

VHDL Team Project Vector Display Processor

Hardware Files (under hardware subdirectory)

- **Config_pack.vhd.** Package of user-configurable constants. You may alter this e.g. to change testbench clock frequency. However don't put your own constants in here, this package will be replaced when your code is tested for assessment.
- **Project_pack.vhd.** Use this for your own constants, types, functions, procedures. The given record **db_2_rcd** is used in the sample code for part of the DB->RCD interface. You may use this, change it or ignore it.
- **Vdp.vhd.** Skeleton VDP entity & architecture . Replace the architecture with your own code but DON'T ALTER the entity. You will need to write code that connects together DB and RCB blocks, implements the **finish** hardware, etc.
- **Db.vhd.** Skeleton for sample draw block. You may change this as you like.
- **Rcb.vhd.** Skeleton for sample ram control block. You may change this as you like.

Testbench Files

- **Utility_pack.vhd.** DON'T ALTER. Conversion function definitions. DONT use this in synthesisable code.
- **Vdp_pack.vhd.** DON'T ALTER. Example definitions for a VDP testbench which you are free to use in your own code, or to copy and modify in your own package as required.
- **Vdp_testbench.vhd.** Working testbench for the VDP. DON'T ALTER
- **Vram.vhd.** Behavioural video ram entity used in vdp_testbench. DON'T ALTER this entity: use it as required in your testbenches.
- **sample_host_commands.txt.** Text file containing sequence of draw commands that perform a simple test of the VDP commands. You should change/add to these as necessary.
- **Vtrun.do.** Modelsim "do" script file to automate startup of VHDL simulation. See Modelsim tutorial

Simulated (behavioural) Hardware

- **vdp_behav.vhd.** DON'T ALTER. This entity is a working behavioural version of the VDP, for testing purposes. Your VDP should replace this but use less RAM cycles (because of ram_control_block) and be fully synthesisable. You don't need to examine this file to do the project – it was written to allow the testbench components to be tested. **This entity will not form part of your submitted code and is included only for information and so you can test the testbench.**

Using the Testbench

NB - ensure vdb_pack, vdp_testbench, vdp_behav, db_behav, rcb_behav are all compiling with 1993 VHDL. (right click on file under project windows->properties->VHDL-> 1993). If you do not do this they will not compile!

To test the testbench

- Compile: utility_pack, config_pack, project_pack, vdp_pack, vram, vdp_behav, vdp_testbench

- Ensure that the required testbench control command file `full_commands.txt` (or other as determined by `config_pack`) is put in your Modelsim project directory.
- Run simulation of `vdp_testbench`. This should print out a correct result and terminate (failure ASSERT) saying that all tests have been passed.

To test a full design

- Replace `vdp_behav` vdp entity with your own hardware top-level vdp entity and also any necessary sub-entities (e.g. `db`, `rcb`, `ex1-ex4`) and your `project_pack`.
- **Make very sure you have compiled the new version of vdp, otherwise vdp_behav will still be simulated and pass anyway**
- Run the testbench simulation again. A correct VDP should pass the test as above.
- The testbench will print out an error message if problems are encountered:
 - VRAM address/data timing errors will result in ASSERT failures from the vram component.
 - Pixel draw errors will result in a non-zero error count. You can see what is wrong from the vram contents screen printout.

Modifying the testbench

You will want to modify the testbench to test your code:

- Alter **`sample_host_commands.txt`** to make your own tests - the testbench will work with any file. Syntax: each line represents a host bus command, or if started with - a comment. See lines 121-144 of `vdp_testbench` which reads the first two characters in each line and determine what command to send, for the meaning of the first two chars. The rest of the line specifies X,Y coordinates for the command, if required. You can run your command file under a different name by altering `config_pack`.
- Test your new command file by running the testbench with `vdp_behav` to see what should be printed out for a given set of commands.
- Use `config.vhd` to adjust the clock frequency (this should make no difference to correctness) as you wish. For post-synthesis testing you may need a lower clock frequency to ensure post-synthesis code delays do not cause problems.
- Adjust the RAM parameters to make RAM slower if you have implemented RCB which can cope with this. Your RCB should (in principle) use the RAM parameters to determine number of cycles needed for RAM access. However this is not required in the basic deliverable.