST releases linked to DC versions prior to 2007.03 will still incude these implementations.

Table 1-5 Simulation Models

Model	Function
DW01.DW01_INCDEC_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW01_incdec_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW01_incdec.v	Verilog simulation model source code

Related Topics

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

HDL Usage Through Operator Inferencing - VHDL

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity  DW01_incdec_oper is
  generic(wordlength: integer := 8 );
  port(in1     : in STD_LOGIC_VECTOR(wordlength-1 downto 0);
       inc_dec : in STD_LOGIC;
       sum     : out STD_LOGIC_VECTOR(wordlength-1 downto 0) );
end DW01_incdec_oper;

architecture oper of DW01_incdec_oper is
  signal in_signed,sum_signed: SIGNED(wordlength-1 downto 0);
begin

  in_signed <= SIGNED(in1);
  process (in_signed, inc_dec)
  begin
    if (inc_dec = '1') then
      sum_signed <= in_signed - 1;
    else
      sum_signed <= in_signed + 1;
    end if;
  end process;
  sum <= STD_LOGIC_VECTOR(sum_signed);
end oper;
```

HDL Usage Through Operator Inferencing - Verilog

```
module DW01_incdec_oper(in1,inc_dec,sum);
  parameter wordlength = 8;

  input [wordlength-1:0] in1;
  input inc_dec;
  output [wordlength-1:0] sum;
  reg [wordlength-1:0] sum;

always @(in1 or inc_dec)
begin
  if (inc_dec == 1)
    sum = in1 - 1;
  else
    sum = in1 + 1;
  end
endmodule
```

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_incdec_inst is
  generic ( inst_width : NATURAL := 8 );
  port ( inst_A       : in std_logic_vector(inst_width-1 downto 0);
         inst_INC_DEC : in std_logic;
         SUM_inst     : out std_logic_vector(inst_width-1 downto 0) );
end DW01_incdec_inst;

architecture inst of DW01_incdec_inst is
begin
  -- Instance of DW01_incdec
  U1 : DW01_incdec
    generic map ( width => inst_width )
    port map ( A => inst_A, INC_DEC => inst_INC_DEC, SUM => SUM_inst );
end inst;

-- pragma translate_off
configuration DW01_incdec_inst_cfg_inst of DW01_incdec_inst is
  for inst
  end for; -- inst
end DW01_incdec_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW01_incdec_inst( inst_A, inst_INC_DEC, SUM_inst );

  parameter width = 8;

  input [width-1 : 0] inst_A;
  input inst_INC_DEC;
  output [width-1 : 0] SUM_inst;

  // Instance of DW01_incdec
  DW01_incdec #(width)
    U1 ( .A(inst_A), .INC_DEC(inst_INC_DEC), .SUM(SUM_inst) );
endmodule
```

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