COMP22111: Processor Microarchitecture

Tool chain using MU0 as a design example

The following steps follow the stages detailed in the toolchain chapter. Here the MU0 is chosen as a design example. If you would like further information on MU0, see your COMP12111 notes.

Step 1 – Editing & Compile the MU0 design

Start the Cadence environment using 'start_cadence 22111' to open up the Cadence NCLaunch window. Open the MU0 module from the left-hand pane by selecting the MU0.v design in the src directory, then right-click -> Edit. The module is written in Verilog and describes the whole processor; have a look through the example.

To compile the design, ensure that your Work Library is 'mu0' (File->Set Work Library->mu0) – the 'hard hat' should be next to mu0 in the right hand pane. Select your design in the NCLaunch window left-hand pane and compile it by selecting the "Launch Verilog Compiler with current selection" button on the toolbar (the button with VLOG on it and some cogs), as shown in Figure 1.

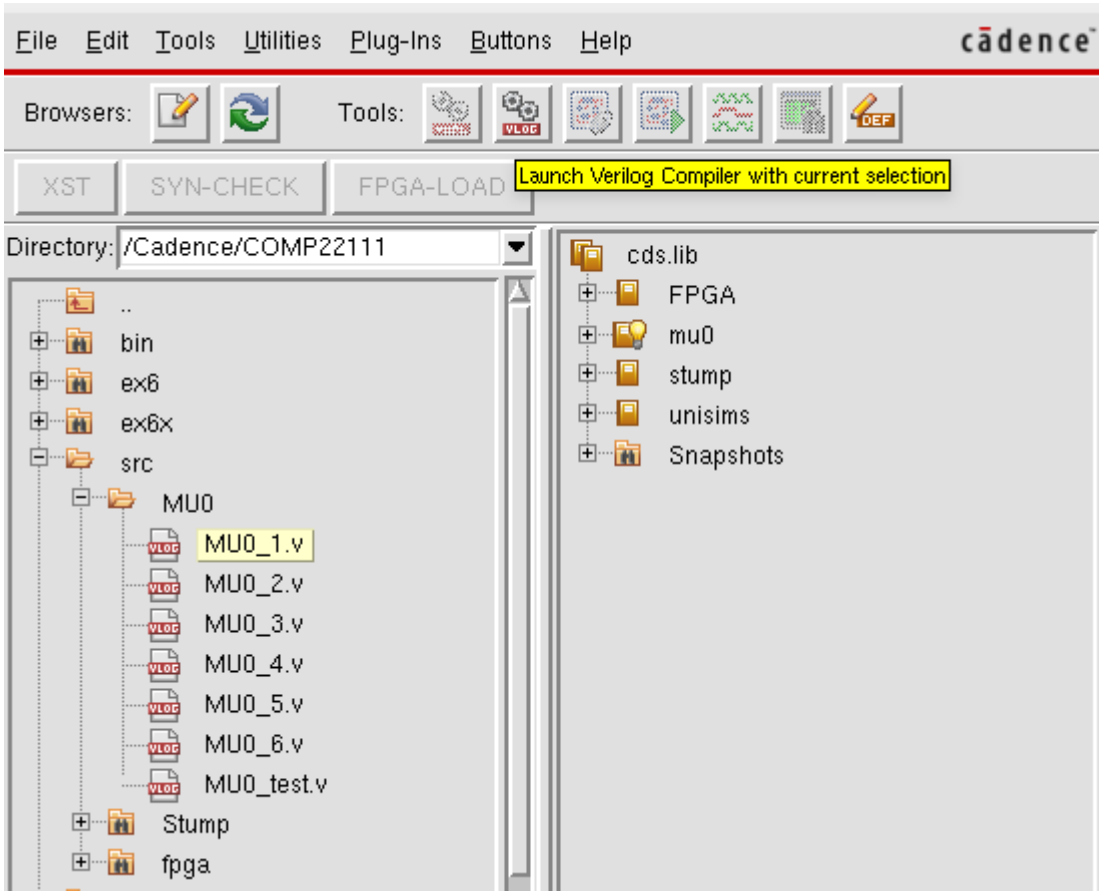

Figure 1: Compiling the MU0 design

It should compile without any errors and an instance should appear in the working library (here mu0) in the right-hand pane.

