



HDL Bencher User's Guide

The screenshot displays the HDL Bencher interface for a testbench named ADDSUB_TB.TBW. The top window shows a timing diagram with the following data:

| Time (ns) | 0 | 100 | 200 | 300 | 400 | 500 |
|-------------|---|-----|-----|-----|-----|-----|
| AIN[3:0] | 0 | 0 | 12 | 11 | 0 | 0 |
| BIN[3:0] | 0 | 0 | 0 | 0 | 1 | 0 |
| ADD_SUB | 0 | 0 | 0 | 0 | 1 | 0 |
| RESULT[3:0] | 0 | 0 | 12 | 11 | 12 | 0 |

The bottom window shows the HDL source code for the ALU entity:

```
5  entity ALU is
6      generic (NWIDE : integer := 4);
7      port (
8          AIN,BIN  : in STD_LOGIC_VECTOR (NWIDE-1 downto 0);
9          ADD_SUB  : in STD_LOGIC ;
10         RESULT   : out STD_LOGIC_VECTOR (NWIDE-1 downto 0)
11     );
12 end ALU;
```

Xilinx, Inc.



The Xilinx logo shown above is a registered trademark of Xilinx, Inc.

ASYL, FPGA Architect, FPGA Foundry, NeoCAD, NeoCAD EPIC, NeoCAD PRISM, NeoROUTE, Timing Wizard, TRACE, XACT, XILINX, XC2064, XC3090, XC4005, XC5210, and XC-DS501 are registered trademarks of Xilinx, Inc.



The shadow X shown above is a trademark of Xilinx, Inc.

All XC-prefix product designations, A.K.A Speed, Alliance Series, AllianceCORE, BITA, CLC, Configurable Logic Cell, CoolRunner, CORE Generator, CoreLINX, Dual Block, EZTag, FastCLK, FastCONNECT, FastFLASH, FastMap, Fast Zero Power, Foundation, HardWire, IRL, LCA, LogiBLOX, Logic Cell, LogiCORE, LogicProfessor, MicroVia, MultiLINX, PLUSASM, PowerGuide, PowerMaze, QPro, RealPCI, RealPCI 64/66, SelectI/O, SelectRAM, SelectRAM+, Silicon Xpresso, Smartguide, Smart-IP, SmartSearch, Smartspec, SMARTSwitch, Spartan, TrueMap, UIM, VectorMaze, VersaBlock, VersaRing, Virtex, WebFitter, WebLINX, WebPACK, XABEL, XACTstep, XACTstep Advanced, XACTstep Foundry, XACT-Floorplanner, XACT-Performance, XAM, XAPP, X-BLOX, X-BLOX plus, XChecker, XDM, XDS, XEPLD, Xilinx Foundation Series, XPP, XSI, and ZERO+ are trademarks of Xilinx, Inc. The Programmable Logic Company and The Programmable Gate Array Company are service marks of Xilinx, Inc.

All other trademarks are the property of their respective owners.

Xilinx, Inc. does not assume any liability arising out of the application or use of any product described or shown herein; nor does it convey any license under its patents, copyrights, or maskwork rights or any rights of others. Xilinx, Inc. reserves the right to make changes, at any time, in order to improve reliability, function or design and to supply the best product possible. Xilinx, Inc. will not assume responsibility for the use of any circuitry described herein other than circuitry entirely embodied in its products. Xilinx, Inc. devices and products are protected under one or more of the following U.S. Patents: 4,642,487; 4,695,740; 4,706,216; 4,713,557; 4,746,822; 4,750,155; 4,758,985; 4,820,937; 4,821,233; 4,835,418; 4,855,619; 4,855,669; 4,902,910; 4,940,909; 4,967,107; 5,012,135; 5,023,606; 5,028,821; 5,047,710; 5,068,603; 5,140,193; 5,148,390; 5,155,432; 5,166,858; 5,224,056; 5,243,238; 5,245,277; 5,267,187; 5,291,079; 5,295,090; 5,302,866; 5,319,252; 5,319,254; 5,321,704; 5,329,174; 5,329,181; 5,331,220; 5,331,226; 5,332,929; 5,337,255; 5,343,406; 5,349,248; 5,349,249; 5,349,250; 5,349,691; 5,357,153; 5,360,747; 5,361,229; 5,362,999; 5,365,125; 5,367,207; 5,386,154; 5,394,104; 5,399,924; 5,399,925; 5,410,189; 5,410,194; 5,414,377; 5,422,833; 5,426,378; 5,426,379; 5,430,687; 5,432,719; 5,448,181; 5,448,493; 5,450,021; 5,450,022; 5,453,706; 5,455,525; 5,466,117; 5,469,003; 5,475,253; 5,477,414; 5,481,206; 5,483,478; 5,486,707; 5,486,776; 5,488,316; 5,489,858; 5,489,866; 5,491,353; 5,495,196; 5,498,979; 5,498,989; 5,499,192; 5,500,608; 5,500,609; 5,502,000; 5,502,440; 5,504,439; 5,506,518; 5,506,523; 5,506,878; 5,513,124; 5,517,135; 5,521,835; 5,521,837; 5,523,963; 5,523,971; 5,524,097; 5,526,322; 5,528,169; 5,528,176; 5,530,378; 5,530,384; 5,546,018; 5,550,839; 5,550,843; 5,552,722; 5,553,001; 5,559,751; 5,561,367; 5,561,629; 5,561,631; 5,563,527; 5,563,528; 5,563,529; 5,563,827; 5,565,792; 5,566,123; 5,570,051; 5,574,634; 5,574,655; 5,578,946; 5,581,198; 5,581,199; 5,581,738; 5,583,450; 5,583,452; 5,592,105; 5,594,367; 5,598,424; 5,600,263; 5,600,264; 5,600,271; 5,600,597; 5,608,342; 5,610,536; 5,610,790; 5,610,829; 5,612,633; 5,617,021; 5,617,041; 5,617,327; 5,617,573; 5,623,387; 5,627,480; 5,629,637; 5,629,886; 5,631,577; 5,631,583; 5,635,851; 5,636,368; 5,640,106; 5,642,058; 5,646,545; 5,646,547; 5,646,564; 5,646,903; 5,648,732; 5,648,913; 5,650,672; 5,650,946; 5,652,904; 5,654,631; 5,656,950; 5,657,290; 5,659,484; 5,661,660; 5,661,685; 5,670,896; 5,670,897; 5,672,966; 5,673,198; 5,675,262; 5,675,270; 5,675,589; 5,677,638; 5,682,107; 5,689,133; 5,689,516; 5,691,907; 5,691,912; 5,694,047; 5,694,056; 5,724,276; 5,694,399; 5,696,454; 5,701,091; 5,701,441; 5,703,759; 5,705,932; 5,705,938; 5,708,597; 5,712,579; 5,715,197; 5,717,340; 5,719,506; 5,719,507; 5,724,276; 5,726,484; 5,726,584; 5,734,866; 5,734,868; 5,737,234; 5,737,235; 5,737,631; 5,742,178; 5,742,531; 5,744,974; 5,744,979; 5,744,995; 5,748,942; 5,748,979; 5,752,006; 5,752,035; 5,754,459; 5,758,192; 5,760,603; 5,760,604; 5,760,607; 5,761,483; 5,764,076; 5,764,534; 5,764,564; 5,768,179; 5,770,951; 5,773,993; 5,778,439; 5,781,756; 5,784,313; 5,784,577; 5,786,240; 5,787,007; 5,789,938; 5,790,479; 5,790,882; 5,795,068; 5,796,269; 5,798,656; 5,801,546; 5,801,547; 5,801,548; 5,811,985; 5,815,004; 5,815,016; 5,815,404; 5,815,405; 5,818,255; 5,818,730; 5,821,772; 5,821,774; 5,825,202; 5,825,662; 5,825,787; 5,828,230; 5,828,231; 5,828,236; 5,828,608; 5,831,448; 5,831,460; 5,831,845; 5,831,907; 5,835,402; 5,838,167; 5,838,901; 5,838,954; 5,841,296; 5,841,867; 5,844,422; 5,844,424; 5,844,829; 5,844,844; 5,847,577; 5,847,579; 5,847,580; 5,847,993; 5,852,323; 5,861,761; 5,862,082; 5,867,396; 5,870,309; 5,870,327; 5,870,586; 5,874,834; 5,875,111; 5,877,632; 5,877,979; 5,880,492; 5,880,598; 5,880,620; 5,883,525; 5,886,538; 5,889,411; 5,889,413; 5,889,701; 5,892,681; 5,892,961; 5,894,420; 5,896,047; 5,896,329; 5,898,319; 5,898,320; 5,898,602; 5,898,618; 5,898,893; 5,907,245; 5,907,248; 5,909,125; 5,909,453; 5,910,732; 5,912,937; 5,914,514; 5,914,616; 5,920,201; 5,920,202; 5,920,223; 5,923,185; 5,923,602; 5,923,614; 5,928,338; 5,931,962; 5,933,023; 5,933,025; 5,933,369; 5,936,415; 5,936,424; 5,939,930; 5,942,913; 5,944,813; 5,945,837; 5,946,478; 5,949,690; 5,949,712; 5,949,983; 5,949,987; 5,952,839; 5,952,846; 5,955,888; 5,956,748; 5,958,026; 5,959,821; 5,959,881; 5,959,885; 5,961,576; 5,962,881; 5,963,048; 5,963,050; 5,969,539; 5,969,543; 5,970,142; 5,970,372; 5,971,595; 5,973,506; 5,978,260; 5,986,958; 5,990,704; 5,991,523; 5,991,788; 5,991,880; 5,991,908; 5,995,419; 5,995,744; 5,995,988; 5,999,014; 5,999,025; 6,002,282; and 6,002,991; Re. 34,363, Re. 34,444, and Re. 34,808. Other U.S. and foreign patents pending. Xilinx, Inc. does not represent that devices shown or products described herein are free from patent infringement or from any other third party right. Xilinx, Inc. assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction if such be made. Xilinx, Inc. will not assume any liability for the accuracy or correctness of any engineering or software support or assistance provided to a user.

Xilinx products are not intended for use in life support appliances, devices, or systems. Use of a Xilinx product in such applications without the written consent of the appropriate Xilinx officer is prohibited.

Copyright 1991-2000 Xilinx, Inc. All Rights Reserved.

Contents

| | |
|-------------------------------------|-----------|
| Introduction | 1 |
| What is HDL Bencher?..... | 1 |
| HDL Bencher Windows | 2 |
| Conventions | 3 |
| Text Conventions | 3 |
| White Characters | 3 |
| Executing Commands | 4 |
| Toolbar Commands | 4 |
| Menu Commands | 4 |
| Resize Bar | 5 |
| Process Toolbar | 5 |
| Status Bar | 5 |
| HDL Source Viewer..... | 6 |
| Tool tips..... | 6 |
| Radix Control | 6 |
| Invoking HDL Bencher | 7 |
| Ending HDL Bencher Sessions..... | 8 |
| Refresh..... | 8 |
| About | 8 |
| Help | 9 |
| Context Sensitive Help | 9 |
| | |
| Basic Operations | 10 |
| Undo | 10 |
| Moving Through The Waveform..... | 11 |
| Zoom In/Out | 11 |
| Zoom To..... | 11 |
| Text Compression..... | 11 |
| Scrolling | 12 |
| Automatic Scrolling | 12 |
| Importing Designs | 13 |
| HDL Import Errors..... | 14 |
| Unknown Data Types (Import) | 14 |
| Unknown Value (Import) | 15 |
| Mapping Data Types | 15 |
| Mapping Package Constants..... | 16 |
| Manual Port Update (Reimport) | 16 |
| Modifying Ports/Remapping | 16 |
| Generics and Parameters..... | 17 |
| Revision Control..... | 17 |
| End of Testbench (Termination)..... | 18 |
| Opening Waveforms..... | 19 |
| Save As Waveform | 20 |
| Save Waveform | 21 |
| Close Waveform | 21 |
| Configuration Control..... | 22 |
| VHDL Compatibility Mode | 22 |
| Automatically End Testbench | 22 |

| | |
|--------------------------------------|-----------|
| Testbench Extension..... | 22 |
| Grid..... | 23 |
| Cell Enable..... | 23 |
| Reordering Signals..... | 23 |
| Developing Waveforms | 24 |
| Introduction To Drawing | 24 |
| Entering Values..... | 25 |
| Toggling Bit Values | 25 |
| WaveTable (Spread Sheet) Entry | 25 |
| Column Mode WaveTable Entry..... | 25 |
| Pattern Wizard..... | 26 |
| Waveform Selection | 27 |
| Selecting A Region..... | 27 |
| Select All | 27 |
| Unselecting | 27 |
| Clipboard and Data Transfer..... | 28 |
| Copy | 29 |
| Cut | 30 |
| Insert..... | 30 |
| Delete..... | 30 |
| Set Value | 30 |
| Clear | 31 |
| Clear All | 31 |
| Paste..... | 31 |
| Copy Text..... | 32 |
| Cut Text..... | 32 |
| Paste Text | 32 |
| Timing Constraints | 33 |
| Coverage..... | 34 |
| Coverage Statistics | 34 |
| Documentation | 35 |
| Print Command..... | 35 |
| Selecting A Printer..... | 36 |
| Print Preview | 37 |
| Fonts | 38 |
| Comments | 39 |
| How to Add Comments | 39 |
| How to Edit Comments | 39 |
| Line Breaks in Comments | 39 |
| Comment Borders..... | 39 |
| Comment Background..... | 39 |
| Resizing Comments..... | 39 |
| Moving Comments | 39 |
| Product Limitations | 40 |
| Limits And Constraints | 40 |
| Glossary of Terms | 41 |
| Index | 45 |

Introduction

What is HDL Bencher?

