VHDL-2008 Support Library Documentation

Release 1.0.0

David W. Bishop

October 13, 2016

Contents

These packages were designed as a bridge between VHDL-93 and VHDL-2008. I replicated as many of the new functions as possible. Note that all of these packages are design to be synthesizable in VHDL-93. So, as long as you stick to the subsets defined in the “README” files for the various vendors you should be able to take your code through the entire flow.

VHDL-2008 is finally getting some traction. What started out as just a fixed and floating point package got merged into the VHDL LRM. On this page you will find definitions of the functions available in the VHDL-2008 libraries. You will also find VHDL-93 compatible code for those that do not yet have access to VHDL-2008 compilers.

There is a Fixed Point user’s guide and a Floating Point user’s guide. Please check the Fixed and floating point FAQ (NEW!) if you have any quesiton.

The VHDL-2008 packages will eventually be included in your vendor’s environment. In some cases I have found that they may be encrypted due to IEEE rules. The packages available on this page are NOT the released packages, but VHDL-93 versions of those packages, which I published BEFORE the release of the LRM. They are free of copyright restrictions, and may be used for whatever purpose is needed.

CHAPTER 1

VHDL-93 versions of the VHDL-2008 packages

Description	File	User Guide
Additions to std.standard	standard_additions_c.vhdl	user's guide
New package std.env	env_c.vhdl	
Additions to std.textio	standard_textio_additions_c.vhdl	(user's guide)
Additions to ieee.std_logic_1164	std_logic_1164_additions.vhdl	(user's guide)
Additions to ieee.numeric_std	numeric_std_additions.vhdl	(user's guide)
New package numeric_std_unsigned	numeric_std_unsigned_c.vhdl	(user's guide)
New package fixed_float_types	fixed_float_types_c.vhdl	
New package fixed_pkg	fixed_pkg_c.vhdl	(Fixed Point user's guide)
New package float_pkg	float_pkg_c.vhdl	(Floating Point user's guide)

I use this code in most of my designs. Many times I find that I have to modify the code slightly in some tools, so I made this list. Included in the "source code" section for each tool is source code specifically debugged for that particluar tool. Click on the "documentation" link to see what changes I had to make, and how to use this code in the specific tool.

CHAPTER 2

Tool specifc modifications

Vendor / Tool	ZIP File	Notes
Altera Quartus		

Some helpful code to go with these packages

- **fixed_noresize.vhdl** Similar to `fixed_pkg`, however this version uses the same rules that `numeric_std` does for the size of the result. This is done by calling the function in `fixed_pkg` and resizing the result.
- **float_noround_pkg.vhdl** Similar to `float_pkg`, however this version turns off all of the IEEE rounding and overflow, and defaults to a 26 bit floating point number. This package saves off 1/3 of the logic needed for full 32 bit floating point.
- **fixed_synth.vhdl** Synthesis testcase for the fixed point package.
- **test_fixed_synth.vhdl** Testbench for the fixed point synthesis testcase.
- **float_synth.vhdl** Synthesis testcase for the floating point package.
- **test_float_synth.vhdl** Testbench for the floating point synthesis package
- **Matrix Math package** for type REAL which has a user guide (Done in conjunction with IEEE 1076.1 VHDL-AMS)
- Testbenches to verify an implimentation of VHDL-2008.

3.1 New Packages

- numeric_std_unsigned
- fixed_float_types
- fixed_pkg
- float_pkg

<!DOCTYPE HTML PUBLIC “-//IETF//DTD HTML//EN”> <html><head><title>Package numeric_std_unsigned</title> <!– Changed by: , 26-Aug-2009 –>

<meta name=”description” content=”VHDL package numeric_std_unsigned”> <meta name=”author” content=”dbishop@vhdl.org“>

<META NAME=keywords CONTENT=”standard_addtions, VHDL, VHDL-2008”> <META NAME=identifier-url CONTENT=”http://vhdl.org/fphdl/numeric_std_unsigned.html“> <META NAME=distribution CONTENT=Global> <META NAME=rating CONTENT=General> <META NAME=copyright CONTENT=> <META NAME=resource-type CONTENT=document> <META NAME=reply-to CONTENT=”dbishop@vhdl.org“> <META NAME=robots CONTENT=all> <META NAME=language CONTENT=en> <META HTTP-EQUIV=CONTENT-type CONTENT=text/html; charset=iso-8859-1> </head> <body bgcolor=”#FFFFFF” text=”#000000” link=”#0000FF” vlink=”#007F00”> <p> <p>