Step 2 – Simulate the MU0 design

To simulate your design the module should be instantiated in another (testing) module which will provide the stimulus to test the module under test, for MU0 this is given to you and is called 'MU0_test.v'. Open MU0_test.v to familiarise yourself with the structure of the test file. You will need to compile this design in the same way you compiled MU0.v, and an instance should then appear in the working directory if there are no errors in the design. Once MU0_test.v has been compiled with no errors then the design needs elaborating (linking), to do so select MU0_test in the mu0 working directory in the right-hand pane and select the "Launch Elaborator with current selection" button on the toolbar (the button with a paperclip on it), as illustrated in the Figure 2.

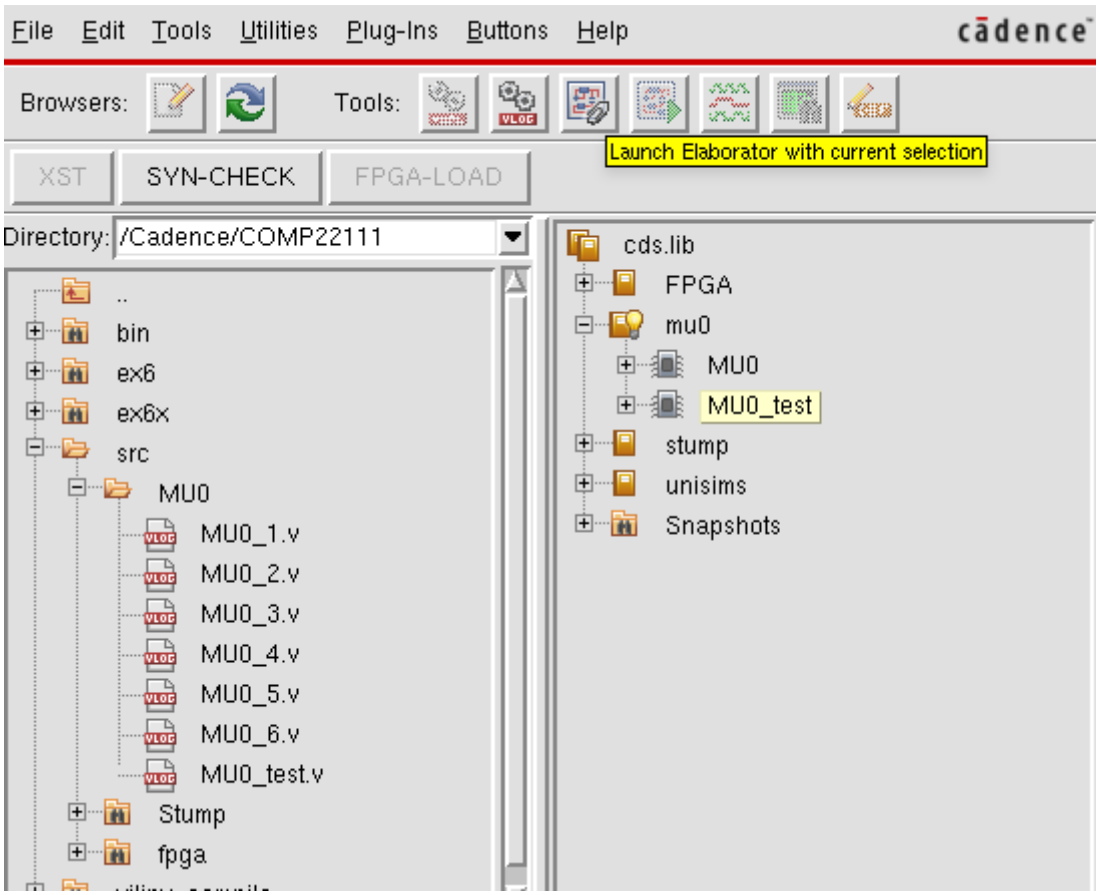

Figure 2: Linking the MU0 design

MU0_test.v models a simple memory that is preloaded ('initial'ised) with some software. You can change the program that is loaded by uncommenting the appropriate line. It also 'prints' some trace information.