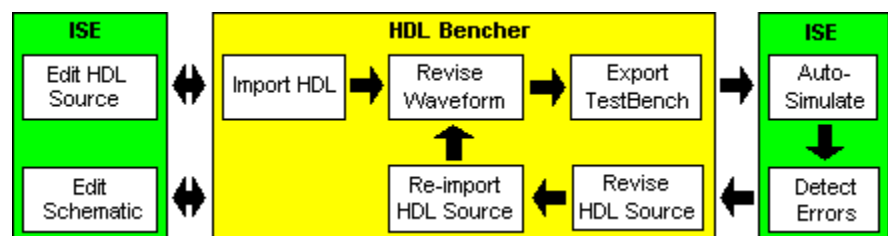
HDL Bencher automates verification of VHDL sources, Verilog sources, and schematics created within ISE. Design sources are imported, a waveform is created, and stimulus is specified by filling in the WaveTable spread sheet cells. Outputs may be auto-simulated via a command from ISE. A self checking test bench is exported whenever the waveform is saved. No knowledge of HDL or language scripting is needed to verify the design functions as intended.

Multiple layers of simulation is supported. Waveforms are developed for behavioral designs which include the expected timing results. The waveforms may be simulated behaviorally, after translation, after routing, or after device map.

HDL Bencher constrains the test run to a specific sequence of events, initial conditions, and user determined results. With HDL Bencher you can quickly validate your design functions as intended. Use it to answer questions like...

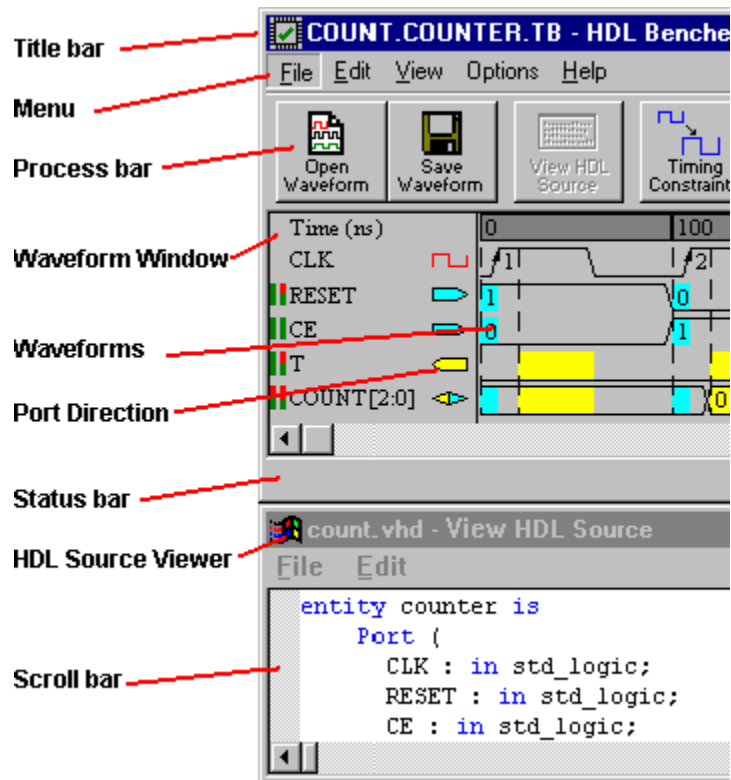
- ▶ Does count = 5 five clocks after reset?
- ▶ Will my design run at 300 MHz?
- ▶ Does IO_READ meet the required 20ns set up time?
- ▶ Is the accumulator being loaded with the value presented to the data bus?

HDL Bencher helps you validate your design functions under specific input conditions. All you need to do is specify the stimulus and the expected results, and HDL Bencher will create a test bench that will tell you if the design actually performed as intended.



HDL Bencher Windows

The Waveform Window is used to display and edit waveforms. HDL Bencher uses a fixed color scheme. The Waveform Window drawing area is always shown in gray, and waveforms are colored according to the IO mode (input, output, bi-directional etc.) These colors are independent of the colors selected via the control panel. This color scheme ensures waveforms are clearly visible and that objects may be easily identified when selected, resized, etc. Only one HDL Bencher window may be opened at a time. The size and position of HDL Bencher's windows and tool bars are set up automatically. The Source Viewer window allows the source file associated with the test bench to be viewed as test bench development proceeds. The HDL source window is automatically positioned below the waveform window whenever a waveform file is opened. HDL Bencher's windows are shown below.



- Waveform Window.....** The area where waveforms are drawn and edited.
- HDL Source Viewer.....** Shows the HDL code that is being tested.
- Menu Bar.....** Displays the names of the drop down menus.
- Minimize Control.....** This control shrinks the waveform window into its icon. Does not minimize if dialogs are open.
- Scroll Controls.....** Dragging/clicking repositions the viewing area.
- System Menu.....** Opens the system menu. This menu allows manipulation of the window.
- Title Bar.....** Displays the name of the application and the waveform the application is currently working on.
- Status bar.....** Shows information on the current command selected (or under) the cursor
- Tool Tips.....** When the cursor is moved over a tool bar button, a popup window shows the buttons name / function.
- Port Direction.....** Graphics to the right of signal names indicating the signal direction (in, out, bi-directional)

Conventions

The following sections detail the conventions used in this manual.

Text Conventions

The following table details the special characters and markings used in this manual.

| Convention | Meaning |
|-------------|---|
| Bold | Menu names and commands. |
| [] | Denotes keyboard keys. [X] is an example |
| + | Hold the first key down, press the key following the [+]. For example [ALT]+[F4] means hold [ALT] down, press [F4] , then release both keys. |
| , | Separates keyboard sequences. For example [Q], [Y] means press [Q] , release the key, then press [Y] . |

White Characters

White characters (including blanks, tabs, and new line characters) are used to separate text strings. There is a special standard for entering the tab character **[CTRL]+{TAB}**, as this key is used for other purposes in dialog boxes and menus.

Executing Commands

Commands may be executed from the menu, the process tool bar, or by directly manipulating the appropriate window locally.



All high priority command are available on the process tool bar. The menu provides many of the commands available from the toolbar as well as many others which are not used as often.

Toolbar Commands

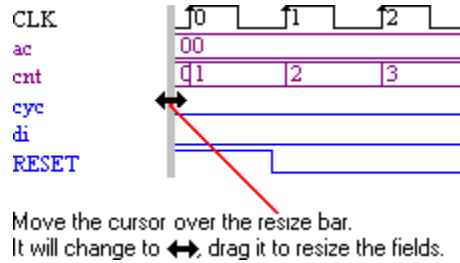
The most frequently used commands are available from the tool bar. Easy access to these commands facilitates test bench development. The commands are grouped according to function, and include a picture and a textual caption for easy identification.

Menu Commands

Most commands are available through the menu. Commands are indicated using the notation Menu ->Submenu. This syntax means: “from the high level Menu choose Submenu underneath it”. For example File->Exit means: move the cursor over the File menu, press the left mouse button, drag the cursor until Exit is selected, then release the left mouse button.

Resize Bar

Dragging the resize bar allows the width of the signal name field in the waveform to change:

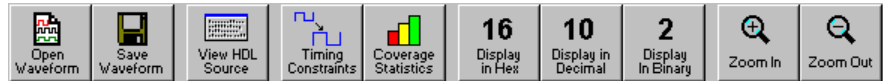


When a waveform is initialized, the width of the name field is set automatically to show the widest signal name in the design.

Process Toolbar

| Menu | Keyboard |
|---------------------------------------|----------------|
| <u>V</u> iew, <u>P</u> rocess Toolbar | [ALT]+[V], [P] |

The complete development process is clearly visible from the process bar. The process bar facilitates test bench development by bringing the most powerful features of HDL Bencher to the forefront.



Status Bar

| Menu | Keyboard |
|----------------------------------|----------------|
| <u>V</u> iew, <u>S</u> tatus bar | [ALT]+[V], [S] |

A status bar at the bottom of the waveform window displays information on the current command. When the cursor is moved over a menu item or tool bar control, the command is described in the status bar - making it easy to understand what each command does.

HDL Source Viewer

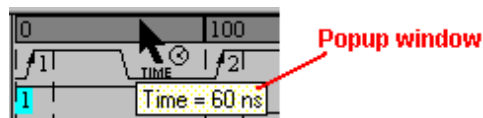
| Menu | Keyboard |
|----------------------------------|----------------|
| <u>V</u> iew, <u>H</u> DL Source | [ALT]+[V], [H] |

When a waveform file is opened, the associated HDL source is shown in the HDL Source Viewer. HDL Bencher tracks source dates, and when changes are made in third party editors, they are automatically reflected in the Viewer. In the source viewer, text strings can be located. HDL Bencher tracks port IOs. When a signal name is double clicked in the waveform window, the port signal is automatically highlighted in the Source Viewer.

When a waveform file is opened, HDL Bencher automatically searches for the associated HDL source file. If it cannot locate it, HDL Bencher will ask you to help locate the file, then it will automatically copy the file to the waveform directory.

Tool tips

Moving the cursor over a signal name causes a popup window to show the signal attributes. Moving over a waveform shows the signal's value at that time. Moving over the small graphs to the left of signal names shows the percent of test coverage achieved so far (see Coverage). Moving over a control shows the controls name and function. Moving over a column header shows the time.



