Xilinx Academy - Foundation Series training, part I

This tutorial demonstrates the Schematic-based design entry and the Foundation Series point tools. It also shows the functional simulator. This tutorial has been extracted from the F1.5 Quick-Start guide. The Quick-Start guide's In-Depth schematic tutorial is much more extensive, and you may wish to work on that tutorial later, if you need additional familiarity with the point tools.

Before working on this tutorial, make sure you have properly installed the Foundation Series tools and tutorials. Instructions for installation have already been provided to you, in the document called "Foundation Series training preparation."

Schematic-Based Design

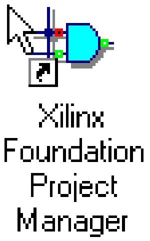
This tutorial guides you through a typical FPGA schematic-based design procedure using a design of a runner's stopwatch called "Watch". In the first part of the tutorial, you will use the Foundation design entry tools to complete the design. The design is composed of schematic elements, a state machine, a LogiBLOX component, and an HDL macro.

After the design is successfully entered in the Schematic Editor, it is ready for functional simulation with the Foundation Logic Simulator, implementation with the Xilinx Implementation Tools, timing simulation, and, finally, downloading and hardware debugging in a Xilinx FPGA on the FPGA Demonstration Board.

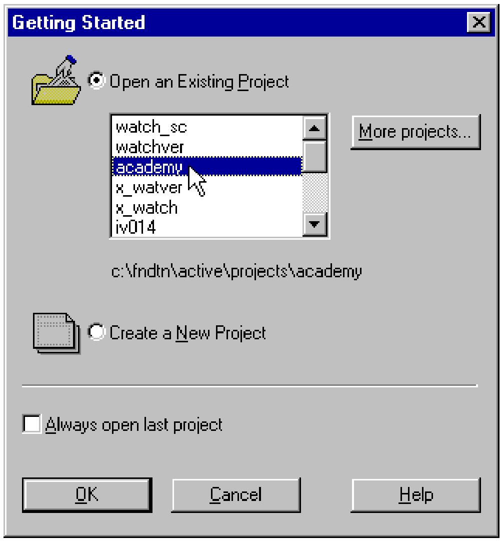
This tutorial will cover design entry and functional simulation. You will use the tutorial project provided by your instructor, and it should be installed in C:\fndtn\Active\projects\ACADEMY. The solutions project, WATCH_SC, contains all completed input and output files and is provided as one of the installed sample tutorials.

Starting the Project Manager

1. Double click the Foundation Series Project Manager icon on your desktop or select **`Start`** → **`Programs`** → **`Xilinx Foundation Series`** → **`Xilinx Foundation Project Manager`** from the Start menu.



2. A Getting Started dialog box displays, allowing you to select a project to open. If you have not opened this tutorial project before now, click the **`More Projects...`** button.



Getting Started

● Open an Existing Project

watch_sc
watchver
academy
x_watver
x_watch
iv014

c:\fndtn\active\projects\academy

○ Create a New Project

□ Always open last project

Figure 1 Getting Started Dialog Box

3. Browse to the c:\fndtn\Active\projects directory in the Directories list and select the ACADEMY project in the Projects list (the left side) of the Open Project dialog box. Select **Open** to open the ACADEMY project.

Copying the Tutorial Files (Optional)

You can either work within the ACADEMY directory as it has originally been installed, or you can make a copy to work on. Make a copy if you want to work the complete tutorial again later. Perform the following steps to make a working copy of the tutorial files. Whenever copying projects in Foundation, it is important to use the “Copy Project” feature in the Project Manager to ensure that the project’s directory structure is kept intact.

- Select **File** → **Copy Project**.
- Under the Destination section, type **Myproj** (or a unique name of your choice) in the Name field.
- Click **OK**.
- Select **File** → **Open Project**.
- Scroll down in the project list and select **Myproj**. Click **Open**.
- The Myproj project may contain two UCF files. If this is the case, select the ACADEMY.ucf file. Select **Document** → **Remove** or press **Del** to remove the file from the project. Click **Yes** to confirm the removal of the file.

  This does not delete the file from disk. It only removes it from the project so that it is not used during compilation. The file still exists in the project directory on the disk. If you mistakenly remove a file from a project, select **Document** → **Add** to add it back.

Design Description

The design used in this tutorial is a hierarchical, schematic-based design, meaning that the top-level design file is a schematic sheet which refers to several other lower-level macros. The lower-level macros are a variety of different types of modules including schematic-based modules, LogiBLOX modules, state machine modules, and HDL modules.

The design begins as an unfinished design. Throughout the tutorial, you will complete the design by creating some of the modules, and by completing some others from existing files. After the design is complete, you will simulate it to verify the functionality.

Watch is a simple runner’s stopwatch. The completed schematic is shown in the following figure.

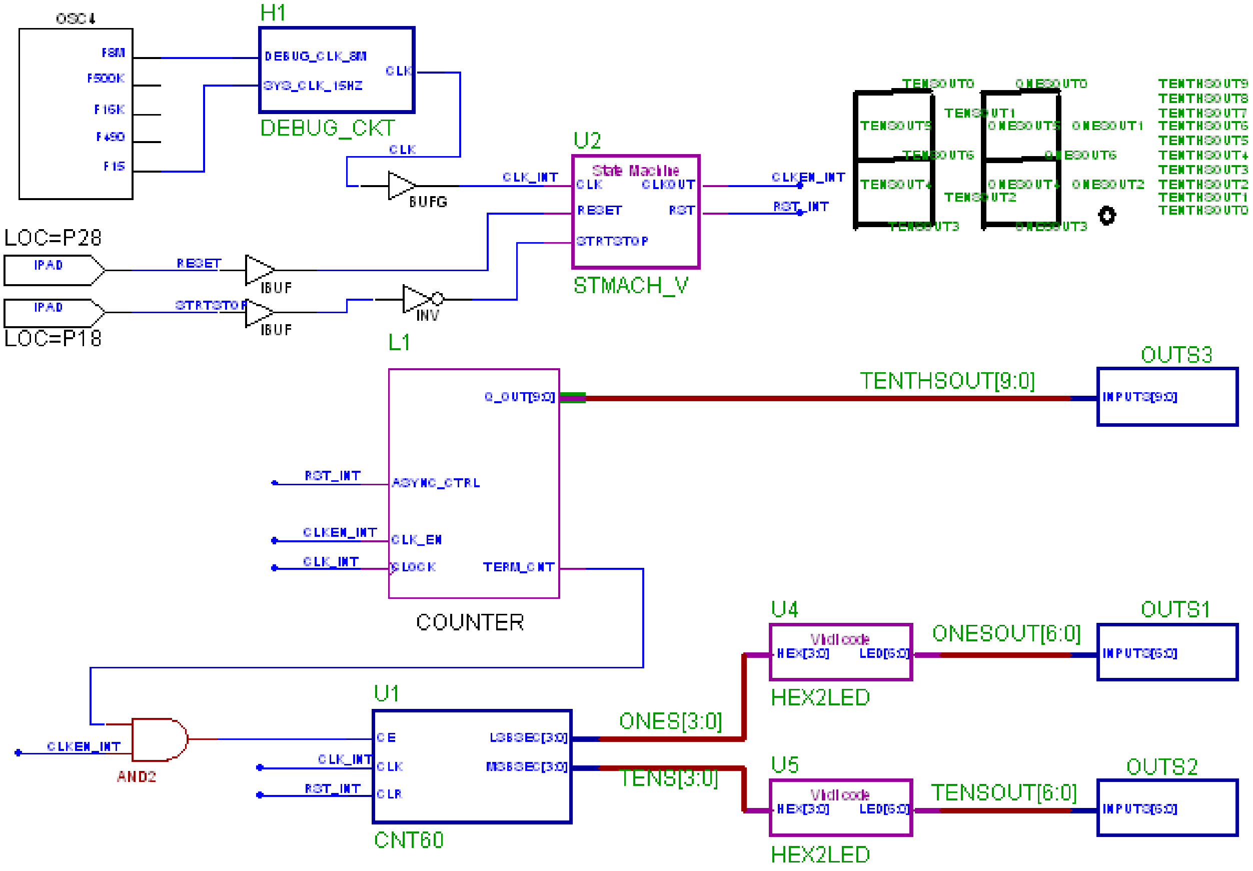


Figure 2 Completed Watch Schematic

There are two external inputs and three external outputs in the completed design. The system clock is an internally generated signal produced by OSC4, the internal oscillator in the XC4000 devices. The following list summarizes the inputs and outputs and their functions.

Inputs:

- STRTSTOP—Starts and stops the stopwatch. This is an active-low signal which acts like the start/stop button on a runner’s stopwatch.
- RESET—Resets the stopwatch to 00.0 after it has been stopped.

Outputs:

- TENSOUT[6:0]—7-bit bus which represents the Tens digit of the stopwatch value. This bus is in 7-segment display format to be viewable on the 7-segment LED display on the Xilinx demonstration board.
- ONESOUT[6:0]—similar to TENSOUT bus above, but represents the Ones digit of the stopwatch value.
- TENTHSOUT[9:0]—10-bit bus which represents the Tenths digit of the stopwatch value. This bus is one-hot encoded.

The completed design consists of the following functional blocks. Most of these blocks do not appear yet on the schematic sheet in the tutorial project since they will be created during this tutorial.

