ECSE-221-222
Digital System Design

Lab #1 – *Using the Altera Quartus II Software* Winter 2018

Based on Labs developed from work of Dr. J. Clark for ECSE 323

Introduction

In this lab you will learn the basics of the Altera Quartus II FPGA design software through following a step-by-step tutorial, and use it to design two simple combinational logic circuits.

Learning Outcomes .

After completing this lab you should know how to:

- Startup the Altera Quartus II software
- Create the framework for a new project
- Name and connect nodes and busses using the block diagram editor
- Generate a functional simulation netlist
- Create a Vector Waveform File (for simulation inputs)
- Do functional simulation of a circuit

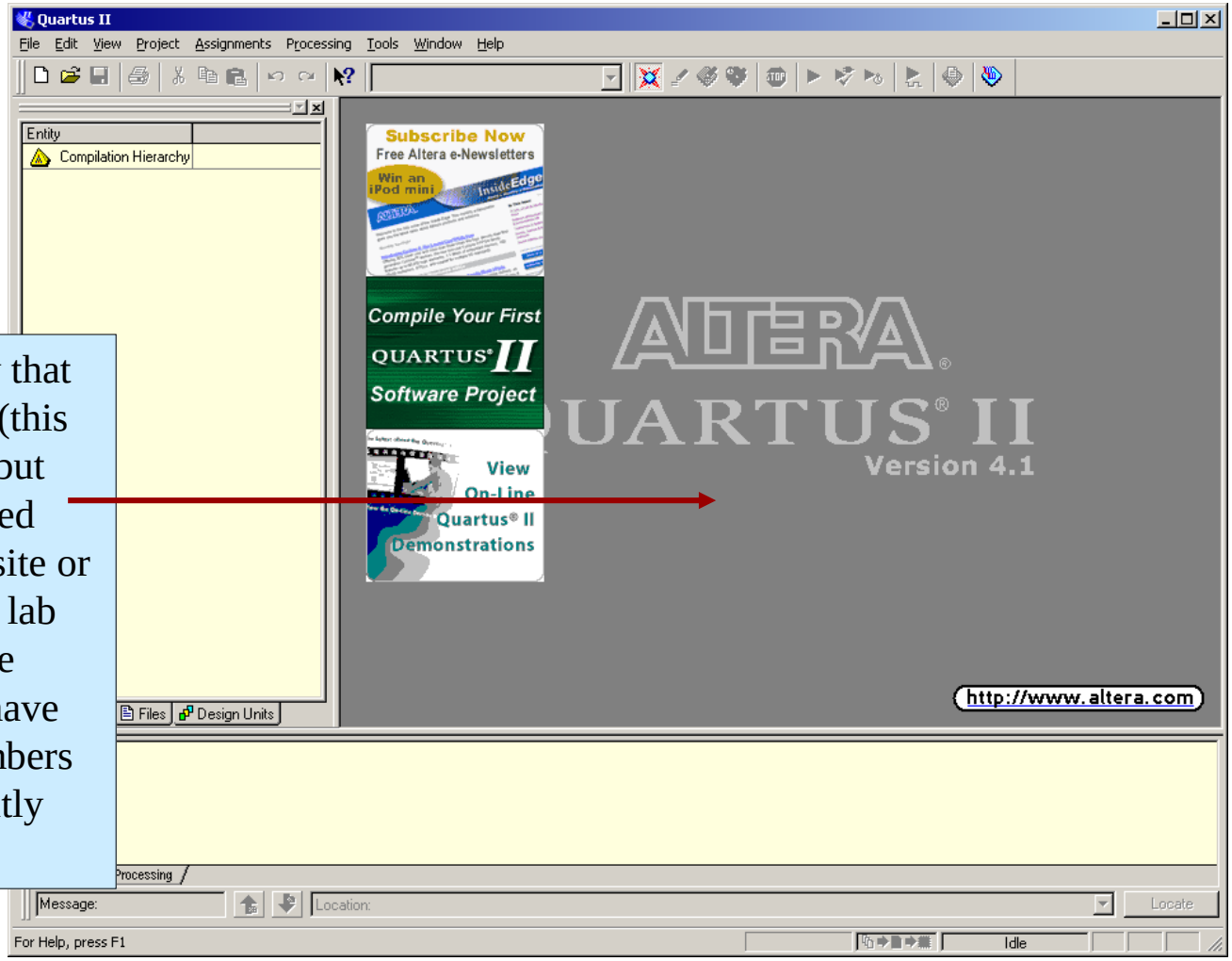
2. Startup and creation of the project framework

In this course you will be using commercial FPGA design software, the Altera ***Quartus II*** program. A slightly restricted version of this program is contained on the CD that comes with the course textbook, and can also be downloaded from the Altera web site (preferred, since the Altera site will have the most recent version). The program restrictions will not affect any designs you will be doing in the course, so you can install the program on your personal computer, so that you can work on your project outside of the lab.

The purpose of this first lab is to familiarize yourself with the Quartus program, by going through a step-by-step tutorial on its use. You will use it to develop two simple circuits using schematic capture techniques.

To begin, start the ***Quartus II*** program by either by selecting the program in the Windows Start menu or double-clicking on the desktop shortcut icon (if one exists).

This is the window that appears on startup (this shows version 4.1 but versions downloaded from Altera's web site or the versions on the lab computers or on the textbook CD will have higher version numbers and may look slightly different)

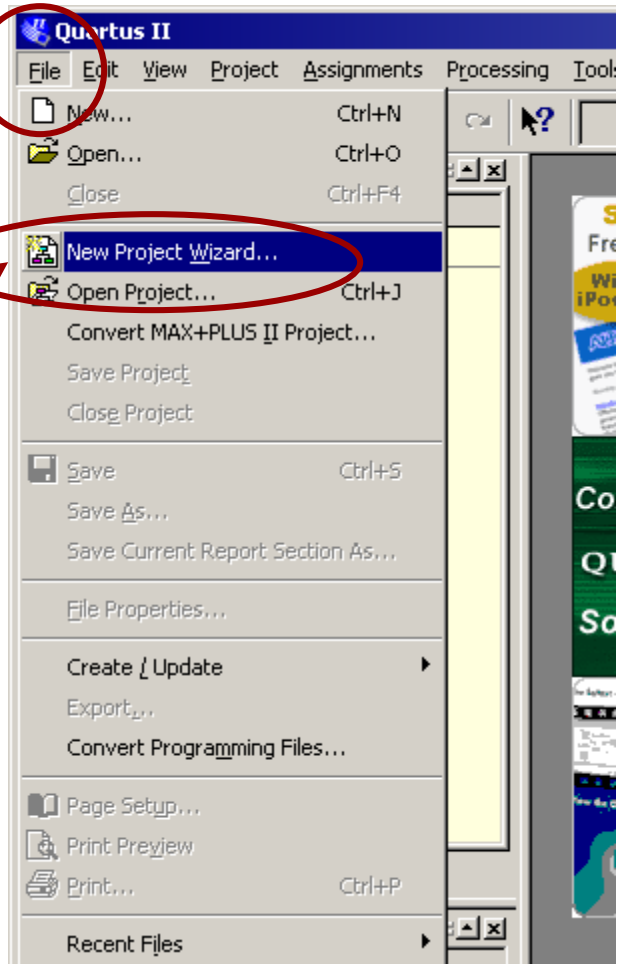


The Altera ***Quartus II*** program employs a ***project-based approach***. The goal of a Quartus project is to develop a hardware implementation of a specific function, targeted to an FPGA (*Field Programmable Gate Array*) device.

Typically, the project will involve a (large) number of different circuits, each designed individually, or taken from circuit libraries. Project management is therefore important. The Quartus II program aids in the project management by providing a project framework, that keeps track of the various components of the project, including ***design files*** (such as schematic block diagrams or VHDL descriptions), ***simulation vector files***, ***compilation reports***, ***FPGA configuration or programming files***, project specific program settings and assignments, and many others.

The first step in designing a system using the Quartus II approach is therefore to create the project framework. The program simplifies this by providing a "Wizard" which guides you through a step-by-step setting of the most important options.

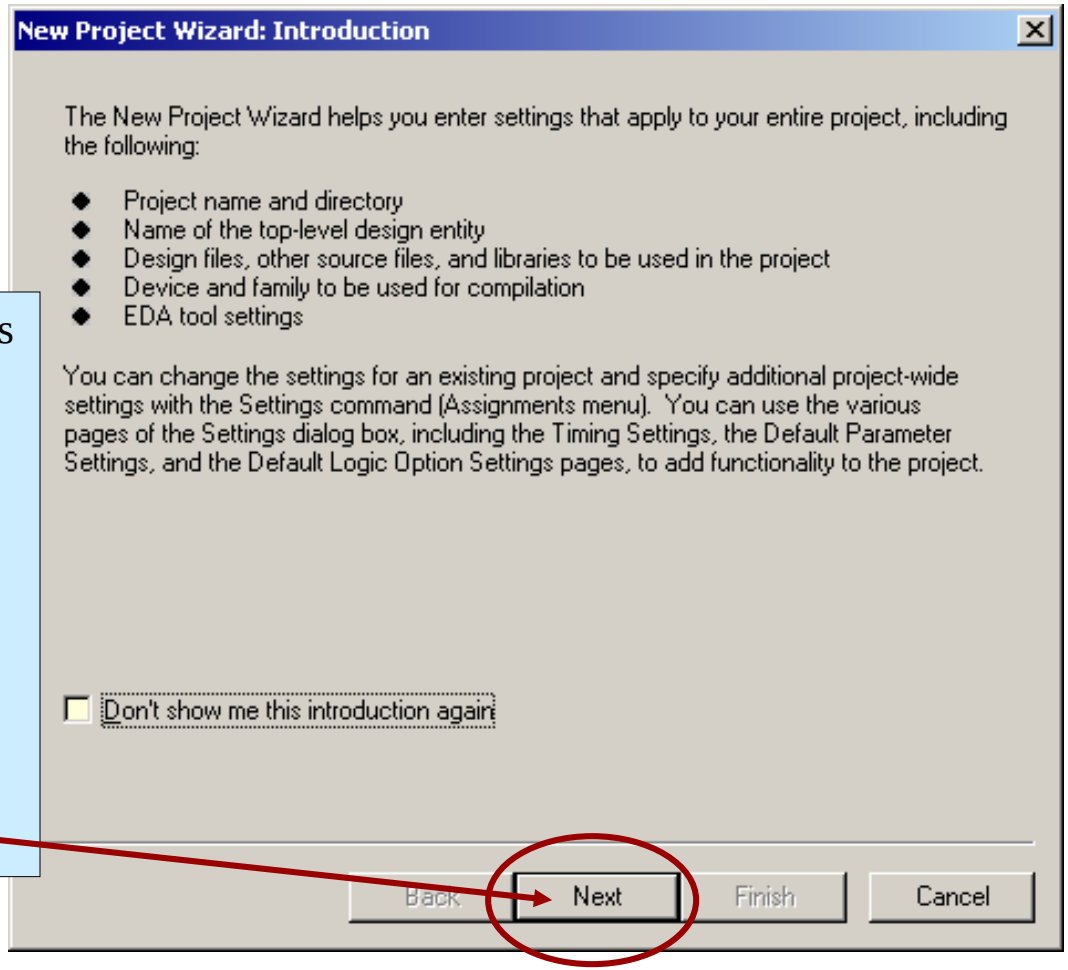
To run the Project Wizard, click on the "***File***" menu and select the "***New Project Wizard***" entry.



The New Project Wizard involves going through a series of windows.

The first window, shown at right, is an introduction, listing the settings that can be applied.

After reading the text on this window, click on "Next" to proceed.

