University Of Cambridge Computer Laboratory

Cambridge CSIM Verilog Simulator

User Manual.

License Warning

1 CSIM Introduction

CSIM is a Verilog logic simulator. It takes circuit designs in structural forms and functional forms. It is used to verify circuits are functionally correct before they are placed into chips and for verification of back-annotated chips. CSIM can be used to simulate small parts of circuits, complete functional units such as FPGAs or complete board designs.

All designs files contain Verilog. Most of the regular Verilog constructs are supported by CSIM, but many exotic features of Verilog are not supported. Not supported constructs currently include: *charge strengths, power supplies, built-in gates, upward references, event-counts, functions, disable, user defined primitives, global events and event types, real numbers, named blocks, procedural continuous assignments, specify.*

The current version of CSIM is different in its semantics from standard Verilog systems in that the uncertain logic state 'x' is not represented on a bit-wise basis inside expression evaluation. Instead, if any bit is uncertain, the whole expression is so.

2 Unix Environment Variables and Starting CSIM

The CSIM executable must be on your PATH. Secondly, the environment variable CVPATH must be set as the search path for Verilog files. Thirdly, you need an 'app-defaults' sub-directory pointed to by the XAPPLRESDIR variable. This directory must contain the file 'csimdefs', suitably modified, if desired, for your own preferences.

On Thor, using the bash shell, the following will serve

```
PATH=$PATH:/group/clteach/djg11/bin:/group/clteach/ckh11/arm/bin
CVPATH=.:/group/clteach/djg11/techlibs:/group/clteach/djg11/teachlibs
XAPPLRESDIR=$HOME/.app-defaults
export CVPATH PATH XAPPLRESDIR
```

provided you create the directory pointed to be XAPPLRESDIR and copy in 'csimdefs' from djg's 'app-defaults' directory.

To run the simulator, simply type the command

```
$ cs fn1 fn2 fn3 ...  &
```

where the optional '&' runs the simulator as a background process. You must add all suffixes to file names. These will typically be `.vnl`, `.v` or `.cv`.

For example, to run CSIM with the source file 'trivial.cv' type the command '`cs trivial.cv &`'. If CSIM's main window appears on the screen you have proved that you are able to run CSIM. For now, simply click on the 'Quit' button, followed by 'Yes' to exit.

The advantage of running the simulator as a background process is that you are always free to return to the parent window and do other things during the course of your simulator session without having to exit and then restart the simulator and re-enter your current simulation parameters. One reason for wanting to do this is to update your simulator source files to fix any errors found during simulation. If any source files are changed, CSIM can automatically detect this and reload any affected files before re-simulation using its 'Reload' command.

3 CSIM Windows

CSIM uses three windows, the main window, the navigate window, and the trace window. If you are running with a window manager that uses titlebars, the bars will show 'main', 'csimnav', and 'csimtrace', respectively. The main window contains a toolbar and a text display window where simulator messages and output from Verilog meta-statements such as `$display` are displayed.

The navigate window is used to explore the net list and to select which signal nets to view in the trace window.

The trace window shows the timing waveforms of traced signal nets.

When the simulator is initially invoked only the main window is displayed. To display the navigate window click on the 'Navigate' button in the main window. To display the trace window click on the 'Sim' button in the main window. To close the navigate window click on the 'Exit' button in the window (if it exists!), or alternatively 'Quit' to close all displayed CSIM windows and exit from the simulator.

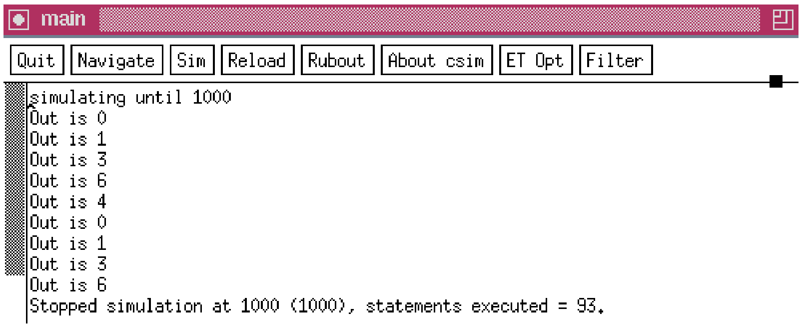

Figure 1: Main window display

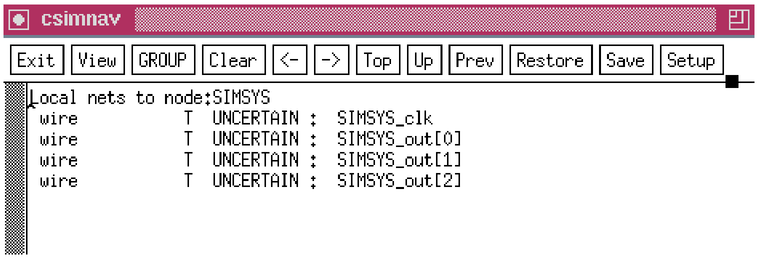

Figure 2: Navigate window display

In most circumstances you would normally want to display all three CSIM windows at once by simply clicking on the 'Navigate' and 'Sim' buttons in the main window immediately after invoking CSIM, and likewise at the end your CSIM session close all three windows and exit from the simulator by clicking on the 'Quit' button.

