

Dinotrace

Signal Waveform Tracing
for Dinotrace 9.4f
October 2019

Wilson Snyder, wsnyder@wsnyder.org

Copyright © 1998-2014 Wilson Snyder

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the author.

1 Summary

Dinotrace is a tool designed to aid in viewing Verilog Value Change Dump, ASCII, Verilator, Tempest CCLI, COSMOS, Chango and Decsim Binary simulation traces. It is optimized for rapid design debugging using X-Windows Mosaic. A special interface allows signal information to be annotated into source code using Emacs.

Please see the file `README.adoc` in the distribution directory for installation instructions.

1.1 Copying

Dinotrace is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3, or (at your option) any later version.

Dinotrace is distributed in the hope that it will be useful, but **without any warranty**; without even the implied warranty of **merchantability** or **fitness for a particular purpose**. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with Dinotrace; see the file `COPYING`. If not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

2 History

Dinotrace was conceived in the early 1980's by Allen Gallotta at Digital Equipment Corporation, who wrote the code and supported it through version 4.2. When created, it was the first graphical display tool for the simulators being designed at Digital.

In 1992 I joined Digital Semiconductor (a division of Digital Equipment Corporation) and took over the support of Dinotrace. I converted it from Digital's X toolkit to Motif, added color, highlighting, searching, and other features. Several formats were added to Dinotrace to support Verilator, our group's own Verilog to C translator, written by Paul Wasson. As the years went on, other groups started using the typical industry standard Verilog simulators. Although programs like vwaves were available, they had some silly limitations. Besides, we were hooked on Dinotrace, so I added Verilog format, and Dinotrace lived on.

In 1998 I moved to Maker Communications, where I am currently employed. Life, however, was sad and incomplete without Dinotrace. Through the efforts of existing DECies, mainly Duane Galbi, Digital graciously agreed to release the source code. (Subject to the code not being resold, which is compatible with the GNU Public License.)

Still at Maker Communications, life is now complete as Dinotrace can be shared by all. This release is based upon the Digital version, with many enhancements, most notably portability improvements and true 4-state signal support.

2.1 Contributors

Many people have provided ideas and other assistance with Dinotrace.

The major corporate sponsors of Dinotrace, by providing funds or equipment grants, are Compaq Corporation, Digital Equipment Corporation, Intel Corporation, and Maker Communications.

The people who have contributed code or other major functionality are Allen Gallotta, Steve Hoover, Nathan Hughes, and Craig Ziegler.

Some of the people who have provided ideas and feedback for Dinotrace include Edward Arthur, Sally Barry, Josh Belkin, Yifat Ben-Shachar, Lee Bradshaw, Rachel Berman, Sass Berman, Mike Blake, Serge Bogdanov, Ronen Boneh, Gregg Bouchard, Derek Bosch, Bill Bruce, Lauren Carlson, Paul Chan, Ta-Chung Chang, David Chen, Dave Colson, Derrick Dacosta, Erik Debriac, Paul Dormitzer, Duane Galbi, Allen Gallotta, Steve Glaser, Mike Goulet, Larry Gust, Larry Herman, Dave Horan, Steve Hoover, Nathan Hughes, Tom Hunt, Tracey Jones, Mike Kagen, Dan Katz, Sol Katzman, Steve Kolecki, Steve Lang, Charlie Lind, Greg Loxtercamp, Dan Lussier, Ben Marshall, Bob McNamara, Harunobu Miyashita, John Murphy, Aki Niimura, Lisa Noack, Mitch Norcross, Michael Quadri, Reinhard Schumann, Simcha, Jai Singh, Dean Sovie, Julie Staraitis, Dominik Strasser, Bob Walsh, Paul Wasson, Steve Wilson, Bob Yodlowski, and Craig Ziegler.

3 Invoking Dinotrace

Dinotrace accepts switches and filenames on the command line. Any filenames on the command line will be read in before Dinotrace gives control to the user.

Dinotrace accepts switches with leading `-`, and everything else is a trace filename. Filenames are in standard Unix format, with a optional `.gz` or `.Z` extension to indicate the trace should be piped before opening through **gunzip** or **uncompress** respectively.

These switches are supported, in addition to the standard **X-11** switches, like `-display`.

`-verilog`

Read following trace filenames using Verilog format. This is the default format.

`-vpd`

Read following trace filenames using Verilog Plus Dump format.

`-decsim`

Read following trace filenames using Decsim format. If running on a VMS system, Decsim Binary format is also supported.

`-tempest`

Read following trace filenames using Tempest/CCLI format.

`-noconfig`

Skip reading the standard global, user, and directory `.dino` files. Do read any trace-specific `.dino` files.

`-geometry geometry`

Specifies the window size and position in standard X format (*widthXheight+xpos+ypos*).

`-res resolution`

Specify the trace resolution.

4 Screen Description

This chapter describes the layout of the screen, and the function of the various widget and menu items.

4.1 Window Widgets

The main window is broken into the display area, with signal names on the left and values stretching to the vertical scroll bar.

The state of each signal across time is indicated by the way the line is drawn. As usual, low, tristate, and high are low, middle, and high strait lines. The high value has a bolder line to aid in discriminating it from a low. Unknowns are filled areas.

Busses are drawn in one of four ways; if zero they are low, if all bits are one it is two parallel lines with the upper one bolded. If the bus is some mix of zeros and ones it is two narrow parallel lines. If there is space the value of the bus will be shown based on the **Signal Radix** menu option. Finally, the **Signal Waveform** allows analog waveforms, or signed waveforms (two's complement with 0 making a centered trace.)

Gadgets surround the main area. The vertical scroll bar on the right is the *Signal Slider* used to scroll signals, if they all cannot fit on the screen at once. At the bottom of the screen is the *Time Slider* on the right, and the *Name Slider* on the left.

4.1.1 Time Slider

On the left bottom of the main window is the *Time Slider*.

Drag

Dragging the *Time Slider* will scroll the time across the screen. Lines are printed in the scroll area to indicate where cursors are located.

Unit Dec

Clicking the <- arrow at the left of the *Time Slider*, or pressing **left** will move the time one grid #0 (or the first visible grid) resolution earlier.

Unit Inc

Clicking the -> arrow at the right of the *Time Slider*, or pressing **right** will move the time one grid #0 resolution (usually one clock) later.

Page Dec

Clicking the blank area at the left of the *Time Slider*, or pressing **S-right** will move the time one screen width earlier, so the time at the left of the screen is now at the right. (Or 1/2 or 1/4 screen width, depending on the **page_inc** customization setting.)

Page Inc

Clicking the blank area at the right of the *Time Slider*, or pressing **S-right** will move the time one screen width later, so the time at the right of the screen is now at the left. (Or 1/2 or 1/4 screen width, depending on the **page_inc** customization setting.)

4.1.2 Name Slider

The *Name Slider* is the horizontal scroll bar left of the *Time Slider*.

Drag

Dragging the *Name Slider* will scroll the signal names left and right so that they may be completely seen. Dinotrace aligns the signals on the last dot (hierarchy character).

4.1.3 Signal Slider

The *Signal Slider* is the vertical scroll bar to the right of the window.

Drag

Dragging the *Signal Slider* will scroll the signals so that new signals can be seen. Lines are printed in the scroll area to indicate where highlighted signals are located.

Unit Dec

Clicking the <- arrow at the top of the *Signal Slider*, or pressing **up** will move the signals down one, displaying a new signal at the top.

Unit Inc

Clicking the -> arrow at the bottom of the *Signal Slider*, or pressing **down** will move the Moves signals up one, displaying a new signal at the bottom.

Page Dec

Clicking the blank area above the *Signal Slider*, or pressing **pageup** will move the signals up one screen full, so the previous top signal is now at the bottom.

Page Inc

Clicking the blank area below the *Signal Slider*, or pressing **pagedown** will move the signals down one screen full, so the previous bottom signal is now at the top.

4.1.4 Buttons

The buttons below the lower sliders perform common functions that affect the display.

Begin

Clicking the *Begin* button below the *Time Slider*, or pressing **begin** will move to the earliest time in the trace.

End

Clicking the *End* button below the *Time Slider*, or pressing **end** will move to the earliest time in the trace.

Goto

Clicking the *Goto* button below the *Time Slider*, or pressing **g** will make a popup that prompts for a time. The window will scroll horizontally so that the entered time is centered on the screen. A option menu will place a cursor of the selected color at the time entered. A note may be attached to the cursor, and seen with **Value Examine**.

Refresh

Clicking the *Refresh* button below the scroll bar will refresh the screen. Normally this is automatic. When using Dinotrace over slow links, it may be desirable to refresh manually, See Section 6.2.11 [refreshing], page 17.

Resolution

Clicking the *Resolution* number button at the middle bottom will make a popup that can change the resolution to the number specified.

Inc Res

Clicking the arrow button to the right of the *Resolution* button expands the trace so the same amount of time takes more space.

Dec Res

Clicking the arrow button to the left of the *Resolution* button expands the trace so the same amount of time takes less space.