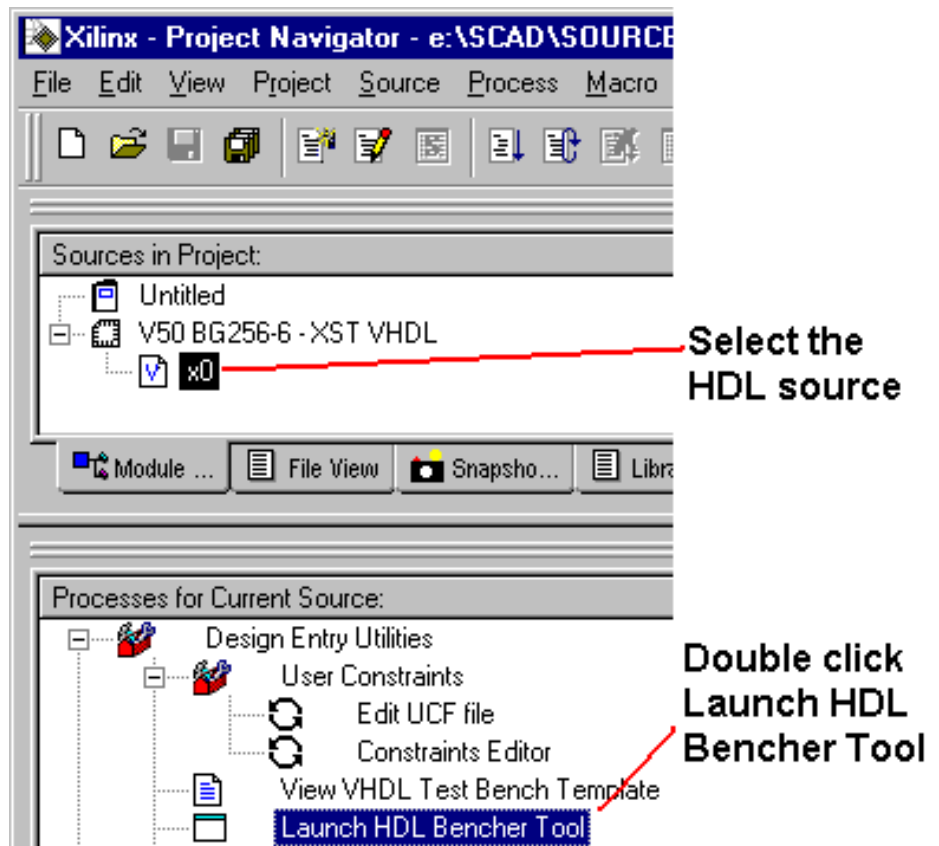
Radix Control

| Menu | Keyboard | Process bar | | | |
|--|---------------------------------|--|-----------------------------|---------------------------------|-------------------------------|
| <u>V</u> iew, <u>H</u> ex / <u>D</u> ecimal / <u>B</u> inary | [ALT]+[V], [H] | <table border="1"> <tr> <td>16 Display in Hex</td> <td>10 Display in Decimal</td> <td>2 Display In Binary</td> </tr> </table> | 16 Display in Hex | 10 Display in Decimal | 2 Display In Binary |
| 16 Display in Hex | 10 Display in Decimal | 2 Display In Binary | | | |

HDL Bencher displays numbers in hex, decimal, or binary. If the radix is changed, all further usage of numbers uses the new radix as the default. Also, the current radix is stored in the waveform files, and is remembered between invocations. HDL Bencher supports negative numbers in decimal mode. Negative numbers entered for vectors are automatically shown as their positive values. Negative values for integers are shown as negative values only in decimal mode (not in hex or binary).

Invoking HDL Bencher

When a design source (VHDL, Verilog, or schematic) file is selected within ISE, HDL Bencher is one of the processes available from the source. To invoke HDL Bencher (from ISE) select a VHDL or Verilog source file in the **Sources in Project** window then choosing **Launch HDL Bencher Tool** in the **Processes for Current Source** window. This method opens HDL Bencher and automatically imports the selected source file.



Ending HDL Bencher Sessions

| Menu | Keyboard |
|----------------------------|----------------|
| <u>F</u> ile, <u>E</u> xit | [ALT]+[F], [X] |

When this command is executed, HDL Bencher is closed. If any unsaved changes have been made to the waveform, you are asked if you would like to save the changes before exiting. The waveform is saved, and a test bench is exported.

Refresh

| Menu | Keyboard | Right Mouse Popup |
|-------------------------------|----------------|-------------------|
| <u>V</u> iew, <u>R</u> efresh | [ALT]+[V], [R] | <u>R</u> efresh |

Occasionally, as a result of graphical operations, extra pixels are left on the screen. In these instances refresh can be used to clean up the screen.

About

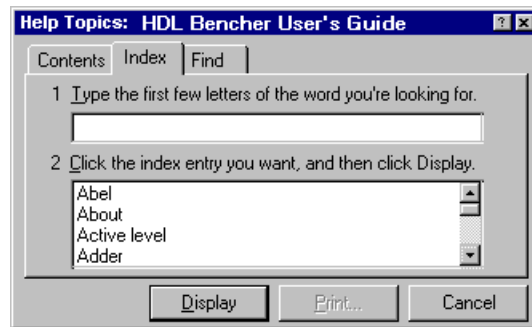
| Menu | Keyboard |
|--------------------------------|----------------|
| <u>H</u> elp, <u>A</u> bout... | [ALT]+[H], [A] |

Executing this command shows the About dialog box. This dialog box shows the version of HDL Bencher running and the copyright notice.

Help

| Menu | Keyboard |
|-----------------------------|----------------|
| H elp, I ndex | [ALT]+[H], [I] |

When information is required on a procedure or topic, HDL Bencher provides support through the Windows help engine. Help may be accessed through the help menu or through context sensitive help. HDL Bencher also provides the **About** dialog box which describes the currently running version and includes ordering information. Help is accessible by pressing **F1**, or from the **Help** menu item. The help index provides a table of all the topics available from help.



Help information may be obtained on a menu item by pressing [**F1**] when the menu item is highlighted. Help will immediately be displayed on the selected menu item. Help information is available in dialog boxes, and can be accessed by pressing [**F1**] when a dialog box is open.

Context Sensitive Help

| Menu | Keyboard |
|--|--------------------------------|
| H elp, C ontext S ensitive H elp | [SHIFT]+[F1] [ALT]+[H], [S] |

Help may be found on menu items and dialog boxes using context sensitive help. Executing the Context Sensitive Help command causes the Help Mode cursor to appear:



When this cursor is shown, clicking a menu item causes the help information on the menu item to be shown.

Information may also be found using help when a menu item (or a dialog box) is selected by pressing [**F1**]. This causes the help topic on the selected menu item (or dialog box) to be shown. When a menu item is grayed, help information may be obtained using [**F1**], but not using the context sensitive help mode ([SHIFT]+[F1]).

Basic Operations

Undo

| Menu | Keyboard |
|----------------------------|------------------------------|
| <u>E</u> dit, <u>U</u> ndo | [CTRL]+[Z] [ALT]+[E], [U] |

Changes are made to a waveform when editing, drawing, or developing a design. Sometimes a mistake is made while entering or modifying a waveform, and undo frequently can fix the problem. If a waveform is accidentally moved, edited, or deleted, these actions can be undone, and the waveform restored to the state it was in prior to the erroneous action. This command is also available from the pop up menu (click the right mouse button on the waveform window to make the pop up menu appear. The command which is to be undone is shown in the menu.


The operations which change a waveform graphically (including cut, copy, paste, delete and edit) can be undone. Non-graphical operations usually cannot be undone.

The operations which cannot be undone include file operations (open, save, new etc.), printing and print set up, all actions under the view menu, Font and Configuration dialogs, all item under the Windows menu, Wizards, and all items under the Help menu. Also, changes to the view of a window including scrolling, and resizing of the waveform window cannot be undone. If an operation cannot be undone, the Undo command is grayed.

Moving Through The Waveform

Large waveforms cannot always be shown on the screen at one time. To allow the viewing and editing of large test cases, zooming and scrolling are used. This section describes the techniques for moving through a large waveform.

Zoom In/Out

| Menu | Keyboard | Right Mouse Popup | Process bar |
|---|-------------------------------------|--|---|
| <u>V</u> iew, <u>Z</u> oom <u>I</u> n or <u>V</u> iew, <u>Z</u> oom <u>O</u> t | [ALT]+[V], [I] or [ALT]+[V], [O] | <u>Z</u> oom <u>I</u> n <u>Z</u> oom <u>O</u> t |  |

These commands allow you to adjust the number of cycles that you can view in the HDL Bencher Window, as well as the amount of detail that can be displayed at different magnification levels. These buttons turn gray when no further magnification or compression is possible. If a region is selected, it is auto panned into view when zoomed.

Zoom To



Changing the value in the zoom to box causes the waveform to be rescaled. Scaling can be done on 1 % increments from 6% to 400%.

Text Compression

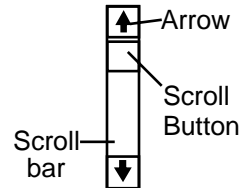
As the waveforms shrink (during zoom) it is often not possible to display the complete values of signals. HDL Bencher automatically compresses values to make them readable while zoomed. HDL Bencher will strip leading zeros, and use ellipsis to denote same value for bits of range (i.e. 3...3 instead of 33333333).

Scrolling

Scrolling allows viewing of waveforms that are too large to fit on the screen. A whole waveform may be viewed in detail by changing the viewing area.

Scroll Bars

The viewing area may be moved about the waveform by clicking the scroll arrows, clicking the scroll bars, or dragging the scroll buttons:



The scroll arrow moves the diagram by a small amount. The scroll bar moves the waveform one page, and the scroll button moves the waveform to a specific position. In addition, the keyboard may be used to maneuver through the waveform or the HDL Source Viewer Window.

Waveform Scroll Keys

The following table lists the scroll keys and their operation. These keys scroll the currently active window (when not in edit current cell mode).

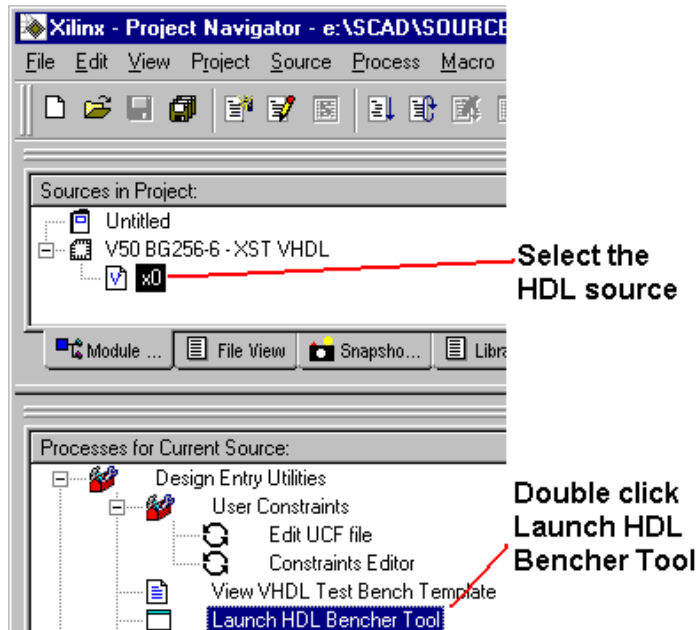
| Key Name | Scroll... |
|-------------------------|--|
| [HOME] | to the top left area of the window |
| [END] | to the end of the waveform in the Waveform Window or to the bottom of the HDL in the Source Viewer window. |
| [PGUP] | up one page |
| [PGDN] | down one page |
| [UP ARROW] | up one line |
| [SHIFT] + [UP ARROW] | up one page |
| [DOWN ARROW] | down one line |
| [SHIFT] + [DOWN ARROW] | down one page |
| [RIGHT ARROW] | right one line |
| [SHIFT] + [RIGHT ARROW] | right one page |
| [LEFT ARROW] | left one line |
| [SHIFT] + [LEFT ARROW] | left one page |

Automatic Scrolling

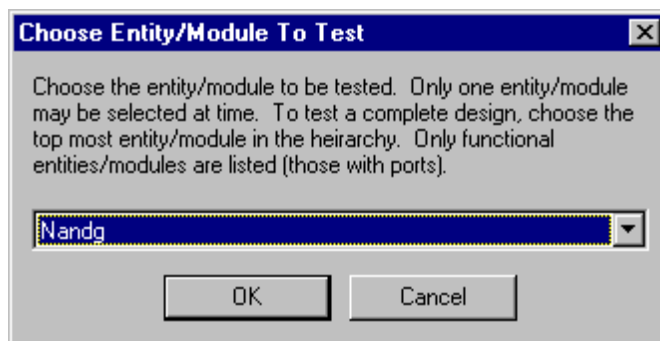
When selecting a region (or moving signals) in the waveform window, moving the cursor near the edge of the screen causes the viewing region to scroll automatically, allowing large regions to be selected.

Importing Designs

Within ISE select a VHDL, Verilog, or schematic source file in the **Sources in Project** window then choosing **Launch HDL Bencher Tool** in the **Processes for Current Source** window. This opens HDL Bencher and automatically imports the selected source file.