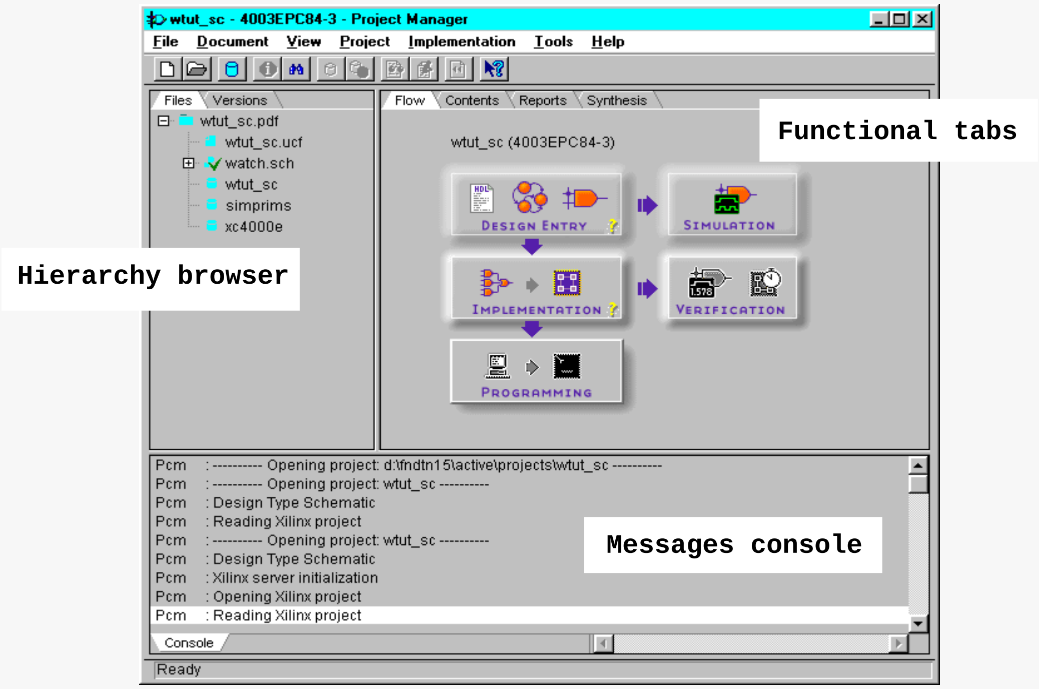
Functional Blocks

- OSC4 - Xilinx Unified Library component which represents the XC4000 on-chip oscillator.
- STMACH_A or STMACH_V - State Machine macro. This module uses the Foundation State Editor to enter and implement the state machine. One is an ABEL version; the other is a VHDL version.
- CNT60 - Schematic-based module which counts from 0 to 59, decimal. This macro has two 4-bit outputs, which represent the 'ones' and 'tens' digits of the decimal values, respectively.
- TENTHS - LogiBLOX 10-bit, one-hot encoded counter. This macro outputs the 'tenths' digit of the watch value as a 10-bit one-hot encoded value.
- HEX2LED - HDL-based macro. This macro decodes the ones and tens digit values from hexadecimal to 7-segment display format to view on the FPGA Demonstration Board.
- OUTS1, OUTS2, OUTS3 - Schematic-based macros which define the external output pin assignments for TENSOUT, ONESOUT, and TENTHSOUT output buses.
- DEBUG_CKT - Schematic-based macro containing the necessary logic to perform hardware debugging and readback using the Hardware Debugger

The Project Manager (reference section)

The Project Manager controls all aspects of the design flow. You can access all of the various design entry and design implementation tools as well as the files and documents associated with your project. The Project Manager also maintains revision control over multiple design iterations.

The Project Manager is divided into three main subwindows. To the left is the Design Hierarchy Browser which displays the project elements. To the right is a set of tabs, each one opens a separate functional window. The third window at the bottom of the Project Manager is the Message Console and shows status messages, errors, and warnings, and is updated during all project actions. These windows are discussed in more detail in the following sections.



```
Pcm    : ---------- Opening project: d:\fndtn15\active\projects\wtut_sc ----------
Pcm    : ---------- Opening project: wtut_sc ----------
Pcm    : Design Type Schematic
Pcm    : Reading Xilinx project
Pcm    : ---------- Opening project: wtut_sc ----------
Pcm    : Design Type Schematic
Pcm    : Xilinx server initialization
Pcm    : Opening Xilinx project
Pcm    : Reading Xilinx project
```

Figure 3 Project Manager

Hierarchy Browser

The Hierarchy Browser displays the project source files in a hierarchical tree. Within this display, you can quickly navigate to any point in your design.

In the Files tab of the Hierarchy Browser, the design source files and libraries are displayed. Next to each filename, an icon tells you the file type (schematic, HDL file, state machine, library, text file). If a file contains lower levels of hierarchy, the icon has a “+” in the lower right corner. You can expand the tree by clicking this icon. You can open a file to edit by simply double-clicking the filename in the browser.

A Versions tab is also available behind the Files tab. This tab displays a design’s implementation revisions. Because this is a new design which has not yet been implemented, the Versions tab does not yet contain any revision information. Versions are discussed in more detail later in the tutorial during design implementation.

Functional Tabs

As mentioned previously, the right-hand side of the Project Manager contains a series of functional tabs. Briefly, the functions of these tabs follow.

- Flow—Provides access to tools you use to complete your entire design, arranged in a flow-chart style to guide you through the design flow. Status indicators in the lower right corner of each phase button indicate whether the step has been completed successfully.
- Contents—Lists contents and date the file selected in the Hierarchy Browser was last modified.
- Reports—Displays design flow reports.
- Synthesis—Displays all of the HDL macros contained in the project, and, from this tab, you can update these macros.

You have the option to browse through these tabs to see how the tabs are updated during the design flow process.

Message Console Window

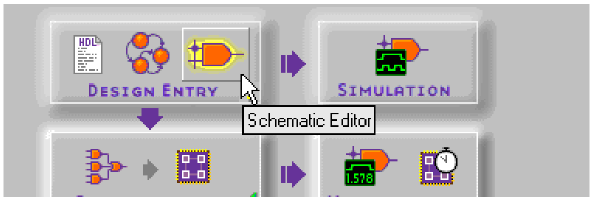
Errors, warnings, and informational messages are displayed in the Message Window. Errors are displayed in red, warnings in blue, and informational messages in black.

Design Entry

Starting the Schematic Editor

4. There are two different ways to open the Schematic Editor.

- From the Flow tab, click the Schematic Editor icon in the Design Entry phase button. This instructs the Schematic Editor to open the project's top level schematic sheet.



or,

- Double click the file name WATCH.SCH in the Files tab.

The Schematic Editor opens with the Watch schematic sheet loaded. The Watch schematic is incomplete at this point. Throughout the tutorial, you create the components to complete the design. The unfinished design is shown in the figure below.

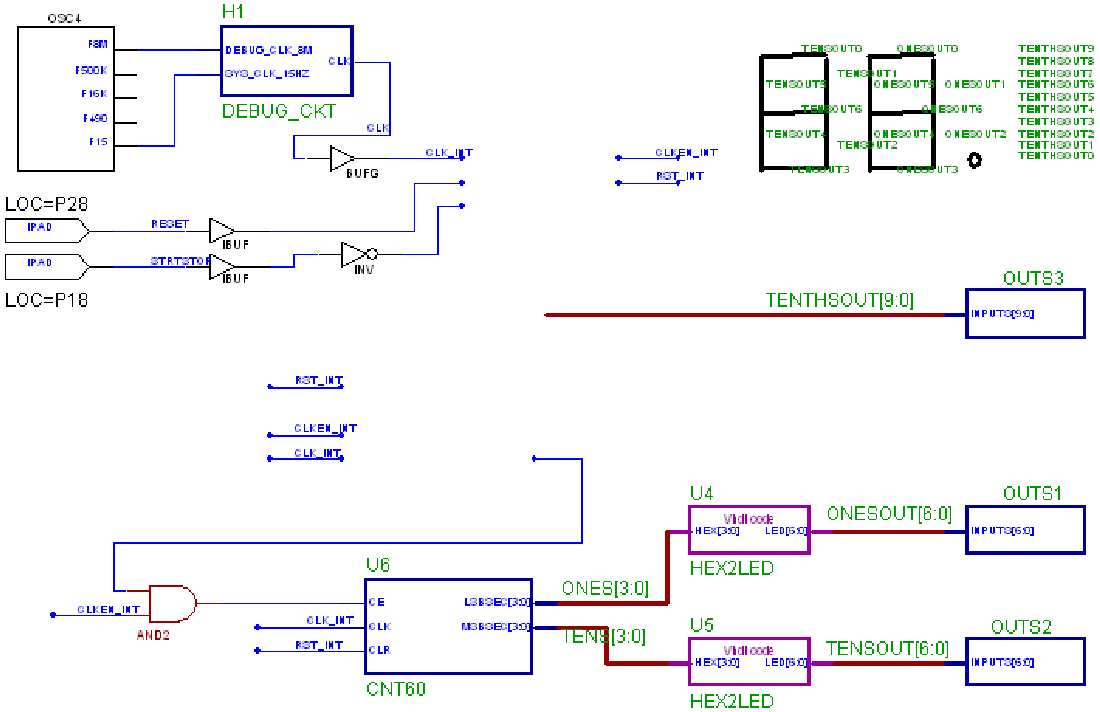


Figure 4 Incomplete Watch Schematic

5. If you need to stop the tutorial at any time, save your work by selecting **File** → **Save** from the pull-down menus.

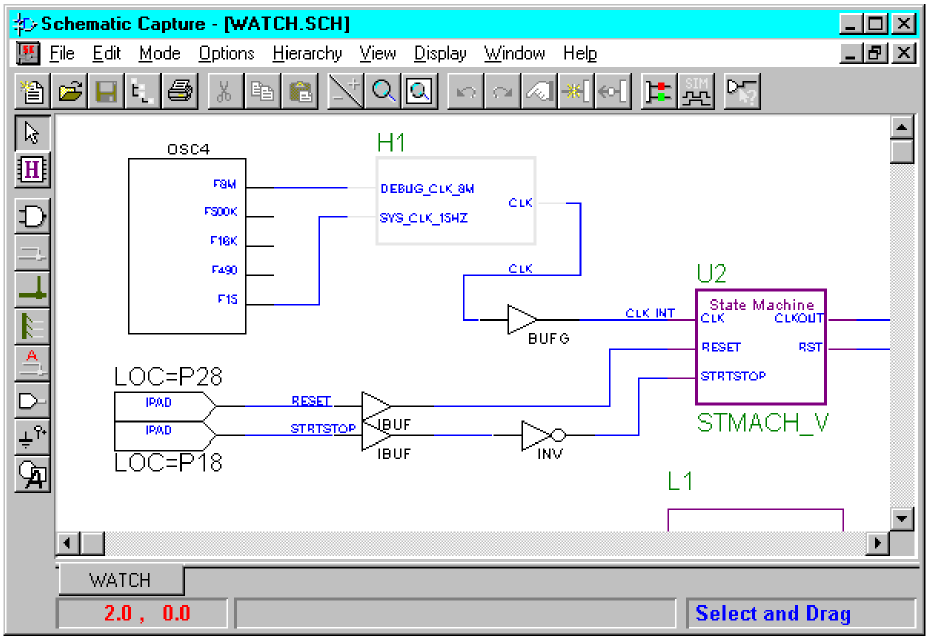


Figure 5 Schematic Editor

6. Under the Display pulldown menu is a series of commands that modify the viewing area of the Schematic Editor window. Zoom in the schematic to comfortably view it.