Zoom

Clicking the *Zoom* button, then clicking MB1 on two times expands the trace so that the time between the clicks occupies the entire screen.

Full

Clicking the *Full* button shrinks the resolution so that the entire trace fits onto the screen. It is often quicker to press *Full* then zoom to the desired location rather than scrolling.

4.2 Trace Menu

4.2.1 Trace Read...

Selecting the **Trace Read...** menu option, or pressing **C-o** pops up a file requester to read a trace. If a file ends in a **.Z** or **.gz** extension, it will be piped through **uncompress** or **gunzip** respectively before being read. The type of trace may be set with the option box; the default is set by the **file_format** configuration command. If *Preserve Signal Ordering* is checked, the signals in the new trace will be reorganized to attempt to match the ordering of the current trace in the window.

4.2.2 Trace ReRead

Selecting **Trace ReRead** checks if there is a new version of the file currently begin displayed and reads it. The file will only be read if a new one exists. If no file is displayed, it is the same as the **Trace Read** option.

This is useful for rereading a trace after it has changed due to a new simulation run.

4.2.3 Trace ReRead All

Selecting **Trace ReRead All**, or pressing **C-r**, checks every window to see if that window's file has been updated, and rereads if it has.

4.2.4 Trace Open...

Selecting **Trace Open...** shrinks the current window in half, and opens another Dinotrace window, linked in time to the original. The two traces will then pan together and share cursors, etc. Multiple window support is intended for traces that have the same time lengths. See Section 6.2.35 [open_geometry], page 20, for how to change the default window sizes, such as to not shrink the original window when opening.

4.2.5 Trace Close

Selecting **Trace Close** closes the current window if more than one is open. If this is the only one remaining open, use **Trace Exit** instead.

4.2.6 Trace Clear

Selecting **Trace Clear** closes all other windows and resets Dinotrace as if Dinotrace was exited and ran again.

4.2.7 Trace Print...

Selecting **Trace Print...** pops up a file requester allowing the current screen to be printed to a file. The amount of time, and number of signals on the screen will match the time and signals put onto one page of paper.

The *Filename* and *Note* specify where to put the output, and a note to be placed on the bottom of each page.

The *Layout* option selects the size of the page. *EPS* will make a Encapsulated Postscript file that does not include all of the headers. This is suitable for making figures for inclusion in other documents.

The *Include off-screen times* button will print multiple *horizontal* pages of paper, so that all times in the trace is printed. The *Include off-screen signals* button will print multiple *vertical* pages, so that all signals are printed. Both may be checked and taped together to get a nice wall hanging.

The *Begin printing at* and *End printing at* select the region to be printed. This may be based on the whole trace, what is displayed on the screen, or the cursors.

4.2.8 Trace Print

Selecting **Trace Print** performs a print using the last **Trace Print...** settings.

4.2.9 Trace Exit

Selecting **Trace Exit** will exit Dinotrace.

4.3 Customize Menu

4.3.1 Customize Change...

Selecting **Customize Change** pops up a requester that allows the configuration to be customized. See Chapter 6 [Customization Files], page 15, for permanent configuration, and additional options that are not selectable here.

The *Page Increment* option selects how much of a screen clicking on the *Page Inc* space on the *Time Slider* will increment.

The *Time Representation* option selects how times are displayed and entered in the *Goto* popup. Selecting *Grid #0 Cycles* will put all times in the period defined by Grid #0.

The *Signal Height* option controls how many signals can fit on the screen and printout. Small sizes have more information, but may cause the values to be hard to read.

The *Click To Edge* button enables **Cursor Add** and other menu items that require mouse clicks to find the closest transition and add it at that point. The search will be only within 20 pixels; if there is no edge within that distance, the cursor is added right where the cursor is.

The *Rise/Fall Time* slider controls how fast signals transition.

The *Draw Prefix* enables displaying the entire signal name, instead of the default where a prefix common to every signal in the design is not shown.

The *Manual Refreshing* button disables automatic refreshing, for dialup lines. The *Refresh* button must be pressed to refresh the screen.

4.3.2 Customize Grid...

Selecting **Customize Grid** pops up a requester that allows the configuration to be customized. See Chapter 6 [Customization Files], page 15, for permanent configuration. Note that the lowest numbered grid that is visible is used for moving one increment, stepping cursors forward and back, and for counting cycles.

4.3.3 Customize Read...

Selecting **Customize Read** pops up a requester that reads the specified `.dino` file. The requester also allows enabling and disabling of which `.dino` files will be read when a new trace is read.

4.3.4 Customize ReRead

Selecting **Customize ReRead** will reread the customize `.dino` files. This is useful for applying changes to Dinotrace after a `.dino` file has been edited outside Dinotrace.

4.3.5 Customize Save As...

Selecting **Customize Save As...** will create a popup. A filename may then be entered to save the current customization settings. Buttons allow different portions of the configuration to be written. Note that many of the settings that will be saved can be applied only to one window, and should be edited out before being included in any global `dinotrace.dino` files.

4.3.6 Customize Restore

Selecting **Customize Restore** will restore the customization options to their program defaults. It will not read any `.dino` files.

4.4 Cursor Menu

4.4.1 Cursor Add

Selecting **Cursor Add**, or pressing `c`, then clicking **MB1** will insert cursors at the clicked on times. The **Next** color button, or pressing `C` will select the next color down in the menu

from the last color selected. See Section 6.2.28 [cursor_add], page 19, for the equivalent configuration file command.

4.4.2 Cursor Highlight

Selecting **Cursor Delete** then clicking MB1 on cursors will change the color of those cursors.

4.4.3 Cursor Move

Selecting **Cursor Move** then Dragging MB1 on a cursor will move that cursor. Press and hold mouse button one and release when the cursor is at the new time desired. Moving a cursor that was created with **Signal Search** will make that cursor permanent; the cursor will not go away if the search is changed.

4.4.4 Cursor Delete

Selecting **Cursor Delete** then clicking MB1 on cursors will delete those cursors. To delete cursors that were created by **Value Search** (those with dotted lines), turn off the second button on the **Value Search** requester.

4.4.5 Cursor Note

Selecting **Cursor Note** then clicking MB1 on a cursor will pop up a dialog box to change the note attached to that cursor.

4.4.6 Cursor Step Forward

Selecting **Cursor Step Forward** or pressing > will move all user cursors forward the by grid #0's (or the first visible grid's) period.

4.4.7 Cursor Step Backward

Selecting **Cursor Step Back** or pressing < will move all user cursors backward the by grid #0's (or the first visible grid's) period.

4.4.8 Cursor Clear

Selecting **Cursor Clear** will delete all cursors.

4.4.9 Cursor Cancel

Selecting **Cursor Cancel**, or pressing esc will stop MB1 mouse clicks from having any affect.

4.5 Signal Menu

4.5.1 Signal Add...

Selecting **Signal Add** will pop up a popup allowing a signal to be selected then added at the point where MB1 is clicked. See Section 6.2.39 [signal_add], page 21, for the equivalent configuration file command.

4.5.2 Signal Highlight

Selecting **Signal Highlight**, or pressing s, then clicking MB1 on signals changes the color of those signals. See Section 6.2.38 [signal_highlight], page 21, for the equivalent configuration file command.

4.5.3 Signal Move

Selecting **Signal Move** then clicking MB1 on a signal and clicking MB1 again on a new position moves a signal. Signals may be moved between traces, but closing a trace eliminates all signals that were originally from that trace. See Section 6.2.43 [signal_move], page 21, for the equivalent configuration file command.

4.5.4 Signal Copy

Selecting **Signal Copy** then clicking MB1 on a signal and clicking MB1 again on a new position copies a signal. Signals may be copied between traces, but closing a trace eliminates all signals that were originally from that trace. See Section 6.2.42 [signal_copy], page 21, for the equivalent configuration file command.

4.5.5 Signal Delete

Selecting **Signal Delete** then clicking MB1 on signals deletes those signals. **Signal Select** may be used to recover the deleted signals. See Section 6.2.40 [signal_delete], page 21, for the equivalent configuration file command.

4.5.6 Signal Note

Selecting **Signal Note** then clicking MB1 on a signal will pop up a dialog box to change the note attached to that signal.

4.5.7 Signal Radix

Selecting **Signal Radix** then clicking MB1 on signals changes the radix of the signals to the one selected. Note that the **real** radix is detected for Verilog automatically, and changing other non-real signals to a real format probably won't display the values you wish. See Section 6.2.46 [signal_radix], page 22, for the equivalent configuration file command.

4.5.8 Signal Waveform

Selecting **Signal Waveform** then clicking MB1 on signals changes the signal to be displayed as either analog or digital. See Section 6.2.47 [signal_waveform], page 22, for the equivalent configuration file command.

4.5.9 Signal Search...

Selecting **Signal Search** opens a popup for highlighting signal names. Nine signal names which will be searched for in the trace are specified. Any signal name which matches the signal name pattern will be highlighted using the color of the toggle button, just as if **Signal Highlight** was used to color it. Unlike **Signal Search**, highlighting a signal with **Signal Highlight** makes it permanent; it will not go away when the search that found that signal changes.