Once all simulatable entities are identified, you are asked to select the unit under test (if there is more than one in the design file):

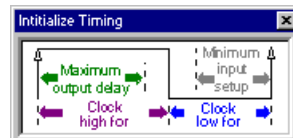


HDL Import Errors

If one entity (or module) exists in the design, it is automatically used as the unit under test. Once the unit under test has been selected, it is analyzed for port problems (syntax violations, semantic errors, and inconsistencies). When problems are detected, they are highlighted, a description is provided, and a solution is suggested:

```
Problem:
Error[0112] Expected terminating semicolon
Solution:
Add terminating semicolon
Source File:
COUNT: inout integer range 0 to 7 := 0
T: out std_logic
```

Once a design's ports are error free, all IO signals are stored away, and all generics (parameters) are captured. The HDL is analyzed, to identify if the design is clocked or combinatorial. You are then asked to select the default timing to be used in the test bench:

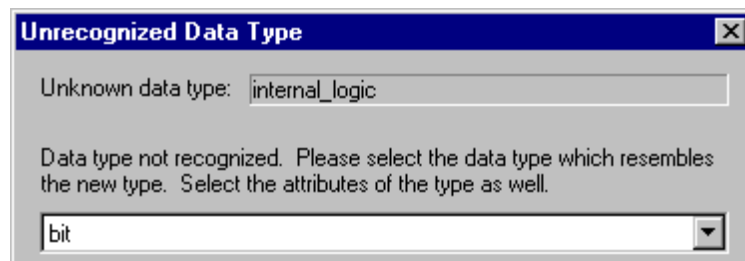


The waveform is automatically initialized, the clock pattern is automatically set, waveforms are sorted by type (input, output, bi-directional) and inputs are assigned initial values.

The HDL Source Viewer window is opened, and automatically scrolled to the entity under test (unit under test). Upon completion of the waveform specification, the waveform is automatically exported as a VHDL (or Verilog) test bench when the waveform is saved.

Unknown Data Types (Import)

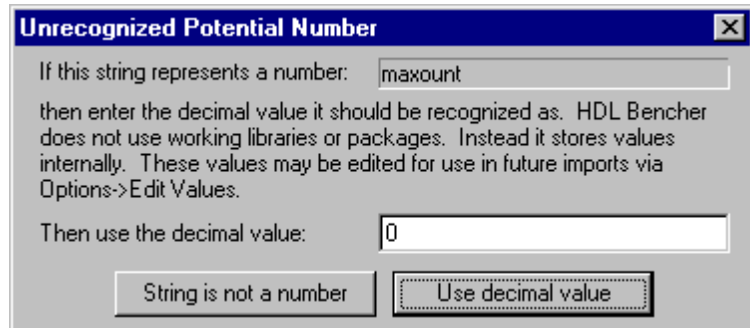
During import, when an unrecognized data type is located, you are asked to choose a data type (and value range) that best matches the one not recognized:



The default range may be chosen (for vector and integer types) so that unconstrained signals (of the selected types) will use the default range. Also, if a type includes a range operator in the definition, HDL Bencher can be instructed to flag a range specification in the port as an error.

Unknown Value (Import)

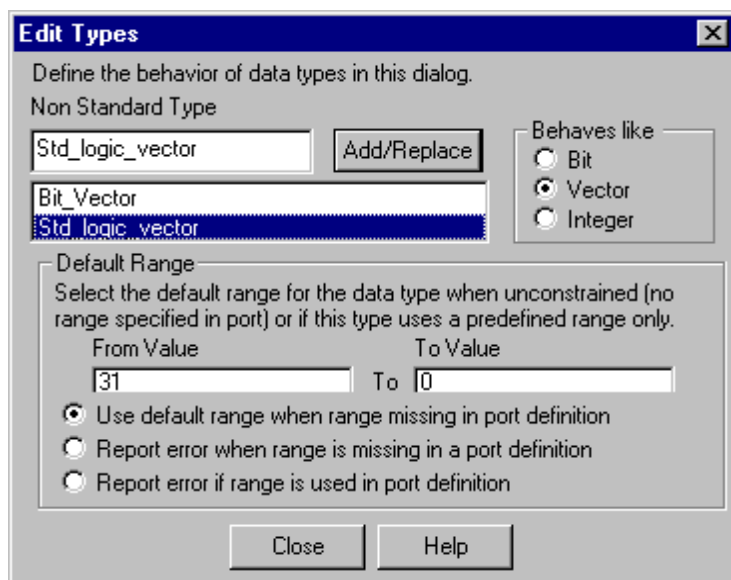
During import, when an unrecognized name is located in a port definition, you are asked if it is a constant (value). If so, you must supply the value (which is stored for future re-use):



Mapping Data Types

| Menu | Keyboard |
|---------------------------------|----------------|
| Options, <u>M</u> ap Data Types | [ALT]+[O], [M] |

When the **Import** command is executed, a signal may be located which uses a non-standard data type. HDL Bencher uses a local library of data types during import. The data types may be edited using this command:

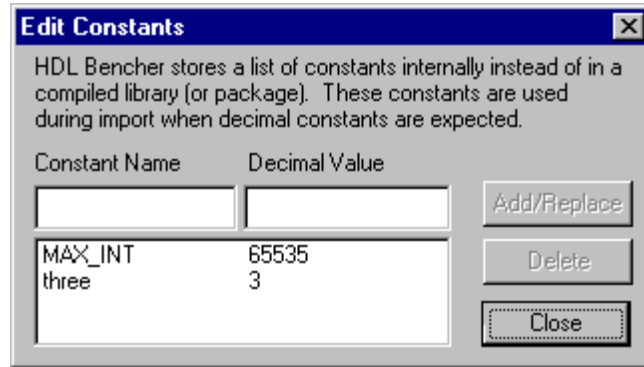


Data types are used during the import process. If a data type is found, its width (and other attributes) is resolved based on the type specification defined in this dialog.

Mapping Package Constants

| Menu | Keyboard |
|--|----------------|
| O ptions, Map P ackage Constants | [ALT]+[O], [P] |

When the **Import** command is executed, global constants may be located in the port definitions. To complete the import, the constants must be assigned a specific value. The values may be set in this dialog:



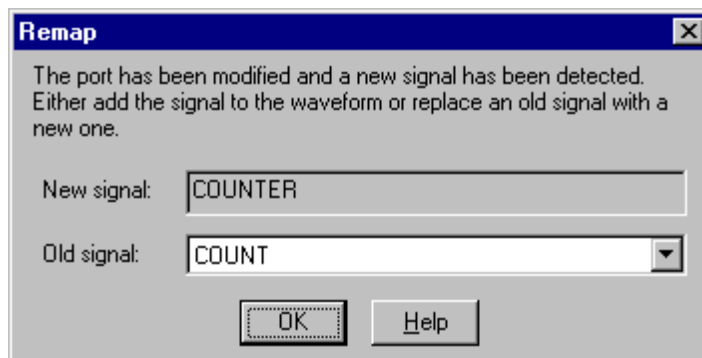
Manual Port Update (Reimport)

| Menu | Keyboard |
|--|----------------|
| O ptions, U ppdate Ports | [ALT]+[O], [U] |

If the time stamps of the HDL source are corrupted, or incorrectly set to be older than the associated waveform file, no automatic update will occur. It is possible to force the update, to guarantee the waveform is up to date using the **Update Ports** command.

Modifying Ports/Remapping

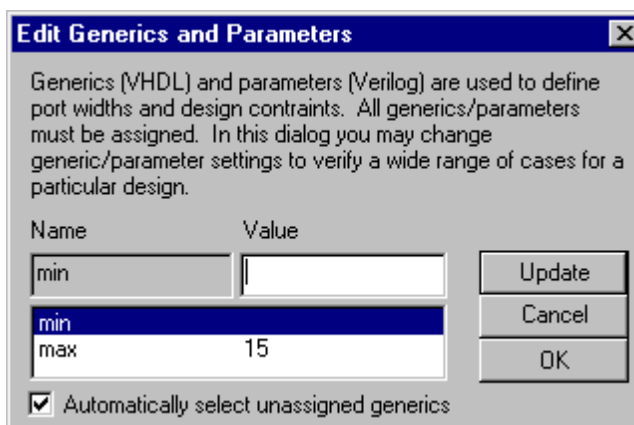
When a signal in the port definition of the unit under test is modified, and the ports are updated by HDL Bencher, the ports are reanalyzed. If a signal's direction, width, or port order has changed, the waveform is updated automatically. When signals are added (or deleted) the change is automatically reflected in the waveform. When signals are renamed, you are asked which signal to remap the old name to:



Generics and Parameters

| Menu | Keyboard |
|--|----------------|
| <u>O</u> ptions, <u>E</u> dit <u>G</u> enerics/ <u>P</u> arameters | [ALT]+[O], [G] |

VHDL Generics (parameters in Verilog) allow a design to be used with different size vectors, timing constraints etc. To instantiate a unit under test, all generics/parameters must be initialized. HDL Bencher uses the default values assigned to the generics (when found). If no value has been found for a generic, the Edit Generic and Parameters dialog is invoked:



In this dialog, unassigned generics are automatically selected (when Automatically select unassigned generics is checked), for you to enter values. Generics with data types integer, natural and positive are checked for range violations. Other data types, including user defined types and externally packaged types are allowed, however the values are stored as strings without any type validation. When the values of generics are changed in the HDL source, the design is automatically reimported to correctly reflect the updated HDL source using the new values from the HDL source.

Revision Control

When changes are made to a diagram, and the current diagram is to be closed (via the **Exit**, **File Open**, **File Close**, or **File New** commands) HDL Bencher queries if the test bench and the waveform should be updated. This avoids stale test cases.

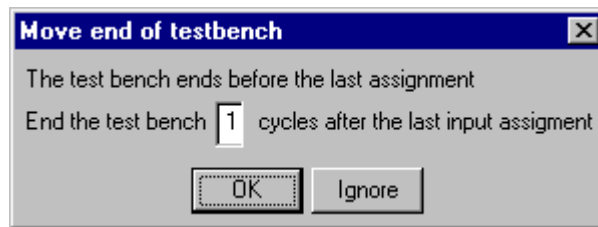
The time stamp of the HDL source file is automatically checked by HDL Bencher, each time the HDL Bencher is made the active window, and on closing a waveform. If the HDL source is out of date, you will be asked if the waveform should be updated automatically. Choosing yes will update all ports, and update the waveform so that when exported you will have an updated, valid test bench.

Port IOs are automatically updated under a number of circumstances. Ports are updated when the HDL is modified, and the waveform window is made the active windows. Likewise, opening a waveform file that is older then the HDL source will cause an automatic update.

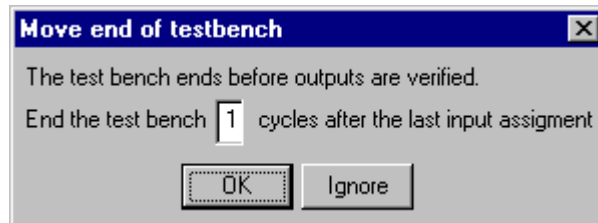
End of Testbench (Termination)

HDL Bencher allows you to determine the end of the test bench (when exporting) either automatically or manually. When a waveform is first imported, and values are assigned, the end of test bench is determined automatically, and is revised each time a design change is made. In automatic mode, the last assignment (inputs) or assertion (outputs) made to the waveform is used as the end location. A vertical blue bar shows the termination position. Moving this bar disables automatic termination. Repositioning the termination bar allows the beginning portion of a waveform to be used for testing, or allows clocking to continue beyond the last input assignment. To re-enable automatic end of test bench, execute the **Options, Configuration** command.

When the waveform is saved, the end of test bench marker is checked. If the marker is set before the last assigned value, you are asked if the end of test bench should be moved to or past the end of the waveform:




If the end of testbench marker is set after all assignments, but before outputs are checked, then you are asked if the end of test bench should be moved to or past the end of the waveform:



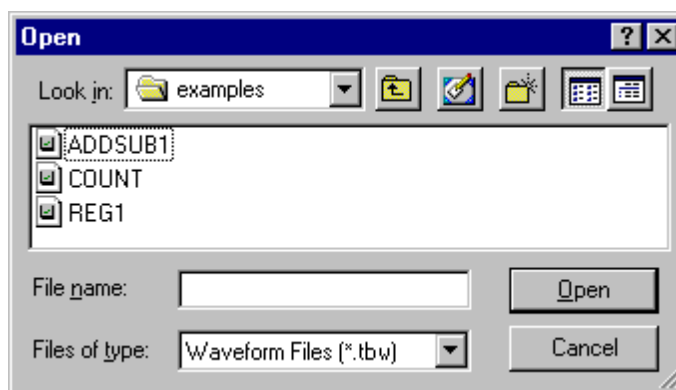
Opening Waveforms

Within Xilinx ISE you can open HDL Bencher waveform files by double clicking the associated .TBW file in the Sources in Project window. While in Bencher, you can edit different waveform files using the **File Open Waveform** command and the Most Recent Files list. These methods have the same effect; they allow a waveform file to be read from disk into HDL Bencher. Once a file has been opened, the diagram is shown in the Waveform Window.