Once linked, you will notice that a MU0_test design has appeared in the "Snapshots" directory. To start the simulation, select the appropriate MU0_test 'Snapshot' and select "Launch Simulator with current selection" from the toolbar (the button with the 'play' symbol in it), as illustrated in Figure 3.

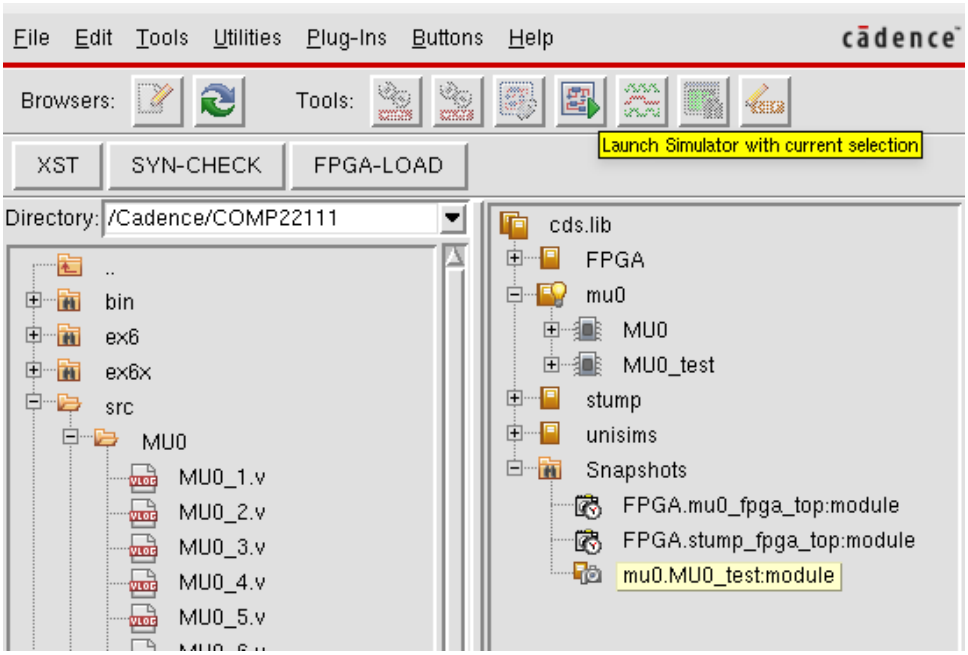

Figure 3: Simulating the MU0 design

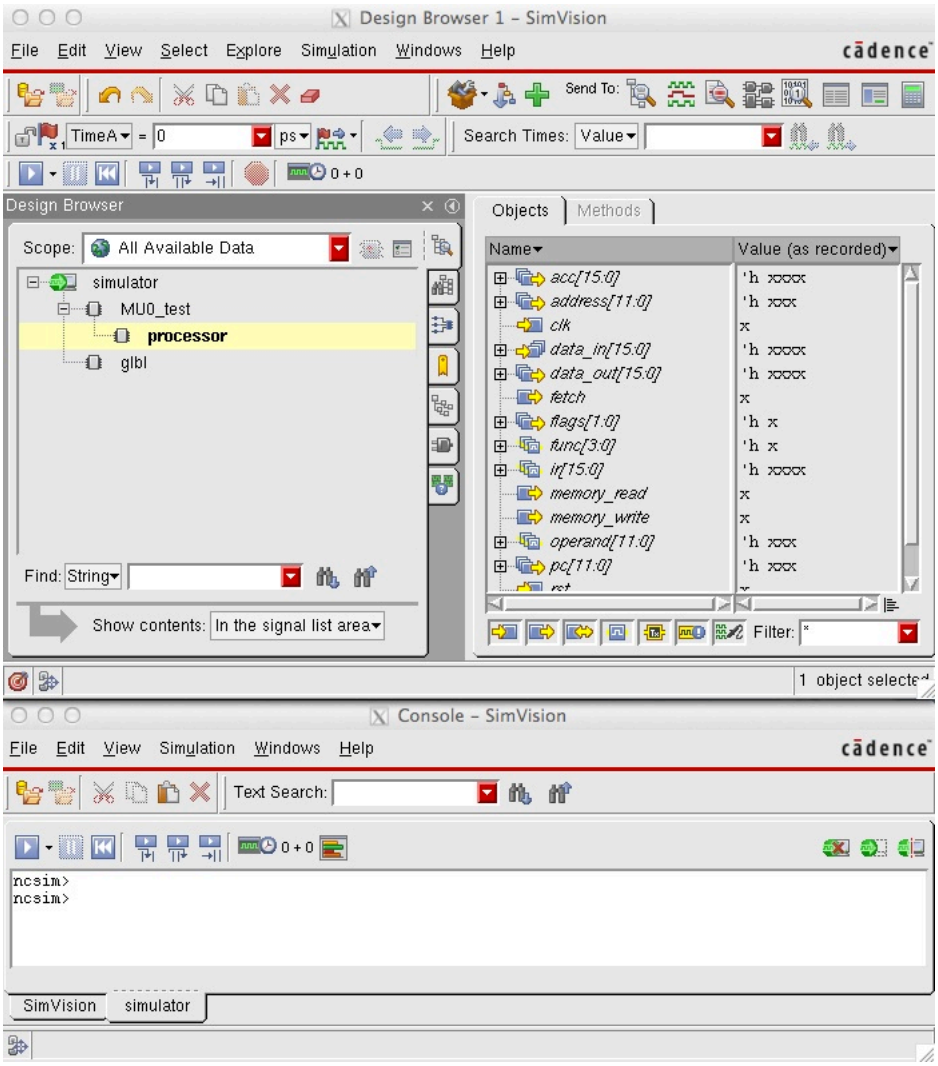

Figure 4: SimVision Simulation Windows

This will open the Simvision windows as illustrated in Figure 4.

In the Design Browser window if you expand MU0_test in the left-hand pane you can see a module “processor”, which is the main instance of the MU0 processor. Selecting this will open the signals that you can view in the left-hand pane (as illustrated in figure A4). Select the signals you wish to view. Individual buses can be selected in the right-hand panel. Send these to a waveform viewer using the button with the waveforms on it (hover for tooltips if unsure); the first such action should create the window.

Different selections may be made using the ‘Select’ menu or using ctrl or shift in combination with the mouse, ctrl-A will select all the signals in a chosen module. It is recommended that you display at least the clock, reset and processor state {PC, Acc}; other signals may also aid understanding.

When happy with your signal set (you can change it and rerun as needed) run the simulation using the ‘play’ button in the Waveform window, as illustrated in Figure 5.