Completing the CNT60 Schematic Macro

The CNT60 schematic macro appears on the top level schematic sheet. The underlying schematic is incomplete. You will complete it in this section.

Opening the Macro

7. From the schematic editor pulldown menus, select **`File`** → **`Open Macro`**. The Open Macro dialog box opens.Select the CNT60 macro from the files list, then click **`OK`**.

8. Zoom in or out until all of the Hierarchy Connectors are clearly visible. The incomplete CNT60 macro is shown in the following picture.

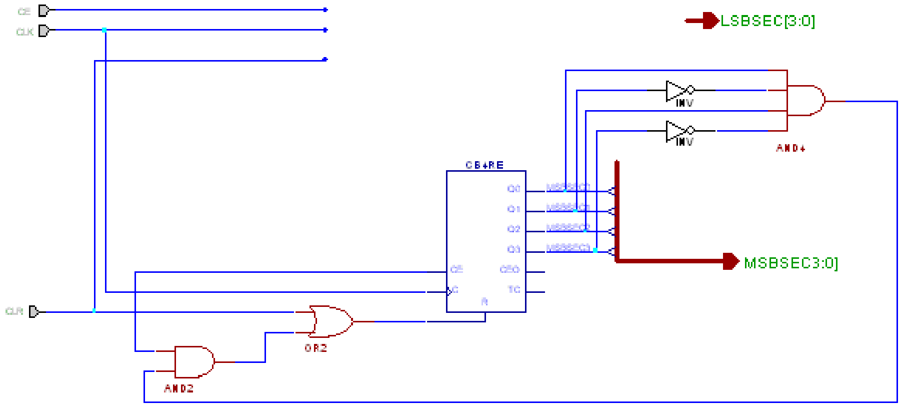
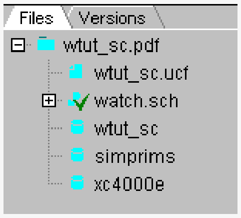


Figure 6 Incomplete CNT60 schematic macro

Project Libraries (reference section)

When you create a new project in Foundation, three libraries are automatically added to the project: the appropriate device family library based on the target family you have chosen (for example, xc4000e), the project library (with the same name as the project), and the SIMPRIMS library (for simulation). All libraries which are part of the project are listed in the Files tab of the Project Manager. You can double click on any of these libraries to see the contents of the library.



wtut_sc.pdf
wtut_sc.ucf
watch.sch
wtut_sc
simprims
xc4000e

Figure 7 Project Libraries

The device family library (XC4000E for this project) contains all of the Xilinx Unified Library components for the given family. A complete description of all of these components can be found in the DynaText Xilinx *Libraries Guide*.

The project library (ACADEMY for this project) is a writable library containing user-created macros. Any macro you create in this project is automatically placed in this library.

Additionally, you can copy macros from other libraries into this project library and vice versa using the Schematic Symbols Library Manager which you can open with the **Tools** → **Utilities** menu in the Project Manager.

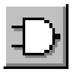
To facilitate simulation with the Foundation Logic Simulator, the SIMPRIMS is added to the project. This library contains the simulation models for the Xilinx devices.

You can add more libraries to the project by choosing **File** → **Project Libraries** from the Project Manager. After you add a library to the project, you can use any component from that library in the current project.

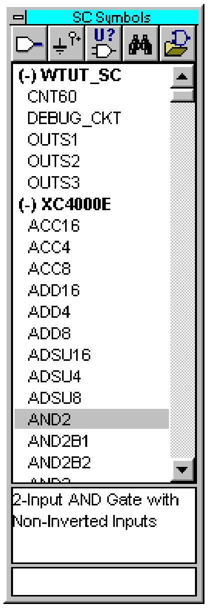
Adding Components to CNT60

Components from all of the libraries (except SIMPRIMS) for the given project are available from the SC Symbols toolbox to place on the schematic. The available components listed in this toolbox are arranged alphabetically within each library.

9. From the menu bar, select **Mode** → **Symbols** or click the Symbols Toolbox button in the vertical toolbar on the left side of the Schematic Editor.



This opens the SC Symbols window and displays the libraries and their corresponding components.



(-) WTUT_SC
CNT60
DEBUG_CKT
OUTS1
OUTS2
OUTS3
(-) XC4000E
ACC16
ACC4
ACC8
ADD16
ADD4
ADD8
ADSU16
ADSU4
ADSU8
AND2
AND2B1
AND2B2

2-Input AND Gate with Non-Inverted Inputs

Figure 8 SC Symbols Toolbox

10. The first component you will place is an AND2, a 2-input AND gate. You can select this component by either scrolling down the list and selecting it or by typing **AND2** in the bottom of the SC Symbols Window. Then move the mouse back into the schematic window.

    In the SC Symbols window, when the AND2 component is selected, a description of the component appears in the bottom of the window.

11. Move the symbol outline to the location shown in the following figure and click the left mouse button to place the object.

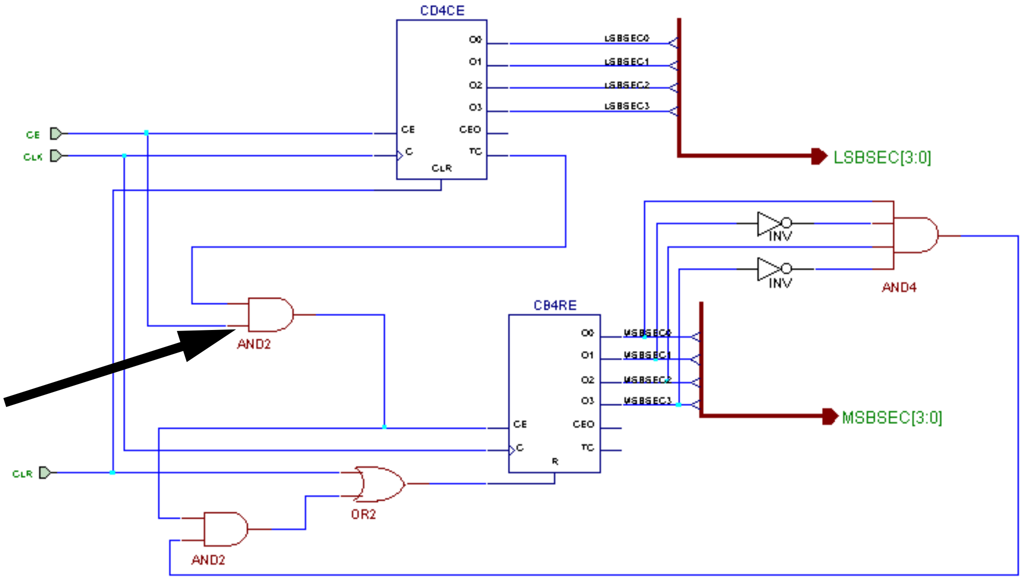


Figure 9 Completed CNT60 Schematic

Note: The preceding schematic illustrates the completed CNT60 schematic. Use this figure as a reference for drawing nets and buses in the following subsections.

Correcting Mistakes

12. If you make a mistake when placing a component, you can easily move or delete the component:

- Press the **`Esc`** key on the keyboard to exit the Symbols Mode.
- Select the component you want to move or delete. Make sure that no other components are selected (clicking on a blank area of the schematic de-selects everything).
- Click and drag to correctly place the component, or press the **`Del`** key on the keyboard or the Cut icon in the toolbar to delete the component.

Placing the Remaining Component

13. Follow the steps listed previously in the “Adding Components to CNT60” section to place the CD4CE component on the schematic sheet as shown in the “Completed CNT60 Schematic” figure. For a detailed description of the functionality of this and each of the other components, refer to the Xilinx *Libraries Guide*.

Drawing Nets

You use the Draw Wires icon in the vertical toolbar to draw wires (also called nets) between the various components on the schematic. Use Nets to physically connect single bits together.

Signals can also logically be connected by naming multiple segments identically. In this case, the nets do not need to be physically connected on the schematic to make the logical connection. In the CNT60 schematic, you will draw nets to connect the components together. Do not yet worry about drawing the nets for the LSBSEC and MSBSEC buses. These nets will be drawn in the next section.

Follow these steps to draw a net between the AND2 and the CB4RE component on the CNT60 schematic.

14. Click the Draw Wires icon in the vertical toolbar.

15. Click the source symbol pin (output pin of the AND2), then click on the destination pin (CE pin on the CB4RE). The net will automatically be drawn between the two pins.

    Note: You can specify the shape of the net by moving the mouse in the direction you want to draw the net and then single-clicking to create a 90-degree bend in the wire.

16. Draw the nets to connect the remaining components as shown in the "Completed CNT60 Schematic" figure. To draw a net between an already existing net and a pin, click once on the component pin and once on the existing net. A junction point will be drawn on the existing net.

Adding Buses

Sometimes it is convenient to draw a set of signals as a bus rather than as several separate wires. You have the option to group signals in the form of a bus and "tap" this bus off to use each signal individually. In this CNT60 schematic, there are two buses, each comprised of the 4 output bits of each counter. These buses are named LSBSEC[3:0] and MSBSEC[3:0], and they will also be connected to hierarchy connectors to connect them to the CNT60 symbol.

Addthe LSBSEC[3:0] bus to the schematic as follows.

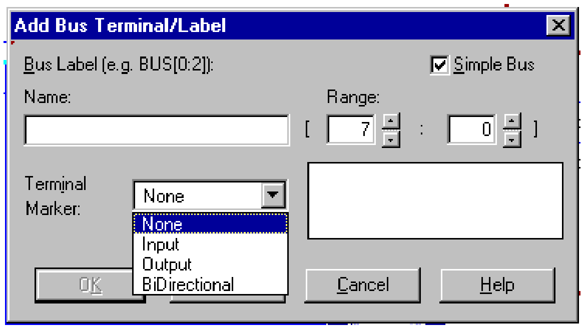
17. Select **`Mode`** → **`Draw Buses`** or click the Draw Buses button in the vertical toolbar to get into the Draw Buses mode.

18. The CNT60 schematic has some bus "stubs" connected to Hierarchy Connectors which represent the symbol pins on the CNT60 macro symbol.

