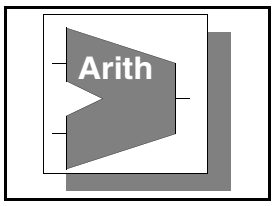
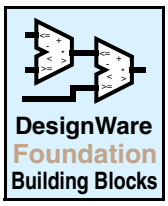




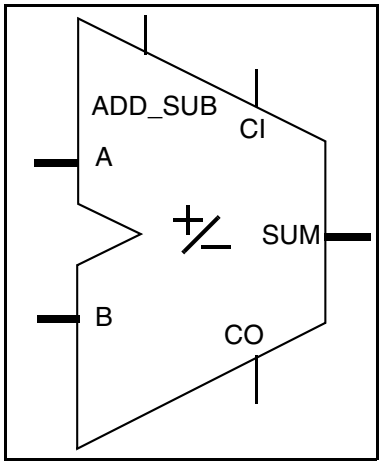
DW01_addsub

Adder-Subtractor

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterized word length
- Carry-in and carry-out signals



Description

DW01_addsub is a two-input adder-subtractor. DW01_addsub adds or subtracts two operands A and B with a carry-in (CI) to produce the output SUM with a carry-out (CO).

The ADD_SUB signal determines whether the operation to perform is an addition (ADD_SUB=0) or a subtraction (ADD_SUB=1).

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*width* bit(s)	Input	Input data
B	*width* bit(s)	Input	Input data
CI	1 bit	Input	Carry/borrow-in
ADD_SUB	1 bit	Input	Addition/subtraction control
SUM	*width* bit(s)	Output	Sum (A + B + CI) or difference (A – B – CI)
CO	1 bit	Output	Carry/borrow-out

Table 1-2 Parameter Description

Parameter	Values	Description
width	≥ 1	Word length of A, B, and SUM

Table 1-3 Synthesis Implementations[a]

Implementation Name	Implementation	License Feature Required
rpl	Ripple-carry synthesis model	none
cla	Carry-look-ahead synthesis model	none
pparch	Delay-optimized flexible parallel-prefix	DesignWare

Table 1-3 Synthesis Implementations[a] (Continued)

Implementation Name	Implementation	License Feature Required
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
bk	Brent-Kung architecture synthesis model	pparch
csm	Conditional-sum synthesis model	pparch
rpcs	Ripple-carry-select architecture	pparch

a. DC versions prior to 2007.03 will still include these implementations.

Table 1-5 Simulation Models

Model	Function
DW01.DW01_ADDSUB_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW01_addsub_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW01_addsub.v	Verilog simulation model source code

Table 1-6 Functional Description

ADD_SUB	A	B	CI	SUM	CO
0	A	B	0	A + B	Carry-out
0	A	B	1	A + B + 1	Carry-out
1	A	B	0	A - B	Carry-out
1	A	B	1	A - B - 1	Carry-out

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_addsub_oper is
  generic(wordlength: integer := 8);
  port(in1, in2 : in STD_LOGIC_VECTOR(wordlength-1 downto 0);
       ctl      : in STD_LOGIC;
       sum      : out STD_LOGIC_VECTOR(wordlength-1 downto 0));
end DW01_addsub_oper;

architecture oper of DW01_addsub_oper is
  signal in1_signed, in2_signed, sum_signed: SIGNED(wordlength-1 downto 0);
begin
  in1_signed <= SIGNED(in1);
  in2_signed <= SIGNED(in2);
  process (in1_signed, in2_signed, ctl)
  begin
    if (ctl = '0') then
      sum_signed <= in1_signed + in2_signed;
    else
      sum_signed <= in1_signed - in2_signed;
    end if;
  end process;
  sum <= STD_LOGIC_VECTOR(sum_signed);
end oper;
```

HDL Usage Through Operator Inferencing - Verilog

```
module DW01_addsub_oper(in1,in2,ctl,sum);
  parameter wordlength = 8;

  input [wordlength-1:0] in1,in2;
  input ctl;
  output [wordlength-1:0] sum;
  reg [wordlength-1:0] sum;

  always @(in1 or in2 or ctl)
  begin
    if (ctl == 0)
      sum = in1 + in2;
    else
      sum = in1 - in2;
  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_addsub_inst is
  generic ( inst_width : NATURAL := 8 );
  port ( inst_A       : in std_logic_vector(inst_width-1 downto 0);
         inst_B       : in std_logic_vector(inst_width-1 downto 0);
         inst_CI      : in std_logic;
         inst_ADD_SUB : in std_logic;
         SUM_inst     : out std_logic_vector(inst_width-1 downto 0);
         CO_inst      : out std_logic );
end DW01_addsub_inst;

architecture inst of DW01_addsub_inst is
begin

  -- Instance of DW01_addsub
  U1 : DW01_addsub
    generic map ( width => inst_width )
    port map ( A => inst_A, B => inst_B, CI => inst_CI,
               ADD_SUB => inst_ADD_SUB, SUM => SUM_inst, CO => CO_inst );
end inst;

-- pragma translate_off
configuration DW01_addsub_inst_cfg_inst of DW01_addsub_inst is
  for inst
  end for; -- inst
end DW01_addsub_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW01_addsub_inst( inst_A, inst_B, inst_CI, inst_ADD_SUB,
                         SUM_inst, CO_inst );

  parameter width = 8;

  input [width-1 : 0] inst_A;
  input [width-1 : 0] inst_B;
  input inst_CI;
  input inst_ADD_SUB;
  output [width-1 : 0] SUM_inst;
  output CO_inst;

  // Instance of DW01_addsub
  DW01_addsub #(width)
    U1 ( .A(inst_A), .B(inst_B), .CI(inst_CI), .ADD_SUB(inst_ADD_SUB),
         .SUM(SUM_inst), .CO(CO_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