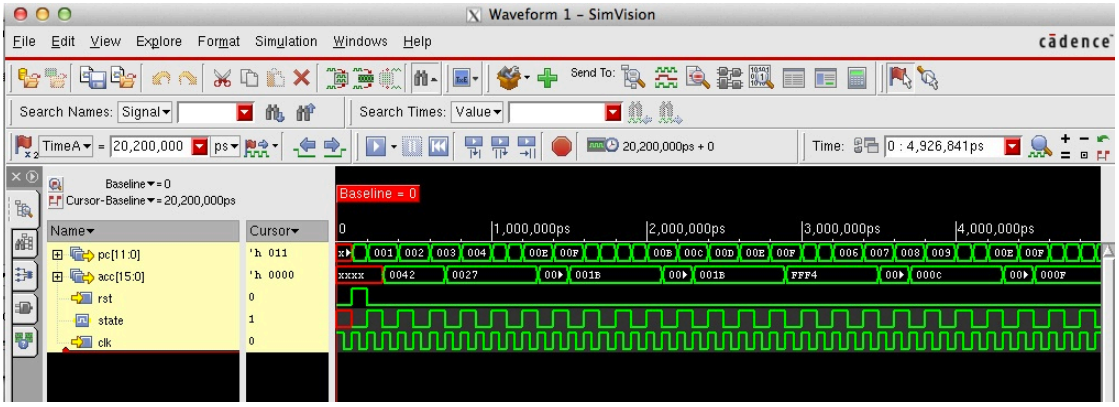

Figure 5: Waveform Viewer

In this case the simulation will run for a defined time and then halt. You can observe the signal evolution by zooming and scrolling. The behaviour of the PC and the accumulator should be visible and can be compared with the expected behaviour. If you wish to reference your simulation to sample code then there is a library in /opt/info/courses/COMP22111/MU0_src/- where the ‘list’ file, which shows addresses and op-codes, may be useful.

Output which is generated by ‘$display()’ functions will appear in ~/Cadence/COMP22111/ncsim.log.

Step 3 – Synthesize the MU0 design

Now synthesize the design for the FPGA. The potential problem here is that the processor is not useful in isolation - it needs to be part of a computer, with memory etc. We have provided the environment for this in the cell ‘MU0_fpga_top.v’ in which MU0 has been instantiated. If the contained MU0.v has simulated correctly it should synthesize without problems since we have already tested this environment and have introduced no deliberate errors.

Synthesize the design by selecting its instance “FPGA.mu0_fpga_top.module” in the Snapshot directory, as illustrated in Figure 6.

