DesignWare® Building Block IP

User Guide

R-2020.09
DWBB_202009.0
September 2020

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.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

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.
www.synopsys.com

Contents

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For web access to all Synopsys DesignWare Building Block IP components and the latest documents, see:

https://www.synopsys.com/dw/buildingblock.php

For a version of this user guide with visible change bars, click here.

Date	Version	Description
December 2019	Q-2019.12	■ Updated title of the DW_crc_s component in Table 2-1 on page 17
March 2019	P-2019.03	■ Re-organized the structure and refreshed all content in this user guide
January 2019	O-2018.06-SP5	■ Updated comments in example in “Example in Verilog” on page 40
December 2018	O-2018.06-SP4	■ Updated “Library” statements in blocks of VHDL code ■ Added this Revision History table and the document links on this page

Preface

1.1 About This Manual

This manual is the entry point to the Synopsys® DesignWare® Building Block (DWBB) IP product and documentation. It is intended for users of Synopsys synthesis tools. The DWBB IP are part of the overall DesignWare IP Library.

These building blocks are technology-independent, micro architecture-level components that are tightly integrated into the Synopsys synthesis environment.

1.1.1 Manual Overview

This manual contains the following chapters and appendixes:

Chapter 1, “Introduction”	An introduction to the DWBB product and a starting point for its usage
Chapter 2, “DWBB Components”	A list of all DWBB components with hyperlinks to datasheets
Appendix A, “Standard Synthetic Operators”	A list of synthetic operators
Appendix B, “minPower Components By Category”	A set of lists of minPower components by category

1.2 Customer Support

- First, prepare debug information, if applicable. For example:
  - Create a waveforms file (such as VPD or VCD)
  - Identify the hierarchy path to the DWBB instance
  - Identify the timestamp of any signals or locations in the waveforms that are not understood
- *For fastest response*, enter a case through SolvNetPlus:

  a. https://solvnetplus.synopsys.com

  > **Note** SolvNetPlus does not support the Internet Explorer browser. Use a supported browser such as Google Chrome, Mozilla Firefox, Microsoft Edge, or Apple Safari.

  b. Click the **Cases** menu and then click **Create a New Case** (below the list of cases).

  c. Complete the mandatory fields that are marked with an asterisk and click **Save**.
  Note the following information:

  - **Product L1:** *DesignWare Building Blocks*
  - **Product L2:** Choose the component category
  - **Product L3:** Choose the component (for example, DW01_add)
  - **Release:** *R-2020.09*
  - Describe the details to clarify your problem; include configurations of DWBB components you are using and any warning or error messages. For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood.

  d. After creating the case, attach any debug files you created in the previous step.

  For more information about general usage information, refer to the following article in SolvNetPlus:

  https://solvnetplus.synopsys.com/s/article/SolvNetPlus-Usage-Help-Resources

- Or, send an e-mail message to support_center@synopsys.com (your e-mail will be queued and then, on a first-come, first-served basis, manually routed to the correct support engineer):
  - Include the Product L1 and Product L2 names, and Version number in your e-mail so it can be routed correctly.
  - For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood
  - Attach any debug files you created.
  - Or, telephone your local support center:
    - North America:
      Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.
    - All other countries:
      https://www.synopsys.com/support/global-support-centers.html

1
Introduction

The Synopsys® DesignWare® Building Blocks (DWBB) is a collection of reusable intellectual property blocks that are tightly integrated into the Synopsys synthesis environment.

The DWBB is part of the DesignWare Library. For an overview of the DesignWare Library, see:

https://www.synopsys.com/designware-ip/soc-infrastructure-ip/designware-library.html

This document is the entry point to the DWBB product and documentation. This chapter contains the following main sections:

- "Features and Benefits" on page 10
- "License Requirements" on page 11
- "Documentation" on page 11
- "File Structure" on page 12
- "DesignWare Library for DWBB" on page 13
- "Synthesis Optimization Flow" on page 14
- "minPower Overview" on page 15

1.1 Features and Benefits

The DWBB supports the entire design flow by providing multiple views, including synthesizable models, simulation models, datasheets, and examples. It includes the following:

- Synthesis models for Synopsys synthesis tools
  Note: third-party synthesis tools are not supported
- Separate simulation models that are supported for all simulators
- Formal verification resources for Formality

The DWBB components are:

- Pre-verified for quality
- Linked to high-level synthesis
- Parameterized in size and, for some components, functionality
- Technology-independent

The DWBB components can be organized into the following groups:

- **Logic components**: Combinational and sequential
- **Math components**: Arithmetic and trigonometric
- **Floating point components**: Arithmetic and trigonometric
- **Memory**: Registers, FIFOs, and FIFO controllers; RAMS; stack components
- **Clock Domain Crossing components**: Synchronization
- **Application-specific components**: Data integrity, interface, and JTAG components
- **DSP components**: Digital filters for digital signal processing (DSP) applications
- **GTECH components**: A technology-independent, gate-level library
- **minPower components**: Low-power versions of frequently used instantiated IP

  While several minPower components exist only as low-power components, many are DWBB components that have low-power features such as:

  - Datapath gating (with enable control)
  - Low-power pipeline control
  - Enhanced clock gating

1.2 License Requirements

For DWBB, the license requirements depend on which version you are using, as listed in Table 1-1:

Table 1-1 License Requirements

Version	License Feature Required
P-2019.03 and later	DesignWare
Before P-2019.03	DesignWare DesignWare-LP (for minPower components)

For general license setup and usage when running Synopsys synthesis tools, refer to the synthesis tool documentation.