The *Signal name pattern* indicates which signal names to highlight, potentially containing * to match any number of characters, or ? to match any single character. Case never matters when matching signal names.

The *Enable* button enables the search, and shows the color to be used for highlighting when the signal is found.

4.5.10 Signal Select...

Selecting **Signal Select**, or pressing `C-f`, presents a requester which lists the names of the signals which are being displayed, and those which are deleted. Both lists have a pattern at the top which a signal must match to be displayed. Clicking on a signal will add it to the bottom of the other list. Clicking on the *Add All* or *Delete All* will affect all of those in the list. By specifying a pattern then *Add All* or *Delete All*, signals which match a pattern can be added or deleted.

The *Delete Constants* button will delete only those signals which have a single constant value for all times in the trace. Likewise the *Delete Const or X/Z* button will delete only those signals which have a constant value, or are unknown or tristated. This is useful for deleting those signals which have uninitialized values before reset then later in the trace are initialized to constants.

The *Sort Wholename* and *Sort Basename* buttons will sort the selected signals and place them at the bottom of the trace. The former uses the entire signal name, while the latter sorts by the basename, ignoring the hierarchy portion of the signal name.

4.5.11 Clear Highlight

Selecting **Signal Clear Highlight** will remove all highlighting from signals.

4.5.12 Keep Highlight

Selecting **Signal Keep Highlighted** will delete signals from the trace which are not highlighted. This allows signals to be highlighted that match a given pattern using *Signal Select...*, then all others deleted using this selection.

4.5.13 Signal Cancel

Selecting **Signal Cancel**, or pressing `esc` will stop MB1 mouse clicks from having any affect.

4.6 Value Menu

4.6.1 Value Annotate...

Selecting **Value Annotate...** makes a file with the highlighted signals, and values under every cursor. Behavioral code can then be highlighted with the values by `dinotrace.el`'s reading the written file. Cursors which are exactly on a edge of a signal will get the value that the signal is changing *to*, not from.

Buttons specify which cursor and signal colors are included in the file. If a button is on, the color is included, else it is as if the cursor did not exist, or the signal was not highlighted.

User cursors are those placed using the **Cursor Add** menu. Automatic cursors are from searching for values or cursors placed by `.dino` files.

See Chapter 9 [Dinotrace and Emacs], page 25,

4.6.2 Value Annotate

Selecting **Value Annotate**, or pressing `a` annotates code using the last **Annotate...** settings. See Section 6.2.14 [annotate], page 17, for the equivalent configuration file command.

4.6.3 Value Highlight

Selecting **Value Highlight**, or pressing *v*, then clicking MB1 on a value loads that value into one of the 9 search values, and highlights it where it occurs in the trace. The search value can then be seen and modified with the **Value Search** popup. See Section 6.2.37 [value_highlight], page 20, for the equivalent configuration file command.

4.6.4 Value Examine

Selecting **Value Examine** then holding MB1 will show the state of the entity under the mouse. MB2 can always be used for this feature.

Signal transitions can be examined when over the transition lines. Signal names and base information can be seen when over the signal names. Cursors can be examined when at the bottom of the screen, and grids when at the top.

4.6.5 Value Search...

Selecting **Value Search...** allows specification of 9 values to be searched for in the trace. Each value is entered into a text widget and one or two toggle buttons to the left are activated. The **Value Highlight** menu can be used to load a search automatically. See Section 6.2.37 [value_highlight], page 20, for the equivalent configuration file command.

The *Highlight value* button will make anywhere that the value appears in a trace to be printed using the color of the toggle button.

The *Add cursor* button will add a cursor at the time where the value begins. Cursors that are placed in this fashion are displayed as dotted lines, and will go away if the search value is changed. Highlighting or moving a cursor makes it permanent; it will not go away when the search changes.

The *Signal name pattern* indicates which signals names to search for the value, potentially containing *** to match any number of characters, or *?* to match any single character. Often a single *** is useful to search all signals.

The *Value* indicates the value to be found. Unfortunately wildcards are not supported. A standard verilog or VHDL prefix may be added to specify the radix of the number being entered. (The radix of the signal found does not have to match the radix entered; `'h10'` will match `'d16'`).

`'h<num>`

`X"<num>"` Hex

`'o<num>`

`O"<num>"` Octal

`'b<num>`

`B"<num>"` Binary

`'d<num>`

`D"<num>"` Decimal

`'r<num>`

`R"<num>"` Real

`'s<strg>`

S"<strg>"

"<strg>" Ascii (String)

4.6.6 Value Cancel

Selecting Value Cancel, or pressing `esc` will stop MB1 mouse clicks from having any affect.

5 Keyboard Shortcuts

This chapter describes the keyboard shortcuts for Dinotrace.

Any menu function may be invoked with the standard Motif accelerators; hold down **ALT** then press the first letter of the menu name. Release **ALT** and press the underlined letter of the sub menu item, and so on.

The following keys are also hard-coded: Function keys have the identical function when using GNU Emacs's Verilog, C, Merlin, or CCLI modes.

MB2	Value Examine.
up	Signal scroll up one signal.
S-up	Signal scroll up a screen full.
down	Signal scroll down one signal.
S-down	Signal scroll down a screen full.
pageup	Signal scroll up a screen full.
pagedown	Signal scroll down a screen full.
left	Time scroll left a grid #0 increment.
S-left	Time scroll left a screen full.
right	Time scroll left a grid #0 increment.
S-right	Time scroll left a screen full.
begin	Beginning of time.
end	End of time.
esc	Signal Cancel.
>	Cursor Step Forward.
<	Cursor Step Backward.
a	Value Annotate.
c	Cursor Add with current color.
C	Cursor Add with next color.
g	<i>Goto</i> time.
s	Signal Highlight with current color.
S	Signal Highlight with next color.
v	Value Highlight with current color.
V	Value Highlight with next color.
C-f	Signal Search....
C-o	Trace Read....
C-r	Trace ReRead All.

6 Customization Files

This chapter describes customization files, and the commands that may be used in them.

Customization files allow the options and display in Dinotrace to be customized to suit a specific site, user, directory, and trace.

6.1 Automatically Read Files

When a trace file is read using the Trace Read menu option, several configuration files are read automatically:

`$DINODISK:dinotrace.dino`

The `$DINODISK:dinotrace.dino` file is intended to have site wide defaults. This will probably contain a `file_format verilog` or similar commands. (On Unix, `$DINODISK` represents the contents of the `DINODISK` environment variable. On VMS, it represents the value of the system define.)

`$DINOCONFIG`

The `$DINOCONFIG` file is intended to be used for session specific start-up, like setting screen size depending on what type of system you are using. Users will probably point this to several possible sources using their `.login`.

`~/dinotrace.dino`

The `~/dinotrace.dino` file is for user defaults. The author for example has a `signal_height 15` and `signal_delete_constant *`, which are both personal preference items. (On VMS, `SYS$LOGIN:` is used instead of `~/.`)

`{current/trace/directory}/dinotrace.dino`

The `{current/trace/directory}/dinotrace.dino` is for `signal_states` and other information on a per-simulation setup basis. This probably includes `signal_state` commands (unless they are in the global `dinotrace.dino`).

`{current/trace/directory}/CURRENT_TRACE_NAME.dino`

The `{current/trace/directory}/CURRENT_TRACE_NAME.dino` is for highlighting and such that only apply to the current file. See the enclosed example **Perl** script that will parse a log file for times that errors occurred and write a dino that places a cursor at every such time.

6.2 Configuration Commands

Configuration files are organized with one command per line, with the exception of `signal_states`, which may be multiple lines. Case is ignored in all but state names. The commands are:

6.2.1 `#`

`# comment`

Comments out the remainder of the line.

6.2.2 page_inc

`page_inc 4 | 2 | 1`

Changes the increment used during *Page Inc* and *Page Dec* clicks on the horizontal scroll-bar. 1=full page, 2=half, 4=quarter. Same as the page option menu in *Customize Change*.

6.2.3 rise_fall_time

`rise_fall_time number`

Changes the slope used when drawing signal assertions and deassertion. Use 0 to minimize the `rise_fall_time`.

6.2.4 signal_height

`signal_height number`

Changes the vertical spacing between signals on the screen. Same as the slider in *Customize Change*.

6.2.5 grid

`grid grid_number ON | OFF`

Enables or disables the grid. Same as the visible toggle buttons in *Customize Grid*.

6.2.6 grid_resolution

`grid_resolution grid_number number | AUTO | EDGE`

Changes the horizontal space between grids, normally the period of a clock. Specify a number in nanoseconds, or `AUTO` to compute the grid based on the `grid_signal`. Use `EDGE` to place a grid on each change of the `grid_signal`.

