

## Table of Contents

Purpose:.....	1
1. Getting Started: .....	1
1.1. If you have not already created the area: .....	2
1.2. If you have already created the area: .....	2
1.3. Start the VHDL simulation tool using the command: .....	2
2. Exercise 1 .....	2
1.4. Setting up for Simulation.....	3
1.4.1. Compiling my_and.vhd.....	3
1.4.2. Compiling the testbench_my_and.vhd.....	3
1.4.3. Elaborating the testbench_my_and worklib entry .....	4
1.5. Simulating the testbench_my_and snapshot.....	5
1.5.1. Selecting Signals and Running a Simulation .....	6
1.5.1.1. Selecting Signals .....	6
1.6. Congratulations.....	10
1.7. Using the other my_and architectures .....	10
3. Additional Exercise.....	11
4. Hints.....	11

### ***Purpose:***

This document is intended to cover the steps required to compile, simulate and test a basic gate such as the AND gate using the VHDL file and VHDL test bench provided in the ECE2044 Walkthroughs area.

The intention is for this to also to function as a aide memoire for you during the next couple of weeks.

## **1. Getting Started:**

As per the instructions last week:

- Login to the Linux CAD system
- Open a Terminal Window
- Open an etransport window
- Select the AMI -> BEng2 -> ECE2044 -> Walkthroughs area

### 1.1. If you have not already created the area:

Select area and use the “Create Module Area” from the etransport form.

### 1.2. If you have already created the area:

Use the “Enter Module Area” to open up a configured terminal in the ECE2044 walkthroughs area.

### 1.3. Start the VHDL simulation tool using the command:

- **nclaunch**

If you have previously configured the VHDL tool then it should start automatically in the correct area. If you did not do this, then follow the instructions from last week. Which hopefully you have brought with you?

You should see the files displayed in the tool and it should look something like figure 1.

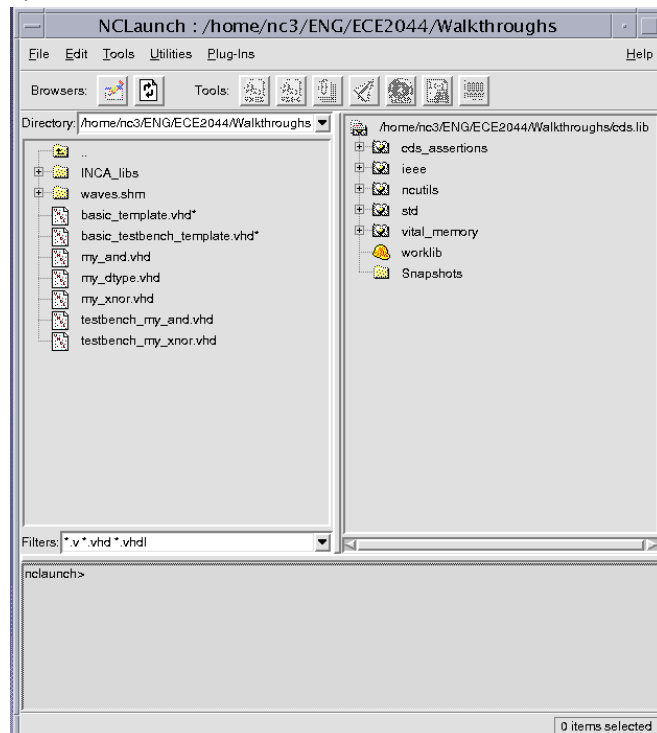


Figure 1: Basic NCVHDL Window

## 2. Exercise 1

We are going to open and examine the my\_and.vhd and testbench\_my\_and.vhd files. We do this by clicking on them once to select then using the right hand side mouse button to open the context sensitive entry. We can then select “Edit” from the available options. Having reviewed the entity and architecture as your assigned work last week it should be relatively familiar.

Resize the two windows until you are comfortable with their size.

## 1.4. Setting up for Simulation

### 1.4.1. Compiling my\_and.vhd

First we need to successfully compile the my\_and.vhd file.