1.3 Documentation

Documents for DWBB reside at `$SYNOPSYS/dw/doc` and include the following:

- User Guide (this document)
- Component datasheets
- Release Notes

Related documentation includes:

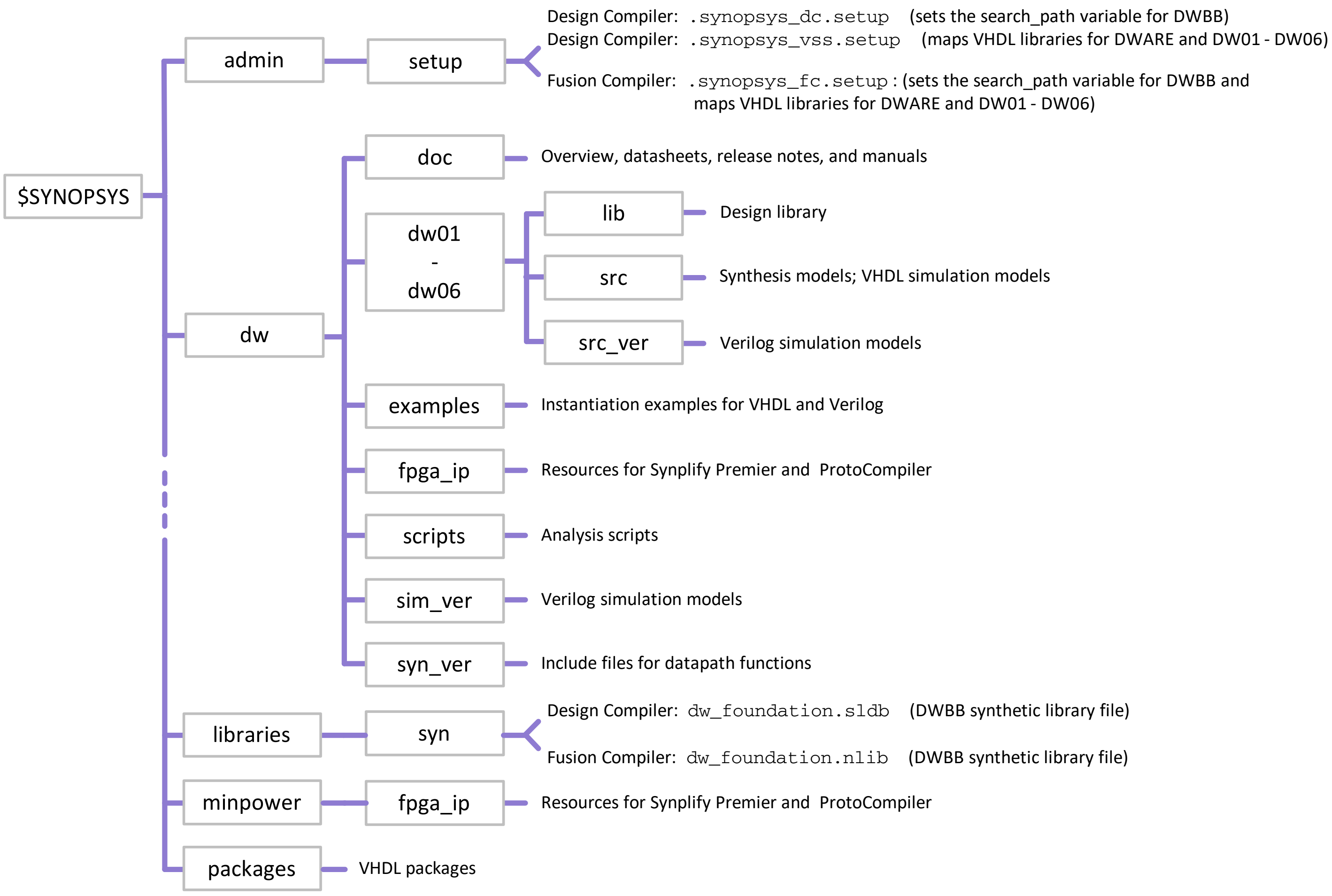
- Synopsys synthesis tools (see SolvNetPlus)
- *DesignWare GTECH Library Databook*
- *DesignWare Developers Guide* (for third-party model developers)

For the most recent DWBB documentation, see https://www.synopsys.com/dw/buildingblock.php. Click on a specific component and click Show Documents.