The File Open Command

| Menu | Keyboard | Process Bar |
|----------------------------|---------------------------------|---|
| File, Open Waveform | [ALT]+[F], [O] or [CTRL]+[O] |  |

The **Open Waveform** command allows waveforms to be read into HDL Bencher for editing. Executing **Open** causes the Open dialog box to appear:



Type or select the desired file name into the File Name field. If the current directory is not the one holding the desired file, either type the path in front of the file name or maneuver the directories by clicking over the appropriate selection in the directories list. If the current drive is not the correct drive for the desired file, type the drive in front of the path. Finally, open the file by clicking the **Open** button. The new diagram is opened, shown on the waveform window, and the title bar reflects the new file name. HDL Bencher waveform diagrams normally use the extension ".tbw". File names may be typed without specifying the extension, as ".tbw" is automatically appended when no extension is given. Other extensions may be used but are not recommended. When the waveform is opened, the associated HDL source file is also opened.

Opening Files Using Most Recent Used List

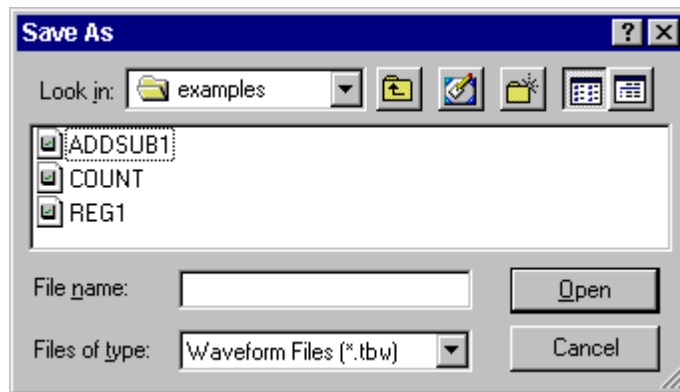
| Menu | Keyboard |
|----------------|----------------|
| File, 1 | [ALT]+[F], [1] |

HDL Bencher remembers the last four waveforms edited (MRU list), and updates the MRU list each time a diagram is opened or the Save As command is executed. To open a file recently edited, from the File menu (at the bottom of the list) is the MRU, select the file name (click on it), the diagram will be opened.

Save As Waveform

| Menu | Keyboard |
|---|----------------|
| F ile, S ave A s Waveform... | [ALT]+[F], [A] |

This command allows a diagram to be saved to a specific file name, path, and drive. After execution of **Save As** the title bar changes to reflect the file name used in the command. No extension is necessary when saving a file, as the ".tbw" extension is automatically appended to the file name. If an extension is specified, then the file will be saved using the extension. It is not recommended to use an extension other than ".tbw". If the filename entered into the **Save As** dialog box already exists, HDL Bencher will ask if the file should be written over. **Save As** causes the following dialog box to appear:




Type the desired file name into the **File Name** field. If the current directory is not the correct one, type the path in front of the file name, or maneuver the directories by clicking over the appropriate selection in the directories list. If the current drive is not correct, type the drive in front of the path and filename, or select the drive from the **Drives** combination box. Finally, save the file by clicking **OK**. Once this command executes, the title of the diagram is updated to the new name. Saving a waveform to a different directory will cause HDL Bencher to ask if it should copy the HDL source to the new directory.

To handle multiple waveforms for each HDL source, you should save each waveform to a unique file name (the waveforms are automatically added to ISE when they are saved). The waveform file (.tbw) can later be opened by double clicking the waveform icon in the ISE project. After saving the waveform, the associated test bench is exported automatically.

Note: If ISE is closed (not running), and a waveform is saved within Bencher, it will not be added as a source in the ISE project. When you re-invoke the ISE project, you will need to add the waveform source to the project. To add a source within ISE, open the project file, then press **[INSERT]** to open the add source dialog. Choose the appropriate waveform file (.tbw), then click **OK**.

Save Waveform

| Menu | Keyboard | Process Bar |
|-------------------------------------|---------------------------------|---|
| <u>F</u> ile, <u>S</u> ave Waveform | [ALT]+[F], [S] or [CTRL]+[S] |  Save Waveform |

This command writes the diagram to the current filename shown in the title bar. When a waveform is saved, it is added to the current ISE project. The waveform file (.tbw) can later be opened by double clicking the waveform icon in the ISE project. After saving the waveform, the associated test bench is exported automatically.

File Naming Convention

HDL Bencher allows multiple test benches to be created for a single HDL source file. HDL Bencher keeps track of the test bench waveform files and the test benches. When HDL Bencher imports a source file (filename.vhd), it automatically names the waveform file (filename_tb.tbw), using the filename (i.e. filename_tb.tbw). When the waveform is exported (as a test bench), the resulting test bench file has the same base filename, with the suffix _TB added, and the appropriate extension (i.e. filename_TB.vhd).

Close Waveform

| Menu | Keyboard |
|-----------------------------|---------------------------------|
| <u>F</u> ile, <u>C</u> lose | [ALT]+[F], [C] or [CTRL]+[N] |

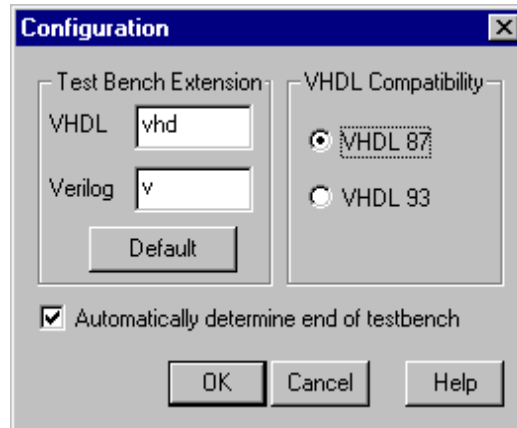
This command closes the current waveform. If any changes were made, HDL Bencher asks if it should save the changes to the waveform (and update the test bench) before clearing the old waveform. After **File Close** executes, the title bar shows the current filename as "DEFAULT.TBW".

The options used for the last waveform diagram are used whenever a new design is imported. This allows many diagrams with the same feature sets to be created without having to specify the options each time. These options include the font, timing constraints, and VHDL mode (87 or 93).

Configuration Control

| Menu | Keyboard |
|--|----------------|
| <u>O</u> ptions, <u>C</u> onfiguration | [ALT]+[O], [C] |

This dialog allows design specific attributes to be modified.



VHDL Compatibility Mode

This option allows you choose to have the test bench made VHDL 93 or VHDL 87 compliant. If you compile the exported test bench in a third party simulator, and receive warnings VHDL 87 mode used, VHDL 93 mode expected (or the other way around), simply change modes in this dialog, then re-save the design.

Automatically End Testbench

HDL Bencher can automatically terminate the test bench on the last assignment. As the waveform is modified, the end of test bench marker is relocated appropriately. If this marker is dragged (by the user), the automatic end of test bench detection is disabled. To re-enable automatic detection, use this command.

Testbench Extension

The target test benches file name is the waveform file name with the _TB suffix, and the file extension as set in the configuration dialog. When you change the extension in this dialog, the change is affected for all new waveforms, but existing waveforms use the settings specified for them. Change the extension if your simulation tool requires an alternate extension to properly recognize the test bench as a test fixture.

Grid

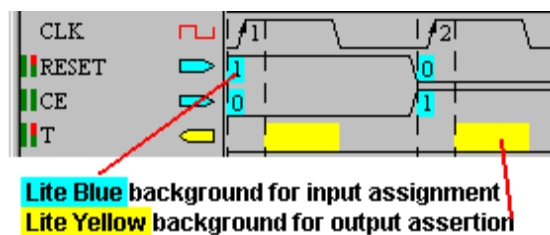
| Menu | Keyboard |
|---------------------------------|----------------|
| <u>V</u> iew, Show <u>G</u> rid | [ALT]+[V], [G] |

HDL Bencher can automatically display vertical lines at each active clock edge to make following the waveforms easier.

Cell Enable

| Menu | Keyboard |
|--|----------------|
| <u>V</u> iew, Show <u>C</u> ell <u>E</u> nable | [ALT]+[V], [N] |

HDL Bencher automatically assigns inputs, and checks outputs at the times selected in the Revise Timing dialog. Cell Enable shows the times when signals are active. Blue background is used when inputs are assigned, and yellow background is used when outputs are asserted (checked). The port direction is also indicated graphically to the right of each signal name.



Reordering Signals

Pressing the left mouse button down over a signal name, then dragging it (up or down) will reorder the signals in the waveform (this has no impact on the port order).

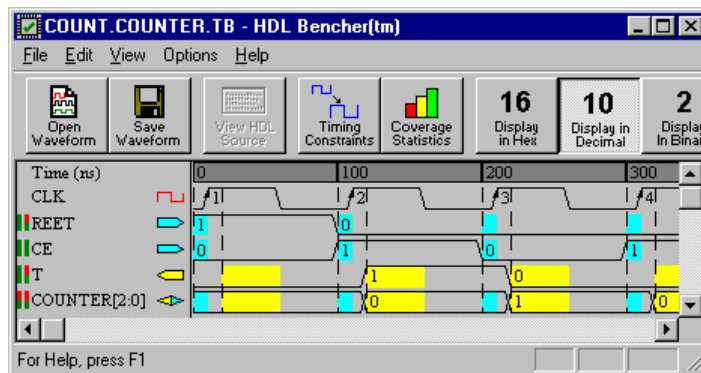
Developing Waveforms

Introduction To Drawing

HDL Bencher has been optimized for creating behavioral test benches. Numerous assignment modes greatly simplify the development process. Double clicking a bit signal toggles its value. The pattern wizard allows a range of cell values to be assigned. Also, data values can be entered using the WaveTable™ approach where “cell” are assigned values just like in a spreadsheet. HDL Bencher supports bits, vectors, and integers. The multi-value system is provided (1, 0, X, Z, U, H, L, -).

HDL Bencher's WaveTable spreadsheet entry handles all timing related issues in a transparent manner. When values are assigned, they are set at the correct time to perform the expected action. Waveform values are checked as they are being entered, guaranteeing the test bench is always correct by construction.

HDL Bencher includes support for the Windows standard graphics operators (cut, copy, paste etc) to facilitate manipulation of the waveforms. Special operations like insert and clear all are specifically tailored to the WaveTable nature of this tool.



Entering Values

Toggleing Bit Values

The easiest way to assign bit signals is to click directly on the signal's waveform at the time where you want the change to take place. If the signal is a bit value, it is toggled (0↔1).

Input signals are automatically assigned at the input set up time, and outputs are validated after the delay specified in the timing constraints dialog.



Click once in clock 3 to set T, then again in clock 4 to clear it

WaveTable (Spread Sheet) Entry

HDL Bencher implements a table-waveform approach geared toward fast behavioral verification. Inputs are assigned at the set up time, and outputs are checked at the minimum valid delay. This approach allows the waveform to be represented as a spreadsheet of enabled (and disabled) cells. Data entry is performed using a standard spreadsheet interface. This method is very fast, because the user simply fills in the blanks to complete the behavioral description.

To enter values, double click a signal at the time it should be changed. This causes the value editor to appear. In this edit box, you can enter the value of the signal at the specified time. When done assigning the value, press the [ENTER] key to move to the next time frame (to the right). The arrow keys allow the current cell to be selected. Pressing an arrow key moves the current cell to the next valid time slot.



Double clicking invokes the cell editor

Values may be entered without leading zeros, and the radix used for data entry is the display radix. Incorrect values are prohibited, and range validation occurs during data entry. Values supported are 1, 0, X, Z, U, H, L, W, - (for bits), and constants in the valid range for vectors (and integers) as well as X, Z, and U. When the display mode is not binary, and there are conflicting bit values assigned in a given digit (i.e. 01x0 0000) then the offending digit is displayed as a question mark (i.e. ?0). Values entered in quotes are always interpreted as binary (independent of display mode).

Column Mode WaveTable Entry

To facilitate data entry when in WaveTable mode (editing cells), a row of values can be entered as a comma delimited list. When no value is specified between commas, the current value is maintained. After typing a comma delimited list, press [ENTER] to store the list, update the row, and move down to the next row.