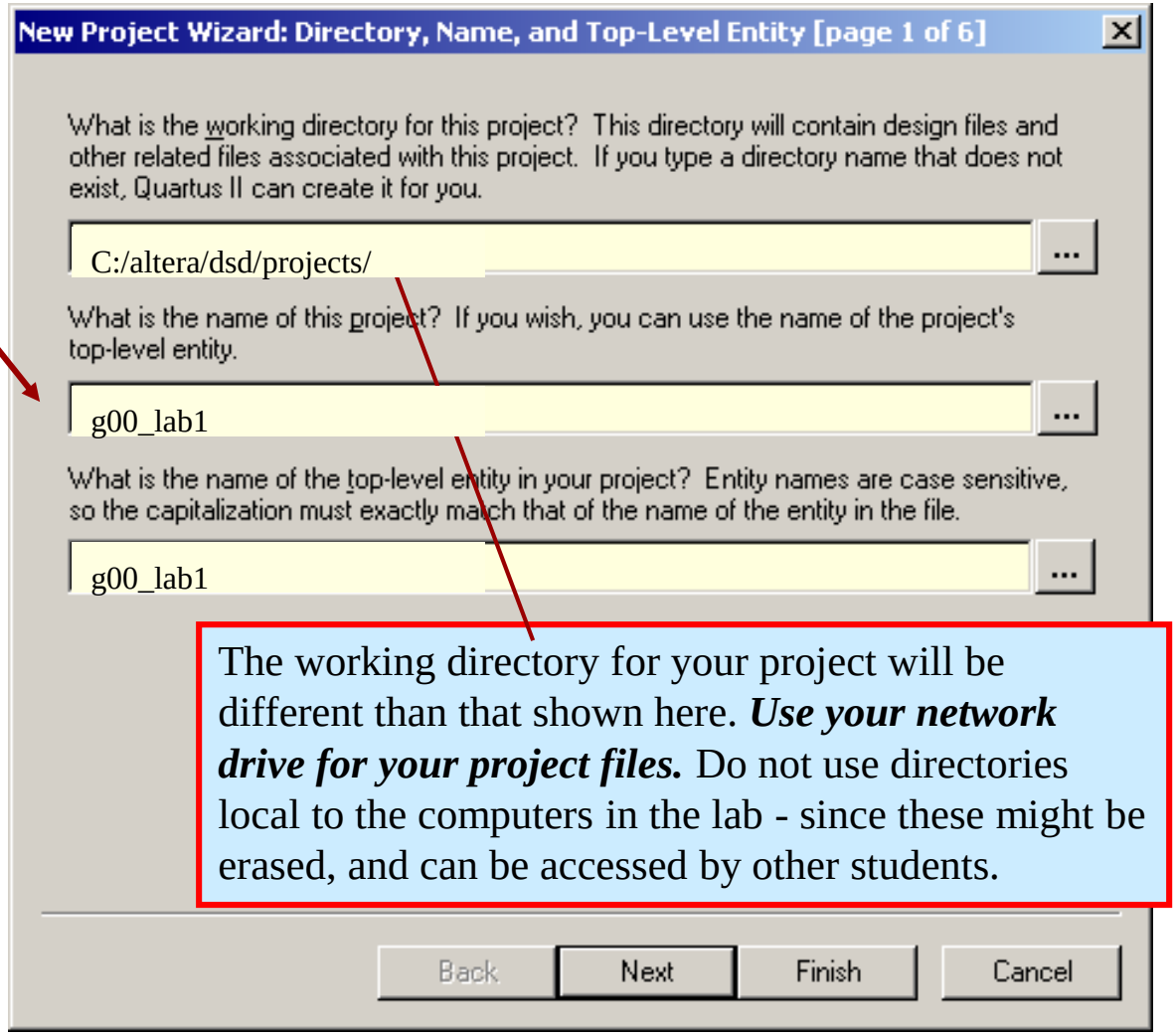


In the second window you should give the project the following name:

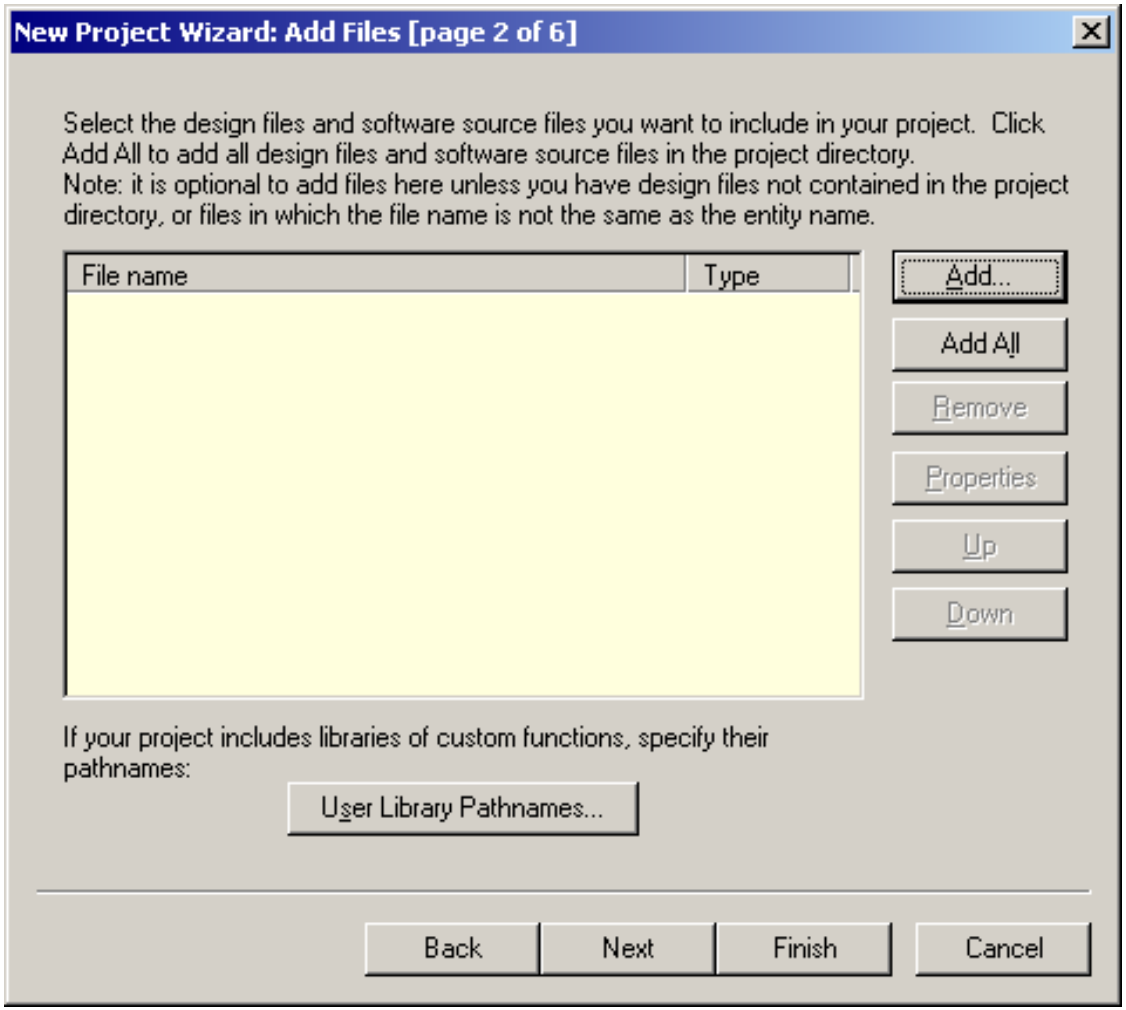
gNN_lab1

where NN is replaced with your 2-digit group number - e.g. if you are in group 21, then call the project:
g21_ lab1
Use this naming convention throughout the course for all of your designs.

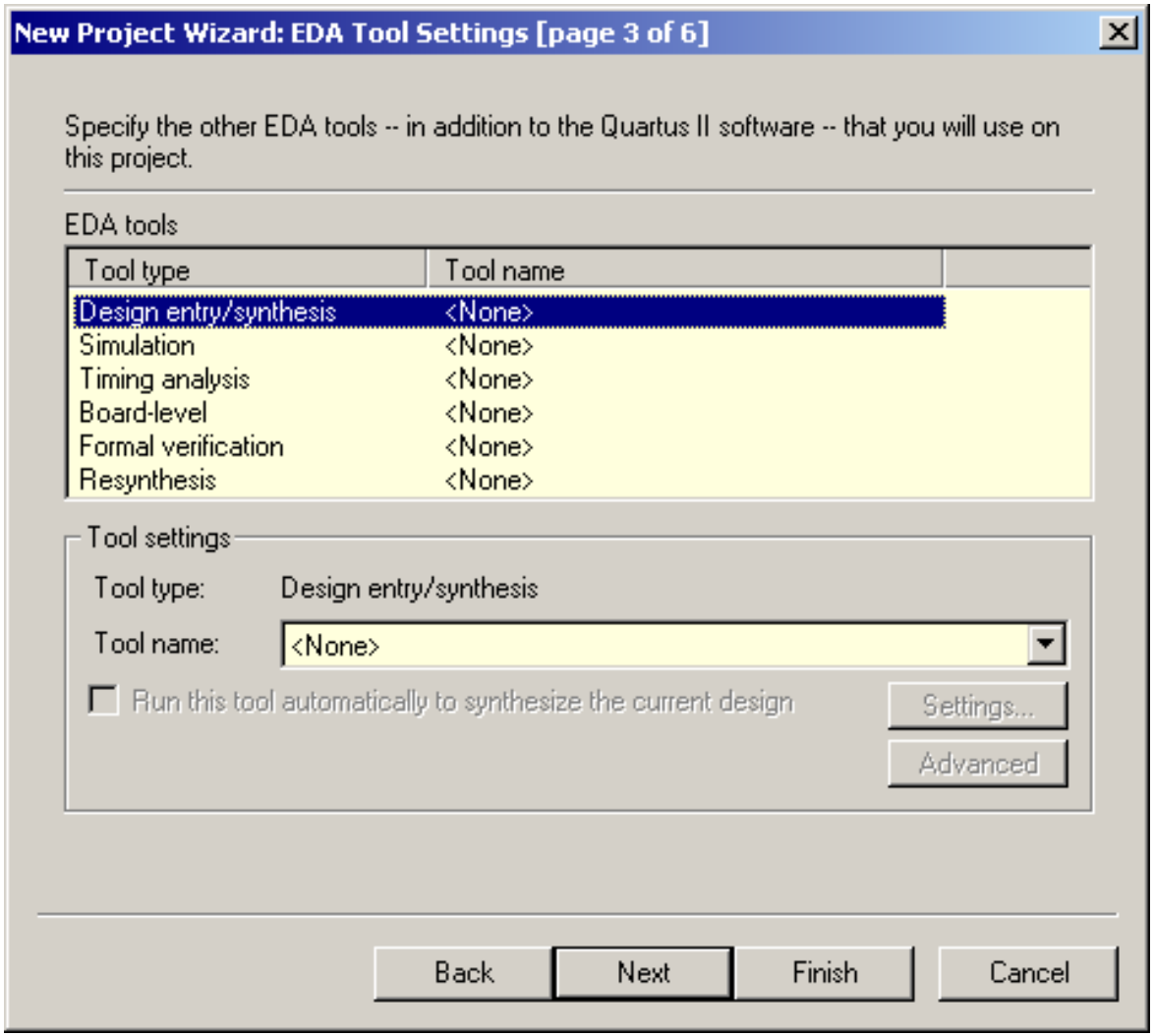
Note: ***g00_*** will be used for designs developed by the course instructors



The working directory for your project will be different than that shown here. ***Use your network drive for your project files.*** Do not use directories local to the computers in the lab - since these might be erased, and can be accessed by other students.



You will add files later, so for now, just click on "Next".



This dialog box permits the designer to specify 3rd-party tools to use for various parts of the design process.

We will not be using any tools other than those provided by the Quartus II program, so just click on "Next".

Entering a VHDL file:

From toolbar selection: Select New

-Then select VHDL file

This will invoke your text editor

You will then type in your VHDL code. Remember to save often.

After you have completed your design you will have to make a netlist file.

You select processing from the toolbar. Then start compilation

The 2 circuit components that will be generated are as follows:

1) 1-bit Adder.
2) 1-bit Subtractor

You are to write the VHDL file and compile the VHDL files.
Next you are to simulate the circuits to verify that they work.

One Bit Adder Truth Table

A	B	Carry_in	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

One Bit subtractor Truth Table

A	B	Borrow_in	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

3. Performing Functional Simulation of the Project

Once you have your circuit described in a schematic diagram (or, in later labs, in a VHDL description) you should simulate it.