    Click the end of the LSBSEC[3:0] stub, then move the mouse to a new position. Click to make a corner in the bus.

19. Terminate the bus by double clicking with the left mouse button. This opens the Add Bus Terminal / Label dialog box where you can define the bus name, width, and the type of terminal you want to use.

20. In the Add Bus Terminal / Label dialog box, change the Terminal Marker type to None by choosing this selection from the pulldown menu. This sets the type of terminal for the point where you are terminating the bus. Do not change any of the other settings. Click Bus End (the bus name and width were previously defined so you do not need to do that here).



Add Bus Terminal/Label

Bus Label (e.g. BUS[0:2]):

☑ Simple Bus

Name:

Range: [7 : 0]

Terminal Marker: None

Figure 10 Creating Bus Ends

21. If you make a mistake, press the **Esc** key on the keyboard to exit the Draw Buses mode. Then click the bus you want to delete so that it is highlighted. Press **Del** to remove the bus.
22. After adding the LSBSEC bus, press **Esc** or right-click to exit the Draw Buses mode.

Adding Bus Taps

Next, you add nets to attach the appropriate pins from the CD4CE counter to the bus. You use Bus Taps to tap off a single bit of a bus and connect it to another component. The Schematic Editor can automatically name the bus taps incrementally as they are drawn.

You may enlarge the view of the schematic to make it easier to draw the nets.

23. Select **Mode** → **Draw Bus Taps** or click the Draw Bus Taps button in the vertical toolbar. The cursor changes, indicating that you are now in Draw Bus Taps mode.

24. Click the LSBSEC[3:0] bus *label*.

    The status bar at the bottom of the window displays the message `Expand Bus Tap: LSBSEC3`. This tells you that the next bus tap drawn will be labeled LSBSEC3.

    > **Note:** The default is to start at 3 and decrement as bus taps are drawn. You can use the up and down arrow keys to change which bus bit will be tapped first.

25. Click the Q3 output pin of the CD4CE component to draw the bus tap. The net is automatically drawn and labeled. The status bar now reads `Expand Bus Tap: LSBSEC2`.
26. Click next on each of the other output pins of the CD4CE component. The bus taps will be drawn and labeled incrementally.

    > **Note:** If the bits are not automatically being labeled incrementally, check that you clicked the bus name (label) before clicking the counter output pins.

    > **Note:** If the nets appear disconnected, try selecting **Display** → **Redraw** to refresh the screen.

    If there is an error with the labeling of the bus taps, double click the bus tap net to edit the label.

27. Press **Esc** twice or right-click to exit the Draw Bus Taps mode.
28. Compare your CNT60 schematic again with the “Completed CNT60 Schematic” figure to ensure that all connections are properly made.

Saving the Schematic

The CNT60 schematic is now complete.

29. Save the schematic by selecting **File** → **Save** or clicking the Save icon in the horizontal toolbar.

All errors, warnings, and informational messages are displayed in the Message Window in the Project Manager. If any errors are issued, resolve them and save the schematic again.

30. Close this schematic and return to the top-level schematic. Select **File** → **Close** to close CNT60.

Creating a LogiBLOX Module

LogiBLOX is a graphical interactive design tool that you use to create high-level modules such as counters, shift registers, RAM, and multiplexers. You can customize and pre-optimize the modules to take advantage of the inherent architectural features of the Xilinx FPGA architectures, such as Fast Carry Logic for arithmetic functions and on-chip RAM for dual-port and synchronous RAM.

In this design, you create a LogiBLOX module called Tenths. Tenths is a 10-bit one-hot encoded counter. It counts the tenths digit of the stopwatch's time value. To better see the digit when it is downloaded on the FPGA Demonstration Board, the encoding is set to one-hot. The series of LED lights displays the Tenths digit, where one light is on for each count of the tenths digit.

You use the LogiBLOX Module Selector GUI to select the type of module you want to create, as well as the specific features of the module. You may invoke this GUI from either the Project Manager, the Schematic Editor, or the HDL Editor. The operation of the tool is the same regardless of where you invoke it.

31. From within the Schematic Editor, select **`Options`** → **`LogiBLOX`**.
32. Fill in the Logiblox Module Selector with the following settings:

    - Module Type: Counters

      Defines the type of module.

Note: experiment by selecting different module types, and notice how the symbol's shape changes for each module type. This is the shape of the symbol that will appear on your schematic.

- Module Name: Tenths

  Defines the name of the module.

- Bus Width: 10

  Defines the width of the data bus. You either choose from the pulldown menu, or type in a value.

- Operation: Up

  Defines how the counter will operate. This field is dependent on the type of module selected.

- Style: Maximum Speed

  Defines the type of optimization strategy for the module. This dictates how the layout of the module is defined.

- Encoding: One Hot

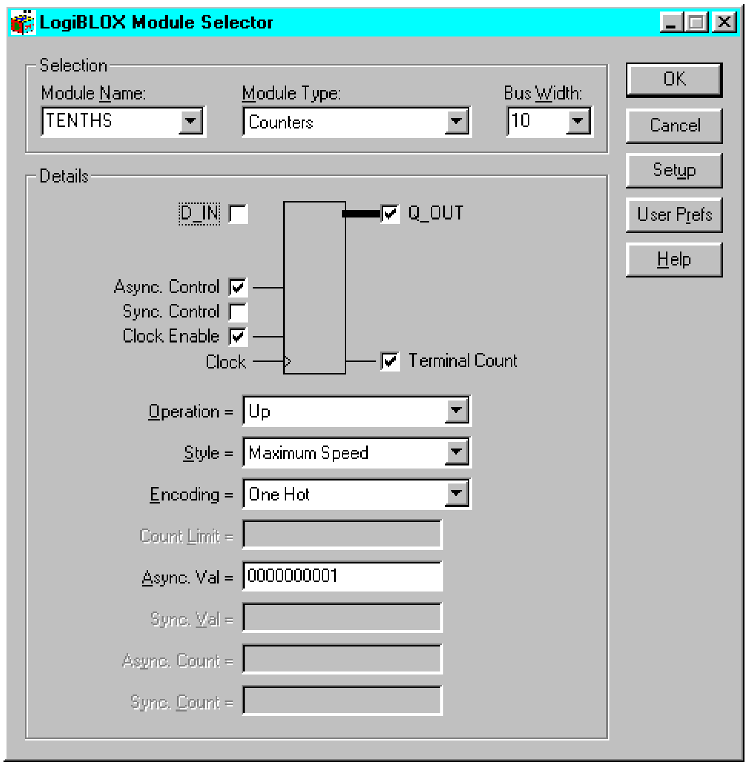
  Defines the register encoding for the module.

- Async Val: 0000000001

  Defines the value of the module on power-up and reset.

33. “Check” or “uncheck” the appropriate boxes on the module diagram so that ***only*** the following pins are used. (*You will need to uncheck default items!*)

    Q_OUT, Clock Enable, Async Control, Terminal Count



LogiBLOX Module Selector

Selection

Module Name: TENTHS

Module Type: Counters

Bus Width: 10

Details

D_IN ☐

Q_OUT ☑

Async. Control ☑

Sync. Control ☐

Clock Enable ☑

Clock

Terminal Count ☑

Operation = Up

Style = Maximum Speed

Encoding = One Hot

Count Limit =

Async. Val = 0000000001

Sync. Val =

Async. Count =

Sync. Count =

Figure 11 LogiBLOX Module Selector

34. Click **OK**. The module is created and automatically added to the project library. Additionally, it will be automatically attached to the cursor to immediately place on the schematic.

    Note: If you do not want to place the symbol at this time, you can press the **Esc** key on the keyboard to get out of the Place Symbol mode. You can then select it at any time from the SC Symbols Toolbox to place on the schematic.

35. Place the newly created Tenths component on the Watch schematic sheet, as shown below. You will connect this symbol to the rest of the schematic later in the tutorial. The symbol is labeled “L1” on the schematic sheet.

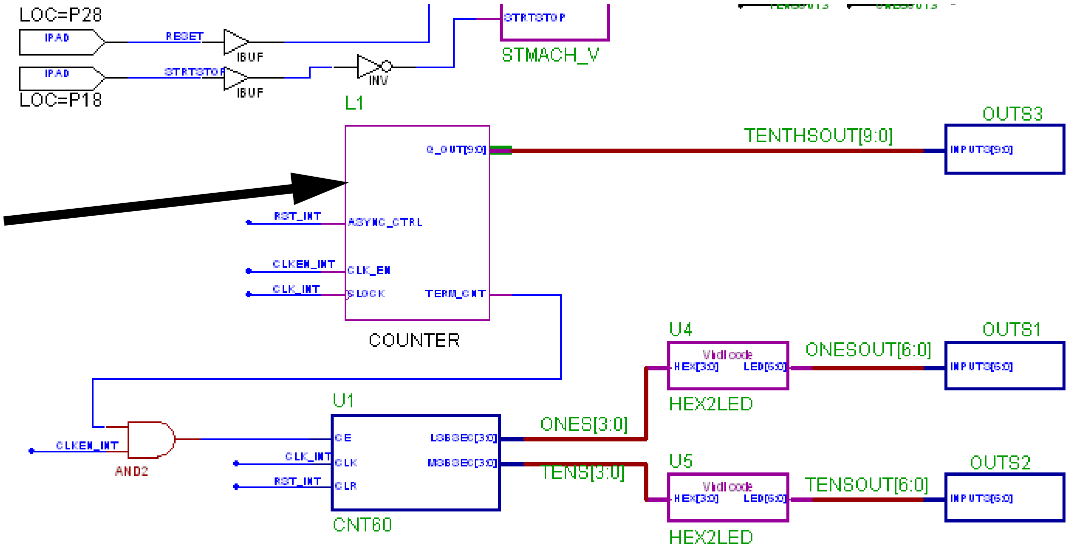


Figure 12 Placing the LogiBLOX "TENTHS" component

36. You have lined up your LogiBLOX symbol next to the bus and wires that it must be connected to. In order to connect the symbol with one operation, you will bump the LogiBLOX symbol's pins up next to the bus and wires, signaling a connection is to be made.

37. Select the LogiBLOX symbol (single click on the symbol, it will be highlighted with a red frame). Using the first mouse button, press and drag the symbol such that the pins touch the nets that they are lined up next to. To verify that the symbol is connected to its wires and output bus, press and drag the symbol with the left mouse button. If it has been successfully connected, the wires will drag with it.