1.4 File Structure

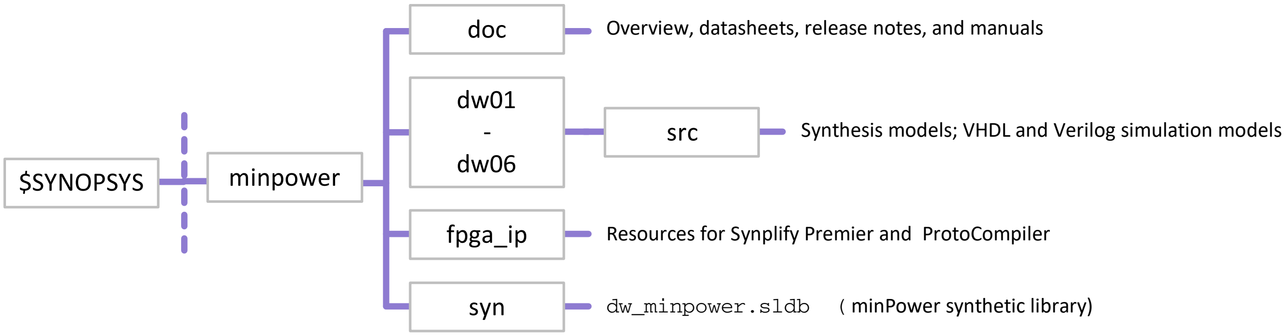
The high-level file structure for DWBB is shown in Figure 1-1.

Figure 1-1 DWBB File Structure


$SYNOPSYS
admin
setup
Design Compiler: .synopsys_dc.setup (sets the search_path variable for DWBB)
Design Compiler: .synopsys_vss.setup (maps VHDL libraries for DWARE and DW01 - DW06)
Fusion Compiler: .synopsys_fc.setup : (sets the search_path variable for DWBB and maps VHDL libraries for DWARE and DW01 - DW06)
dw
doc
Overview, datasheets, release notes, and manuals
dw01 - dw06
lib
Design library
src
Synthesis models; VHDL simulation models
src_ver
Verilog simulation models
examples
Instantiation examples for VHDL and Verilog
fpga_ip
Resources for Synplify Premier and ProtoCompiler
scripts
Analysis scripts
sim_ver
Verilog simulation models
syn_ver
Include files for datapath functions
libraries
syn
Design Compiler: dw_foundation.sldb (DWBB synthetic library file)
Fusion Compiler: dw_foundation.nlib (DWBB synthetic library file)
minpower
fpga_ip
Resources for Synplify Premier and ProtoCompiler
packages
VHDL packages

Prior to version P-2019.03, note that the minPower file structure is different, as shown in Figure 1-2.

Figure 1-2 minPower File Structure (Versions Before P-2019.03)


$SYNOPSYS
minpower
doc
Overview, datasheets, release notes, and manuals
dw01 - dw06
src
Synthesis models; VHDL and Verilog simulation models
fpga_ip
Resources for Synplify Premier and ProtoCompiler
syn
dw_minpower.sldb (minPower synthetic library)

1.5 DesignWare Library for DWBB

To make DWBB components accessible to the Synopsys tools, you specify paths to the necessary synthetic library files and design library directories.

- A synthetic library is a binary file that Synopsys synthesis tools use. It includes the following:
    - Port information
    - Parameters
    - Implementations (multiple architectural realizations of a synthetic module)
    - Bindings to synthetic operators
    - License requirements
- A design library is a UNIX directory that contains circuit descriptions (elaborated synthesis model) stored in binary formats.

The following synthetic library files, which reside at `$SYNOPSY/libraries/syn`, are needed for DWBB components:

- standard.sldb
- dw_foundation.sldb
- dw_minpower.sldb[1]

These synthetic libraries are added to the search_path variable in the setup file in `$SYNOPSYS/admin/setup/`. If you alter the search path, make sure it includes the synthetic libraries needed for DWBB.

In Fusion Compiler, the DWBB synthetic libraries are automatically located for you.

In Design Compiler, the synthetic_library variable specifies the synthetic libraries you can use. (The physical location of synthetic library files is specified in the search_path variable.) For example, to use the DWBB synthetic library, set the following:

```
set synthetic_library dw_foundation.sldb
set link_library {* $target_library $synthetic_library}
```

To locate the design library for DWBB, the synthesis tools use the following files:

- Design Compiler:

  ```
  $SYNOPSYS/admin/setup/.synopsys_vss.setup
  ```

- Fusion Compiler:

  ```
  $SYNOPSYS/admin/setup/.synopsys_fc.setup
  ```

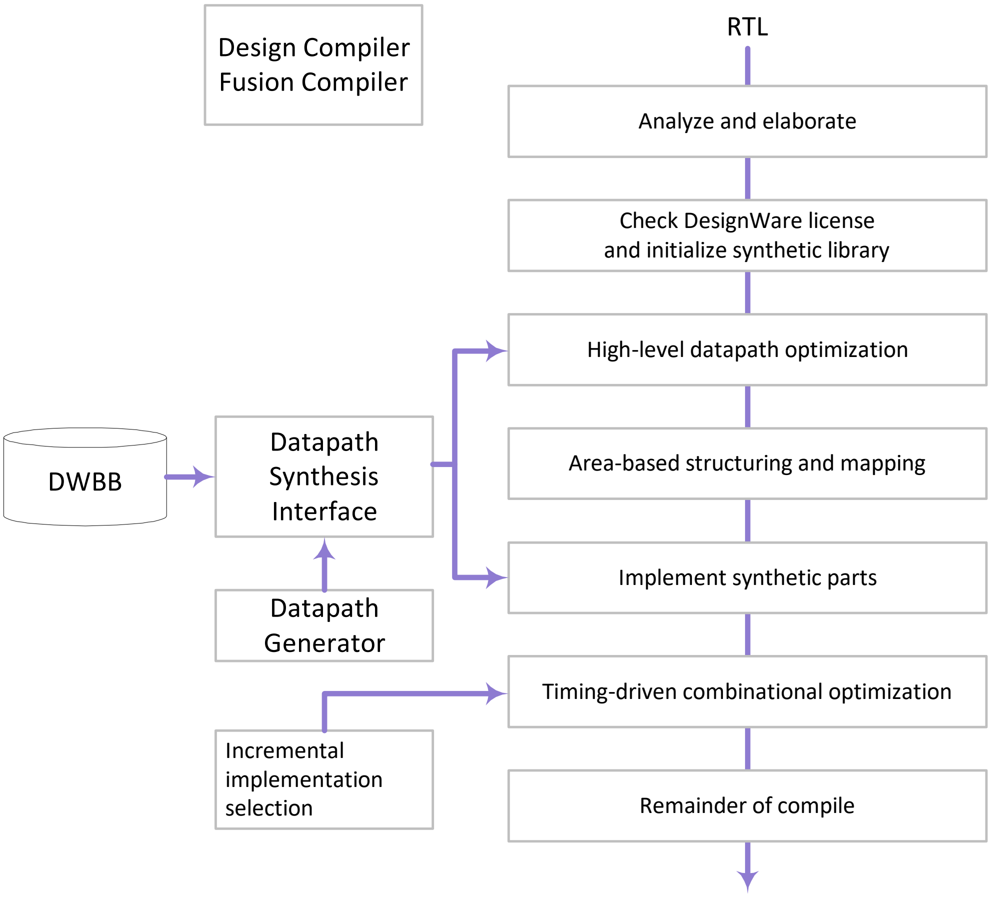
[1] Needed only for versions before P-2019.03 when using minPower components. To use minPower components in pre-P-2019.03 versions, you must have the DesignWare-LP license and specify the necessary synthetic libraries:

```
set synthetic_library {dw_foundation.sldb dw_minpower.sldb}
set link_library {* $target_library $synthetic_library}
```

1.6 Synthesis Optimization Flow

Figure 1-3 shows how Synopsys synthesis tools optimize DWBB components.

Figure 1-3 Optimization Flow


Design Compiler
Fusion Compiler
RTL
Analyze and elaborate
Check DesignWare license and initialize synthetic library
High-level datapath optimization
Area-based structuring and mapping
Implement synthetic parts
Timing-driven combinational optimization
Remainder of compile
DWBB
Datapath Synthesis Interface
Datapath Generator
Incremental implementation selection

In the flow shown in Figure 1-3, the Synopsys synthesis tool performs the following:

1. During elaboration, maps HDL operators to synthetic operators that appear in the generic netlist. For a list of the standard synthetic operators, see Appendix A, "Standard Synthetic Operators".
2. Checks for the DesignWare licenses and initializes the synthetic library.
3. During high-level datapath optimization, manipulates the synthetic operators and applies optimizations such as arithmetic simplifications and resource sharing.
4. During implement synthetic parts, maps synthetic modules to DWBB implementations (architectural representations), and uses the datapath generator to implement arithmetic components and generate the best implementations. In addition, if you are using DC Ultra, the tool performs advanced datapath transformations on the extracted datapath blocks.
5. During incremental implementation selection, explores alternative implementations for each arithmetic component. The tool evaluates and replaces synthetic implementations along the critical path to improve delay cost.

1.7 minPower Overview

The minPower components include low-power datapath generators and reusable intellectual property blocks for many basic datapath functions, and include IP that implement more advanced arithmetic and floating-point functions.

To reduce power usage, minPower components focus on the following:

- Maximizing clock gate insertion structures within components
- Inserting datapath gating structures with enable control
- Pipeline management for pipelined components

Power saving techniques include:

- **Innovative low-power architectures:** Unique low-power datapath structures that can suppress switching and glitch activities.
- **Power costing and switching-activity-aware optimization:** Datapath tree structures and operand encodings are optimized for power savings. This is accomplished by including power and switching activity into the cost function along with other design constraints.
- **Built-in datapath gating logic:** An intelligent scheme isolates the operands from activity when the output of the datapath structure is not required. Isolation logic is inserted and optimized so there is no negative impact to the overall design. Many minPower components have built-in isolation logic that is strategically placed on paths with minimal area overhead and no timing impact.
- **Low-power instantiated IP:** Low-power versions of frequently used instantiated IP are included. The power optimizations result from efficient coding to maximize clock gating, support for datapath gating, and so on. Also, pipelined components reduce register switching for better dynamic power performance.

For sets of minPower components that fit several categories, see Appendix B, “minPower Components By Category”.

Note the following about minPower components:

- All minPower components are used by instantiating them in RTL.
- In version P-2019.03 (and later), all minPower components are defined in the DWBB synthetic library file and require no special license.
- For versions prior to P-2019.03, you need the DesignWare-LP license and you must explicitly include the minPower synthetic library (dw_minpower.sldb), as mentioned in “DesignWare Library for DWBB” on page 13.

2 DWBB Components

Table 2-1 summarizes all DWBB components and provides a link to the detailed datasheet.

Datasheets include coding examples for instantiation, as well as for operator and function inference, where appropriate.

