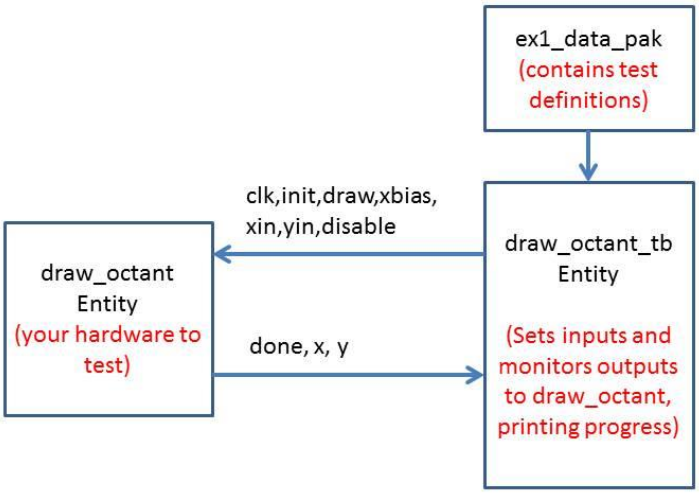
TESTING GUIDE

Understanding the testbench

As you can see from the testbench file, the draw_octant_tb entity declaration is empty. The testbench does not have input and output ports because it is not hardware. The testbench has a package which contains data to drive the test. Every set of data contains an instruction, values of **x, y, xin, yin** and **xbias**. (Exactly what **xbias** does is complex – it is tested in test #5 and #6 where a line is drawn that changes slightly according to the value of xbias).
The testbench generates clock signals, and the clock is connected to *draw_octant*. Every clock cycle, **init**, **draw**, **xbias**, **xin**, **yin**, **disable** are fed into *draw_octant*, then **done**, **x** and **y** generated from *draw_octant* accordingly are fed back into the testbench. The testbench will compare **x**, **y** and **done** generated by *draw_octant* with x, y and the instruction in the data set.



The package **ex1_data_pak** (automatically generated, see later) describes the correct behavior of your hardware.

When the hardware is all correct the testbench will run to the end of the code and then stop with a VHDL “failure” assert and a message:

```
"All tests finished OK, terminating with failure ASSERT."
```

. This means success not failure!

Debugging draw_octant

Debug with **ex1_data_pak** which contains a sequence of tests, starting with simple ones. The tests used (in terms oflines being drawn) can be found from the list of data items in the draw function call at the end of ex1_data_gen. Each line of this list represents a different line to be drawn. For example (note coordinates are always (x,y):

((0,0),(1,2),0), # first line of given ex1_data_pak tests, start: (0,0) end : (1,2) xbias 0.

Note that the testbench also generates a **disable** signal. The **disable** signal controls operation of draw_octant. In the testbench it is set to be high in the 3rd, 4th and 5th clock cycle. Your hardware is supposed to be 'frozen' when the disable signal is high. Thus whatever the waveform on disable the final result will be the same but during clock cycles when the hardware is disabled nothing changes. Check the wave window for this functionality first.

Generating specific tests

You don't need anything other than the given (long) test. However you can generate your only tests easily using the test generator. This is a python script which will run under python 2.7, If not familiar with python, and wanting to use this on a windows system:

(1) Install python 2.7 32 bit standard distribution (google python 2.7)
(2) Install pyscripter (an easy python IDE)
(3) Open ex1_data_gen_2013.py in pyscripter and run it – it will generate an output file corresponding to the tests set in the python file draw command. You can change the file name and the tests (default ex1_data_pak and all tests).