

# Fast and Easy Digital Simulation

©SynaptiCAD Inc.

This Appendix introduces a new type of simulation technology, called Interactive Simulation, which allows engineers to begin to simulate and analyze design ideas before a complete simulation model or schematic is available. This technology is particularly useful for students, because they can quickly enter Boolean and RTL equations to check equivalency or to experiment with flip-flops and latch designs.

Designers with an Interactive Simulator have two major advantages over engineers using traditional gate-level and HDL simulators. First, the Interactive simulator works with circuits at the same level of abstraction (Boolean equations, delay paths, timing requirements) as engineers do when they first start formalizing their design. This means designers can rapidly enter the essential elements of their design without having to create schematics or HDL models. The second major advantage is unlike most EDA tools, the Interactive Simulator is *fully* interactive - it provides instant feedback to designers as changes are made to the design. The combination of these two features, rapid design entry and rapid feedback on system functionality and performance, makes the Interactive Simulator the ideal tool for the iteration intensive process of designing digital circuits and examining design tradeoffs.

## 1.0 Getting Started

This Appendix is in the form of a tutorial that will demonstrate the drawing features, and the Boolean and RTL equation interface using SynaptiCAD's WaveFormer Pro or VeriLogger Pro products.


Install either the VeriLogger Pro or WaveFormer Pro evaluation version from the CDROM in the back of this book. If the CD is missing you can download a version from the [www.syncad.com](http://www.syncad.com) site. Both products contain the interactive simulator that is used in the following tutorial. VeriLogger Pro also contains a traditional Verilog simulator that can compile and simulate the Verilog examples on the CDROM.

## 2.0 Sketching Timing Waveforms

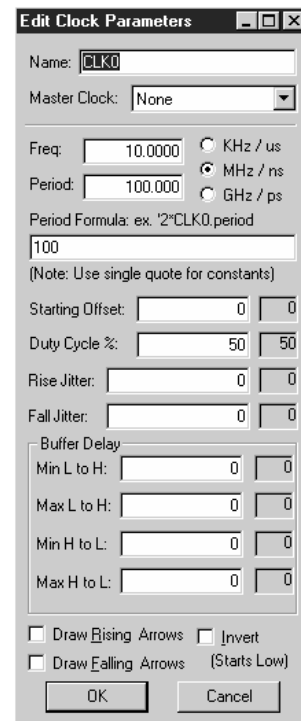
Design specifications are usually in the form of timing diagrams, waveforms and timing requirements. So the Interactive Simulator ships with a timing diagram editor that lets you sketch waveforms and perform timing analysis.

### 2.1 Add a Clock

Clocks are special repetitive signals that draw themselves based on their attributes: period, frequency, duty cycle, edge jitter, offset, and other parameters. To add a clock:


- Press the **Add Clock**  button, located in the top left hand corner of the Diagram Window. This causes the *Edit Clock Parameters* dialog to open.
- We will use the default settings of a 100ns period and 50% duty cycle to define the clock for this example. Click the **OK** button to add the clock and close the dialog.

Note: For more information on clocks, master clocks, clocks with formulas, and clock grids read *Chapter 2: Clocks* in the on-line help (select the **Help > Timing Diagram Editor Table of Contents** menu option). If you made a mistake designing the clock, then double click on a clock segment to reopen the *Edit Clock Parameters* dialog box. Double left clicking on a clock edge opens up the *Edge Properties* dialog, which displays the edge time. You may also reach the *Edit Clock Parameters* dialog by double clicking on the clock name and choosing the clock properties button in the *Signal Properties* dialog.





## 2.2 Add Signals

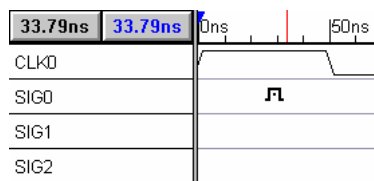
Next we will add three signals that we will use to experiment with the waveform drawing features. To add the signals:



- Press the **Add Signal**  button three times to add three signals with empty waveforms.

## 2.3 Drawing Signal Waveforms

Next, we will draw some random waveforms to become familiar with the drawing environment.


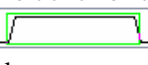

1. Notice the buttons with the waveforms drawn on them . These are the State Buttons. The active button is colored red and indicates the type of signal state that will be drawn next. In this case, the HIGH signal state is active .
2. Move the mouse cursor to inside the Diagram window at the same level as the signal name SIG0, and at about 40ns.



