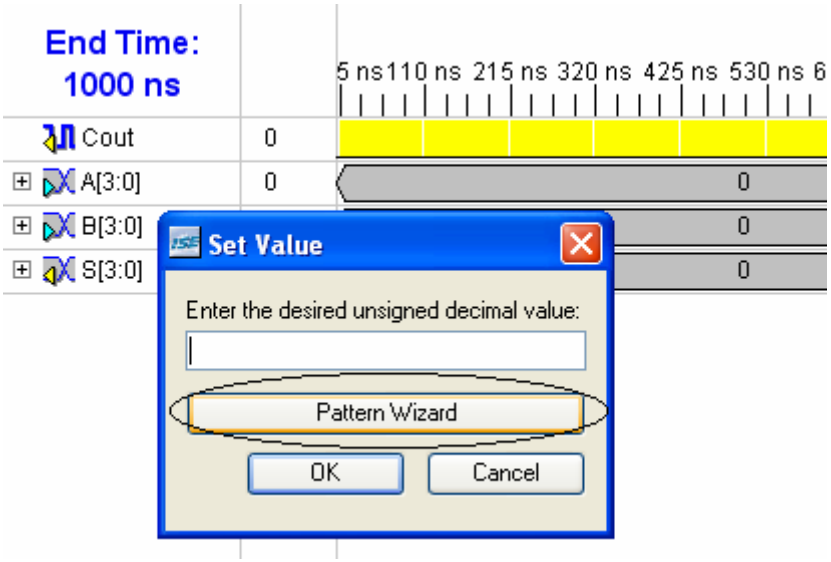
University of Pennsylvania
Department of Electrical and Systems Engineering

Digital Design Laboratory

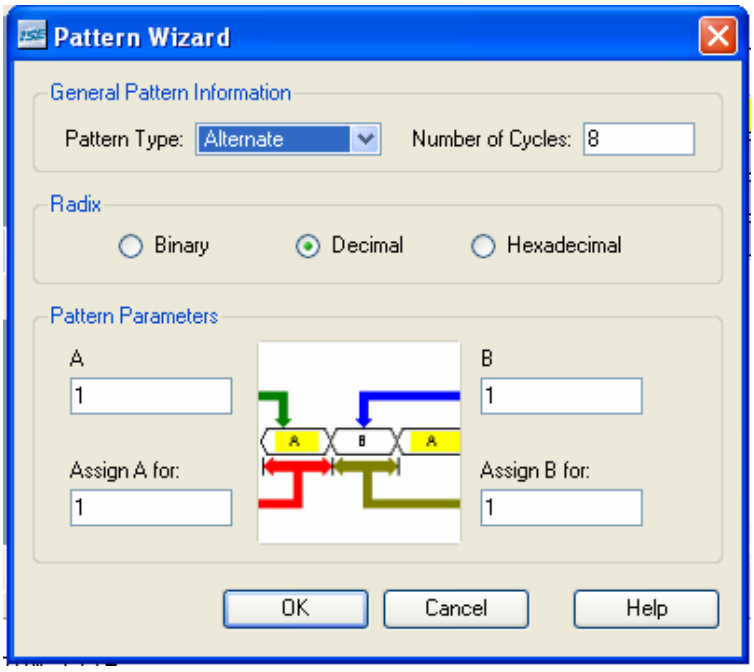
Tutorial: Pattern Wizard

When assigning values to a bus in Xilinx during the behavioral simulation, you should use the pattern wizard to help you ease the process. The pattern wizard let you assign binary or decimal values to a group of signals and it is best used for multiple bit arithmetic. In this tutorial we will go through how to use this Xilinx function by testing a four-bit adder (with inputs A and B buses).

