Versatile library

A collection of frequently used modules with synthesis support

Brought to You By ORSoC / OpenCores

Legal Notices and Disclaimers

Copyright Notice

General Disclaimer

The Publisher has strived to be as accurate and complete as possible in the creation of this ebook, notwithstanding the fact that he does not warrant or represent at any time that the contents within are accurate due to the rapidly changing nature of information.

The Publisher will not be responsible for any losses or damages of any kind incurred by the reader whether directly or indirectly arising from the use of the information found in this ebook.

This ebook is not intended for use as a source of legal, business, accounting, financial, or medical advice. All readers are advised to seek services of competent professionals in the legal, business, accounting, finance, and medical fields.

No guarantees of any kind are made. Reader assumes responsibility for use of the information contained herein. The Publisher reserves the right to make changes without notice. The Publisher assumes no responsibility or liability whatsoever on the behalf of the reader of this report.

Distribution Rights

The Publisher grants you the following rights for re-distribution of this ebook.

[YES] Can be given away.
[YES] Can be packaged.
[YES] Can be offered as a bonus.
[NO] Can be edited completely and your name put on it.
[YES] Can be used as web content.
[NO] Can be broken down into smaller articles.
[NO] Can be added to an e-course or auto-responder as content.
[NO] Can be submitted to article directories (even YOURS) IF at least half is rewritten!
[NO] Can be added to paid membership sites.
[NO] Can be added to an ebook/PDF as content.
[NO] Can be offered through auction sites.
[NO] Can sell Resale Rights.
[NO] Can sell Master Resale Rights.
[NO] Can sell Private Label Rights.

Table of Contents

Description

A Verilog HDL library with frequently used functions.

Care have been taken to fully support synthesis of all modules. Different versions exist for optimal synthesis support. Currently ACTEL and ALTERA are supported

The following types of functions are included

1. Clock and reset
   Global buffers, PLL and sync reset
2. Registers
   with clock enable, async set and reset, ...
3. Memories
   RAM, dual port RAM, ...
4. Counters
   Binary, Gray, LFSR
5. Wishbone
   Wishbone system-on-chip bus realted logic

Most modules uses parameter for per instance uniqeness.

Different version of the library exist. All included module names are identical in all library versions. Retargeting a design can be to use a different library.

1. versatile_library.v
   Target independent library
2. versatile_library_actel.v
   ACTEL version with synthesis constraints for synplify / Libero
3. versatile_library_altera.v
   ALTERA version with synthesis constraints for QuartusII

Note: The target independent could also be used with target specific constarints simply by adding the following switch during simulation and or synthesis

+define+ACTEL or
+define+ALTERA

Clock and reset

vl_gbuf

A global buffer to be used to ensure global routing resources for high fanout signals such as clock and reset networks.

Signal	Direction	Comment
i	Input	
o	Output	Global signal

vl_sync_reset

To ensure proper startup behaviour each clock network should have a dedicated synchronous reset signal.

Signal	Direction	Comment
rst_n_i	Input	Async active low reset input, normally driver from external reset and/or PLL lock signal
rst_o	Input	Global synchronous reset signal, active high
clk		Clock source

vl_pll

Most FPGA have PLL functions built-in. This function is one of few which are not possible to describe in HDL and have synthesis support. A functional model can be described.

This library contains the following:

1. vl_pll in versatile_library.v
   a functional model for simulation only
2. vl_pll in versatileLibrary_actel.v
   a wrapper for one or more actual PLL
   a functional model for simulation only
   the later used if define SIM_PLL is applied at simulation time

Simulation model

Parameters

Parameter	Default value	Comment
index	0	Only used for synthesis. Parameter used in PLL wrapper to give a unique name to actual PLL. Parameter is concateneted with pll to give actual name. Example: index=0 => instance name pll0
number_of_clocks	3	Defines number of generated clocks
clk_i_period_time	20	Defines input period time of input clock source, in ns
post_div[0:nu	[1,1,1]	Actual frequency of output clocks is

Parameter	Default value	Comment
mber_of_clocks-1]		frequency of clk_o[i] = 1/clk_i_period_time * mult[i] / div[i]
lock_delay	2000	Time to lock PLL, used in simulation only. Time in ns

Top level signals

Signal	Direction	Comment
clk_i	Input	Clock source
rst_n_i	Input	Async active low reset input used for synchronous reset generation
lock	Output	Signal indicating PLL is in locked state and that clock outputs are valid
clk_o [0:number_of_clocks-1]	Output	Vector with generated clocks

By specifying define SIM_PLL when cmpiling in simulator a functional model is used for PLL.

Appendix A have a detailed usage example of PLLs when used with ACTEL ProASIC3 as target.

Registers

Dff

dff_ce

dff_sr

Appendix A
vl_pll usage example with ACTEL ProASIC3 target

Requirements:

1. clk_i1 external clock, 66.666666667 MHz
   shall generate
   clk0 : 66.666666667 MHz
   clk1 : 25.000 MHz
   clk2 : 48 MHz
2. clk_i2 external 125.000 MHz
   shall generate
   clk3 : 125.000 MHz
3. each clock domain should have a global synchronised reset signal

Recommended Resources

ORSoC – http://www.orsoc.se

ORSoC is a fabless ASIC design & manufacturing services company, providing RTL to ASIC design services and silicon fabrication service. **ORSoC** are specialists building complex system based on the OpenRISC processor platform.

Open Source IP – http://www.opencores.org

Your number one source for open source IP and other FPGA/ASIC related information.