3. Left click to draw a waveform segment from 0ns to the cursor. Notice that a HIGH signal was created.
4. A different state button is now activated. The State Buttons automatically toggle between the two most recently activated states. The small red T above the signal name denotes the toggle state, for instance, .
5. Move the cursor to about 80ns on the same signal and left click. Now a LOW segment is drawn from the end of the HIGH signal to the location of the cursor.
6. Left click on the TRI button  to activate the tristate State Button and draw another waveform segment.
7. Draw more segments, using all the states except the HEX button. The HEX state button is used in defining multi-bit signals, and for signals with a user defined VHDL type. This button is covered in later tutorials. For the time, experiment with the graphical states.

## 2.4 Edit signal waveforms

There are four main editing techniques used to modify existing signals (Note: these techniques will not work on clocks). The most commonly used technique is the dragging of signal transitions to adjust their location. The other three techniques all act on signal segments, the waveforms between any two consecutive signal transitions. The segment waveform can be changed, deleted, or a new segment can be inserted within another segment. Use each of the following techniques:

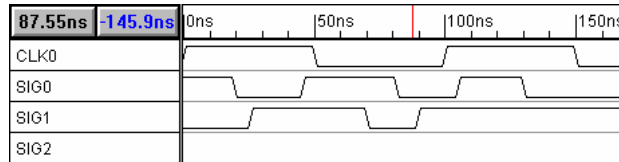
- **Move a signal transition:** Left click down on a signal transition and drag it to the desired location. A green bar will appear that follows the mouse cursor . Release the mouse button when the green bar is at the location where you wish to place the transition.
- **Change the state of a segment:** A segment is the waveform between two consecutive signal transitions. Left click on the segment to select it (a selected segment has a highlighted box drawn around it) . Then left click on the State Button of the new state desired. Note: If you try to select a narrow segment and one of the transitions gets selected, widen the segment by clicking the **Zoom In** button , located on the right hand corner of the button bar.
- **Delete a segment:** Select a segment (see above) and then press the delete key on the keyboard.
- **Insert a segment:** Inside a large segment, hold the left mouse button down and drag to the right then release. A new segment will be added in the middle of the original segment. For this operation to work the original segment must be wide enough to be selected.

These techniques will not work on clocks. This is because clocks have fixed edges and segments. To edit a clock, double click on a segment of the clock waveform in the Diagram window to open the *Edit Clock*

Parameters dialog. All clock parameters can be changed in this dialog box. *Chapter 1: Signals and Waveforms* in the on-line help has more information about drawing and editing waveforms.

### 2.5 Adjust your diagram so it resembles the figure below

Now use the above techniques to edit the signals so they have roughly the same transitions as the signals in the figure below. This is not the normal way to create a timing diagram, but it will teach you how to use the waveform editing features. Make sure you try all the editing techniques.



For this tutorial we will only discuss the drawing features. But for a real design you would also want to analyze circuit timing and would also want to add delays, setups, and holds to your design data. These concepts are covered in the *Basic Drawing and Timing Analysis* tutorial (select the **Help > Tutorials** menu option).

### 3.0 Simulating Boolean Equations

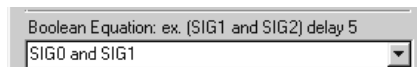
The Interactive simulator accepts Boolean equations in either VHDL, Verilog, or SynaptiCAD's enhanced equation syntax. The SynaptiCAD format supports the following operators (symbolic operators follow the operators that they represent):

and, &	xor, ^
or,	not, ~, !
nand	delay
nor	

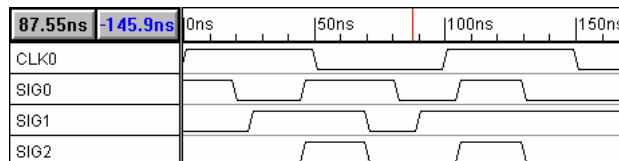
#### 3.1 Simulate a Boolean equation

We will begin by simulating a Boolean Equation.

1. Double click the **SIG2** signal name to open the *Signal Properties* dialog. Arrange the Signal Properties dialog so that you can see the dialog and the 3 signals at the same time. This dialog is modeless, so leave it open during this section. All controls and buttons used in this section are contained in the *Signal Properties* dialog.
2. Make sure the **Boolean Equation** radio button  Boolean Equation is selected.
3. Type the following equation into the Boolean equation edit box (signal names are case sensitive): **SIG0 and SIG1**. This causes the waveform of SIG2 to be the results of SIG0 and SIG1 passing through an AND gate.