To do this we can left click twice on the my\_and.vhd file in the file browser.

This will compile the file.

Alternatively if you select the file and right click the options form will be displayed.

Select the “compile” option. Whilst doing this make sure that the VHDL 93 options are selected!

Check the window at the base of the NCVHDL tool for any compiler error messages. If you are successful a new entry should appear in the worklib. Similar to the example in figure 2.

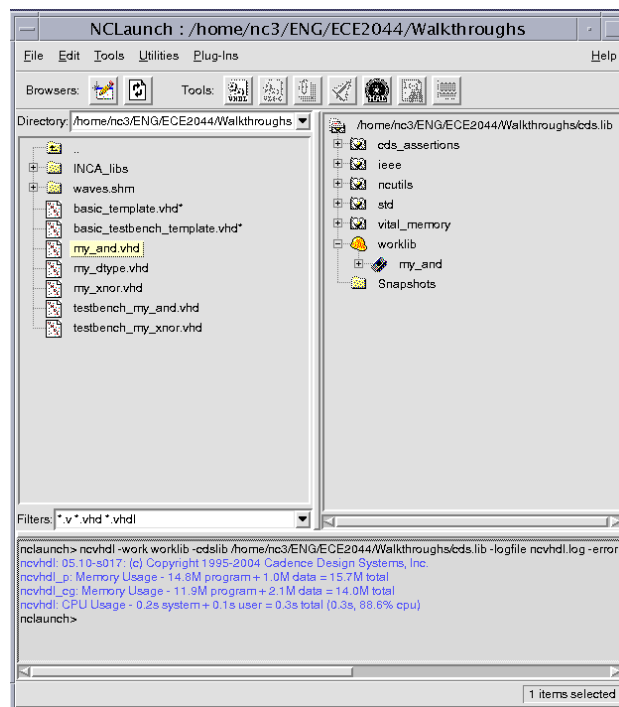


Figure 2 : Worklib example

### 1.4.2. Compiling the testbench\_my\_and.vhd

Now we need to compile the testbench file for the and gate. The one used in this case is called testbench\_my\_and.vhd file.

Repeat the procedure used on the my\_and.vhd file to compile the testbench\_my\_and.vhd file.

Check the window at the base of the window for error messages. The result should resemble figure 3.

