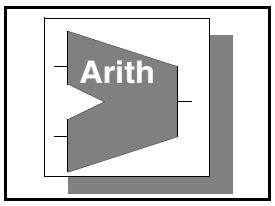
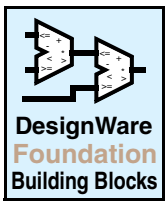




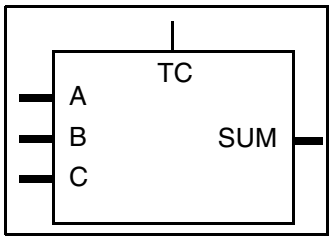
DW02_prod_sum1

Multiplier-Adder

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterized word length



Applications

- Multiply and accumulate
- Digital filtering

Description

DW02_prod_sum1 performs the mathematical operation `SUM = A × B + C`. This component is an extended version of DW02_mac, offering automatic sign extension on the output port `SUM` according to the parameter SUM_width. This component can also be considered a special case of DW02_prod_sum.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
A	*A_width* bit(s)	Input	Input data
B	*B_width* bit(s)	Input	Input data
C	*SUM_width* bit(s)	Input	Input data
TC	1 bit	Input	Two's complement 0 = unsigned 1 = signed
SUM	*SUM_width* bit(s)	Output	Sum of products

Table 1-2 Parameter Description

Parameter	Values	Description
A_width	≥ 1	Word length of A
B_width	≥ 1[a]	Word length of B
SUM_width	≥ 1	Word length of C and output SUM

a. For nbw implementation, *A_width*+*B_width* ≤ 36. Due to concern of implementation selection run time, a limitation is set for *A_width* and *B_width*.

Table 1-3 Synthesis Implementations[a]

Implementation Name	Function	License Feature Required
pparch	Delay-optimized flexible parallel-prefix	DesignWare
apparch	Area-optimized flexible parallel-prefix	DesignWare

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

Table 1-4 Obsolete Synthesis Implementations[a]

Implementation	Function	Replacement Implementation
csa	Carry-save array synthesis model	pparch
wall	Booth-recoded Wallace-tree synthesis model[b]	pparch
nbw	Either a non-Booth (*A_width*+*B_width* ≤ 41) or a Booth Wallace-tree (*A_width*+*B_width* > 41) synthesis model[c]	pparch

a. DC versions and DesignWare EST releases linked to DC versions prior to 2007.03 will still incude these implementations.
b. In most cases, the wall implementation generates both faster and smaller circuits for medium- to large-sized multipliers.
c. In cases where *A_width*+*B_width* ≤ 41, the nbw implementation generates a non-Booth recoded Wallace-tree multiplier. For multipliers having products larger than 41 bits (such as, *A_width*+*B_width* > 41) the nbw implementation produces a Booth-recoded multiplier identical to the wall implementation.

Table 1-5 Simulation Models

Model	Function
DW02.DW02_PROD_SUM1_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW02_prod_sum1_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW02_prod_sum1.v	Verilog simulation model source code

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 DW02_prod_sum1_inst is
  generic ( inst_A_width : NATURAL := 5;
            inst_B_width : NATURAL := 5;
            inst_SUM_width : NATURAL := 11 );
  port ( inst_A   : in std_logic_vector(inst_A_width-1 downto 0);
         inst_B   : in std_logic_vector(inst_B_width-1 downto 0);
         inst_C   : in std_logic_vector(inst_SUM_width-1 downto 0);
         inst_TC  : in std_logic;
         SUM_inst : out std_logic_vector(inst_SUM_width-1 downto 0) );
end DW02_prod_sum1_inst;

architecture inst of DW02_prod_sum1_inst is
begin

  -- Instance of DW02_prod_sum1
  U1 : DW02_prod_sum1
    generic map ( A_width => inst_A_width,   B_width => inst_B_width,
                  SUM_width => inst_SUM_width )
    port map ( A => inst_A,   B => inst_B,   C => inst_C,
               TC => inst_TC,   SUM => SUM_inst );
end inst;

-- pragma translate_off
configuration DW02_prod_sum1_inst_cfg_inst of DW02_prod_sum1_inst is
  for inst
  end for; -- inst
end DW02_prod_sum1_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW02_prod_sum1_inst( inst_A, inst_B, inst_C, inst_TC, SUM_inst );

  parameter A_width = 5;
  parameter B_width = 5;
  parameter SUM_width = 11;

  input [A_width-1 : 0] inst_A;
  input [B_width-1 : 0] inst_B;
  input [SUM_width-1 : 0] inst_C;
  input inst_TC;
  output [SUM_width-1 : 0] SUM_inst;

  // Instance of DW02_prod_sum1
  DW02_prod_sum1 #(A_width, B_width, SUM_width)
    U1 ( .A(inst_A), .B(inst_B), .C(inst_C), .TC(inst_TC), .SUM(SUM_inst) );

endmodule
```