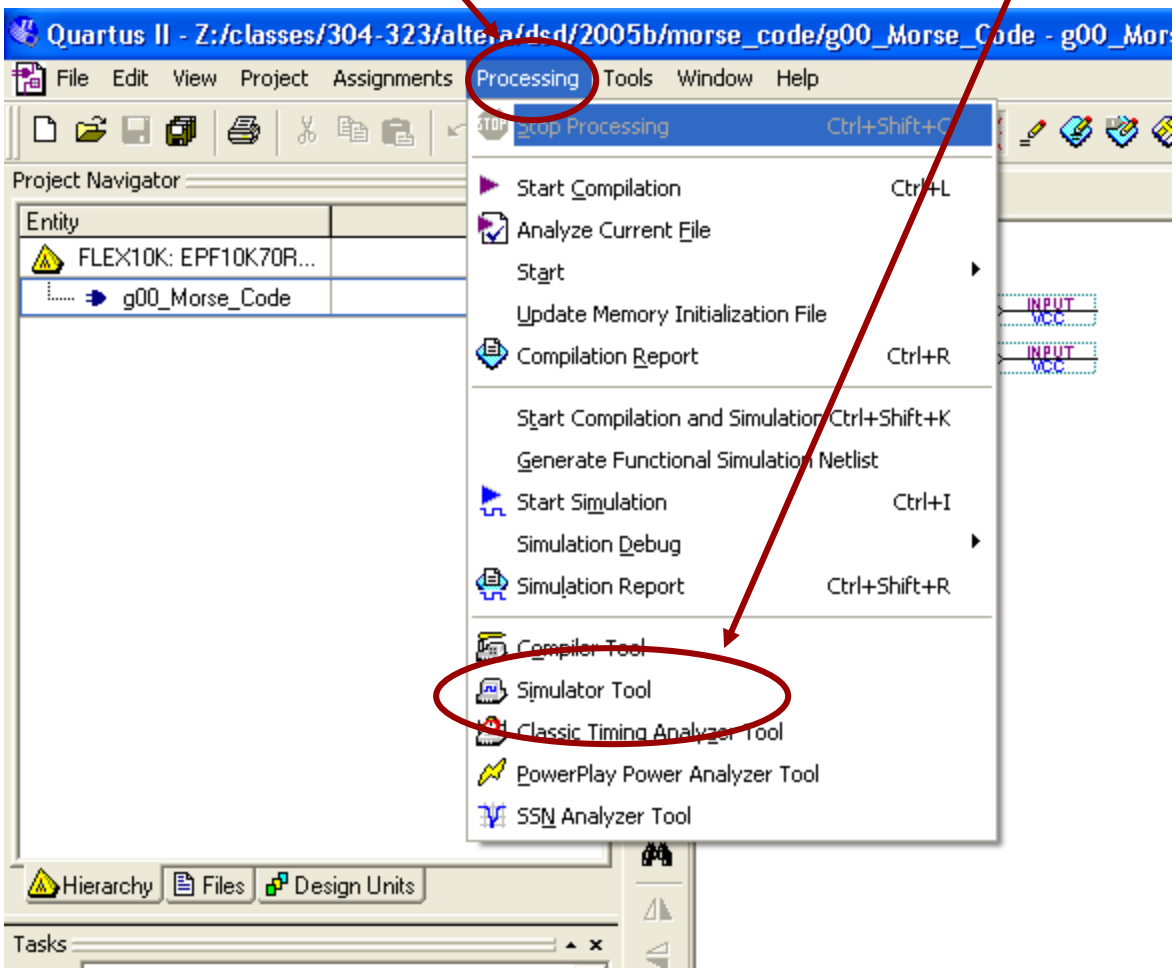
The purpose of simulation is generally two-fold:

1. To determine if the circuit performs the desired function
2. To determine if timing constraints are met

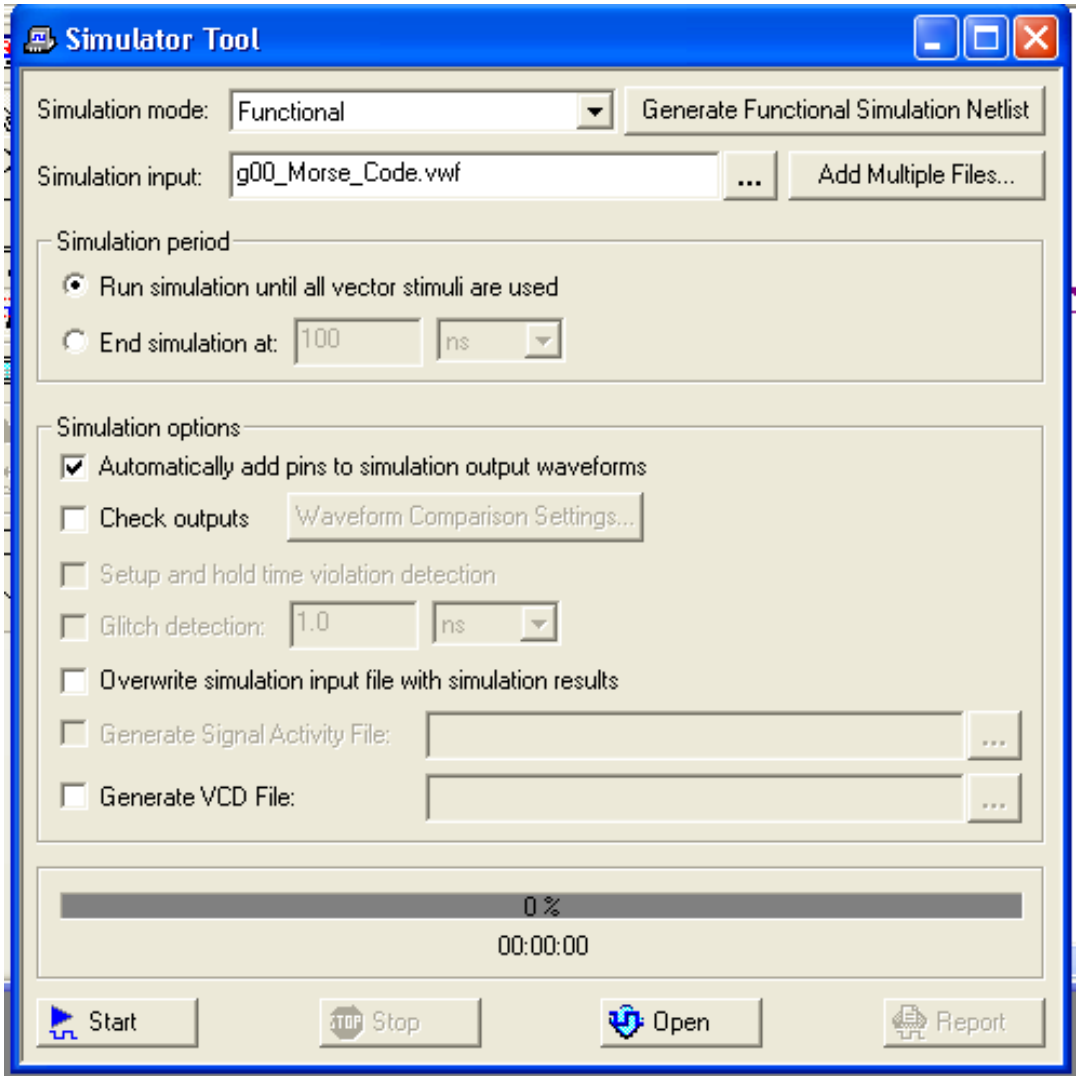
In the first case, we are only interested in the functionality of our implementation. We do not care about propagation delays and other timing issues. Because of this, we do not have to map our design to a target hardware. This type of simulation is called ***functional simulation.*** *This is the type of simulation we will learn about in this lab.*

The other form of simulation is called timing simulation, or hardware simulation. It requires that the design be mapped onto a target device, such as an FPGA. Based on the model of the device, the simulator can predict propagation delays, and provide a simulation that takes these into account. Thus, the timing simulation may produce results that are quite different from the purely functional simulation.

To begin setting up to do a functional simulation, select the "***Simulator Tool***" item in the "***Processing***" menu.



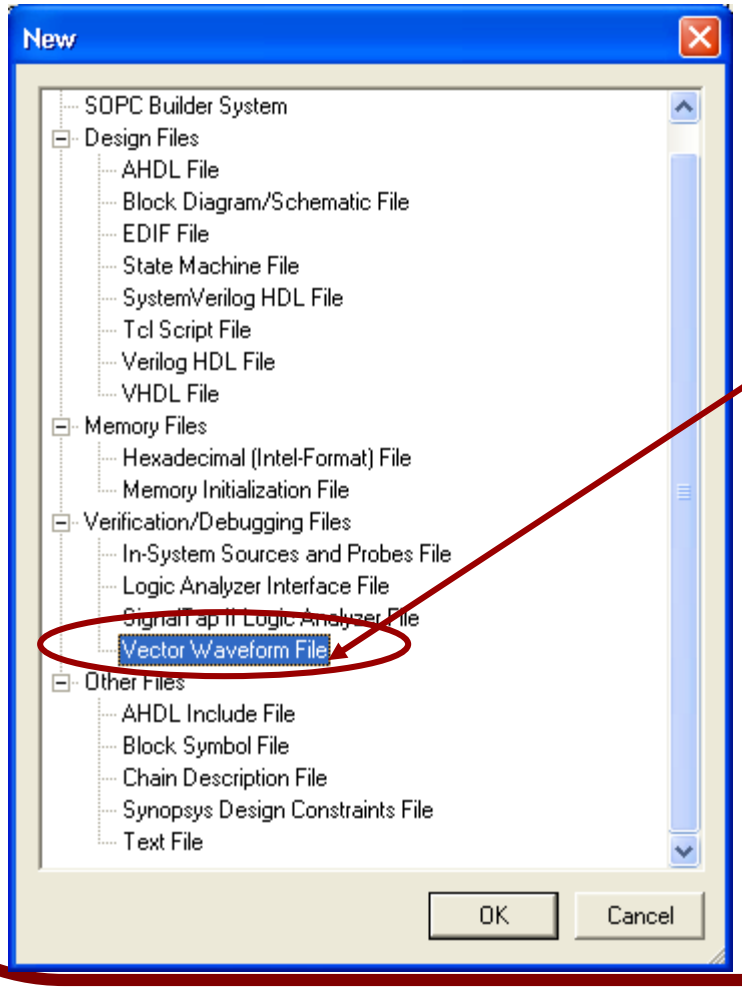
The "***Simulator Tool***" window will appear…



Click on the "***Generate Functional Simulation Netlist***" button in the simulator tool window. This will analyze the circuitry in the project and extract a set of logical equations which describe the functionality of the circuit.

When running a simulation it is useful to have a means of setting the inputs to certain patterns, and of observing the outputs' responses to these inputs.

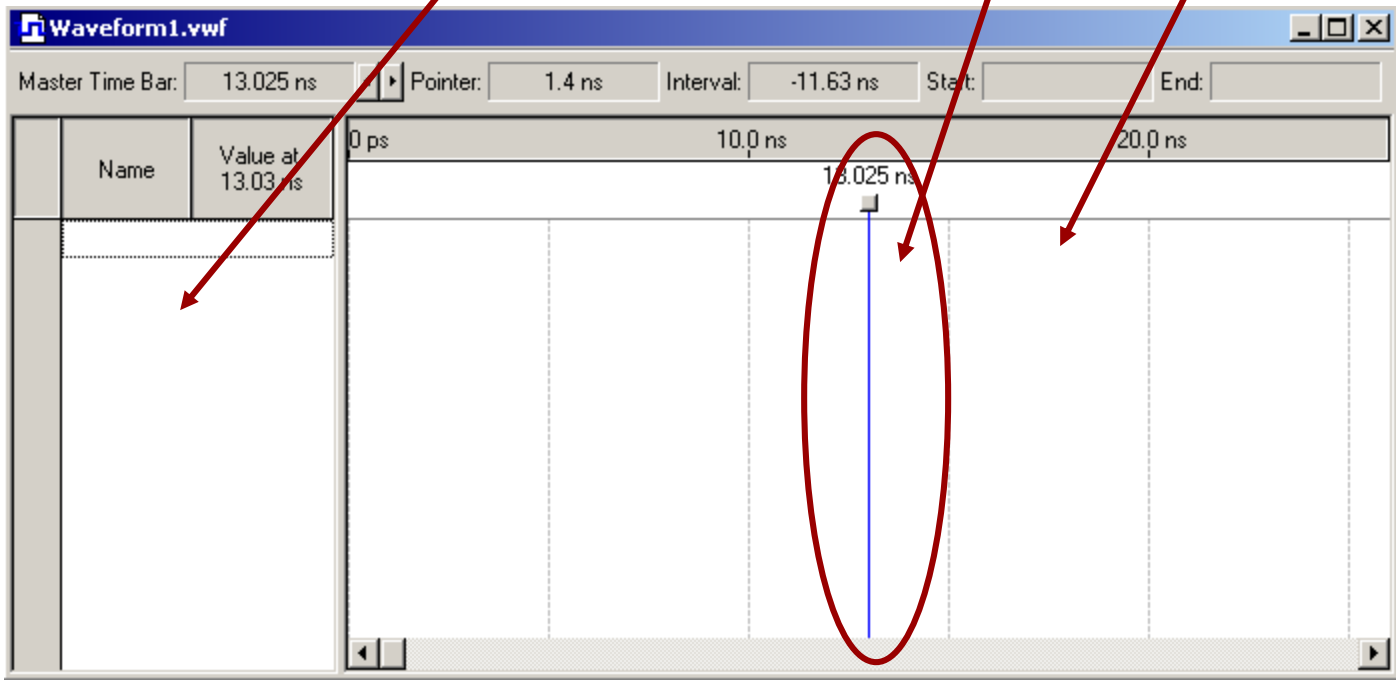
In Quartus, these tasks are done through graphical display of a "***Vector Waveform File***" (.vwf). To create a new vwf file, select the "***New***" command from the "***File***" menu.