38. Save the schematic by selecting **`File`** → **Save**. Close the Schematic Editor.

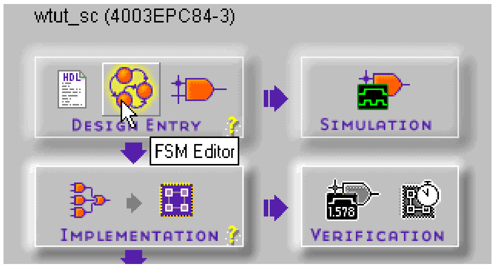
Creating a State Machine Module

With the Foundation State Editor, you graphically create finite state machines. You draw states, inputs/outputs, and state transition conditions on the diagram using a simple windows GUI. Transition conditions and state actions are typed into the diagram in appropriate VHDL, Verilog, or ABEL syntax. The State Editor then generates either VHDL, Verilog HDL or ABEL code from the diagram. The resulting HDL file is finally synthesized to create a netlist and/or macro for you to place on a schematic sheet.

For this tutorial, a complete state machine diagram is provided. In the next section, you will synthesize the module into a macro to place on the Watch schematic.

Opening the State Editor

39. To invoke the State Editor, click the State Editor button in the Flow tab of the Project Manager.



40. A dialog box prompts you to select a document. Click **Existing Document**, click **OK**, and then select STMACH_V.ASF to open the completed stopwatch state machine.

The State Machine diagram is shown below.

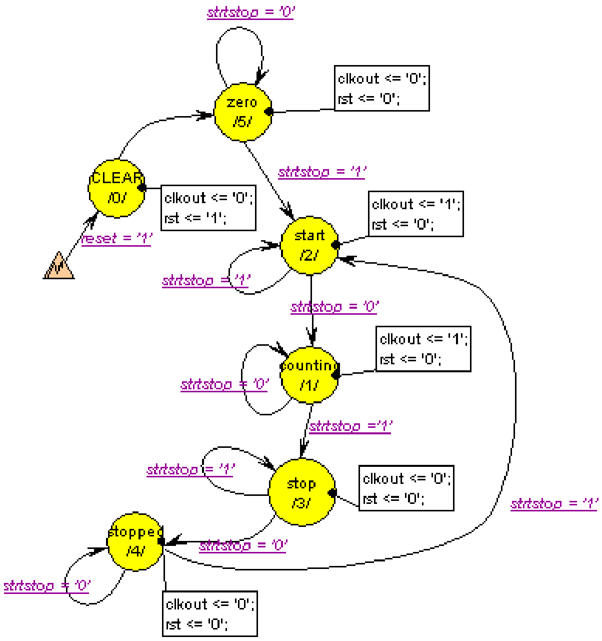


Figure 13 State Machine Diagram

- The circles represent the various states.
- The purple underlined expressions are the transition conditions, defining how you move between states.
- The boxes containing expressions attached to each state are output actions for each state, defining how the outputs behave in each state.

In the State Machine diagrams, the transition conditions and the state actions are written in proper HDL syntax.

The state machine diagram is complete. In the following section, you will create HDL from it and add it to the project/schematic sheet.

Creating the State Machine Macro

You will now synthesize the state machine and a macro will be created that you can place on the Watch schematic. The macro symbol will automatically be added to the project library. The synthesis process encompasses the creation of the HDL code from the state machine diagram and the synthesis of the HDL code by the Foundation Express compiler. Additionally, you have the option to use the State Editor to create a symbol for the state machine which you can place on the schematic.

41. Select **Project** → **Create Macro**. This synthesizes the design as well as creates the macro symbol and adds the symbol to the SC Symbols toolbox.
42. To view the HDL code which the State Editor produced, select **Tools** → **HDL Editor**.
43. Close the State Editor by clicking the X in the upper right corner of the window.

Placing the STMACH symbol

You can now place the STMACH state machine macro on the Watch schematic.

44. If it is not already opened, re-open the Schematic Editor. Open the SC Symbols Toolbox to view the list of available library components. You should now be able to locate the STMACH_V macro in this list. (If the SC Symbols Toolbox was already open, and you do not see the STMACH macro, select **`File`** → **`Update Libraries`**.)

45. Select the symbol, and add it to the Watch schematic as shown below.

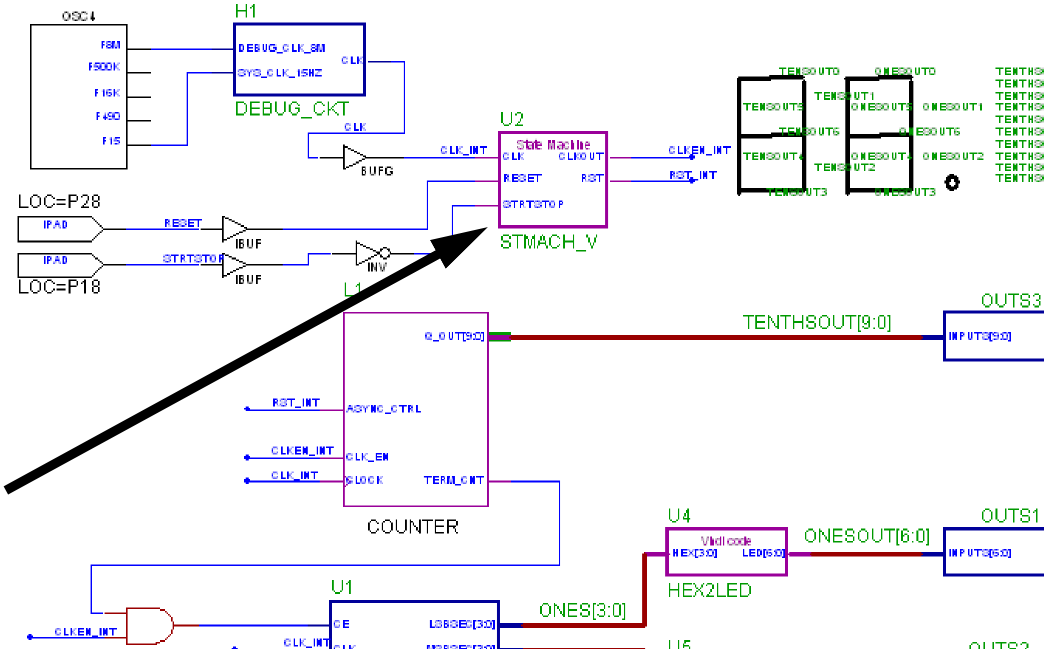


Figure 14 Placing the State Machine Macro

46. Connect the STMACH symbol to its wires by "bumping" its pins up to the existing wires (in the same manner that you connected the LogiBLOX symbol in the last section). Save the schematic.

Hierarchy Push/Pop - exploring the schematic hierarchy

Components on the schematic sheet are hierarchical. In order to descend into a lower-level of hierarchy to view the underlying file, you will use the Hierarchy Push / Pop feature of the schematic editor.

47. To push down into CNT60 (U1), the lower-level schematic sheet that you have already completed, click the Hierarchy Push / Pop button (the letter "H" on the left-side vertical tool bar). The mouse cursor changes to the letter "H". Double click the CNT60 symbol.

You may descend further down into the hierarchy; for example,

48. Try double-clicking in the CD4CE element. CD4CE is a library element, but using Hierarchy push, you may view its primitives. Note you will not be allowed to push into primitive cells.

49. To "Pop" back out of the components, you may select the Hierarchy Push / Pop icon, then double click in an empty space on the schematic. Or, you may click the Watch tab at the bottom of the Schematic Editor to return to the top-level Watch schematic sheet. You may also descend into HDL components (bringing up the HDL editor) and FSM components.

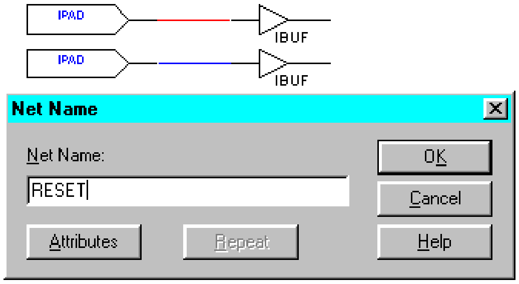
Labeling Nets

It is important to label nets and buses for several reasons. It aids in debugging and simulation, as you will more easily trace nets back to your original design. Any nets which remain unnamed in the design will be given machine-generated names which will mean nothing to you later in the implementation process. Naming nets also enhances readability and aids in documenting your design.

> **Note:** Nets are logically connected if their names are the same, even if the net is not physically drawn as a connection in the schematic. This method has been used to make the logical connection of the RST_INT, CLKEN_INT and CLK_INT signals.

You will now label the two input nets on this design. When naming input and output pins, it is advisable to label the net between the pad and the buffer. This name is carried through the entire design flow including place and route. If you label only the output of the buffer (in the case of an input pin) or input of the buffer (in the case of an output pin), you will not be able to easily trace your I/O pins in implementation tools and reports.

50. Double click the RESET net.
51. In the Net Name field, type **RESET** as shown below.



Net Name: RESET

Figure 15 Labeling Nets

52. Click **OK**.
53. Repeat the previous 3 steps for the **STRTSTOP** pin. You have the option to click and drag the new attributes to better place them on the schematic.

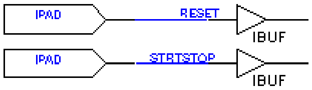


Figure 16 Labeled Nets

Assigning Pin Locations

Xilinx recommends that you let the automatic placement and routing program, PAR, define the pinout of your design. Pre-assigning locations to the pins can sometimes degrade the performance of the place and route tools. However, it is usually necessary, at some point, to lock the pinout of a design so that it can be integrated into a Printed Circuit Board (PCB).