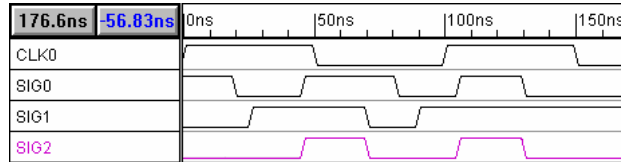
4. Click the **Simulate Once** button and watch the signal draw itself. Notice that SIG2 is the result of the Boolean Equation "SIG0 and SIG1". By default, the **Simulate** radio button is not checked, so if you moved an edge on SIG0, SIG2 is not automatically re-simulated.



#### 3.2 Continuously Simulate the Boolean Equation

Next we will setup the equation to automatically re-simulate whenever a change is made to an input signal.

1. Check the **Simulate** radio button. Notice that the SIG2 is now drawn in purple. This color means that the signal is being continuously simulated, and changes to the waveforms of SIG0 and SIG1



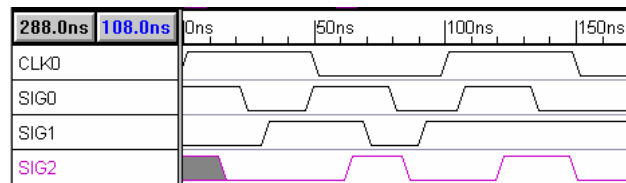
will cause a new simulation to occur automatically. If you are using VeriLogger Pro, make sure that the program is in the **Auto Run**  simulation mode. The simulation mode button, which toggles between the Auto Run and Debug Run modes, is located on the simulation toolbar, just below the main application menus. Debug Run mode will not continuously update signals.

2. Move some of the edges on SIG0 and SIG1 and watch SIG2 re-simulate. (Notice that you cannot drag and drop SIG2's signal edges because they are calculated edges).

### 3.3 Simulating Boolean Equations with Delays

Next we will modify the Boolean equation to take into account the propagation delay through the AND gate. First we simulate a simple 15ns delay:

1. Enter one of the following Verilog, VHDL, or SynaptiCAD equations into the **Boolean equation** edit box of SIG2:  
**#15 (SIG0 & SIG1)**  
**(SIG0 and SIG1) after 15**  
**(SIG0 and SIG1) delay 15**
2. Click the **Apply** button and verify that SIG2 is correctly drawn. The diagram from section 3.2 would be changed to the diagram shown below, for instance.



### 3.4 Simulate a true min/max delay using SynaptiCAD syntax:

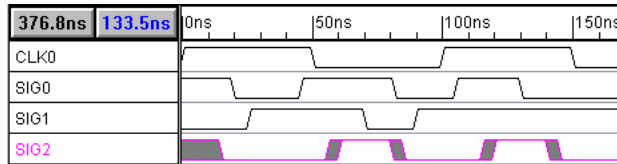
Next we will use the delay operator to simulate a true min/max delay. The propagation delay for a particular part will vary with temperature, loading conditions, and manufacturing process variations. The delay variations are listed on the data sheets of each part as the min and max times (sometimes only max times are given). Often the min/max delays are different for each possible state transition combination a part might go through (high-to-low, low-to-high, low-to-tristate, tristate-to-low, high-to-tristate, and tristate-to-high). If you don't know whether or not you will have low-high or high-low transition you must use the extremes in calculating the timing (smallest min time and the largest max time).

As a design engineer you will need to make sure that your circuit will perform correctly given any combination of valid delays. Keeping track of each delay by hand can become a nightmare for all but the simplest circuits. The Interactive Simulator can simulate the true min/max timing using the delay operator.

The **delay** operator takes a signal on the left, and a time or parameter name on the right, and returns a signal. A parameter is a construct used by the simulator to represent timing information. If a parameter name is used on the right hand side of the delay operator, then the equation will simulate true min/max timing. This true min/max timing is the main advantage that SynaptiCAD's format has over the VHDL or Verilog format. In this section we will add a parameter and then reference it in the delay equation.

1. Click the **Add Free Parameter**  button in the Parameter Window. This causes a free parameter named **F0** to be added to the Parameter Window.
2. Double click on the name of the free parameter (F0, in the Parameter window) to open the *Parameter Properties* dialog.

3. Set the min time to **10** and the max time to **15**. Push the **OK** button to close the *Parameter Properties* dialog.
4. Modify the **Boolean Equation** of SIG2 in the *Signal Properties* dialog so that it reads: **(SIG0 and SIG1) delay F0**. By referencing the free parameter F0 in the equation, the min and max propagation delay of the AND gate is incorporated into the waveform of SIG2.
5. Click the **Apply** button to apply the changes. A new simulation will be performed. Notice the gray uncertainty regions on the waveform of SIG2. This true min/max timing is the main advantage that SynaptiCAD's format has over the VHDL or Verilog format.