Table 2-1 List of DesignWare Building Block IP

Component	Inference?	Description
Application Specific: Control Logic		
DW_arb_2t	No	Two-Tier Arbiter with Dynamic/Fair-Among-Equal Scheme
DW_arb_dp	No	Arbiter with Dynamic Priority Scheme
DW_arb_fcfs	No	Arbiter with First-Come-First-Served Priority Scheme
DW_arb_rr	No	Arbiter with Round Robin Priority Scheme
DW_arb_sp	No	Arbiter with Static Priority Scheme
Datapath: Arithmetic Components		
DW01_absval	Function	Absolute Value
DW01_add	Operator	Adder
DW01_addsub	Operator	Adder-Subtractor
DW_addsub_dx	No	Duplex Adder/Subtractor with Saturation and Rounding
DW01_ash	Function	Arithmetic Shifter
DW_bin2gray	Function	Binary to Gray Converter
DW01_bsh	Function	Barrel Shifter
DW01_cmp2	Operator	2-Function Comparator
DW01_cmp6	No	6-Function Comparator
DW_cmp_dx	No	Duplex Comparator

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
DW_cntr_gray	No	Gray Code Counter
DW_lp_cntr_up_df	No	Low Power “Up” Counter with Dynamic Terminal Count Flag
DW_lp_cntr_updn_df	No	Low Power “Up/Down” Counter with Dynamic Terminal Count Flag
DW01_csa	No	Carry Save Adder
DW01_dec	Operator	Decrementer
DW_div	Function/Operator	Combinational Divider
DW_div_sat	No	Combinational Divider with Saturation
DW_div_pipe	No	Stallable Pipelined Divider
DW_lp_piped_div	No	Low Power Pipelined Divider
DW_exp2	No	Base 2 Exponential (2a)
DW_gray2bin	No	Gray to Binary Converter
DW01_inc	Operator	Incrementer
DW01_incdec	Operator	Incrementer-Decrementer
DW_inc_gray	Function	Gray Incrementer
DW_inv_sqrt	No	Reciprocal of Square-Root
DW_lbsh	Function	Barrel Shifter with Preferred Left Direction
DW_ln	No	Natural Logarithm (ln(a))
DW_log2	No	Base 2 Logarithm ($\log_2(a)$)
DW02_mac	Function	Multiplier-Accumulator
DW_minmax	Function	Minimum/Maximum Value
DW02_mult	Function*/Operator	Multiplier * Does not support function inference for VHDL
DW02_multp	No	Partial Product Multiplier
DW02_mult_2_stage	No	Two-Stage Pipelined Multiplier
DW02_mult_3_stage	No	Three-Stage Pipelined Multiplier
DW02_mult_4_stage	No	Four-Stage Pipelined Multiplier
DW02_mult_5_stage	No	Five-Stage Pipelined Multiplier
DW02_mult_6_stage	No	Six-Stage Pipelined Multiplier
DW_mult_dx	No	Duplex Multiplier

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
DW_mult_pipe	No	Stallable Pipelined Multiplier
DW_lp_piped_mult	No	Low Power Pipelined Multiplier
DW_lp_multifunc	No	Low Power Fixed-Point Multi-Function Unit
DW_lp_multifunc_DG	No	Low Power Multi-Function Unit with Datapath Gating
DW_norm	No	Normalization for Fractional Input
DW_norm_rnd	No	Normalization and Rounding
DW_piped_mac	No	Pipelined Multiplier-Accumulator
DW_lp_pipe_mgr	No	Low Power Pipeline Manager
DW02_prod_sum	No	Generalized Sum of Products
DW02_prod_sum1	No	Multiplier-Adder
DW_prod_sum_pipe	No	Stallable Pipelined Generalized Sum of Products
DW_lp_piped_prod_sum	No	Low Power Pipelined Sum of Products
DW_rash	Function	Arithmetic Shifter with Preferred Right Direction
DW_rbsh	Function	Barrel Shifter with Preferred Right Direction
DW01_satrnd	No	Arithmetic Saturation and Rounding Logic
DW_shifter	Function	Combined Arithmetic and Barrel Shifter
DW_sla	No	Arithmetic Shifter with Preferred Left Direction (VHDL style)
DW_sra	No	Arithmetic Shifter with Preferred Right Direction (VHDL style)
DW_square	Function	Integer Squarer
DW_squarep	No	Partial Product Integer Squarer
DW_sqrt	Function/Operator*	Combinational Square Root * Does not support operator inference for Verilog
DW_sqrt_pipe	No	Stallable Pipelined Square Root
DW_lp_piped_sqrt	No	Low Power Pipelined Square Root
DW01_sub	Operator	Subtractor
DW02_sum	Function	Vector Adder
DW02_tree	No	Wallace Tree Compressor