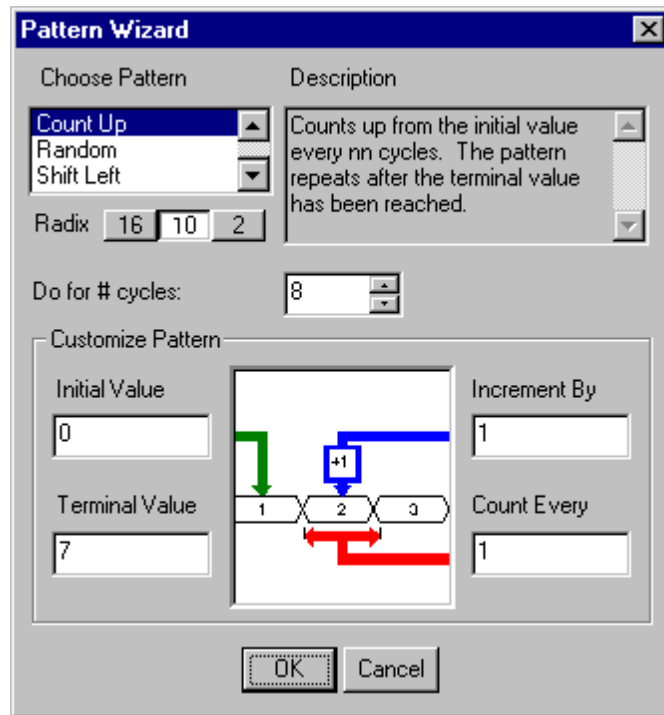
Pattern Wizard

Setting complex waveforms is made easier with the pattern wizard. Double click a signal to invoke the value editor. To the left of the editor is the Pattern button. Click the Pattern button to invoke the wizard.



Double clicking invokes the cell editor

The pattern wizard allows bit signals to be toggled, pulsed, or randomly assigned. Vectors can be counted, shifted, or randomly assigned. In the wizard, choose the pattern to be used. A description provides more information on the specific pattern. As pattern types are changed, the template is updated to allow you to quickly set up the desired pattern.



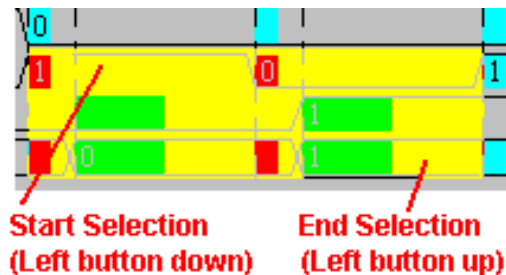
The initial value and terminal value fields are used to initialize the pattern and to determine when to roll over. The increment by indicates how much to count by.

Waveform Selection

Waveforms may be selected singly or as a group. Selecting a group allows many waveforms to be manipulated simultaneously. Selected waveforms may be cut, copied, pasted, cleared, and deleted.

Selecting A Region

Portions of the waveform (and comments) are selected by defining a rectangular region around them. To select a region, press the left mouse button at a corner of the region, drag the mouse until the desired area is enclosed, then release the button. Selectable objects in the region are highlighted, and a bounding box surrounding the region is drawn. When whole columns of waveforms are selected, the associated comments are automatically included in the selection areas. The following diagram shows a selected region:



Select All

| Menu | Keyboard |
|--------------------------|------------------------------|
| <u>E</u> dit, Select All | [ALT]+[E], [A] [CTRL]+[A] |

All waveforms and comments in a diagram may be selected by executing this command. Selected waveforms are highlighted, and the region is bounded by a rectangle. The region may be cut, copied, deleted etc.

Unselecting

To unselect, press [ESC]. All selected objects (highlighted) are unselected, and the surrounding rectangle disappears.

Clipboard and Data Transfer

HDL Bencher supports the transfer of waveform regions (as bitmaps or metafiles) to other programs. It also supports the exchange of waveform fragments within a particular waveform diagram. To facilitate data transfer, HDL Bencher supports both the clipboard and an internal buffer. The internal buffer allows waveform regions to be transferred within HDL Bencher. Related commands (insert, delete, clear all) do not operate on the clipboard, but are described here because of their similarity.

Whenever a **Cut Image** or **Copy Image** command is executed on a selection, a full color image of the selection is copied to the clipboard (metafile and bitmap formats) and a description image is copied to the internal buffer. Once an image has been placed into the clipboard, it may be pasted into any Windows package that accepts bitmaps or metafiles from the clipboard. This is how many of the pictures shown in this document were developed.

When an area is selected into the clipboard the zoom level is ignored. Anything sent to the clipboard is shown in the clipboard in its unzoomed size. This makes it easy to see information about the design when zoomed.

Images are copied as metafiles and bitmaps. Applications automatically select the best format.

When a selection is copied to the clipboard, grid lines and cell enabling are suppressed. To copy the video screen to the clipboard, use the Windows feature, print screen (**[CTRL]+[PRINT SCR]**), or use print window (**[ALT]+[PRINT SCR]**) to copy the currently active window.

The internal buffer operates independent of the clipboard. This allows other applications to use the clipboard while HDL Bencher retains the last items selected into the buffer.

When other applications change what is in the clipboard, HDL Bencher maintains its internal buffer. This allows the last selection to be pasted inside HDL Bencher regardless of the clipboard status. If there is not enough memory to send an image to the clipboard the internal image still may be pasted inside HDL Bencher.

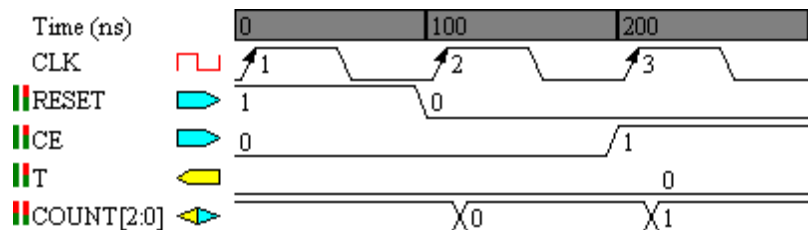
HDL Bencher supports the transfer of text between comments and applications using the clipboard. The standard Windows commands **Cut Text** (**[CTRL]+[X]**), **Copy Text** (**[CTRL]+[C]**) and **Paste Text** (**[CTRL]+[V]**), are supported for text-based objects. Also, clicking the right mouse button in edit field pops up a menu allowing a selection to be cut, copied, and pasted.

Copy

| Menu | Keyboard | Right Mouse Popup |
|----------------------------|------------------------------|-------------------|
| <u>E</u> dit, <u>C</u> opy | [ALT]+[E], [C] [CTRL]+[C] | <u>C</u> opy |

The **Copy** command places a selection image into the clipboard and the internal buffer. The internal buffer may be pasted to the waveform diagram, and the image (bitmap and metafile) in the clipboard may be pasted into other applications. To copy a waveform, select a region and execute **Copy**.

HDL Bencher supports the transfer of waveform fragments (as bitmaps or metafiles) to other programs via the clipboard. Whenever the **Copy** command is executed on a selection, a full color image of the selection is copied to the clipboard (metafile and bitmap formats). Once an image has been placed into the clipboard, it may be pasted into any Windows package that accepts bitmaps or metafiles from the clipboard. This is how many of the pictures shown in this document were developed. An example of a waveform image transferred via the clipboard is shown below:



When an area is selected into the clipboard the zoom level is ignored. Anything sent to the clipboard is shown in its unzoomed size.

When a selection is copied to the clipboard, grid spacing and the selection cursors are suppressed. To copy the video screen to the clipboard, use the Windows feature, print screen ([CTRL]+[PRINT SCR]), or use print window ([ALT]+[PRINT SCR]) to copy the currently active window.

Cut

| Menu | Keyboard | Right Mouse Popup |
|---------------------------|------------------------------|-------------------|
| <u>E</u> dit, <u>C</u> ut | [ALT]+[E], [T] [CTRL]+[X] | <u>C</u> ut |

The **Cut** command places images of a selection into the clipboard and the internal buffer, and then deletes the selection from the diagram. Once the internal buffer is filled, the selection in it may be pasted into the diagram, and the clipboard image (bitmap and metafile) may be pasted into other applications. To cut an image, select a region and execute the **Cut** command.

Insert

| Menu | Keyboard | Right Mouse Popup |
|------------------------------|------------------------------|-------------------|
| <u>E</u> dit, <u>I</u> nsert | [ALT]+[E], [I] [CTRL]+[I] | <u>I</u> nsert |

The **Insert** command inserts a blank column (or cell) at the selection point on the waveform.

Delete

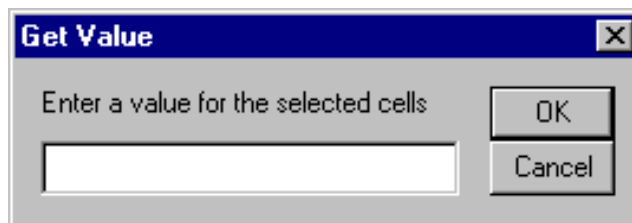
| Menu | Keyboard | Right Mouse Popup |
|------------------------------|--|-------------------|
| <u>E</u> dit, <u>D</u> elete | [ALT]+[E], [D] [CTRL]+[D] [DEL] [BACKSPACE] | <u>D</u> elete |

The **Delete** command deletes the selection (shifting the selected waveform cells to the left) without placing an image in the clipboard or internal buffer.

Set Value

| Menu | Keyboard | Right Mouse Popup |
|---------------------------------|----------------|-------------------|
| <u>E</u> dit, <u>S</u> et Value | [ALT]+[E], [S] | <u>S</u> et Value |

When a single row of cells (or part of a row) is selected, the **Set Value** command becomes active in the popup menu. Executing this command causes the **Get Value** dialog to appear. Enter the desired value (in the current radix) into this dialog.



Clear

| Menu | Keyboard | Right Mouse Popup |
|------------------------------|------------------------------|-------------------|
| <u>E</u> dit, C <u>l</u> ear | [ALT]+[E], [L] [CTRL]+[L] | C <u>l</u> ear |

When you execute **Clear** on a selection, you are asked should the selection be cleared, or the selection and all cells to the right. The command deletes the selection without shifting the selected waveform. No image is placed in the clipboard or internal buffer. This command is also available from the pop up menu (click the right mouse button on the diagram window to make the popup menu appear.)

Clear All

| Menu | Keyboard |
|--|----------------|
| <u>E</u> dit, C <u>l</u> ear <u>A</u> ll | [ALT]+[E], [R] |

The **Clear All** command deletes all cells of the waveform without shifting the selected waveform. No image is placed in the clipboard or internal buffer.

Paste

| Menu | Keyboard | Right Mouse Popup |
|-----------------------------|------------------------------|-------------------|
| <u>E</u> dit, <u>P</u> aste | [ALT]+[E], [P] [CTRL]+[V] | <u>P</u> aste |

Paste causes a copy of the internal buffer to be placed into the waveform diagram. To paste, copy a region into the buffer (via **Cut** or **Copy**), then select a position to paste into (when waveforms are being pasted), then execute **Paste Image**. If only comments are copied, a rectangle appears on the screen that should be moved to the region where the pasted image is desired (click the left mouse button to complete the operation). Waveforms may be pasted within HDL Bencher from the internal buffer. Images in the clipboard may not be pasted into HDL Bencher. Once HDL Bencher has been exited the internal buffer is flushed, but the clipboard retains whatever information it had up to that point. Pasting may be done within a waveform diagram, as long as HDL Bencher is not exited. This command is also available from the pop up menu (click the right mouse button on the diagram window to make the popup menu appear.

Copy Text

| Keyboard | Right Mouse Popup |
|------------|-------------------|
| [CTRL]+[C] | <u>C</u> opy |

Selected text may be copied to the clipboard by executing **Copy Text**. Text in the clipboard may be pasted into comments or other applications.

Cut Text

| Keyboard | Right Mouse Popup |
|------------|-------------------|
| [CTRL]+[X] | <u>C</u> t |

Cut Text, Copy Text and Paste Text are Windows standard features supported by most applications.