<a href=”numeric_std_unsigned_c.vhdl”>numeric_std_unsigned.vhdl</a> This package

is a "standardized" verion of "std_logic_unsigned" which appears in many vendor tools. </p> <p>

Use model:<br>

<b>use ieee.std_logic_1164.all;<br> use ieee.numeric_std.all;<br> use ieee_proposed.numeric_std_additions.all;</b> use ieee_proposed.numeric_std_unsigned.all;</b>

</p> <p>VHDL-2008 use model:<br> <b>use ieee.numeric_std_unsigned.all;</b> </p> <p>

Dependencies: ieee.std_logic_1164, ieee.numeric_std

</p> <p> This package treats "std_logic_vector" and "std_ulogic_vector" just like the "unsigned" type in ieee.numeric_std. It has all of the funcitionality of the old "std_logic_unsigned" package with the ability to use "std_ulogic_vector". </p> Please send feedback to <em>David W. Bishop <a href=”mailto:dbishop@vhdl.org“>dbishop@vhdl.org</a></em>. &nbsp

</font> </body> </html>

3.1.1 fixed_float_types

3.1.2 fixed_pkg

3.1.3 float_pkg

3.2 Additions to...

- std.standard
- std.env
- std.textio
- ieee.std_logic_1164
- ieee.numeric_std

3.2.1 std.standard

3.2.2 std.env

3.2.3 std.textio

3.2.4 ieee.std_logic_1164

3.2.5 ieee.numeric_std

3.3 Tool Specific Patches

- altera
- cadence_ncvhdl
- cadence_rc
- leonardo

- modelsim
- modeltech
- synopsys
- synplicity
- vcs
- xilinx
- xilinx_11

3.3.1 Altera Quartus

Tested with Altera Quartus 2 version 9.2

First, enter Quartus, and create a project. Next, add the files `fixed_float_types_c.vhdl`, `fixed_pkg_c.vhdl` and `float_pkg_c.vhdl` to the project. Quartus seems to simply ignore library names, so you can just run from there.

When loaded, hit the "run" arrow. This will take some time to run, but it seems to give a functional result.

I needed to modify no files for this version of Quartus.

3.3.2 Cadence ncvhdl

Tested with ncvhdl 08.20-s004.</P>

To compile: You will need to setup your `cds.lib` and `hdl.var` files, then create an `ieee_proposed` and `worklib` directory. Then you can just run `compile.ncvhdl`.

See the README for an explination of the new functions in these packages.

Note: ncvhdl is stricter on syntax then Modeltech. It actually found a few minor things that I missed. The only error I found was in the `env_c.vhdl` file. It didn't like the 10 hr and 100 hr time resolutions.

3.3.3 Cadence RC

Checked with RC version 11.20

To Compile: just run "compile.rc"

Notes: Had to remove the "match_table". This gave me the error: Error : Incompatible bitwidths in assignment. [CDFG-283] [elaborate]

> : Width of target 'match_logic_table' (1) doesn't match the width of the assigned value (81) in file 'fixed_pkg_c.vhdl' on line 1595. : Use 'set_attribute hdl_vhdl_assign_width_mismatch true /' to allow such assignments with mismatching widths of lhs and rhs.

So, I set the variable, and still had the error.... Replaced it with an "rtl" version of the match table. In synthsis it will behave the same way as rtl, but not in simulaiton.

Version 11.20 hung on the floating point package. I reported this to Cadence, and they said to change the synthesis effort to "medium" and use verion 12.10.

To run these testcases:

rc -f fixed_synth.tcl rc -f float_synth.tcl

3.3.4 Leonardo Spectrum

Leonardo Spectrum (2009a)

Spectrum doesn't like the fixed point "divide" functions, so they are commented out in this release. In order to get a divide to work you will have to write your own divider. I've also had to comment out the "alias" statement.

To run this code: `spectrum -f fixed_synth.tcl`

For some reason Spectrum picks slow multipliers. You may need to use the Modgen multipliers to meet timing.

Floating point functions, but the synthesis is very slow. I would recommand that you use smaller accumulators with this tool.