6.2.7 grid_align

`grid_align grid_number number | ASSERTION | DEASSERTION | BOTH`

Changes the horizontal location of the first grid mark. Specify a number in nanoseconds, 'ASSERTION' to align with the asserting edge of the `grid_signal`, or 'DEASSERTION' to align with the deasserting edge of the `grid_signal`, or 'BOTH' to align with both edges of the `grid_signal`. The grid may also align with changes in a bus, although in that case 'ASSERTION' and 'DEASSERTION' have the same effect.

6.2.8 grid_signal

`grid_signal grid_number pattern`

Specifies which signal is to be used when calculating the period and alignment with `grid_resolution` and `grid_alignment` respectively. The first signal that matches the pattern is used, it defaults to "*", which is the first signal in the trace. If no signal is found that matches the pattern, that grid will be disabled.

6.2.9 grid_type

`grid_type pattern NORMAL | WIDE`

Specifies normal grids, or extra wide grids. Wide grids are plotted 2 pixels wide, and help low-contrast colors show up better.

6.2.10 `grid_color`

`grid_color` *grid_number* *color*

Changes the color used to display the grid.

6.2.11 `refreshing`

`refreshing` AUTO | MANUAL

In automatic refresh mode, any change in the screen such as zooming, Dragging, or exposing refreshes (redraws) the screen. When set to manual the screen will only redraw when the refresh gadget is pressed. Note that when in manual mode, it is just as if the display *was* refreshed, so for example clicking on the display is relative to the *new* display. Useful when using slow network links, such as dialup **eXcursion** or **exceed**.

6.2.12 `refresh`

`refresh`

Refresh the screen, even if in manual refresh mode. When a `.dino` file is being read, the screen will not be refreshed until the entire file is completely read, unless a `refresh` command is used.

6.2.13 `draw_prefix`

`draw_prefix`

By default, a prefix common to every signal in the design is not shown in the main window. With this setting, the signal name will be shown in it's entirety.

6.2.14 `annotate`

`annotate`

Makes the annotation file, like the Value Annotate menu.

6.2.15 `resolution`

`resolution` *number*

Sets the screen resolution, like the *Resolution* Button on the bottom of the screen.

6.2.16 `save_enables`

`save_enables` ON | OFF

When reading in verilog CCLI traces, Dinotrace converts two-state enables and data signals into a single four-state signal, then deletes the original signals. When `save_enables` is on, the original signals are not deleted. This is useful for debugging problems related to tristating.

6.2.17 `save_duplicates`

`save_duplicates` ON | OFF

When reading in verilog traces, Dinotrace by default drops signals that connect to a signal higher in the hierarchy. This saves space and reduces clutter. When `save_duplicates` is on, the duplicate signals are not deleted, except under rare situations.

6.2.18 `save_ordering`

`save_ordering` ON | OFF

When set, Dinotrace will attempt to preserve signal colors, ordering, and deletions between the trace that existed before and the new trace being read in. Signals that were deleted because they were constants are not preserved, they will be recalculated. When clear, the new trace does not preserve any ordering information.

6.2.19 `signal_goto`

`signal_goto` *signal*

Move the window so that *signal* is in the center of the window. If the signal was already on the screen, do nothing.

6.2.20 `time_goto`

`time_goto` *time*

Move the window so that *time* is in the center of the window. If *time* was already on the screen, do nothing.

6.2.21 `time_rep`

`time_rep` *number* | FS | PS | NS | US | CYCLE

Changes the time representation to be in picoseconds, nanoseconds, microseconds or in number of grid #0 cycles. Cycle based timing is good for performance estimates. Same as the buttons in **Customize Change**. If a number, then that number is the baseline in picoseconds. For example, '2000' will make one unit on the screen be equal to 2 ns in the trace file. Default is nanoseconds.

6.2.22 `time_precision`

`time_precision` FS | PS | NS | US

Changes the internal precision to be in picoseconds, nanoseconds, or microseconds. Traces that are read in will have all times rounded to this number. Setting to higher units requires less memory and speeds up execution. This parameter must only be changed in the global or user `dinotrace.dino` files, as it should not change between reading different traces. Default is nanoseconds.

Only Decsim Binary and Verilog formats contain absolute time periods, and these will be scaled by the precision. Decsim Ascii traces are always assumed to be in nanoseconds. The Tempest format will be assumed to be in the same units as the `time_precision`, and will not be scaled.

6.2.23 `time_format`

`time_format` %0.6lf | %0.6lg | %lg | ""

Changes the number of decimal places used to print times. This format string is the same as that used by the C-Language `printf` command. (See *man printf* or *HELP CC RUN PRINTF*.) Some of the useful choices are '%0.6lf' which will print 6 decimal places with trailing zeros (123.450000), '%0.6lg' which will print up to 6 decimal places (123.45), and '%lg' which will print the number of decimal places required to fully show the number.

The default is null (""). When null `time_precision` and `time_rep` will set the preferred format for time stamps. For example, setting `time_precision PS` and `time_rep NS` will make times display as "nn.ppp ns" where n is the nanoseconds, and p is the picoseconds.

6.2.24 `time_divisor`

`time_divisor number`

When reading Ascii traces, all times are divided by this number. Generally the `time_precision` is what is wanted instead, as `time_precision` doesn't change the internal times but instead the display. `Time-divisor` changes the value that is read in from the file, and is most useful for Ascii traces which do not specify any time units in the file itself.

6.2.25 `time_multiplier`

`time_multiplier number`

When reading Tempest traces, the multiplier will be used to convert the phase counter to the time. Defaults to 2.

6.2.26 `file_format`

`file_format DECSIM | TEMPEST | VERILOG | VPD`

Changes the file format to be read. Same as the buttons in the `Trace Open` menu.

6.2.27 `cursor`

`cursor ON | OFF`

Enables or disables the cursors. Same as the toggle buttons in `Customize Change`.

6.2.28 `cursor_add`

`cursor_add time color [-USER] [comment]`

Adds a cursor of color number `color` (0 to 9) at the specified time. If specified, add `comment` that may be seen with `Value Examine`. The cursor will be dotted, and will disappear the next time a file is read in. The `-USER` flag makes the cursor like a manual cursor; it will be a solid line and will not disappear on rereads.

6.2.29 `click_to_edge`

`click_to_edge ON | OFF`

Enables or disables whether adding a cursor or grid close to a signal edge will place the cursor on that edge.

6.2.30 `debug`

`debug ON | OFF`

Enables or disables debugging mode.

6.2.31 `print`

`print number | ON | OFF`

Enables or disables debugging print mode at the specified informational level.

6.2.32 print_size

`print_size` A | B | EPSPORT | EPSLAND

Selects the paper size, A (8 1/2" by 11") or B (11" by 17"). EPSPORT and EPSLAND make an Encapsulated Postscript file without headers, in portrait or landscape layout, suitable for inclusion in other documents.

Same as the buttons in the Trace Print popup.

6.2.33 vector_separator

`vector_separator` "character"

Sets the character that separates the base signal name from vector numbers. The default is '[' , which means that the signal 'alpha[1]' consists of vector number 1 in the signal named alpha. Three other useful combinations are '_' for spider signals 'alpha_1', '<' for Decsim signals 'alpha<1>', and null ("") for no separator 'alpha1'. If the separator is null ("") then '<' will still be accepted. Tempest traces are assumed to use the '<' separator, and setting it otherwise will split up all of the bits into individual entries.

6.2.34 start_geometry

`start_geometry` widthxheight+xoffset+yoffset

Specifies the geometry to be used when the first window is opened by Dinotrace. Specified in standard X-Windows format, for example `start_geometry 1200x900+20+95` specifies a window at position 20, 95 of width 1200 and height 900.

6.2.35 open_geometry

`open_geometry` width[%]xheight[%]+xoffset[%]+yoff[%]

Specifies the geometry to be used when a new trace window is created with Trace Open menu. The format may be the same as `start_geometry`, in which case the exact dimensions specified will be used for the new window. Percentage may be used to specify a position and size relative to the old window. For example the default is '`open_geometry 100%x50%+0%+50%`' which means that the new window will have the same width and half the height of the old window. The position will be the same in the x direction and 50% more (not 50% of the value) in the y direction.

6.2.36 shrink_geometry

`shrink_geometry` width%xheight%xoffset%+yoff%

Specifies the geometry to be used when a old trace window is being shrunk with Trace Open menu option. Absolute positions should not be used, instead percentages should be used to specify a position and size relative to the original window size. For example the default is '`shrink_geometry 100%x50%+0%+0%`' which means that when the new opened window is created the old window will have the same width, X, and Y position, but half the height of the old window.

6.2.37 value_highlight

`value_highlight` value color [signal_pattern] -COLOR -CURSOR

Searches for the value in signals matching the pattern. If the -COLOR flag is set, changes the color of the value to color (0 to 9). If the -CURSOR flag is set, adds a cursor of the specified

color. The *signal_pattern* is the name of the signal inside the trace, with * being a wild card that matches any number of characters.

6.2.38 **signal_highlight**

`signal_highlight signal_pattern color`

Changes the color of any signals matching *signal_pattern* to the given color number *color* (0 to 9). The *signal_pattern* is the name of the signal inside the trace, with * being a wild card that matches any number of characters.