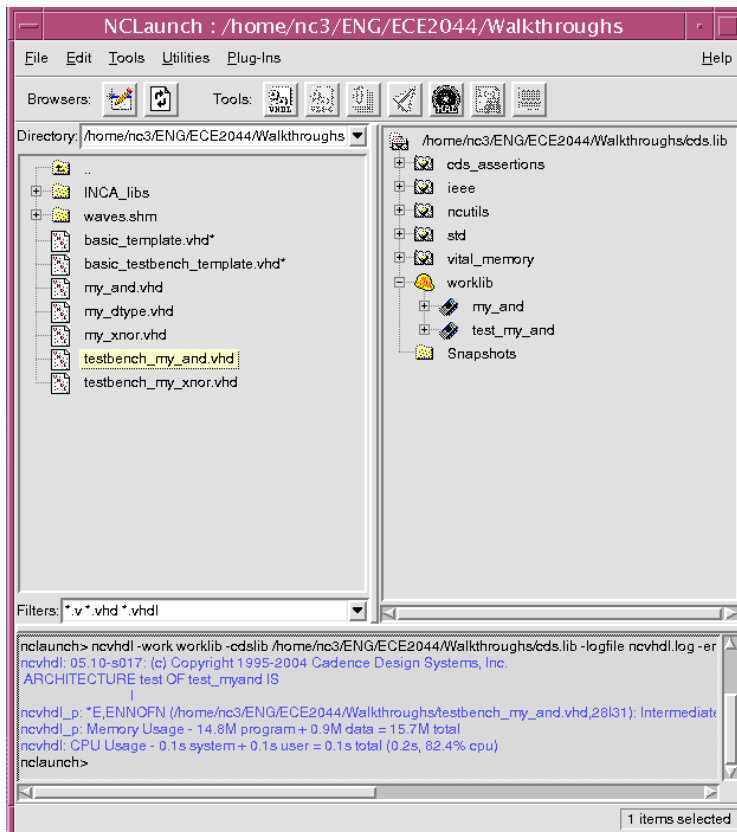


Figure 3: Worklib with my\_and and test\_my\_and files successfully compiled

### 1.4.3. Elaborating the testbench\_my\_and worklib entry

We now need to use “NCElab” to elaborate the test bench file test\_my\_and. We can only “elaborate” files that have compiled successfully. Hence we need to use the version of testbench\_my\_and in the worklib. Single click on the testbench\_my\_and entry in the worklib.

Use the right hand mouse button to bring up the context sensitive menu and select “Elaborate”.

This will bring up the elaboration form, as shown in figure 4, “OK” this form and continue.

Hint:

1. Note that you can elaborate the “my\_and” file successfully. However if you think about it the my\_and file has nothing driving any of the signals. Hence any simulation will show no activity. Again a common error made by students.
2. If you have a pre-existing successfully compiled file and make changes to the design so the compile fails then the system will most likely use the “successful” compilation when you run a simulation. Causing confusion and a great waste of time!

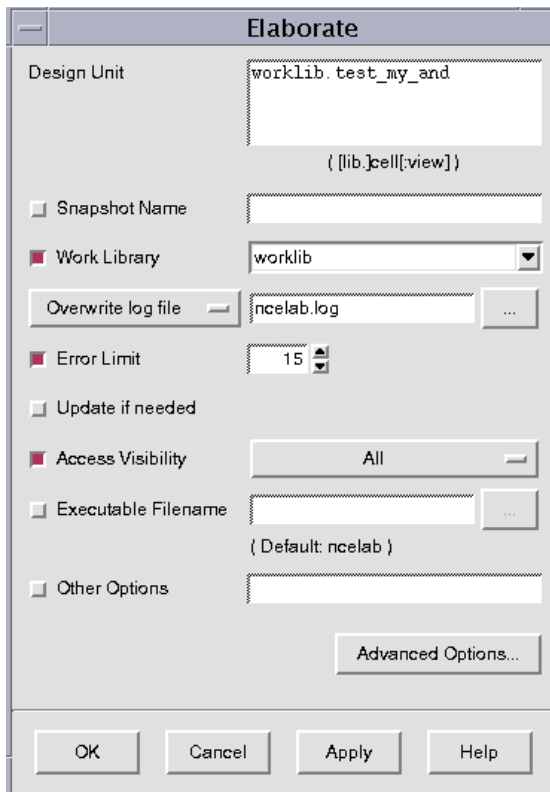


Figure 4: Elaborate Form

Check the compiler window for any possible errors or problems, again.  
 If the elaboration was successful then the snapshot entry below “worklib” should now contain an entry for the “worklib:test\_my\_and:test”. As shown in figure 5.