Here is an example file to demonstrate the simulator. We assume that the following is the contents of `cstest.cv` and that a library file called `teachlib.cv` is available containing a 10MHz clock.

```
// cstest.cv
     module SIMSYS();
       wire clk;
       wire [2:0] out;
       CSTEST cstest(clk, out);
       CLK10MHz clock(clk);
     endmodule


     // Here is Johnson counter.
     module CSTEST(clk, out);
       input clk;
       output [2:0] out;
       reg [2:0] out;

       initial out = 0;
       always @(posedge clk)
         begin
         $display("Out is %d", out);
         out <= { out[1], out[0], ~(out[2] | out[1]) };
         end
     endmodule
```

Now run the simulator with the command

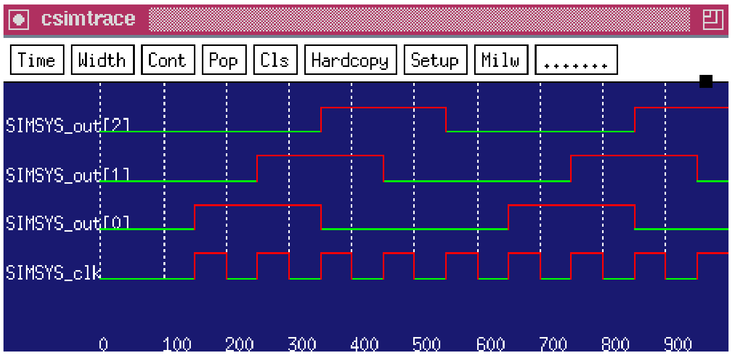

Figure 3: Trace window display

```
$ cs cstest.cv teachlib.cv
```

To perform the example simulation simply select the 'Time' command in the trace window and 'OK' the time shown. At this point the simulation waveforms should appear in the trace window.

Figures 1, 2 and 3 show the status of the three windows after a simulation has been performed.

Once your circuit design has been simulated, the resulting waveforms should be carefully examined and analysed to determine if it is functioning as expected.

The waveform representation in the trace window of the four signal states Low ('L'), High ('H'), High-impedance ('z'), and Uncertain ('x') varies according to the type of monitor being used, i.e. monochrome or colour. When displayed in colour, the waveform signal states are primarily differentiated by colour, but when displayed in monochrome they are primarily differentiated by line style. On all displays the signal levels are positionally represented: 'L' is drawn lower than the other signal states, 'H' is drawn higher than the other states, and 'z' and 'u' are drawn midway between the 'L' and 'H' states. The trace window shows vectored registers and integers as a pair of lines (unless they are at their all-ones or all-zeros state) and prints the value of these variables each time they change.

We have now covered the basics of using CSIM. The following sections cover the other main simulator functions and how to use them.

4 Simulation Process

CSIM commences the simulation when the 'Time' command in the trace window is given (or invoked via mouse zoom and pan commands). The simulation time is measured in nano-seconds. The simulation starts at time zero and runs until the time given in the 'Time' command is reached or else the simulation is stopped with an interrupt (see below) or a `$finish` command is encountered. It is only when simulation is in process that output messages will be yielded to the main window. When panning or zooming in the trace window (described later), it is only if the requested new end time is greater than the current end of simulation that further simulation is performed. The values of nets given in the navigate window is for the current end of simulation time.

When running a long simulation, you may interrupt the simulation whilst it is in progress by pressing 'i' from within either the navigate window or the trace window. This is needed, for instance, if you accidentally zoom-out too far.

In a large simulation, the number of events will exceed the number which CSIM stores in its internal event history buffer, and storage for early events is reclaimed. This means that it is impossible to pan back beyond a certain time in the past. To see what happened then, the simulation must be restarted from the beginning with a hard reset. Hard reset can be achieved using 'Reload'.

5 Main Window Functions

'Navigate' opens the navigate window for exploration and signal tracing control.

'Sim' opens the graphical trace window, in which the simulation waveforms for any traced signals are displayed.

'Reload' is used to reload any Verilog source files which have been edited since the start of the CSIM session or the last reload. This is used to ensure that CSIM is up-to-date with any changes made. Syntax errors introduced and encountered during a reload will be displayed in the window where CSIM was started, and another reload is needed after their correction.

'Rubout' simply clears the message window area.