6.2.39 **signal_add**

`signal_add signal_pattern [after_signal] [color]`

Adds signals matching *signal_pattern* from the trace, if the signal was deleted. If the signal hasn't been deleted, nothing happens. If the *after_signal* is provided, the signal is added after the first signal that matches the *after_signal* pattern. If *after_signal* is not provided, the new signal is added to the end of the trace.

6.2.40 **signal_delete**

`signal_delete signal_pattern`

Deletes signals matching *signal_pattern* from the trace.

6.2.41 **signal_delete_constant**

`signal_delete_constant signal_pattern [-IGNOREXZ]`

Deletes signals matching the pattern, which have a constant value, from the trace. If *-IGNOREXZ* is specified, don't consider transitions to/from unknown or tristate as making a signal non-constant. Useful for reducing a large number of signals down to only those which change.

6.2.42 **signal_copy**

`signal_copy signal_pattern [after_signal] [color]`

Copies signals matching *signal_pattern* from the trace to a new position. If the *after_signal* is provided, the signal is added after the first signal that matches the *after_signal* pattern. If *after_signal* is not provided or is not found, the new signal is added to the end of the trace.

6.2.43 **signal_move**

`signal_move signal_pattern [after_signal] [color]`

Move signals matching *signal_pattern* to a new position. If the *after_signal* is provided, the signal is moved after the first signal that matches the *after_signal* pattern. If *after_signal* is not provided, the new signal is added to the end of the trace.

6.2.44 **signal_note**

`signal_note signal_pattern note`

Add the specified note to signals matching *signal_pattern*. The signal note may be viewed by pressing with the Examine button over the signal name.

6.2.45 `signal_rename`

```
signal_rename signal_pattern new_signal_name [color]
```

Changes the **first** signal matching *signal_pattern* to have the name *new_signal_name*. This can be used to make an alias copy of a signal by copying a signal, then renaming it.

6.2.46 `signal_radix`

```
signal_radix signal_pattern radix
```

Changes signals matching *signal_pattern* to have the specified radix. Note that the **real** radix is detected for Verilog automatically, and changing other non-real signals to a real format probably won't display the values you wish.

6.2.47 `signal_waveform`

```
signal_waveform signal_pattern ANALOG | ANALOG_SIGNED | DIGITAL
```

Changes signals matching *signal_pattern* to have the specified waveform representation on the display.

6.2.48 `signal_states`

```
signal_states signal_pattern = {State, State...};
```

Allows the value of a bus to be translated to a text string. The *signal_pattern* is the name of the signal inside the trace, with `*` being a wild card that matches any number of characters. Each state is the text to be displayed, using the characters `0-9A-Z%._/`. States are assigned in numerical order starting with zero, unless a `##=` proceeds a state name: If a state name is of the form `##=name`, for example in `7=Read` the `Read` name is assigned to the bus value seven. (The number is always in decimal.) The next assignment will be to 8 if no number is specified.

Example:

```
signal_states *cpucreq[2:0] = {
    1=Barrier, Fetch, FetchM, Read, Write, LDx_L, STx_C};
```

In this example, any signal ending in `'cpucreq[2:0]'` (any case) will convert 1 to `Barrier`, 2 to `Fetch`, etc. Unspecified values, such as 0 and 8 will be displayed as a number. If the state does not fit on the screen, the number will be printed instead. Text strings may only be defined for state numbers 0-255.

7 X Resources

This chapter describes the X-Windows resources that Dinotrace supports.

Dinotrace uses the X resource database for color and font selection.

To use a resource, put in your `~/.Xdefaults` file

```
Dinotrace*Foreground: Black
Dinotrace*Background: #caca9999
Dinotrace*color1: DarkOliveGreen
```

Then type at the shell `'xrdb ~/.Xdefaults'` to reread the database.

Dinotrace adds the following resources over and above the core X Toolkit resource names and classes (such as foreground and background):

color#

color#, where **#** is 1-9, specifies the colors for searches, highlighting, and cursors. **color0** is fixed to the foreground color.

barcolor

barcolor specifies the color to be used for the background of alternate signals. Defaults to 7% greener than the background color. Those that remember green bar tractor-feed paper should feel comfortable with this default.

signalfont

signalfont specifies the font to be used for signal names along the sides of the trace window. Defaults to `--Fixed-Medium-R-Normal--*-120-*-*-*--*-1`

timefont

timefont specifies the font to be used for time values along the top of the trace window. Defaults to `--Courier-Medium-R-Normal--*-120-*-*-*--*-1`

valuefont

valuefont specifies the font to be used for values inside buses. Defaults to `--Fixed-Medium-R-Normal--*-100-*-*-*--*-1`

8 Remote Control

Dinotrace can be triggered to reread traces by sending a `USR1` signal to it. This allows build environments to automatically refresh dinotrace when a new waves file is available. See the `kill` manpage for details on sending a signal, usually it's `kill -USR1 %?dino`.

After annotation, Emacs creates a socket that can be used to send commands into it. See the Emacs chapter and internal code for details.

9 Dinotrace and Emacs

This chapter describes the interaction between Dinotrace and Emacs.

Dinotrace provides the means for back-annotating highlighting and values into Emacs buffers, and from Emacs, highlighting signals, values, and the like. This is what separates Dinotrace from the rest.

9.1 Installation

The appropriate Emacs code should have been installed along with the rest of Dinotrace. If you encounter problems, you should return to this section.

You'll probably also need `verilog-mode.el` from <https://www.veripool.org/verilog-mode>. Make sure that file, and the distribution's files `lisp/dinotrace.el` and `lisp/sim-mode.el` are installed in your Emacs `site-lisp` directory. (In Emacs `C-h v load-path RET` should list the name of that directory.) Check that in your `.emacs` file, or better yet, the site's `site-start.el` file there are the lines:

```
(autoload 'sim-log-mode "sim-log" "Mode for Simulation Log files." t)
(setq auto-mode-alist (append (list '("\\.log$" . sim-log-mode)) auto-mode-alist))
(autoload 'dinotrace-update "dinotrace" "Update dinotrace annotations in this buffer" t)
(autoload 'dinotrace-mode "dinotrace" "Toggle dinotrace annotations in this buffer" t)
(global-set-key "\C-x\C-aa" 'dinotrace-update)
(global-set-key "\C-x\C-ad" 'dinotrace-mode)
```

9.2 Annotation

With a trace up in Dinotrace, select the `Value Annotate...` menu item. This will create a magic file in your home directory.

In Emacs, visit a Verilog or C file which has some signals that are in the trace. Press `C-x C-a a`. Emacs will read the annotation file and add highlighting to your buffer. A new menu will appear on the screen, and the mode line will indicate `'Dinotrace'`.

In a simulation log file (ending in `.log`), Pressing `C-x C-a a` will likewise add annotations of where cursors exist.

9.3 Signal Matching

Emacs chooses what to highlight based upon only the base part of the signal name; all hierarchy and bus descriptors are stripped. If multiple signals exist with the same basename, all signals other than those with the desired hierarchy should be deleted so they won't participate in the annotation.

9.4 Entering Dinotrace-Mode

There are two keys used to enter Dinotrace Mode:

`C-x C-a a`

`M-x dinotrace-update`

`C-x C-a a` will enter `dinotrace-mode` in this buffer. If there is a new annotation file, read it, update all annotations in all visible widows, and remove outdated annotations from invisible buffers.

C-x C-a d

M-x dinotrace-mode

C-x C-a d will toggle Dinotrace Mode in this buffer. With arg, turn Dinotrace mode on if and only if arg is positive.

It will not read a new annotation file if one has already been read in.

9.5 Exiting Dinotrace Mode

There are two ways to exit Dinotrace Mode:

C-x C-a d

M-x dinotrace-mode

C-x C-a d will toggle Dinotrace Mode in this buffer. With zero or negative arg, turn off Dinotrace mode.

C-x C-q

M-x dinotrace-toggle-read-only

C-x C-q will toggle read-only status. Dinotrace traps this key and will remove annotations and make the buffer writable.

9.6 Configuring Dinotrace-Mode

These keys configure Dinotrace Mode:

C-x C-a a

M-x dinotrace-hierarchy-set

C-x C-a a will specify the hierarchy as a regular expression. Any signals must match this regular expression in order to be annotated. This is useful if the same signal name appears in many modules with different values; you can thus choose which modules the signals will come from.

9.7 Keys in Dinotrace Mode

When inside a Dinotrace Mode buffer, many keys are defined to affect both Emacs and also Dinotrace. All begin with *C-x C-a*, and are also selectable from the Emacs menubar. Most match the keyboard accelerator keys for Dinotrace itself.

C-x C-a c

M-x dinotrace-goto-and-cursor-time

Take the time that the point is over, and scroll Dinotrace so time is visible. Add a cursor using the current color.

C-x C-a C

M-x dinotrace-goto-and-cursor-time-next-color

Take the time that the point is over, and scroll Dinotrace so time is visible. Add a cursor using the next color.