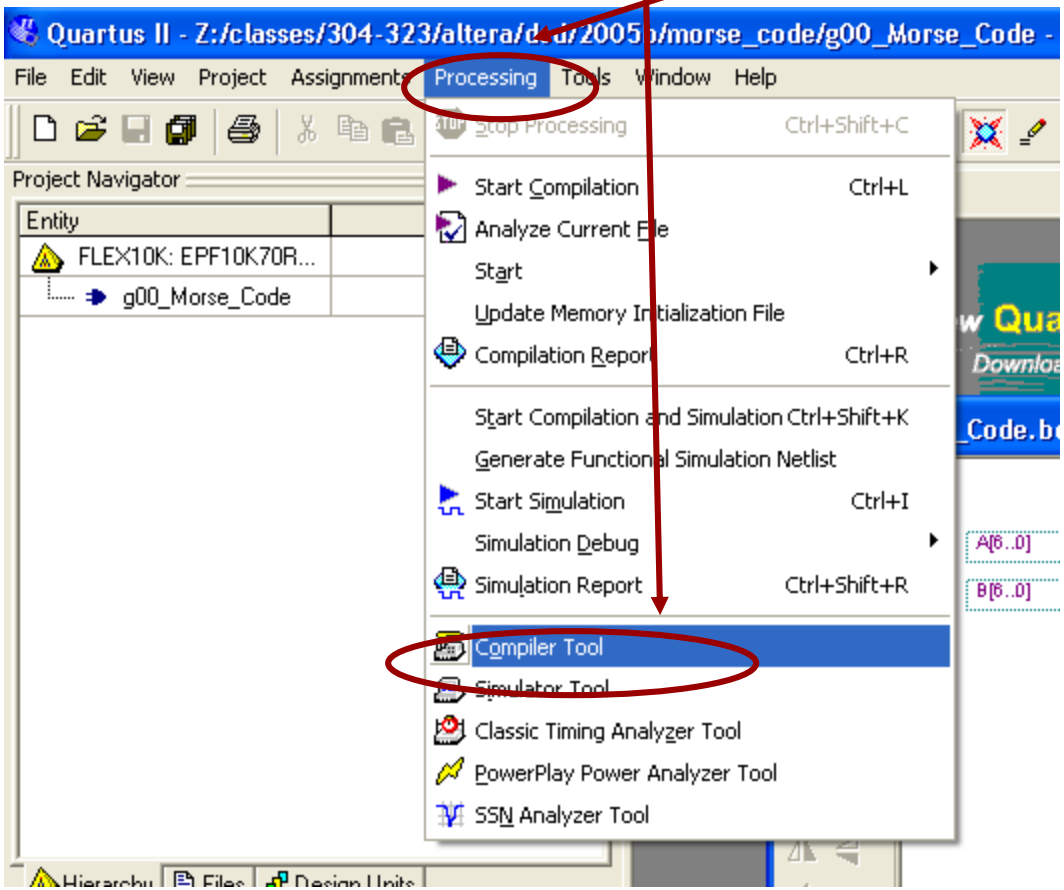
Select the "Vector Waveform File" entry and click on OK.

An empty waveform editor window will appear, as shown below.
It contains two primary display panes - the left hand pane shows the name of the nodes or busses being displayed, and their value at the time indicated by the time-bar, and the right hand pane shows the waveform - which is the node value as a function of time.

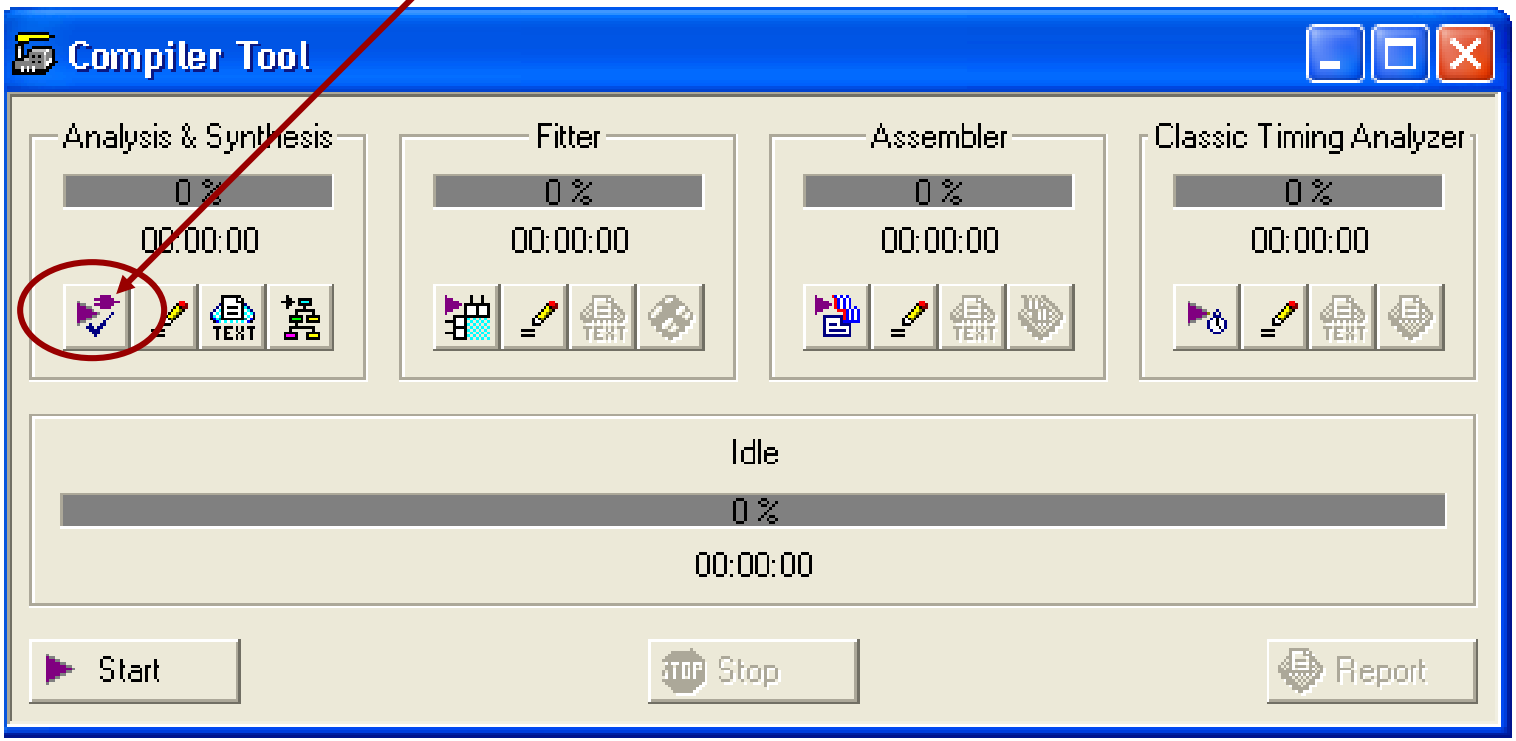
When the VWF is first created, the waveform window will be empty. So, the next thing we must do is specify the nodes and busses that are to be displayed…



Before specifying the nodes to be displayed in the simulator, we must do a ***Compilation*** of the design. To do this, select "***Compiler Tool***" from the "***Tools***" menu.

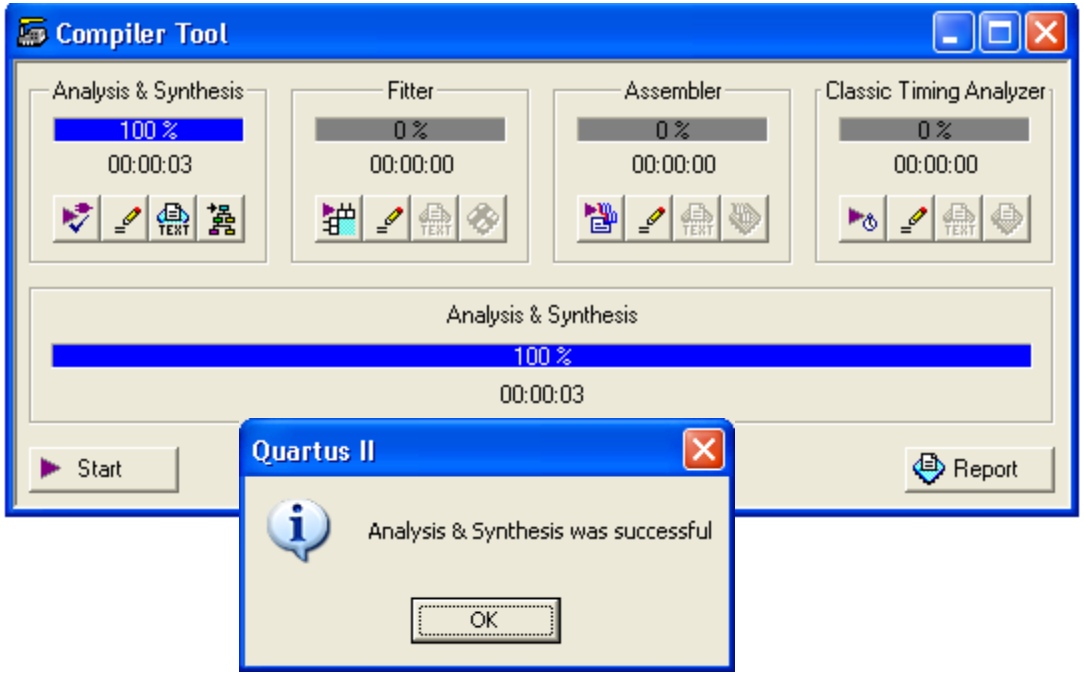


The compiler window will appear. In order to do a functional simulation we do not need to do a complete compilation. We only need to do the first part, Analysis & Synthesis. To run the Analysis/Synthesis, click on the left-most icon in the Analysis & Synthesis block.



On completion the compiler window should look like that shown below.

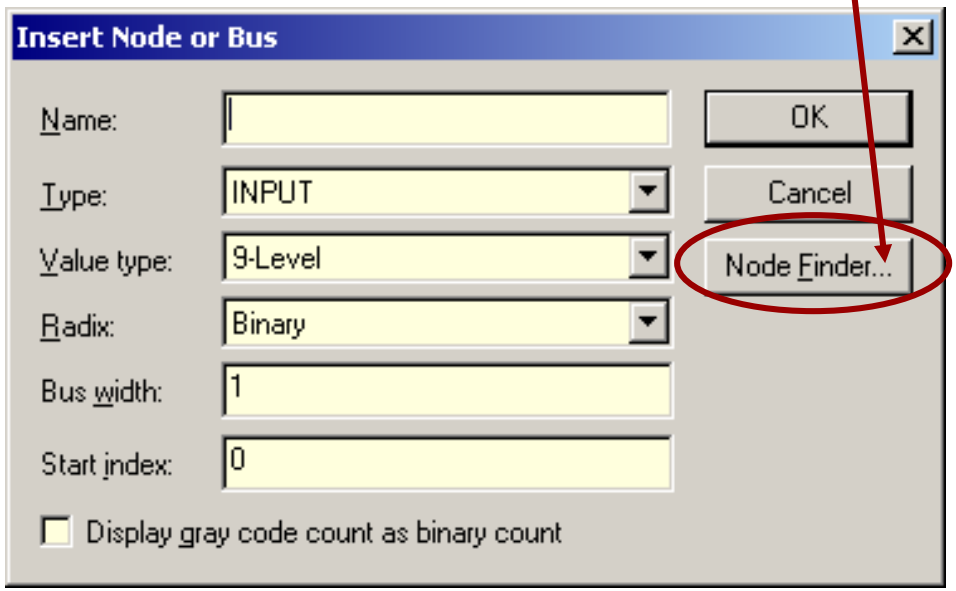
You can now go back to the simulator window and insert some nodes.