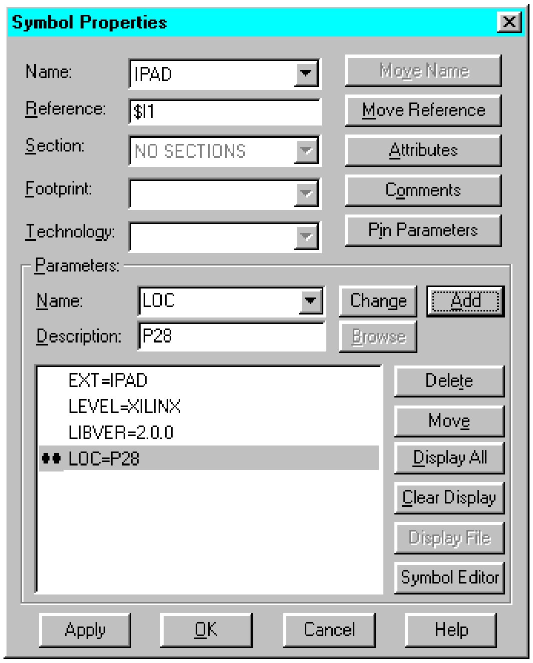
Define the initial pinout by running the place-and-route tools without pin assignments, then locking down the pin placement so that it reflects the locations chosen by the tools. In this design, you assign locations to the pins in the Watch design so that the design can function in a Xilinx demonstration board. Because the design is simple and timing is not critical, these pin assignments will not adversely affect the ability of PAR to place and route the design.

Specify pin locations by attaching a LOC parameter to a pad component. Assign a LOC parameter to the pad associated with the RESET signal on the Watch schematic as follows.

54. Double click the IPAD connected to the net labeled RESET. The Symbol Properties dialog box opens.
55. In the Parameters section, add a new parameter with these values:

    Name: **`LOC`** (find LOC in the pulldown menu associated with Name)

    Description: **`P28`** (type “P28” in the Description field)



Symbol Properties

Name: IPAD

Reference: $I1

Section: NO SECTIONS

Footprint:

Technology:

Parameters:

Name: LOC

Description: P28

EXT=IPAD
LEVEL=XILINX
LIBVER=2.0.0
●● LOC=P28

Figure 17 Assigning Pin Locations

56. Click **Add**. The parameter appears in the list box. This step assigns the RESET signal to pin P28 of the target device.

    Note: Notice the two black dots to the left of the parameter. This indicates that both the Name field and the Description field of the parameter will be displayed on the schematic. You can double click on the parameter to change the number of dots shown.

    - One dot—only the Description field will show on the schematic
    - Zero dots—neither the Description field nor the Name field will appear on the schematic.

    This function only affects what is displayed on the schematic; in all cases, the parameter has the same effect on the tools.

57. Click **`Apply`**. You see the parameter next to the IPAD.
58. Click **`OK`** to close the window.
59. Repeat the previous five steps to assign the STRTSTOP input pin to pin P18.

    Note: You may click and drag the attributes to position them where you wish on the schematic.

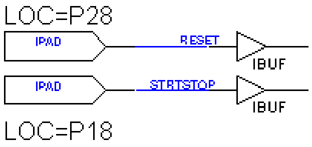


Figure 18 STRTSTP and RESET Pin Assignments

Completing the Schematic

60. Complete the schematic by wiring the components you have created and placed, adding any additional necessary logic, and labeling nets appropriately. The finished schematic is shown in the following figure as a guide.

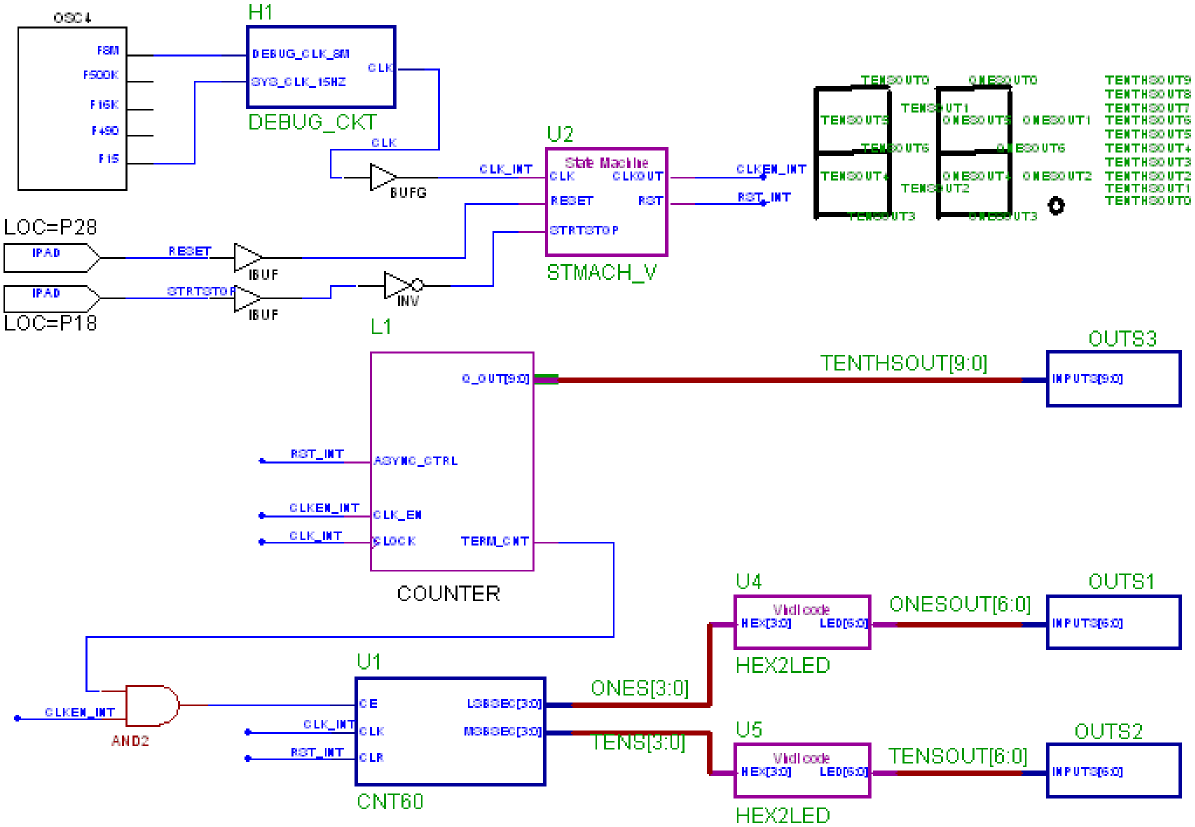


Figure 19 Completed Watch Schematic

61. Save the design by selecting **File** → **Save**.

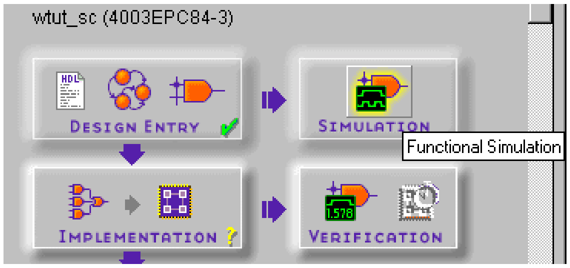
Functional Simulation

You can perform functional simulation before design implementation to verify that the logic that you have created is correct. Foundation provides a Logic Simulator, which is a gate-level simulator. You can perform functional simulation on a schematic-based design immediately after the design is captured in the Schematic Editor. In the case of an HDL-based design, you can perform functional simulation immediately following synthesis. Later, you can perform timing simulation, which takes place after the design is implemented (placed and routed) with the Xilinx Implementation Tools.

In this section you will functionally simulate the Watch schematic design that you have just completed.

Starting the Logic Simulator

1. Click the Functional Simulation phase button under the Project Flow tab.



2. You may be prompted to update the schematic netlist if you modified the schematic but did not write out a netlist. In this case, click **Yes** to update the netlist.

The Logic Simulator is invoked, and the project netlist is automatically loaded into the simulator.

Performing Simulation

There are three basic steps needed to simulate your design:

- Adding signals
- Adding stimulus
- Running the simulation

These methods are discussed briefly in the following sections. In this tutorial, you use the simulator in various ways, and then you can decide what is best for you with your own designs.

Adding Signals

In order to view signals during the simulation, you must first add them to the Waveform Viewer in the Simulator. The signals are then listed in the Waveform Viewer. You can view and monitor the waveforms next to the corresponding signal names, as well as monitor the state of these signals in the schematic during the simulation.

There are two basic methods for adding signals to the Simulator Waveform Viewer.

- Using Probes from the Schematic Editor
- Using the Component Selection window in the Simulator

Adding Signals Using Probes

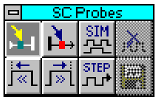
> **Note:** You may use the functional simulator to simulate HDL-based designs as well as schematic-based designs such as you are doing here. However, probes may only be used with schematic-based design flows. During functional simulation of HDL flows, signals may only be added with the Component Selection window, which will be described in the next section.

In order to add signals for the Watch design simulation, you use Probes from the Schematic Editor to identify signals that you want to view in the Simulator.

3. Bring up the Schematic Editor from within the Simulator by clicking the Schematic Capture (SC) icon in the Simulator toolbar.

4. After the schematic has opened, click the Simulation Toolbox icon in the Schematic Editor toolbar.

   This opens the SC Probes toolbox which has several buttons you can use to control the simulation from within the Schematic Editor.



> **Note:** You can view the results of the simulation either in the Simulator Waveform Viewer or by looking at the annotated values that appear directly on the schematic. These methods are examined more closely later in the tutorial.

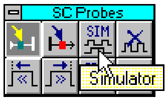
When the SC Probes toolbox is open, the cursor is automatically put into the Add Probes mode. You can see a probes icon attached to the cursor as shown in the following figure, and the Add Probes button in the SC Probes toolbox is depressed. While adding probes to the schematic, you must remain in this Add Probes mode.