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
Datapath: Floating Point		
DW_fp_add	No	Floating Point Adder
DW_fp_add_DG	No	Low Power Floating Point Adder with Datapath Gating
DW_lp_piped_fp_add	No	Low Power Pipelined Floating Point Adder
DW_fp_addsub	No	Floating Point Adder/Subtractor
DW_fp_addsub_DG	No	Low Power Floating Point Adder/Subtractor with Datapath Gating
DW_fp_cmp	No	Floating Point Comparator
DW_fp_cmp_DG	No	Low Power Floating Point Comparator with Datapath Gating
DW_fp_div	No	Floating Point Divider
DW_fp_div_DG	No	Low Power Floating Point Divide with Datapath Gating
DW_lp_piped_fp_div	No	Low Power Pipelined Floating Point Divider
DW_fp_div_seq	No	Floating Point Sequential Divider
DW_fp_dp2	No	2-Term Floating Point Dot-product
DW_fp_dp3	No	3-Term Floating Point Dot-product
DW_fp_dp4	No	4-Term Floating Point Dot-product
DW_fp_exp	No	Floating Point Exponential (e^a)
DW_fp_exp2	No	Floating Point Base-2 Exponential (2a)
DW_fp_flt2i	No	Floating Point to Integer Converter
DW_fp_i2flt	No	Integer to Floating Point Converter
DW_fp_invsqrt	No	Floating Point Reciprocal of Square Root
DW_fp_ln	No	Floating Point Natural Logarithm (ln(a))
DW_fp_log2	No	Floating Point Base 2 Logarithm ($\log_2$(a))
DW_fp_mac	No	Floating Point Multiply-and-Add
DW_fp_mac_DG	No	Low Power Floating Point Multiply-and-Add with Datapath Gating
DW_fp_mult	No	Floating Point Multiplier
DW_fp_mult_DG	No	Low Power Floating Point Multiplier with Datapath Gating
DW_lp_piped_fp_mult	No	Low Power Pipelined Floating Point Multiplier
DW_lp_fp_multifunc	No	Low Power Floating-Point Multi-Function Unit

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
DW_lp_fp_multifunc_DG	No	Low Power Floating Point Multi-Function Unit with Datapath Gating
DW_fp_recip	No	Floating Point Reciprocal (1/a)
DW_fp_recip_DG	No	Low Power Floating Point Reciprocal with Datapath Gating
DW_lp_piped_fp_recip	No	Low Power Pipelined Floating Point Reciprocal
DW_fp_sincos	No	Floating Point Sine and Cosine
DW_fp_sqrt	No	Floating Point Square Root
DW_fp_square	No	Floating Point Square
DW_fp_sub	No	Floating Point Subtractor
DW_fp_sub_DG	No	Low Power Floating Point Subtractor with Datapath Gating
DW_fp_sum3	No	3-input Floating Point Adder
DW_fp_sum3_DG	No	Low Power 3-input Floating Point Adder with Datapath Gating
DW_lp_piped_fp_sum3	No	Low Power Pipelined 3-input Floating Point Adder
DW_fp_sum4	No	4-input Floating Point Adder
Datapath: Sequential		
DW_div_seq	No	Sequential Divider
DW_mult_seq	No	Sequential Multiplier
DW_sqrt_seq	No	Sequential Square Root
Datapath: Trigonometric		
DW_sincos	No	Combinational Sine - Cosine
Data Integrity		
DW_crc_p	No	Universal Parallel (Combinational) CRC Generator/Checker
DW_crc_s	No	Universal Sequential CRC Generator/Checker
DW_ecc	No	Error Checking and Correction
DW_lp_piped_ecc	No	Low Power Pipelined Error Correction Code (ECC)
DW04_par_gen	Function*	Parity Generator and Checker * Does not support function inference for Verilog

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
Data Integrity: Coding		
DW_8b10b_dec	No	8b10b Decoder
DW_8b10b_enc	No	8b10b Encoder
DW_8b10b_unbal	No	8b10b Coding Balance Predictor
Digital Signal Processing (DSP)		
DW_fir	No	High-Speed Digital FIR Filter
DW_fir_seq	No	Sequential Digital FIR Filter Processor
DW_iir_dc	No	High-Speed Digital IIR Filter with Dynamic Coefficients
DW_iir_sc	No	High-Speed Digital IIR Filter with Static Coefficients
DW_dct_2d	No	Two Dimensional Discreet Cosine Transform
DW_thermdec	No	Binary Thermometer Decoder and Enable
Interface: Clock Domain Crossing		
DW_data_qsync_hl	No	Quasi-Synchronous Data Interface for H-to-L Frequency Clocks
DW_data_qsync_lh	No	Quasi-Synchronous Data Interface for L-to-H Frequency Clocks
DW_data_sync	No	Data Bus Synchronizer with Acknowledge
DW_data_sync_na	No	Data Bus Synchronizer without Acknowledge
DW_data_sync_1c	No	Single Clock Filtered Data Bus Synchronizer
DW_gray_sync	No	Gray Coded Synchronizer
DW_pulse_sync	No	Dual Clock Pulse Synchronizer
DW_pulseack_sync	No	Pulse Synchronizer with Acknowledge
DW_reset_sync	No	Reset Sequence Synchronizer
DW_stream_sync	No	Data Stream Synchronizer
DW_sync	No	Single Clock Data Bus Synchronizer
Logic: Combinational Components		
DW01_binenc	Function	Binary Encoder
DW01_decode	Function	Decoder
DW_decode_en	No	Binary Decoder with Enable