'About csim' prints a copyright message.

'ET Opt' is a toggle to enable detailed event tracing for debugging behavioural Verilog. When ET is enabled, each assignment to a wire or register results in a print statement to the console output.

'Filter' is used to define output message filters. These suppress the output of predictable and unwanted error or warning messages generated during simulation. Filters are defined using the Filter 'add' option, and take the form of UNIX 'ed'-like regular expressions. They are typically used to suppress setup and hold-time violation messages, which can be quite numerous (and not very significant) at times; e.g. '.*DFF dplltest.dpll.s0: D input violated .*time' would suppress any setup/hold-time violations on the specified flip-flop's D-input. The Filter 'clear' option deletes any currently defined filters.

Filters get saved (by default) in signal tracing ('.sig') files when using the navigate window function 'Save', and are subsequently restored (by default) when using the 'Restore' function.

6 Navigate Window Functions

In the navigate window there is a concept of a current module, which can be either a net or a node. Initially this module is set to the top node. There are also four modes of display, which are toggled using the 'View' command or the 'v' key. The modes are:

- Nets: nets local to the current module
- Nodes: local nodes (modules) below the current module
- Contacts: port contacts to the current module
- Traces: list of traced nets (not a function of the current module)

The functions of the mouse keys in the navigate window are:

- Left-button: makes the line under the mouse the current (selected) line
- Middle-button: selects and traces a net (if the current line is a net)
- Right-button: selects and untraces the net on the current line

Multiple nets may be traced or untraced by dragging across a number of lines with the mouse.

Any line in the listing can be selected by moving the text cursor to it, either with the left mouse button or the arrow keys on the keyboard. This line can then be focussed on by pressing the 's' key. If the line is a node, focussing on it results in it becoming the current module. If the line is a net, focussing on it causes its contacts to be displayed. (Nets can also be focussed on using the mouse in the trace window, see below.) If the line represents a signal then this signal can be traced using the middle mouse button and untraced with the right mouse button.

When a signal is focussed on (from either the navigate or trace window), the node of which it is an output is selected as the current node. This makes following the structure easier.[1]

The 'Prev' button goes to the previously displayed state.

'Save', 'Restore', and 'Clear' operate on all signals, regardless of the current node. 'Save' and 'Restore' both display a list of the available signal tracing files. The 'New' option in the Save and Restore menus is for specifying a new filename, i.e. one which does not appear in the list because it is a new file or it does not have the conventional '`.sig`' filename extension. Save saves the current list of traced signals to a file. The simulation Time and Width, and any output-message filters, are also recorded in the saved file. When a file is restored, all the signals in the file are added to the traced list. Several signal files may be Restored, in which case all the different signals are accumulated in the traced list. Clear simply untraces all signals in the traced list.

By toggling 'Traces On/Off' in the Restore menu, and 'Times On/Off' and 'Filter On/Off' in the Save and Restore menus, you can control what information is saved to the file, and the subsequent behaviour of restores, to prevent or allow loading of the traced list, modification of the simulation Time and Width, and loading of any filter expressions defined in the file. By default all Save and Restore functions are enabled.

'Up', Leftarrow, and Rightarrow change the current node to the current node's parent and sibling nodes. The order of the sibling nodes is dependent on the Verilog read in.

The 'GROUP' functions are used to group-together/ungroup signals with similar names, e.g. bus signals. These functions apply to the currently selected net, and all nets to be included in a group must 1. already be traced, 2. be subscripted, and 3. have the same textual name (i.e. differ in subscript only). When grouped the signals are traced, only one waveform entry appears in the trace window's list of traced signals, i.e. that of the group. All the waveforms for the individual member signals of the group are written on top of one another, giving a composite trace showing different signal states at the same time. The value of each signal-group is displayed for the current position of the currently selected cursor on the right-hand side of the trace window's command bar.[2]

The main reason for grouping signals together is to preserve valuable space in the waveform area. When you need to trace many signals, the waveforms become very cramped if you try to display too many discrete signals at once. Furthermore, it is normally clearer to display a related group of signals data 'as one' and be able to view its value numerically, rather than trying to work out what's happening from a list of individual waveforms. Note that integer and vectored Verilog variables are always shown a bit like grouped nets since they do not have constituent nets. When a set of signals is grouped together, a 'G' appears to the right of each of the grouped signals in the navigate window's signal list.

There are four grouping functions: 'Group', 'Group-Downwards', 'Ungroup', and 'Ungroup-Downwards'. The 'Group' and 'Ungroup' functions add and delete the current signal from the group. The 'Downwards' versions apply the command to all nets of the same name below the selected net in the listing.

[1] However, this is not implemented currently.
[2] Currently not implemented.