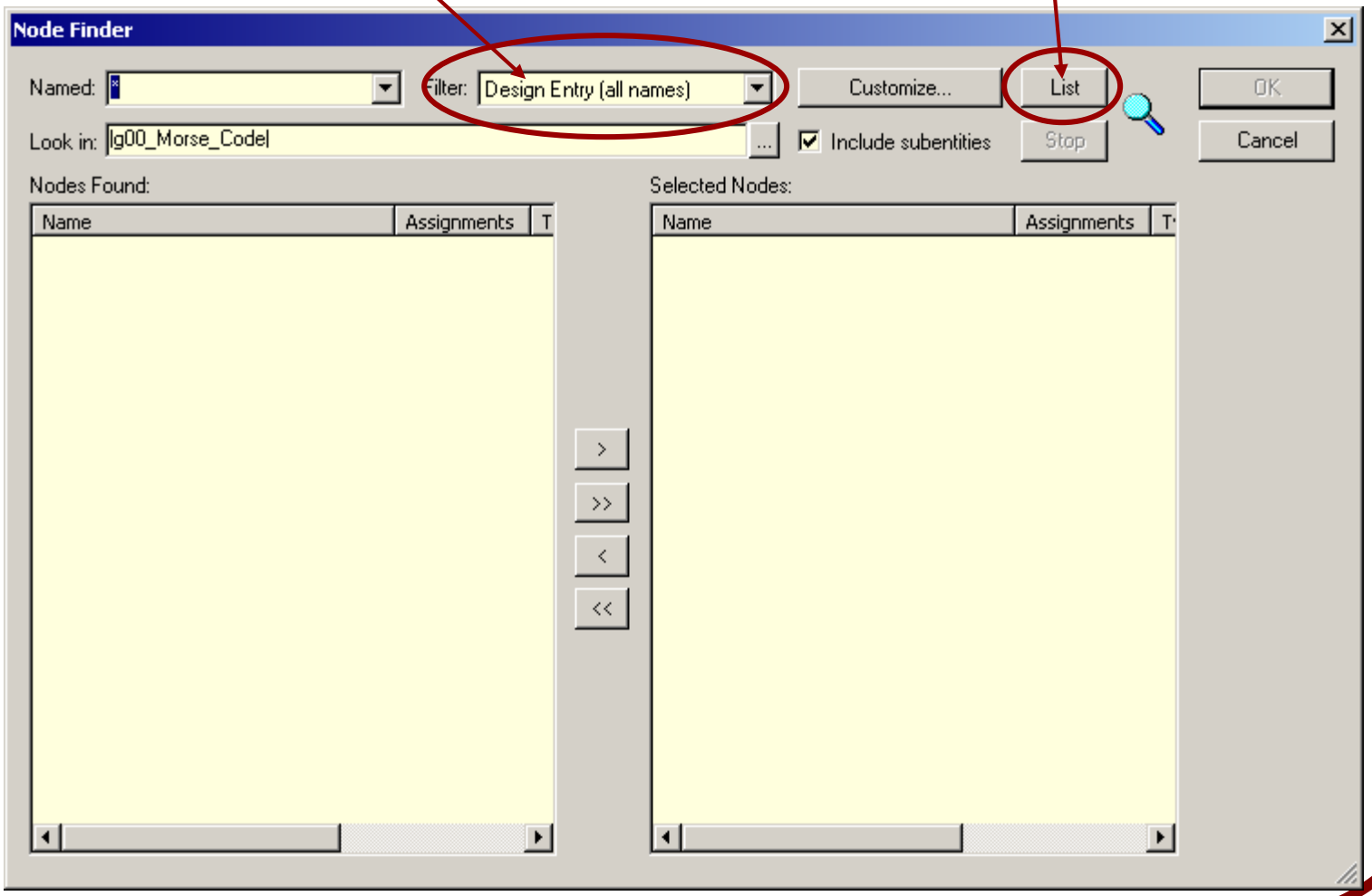
To insert some nodes and busses into the waveform editor window, position the mouse cursor over the left part of the Waveform editor window (the part which says "Name" and "Value"). ***Right***-click the mouse, and select the "Insert Node or Bus…" menu item. A dialog window will popup as shown below.

If you know the name of the node that you want to add then you can type it in to this window. But at the beginning, when you have a lot of nodes to enter, or when you don't know the name of the node, it is better to use the "***Node Finder***".

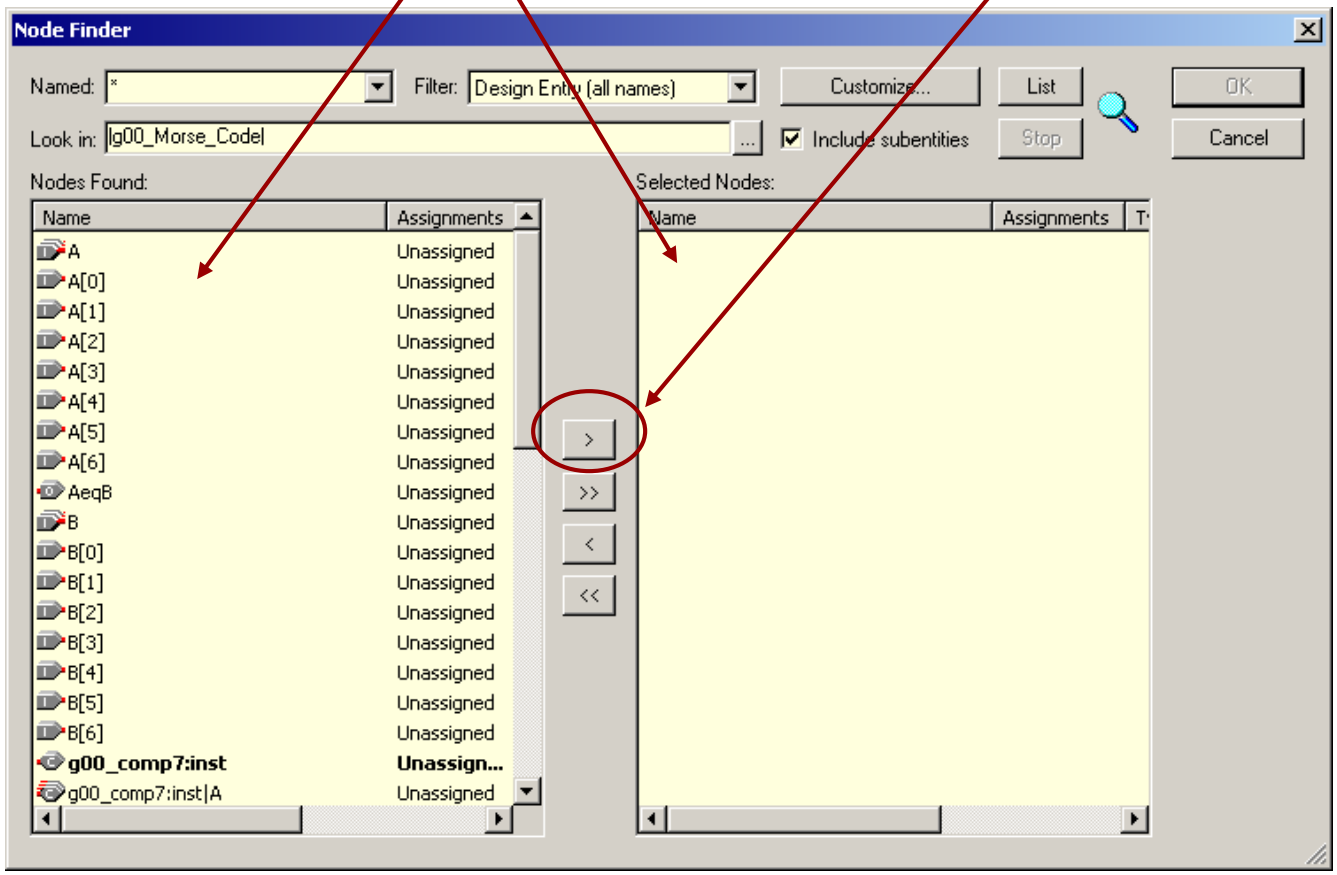
To use the node finder, click on the Node Finder button…



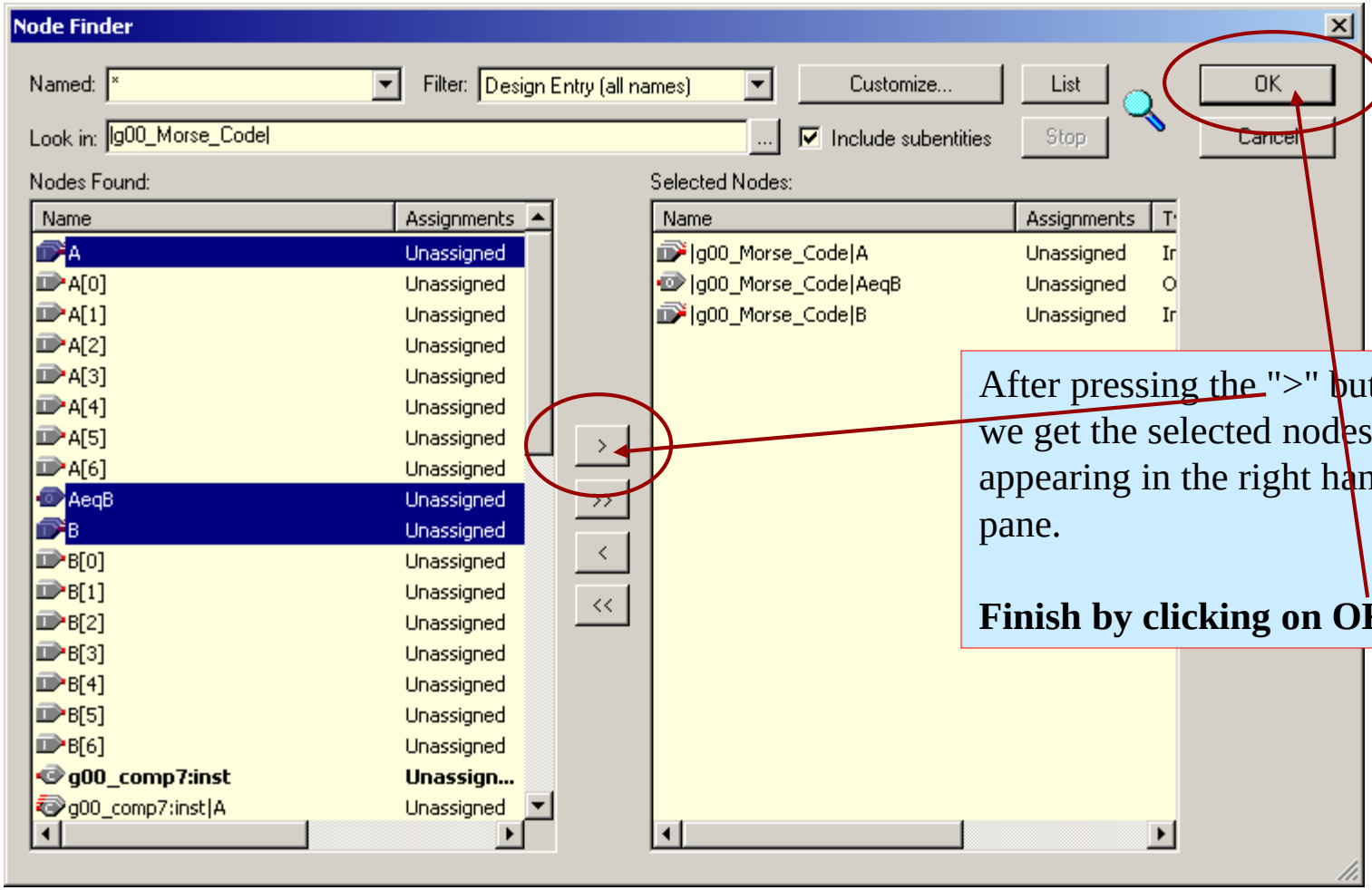
When you click on the Node Finder button, the following window appears. Make sure that the Filter: is set to "***Design Entity (all names)***", and click on the "***List***" button.