3.3.5 Modeltech / ModelSim

Tested with Modeltech 6.2e

To compile: source the "compile.mti" script. This will create the IEEE_PROPOSED VHDL library. The ZIP file contains VHDL-93 compatable versions of several of the new packages. Included in this <a href="modelsim.zip">ZIP file</a> are the following packages: <ul>library ieee_proposed; <li>use ieee_proposed.standard_additions.all; – Additions to packages standard.std <li>use ieee_proposed.standard_textio.all; – Additions to packages standard.textio <li>use ieee_proposed.env.all; – New "env" package <li>use ieee_proposed.std_logic_1164_additions.all; – Additions to std_logic_1164 <li>use ieee_proposed.numeric_std_additions.all; – Additions to numeric_std <li>use ieee_proposed.numeric_std_unsigned.all; – Package to do unsigned math with std_logic_vectors, similar to the (sic) ieee.std_logic_unsigned package. <li>use ieee_proposed.math_utiliti_pkg.all; – Types for the fixed and float packages <li>use ieee_proposed.fixed_pkg.all; – Fixed point package <li>use ieee_proposed.float_pkg.all; – Floating point package </ul>

See the README for an explination of the new functions in these packages. You will also want to look at the <a href="http://www.vhdl.org/vhdl-200x/vhdl-200x-ft/packages/Fixed_ug.pdf">Fixed point docuementation</a> and the <a href="http://www.eda.org/vhdl-200x/vhdl-200x-ft/packages/Float_ug.pdf">Floating point docuementation</a>.

Note: When you simulation, you may need to use the `-novopt` option. Otherwise some of the alias won't get seen correctly.

3.3.6 Modeltech / QuestaSim

Tested with Modeltech/Questasim 6.5b

When you compile you will get a warning: (vcom-1246) Range 0 downto 1 is null. To suppress this warning compile with "-suppress 1236" In the MTI install you will find a premapped library called "floatfixlib". This contains an old version of "fixed_pkg" and "float_pkg". The versions in this release are newer.

To compile: source the "compile.mti" script. This will create the IEEE_PROPOSED VHDL library.

See the README for an explination of the new functions in these packages.

3.3.7 Synopsys

Tested with Synopsys 2006.06 sp4, you may be able to use earlier versions, but if you do make sure that you are using the "Presto" compiler.

To load, The easiest way is to add the following lines into your compile script (You need only load the packages you need):

```
define_design_lib ieee_proposed -path ./ieee_proposed
analyze -w ieee_proposed -f vhdl standard_additions_c.vhdl
analyze -w ieee_proposed -f vhdl std_logic_1164_additions.vhdl
analyze -w ieee_proposed -f vhdl numeric_std_additions.vhdl
analyze -w ieee_proposed -f vhdl numeric_std_unsigned_c.vhdl
analyze -w ieee_proposed -f vhdl math_utility_pkg.vhdl
analyze -w ieee_proposed -f vhdl fixed_pkg_c.vhdl
analyze -w ieee_proposed -f vhdl float_pkg_c.vhdl
# analyze -w work -f vhdl float_synth.vhdl
```

Included in this <a href="synopsys.zip">ZIP file</a> are the following packages (only include the ones you need): <ul>library ieee_proposed; <li>use ieee_proposed.standard_additions.all; – Additions to packages standard.std <li>use ieee_proposed.std_logic_1164_additions.all; – Additions to std_logic_1164 <li>use ieee_proposed.numeric_std_additions.all; – Additions to numeric_std <li>use ieee_proposed.numeric_std_unsigned.all; – Package to do unsigned math with std_logic_vectors, similar to the (sic) ieee.std_logic_unsigned package. <li>use ieee_proposed.math_utiliti_pkg.all; – Types for the fixed and float packages <li>use ieee_proposed.fixed_pkg.all; – Fixed point package <li>use ieee_proposed.float_pkg.all; – Floating point package </ul> The "README" file in the ZIP file will give you a list of the new functions.

See the README for an explination of the new functions in these packages. You will also want to look at the <a href="http://www.vhdl.org/fphdl/Fixed_ug.pdf">Fixed point docuementation</a> and the <a href="http://www.eda.org/fphdl/Float_ug.pdf">Floating point docuementation</a>.

Note:

- Synopsys doesn't accept the 1076.6 "– rtl_synthesis off" metacomment. I placed "– pragma synthesis_off" metacomments around these. &nbsp
- When Synopsys sees fixed_pkg'instance_name it dies on elaboration. Replaced with "fixed_pkg" (and "float_pkg") where necessary.
- Same problem with "integer'image" and "real'image" - commented out.
- I had to replace the "match_logic_table" and "no_match_logic_table" with a logical equivilent. This was done in "std_logic_1164_additions" and in "fixed_pkg_c".