C-x C-a s

M-x dinotrace-goto-and-highlight-signal

C-x C-a s will take the signal name the point is over, and scroll Dinotrace so the signal is visible. The signal is highlighted using the current color.

C-x C-a S

M-x dinotrace-goto-and-highlight-signal-next-color

C-x C-a S will take the signal name the point is over, and scroll Dinotrace so the signal is visible. The signal is highlighted using the next color.

C-x C-a v

M-x dinotrace-highlight-value

C-x C-a v will take the value the point is over, and highlight it using the current color. With a prefix-arg, cursors will also be put where the value occurs.

C-x C-a V

M-x dinotrace-highlight-value-next-color

C-x C-a S will take the value the point is over, and highlight it using the next color. With a prefix-arg, cursors will also be put where the value occurs.

C-x C-a >

M-x dinotrace-cursor-step-forward

C-x C-a > will move all cursors forward one grid #0 period and reannotate the buffer. If the grid is set correctly, this will change the values to represent one clock period later.

C-x C-a <

M-x dinotrace-cursor-step-backward

C-x C-a < will move all cursors back one grid #0 period and reannotate the buffer. If the grid is set correctly, this will change the values to represent one clock period earlier.

10 File Formats

This chapter describes the file formats supported by Dinotrace, and how to produce them with various simulators.

If you're making a quick trace that is small, the ASCII format is probably quickest and dirtiest. If you're making a real 4-state simulator, Tempest XOR format is very compact and fast. A example reader for this format is included with the Dinotrace distribution.

10.1 Verilog Value Change Dumps

Dinotrace can read the Value Change Dump format, common with simulators such as Verilog-XL, VCS, and many others.

VCD files have a `.vcd` extension. Under the **Trace Read** menu, select Verilog Format. Select **Trace Read** to read the file. If this is your standard format, you can set it permanently by placing in the `$DINODISK:dinotrace.dino` file:

```
file_format verilog
vector_separator "["
```

VCD files can contain a great number of signals, potentially every net in the design. To cut down on the clutter, Dinotrace removes any signal that is identical to a signal higher up in the design hierarchy. For example, if the signal `'top.cpu.cache.index_addr[7:0]'` is the same net as `'top.cpu.addr[27:20]'`, then only the latter will be shown in Dinotrace. This can be disabled with `save_duplicates`.

To save disk space you may wish to store files as gzipped. This will result in about a 10 times space savings, and Dinotrace will uncompress them automatically. (By detecting the `.gz` extension and piping through `gunzip`. `Gunzip` must exist in your `$PATH` for this to work.)

10.1.1 Making VCDs in Verilog

To create a value change dump file inside of Verilog, add the code somewhere in a module:

```
// Create a dump file
initial begin
    $dumpfile ("|gzip --fast --stdout >trace_top.vcd.gz");
    $dumpvars (12, top);
end
```

Where `top` is the name of the module to be traced, `trace_top.vcd.gz` is the output file, and 12 is the number of levels of hierarchy underneath `top` to be traced. Dinotrace expects the `.vcd` extension. This command will dump all signals in the module `top`. For more information on dump files, see *Chapter 18 of the Verilog-XL reference manual*.

10.1.2 VCD File Format

For a description of the VCD file format, see *Chapter 18 of the Verilog-XL reference manual*.

10.2 Verilog Value Plus Dumps

Dinotrace can read the Value Plus Dump format, common with VCS.

VPD files have a `.vpd` extension. Under the **Trace Read** menu, select **Verilog VPD Format**. Select **Trace Read** to read the file. If this is your standard format, you can set it permanently by placing in the `$DINODISK:dinotrace.dino` file:

```
file_format vpd
vector_separator "["
```

For this to work, you must have the **vpd2vcd** program installed in your search path. It was distributed with VCS, and requires a key to operate; contact your simulation vendor for details.

Dinotrace will convert the format to a standard VCD file using a pipe. Thus all of the notes above for the Section 10.1 [Verilog Value Change Dumps], page 28, also apply to this format.

10.2.1 Making VPDs in Verilog

To create a value plus dump file inside of Verilog, add the code somewhere in a module:

```
// Create a vpd file
initial begin
    $vcdpluson (12, top);
end
```

Where `top` is the name of the module to be traced, and `12` is the number of levels of hierarchy underneath `top` to be traced. For this to work you may also need a `-I` or other compile time option for your simulator.

10.3 ASCII Traces

The initial format that Dinotrace supported was a Decsim ASCII format. This format is still very useful for reading traces created by **perl** scripts and other quick programs. For large traces, any of the other formats will make significantly smaller files.

10.3.1 Making Decsim ASCII Traces

Decsim provides both ASCII and binary formats. For a Decsim ASCII trace, type at the command prompt:

```
set trace/radix=2/vector/header/exact/logname=name
clock, signal1, signal2, signal3
```

For information, see Section 10.11.1 [Making Decsim Binary Traces], page 33.

10.3.2 ASCII Trace Format

The ASCII Trace format is broken into two pieces, the header, and the data lines.

The header consists of a leading `!`, followed by signal names that are read vertically, with each bit in a independent column. Some character separates the signal name from the bit number, this exact character varies depending on the simulator. The header is actually optional, Dinotrace will create some pretty boring names if the header is not present.

Here's a header, with the signal names SIG[2:0] and OUT. (Read it sideways.)

```
!   SSS
!   III
!   GGG0
!   ___U
!   210T
```

The data follows, with one time per line. The spaces leading the header above were added so that the first signal column lines up with the first value after the time stamp. Actually, the time stamps are optional, in which case Dinotrace will assign successive times, starting at 0 and incrementing by 100 for each data line.

```
0   xxxx
100 000x
103 0000
200 1100
203 0001
```

Dinotrace understands the values 0, 1, the tristate Z, and the unknowns ?, X, and U. Case does not matter.

10.4 Tempest Traces

Dinotrace supports Digital's tempest (BT02) format, as produced by CCLI, COSMOS and other tools. After creating a trace, start up Dinotrace. Under the **Trace Read** menu section, select *Tempest Format*. (To make this the default, read the instructions about configuration files.) Select **Trace Read** to read the file.

Tempest designs use underscores to indicate bus subscripts. To have Dinotrace collapse signals into vectors, create a configuration file with the lines:

```
vector_separator "<"
hierarchy_separator ">"
```

Some CPU groups use a timestamp that is a integer that increases every other cycle. The half cycle is a phase. Dinotrace can either count phases, which is the default, or cycles. Since Dinotrace times are integers, the trick is to change the time resolution so that fractional values can be displayed:

```
time_multiplier 500
time_precision ps
time_rep ns
```

To detect traces that start on phase B, the second signal in the trace, right after the cycle_count, must be "phase", which toggles zero/one every phase. If this signal is not present, it is assumed the trace starts on phase A. (If incorrect - it starts on phase B and there is no phase signal, the first phase in the entire trace will seem to be stretched one cycle.)

To save disk space you may wish to store files as gzipped. This will result in about a 50 times space savings, and Dinotrace will uncompress them automatically.

10.5 Making Verilator Traces

Dinotrace can read in Verilator traces created using CCLI. Dinotrace will convert the two signals that Verilator creates for tristate busses (foo and foo_en) into a single signal with tristates shown normally.

10.6 Making Cosmos Traces

Dinotrace can read in COSMOS traces created under Merlin. Dinotrace will convert the two signals that COSMOS creates for each signal (foo and foo_cos) into a single signal with unknowns shown normally.

10.7 Tempest Trace Format

The Tempest Trace Format is a relatively simple binary format. Files all have a `.bt` extension, and are stored by unix `write()`s of arrays of **little-endian** longwords. (Little is the endianness as Windows NT and Digital Unix, and opposite that of Suns.)

The first record is at the head of the file, and is the identification block.

`'char (4 bytes)'`

File signature plus revision (e.g. 'BT02').

`'uint (32 bits)'`

Number of bytes for the header.

`'uint (32 bits)'`

Number of signals.

`'uint (32 bits)'`

Number of simulation rows (Do not rely on this, it is not correct if not known when the header is written)

`'uint (32 bits)'`

Number of bits per simulation 'row', non-padded.

`'uint (32 bits)'`

Number of bits per simulation 'row', padded to mod 64.

Following that is a signal description block for every signal in the design. The number of such records was read from the above identification block.

`'uint (32 bits)'`

Flags for signal characteristics:

`'bit 0'` Signal is input.

`'bit 1'` Signal is output.

`'bit 2'` Signal is pseudo-pin.

`'bit 3'` Signal contains time-stamp value (not a real signal in design).

`'bit 4'` Signal is 3 or 4 state (not 2 state).

`'uint (32 bits)'`

Offset in body row (in bits).

```

'uint (32 bits)'
    Width of signal (in bits).

'uint (32 bits)'
    Number of chars in signal name. n below.

'char (n bytes)'
    Signal name, not null terminated.

'char padding (8-(n%(8 bytes))) if n%8 != 0'
    Padding characters to insure quadword alignment.