When you click on the List button, the "***Nodes Found***" pane on the left side of the Node Finder window is filled with all of the nodes in the design entity. You can now select individual nodes to be shown in the simulation. Selection of multiple nodes is done in the same way as selecting multiple files in the Windows File Explorer (i.e. ctrl-click to select). Once you have all the files you want selected, click on the ">" button to transfer these to the "***Selected Nodes***" pane.



For this case, the only nodes we care about during simulation are the input busses A and B, and the output node AeqB. We could have also selected the individual bus nodes, A[0], A[1], A[2], etc., but it is more convenient to treat them as a whole bus.

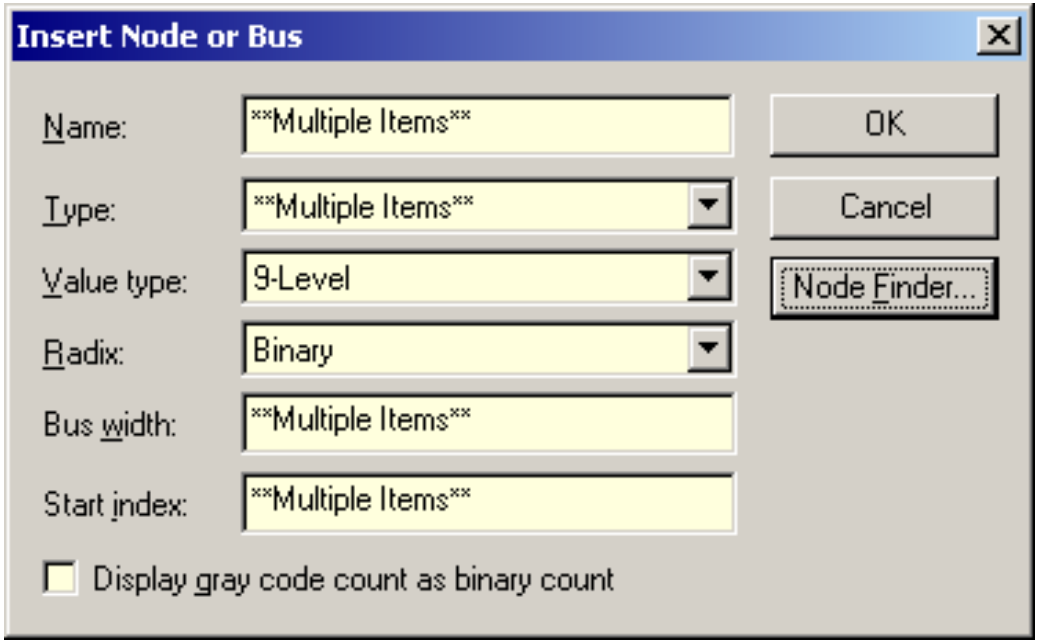


After pressing the ">" button we get the selected nodes appearing in the right hand pane.

Finish by clicking on OK

Returning from the Node Finder dialog, the "Insert Node or Bus" window looks as shown in the figure below.

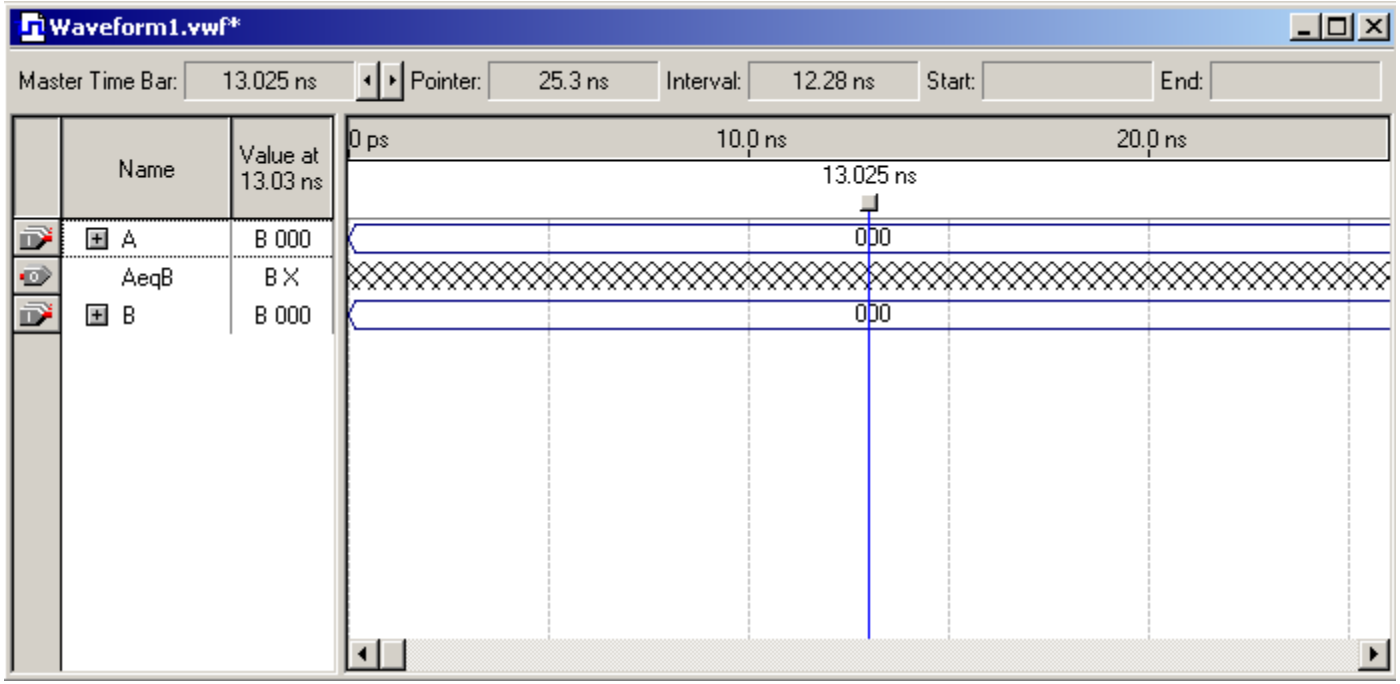
Click on OK to return to the Simulator window.



After insertion of the selected nodes, the Simulator window now looks as in the figure shown below. Note that A and B are both set to zero and the value of AeqB is X, indicating an undefined value (as is expected, since we have yet to run the simulation).

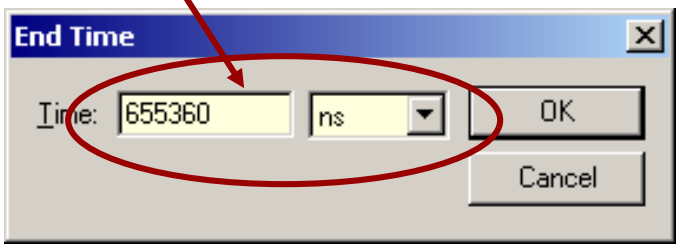
The next step is to specify test input waveforms for the input signals A and B.

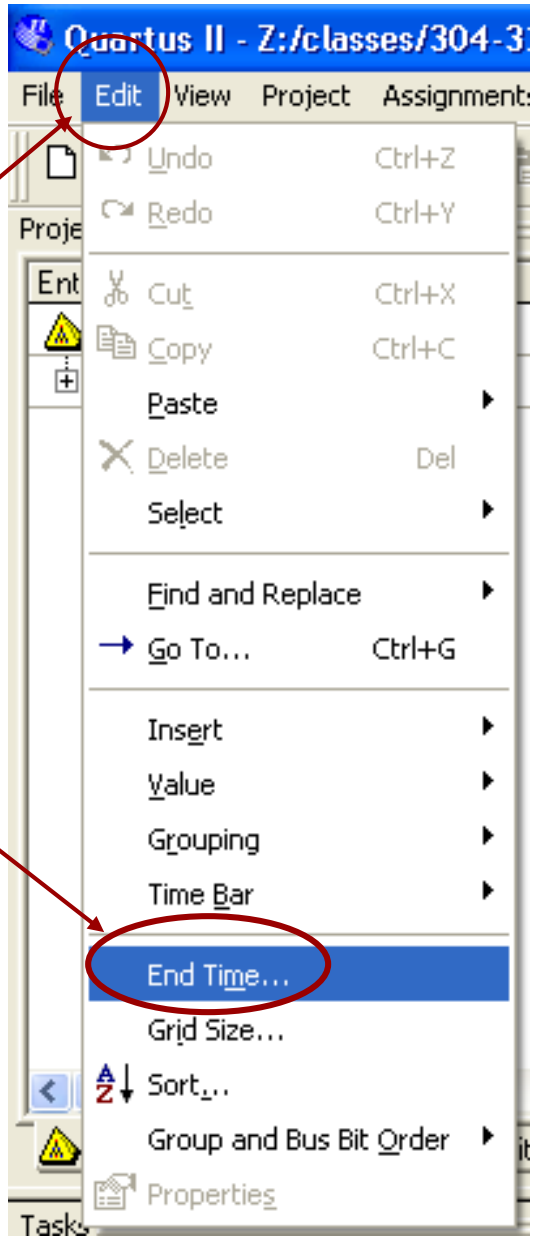
There are 2^14 possible input patterns (16,384), which is a lot to test, but the computer is doing all of the work.



First, set the end time for the simulation, by selecting the "***End Time***" item in the ***Edit*** menu. A window will popup, allowing you to specify the time at which the simulation will end.

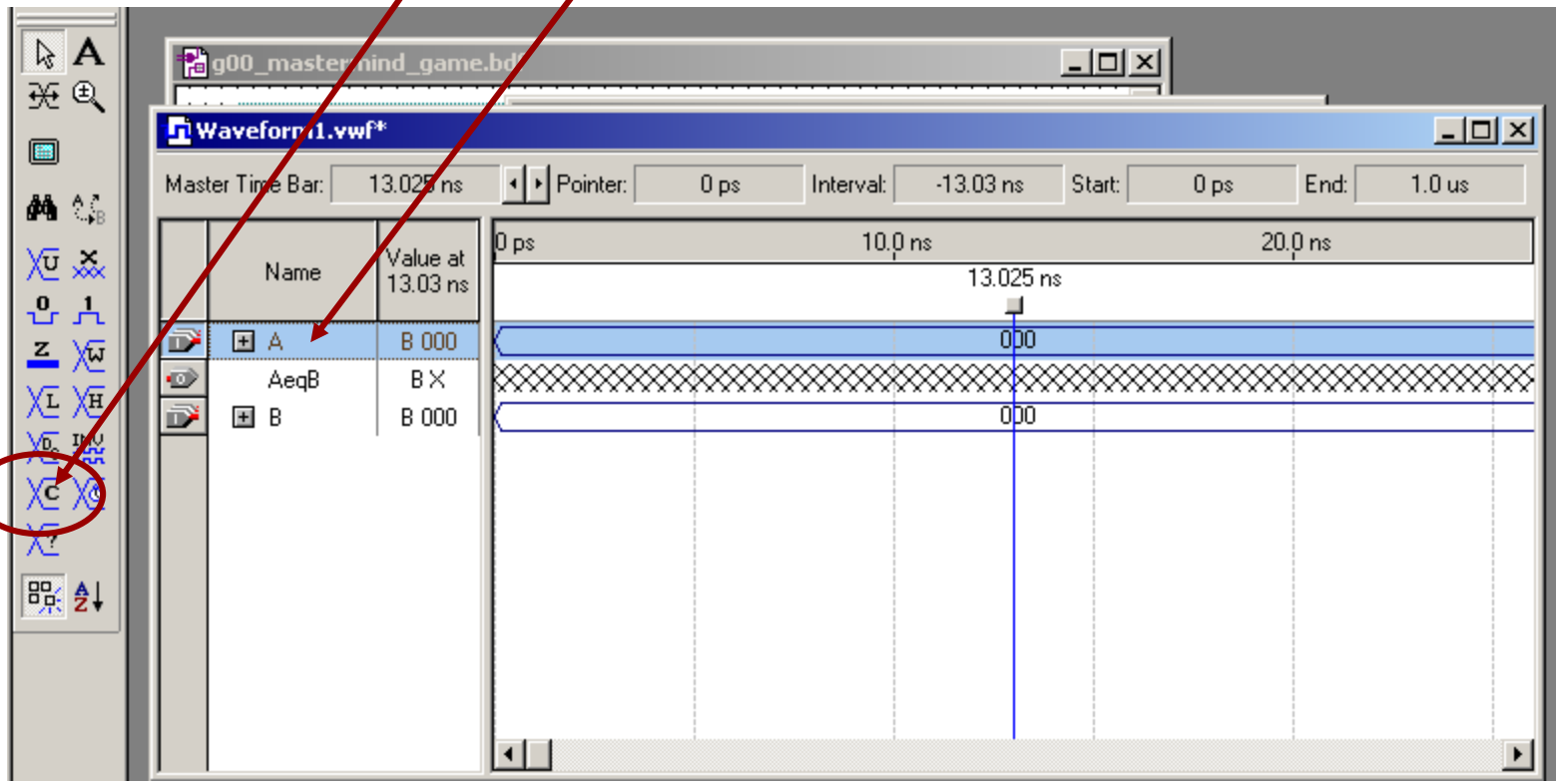
Set the time to 655360 ns 655360 ns (which is 2^14 x 40ns).