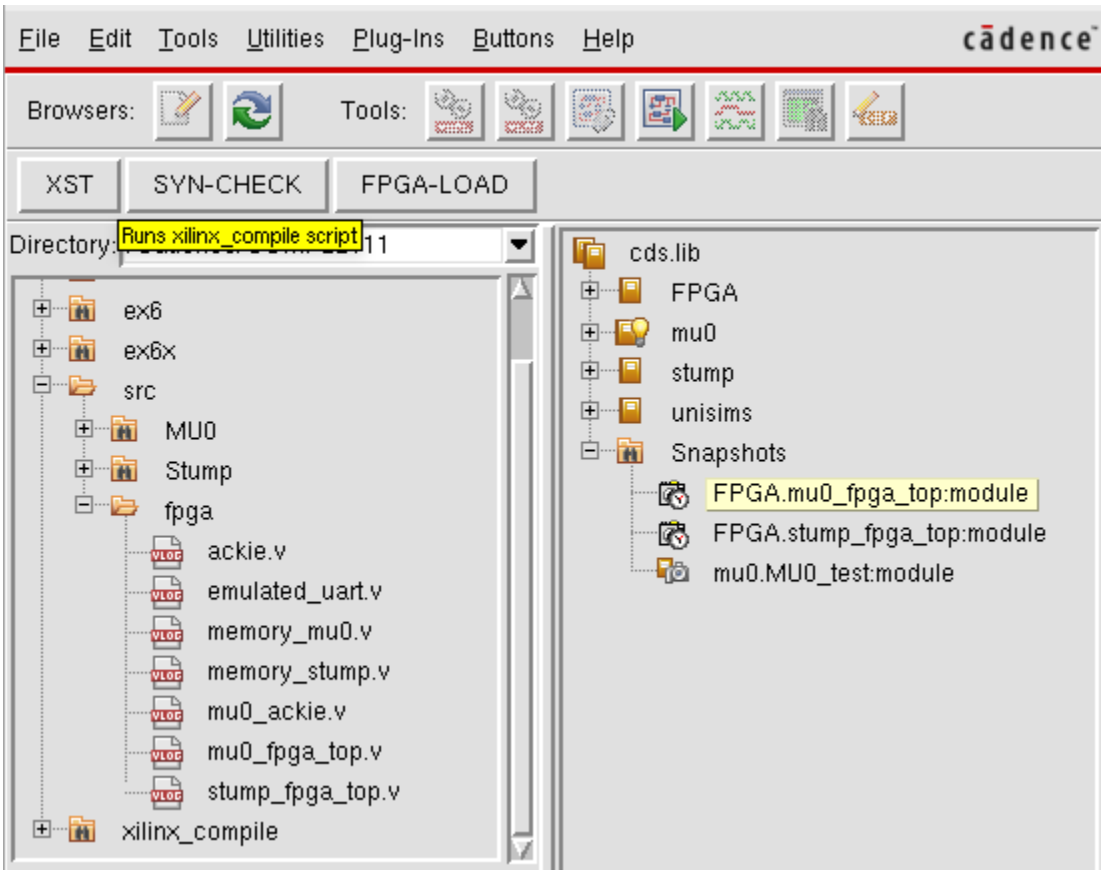

Figure 6: MU0 design for Synthesis

Then press the XST (Xilinx Synthesis Tool) button on the NCLaunch window. The design should synthesise, with reports scrolling through the console pane, without errors (the log files are retained in /home/Cadence/COMP22111/xilinx_compile/ xst.log). If this has worked there should be a MU0_fpga_top.bit file generated in your ~/Cadence/COMP22111/xilinx_compile/ directory.

Step 4 – Download the MU0 design

The design should synthesize without problems. If something did go wrong you have probably rushed and skipped some step above. This will cost you time. If you were making a custom chip at this point it would have cost your company a lot of money too.

The synthesis result is stored as the file mu0_fpga_top.bit in the directory ~/Cadence/COMP22111/xilinx_compile/. Now it needs loading onto the experimental boards in the laboratory.

Ensure you have an FPGA board connected (and switched on). To download your design to the board select theFPGA.mu0_fpga_top:module from the Snapshots and press the FPGA-LOAD button. Check the log to make sure the download is successful. If successful start the Perentie debugger using the script 'start_perentie' at a command prompt. This is an environment like Komodo - which you saw last year - but it will let you view the operation of the design on the FPGA. Note: this is a new tool prepared specifically for this laboratory; please report any faults so that we can fix them as soon as possible.

An initial dialogue box is displayed, as shown in Figure 7, to select the Debugger target. Select the "DEPP" tab and then connect.

When this stage is complete Perentie should recognise you have an MU0 processor and should give you a view of the working design, as illustrated in Figure 8.