To open up pattern wizard, click on the A bus and click on Pattern Wizard:



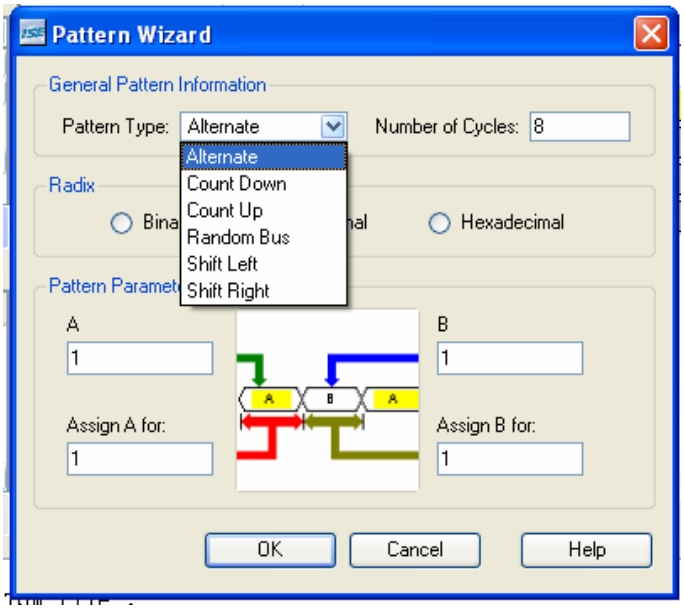
You will enter a screen that looks like this:



This is the screen that you will set all the parameters for your pattern.

1. General Pattern Information

Under the general pattern information there are two fields. The first one is called Pattern Type and it is a pull-down menu that offers six choices:



Alternate means you want the value to go back and forth between two specific values (0 and 1 for example). Count up and count down goes through a series of values from a beginning value and a terminal value. Random bus generates a random number without any specific pattern. Shift let you shift the bits in a specific direction. You will learn about the shift function toward the end of the course. In this specific example, because we are doing binary addition, we would want to choose the Count Up pattern type and let it count in decimal for us.

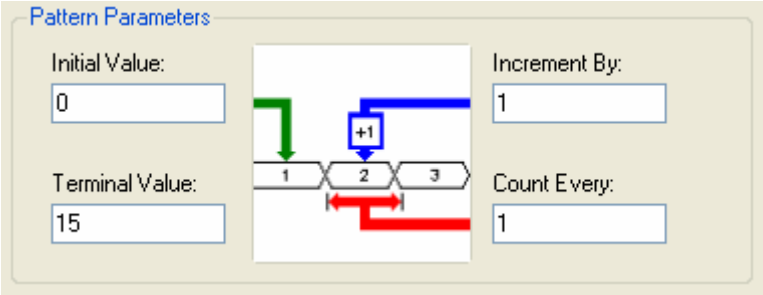
The second field is called the Number of Cycles. This is the number of times you want the pattern wizard to change values. For example, if you want it to count from 0 to 9 (10 numbers in total), you must set this number to at least 10 or else it will not count up all the way. If you set the number to more than 10 it will simply stop at the terminal value. The number of cycles is more important when you choose an alternating pattern type, where the numbers would alternate according to the number of cycles. So we will just set this value to 10 for our purpose.

2. Radix

In this section you can select which number system you wish to use for the pattern wizard, this would let the number you can enter in the next section: pattern parameters. However, Xilinx set Unsigned Decimal as the default display for buses so changing the radix has no effect on display (you can change the way numbers are displayed by right click on the bus in the waveform window). Because decimal is the most intuitive number system, we will use decimal for this simulation.

3. Pattern Parameters

Pattern Parameters varies with the pattern type you selected. So after you select Count Up under pattern type, the pattern parameter section should look like this:

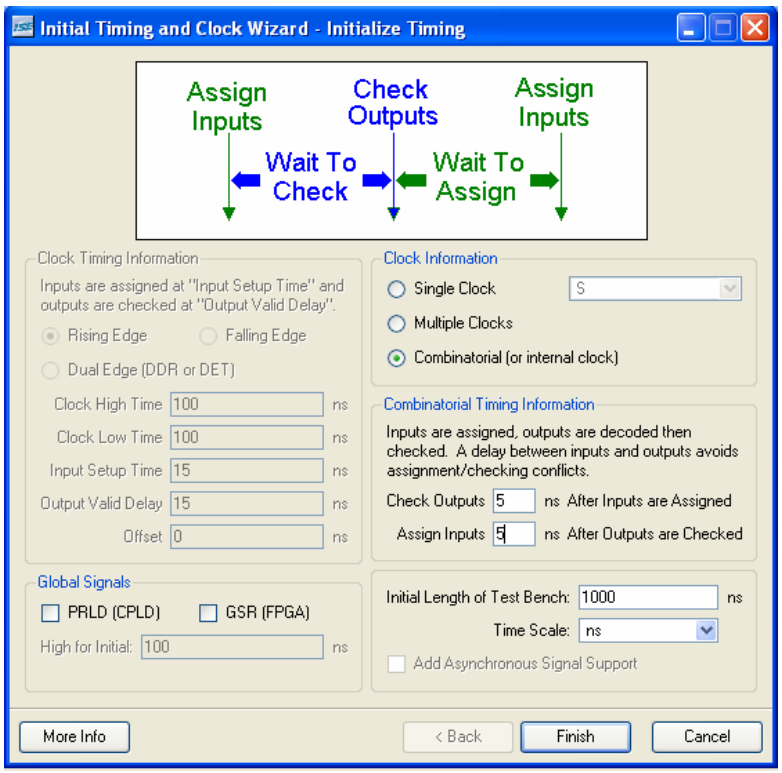


Initial and Terminal values are simply how long and how low you want the numbers to go. Because we are testing our BCD adder, we want to set the initial value to 0 and terminal value to 9.

The "Increment By:" field let you set the difference between adjacent numbers so you can count in a multiple of a number. You can set a terminal value will not necessarily appear on the waveform if you are incrementing by a multiple that doesn't include the terminal value. For example, if you want to increment by 2 starting with 0, then even though you can set your terminal value to 9 and last number that is counted will be 8 and 9 will not appear. Counting will automatically stop at the last number it counted before reaching the terminal value and will never go above it.

The last field is called "Count Every:" and it indicates how long you want each cycle to be. This has to be in a multiple of your combinatorial clock time. To understand this value, let's return to the screen when you first set up the waveform.

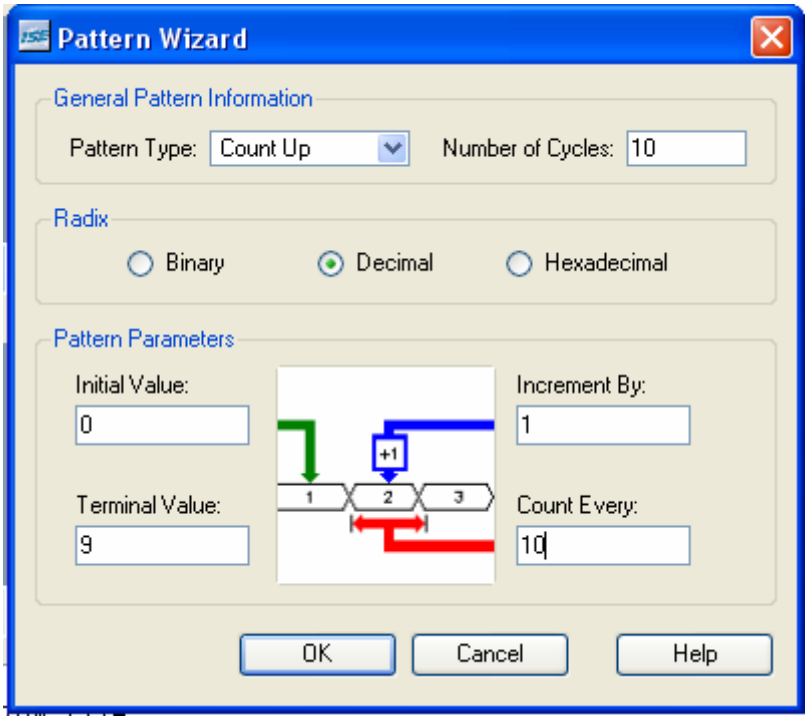
If you remember, sometime during adding a test bench waveform to your project you come across a screen that looks like this:



In the Xilinx tutorial, you were told you should select the Clock Information to be Combinatorial and set Timing to 5ns for both checking outputs and assigning inputs. This means that the combinatorial clock cycle repeats itself after 10ns (5ns for checking + 5ns for assigning).