Selected text may be transferred into the clipboard by executing **Cut Text**. The selected text is copied to the clipboard then deleted. Once the clipboard is filled, the text in it may be pasted into other comments or other applications. The following shows selected text which may be cut or copied:

The following line of text is

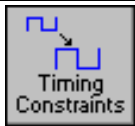
SELECTED and may be cut [CTRL]+[X] or copied [CTRL]+[C]

Paste Text

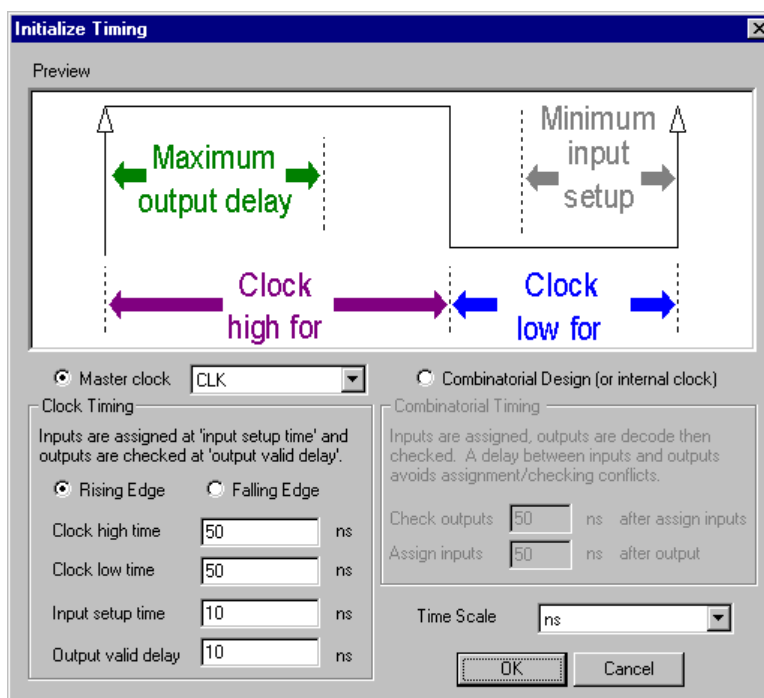
| Keyboard | Right Mouse Popup |
|------------|-------------------|
| [CTRL]+[V] | <u>P</u> aste |

Executing **Paste Text** causes the clipboard text to be placed into comments at the insertion point.

Timing Constraints

| Menu | Keyboard | Process Bar |
|--|----------------|---|
| Options, Timing Constraints | [ALT]+[O], [T] |  |

HDL Bencher lets you specify timing constraints that must be met during post-synthesis verification. All parameters are relative to the clock and are used throughout the waveform. Timing parameters can be changed at any time, allowing test cases to be updated for faster devices. When a design is imported, you are asked to choose the timing parameters.



HDL Bencher supports clocked and combinatorial designs. When a design is imported, it is parsed to determine whether it is clocked or combinatorial. The default settings are selected automatically (but may be overridden when importing).

The **maximum output delay** and the **minimum input setup time** can be specified, and are used by all test benches generated by HDL Bencher. Select the time units to be used during test benching from the **Time scale** list box.


The duty cycle of the clock is specified by setting the **Clock high time** and **Clock low time** parameters. HDL Bencher uses the active edge for the clock as found in the HDL source (if possible). The active edge is shown in the **Preview** window, and is indicated by an arrow.

HDL Bencher outputs timing constrained test benches which can be simulated within ISE. By synthesizing to a specific programmable device, you can use HDL Bencher's output to see if the timing constraints are met after synthesis.

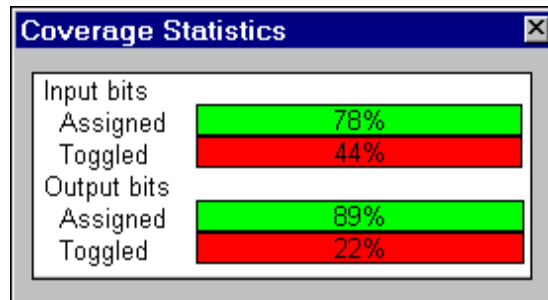
Coverage

HDL Bencher automatically shows testing coverage in a number of ways, allowing you to quickly, identify untested portions of the design, and improve design reliability. As waveforms are developed, small graphs to the left of each signal indicate coverage. The Coverage Statistics dialog provides a numerical and graphical view of total coverage. Also, an overall indicator in the waveform window shows total coverage.

Coverage Statistics

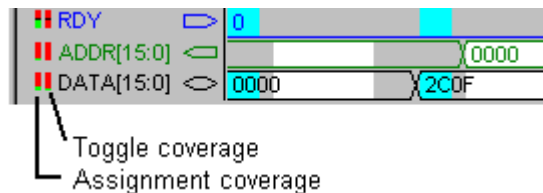
| Menu | Keyboard | Process bar |
|---|----------------|---|
| V iew, C overage Statistics | [ALT]+[V], [C] |  |

This command invokes the **Coverage Statistics** dialog that summarizes how much of the design has been exercised at any point in time. It also outlines how much each type of signal has been exercised.



The coverage statistics dialog shows the total coverage (on a bit-wise basis) for inputs and outputs. The Assign bars measures the percentage of bits that were assigned 0 and 1. The Toggle bars measure the percentage of bits that transitioned from 0 to 1 and from 1 to 0. The coverage dialog can be left open as you work on testing, and it is updated as coverage changes.

As a waveform is developed, coverage of each signal is shown via small graphs to the left of each signal's name. The coverage graphs change from red (untested) to green (tested). Moving the cursor over a graph causes a popup window to display the signals coverage.

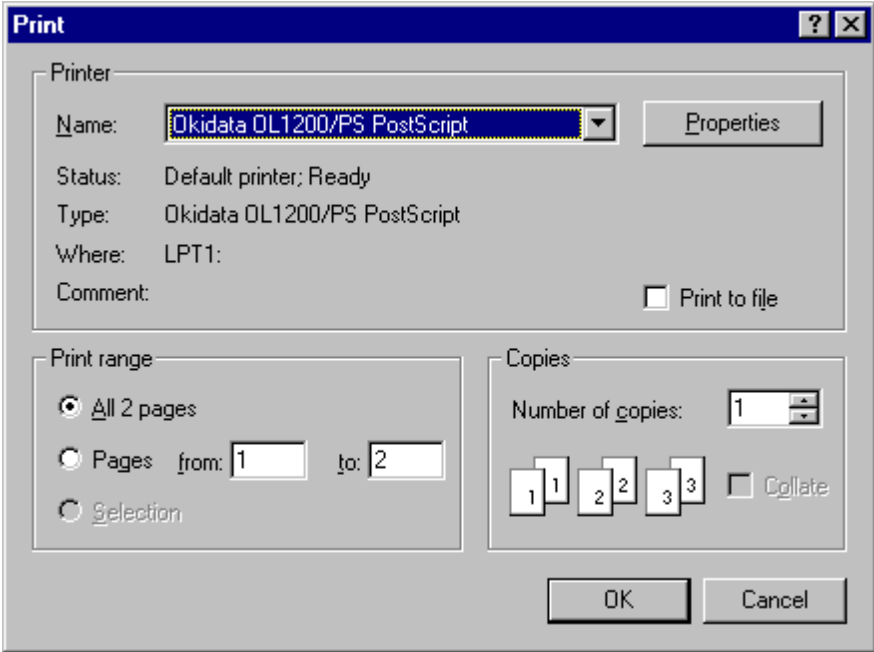


Documentation

Print Command

| Menu | Keyboard |
|-----------------------------|------------------------------|
| F ile, P rint | [ALT]+[F], [P] or [CTRL]+[P] |

Executing this command causes the Print dialog box to appear. Selected pages or the whole waveform may be printed. To print a selection, enter a range (**F**rom and **T**o fields). Clicking **OK** starts the print job.

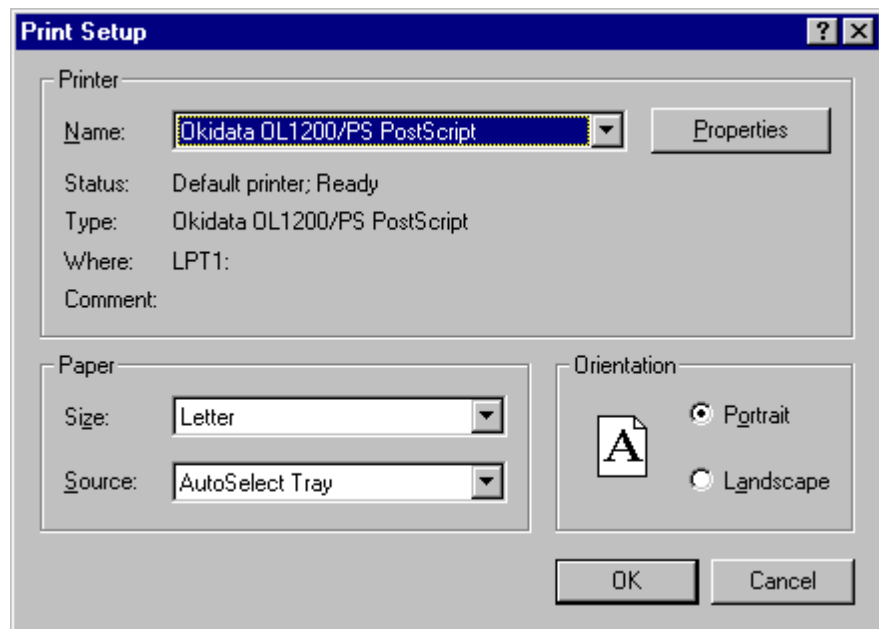


Selecting A Printer

| Menu | Keyboard |
|-----------------------------------|----------------|
| F ile, P rint Setup | [ALT]+[F], [R] |

The **Print Setup** should be used if the current printer is not the desired one or if the printer configuration is not suitable. Changing printer configurations in HDL Bencher has no effect on the printer selection of other programs.

HDL Bencher does not remember the printer last used during prior sessions. When HDL Bencher is invoked, the Windows default is assumed to be the printer that will be used. Selecting **Print Setup** allows the printer selection to be changed. Executing **Print Setup** causes the Print Setup dialog box to appear:

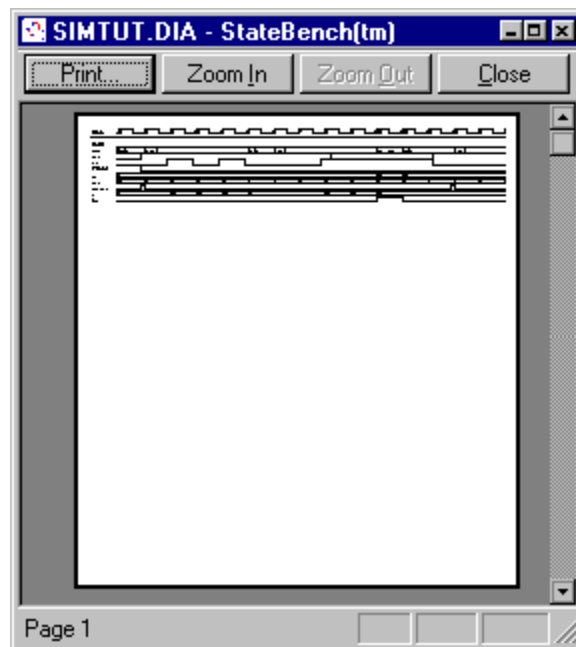


Any of the listed printers may be selected. Effect the change by clicking **OK**. The way the printer works may be modified by clicking the **Options...** button, and responding to the dialog box. Options typically include landscape and portrait orientations. Additional options may be available from the Print Manager. Switching orientations may make a diagram print on less pages.

