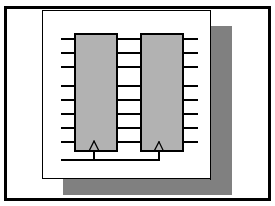
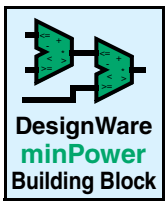




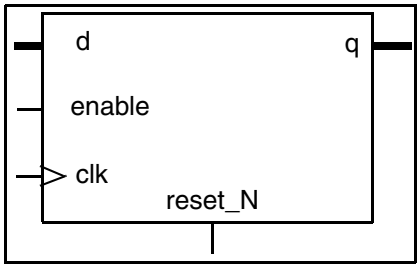
DW03_reg_s_pl

Register with Synchronous Enable Reset

Version, STAR and Download Information: IP Directory

Features and Benefits

- Parameterizable data width
- Parameterized reset to any constant value
- Multiple synthesis implementations
- Provides minPower benefits with the DesignWare-LP license.



Description

DW03_reg_s_pl provides an optimal implementation of a register that is synchronously reset and enabled.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
d	*width* bit(s)	Input	Input data bus
clk	1 bit	Input	Clock
reset_N	1 bit	Input	Synchronous reset
enable	1 bit	Input	Enables all operations
q	*width* bi(s)	Output	Output data bus

Table 1-2 Parameter Description

Parameter	Values	Description
width	1 to 31 Default: 8	Width of `d` and `q` buses
reset_value	0 to 2^{width}–1 when *width* ≤ 31; 0 when *width* ≥ 32 Default: 0	Resets to a constant

Table 1-3 Synthesis Implementations[a]

Implementation Name	Function	License Feature Required
str	Single-bit flip-flops synthesis model	DesignWare
mbstr	Multiple-bit flip-flops synthesis model	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 Simulation Models

Model	Function
DW03.DW03_REG_S_PL_CFG_SIM	Design unit name for VHDL simulation
dw/dw03/src/DW03_reg_s_pl_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW03_reg_s_pl_sim.v	Verilog simulation model source code

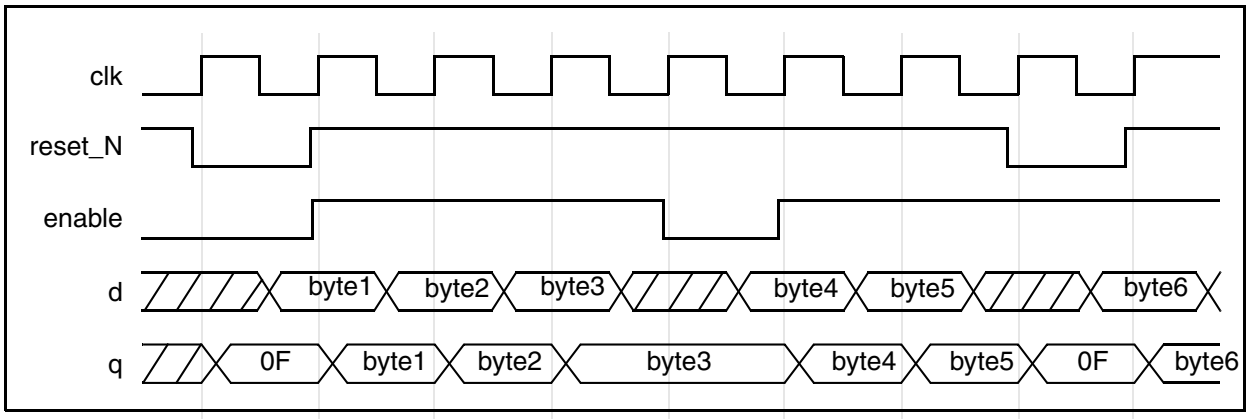
This component is composed of D flip-flops. It can be designed to reset to any constant value. Setup and hold times are relative to the rising edge of the clock signal, `clk`, and are technology dependent.

The `width` parameter configures the width of the part. The `reset_value` parameter indicates the constant value to which you would like the flip-flops set upon reset. This parameter configures logic in front of each internal flip-flop which resets that flip-flop to a one or a zero state corresponding to its bit in the constant `reset_value`.

Due to VHDL and Verilog language limitations, parameters are represented by signed 32-bit integers. The maximum value a parameter can be is $2^{31}-1$. This limitation applies to DW03_reg_s_pl's parameters `width` and `reset_value`. If you set `reset_value` to a value that is larger than 0, then you can only use DW03_reg_s_pl up to 31 bits. If you only want to reset DW03_reg_s_pl to 0, then you can use any *width* without limitation.

Timing Diagram

Figure 1-1 Functional Operation - width = 8, reset_value = 15



Related Topics

- Memory – Registers 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 DW03_reg_s_pl_inst is
  generic ( inst_width : POSITIVE := 8;
            inst_reset_value : INTEGER := 0 );
  port ( inst_d       : in std_logic_vector(inst_width-1 downto 0);
         inst_clk     : in std_logic;
         inst_reset_N : in std_logic;
         inst_enable  : in std_logic;
         q_inst       : out std_logic_vector(inst_width-1 downto 0) );
end DW03_reg_s_pl_inst;
architecture inst of DW03_reg_s_pl_inst is
begin

  -- Instance of DW03_reg_s_pl
  U1 : DW03_reg_s_pl
    generic map ( width => inst_width,  reset_value => inst_reset_value )
    port map ( d => inst_d,  clk => inst_clk,  reset_N => inst_reset_N,
               enable => inst_enable,  q => q_inst );
end inst;

-- pragma translate_off
configuration DW03_reg_s_pl_inst_cfg_inst of DW03_reg_s_pl_inst is
  for inst
  end for; -- inst
end DW03_reg_s_pl_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW03_reg_s_pl_inst( inst_d, inst_clk, inst_reset_N,
                           inst_enable, q_inst );

  parameter width = 8;
  parameter reset_value = 0;

  input [width-1 : 0] inst_d;
  input inst_clk;
  input inst_reset_N;
  input inst_enable;
  output [width-1 : 0] q_inst;

  // Instance of DW03_reg_s_pl
  DW03_reg_s_pl #(width, reset_value)
    U1 ( .d(inst_d), .clk(inst_clk), .reset_N(inst_reset_N),
         .enable(inst_enable), .q(q_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