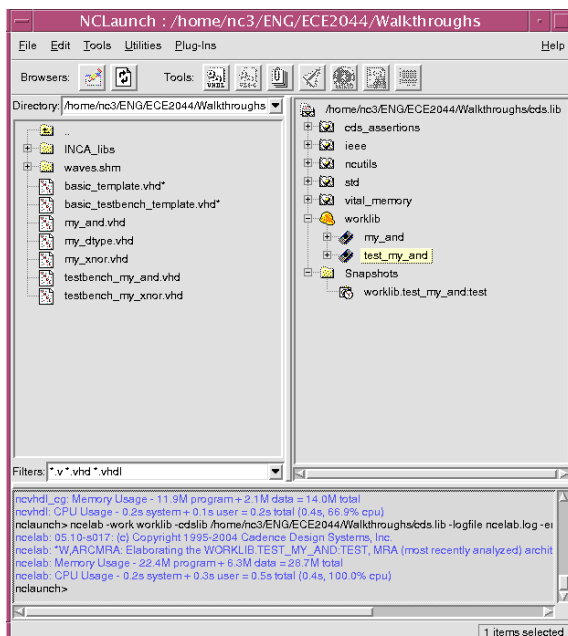


Figure 5: Post Elaborate NCVHDL Tool

## 1.5. Simulating the testbench\_my\_and snapshot

To do this we need to access the snapshot folder below the worklib folder on the NCVHDL tool. Single click on the snapshot file for the test\_my\_and and use the right hand mouse button to bring up the context sensitive menu. This time select the “NCSim” option to open up the simulator.

This will bring up “Simulate” form, shown in figure 6. “OK” this form to open up the simulation windows.

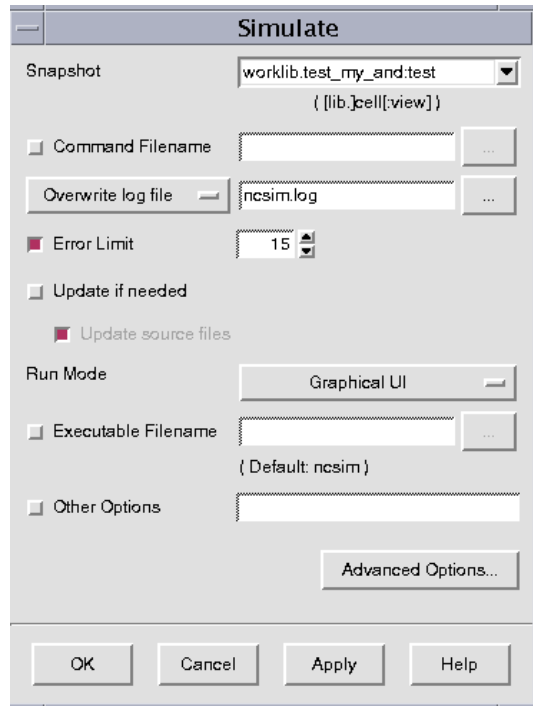


Figure 6: Simulate Form

### 1.5.1. Running a Simulation

This will open up a number of windows including the console and Design Browser , An example of which is shown in figure 7.

#### 1.5.1.1. Selecting Signals

We can use the design browser window to select and display signals we are interested in. We can also use this window to move up and down the design hierarchy ( when we have one) selecting signals other than at the top level.