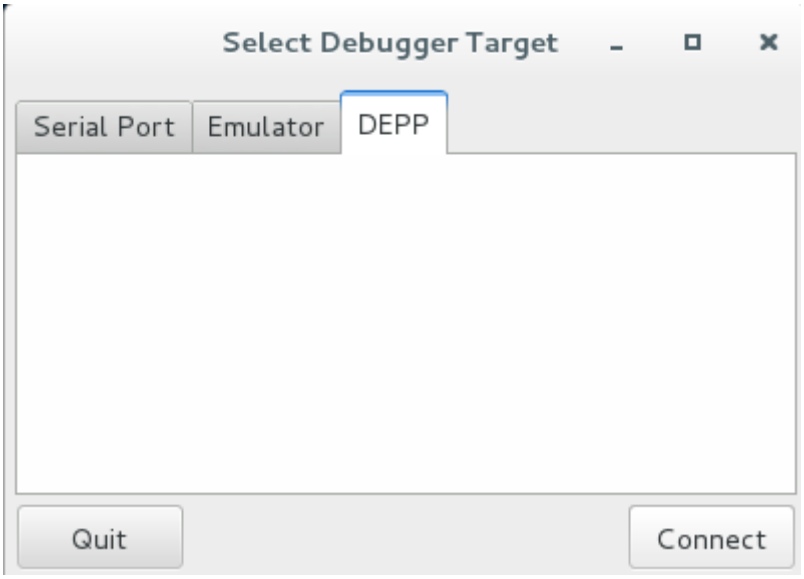

Figure 7: Perentie Window on startup

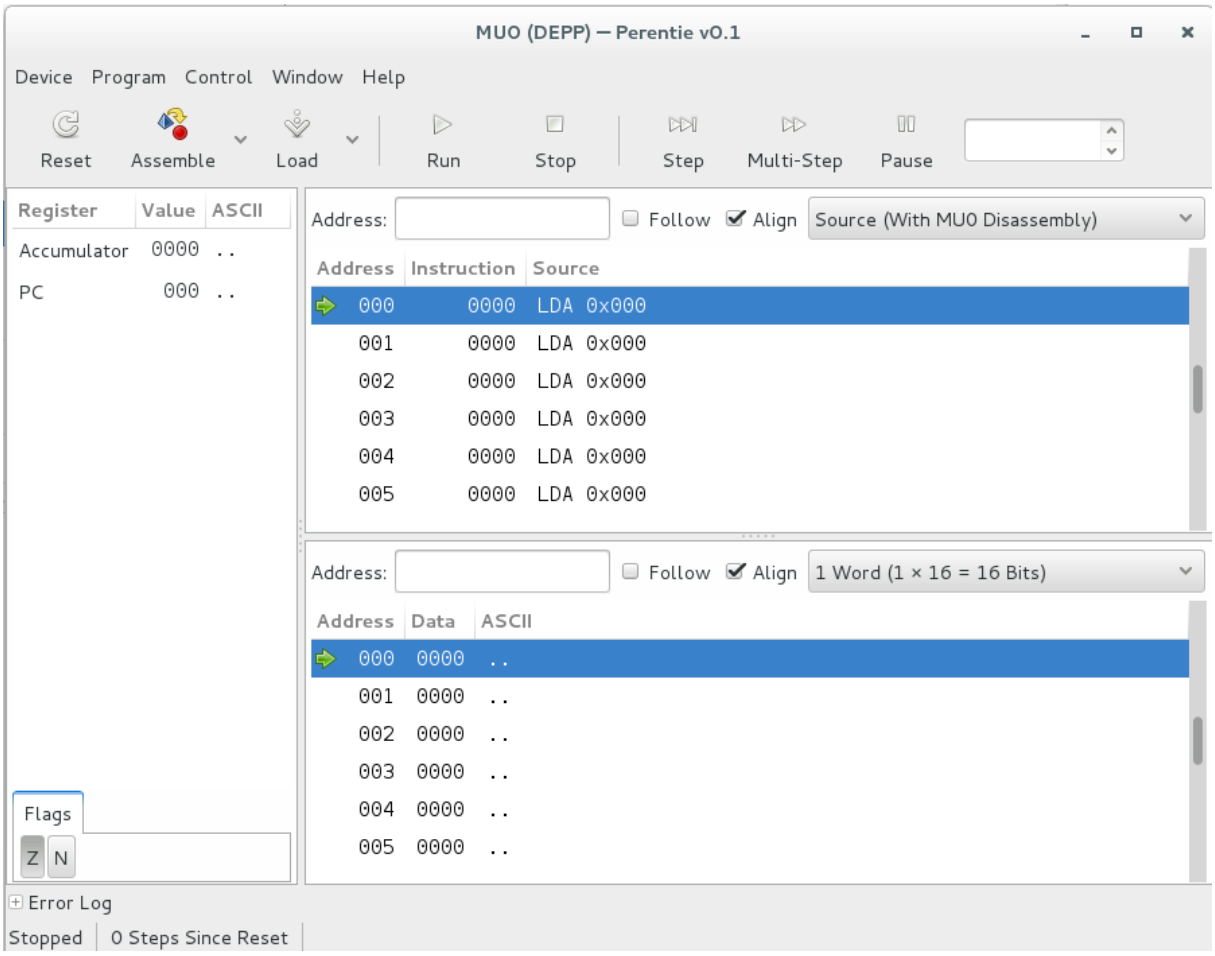

Figure 8: Perentie working with a processor design

A processor also needs software. A small selection of short MU0 programmes have been created in /opt/info/courses/COMP22111/MU0_src/, and they have been assembled to give list (.lst) files which can be loaded into Perentie's source debugger. The sources are commented to say a little about what they do. Select ‘Load’ and navigate to the MU0_src directory (from the “File System” root directory). Select a file and ‘Open’ to load the memory.

You can single step or run the processor from here for the programmer's view of execution, observing changes in registers and memory. If you wish to restart you can also ‘Reset' to initialise the processor state.

You should now have an MU0 design working on the experimental board running one of the sample programs.