3.3.8 Synplify

Tested with Synplify 2009.06

I tried the new "VHDL 2008 (beta)" VHDL option in this tool. There were major issues with function overloading, so don't use it. Instead, include the VHDL-93 package versions. They all work find in this tool (no modification needed).

To load, The easiest way is to add the following lines into your project file (You need only load the packages you need):

```
add_file -vhdl -lib ieee_proposed "standard_additions_c.vhdl"
add_file -vhdl -lib ieee_proposed "std_logic_1164_additions.vhdl"
add_file -vhdl -lib ieee_proposed "numeric_std_additions.vhdl"
add_file -vhdl -lib ieee_proposed "numeric_std_unsigned_c.vhdl"
add_file -vhdl -lib ieee_proposed "math_utility_pkg.vhdl"
add_file -vhdl -lib ieee_proposed "fixed_pkg_c.vhdl"
add_file -vhdl -lib ieee_proposed "float_pkg_c.vhdl"
```

The “README” file in the ZIP file will give you a list of the new functions.

See the README for an explination of the new functions in these packages. You will also want to look at the <a href=”http://www.vhdl.org/fphdl/Fixed_ug.pdf“>Fixed point docuementation</a> and the <a href=”http://www.vhdl.org/fphdl/Float_ug.pdf“>Floating point docuementation</a>.

3.3.9 VCS

Tested with VCS C-2009.06

Textio issues: VCS reports time with capital letters. So instead of “5 ns” you will get “5 NS”. It also doesn’t know what an NBSP is (it’s in the spec, but not in VCS). ‘instance_name works, but you don’t get a seperator at the end. (not reported). The Textio “NBSP” does not appear to be recognized by VCS. (not reported)

This version does not seem to like the floating point packages, however previous versions did.

3.3.10 Xilinx ISE

Tested with Xilinx M9.1i sp1

Go through “new project” and add the files to the project. Go to the “Source Libraries” tab under “Sources” Click on a blank area of that window, Select “New Source” Select “VHDL Library”, enter the name “ieee_proposed” and hit “Finish”. Select “math_utility_pkg.vhdl”, “fixed_pkg_c.vhdl”, and “float_pkg_c.vhdl” one at a time and move them into “ieee_proposed” library.

Click on “Synthesis” and select “Run”.

Things Xilinx m9.1i didn’t like about these packages: 1) Didn’t like the “alias” statements on functions and type. Had to comment these out or replace them with subtypes. 2) ‘instance_name - Doesn’t like this attribute, replace with package name. 3) “to_stdlogicvector(to_suv(arg))” shows as a type conversion error, replace with casting “std_logic_vector()” 4) “to_stdulogicvector(arg)” shows as a type conversion error, replace with casting “std_ulogic_vector()”

After fixing everything, it gave me the error: INTERNAL_ERROR:Xst:cmain.c:3111:1.8.6.1 - To resolve this error, please consult the Answers Database and other online resources at http://support.xilinx.com

Which according to Xilinx has to do with the loops in my synthesis testcase, and not the packages. This is a “use at your own risk” one I guess. I would recommend Synplicity, which seems to work much better. Xilinx has said that they plan to fix this problem in version 9.2.

3.3.11 Xilinx ISE

Tested with Xilinx M11.2i

Go through “new project” and add the files to the project. When the “Adding Source files...” windows comes up, change the library for “fixed_float_types.vhdl”, “fixed_pkg_c.vhdl” and “float_pkg_c.vhdl” to “ieee_porposed”

Click on “Synthesis” and select “Run”.

I had to really fight to get these packages to synthesize in this tool. I could not check everything, so “use at your own risk”. For a real Xilinx project I would use still Synplicity or Leonardo. However, I was able to get both my fixed and floating point testcases (with some modification) to place and route. This is a major improvement over M9.1i

Things Xilinx m11.1i didn't like about these packages: 1) 'instance_name showed as a syntax error, replace with package name 2) "to_stdlogicvector(to_suv(arg))" shows as a type conversion error, replace with casting "std_logic_vector()" 3) Did not like any of the fixed point division routines, had to comment them out. 4) Had to comment out the "?=" routines, XST could not deal with that syntax.