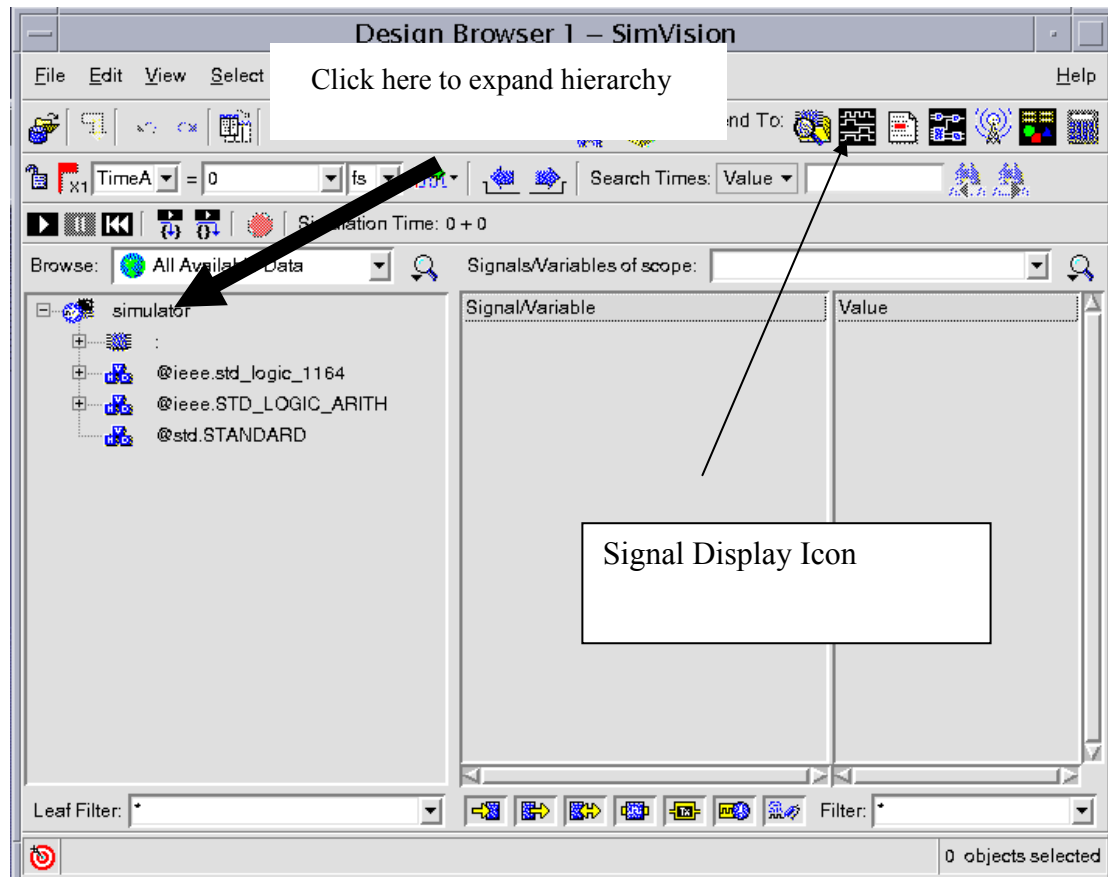
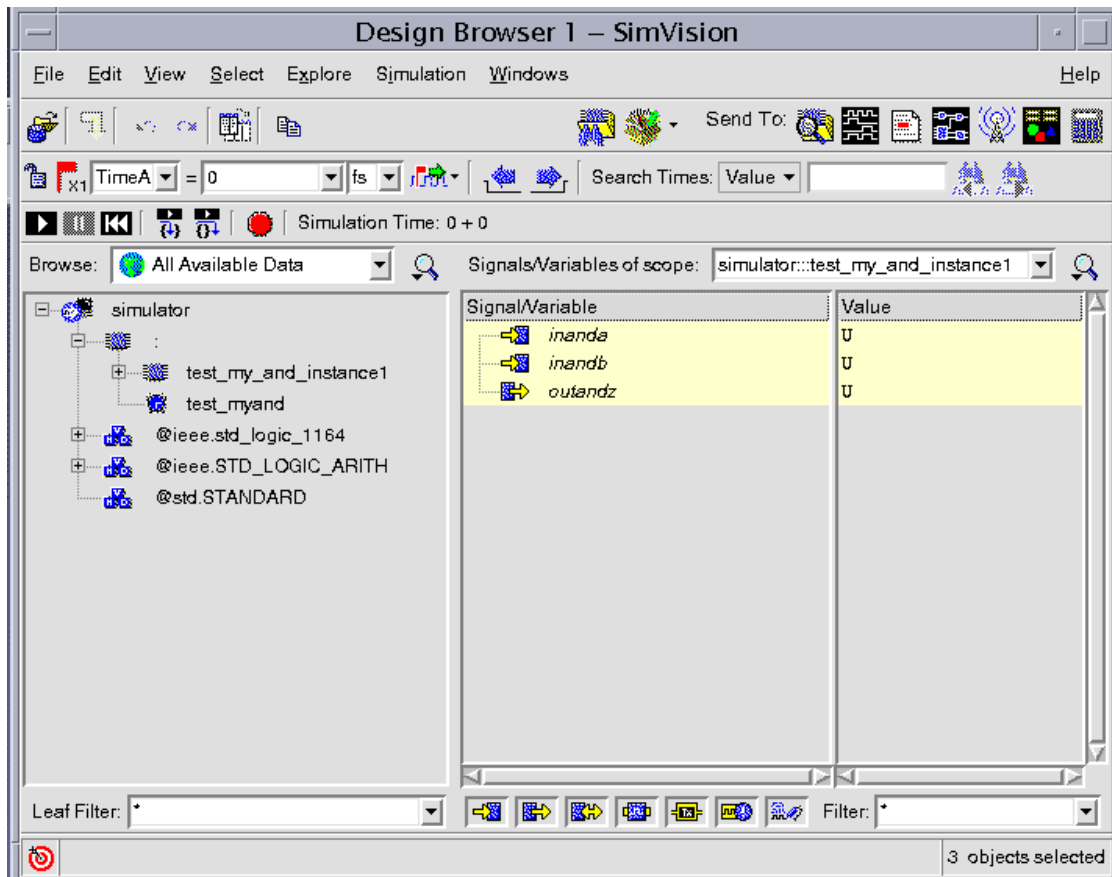