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
DW01_mux_any	No	Universal Multiplexer
DW01_prienc	Function	Priority Encoder
DW_lod	Function	Leading One’s Detector
DW_lsd	Function	Leading Signs Detector
DW_lza	Function	Leading Zero’s Anticipator
DW_lzd	Function	Leading Zero’s Detector
DW_pricod	Function	Priority Coder
Logic: Sequential Components		
DW03_bictr_dcnto	No	Up/Down Binary Counter with Dynamic Count-to Flag
DW03_bictr_scnto	No	Up/Down Binary Counter with Static Count-to Flag
DW03_bictr_decode	No	Up/Down Binary Counter with Output Decode
DW_dpll_sd	No	Digital Phase Locked Loop
DW03_lfsr_dcnto	No	LFSR Counter with Dynamic Count-to Flag
DW03_lfsr_scnto	No	LFSR Counter with Static Count-to Flag
DW03_lfsr_load	No	LFSR Counter with Loadable Input
DW03_lfsr_updn	No	LFSR Up/Down Counter
DW03_updn_ctr	No	Up/Down Counter
Memory: FIFO		
DW_asymdata_inbuf	No	Asymmetric Data Input Buffer
DW_asymdata_outbuf	No	Asymmetric Data Output Buffer
DW_asymfifo_s1_df	No	Asymmetric I/O Synchronous (Single Clock) FIFO with Dynamic Flags
DW_asymfifo_s1_sf	No	Asymmetric I/O Synchronous (Single Clock) FIFO with Static Flags
DW_asymfifo_s2_sf	No	Asymmetric Synchronous (Dual-Clock) FIFO with Static Flags
DW_lp_fifo_1c_df	No	Low Power Single-clock FIFO

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
DW_fifo_2c_df	No	Dual independent clock FIFO
DW_fifo_s1_df	No	Synchronous (Single Clock) FIFO with Dynamic Flags
DW_fifo_s1_sf	No	Synchronous (Single Clock) FIFO with Static Flags
DW_fifo_s2_sf	No	Synchronous (Dual-Clock) FIFO with Static Flags
Memory: FIFO Controllers		
DW_asymfifoctl_2c_df	No	Asymmetric Dual-Clock FIFO Controller with Dynamic Flags
DW_asymfifoctl_s1_df	No	Asymmetric I/O Synchronous (Single Clock) FIFO Controller with Dynamic Flags
DW_asymfifoctl_s1_sf	No	Asymmetric I/O Synchronous (Single Clock) FIFO Controller with Static Flags
DW_asymfifoctl_s2_sf	No	Asymmetric Synchronous (Dual-Clock) FIFO Controller with Static Flags
DW_lp_fifoctl_1c_df	No	Low Power Single-clock FIFO Controller with Dynamic Flags
DW_fifoctl_2c_df	No	Dual Clock FIFO Controller with Dynamic Flags
DW_fifoctl_s1_df	No	Synchronous (Single Clock) FIFO Controller with Dynamic Flags
DW_fifoctl_s1_sf	No	Synchronous (Single-Clock) FIFO Controller with Static Flags
DW_fifoctl_s2_sf	No	Synchronous (Dual-Clock) FIFO Controller with Static Flags
Memory: Registers		
DW03_pipe_reg	No	Pipeline Register
DW_pl_reg	No	Pipeline Register
DW03_reg_s_pl	No	Register with Synchronous Enable Reset
DW03_shftreg	No	Shift Register
DW04_shad_reg	No	Shadow and Multi-bit Register

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
Memory: SRAMs		
DW_ram_r_w_s_dff	No	Synchronous Write-Port, Asynchronous Read-Port RAM (Flip-Flop-Based)
DW_ram_r_w_s_lat	No	Synchronous Write-Port, Asynchronous Read-Port RAM (Latch-Based)
DW_ram_2r_2w_s_dff	No	Synch. Dual Write-Port, Async Dual Read-Port RAM (Flip-Flop-Based)
DW_ram_2r_w_s_dff	No	Synchronous Write-Port, Async Dual Read-Port RAM (Flip-Flop-Based)
DW_ram_2r_w_s_lat	No	Synchronous Write-Port, Async Dual Read-Port RAM (Latch-Based)
DW_ram_rw_s_dff	No	Synchronous Single-Port, Read/Write RAM (Flip-Flop-Based)
DW_ram_rw_s_lat	No	Synchronous Single-Port, Read/Write RAM (Latch-Based)
DW_ram_r_w_a_dff	No	Asynchronous Dual-Port RAM (Flip-Flop-Based)
DW_ram_r_w_a_lat	No	Asynchronous Dual-Port RAM (Latch-Based)
DW_ram_2r_w_a_dff	No	Write-Port, Dual-Read-Port RAM (Flip-Flop-Based)
DW_ram_2r_w_a_lat	No	Write-Port, Dual-Read-Port RAM (Latch-Based)
DW_ram_rw_a_dff	No	Asynchronous Single-Port RAM (Flip-Flop-Based)
DW_ram_rw_a_lat	No	Asynchronous Single-Port RAM (Latch-Based)
Memory: Stacks		
DW_stack	No	Synchronous (Single-Clock) Stack
DW_stackctl	No	Synchronous (Single Clock) Stack Controller
Test: JTAG		
DW_tap	No	TAP Controller
DW_tap_uc	No	TAP Controller with USERCODE Support
DW_bc_1	No	Boundary Scan Cell Type BC_1
DW_bc_2	No	Boundary Scan Cell Type BC_2
DW_bc_3	No	Boundary Scan Cell Type BC_3
DW_bc_4	No	Boundary Scan Cell Type BC_4
DW_bc_5	No	Boundary Scan Cell Type BC_5