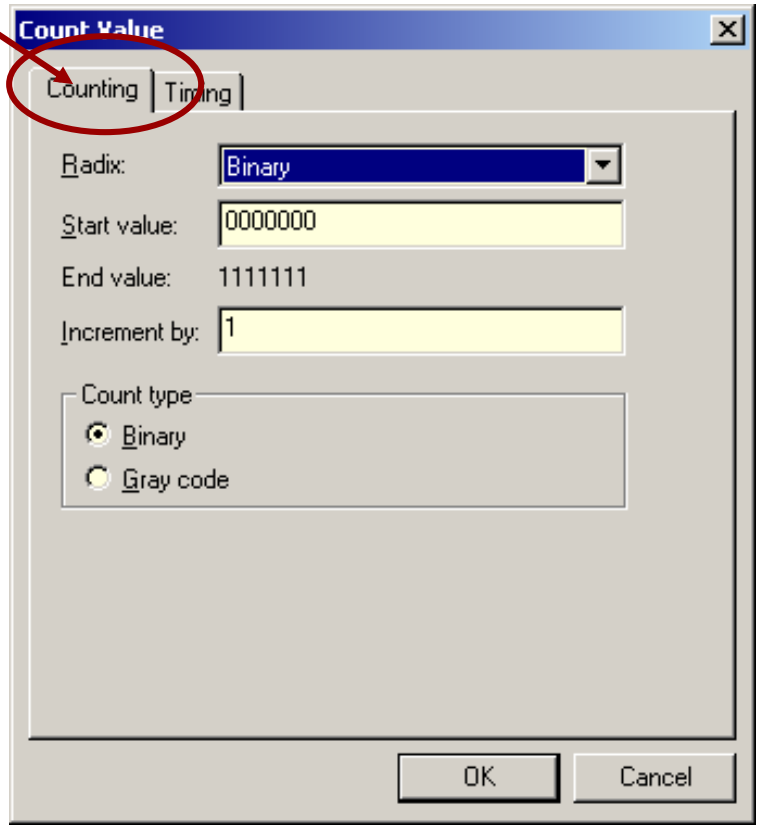


Next, select bus A by left-clicking on its name in the left-hand pane. When it is selected, the entire row will be highlighted.

Now, click on the "Count Value" button in the left-hand vertical toolbar. This is the button with the "C" inside of a pair of crossed lines.
This will allow you to fill the waveform for A with a binary count sequence.

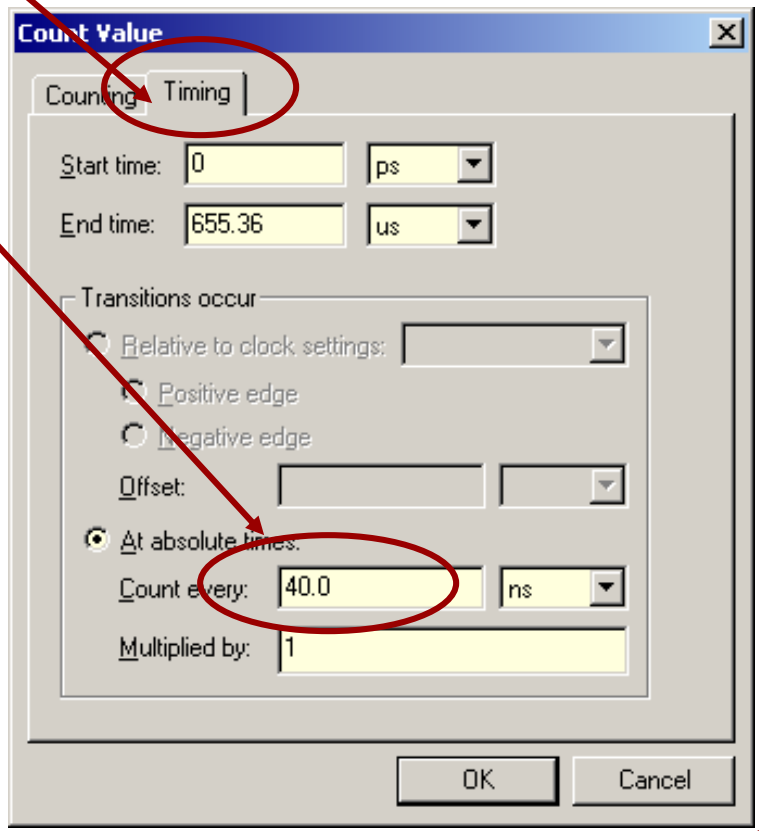


Clicking on the "Count Value" toolbar button brings up a dialog window with two tabs. Select the "Counting" tab. The window shown in the figure below will appear. Set the "Start value" to 0000000 and the "Increment by" value to 1 as shown (these are the default values).

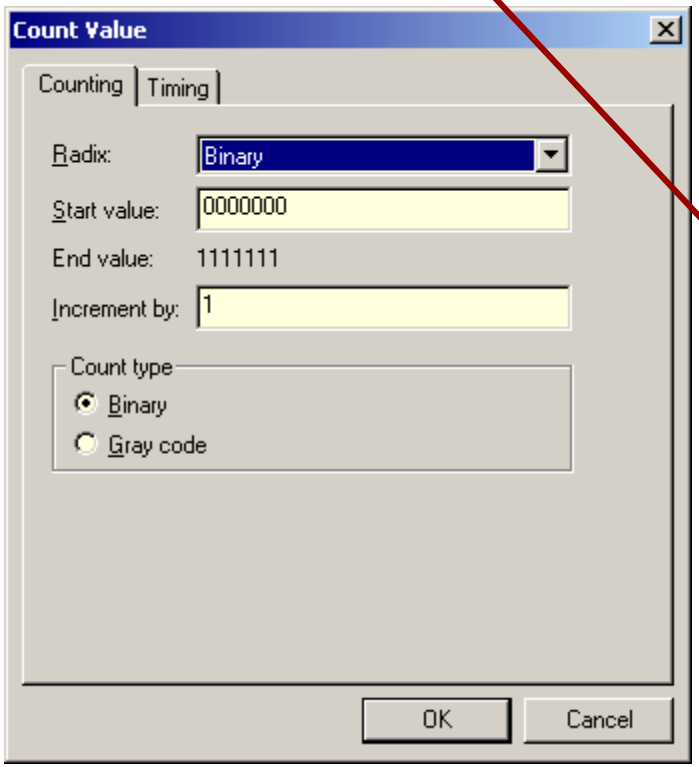


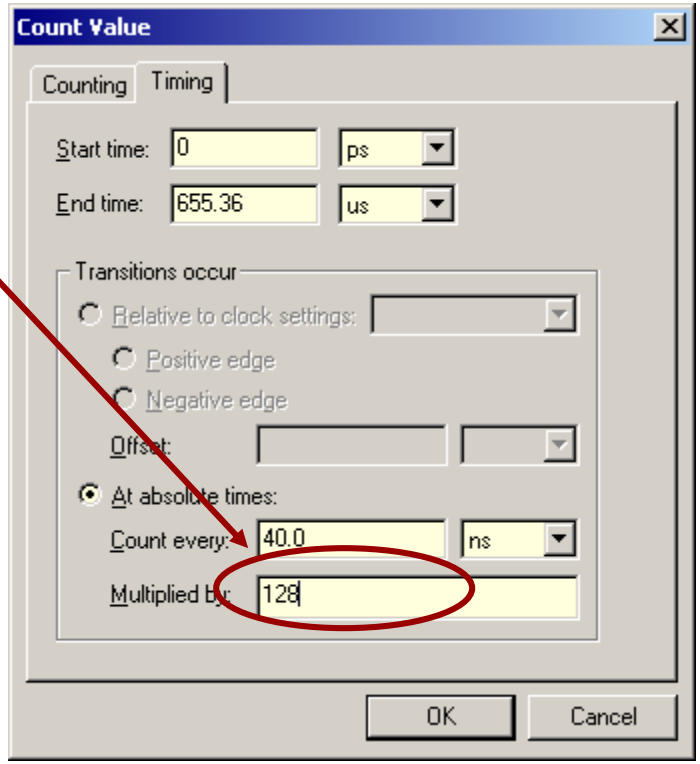
Then select the "Timing" tab. The window shown below will appear. Set the "Count every" setting to 40.0 ns as shown. This will create a waveform which changes every 40 ns.

Click on OK to finish.

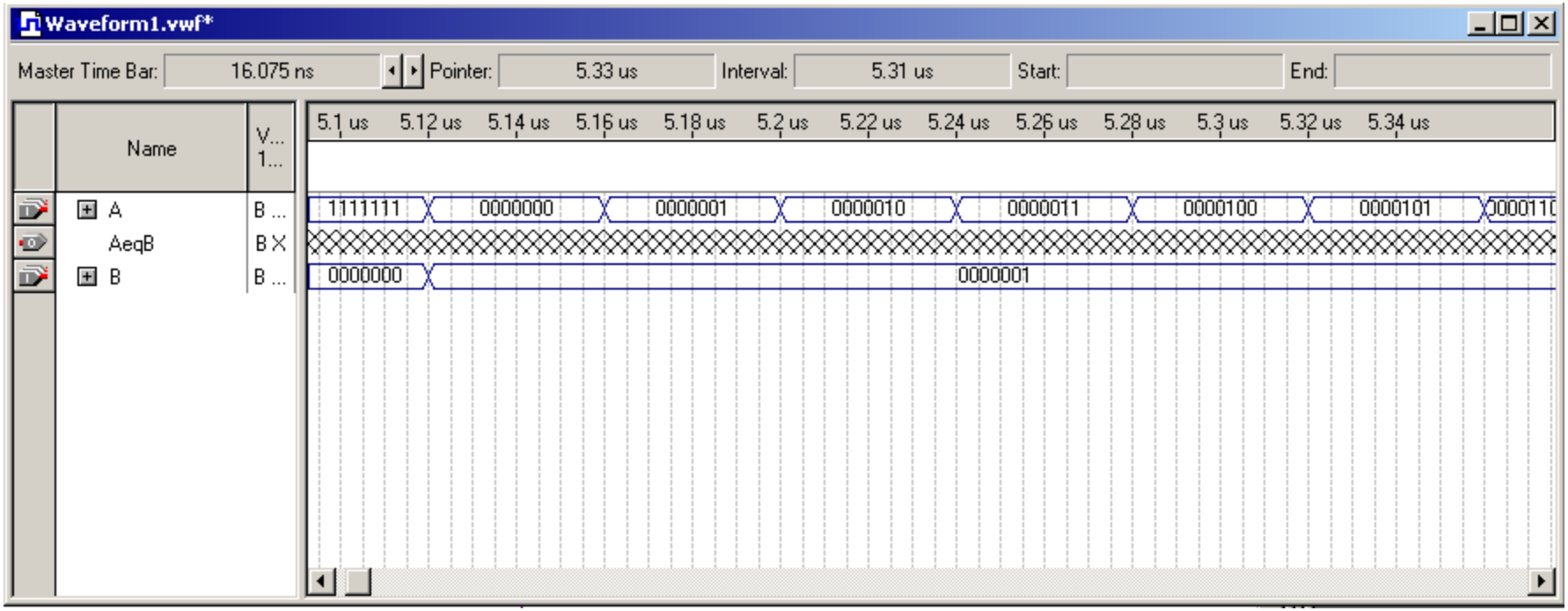


Repeat for bus B. The settings should be the same, except that in the Timing tab view, set the "***Multiplied by***" setting to ***128***. This means that the count sequence for B will change every 5120 ns, which is 128 (2^7) times slower than for A. In this way you get a sequence of the 2^14 possible different values for the combination of A and B.



