DW01_prienc

Priority Encoder

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterized word length
- Inferable using a function call

Description

DW01_prienc encodes the input port A to a binary value on output port INDEX. The encoded value of A is determined by the bit position of the most significant 1 bit. All bits on A lower than the most significant 1 bit are "don't care".

The acceptable values for *INDEX_width* are greater than or equal to ceil($\log_2[A\_width]$), as described in Table 1-2. However, the recommended value for *INDEX_width* is ceil($\log_2[A\_width+1]$). With this value, the output INDEX can cover all the possible legal values. Note that if *INDEX_width* is equal to ceil($\log_2[A\_width]$) with *A_width* greater than 1, all 0s on pin A would result in the same INDEX output value as when a 1 is on the most significant bit of pin A.

Table 1-1   Pin Description

Pin Name	Width	Direction	Function
A	*A_width*	Input	Input data
INDEX	*INDEX_width*	Output	Binary encoded output data

Table 1-2   Parameter Description

Parameter	Values	Description
A_width	≥ 1	Word length of input A
INDEX_width	$\geq$ ceil($\log_2[A\_width]$)	Word length of output INDEX

Table 1-3   Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare
cla[a]	Synthesis model	DesignWare
aot	Synthesis model	DesignWare

a. The 'cla' implementation is only available when A_width ≤ 512.

Table 1-4   Simulation Models

Model	Function
DW01.DW01_PRIENC_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW01_prienc_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW01_prienc.v	Verilog simulation model source code

Table 1-5   Truth Table (A_WIDTH = 8, INDEX_WIDTH = 4)

A(7)	A(6)	A(5)	A(4)	A(3)	A(2)	A(1)	A(0)	INDEX(3:0)
0	0	0	0	0	0	0	0	0000
0	0	0	0	0	0	0	1	0001
0	0	0	0	0	0	1	X	0010
0	0	0	0	0	1	X	X	0011
0	0	0	0	1	X	X	X	0100
0	0	0	1	X	X	X	X	0101
0	0	1	X	X	X	X	X	0110
0	1	X	X	X	X	X	X	0111
1	X	X	X	X	X	X	X	1000

Related Topics

- Logic – Combinational Overview
- DesignWare IP Building Block Documentation Overview

HDL Usage Through Function Inferencing - VHDL

```vhdl
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use DWARE.DW_foundation_arith.all;

entity DW01_prienc_func is
  generic(func_A_width : integer := 8;func_INDEX_width : integer := 4);
  port(func_A: in std_logic_vector(func_A_width-1 downto 0);
        INDEX_func,INDEX_func_TC,INDEX_func_UNS: out
         std_logic_vector(func_INDEX_width-1 downto 0));
end DW01_prienc_func;

architecture func of DW01_prienc_func is
begin

  INDEX_func <= DWF_prienc(func_A,func_INDEX_width);
  INDEX_func_TC <= std_logic_vector(DWF_prienc(signed(func_A),
                                        func_INDEX_width));
  INDEX_func_UNS <= std_logic_vector(DWF_prienc(unsigned(func_A),
                                        func_INDEX_width));

end func;
```

HDL Usage Through Function Inferencing - Verilog

```verilog
module DW01_prienc_func (func_A,INDEX_func);
  parameter func_A_width = 8;
  parameter func_INDEX_width = 4;

  // Passes the widths to the prienc function
  parameter A_width = func_A_width;
  parameter INDEX_width = func_INDEX_width;

  // Please add search_path = search_path + {synopsys_root + "/dw/sim_ver"}
  // to your .synopsys_dc.setup file (for synthesis) and add
  // +incdir+$SYNOPSYS/dw/sim_ver+ to your verilog simulator command line
  // (for simulation).
  `include "DW01_prienc_function.inc"

  input [func_A_width-1:0] func_A;
  output [func_INDEX_width-1:0] INDEX_func;
  assign INDEX_func = DWF_prienc(func_A);

endmodule
```

HDL Usage Through Component Instantiation - VHDL

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

entity DW01_prienc_inst is
  generic (inst_A_width : POSITIVE := 8;
           inst_INDEX_width : POSITIVE := 4);
  port (inst_A : in std_logic_vector(inst_A_width-1 downto 0);
        INDEX_inst : out std_logic_vector(inst_INDEX_width-1 downto 0));
end DW01_prienc_inst;

architecture inst of DW01_prienc_inst is
begin

  -- Instance of DW01_prienc
  U1 : DW01_prienc
    generic map ( A_width => inst_A_width, INDEX_width => inst_INDEX_width )
    port map ( A => inst_A, INDEX => INDEX_inst );
end inst;

-- pragma translate_off
configuration DW01_prienc_inst_cfg_inst of DW01_prienc_inst is
  for inst
  end for; -- inst
end DW01_prienc_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW01_prienc_inst( inst_A, INDEX_inst );

    parameter A_width = 8;
    parameter INDEX_width = 4;

    input [A_width-1 : 0] inst_A;
    output [INDEX_width-1 : 0] INDEX_inst;

    // Instance of DW01_prienc
    DW01_prienc #(A_width, INDEX_width)
      U1 ( .A(inst_A), .INDEX(INDEX_inst) );

endmodule
```

Copyright Notice and Proprietary Information