```

Finally is binary packed data. Each simulation row contains the value of all signals at that time stamp. The location of each bit was defined in the signal description blocks above. For example, storage of 300 signals requires 320 bits per row of simulation data. This data continues to EOF. (The number of simulation rows in the header may not be correct if it was unknown at the time the header was written.)

For three and four state signals, the four state bit in the flag section is set for that signal. Two bits are stored in the file per wire bit, where 00=0, 01=1, 10=tristate, and 11=unknown.

10.8 Tempest XOR Traces

Dinotrace and Verilator have colluded to make a derivative of the Tempest format, called XOR format (format string BX02). This format is detected by Dinotrace automatically when tempest format is selected.

Ungzipped files are only slightly smaller than as traditional Tempest files. Gzipping however yields a 90% shrink in file size. Since the code to generate the trace is compiled in, the actual simulation time to create the trace is also greatly reduced, speeding simulations by 30% or so.

10.9 Making Verilator XOR Traces

Verilator makes Tempest XOR traces using the fast trace `perl` script (see the *Verilator Manual*). A list of signals to be traced must be provided at compile time.

10.10 Tempest XOR Format

Tempest XOR format is very close to tempest normal format. The first difference is that the format characters are BX02 rather than BT02.

The second difference is that the binary data for each time record is a XOR of the previous row's binary data. (Starting with all zeros before the first record.) Thus if a single signal changes in a given time, the first longword will have a couple of bits set to indicate the bits that changed in the time stamp, and a single bit will be set in a later longword for the bit that changed.

This seemingly trivial change greatly improves compression, as 0's are now the majority of the file, and compress to near nothing.

Any Tempest XOR format readers and writers should only operate on gzipped files. Thus the normal file extension is `.bt.gz`.

10.11 Decsim Binary

The Digital internal Decsim simulator has a proprietary format supported by Dinotrace built under VMS only.

Decsim designs use <>'s to indicate bus subscripts. To have Dinotrace collapse signals into vectors, create a configuration file with the line: `vector_separator "<"`.

10.11.1 Making Decsim Binary Traces

Decsim uses the set trace command to create Dinotrace compatible traces. Usually a clock should be the first signal as Dinotrace will automatically align the grid to the first signal in a trace.

At the command prompt, type:

```
set trace/radix=2/vector/binary/exact/logname=name
clock, signal1, signal2, signal3
```

Our group creates a trace file for each sub chip, and turns that trace into a macro in a separate file. This way the same trace may be called from many tests. For example:

TRACE_subchip.IND:

```
macro/command TRACE_subchip [$text] =
set trace/radix=2/vector/binary/exact/logname=$text
clock,
signal1,
signal2
$endmacro
```

A test would then call:

```
@TRACE_subchip.IND
TRACE_subchip testname_subchip
```

The test will create a file testname_subchip.TRA. We find this naming convention convenient as it will group all of the traces from one chip together. For example, running the test ERRORS with the trace files TRACE_SSM.IND and TRACE_CSR.IND creates the trace files ERRORS_SSM.TRA and ERRORS_CSR.TRA.

11 Bugs

This chapter describes bug reporting and the known bugs in the current version.

Please report bugs to wsnyder@wsnyder.org

The following bugs are known to exist:

- When using **lessTIF**, the lines indicating highlighted signals and cursors that appear in the scroll bars may not always appear. To make them reappear, simply press the refresh widget.
- The Ultrix version of Dinotrace requires **Motif V1.1.3**. It will run under older versions, however toolkit warnings will be printed which may be ignored and Trace Open will make the window size too large.
- Decsim does not correctly create binary traces of arrays. For example `signal[1:0]<1:0>` will not display the correct data in Dinotrace because Decsim incorrectly creates the file. Decsim will work correctly if the trace is changed to ASCII.
- There is a problem with the **X-11** Toolkit that causes strange dots to appear in the trace window. This only occurs on some systems. Please contact me if this occurs on your systems.

Index

This is an alphabetical index of this manual.

#

(configuration comment) 15

—

-decsim 3
 -geometry 3
 -noconfig 3
 -res 3
 -tempest 3
 -verilog 3
 -vpd 3

.

.gz 3
 .Z 3

A

a 11
 Adding, Cursors 8, 19
 Adding, Signals 9, 21
 Aligning, Grids 16
 annotate 17
 Annotate... 11
 Annotation 11, 17
 ASCII Trace Format 29
 ASCII Traces 29

B

barcolor 23
 begin 5
 Begin Button 5
 BT02 31
 Bugs 34
 BX02 32

C

c 8
 C 8
 C-f 11
 C-o 6
 C-r 6
 C-x C-a < 27
 C-x C-a > 27
 C-x C-a a 25, 26
 C-x C-a c 26
 C-x C-a C 26
 C-x C-a d 26
 C-x C-a s 26
 C-x C-a S 27
 C-x C-a v 27
 C-x C-a V 27
 C-x C-q 26
 Clear Highlight 11
 Clearing, Cursors 9
 Clearing, Traces 7
 Click to Edge 8, 19
 click_to_edge 19
 Closing, Traces 7
 color# 23
 Color, Bars 23
 Color, Grids 17
 Color, Searching 23
 Comments 15
 Configuration Files, Disabling 3
 Contributors 2
 Copying, Dinotrace 1
 Copying, Signals 10, 21
 Cosmos, Making Traces 31
 cursor 19
 Cursor Add 8
 Cursor Cancel 9
 Cursor Clear 9
 Cursor Delete 9
 Cursor Highlight 9
 Cursor Move 9
 Cursor Note 9
 Cursor Step Backward 9
 Cursor Step Forward 9
 Cursor, Add 19
 Cursor, Enables 19
 cursor_add 19
 Customize Change... 7
 Customize Grid... 8
 Customize Read... 8
 Customize ReRead 8
 Customize Restore 8
 Customize Save As... 8

D

`debug`..... 19
 Debugging, Mode..... 19
 Debugging, Print..... 19
 Decrement Resolution Button..... 6
 Decsim, Arrays Bug..... 34
 Decsim, Making Binary Traces..... 33
 Decsim, Making Traces..... 29
 Deleting, Cursors..... 9
 Deleting, Signals..... 10, 11, 21
 DINOCONFIG..... 15
 DINODISK..... 15
 Divisor, Time..... 19
`down`..... 5
`draw_prefix`..... 17
 Duplicate Signals, Saving..... 17

E

Edge, Clicking to..... 8, 19
 Emacs, Annotation..... 11, 25
 Enables, Saving..... 17
`end`..... 5
 End Button..... 5
 Environment Variables, DINOCONFIG..... 15
 Environment Variables, DINODISK..... 15
`esc`..... 9
 Examining, Values..... 12
`exceed`..... 17
 eXcursion..... 17
 Exiting..... 7

F

File Format, Switch..... 3
`file_format`..... 19
 Finding, Signals..... 10
 Font, Time..... 23
 Font, Value..... 23
 Format, File..... 19
 Format, Time..... 18
 Full..... 6

G

`g`..... 5
 Geometry, Open..... 20
 Geometry, Shrink..... 20
 Geometry, Start..... 20
 Geometry, Switch..... 3
 Goto Button..... 5
 Goto, Signal..... 18
 Goto, Time..... 18
`grid`..... 16
 Grid, Customization..... 8
 Grid..... 8
`grid_align`..... 16
`grid_color`..... 17

`grid_resolution`..... 16
`grid_signal`..... 16
`grid_type`..... 16
 Grids, Stepping Cursors by..... 9
`gunzip`..... 3

H

Height, Signals..... 8, 16
 Highlighting, Cursors..... 9
 Highlighting, Signals..... 9, 11, 21
 Highlighting, Values..... 12, 20
 History..... 2

I

Increment Resolution Button..... 6
 Introduction..... 1
 Invoking Dinotrace..... 3

K

Keep Highlight..... 11
 Keyboard Shortcuts..... 14

L

`left`..... 4
`lessTIF`..... 34

M

Manual Refreshing..... 8, 17
 Motif..... 34
 Moving, Cursors..... 9
 Moving, Cursors by Grid..... 9
 Moving, Signals..... 10, 21
 Multiplier, Time..... 19

N

Name Slider..... 5
 Notes, Cursors..... 9
 Notes, Signals..... 10, 21

O

`open_geometry`..... 20
 Opening, Traces..... 7

P

Page Increment.....	7, 16
Page Size, Print.....	7, 20
<code>page_inc</code>	16
<code>pagedown</code>	5
<code>pageup</code>	5
Precision, Time.....	18
Prefix, Drawing.....	8, 17
<code>print</code>	19
Print, Page Size.....	7, 20
Print.....	7
<code>print_size</code>	20
Printing.....	7

R

Radix, Signals.....	10, 22
Radix, Values.....	12
Reading, Customization.....	8
Reading, Traces.....	6
<code>refresh</code>	17
Refresh Button.....	6
<code>refreshing</code>	17
Refreshing, Doing.....	17
Refreshing, Mode.....	8, 17
Renaming, Signals.....	22
Representation, Time.....	7, 18
Rereading, Traces.....	6
<code>resolution</code>	17
Resolution.....	17
Resolution Button.....	6
Resolution, Decrementing.....	6
Resolution, Grids.....	16
Resolution, Incrementing.....	6
Restoring, Customization.....	8
<code>right</code>	4
Rise-fall time.....	8, 16
<code>rise_fall_time</code>	16

