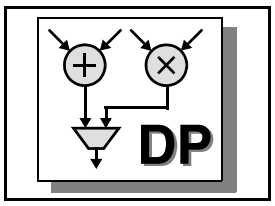

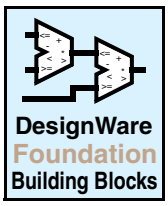

DWF_dp_simd_addc procedures

SIMD add with carries

Version, STAR and Download Information: IP Directory

Description

The DWF_dp_simd_addc procedures implement a configurable SIMD adder with input and output carries. They allow you to either add arguments `a` and `b` as full-width vectors (for example, one 32-bit addition) or to add smaller partitions of `a` and `b` using multiple parallel adders (for example, two 16-bit additions or four 8-bit additions), each with separate input carries `cin` and output carries `cout`. The argument `no_confs` specifies the number of possible configurations, and argument `conf` dynamically selects one configuration. Configuration with number `conf` has 2^{conf} partitions of size $width/2^{conf}$. Arguments `a` and `b` and the output `s` are either unsigned or signed (two's complement).

Table 1-1 Procedure Names

Function Name	Description
DWF_dp_simd_addc	VHDL unsigned SIMD add with carries
DWF_dp_simd_addc	VHDL signed (two's complement) SIMD add with carries
DWF_dp_simd_addc_uns	Verilog unsigned SIMD add with carries
DWF_dp_simd_addc_tc	Verilog signed (two's complement) SIMD add with carries

Table 1-2 Argument Description

Argument Name	Type	Direction	Width / Values	Description
a	Vector [a]	Input	width	Input addend
b	Vector [a]	Input	width	Input addend
cin	Vector [b]	Input	$2^{no\_confs-1}$	Input carries
no_confs	Integer	Input	≥ 2	Number of configurations (VHDL only, constant)
conf	Vector [b]	Input	$ceil(log_2[no\_confs])$	Configuration selection: 2^{conf} partitions of size $width/2^{conf}$
s	Vector [a]	Output	width	Output sum
cout	Vector [b]	Output	$2^{no\_confs-1}$	Output carries

a. `unsigned` or `signed` in VHDL
b. `std_logic_vector` in VHDL

Table 1-3 Parameter Description (Verilog)

Parameter	Values	Description
width	≥ 2, must be a multiple of $2^{no\_confs-1}$	Word length
no_confs	≥ 2	Number of configurations

Verilog Include File: DW_dp_simd_addc_function.inc

Functional Description

```
DWF_dp_simd_addc (a[width-1:0], b[width-1:0], cin[2^no_confs-1-1:0],
                  no_confs, conf[bit_width(no_confs)-1:0],
                    z[width-1:0], cout[2^no_confs-1-1:0])

conf = 0:
  { cout[2^no_confs-1-1],     z[width-1:0] }          =
    a[width-1:0]           + b[width-1:0]          + cin[0]

conf = 1:
  { cout[2^no_confs-1-1],     z[width-1:width/2] }    =
    a[width-1:width/2]     + b[width-1:width/2]    + cin[2^no_confs-1/2]
  { cout[2^no_confs-1/2-1],   z[width/2-1:0] }        =
    a[width/2-1:0]         + b[width/2-1:0]        + cin[0]

conf = 2:
  { cout[2^no_confs-1-1],     z[width-1:width*3/4] }  =
    a[width-1:width*3/4]   + b[width-1:width*3/4]  + cin[2^no_confs-1*3/4]
  { cout[2^no_confs-1*3/4-1], z[width*3/4-1:width/2] } =
    a[width*3/4-1:width/2] + b[width*3/4-1:width/2] + cin[2^no_confs-1/2]
  { cout[2^no_confs-1/2-1],   z[width/2-1:width/4] }  =
    a[width/2-1:width/4]   + b[width/2-1:width/4]  + cin[2^no_confs-1/4]
  { cout[2^no_confs-1/4-1],   z[width/4-1:0] }        =
    a[width/4-1:0]         + b[width/4-1:0]        + cin[0]
...
```

Example: `width = 32, no_confs = 3`

```
conf = 0:
  { cout[3], z[31: 0] } = a[31: 0] + b[31: 0] + cin[0]

conf = 1:
  { cout[3], z[31:16] } = a[31:16] + b[31:16] + cin[2]
  { cout[1], z[15: 0] } = a[15: 0] + b[15: 0] + cin[0]

conf = 2:
  { cout[3], z[31:24] } = a[31:24] + b[31:24] + cin[3]
  { cout[2], z[23:16] } = a[23:16] + b[23:16] + cin[2]
  { cout[1], z[15: 8] } = a[15: 8] + b[15: 8] + cin[1]
  { cout[0], z[ 7: 0] } = a[ 7: 0] + b[ 7: 0] + cin[0]
```

For more information about the DesignWare datapath functions, refer to DesignWare Datapath Functions Overview.

Related Topics

- DesignWare Datapath Functions Overview
- DesignWare Building Block IP Documentation Overview

VHDL Example

```
library IEEE, DWARE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
use DWARE.DW_dp_functions.all;
-- DWARE.DW_dp_functions_arith package if IEEE.std_logic_arith is used

entity DWF_dp_simd_addc_test is
  port (op1, op2  : in  signed(31 downto 0);
        c_in      : in  std_logic_vector(3 downto 0);
        config_no : in  std_logic_vector(1 downto 0);
        sum       : out signed(31 downto 0);
        c_out     : out std_logic_vector(3 downto 0));
end DWF_dp_simd_addc_test;

architecture rtl of DWF_dp_simd_addc_test is
begin
  DWF_dp_simd_addc (a => op1, b => op2, cin => c_in,
                    no_confs => 3, conf => config_no,
                    s => sum, cout => c_out);
end rtl;
```

Verilog Example

```
module DWF_dp_simd_addc_test (op1, op2, c_in, config_no, sum, c_out);

  input  signed [31:0] op1, op2;
  input         [3:0]  c_in;
  input         [1:0]  config_no;
  output signed [31:0] sum;
  output        [3:0]  c_out;
  reg    signed [31:0] sum;
  reg           [3:0]  c_out;

  // Passes the parameters to the function
  parameter width    = 32;
  parameter no_confs = 3;

  // add "$SYNOPSYS/dw/sim_ver" to the search path for simulation
  `include "DW_dp_simd_addc_function.inc"

  always @* begin
    DWF_dp_simd_addc_tc (op1, op2, c_in, config_no, sum, c_out);
  end

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