Table 2-1 List of DesignWare Building Block IP (Continued)

Component	Inference?	Description
DW_bc_7	No	Boundary Scan Cell Type BC_7
DW_bc_8	No	Boundary Scan Cell Type BC_8
DW_bc_9	No	Boundary Scan Cell Type BC_9
DW_bc_10	No	Boundary Scan Cell Type BC_10
Datapath Functions		
DWF_dp_absval	Function	Returns the absolute value (magnitude) of an argument
DWF_dp_blend	Function	Implements an alpha blender or linear interpolator
DWF_dp_count_ones	Function	Performs ones count in argument
DWF_dp_mult_comb	Function	Performs a combined unsigned/signed multiply
DWF_dp_mult_comb_offldet	Function	Performs a combined unsigned/signed multiply and overflow detection
DWF_dp_mult_comb_sat	Function	Performs a combined unsigned/signed multiply and saturation
DWF_dp_mult_ovfldet	Function	Performs a multiplication with overflow detection
DWF_dp_mult_sat	Function	Performs a multiplication and saturation
DWF_dp_rnd	Function	Performs arithmetic rounding
DWF_dp_rndsat	Function	Performs arithmetic rounding and saturation
DWF_dp_sat	Function	Performs arithmetic saturation
DWF_dp_sign_select	Function	Performs sign selection / conditional two's complement
DWF_dp_simd_add	Function	Implements SIMD adder
DWF_dp_simd_addc	Function	Implements SIMD adder with carry
DWF_dp_simd_mult	Function	Implements SIMD multiplier
DWF_dp_sub_abs	Function	Performs a subtraction and returns its absolute value (magnitude)

A
Standard Synthetic Operators

Table A-1 lists the HDL operators that are mapped to synthetic operators.

Table A-1 HDL Operators Mapped to Standard Synthetic Operators

HDL Operator	Synthetic Operator(s)
+	ADD_UNS_OP, ADD_TC_OP
-	SUB_UNS_OP, SUB_TC_OP
*	MULT_UNS_OP, MULT_TC_OP
<	LT_UNS_OP, LT_TC_OP
>	GT_UNS_OP, GT_TC_OP
<=	LEQ_UNS_OP, LEQ_TC_OP
>=	GEQ_UNS_OP, GEQ_TC_OP
==	EQ_UNS_OP, EQ_TC_OP
!=	NE_UNS_OP, NE_TC_OP
if, case	SELECT_OP

B
minPower Components By Category

The categories of minPower components include the following:

- Components with built-in low power features

  Here, power-saving circuitry is brought into your design without special controls or settings. These components include the following:

  DW_lp_cntr_up_df
  DW_lp_cntr_updn_df
  DW_lp_fifo_1c_df
  DW_lp_fifoctl_1c_df
  DW_lp_fp_multifunc
  DW_lp_multifunc

- Components with low-power pipeline control and datapath gating

  Here again, power-saving circuitry is brought into your design without special controls or settings. These components include the following:

  DW_lp_pipe_mgr
  DW_lp_piped_div
  DW_lp_piped_ecc
  DW_lp_piped_mult
  DW_lp_piped_prod_sum
  DW_lp_piped_sqrt
  DW_lp_piped_fp_add
  DW_lp_piped_fp_div
  DW_lp_piped_fp_mult
  DW_lp_piped_fp_recip
  DW_lp_piped_fp_sum3

- Combinational components with datapath gating (with enable control)

  For these components, datapath gating circuitry is implemented and can then be controlled by the `DG_ctrl` port. Details are provided in the component datasheets. These components are as follows:

  DW_fp_add_DG
  DW_fp_addsub_DG
  DW_fp_cmp_DG
  DW_fp_div_DG
  DW_fp_mac_DG
  DW_fp_mult_DG
  DW_fp_recip_DG
  DW_fp_sub_DG
  DW_fp_sum3_DG
  DW_lp_fp_multifunc_DG
  DW_lp_multifunc_DG

- Sequential components with datapath gating

  To achieve minPower benefits for the following components, some extra settings are needed, as explained in the corresponding datasheets:

  DW_8b10b_dec
  DW_8b10b_enc
  DW_div_pipe
  DW_mult_pipe
  DW_piped_mac
  DW_prod_sum_pipe
  DW_sqrt_pipe

- Components with enhanced clock gating

  To achieve minPower benefits for the following components, some extra settings are needed, as explained in the corresponding datasheets:

  DW_arb_dp
  DW_arb_sp
  DW_asymfifoctl_2c_df
  DW_asymfifoctl_s1_df
  DW_asymfifoctl_s1_sf
  DW_cntr_gray
  DW_div_seq
  DW_fifo_2c_df
  DW_fifoctl_2c_df
  DW_fifoctl_s1_df
  DW_fifoctl_s1_sf
  DW_fp_div_seq
  DW_gray_sync
  DW_mult_seq
  DW_sqrt_seq
  DW_stackctl
  DW03_bictr_dcnto
  DW03_bictr_scnto