S

<code>s</code>	9
<code>S-down</code>	5
<code>S-left</code>	4
<code>S-right</code>	4
<code>S-up</code>	5
<code>save_duplicates</code>	17
<code>save_enables</code>	17
<code>save_ordering</code>	18
<code>S</code>	9
Saving, Customization.....	8
Searching, Signals.....	10
Searching, Values.....	12
Selecting, Signal.....	11
Separator, Vector.....	20
<code>shrink_geometry</code>	20
Signal Add.....	9
Signal Cancel.....	11

Signal Copy.....	10
Signal Delete.....	10
Signal Highlight.....	9
Signal Move.....	10
Signal Names, Scrolling.....	5
Signal Note.....	10
Signal Radix.....	10
Signal Search.....	10
Signal Select.....	11
Signal Slider.....	5
Signal Waveform.....	10
Signal, Grids.....	16
Signal, Prefix, Drawing.....	8, 17
Signal, Rise-fall time.....	8, 16
<code>signal_add</code>	21
<code>signal_copy</code>	21
<code>signal_delete</code>	21
<code>signal_delete_constant</code>	21
<code>signal_goto</code>	18
<code>signal_height</code>	16
<code>signal_highlight</code>	21
<code>signal_move</code>	21
<code>signal_note</code>	21
<code>signal_radix</code>	22
<code>signal_rename</code>	22
<code>signal_states</code>	22
<code>signal_waveform</code>	22
<code>signalfont</code>	23
Signals, Deleting.....	11
Signals, Duplicates, Saving.....	17
Signals, Height.....	8
Signals, Highlighting.....	11
Signals, Scrolling.....	5
Signals, SIGUSR.....	24
<code>start_geometry</code>	20
States, Signals.....	22

T

Tempest Trace Format.....	31
Tempest Traces.....	19
Tempest XOR Format.....	32
Time Slider.....	4
Time, Full Window.....	6
Time, Goto.....	5
Time, Moving to Beginning.....	5
Time, Moving to End.....	5
Time, Representation.....	7, 18
Time, Zooming.....	6
<code>time_divisor</code>	19
<code>time_format</code>	18
<code>time_goto</code>	18
<code>time_multiplier</code>	19
<code>time_precision</code>	18
<code>time_rep</code>	18
<code>timefont</code>	23
Trace Clear.....	7
Trace Close.....	7

- | | | | |
|-------------------------------|----|-------------------------------------|--------|
| Trace Exit | 7 | Value Highlight | 12 |
| Trace Open..... | 7 | Value Search..... | 12 |
| Trace Print | 7 | <code>value_highlight</code> | 20 |
| Trace Print..... | 7 | <code>valuefont</code> | 23 |
| Trace Read..... | 6 | Values, indicating Radix | 12 |
| Trace ReRead | 6 | VCS, Making Traces | 28, 29 |
| Trace ReRead All..... | 6 | <code>vector_separator</code> | 20 |
| Tristates, Saving | 17 | Verilator | 2 |
| | | Verilator, Making Traces | 31, 32 |
| | | Verilog, Duplicate Signals | 17 |
| | | <code>verilog-mode.el</code> | 25 |
| | | Verilog-XL, Making Traces | 28, 29 |
| | | | |
| U | | W | |
| <code>uncompress</code> | 3 | Waveform, Signals | 10, 22 |
| <code>up</code> | 5 | | |
| | | Z | |
| V | | Zoom Button | 6 |
| <code>v</code> | 12 | | |
| <code>V</code> | 12 | | |
| Value Annotate | 11 | | |
| Value Annotate..... | 11 | | |
| Value Cancel | 13 | | |
| Value Examine..... | 12 | | |

Table of Contents

1	Summary	1
1.1	Copying	1
2	History	2
2.1	Contributors	2
3	Invoking Dinotrace	3
4	Screen Description	4
4.1	Window Widgets	4
4.1.1	Time Slider	4
4.1.2	Name Slider	5
4.1.3	Signal Slider	5
4.1.4	Buttons	5
4.2	Trace Menu	6
4.2.1	Trace Read.....	6
4.2.2	Trace ReRead	6
4.2.3	Trace ReRead All.....	6
4.2.4	Trace Open.....	7
4.2.5	Trace Close.....	7
4.2.6	Trace Clear.....	7
4.2.7	Trace Print.....	7
4.2.8	Trace Print	7
4.2.9	Trace Exit.....	7
4.3	Customize Menu.....	7
4.3.1	Customize Change.....	7
4.3.2	Customize Grid.....	8
4.3.3	Customize Read... ..	8
4.3.4	Customize ReRead.....	8
4.3.5	Customize Save As.....	8
4.3.6	Customize Restore	8
4.4	Cursor Menu	8
4.4.1	Cursor Add.....	8
4.4.2	Cursor Highlight.....	9
4.4.3	Cursor Move.....	9
4.4.4	Cursor Delete.....	9
4.4.5	Cursor Note	9
4.4.6	Cursor Step Forward.....	9
4.4.7	Cursor Step Backward	9
4.4.8	Cursor Clear.....	9
4.4.9	Cursor Cancel	9
4.5	Signal Menu	9

4.5.1	Signal Add...	9
4.5.2	Signal Highlight	9
4.5.3	Signal Move	10
4.5.4	Signal Copy	10
4.5.5	Signal Delete	10
4.5.6	Signal Note	10
4.5.7	Signal Radix	10
4.5.8	Signal Waveform	10
4.5.9	Signal Search...	10
4.5.10	Signal Select...	11
4.5.11	Clear Highlight	11
4.5.12	Keep Highlight	11
4.5.13	Signal Cancel	11
4.6	Value Menu	11
4.6.1	Value Annotate...	11
4.6.2	Value Annotate	11
4.6.3	Value Highlight	12
4.6.4	Value Examine	12
4.6.5	Value Search...	12
4.6.6	Value Cancel	13
5	Keyboard Shortcuts	14
6	Customization Files	15
6.1	Automatically Read Files	15
6.2	Configuration Commands	15
6.2.1	#	15
6.2.2	page_inc	16
6.2.3	rise_fall_time	16
6.2.4	signal_height	16
6.2.5	grid	16
6.2.6	grid_resolution	16
6.2.7	grid_align	16
6.2.8	grid_signal	16
6.2.9	grid_type	16
6.2.10	grid_color	17
6.2.11	refreshing	17
6.2.12	refresh	17
6.2.13	draw_prefix	17
6.2.14	annotate	17
6.2.15	resolution	17
6.2.16	save_enables	17
6.2.17	save_duplicates	17
6.2.18	save_ordering	18
6.2.19	signal_goto	18
6.2.20	time_goto	18
6.2.21	time_rep	18

6.2.22	time_precision	18
6.2.23	time_format	18
6.2.24	time_divisor	19
6.2.25	time_multiplier	19
6.2.26	file_format	19
6.2.27	cursor	19
6.2.28	cursor_add	19
6.2.29	click_to_edge	19
6.2.30	debug	19
6.2.31	print	19
6.2.32	print_size	20
6.2.33	vector_separator	20
6.2.34	start_geometry	20
6.2.35	open_geometry	20
6.2.36	shrink_geometry	20
6.2.37	value_highlight	20
6.2.38	signal_highlight	21
6.2.39	signal_add	21
6.2.40	signal_delete	21
6.2.41	signal_delete_constant	21
6.2.42	signal_copy	21
6.2.43	signal_move	21
6.2.44	signal_note	21
6.2.45	signal_rename	22
6.2.46	signal_radix	22
6.2.47	signal_waveform	22
6.2.48	signal_states	22
7	X Resources	23
8	Remote Control	24
9	Dinotrace and Emacs	25
9.1	Installation	25
9.2	Annotation	25
9.3	Signal Matching	25
9.4	Entering Dinotrace-Mode	25
9.5	Exiting Dinotrace Mode	26
9.6	Configuring Dinotrace-Mode	26
9.7	Keys in Dinotrace Mode	26

10	File Formats	28
10.1	Verilog Value Change Dumps	28
10.1.1	Making VCDs in Verilog	28
10.1.2	VCD File Format	28
10.2	Verilog Value Plus Dumps	29
10.2.1	Making VPDs in Verilog	29
10.3	ASCII Traces	29
10.3.1	Making Decsim ASCII Traces	29
10.3.2	ASCII Trace Format	29
10.4	Tempest Traces	30
10.5	Making Verilator Traces	31
10.6	Making Cosmos Traces	31
10.7	Tempest Trace Format	31
10.8	Tempest XOR Traces	32
10.9	Making Verilator XOR Traces	32
10.10	Tempest XOR Format	32
10.11	Decsim Binary	33
10.11.1	Making Decsim Binary Traces	33
11	Bugs	34
	Index	35