Print Preview

| Menu | Keyboard |
|---|----------------|
| <u>F</u> ile, <u>P</u> rint <u>P</u> review | [ALT]+[F], [V] |

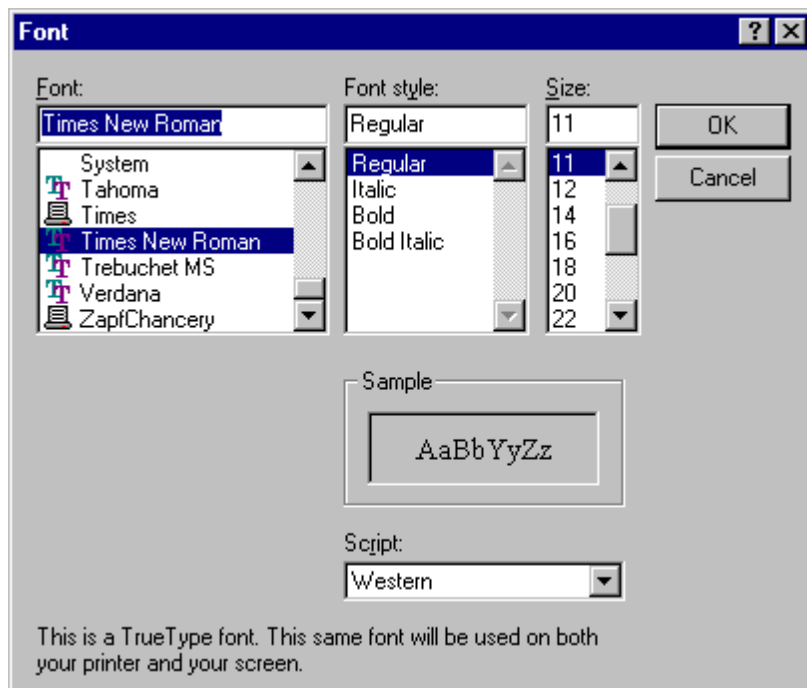
The Print Preview command allows you to see on the screen how waveforms are paginated when printed. Also, text may be clipped differently for the printer and the screen, and Preview allows you to see this as well. You can zoom in and out and see the preview as one or two pages.



Fonts

| Menu | Keyboard |
|-------------------------|----------------|
| Options, Font... | [ALT]+[O], [O] |

The way signal names and values (text) appears may be modified by changing the font used for all objects. Executing the **Font** command causes the Font dialog box to appear:



The current font name, style, and size are shown in the dialog box. When these fields are changed, a sample of the text (the way it will look on the screen) is shown in the Sample text box. To change the font, select the desired font's name from the **Name** list, the desired style from the **Style** list and the desired font size from the **Size** list. If no size is specified Windows provides a default size. The size of characters is specified in points (a point is 1/72 inch). Once these choices have been made click **OK**. The waveform is immediately updated using the new font.

Whatever font was last used is remembered between invocations of HDL Bencher and is used as the default font when a new waveform is created. Only True Type and printer fonts are supported.

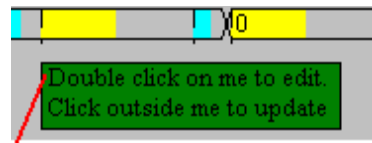
Changing the font rescales the horizontal and vertical spacing of the waveforms. Also, comments are rescaled accordingly.

Comments

Comments make waveforms easier to read. They can be moved onto the waveform to provide detail, or below the waveform as general information. Comments are automatically associated with the nearest waveform cell with respect to the top left corner of the comment. Deleting (or copying) the cell associated with the comment automatically does the same to the comment.

How to Add Comments

Clicking below the waveform automatically adds a comment.



Click here to add a comment

How to Edit Comments

Double clicking on a comment causes the edit dialog box to appear. After updating the text, click **OK**.

Line Breaks in Comments

While editing a comment, at the cursor type **[ENTER]** to enter a line break. To enter a tab character use **[CTRL]+[TAB]**.

Comment Borders

Comments may optionally have a rectangular border shown around them. Borders are enabled when editing by selecting the **Border** check box.

Comment Background

Comments may be made transparent, and placed on the waveforms to provide more detail to a local area. Comments are made transparent when editing by selecting the **Transparent** check box.

Resizing Comments

Clicking left on a comment selects the comment. Once a comment is selected, resize handles appear on the corners. Drag a resize handle to resize the comment.

Moving Comments

Drag a comment to move it.

Product Limitations

Limits And Constraints

HDL Bencher has certain limitations that are documented here to facilitate understanding, and provide workarounds when possible.

Languages

The current version of HDL Bencher supports VHDL, Verilog, and ISE schematics.

Timing Support

HDL Bencher support input set up time, output valid delay, and clock high / low times. Input setup time (and output valid delay) must not be the same as the clock high/low times. Also, hold times are not supported.

Vectors supported up to 128 bits

Vectors longer than 128 bits are not supported in HDL Bencher.

Arbitrary Timing Delays

All signal assignments and assertions occur on fixed time intervals. Signal times cannot be skewed, and inter-clock asynchronous events cannot be specified.

Multiple Clocks

Multiple clocks are not directly supported. To simulate multiple clocks that are even multiples of a master clock, use the “toggle” pattern in the pattern wizard. Repeat the pattern every *nn* (where *nn* is the multiple of the master clock). If multiple inputs are driven by the same clock, then the design should be imported as combinatorial (instead of clocked), and a identical pattern set up to toggle each of the clock lines.

Single Instance

Only one instance of HDL Bencher may run on a machine at a time.

Mode Swapping

HDL Bencher does support re-importing designs that change between clocked and combinatorial, or clock name changes when the original clock is not remapped.

Enumerated and Record data types

Bencher does not support record types, non-integer types (such as reals), or user defined enumerations except those that use the standard values 0,1,X,Z,U,H,L,W,-.

Glossary of Terms

Assertion

A warning indication that is activated when a mismatch in the simulated and expected value occurs. User defined and automatic assertions are supported.

Behavioral Verification

Guarantee the function of a design operates as intended. A simple example is five clocks after reset does the counter contain the value 5?

Cancel

A button found in dialog boxes which voids any changes or actions specified in the dialog box. If the button has changed to "close", then changes have taken place that cannot be reversed.

Click

Press and release the mouse button. Refers to the left mouse unless the right button is specified.

Clipboard

A storage area for holding information. This area is provided by Windows and may be used to transfer text to and from HDL Benchner. It can also be used to transfer bitmap or metafile images of waveform diagrams from HDL Benchner to other packages.

Clock

Provides the timing reference for registered logic. Only one may be defined per HDL Benchner diagram (see limitation). During simulation, the clock refers to the number of synchronous steps taken from the beginning of the simulation.

Combinatorial

An output signal that takes on a new value immediately in response to its inputs and has no dependencies on clock signals.

Concurrent

Operations that occur in parallel such as multiple state machines running at the same time.

Constant

A numeric value used with vectors for comparison and definition purposes. The range (of values) depends on the size of the vector and may be from zero (0) to the maximum binary value that may be represented by the number of bits in the associated vector. If the vector supports negative numbers, then the value is the range where the most significant bit is used to determine the sign.

Coverage

The process of keeping track of which I/O have been assigned, toggled and utilized during a test bench case.

Cursor

The pointer shown on the screen that follows the movement of the mouse. It is used to select, reposition, and edit as well as manipulate windows. The shape changes to reflect the current context.

Dialog box

A window used to collect or to convey information.

Drag

Moving the mouse while holding a mouse button down. Usually refers to moving or resizing objects.

Feedback

The output of a logic equation may be used as an input to the same logic equation. Asynchronous logic and latches use the outputs of logic as the inputs to the same logic.

FPGA

Field Programmable Gate Array. Large scale logic devices that can be configured by programming hardware at an end user site (not at the manufacturer).

HDL

Any one of the different Hardware Description Languages supported by HDL Bencher, such as VHDL or Verilog.

Internal buffer

Memory used by Bencher to hold graphics information. This area is separate from the clipboard and is used to copy graphical objects within HDL Bencher.

ISE

The Xilinx® Foundation Series™ Integrated Synthesis Environment is a complete design environment for Xilinx devices.

Key

Refers to the keyboard switches.

Line break

When text is in a bounded object (such as a comment), the text cannot cross the right boundary of the object. Whenever a word would cross the boundary a new line is started. The wrapping of text is done automatically, and the way text will appear when printed may be viewed using Print Preview. It is also possible to insert new line characters into text (**[ENTER]**) to force word wrapping.

Output Delay

The maximum time allowed for an output to settle and be properly recognized by other logic inputs

Pin

A physical connector on logic devices that may be assigned.

Port

Textual description of interface signals into an out of a design unit.

Post Route Verification

Validating a design's functionality after it has been synthesized and implemented into a device.

Registered

An output that changes value in response to its inputs only when clocked. If the input values change between clocks there is no effect on the output.

Setup

The minimum time for input signals to be stable before they can be sampled correctly on the next active clock edge

Source window

Place where the HDL that is to be tested is located.

Simulate

To determine the output response of a system under varying inputs.

Timing Constraints

A set of parameters that defines clock cycle duty cycle, the minimum input delay and the maximum output delay that are required for the design to behave as intended once synthesized to a specific device.

Toggle

To switch a signal from one logic state to another ($0 \Leftrightarrow 1$).

Verilog

A hardware description language used for synthesis and simulation of large designs. This language maps text based constructs into physical hardware.

VHDL

Acronym for VHSIC Hardware Description Language. VHSIC is an acronym for Very High Speed Integrated Circuits. This language is used for designing digital logic.

Waveform window

The place where waveform diagrams are drawn, edited, and modified. All major operations are accessed from it.

WaveTable

Cell based spreadsheet entry overlaying a graphical waveform. Allows complex data values to be assigned in the correct time positions by simply typing the values.

Xilinx

Xilinx is the leading producer of programmable logic. It produces FPGAs, CPLDs, and software design tools needed to design and program devices.

Index

A

About 8
active 23, 33
add comments 39
assign 25
automatic scrolling 12
automatic test bench 34

B

behavioral verification 1
binary 6
Bit map 28
Borders 39

C

cell enable 23
changing signal width 5
clear 31
clear all 31
clipboard
 copy text 32
Clipboard 28
 Copy image 28
 Copy text 28
 Cut image 28
 Cut text 28, 32
 Paste image 31
 Paste text 28, 32
Clipboard - copy 29
Clipboard - cut 30
Clipboard - paste 21
close waveform 21
Color 28
color scheme 2
comment background 39
comments 39
 text 32
configuration 22
constrain 1
constrain timing 33–34

context sensitive help 9

Conventions 3

Copy 29

Copy Image 28

copy text 32

Copy Text 28

copying

text 32

Copying

 Images 28

 Text 28

Copyright 8

coverage 34

coverage statistics 34

cursor 29

Cut 30

Cut Image 28

Cut text 28, 32

D

data entry 25

data types 15

data values 24

decimal 6

delay 33

delete 30

Directory 19, 20

duty cycle 33

E

edit comments 39

end of test bench 18, 22

enter values 25

entity 14

executing commands 4

exit 8

expected results 1

export 14, 17, 18, 22

extension 22

F

Filename 20, 21
fonts 38

G

generic 14
generics 17
glossary 41
grid 23, 29

H

HDL Bencher 1
help 9
 context sensitive 9
hex 6

I

import 13, 14, 15, 16
importing HDL 13
index 9
input setup time 25
insert 30
Internal buffer 28
 Copy Image 28
 Cut image 28
 Paste image 31
introduction 1
io mode 2
ISE 13, 19

L

Line breaks 39

M

manual port update 16
mapping
 constants 16
 data types 15
menu 4
modify ports 16
move comments 39
Moving through the waveform 11
mru list 19

O

open waveform 19
order 9
ordering information 9
output delay 33
output valid delay 25

P

parameter 14, 17
Paste 31
Paste text 28, 32
Pasting
 Paste text 28, 32
 Pasting images 31
pattern wizard 24, 26
popup menu 10, 31
port 14
port io 6
ports 6, 17
print command 35
print preview 37
print setup 36
printer selection 36
problems 14

R

radix 6
refresh 8
Region 27, 29
reimport 16
remap 16
reorder signals 23
rescale timing 33
resize bar 5
resize comments 39
Revision control 17

S

save as waveform 20
save waveform 21
Scroll 12
scroll keys 12
Select All 27
selection 27
setup 33
setup time 33
signal 34
simulate 33
simulator 33
source editor 2, 6, 14
special characters 3
spreadsheet 24
spreadsheet 25
Status bar 2, 5
synthesis 33

T

technical support 8
termination 18, 22

test bench 1, 17
Text 32
text compression 11
the windows 2
time stamp 17
time unit 33
timing 14, 24
timing verification 33
Title 20, 21
toggle 25
tool bar 4
toolbar 5
toolbar commands 4
tooltips 6

U

Undo 10
unit under test 13, 14
unknown data type 14
unknown value 15
Unselecting 27
untested 34
update ports 16
 uut 13, 14

V

validate 1
value editor 25, 26
values 25
verification 33
verilog 1
VHDL 87 22
VHDL 93 22

W

waveform 25, 34
waveform window 2, 19
wavetable 1, 24, 25
what is HDL Bencher 1
White Characters 3

X

Xilinx 13, 19

Z

zoom 11, 28, 37
zoom to 11