### 3.5 View the HDL code that models the Boolean equation:

The Interactive Simulator takes the Boolean and RTL equations and generates Verilog or VHDL code for a complete circuit model. The generated Verilog is the actual code that is used by the underlying HDL simulator. You can view the code that is generated for each equation using the *Signals Properties* dialog.

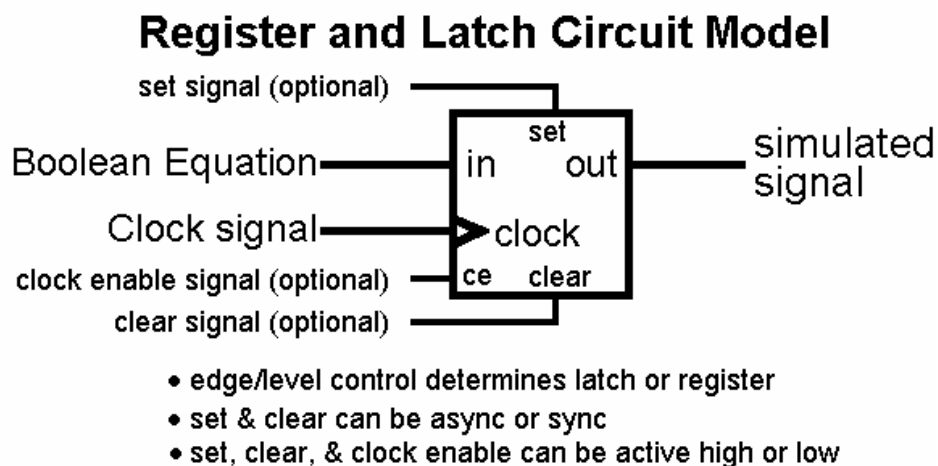
1. Click the **Verilog** or **VHDL** radio button  Verilog  VHDL to view the HDL code that simulates the Boolean equation. If you wanted to, you could add native HDL code here to perform a special function. Do not modify the code now. The code should resemble the following Verilog example:  

```
wire # F0_min SIG2_wf0 = (SIG0 & SIG1);
wire # F0_max SIG2_wf1 = (SIG0 & SIG1);
assign SIG2 = (tb_status[0] == 1'b1) ? (SIG2_wf0 === SIG2_wf1) ? SIG2_wf0 : 'bx : 'hz;
```
2. Click the Boolean Equation radio button to display the Logic Wizard section (or Boolean Equation Section) of the *Signal Properties* dialog.
3. Leave the *Signal Properties* dialog open. We will be using it in the next section.

Note: This example demonstrated true min/max simulation, however **Min-Only** and **Max-Only** simulations can be performed by changing the timing model drop down box. You can find this under the **Options > Diagram Simulation Preferences** menu option. This will open the *Diagram Simulation Preferences* dialog box, and the drop-down list box is in the upper right corner titled **Timing Model**.

### 4.0 Registered and Latched Signals

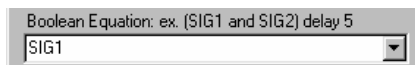
The Interactive Simulator can register or latch the result of a Boolean equation. Figure 4.0 represents the circuit that is modeled.



**Figure 4.0:** Register and Latch circuit modeled by the Interactive Simulator

The *Signal Properties* dialog should still be open and displaying the **SIG2** information from the last section. Let's experiment with the register and latch functions:

1. Enter the equation **SIG1** into the **Boolean Equation** edit box of the *Signal Properties* dialog.



2. Click the **Simulate Once** button to simulate the equation. SIG2 should look like an exact copy of SIG1. When we register SIG2 you can visually compare it to SIG1 to see the effects of the register.
3. Next use the **Clock** drop-down list to choose **SIG0** as the clocking signal. The clocking signal can be any clock or signal in the timing diagram (the default value "Unlocked" means no flip-flop is present).
4. Next use the **Edge/Level** drop-down list box (on the right side of the dialog) and choose **both** as the trigger signal.
5. Click the **Simulate Once** button to simulate the circuit. Notice that SIG2 only transitions when SIG0 has a positive or negative edge transition (move some edges on SIG0 and SIG1 to verify this).



Whether a Register or a Latch is simulated depends on the type of triggering in the **Edge/Level** list box. For a Register circuit choose **neg** for negative edge triggering, **pos** for positive edge triggering, and **both** for positive and negative edge triggering. For a Latch circuit, choose either **low** or **high** level latching.

6. Choose different **Edge/Level** values and press the **Simulate Once** button to verify the operation of the register and latch functions.