Figure 20 Cursor in Add Probes Mode

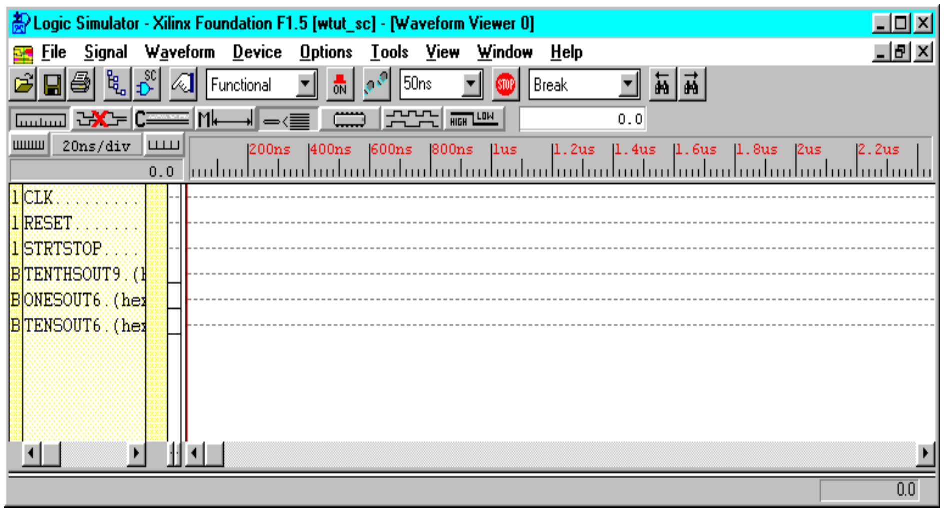
5. With the cursor in Add Probes mode, click once on the CLK signal name on the schematic. A gray box appears to the left of the CLK label. This gray box indicates that a probe has been attached to this signal.

6. Repeat the previous step to add probes to the RESET and STRTSTOP signals and to the TENTHSOUT[9:0], ONESOUT[6:0] and TENSOUT[6:0] buses.

7. Return to the Simulator Waveform Viewer by clicking the SIM button in the SC Probes toolbox.



   You should now see all of the signals you just probed listed in the Simulator Waveform Viewer.


Logic Simulator - Xilinx Foundation F1.5 [wtut_sc] - [Waveform Viewer 0]
File Signal Waveform Device Options Tools View Window Help
Functional
50ns
Break
0.0
20ns/div
0.0
200ns 400ns 600ns 800ns 1us 1.2us 1.4us 1.6us 1.8us 2us 2.2us
0.0

1 CLK.........
1 RESET.......
1 STRTSTOP....
B TENTHSOUT9.(h
B ONESOUT6.(hex
B TENSOUT6.(hex

Figure 21 Simulator Signals List

Adding Signals Using the Component Selection Window

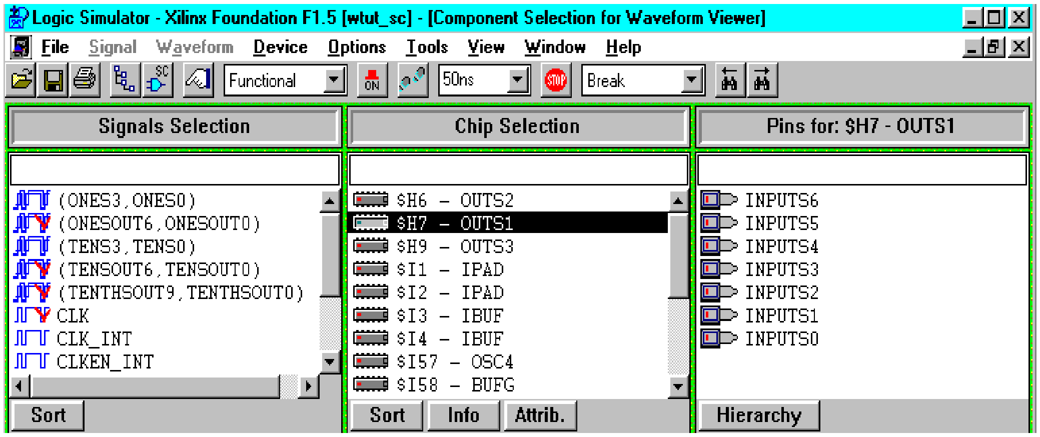
Follow these steps to add more signals using the Component Selection window within the Simulator.

8. Click the Component Selection icon in the toolbar in the Simulator or select **Signal** → **Add Signals**.

   The Component Selection Window opens.

   This window is divided into three panes. The left-most pane is the Signals Selection pane. This pane displays a list of all of the available *signals* for a given level of hierarchy. The middle pane, Chip Selection, displays a list of all of the *components* for a given level of hierarchy.

9. You can select a different level of hierarchy in the right-most pane entitled Scan Hierarchy. For instance, click the OUTS1 macro in the Scan Hierarchy pane. You are now looking at signals and components from the OUTS1 macro in the Signals Selection and Chip Selection panes, respectively.


Logic Simulator - Xilinx Foundation F1.5 [wtut_sc] - [Component Selection for Waveform Viewer]
File Signal Waveform Device Options Tools View Window Help
Functional
50ns
Break
Signals Selection
Chip Selection
Pins for: $H7 - OUTS1
Sort
Sort
Info
Attrib.
Hierarchy

Signals Selection:
(ONES3,ONES0)
(ONESOUT6,ONESOUT0)
(TENS3,TENS0)
(TENSOUT6,TENSOUT0)
(TENTHSOUT9,TENTHSOUT0)
CLK
CLK_INT
CLKEN_INT

Chip Selection:
$H6 – OUTS2
$H7 – OUTS1
$H9 – OUTS3
$I1 – IPAD
$I2 – IPAD
$I3 – IBUF
$I4 – IBUF
$I57 – OSC4
$I58 – BUFG

Pins for: $H7 - OUTS1:
INPUTS6
INPUTS5
INPUTS4
INPUTS3
INPUTS2
INPUTS1
INPUTS0

Figure 22 Scan Hierarchy Signals Selector

10. Return to the Root level of hierarchy by clicking the Hierarchy button and then selecting **Root** in the Scan Hierarchy pane to again view the signals from the top-level of the Watch design.

11. In the Signals Selection pane, several signals have red checkmarks next to their names. These signals have already been added to the Simulator, in this case by using probes in the Schematic Editor. Now you add additional signals to the Waveform Viewer.

    From the Signals Selection pane, you can either double click signals to add them to the Waveform Viewer, or you can single click and then press **Add**. Use whichever method you prefer to add the following buses.

    ONES3, ONES0
    TENS3, TENS0

    Note: It is possible to add these signals using probes on the schematic as you did for the other signals. This is to demonstrate another method for adding signals.

    If you mistakenly add any signals you do not want to add, you double click them again in the Signals Selection pane to remove them from the Waveform Viewer. The red checkmark should then disappear.

12. Close the Component Selection Window by clicking the **`Close`** button.

All of the signals you added are in the Waveform Viewer.

Deleting a Signal

13. To delete any of the signals from the Waveform Viewer, first select the signal in the signal list in the Waveform Viewer, right-click, and then select **Delete Signals** → **Selected**. This operation removes the highlighted signal from the Waveform Viewer.

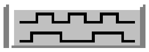
Adding Stimulus

To define the function of the input signals, you must add stimulus to your simulation. There are many ways to define stimulus with the Foundation Simulator. Some of these methods are listed below and will be discussed in this section.

- Keyboard stimulus
- Custom formulas
- Internal binary counter outputs
- Stimulator state selector
- Script file
- Waveform file

In this tutorial, you use the keyboard stimulus, custom formulas, and internal binary counter. The script file method is used later in the training when you are performing a timing simulation. All of these stimulator methods may be used in both functional and timing simulations.

14. Open the Stimulator Selection Window by clicking the Stimulator icon in the toolbar or by selecting **Signal** → **Add Stimulators...**

    The various components of this window are discussed in the following sections.

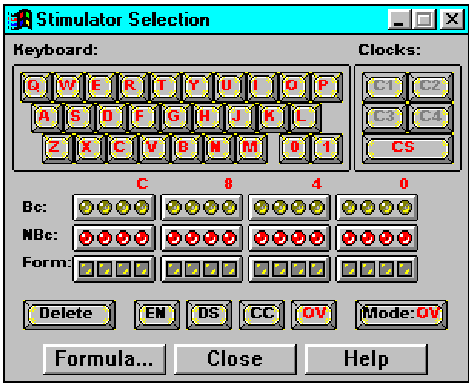


Figure 23 Stimulator Selection Window

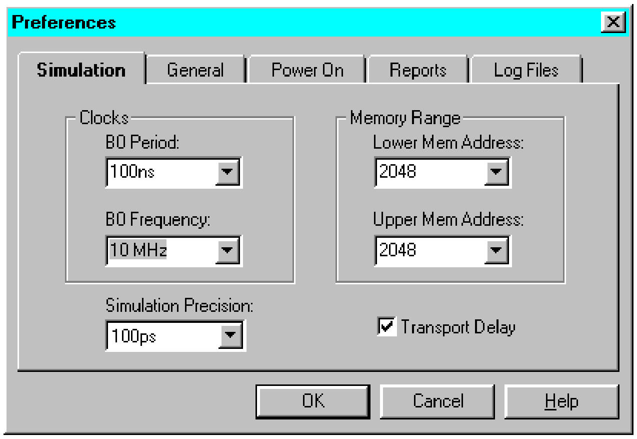
Stimulating with the Internal Binary Counter

The Foundation Simulator includes an internal free-running 16-bit binary counter. You can use each of the 16 output bits of the counter as stimulators. These signals provide 50% duty cycle signals, each bit having half the frequency of the next least significant bit. These are useful when defining clock stimulus. You may define the frequency of the LSB of the counter (B0) and can therefore derive the frequencies of the other counter outputs.

These counter outputs are represented by the round yellow LEDs in the Stimulator Selection window. The row of red round LEDs below it represents the complement of the counter outputs. The B0 output (LSB) of the counter is the farthest LED to the right, and B15 (MSB) is all the way to the left.

In the Watch design, the system clock is generated by the internal oscillator in the XC4000E device. This is represented by the OSC4 component on the schematic. The OSC4 does not have a simulation model, and, thus, cannot be simulated. To simulate the system clock, you assign stimulus to the CLK signal in the simulator. You use the B0 stimulator signal to stimulate the CLK signal in the Watch design.

15. In the Waveform Viewer, select the **CLK** signal by clicking it.
16. In the Stimulator Selection Window, click the B0 stimulator (the right-most yellow round LED). You should now see a B0 next to the CLK signal in the Waveform Viewer indicating that the B0 stimulator is assigned to CLK.
17. Select **Options** → **Preferences** from the Simulator window. This opens the Preferences window. In the Simulation tab of this window, you can set the frequency of the B0 counter output. Set the B0 frequency to 10MHz. This is significantly faster than the actual speed of the system clock used in this design (15Hz), but this frequency is adequate for the purposes of this simulation.