Figure 7 : Design Browser Window SimVision


We can then use the right hand side pane to select the signals we wish to display.



**Figure 8: Design Browser showing Hierarchy structure of design and selected signals for display**

**Hint:**

***If you hold the shift key down you can select multiple signals.***

Once you have selected the signals use the Signal Display Icon  to transfer the selection to the waveform viewing window. If there is no waveform viewing open then it will be opened automatically.

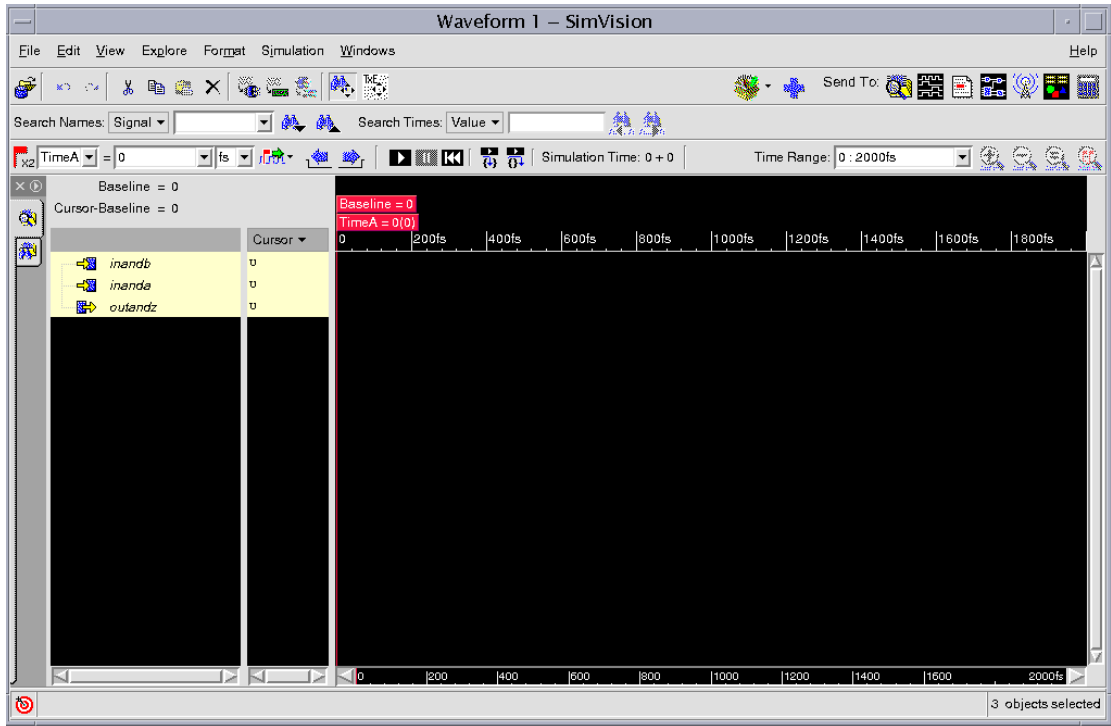


Figure 9: Waveform viewing window with selected signals prior to simulation.