Now let's return to Pattern Wizard, the "Count Every:" means how many combinatorial clock cycles do you want to wait before inputting the next value in your pattern. If you put down 1 in this field, the next value will be inserted after 10ns the previous one was inserted. Because our test bench is 1000ns long, changing the input every 10ns would be impossible to read and unnecessary. If we set the value to 10, then the inputs would be changed every 100ns, allowing 10 different inputs for the entire time we have and it is a must more reasonably number. So let's set the value to 10.

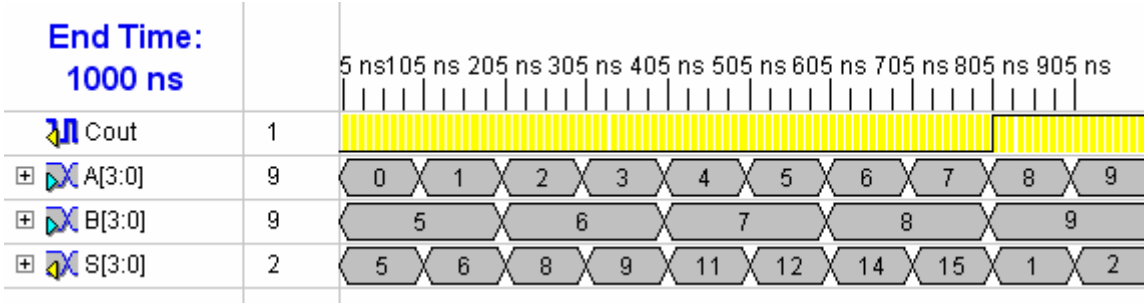
Now you have finished configuring your pattern, the pattern wizard window should look like this after you are done:



Then click OK. You should see the bus is counting from 0 to 9 every 100ns.

Let's repeat the step for B. This time, instead of making it count from 0 to 9, let's make it count from 4 to 9 every 20 clock cycle to generate some more patterns for us to test our circuit.

Then generate the simulated results. You output should look like this:

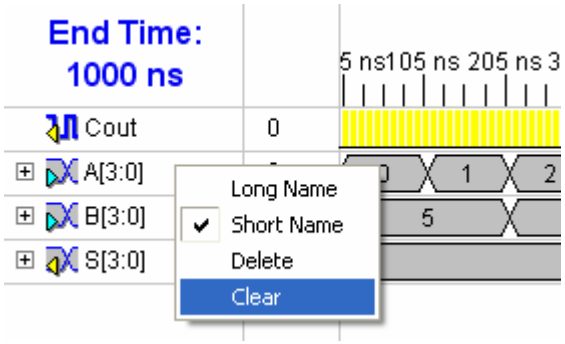


As you can see, Cout went high when the result went above 15.

Clearing Patterns:

In Xilinx you can overlap one pattern over another and it is generally rather confusing and unless you are considering yourself an expert on pattern wizard, we generally recommend clearing out the entire pattern and reconfigure a new one if you wish to change your inputs.

To clear a pattern (or any waveform for this matter), first left click on the bus name to select the bus and then right click on the name and select clear, then you can just reconfigure a new pattern. If you add a new pattern to a bus that already has a pattern there, Xilinx will automatically replace the part where you want the new pattern to be and leave the rest of the old patterns there (It will NOT set it to zero!)



A Note of Caution: If you are assigning a pattern to a bus that had a pattern there or cleared, you MUST click on the bus where you want the new pattern to start. For instance, if you want the pattern to start from the beginning, you must click within the first combinatorial clock. If you click on the middle of the bus, the pattern will start at exactly the point you have first point your mouse to and will leave the first part of the bus unchanged!