Clocks

B0 Period: 100ns

B0 Frequency: 10 MHz

Memory Range

Lower Mem Address: 2048

Upper Mem Address: 2048

Simulation Precision: 100ps

☑ Transport Delay

Figure 24 Simulator Preferences

18. Press **OK** to close the Preferences window.

Stimulating with Keyboard Stimulators

You assign keyboard keys as stimulus for signals in your design with the keyboard in the Stimulator Selection window. After you assign this stimulus, the signal's value toggles between 1 and 0 whenever you press the corresponding key on your PC's keyboard.

Now assign the **R** keyboard stimulus to the RESET signal in the Watch design.

19. Click and drag the **R** key on the keyboard in the Stimulator Selector onto the RESET signal name in the Waveform Viewer.

You should now see an **R** next to the RESET signal in the Waveform Viewer, which indicates that this is the assigned stimulus. Press the **R** key on your PC keyboard a few times to see the state of the stimulus changing in the Waveform Viewer.

Stimulating with Custom Formulas

The 16 square LEDs in the Stimulator Selector represent Custom Formulas. You have the option to define each of these 16 formulas to any custom stimulus pattern you want.

Now create a custom formula and then assign that formula to the STRTSTOP signal in the Watch design.

20. Click the **Formula...** button in the Stimulator Selection Window to bring up the Set Formulas window.

    Note: There are two sections of the Set Formulas window: Clocks and Formulas. Any pattern that you specify for a Clock repeats forever. Any pattern that you specify for a Formula executes just once, and then holds the last specified value for the rest of the simulation.

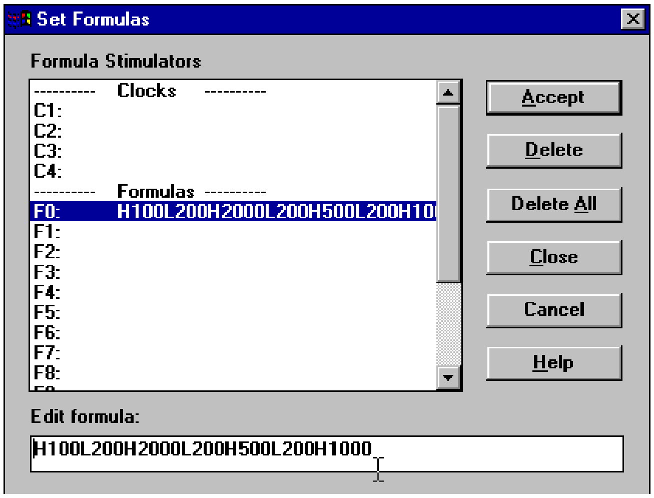
21. Double click on **F0** in the Formulas section. The Edit Formula field at the bottom of the window should now be active.

22. Type the following formula into the Edit Formula field:

    H100L200H2000L200H500L200H1000

    This formula means "High for 100ns, then Low for 200ns, then High for 2000ns, then Low for 200ns, etc...". This defines the stimulus pattern which you assign to STRTSTOP.

23. Click **Accept**. This assigns the formula you just entered to the F0 formula. You should now see it displayed next to the F0.



Formula Stimulators

```
----------  Clocks  ----------
C1:
C2:
C3:
C4:
----------  Formulas ----------
F0:         H100L200H2000L200H500L200H10
F1:
F2:
F3:
F4:
F5:
F6:
F7:
F8:
```

Edit formula: H100L200H2000L200H500L200H1000

Figure 25 Creating Formulas

24. Click **Close**.
25. Assign this newly created F0 formula to the STRTSTOP signal. Click the F0 LED (the farthest square LED to the right) and drag it onto the STRTSTOP signal in the Waveform Viewer. You should now see the F0 next to the STRTSTOP signal indicating that the F0 formula has been assigned as stimulus for that signal.

Other Sections of the Stimulator Selector (reference)

There are a few more sections of the Stimulator Selector that are not used in this tutorial, but are discussed briefly here. For complete documentation on these topics, refer to the Foundation Logic Simulator online help.

The Clocks section contains four custom clock signals. These custom clocks are defined in the Set Formulas window as mentioned above in the Custom Formula section. These custom clocks are useful for clocks with duty cycles other than 50%. You could not use the internal binary counter outputs for those types of clocks or for other repeating functions.

The EN, DS, CC, OV, and CS buttons pertain to the “mode” of the signal and stimulus. These modes control options, such as whether the stimulus is overridden by internally driven signals and whether the stimulus is enabled or disabled at a given time.

Finally, the Delete button deletes the stimulus from a selected signal. This function does not delete the signal from the waveform viewer. It merely deletes the stimulus associated with that signal.

26. Close the Stimulator Selection window by clicking **Close**.

Running the Simulation

Now you should see the three inputs of the Watch design, CLK, RESET, and STRTSTOP, listed in the Waveform Viewer, each having some type of stimulus associated with it. You should also see the outputs TENTHSOUT, TENSOUT, ONESOUT, ONES, and TENS listed. You are now ready to run the simulation.

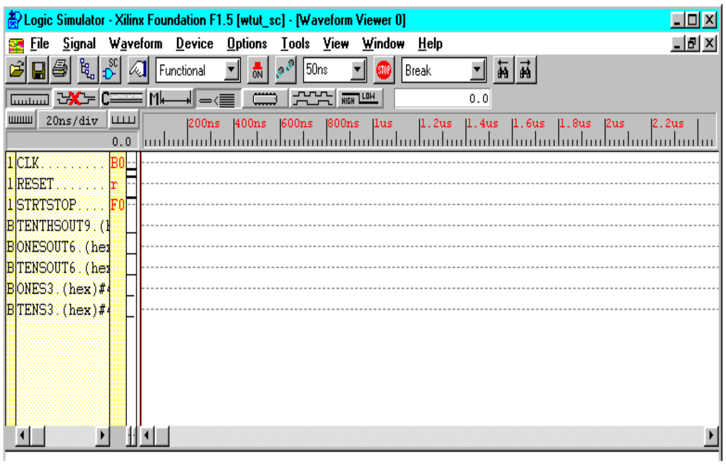

Logic Simulator - Xilinx Foundation F1.5 [wtut_sc] - [Waveform Viewer 0]
File Signal Waveform Device Options Tools View Window Help
Functional
50ns
Break
0.0
20ns/div
0.0
200ns 400ns 600ns 800ns 1us 1.2us 1.4us 1.6us 1.8us 2us 2.2us
CLK
B0
RESET
r
STRTSTOP
F0
TENTHSOUT9.(
ONESOUT6.(hex
TENSOUT6.(hex
ONES3.(hex)#
TENS3.(hex)#

Figure 26 Signals with Stimulus

27. Use the Step button in the Simulator toolbar to advance the simulation for a set amount of time. You can define the size of the Step using the pulldown menu next to the Step button, shown below.

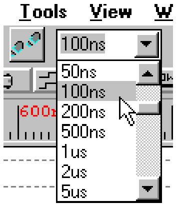
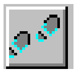


Figure 27 Simulator Step

28. Set the Step size to 100ns.

29. Press the **`r`** key on your PC keyboard until the RESET stimulus state is low.

30. Click the **`Step`** button to advance the simulation. (Shown below - it looks like two footprints.)

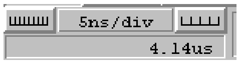


The CLK signal is clocking based on the B0 frequency you set earlier.

> **Note:** The STRTSTOP signal follows the formula created earlier.

31. Continue to click the **`Step`** button to advance the simulation.

32. Does the circuit appear to be working properly? Is the stopwatch counting? Remember that the tenths digit is a one-hot encoded value. To better see the results, you can change the radix of this bus to binary by first clicking the TENTHSOUT bus, right-clicking and selecting **Bus** → **Display** → **Binary**. You may also change the scale of the Waveform Viewer by clicking on the **`Scale`** buttons:



Recall that the ONESOUT and TENSOUT buses are in 7-segment display format, so the value of the bus may not be readily clear. Below is a diagram of the layout of the 7-segment display to help with verification.

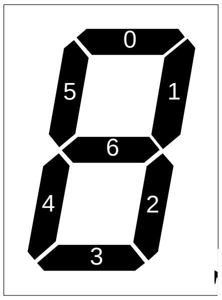


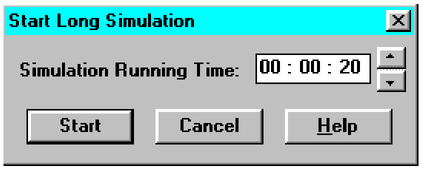
Figure 28 7-Segment Display

33. In this schematic-based design, you are also viewing the ONES and TENS bus in the Waveform viewer. These buses are the 4-bit binary values of the ones and tens digits. To better see these values, you can change the radix of the buses. by clicking the ONES bus, right-clicking and selecting **Bus** → **Display** → **Decimal**. Repeat this procedure for the TENS bus.

34. You can view the results of the simulation in the Waveform Viewer or on the Schematic. To view the simulation on the Schematic, click the Schematic Capture (SC) icon in the Simulator toolbar. This opens the Schematic Editor. You can see simulation values annotated onto the schematic. You can continue stepping the simulation from within the Schematic Editor. Click the Simulation Toolbox icon in the Schematic Editor to open the SC Probes window if it is not already open. Then, click the Step button in the SC Probes window to advance the simulation.

    On the schematic, verify that the value is being displayed properly on the model of the 7-segment display. Green LEDs indicate that the LED is active; red LEDs indicate that it is inactive.

35. As an alternate to manually clicking the **`Step`** button, you may run an extended simulation. Select **Options** → **Start Long Simulation** and set the Simulation Running Time to be 20 sec. Click **`Start`**. The simulation runs for 20 seconds of simulation time.



Start Long Simulation

Simulation Running Time: 00 : 00 : 20

Figure 29 Start Long Simulation

36. Scroll back in the Waveform Viewer using the scroll bar on the bottom of the window to inspect the results of the simulation. Does it still appear to be working?

Saving the Simulation

After you run a simulation, you can save it for future use. You can save the Waveforms you captured as test vectors, and then load them into the simulator to use again later.

37. Select **File** → **Save Waveform**. In the dialog box that opens, you can enter a name for the waveform file (.TVE). You can choose any name and save the waveform file.

    You can load this waveform file into the simulator using the **File** → **Load Waveform** command.

38. Close the Simulator.

You have now completed the first part of the Foundation Series 1.5i lab, showing the design entry tools, schematic-based projects, and functional simulation. Please experiment with other parts of this software and ask the lab instructor if you have questions!