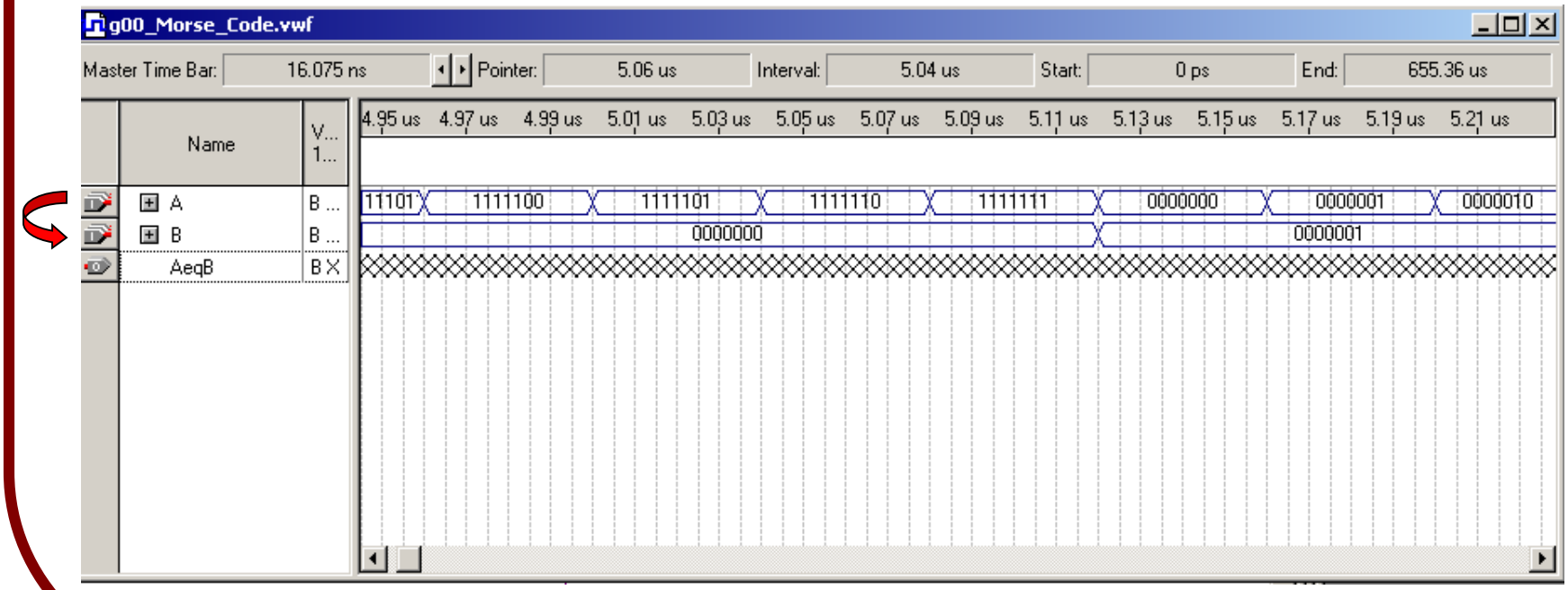


After specifying the count values, the waveforms for A and B will look similar to that shown in the figure below. Note that B changes more slowly than A, as desired, and that all 16,384 possible combinations of the values of A and B are produced.



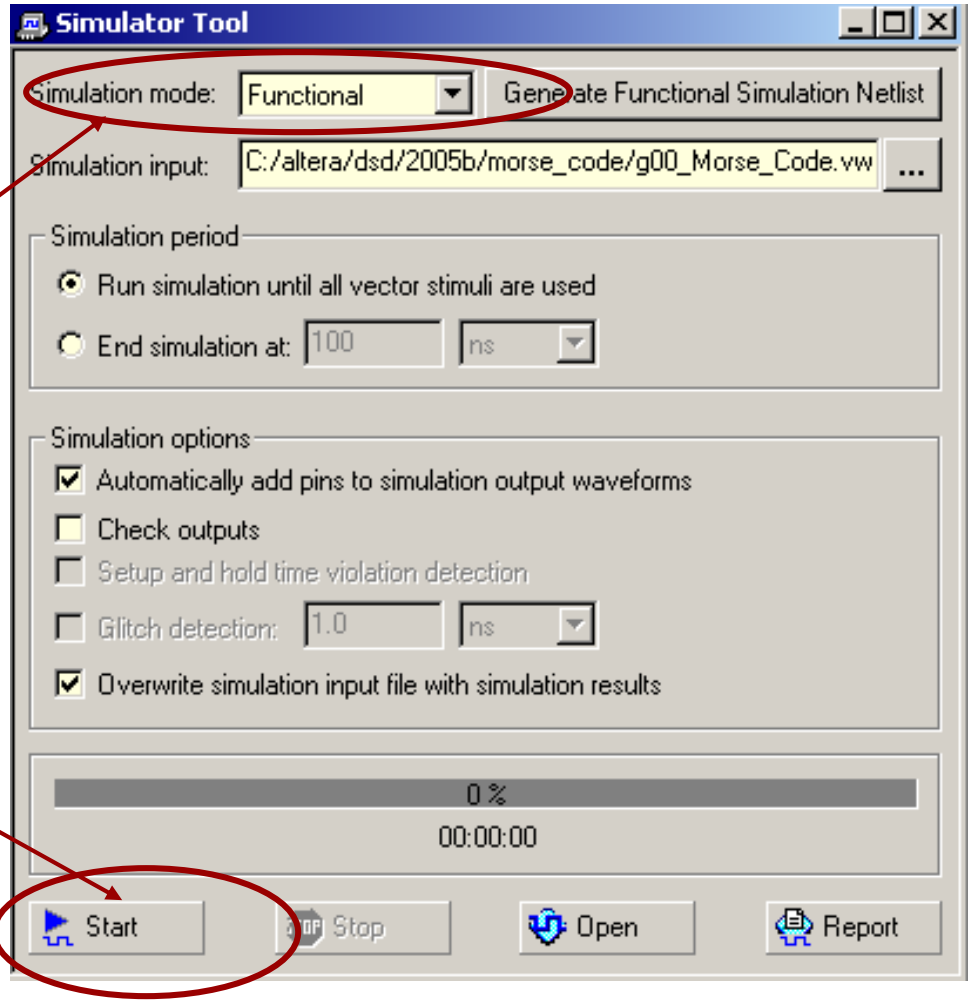
Select "Save As" from the file menu, and save the waveform file as "gNN_lab1.vwf".

We can rearrange the ordering of the waveforms displayed in the simulation window. This is done by selecting the waveform to be moved, then by left-clicking on it, and dragging it to the desired new position, while keeping the mouse button held down. In the diagram below, we have swapped the positions of waveforms B and AeqB, so that waveforms A and B are adjacent.

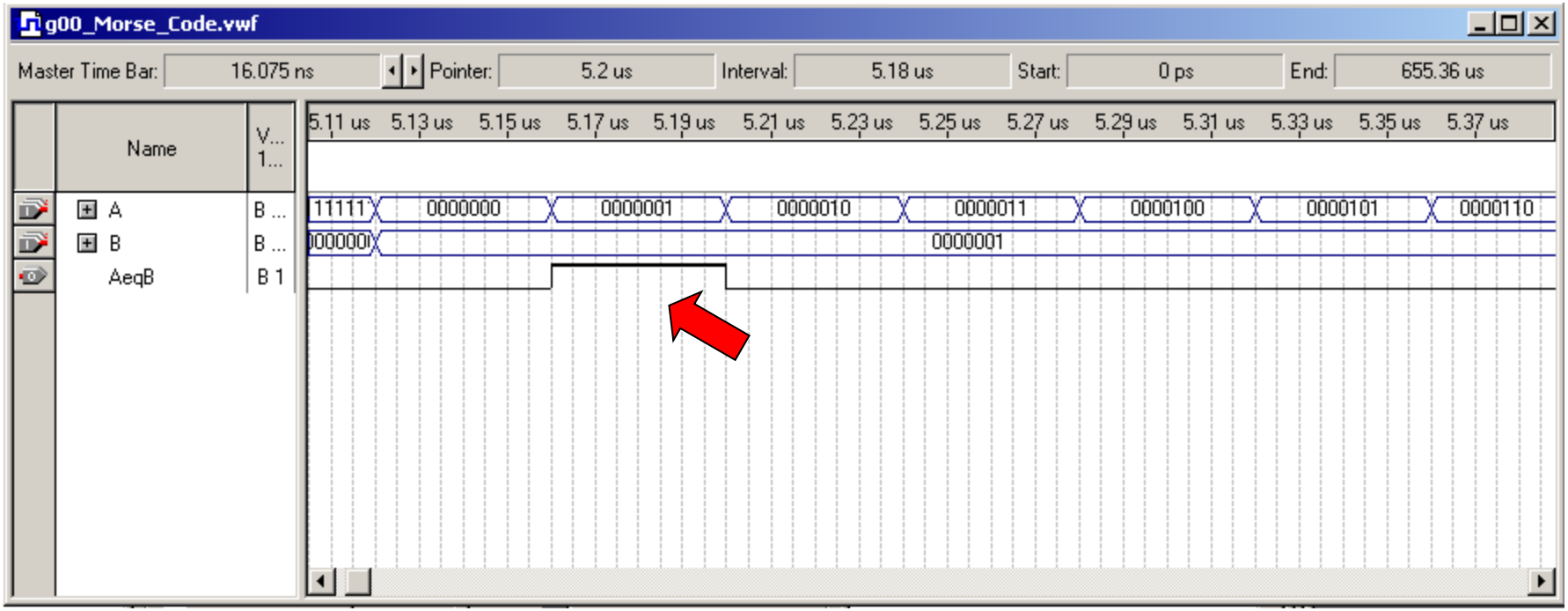


Once your waveforms have been setup, return to the Simulator Tool window. Make sure that the Simulation mode is set to "***Functional***", then click on the Start button to run the simulation.

(Note: you may have to re-run the "Generate Function Simulation Netlist" before starting the simulation)



After the simulation finishes, the output node AeqB, which was previously undefined, should now be filled in with properly defined values. Look at the values for AeqB and see that they are correct, given the inputs A and B (i.e. it should be high when A=B)



Show your simulation results to the TA and have him sign your grade sheet

4. Writeup the Lab Report .

Write up a short report describing the ***gNN_Modulo_13*** circuit that you designed in this lab. This report should be done in html or pdf, or in Microsoft Word (***pdf format is preferred!***).

The report must include the following items:

- A header listing the group number (and company name if you have one), the names and student numbers of each group member.
- A title, giving the name (e.g. ***g00_Adder_16_bit***) of the circuit.
- A description of the circuit's function, listing the inputs and outputs. Provide a pinout or symbol diagram.
- A gate level schematic diagram of the circuit.
- A discussion of how the circuit was tested, showing representative simulation plots. How do you know the circuit works correctly?

5. Submit the Lab Report to WebCT .

The lab report, and all associated design files must be submitted, as an assignment to the myCourses site. Only one submission need be made per group (both students will receive the same grade!).

Combine all of the files that you are submitting into one *zip* file, and name the zip file gNN_LAB_1.zip (where NN is your group number).

Grade Sheet for Lab #1 Winter 2018.

Group Number:_____.

Group Member Name:_____. Student Number:_____.

Group Member Name:_____. Student Number:_____.

Marks			TA Signatures
	1.	VHDL File for 1-bit Adder	_____.
	2.	Waveform Vector File display for the 1-bit Adder	_____.
	3.	VHDL File for 1-bit Subtractor	_____.
	4.	Waveform Vector File display for the 1-bit Subtractor	_____.
	5.	Demonstrated Adder functioning with TA selected input	_____.
	6.	Demonstrated Subtractor functioning with TA selected input	_____.

Each part should be demonstrated to one of the TAs who will then give a grade and sign the grade sheet. Grades for each part will be either 0, 1, or 2. A mark of 2 will be given if everything is done correctly. A grade of 1 will be given if there are significant problems, but an attempt was made. A grade of 0 will be given for parts that were not done at all, or for which there is no TA signature. Make a copy of the signed form for yourself.