Notice that at this point there are no simulation result and the signal states are shown as “u”, for undefined. We now need to run the simulation to examine the results. You can run a simulation for a set time, for example using the command “run 50 us”, you can enter this in the console window. However in this case we added a “wait” to stop simulation in the test bench so we can just use the “Simulation->Run” option from the menu, on most of the Simvision windows.

The result should be:

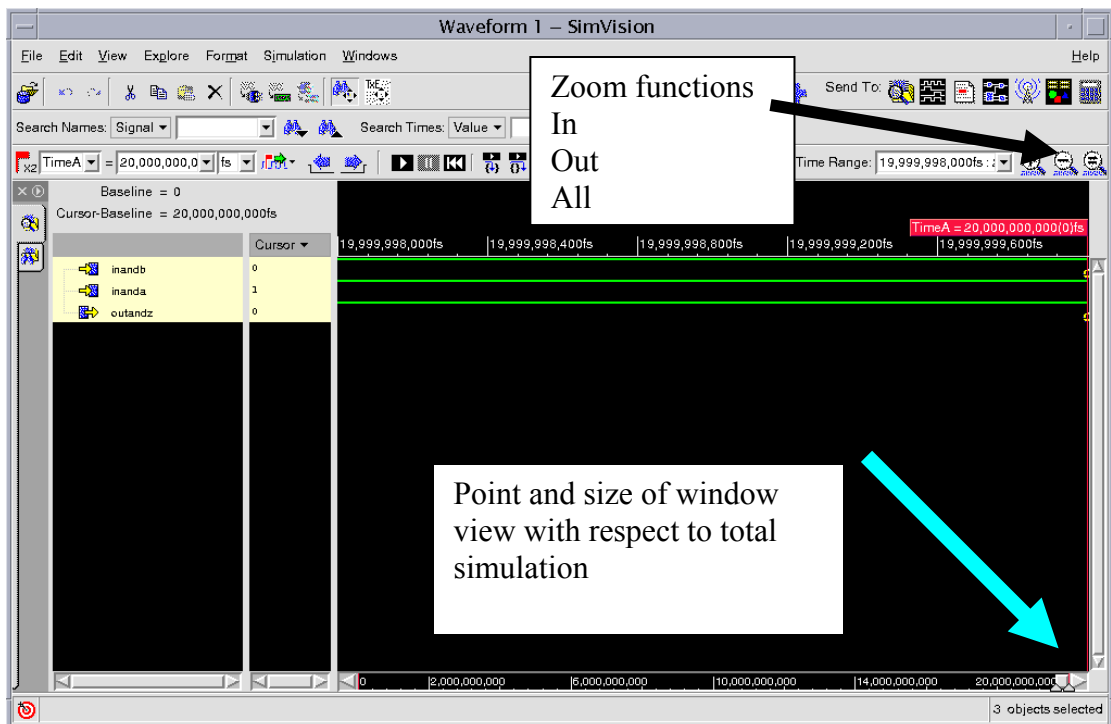


Figure 10: Simvision Waveform tool with simulation results displayed.

However we are not seeing the whole of the simulation. We need to view the whole simulation and we can use the Zoom -> All icon to do this. The result is shown in figure 11. Notice in particular the size of the bar below the waveform pane in figures 10 and 11. Showing how much of the total waveform is displayed.