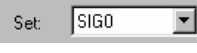

## 5.0 Set and Clear lines

The **set** and **clear** features are useful when defining circuits whose initial value needs to be specified. In this example we will demonstrate how to design a **divide by 2 circuit** using a negative edge triggered register with an asynchronous active-low set line. To specify the initial value:

1. Click the **Add Signal** button to create a new signal named **SIG3**.
2. Double click on the **SIG3** name to open the *Signal Properties* dialog.
3. Enter **!SIG3** into the **Boolean Equation** edit box (it references itself in the Boolean Equation).
4. Choose **CLK0** from the **Clock** drop down list box.
5. Make sure the **Edge/Level** setting is set to **neg**.
6. Check the **Simulate** radio button. Notice that SIG3 signal is completely gray but that status reports **Simulation Good** (lower right corner of the application window). This is because SIG3's Boolean equation references itself but it does not provide the simulator with a known start state.
7. Click the **Advanced Register** button to open the *Advanced Register and Latch Controls* dialog. All the register and latch individual propagation times, setup/hold constraints, clock enable, and set/clear options are set here. Global defaults for these properties can be set using the **Options > Simulation Preferences** menu.
8. Make sure the **Active Low** and the **Asynchronous** check boxes in the *Set and Clear* section are



checked. Click **OK** to close the dialog.

9. Choose **SIG0** in the **Set** drop down list box  of the *Signal Properties* dialog.
10. Click the **Simulate Once**  button. This button is located at the top left corner of the *Signal Properties* dialog box, under the **Name** edit box. Notice that SIG3 now has a simulated waveform. Experiment with SIG0 to see how the active low set line affects the operation of the flip-flop. You may want to redraw **SIG0** so that it goes **low** early in the timing diagram, and then stays **high** for 4 or 5 clock cycles.

The **Clock to Out**, **Setup**, and **Hold** edit boxes in the *Advanced Register and Latch Controls* dialog box accept time values for various timing constraints on the register and latch circuit. For more information on Register and Latch timing read the on-line help *Chapter 12: Interactive HDL Simulation*.

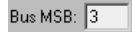

## 6.0 Multi-bit Equations

The Interactive Simulator can automatically generate multi-bit equations for the register, latch and combinatorial logic circuits. To convert a register or latch circuit into a multi-bit signal, change the MSB of the input signal and the MSB of the register or latch. If the sizes of the signals do not match, WaveFormer maps as many of the bits as possible, starting with the LSB.

First setup the diagram to experiment with multi-bit equations:

1. **Delete** the **SIG3** signal.
2. Click on the name of **SIG2** to select then signal we are going to copy.
3. Use the **Edit > Copy Signals** menu option to copy the signal.
4. Select the **Edit > Paste Signals** menu option to paste the signal. There are now two SIG2s in your diagram.
5. Rename the new SIG2 to **SIGX**. SIGX should have the exact same waveform as SIG2.

Next, change the output of SIGX to a multi-bit signal:

1. Double click on the **SIGX** signal name to open the *Signal Properties* dialog.
2. Make sure the **Simulate** radio button is selected.
3. Type **3** into the **MSB**  **LSB**  edit box of SIGX. This will make SIGX a 4-bit signal.
4. Click the **Apply** button. SIGX's waveform is now drawn as a bus with a 4 bit binary display. Only the LSB of SIGX is working because the input signal SIG1 is a single bit. Compare SIG2 and SIGX and verify that they are the same values.

Change the input signal SIG1 to a multi-bit signal:

1. Double click on the **SIG1** signal name to edit it the *Signal Properties* dialog.
2. Change the name of **SIG1** to **SIG1[3:0]**. Changing the name using the bracket notation has the same effect as changing the values in the MSB and LSB edit boxes.
3. Click the **Apply** button to accept the change. Now all four bits of SIGX should be toggling 1111 and 0000. If the radix is in Hex, the signal will toggle between 0 and F. The radix box is located in the lower left part of the Signal Properties dialog.

If you want to further experiment with multi-bit signals, change SIG1's graphical segments to valid regions instead of highs and lows. Then double click on a valid region to open the *Edit Bus State* dialog box. Type different 4-bit values, like 1010 or 0011, into the **Virtual** edit box and watch how it affects the output of SIGX.

## 8.0 Summary

Interactive Simulation differs from traditional HDL simulation in the amount of information that is needed to begin a simulation. With traditional simulation you must write both a complete design model and a test bench model to get simulation results that make sense. With the Interactive Simulator you can design *incrementally*: simulations are performed automatically without the need for complete design information. This allows you to catch mistakes earlier and reduces design time, leaving you more time to optimize your design.

By Donna Mitchell  
VP Marketing, SynaptiCAD Inc.