Debug and Width Parameterization

Shepard Siegel

Atomic Rules LLC

© 2011 Atomic Rules LLC

# Overview

It is often desirable when writing RTL codes intended for reuse to provide a means of specializing the code for a particular application. This document describes two common and independent forms of parameterization and describes their use. The two dimensions of parameterization explored are debug and width.

# Debug

Debug is a logically a Boolean choice that determines whether or not a particular module will be instanced with or without debug logic. The component author specifies which configuration properties and other logics are to be included for debug. The Verilog component author uses a Verilog parameter called HAS\_DEBUG\_LOGIC throughout their code to control the static elaboration, Verilog compile-time Boolean choice of debug or not.

The utility of the debug parameter is that it may left on through development and initial bring up in both functional simulation and hardware. Then, if and when there is a desire to strip the module of non-essential logic, the Debug parameter may be switched off. The module will shrink in area due to the removal of resources. Sometimes Fmax may increase due to a reduced loading on a critical circuit net.

# Width

Width is a selection from a set of choices of the data path width. It is stated for emphasis that width, in this context, in no way implies the selection of data type width. Common data path widths include the integer multiples 1, 2, 4, and 8 of the ubiquitous 32b DWORD. In other words 1, 2, 4, or 8 DWORDs or 4, 8, 16, or 32 Bytes or 32, 64, 128, or 256 bits.

The Verilog component author uses a parameter such as WSI\_S0\_DATAPATH\_WIDTH to specify the width of each port in bits.

The utility of the width parameter is it allows a common adjustment of throughput to be made; provided the component author has provided the capability. The adjustment of width over an 1:8 range generally will adjust the throughput over about that range; and the area of the circuit, generally, over a somewhat smaller range.

# Example Sweep of Width and Debug

The results below show a summary of the outputs automatically generated from sweeping the width and debug for SMAdapter. The complete synthesis reports are checked in under the $OCPI/scripts/reports directory.

Variation: Width:32 Debug:0

Slice Registers: 819

Slice LUTs: 1638

Number of IOs: 426

Block RAM/FIFO: 4

Maximum Frequency: 270

Variation: Width:32 Debug:1

Slice Registers: 1197

Slice LUTs: 2075

Number of IOs: 426

Block RAM/FIFO: 4

Maximum Frequency: 277

Variation: Width:64 Debug:0

Slice Registers: 1099

Slice LUTs: 2101

Number of IOs: 578

Block RAM/FIFO: 6

Maximum Frequency: 278

Variation: Width:64 Debug:1

Slice Registers: 1478

Slice LUTs: 2280

Number of IOs: 578

Block RAM/FIFO: 6

Maximum Frequency: 277

Variation: Width:128 Debug:0

Slice Registers: 1659

Slice LUTs: 2365

Number of IOs: 882

Block RAM/FIFO: 10

Maximum Frequency: 259

Variation: Width:128 Debug:1

Slice Registers: 2034

Slice LUTs: 3080

Number of IOs: 882

Block RAM/FIFO: 10

Maximum Frequency: 258

Variation: Width:256 Debug:0

Slice Registers: 2775

Slice LUTs: 4035

Number of IOs: 1490

Block RAM/FIFO: 18

Maximum Frequency: 222

Variation: Width:256 Debug:1

Slice Registers: 3151

Slice LUTs: 4365

Number of IOs: 1490

Block RAM/FIFO: 18

Maximum Frequency: 22

References

|  |  |
| --- | --- |
| **ID** | **Document Name** |
| 1 | IEEE Standard Verilog Hardware Description Language, March 2001 IEEE Std 1364-2001 |
| 2 | IEEE Standard VHDL Language Reference Manual, IEEE Std 1076-1993, 2002 |