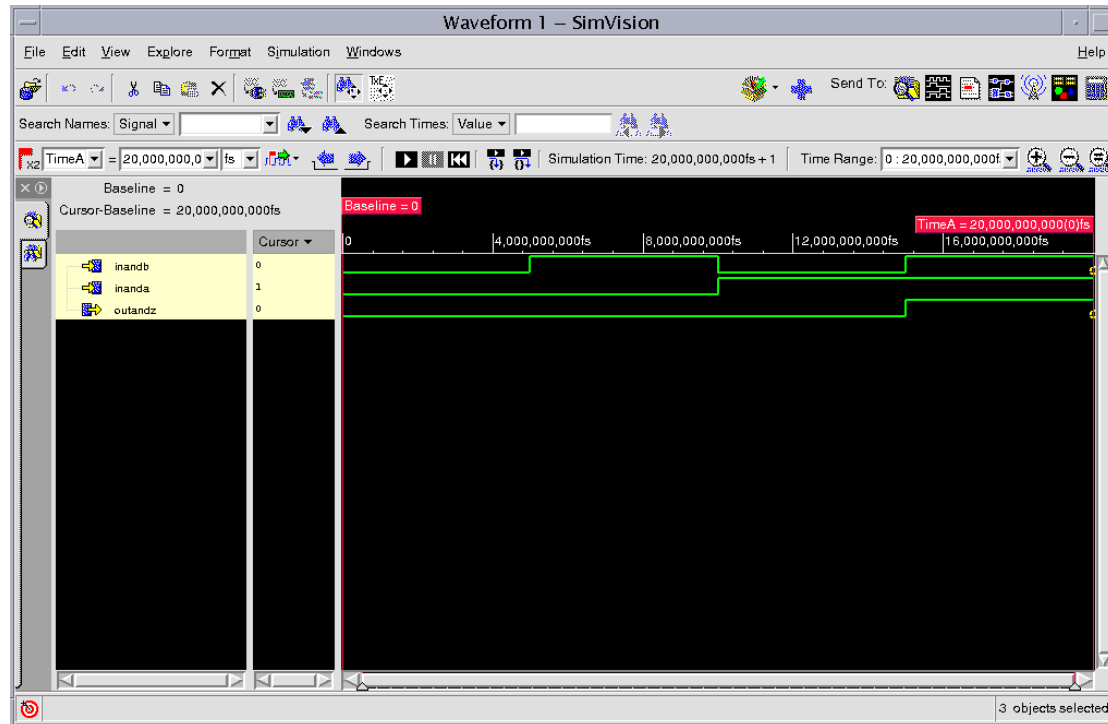


Figure 11: Zoomed Simulation results showing the whole simulation period.

Now we can see the inputs and the outputs from the simulation and we can check to see that the results are correct.

## 1.6. Congratulations.

At this point we should have managed to compile, elaborate and simulate the my\_and.vhd and the testbench\_my\_and.vhd files.

You are expected to experiment and look at the options and how things work. I would strongly recommend that you make notes to refresh your memories when you use the commands again.

## 1.7. Using the other my\_and architectures

You can if you wish change the testbench\_my\_and "use" entries to select other architectures and simulate those should you wish.

Look for the for all entry in the testbench\_my\_and and change it accordingly. You should be able to figure out the requisite changes easily. All the architectures should give identical results.

### 3. Additional Exercise

There are additional exercise for the student:

1. Use the xnor templates to create an xnor gate and test it to ensure correct function. Pick one or more of the architectures and use them as templates.
2. Create an SR Latch using the my\_and gates.
  - a. You will need to add inverts using “not”, or create nand gates in some manner as an SR latch requires nand gates!
3. Use the dtype templates to create a dtype and test it to ensure correct operation
  - a. Create one using “clock”, “indata” and “outdata”
  - b. Now extend this to include “inresetn” and “insetn”
  - c. Make sure the test bench covers all the operations

### 4. Hints

1. If the system does not understand “xor” then make sure you turn on the VHDL 93 options on the VHDL compile.
2. You can generate a clock on a test bench in the following way
  - Clock\_process: Process ()
  - Begin
    - Clock=`0`;
    - Wait for 20 us;
    - Clock=`1`;
    - Wait for 20us;
  - End process;

**In this case you will need to use the “run NN us” command. Where “NN” is a value chosen by you!. As the clock will run for ever. For those of you who forget the “Simulation -> Stop” option will help!**

There are also other ways that this function can be generated.

You are expected to finish these for “homework”.

Using a VHDL manual to clarify any potential issues and trying to write/modify the VHDL would be recommended as an exercise for the next week.