Through the trace window		
	left	move current cursor
	middle	give time delay between two adjacent cursors
	right	move nearest cursor
shift	left	drag time scale
	middle	zoom in
	right	zoom out
control	left	select current trace for print window
	middle	
	right	
Through the navigate window		
	v key	move to next mode
	s key	focus in on selected node
	left	select current line
	middle	trace current net
	right	untrace a traced net

Table 1: CSIM mouse and keyboard handy reference chart

The signal grouping attributes of traced signals are saved to/restored from signal-trace files by the 'Save' and 'Restore' functions described above.

While over the listing area of the navigate window the mouse may be used to perform the following functions. These are also shown in summary form table 1.

While in the navigate window, pressing the 'c' key redraws the screen. This can be useful as the screen is not always updated by all functions as often as it might be!

The 'v', 'u', and 'p' keys are the same as the 'View', 'Up', and 'Prev' buttons respectively. Also, in node mode, 'l' and 'r' are the same as the left '¡-' and right '-¿'.

7 Trace Window Functions

The trace window is controlled by the endtime and the width. A stack of these pairs is maintained and the old pair pushed on each time either of them is changed.

'Time' sets the endtime for the simulation (in nano-seconds) and initiates the simulation.

'Width' sets the period of time to be displayed (in nano-seconds).

'Cont' continues the simulation from the current endtime for the current Width. This is a forward pan.

'Pop' pops the last change of Time and Width off a stack, and redisplays the screen for those values.

'Cls' clears the stack of Time and Width values.

'Hardcopy' dumps the currently displayed trace window contents to a file, which may then be printed out. The data is dumped in postscript.

'Milw' is the interface to the cursor display functions.[3]

'.......' is not a function at all, but simply a place-holder for a subwindow used by a number of the trace window functions.

During simulation it is possible to invoke other commands and scroll other windows. However, it

[3]It is named after Dave Milway, the original author.

might be dangerous to do things like tracing or untracing signals during simulation: the simulator tends to crash. It is best to save your traced signal list at regular intervals.

To focus on a net from the trace window, or change the currently focussed net, point to the net name or its waveform in the trace window and press Ctrl + Left-button. When a net is selected it is displayed in the navigate window. A net selected in this way has a caret inserted before its net name on the trace display and is shown in a highlighted colour. Several cursor functions depend on the selected net.

8 Cursor Functions in the Trace Window

Cursors are vertical lines in the trace window and can be used to obtain information about the traced signals. They are labelled by a number at the bottom of the window. Initially there are two of them, Cursor 0 and Cursor 1, located out of sight off the right side of the trace window. The currently selected cursor (initially Cursor 1) is marked with ‘$>$’ and ‘$<$’ symbols. Cursors may be created, selected, moved, and deleted. Many of the cursor functions depend on the selected net.

The mouse may be used to perform various cursor functions:

- Left-button: moves the currently selected cursor (the position is printed on the right side of command bar).
- Middle-button: prints the difference between the cursors (note: only meaningful if you actually click between two cursors!).
- Right-button: selects and moves the cursor nearest the mouse pointer.

The cursors (‘Milw’) menu in the trace window is the interface to the cursor functions:

Measure difference: calculates the time difference between the minimum and maximum cursor positions.

Count changes: counts the number of positive (rising) edges between the minimum and maximum cursor positions for the currently selected net.

Move to nearest edge: moves the selected cursor to the nearest edge of the currently selected net and prints its position.

Find all cursors: puts all the currently active cursors on the screen, regularly spaced out across the trace display.

Wait for change: waits for a change on the current net after the current cursor. The simulation is run until the change, so this is not visible on the screen until a pan is performed.

Create cursor: creates a new cursor (up to a maximum of nine).

Remove cursor: removes the highest numbered cursor.

Bind all cursors: binds cursors together such that all cursors move and are fixed relative to one another.

Unbind all cursors: unbinds cursors such that they can all be moved independently of one another again.

Cursor n selects cursor n as current cursor from list of cursors.

Note that some of the above functions, namely ‘Count changes’, ‘Move to nearest edge’, and ‘Wait for change’, work on the currently selected net. If no net is currently selected these functions have no effect.

9 Display Control using the Mouse in the Trace Window

In addition to the Time, Width and Cont commands, the trace window width and endtime settings can be changed with the mouse. While over the trace area the mouse can be used to zoom in and out, and pan the waveform display, changing the simulation times, as follows:

Shift + Left-button (press, move, release): pan display: sets the time to the current time +/- the difference in time between press and release; the width is unaffected (except that the time origin, time 0, cannot be brought to the right of the left margin).

Shift + Middle-button (press, move, release): zoom-in: sets the width to the difference in time between press and release, and the time to the point of release.

Shift + Right-button (press, move, release): zoom-out: (see detailed explanation below).

Note: The Shift key must be held down for the full duration of any mouse-button press, move, release (i.e. drag) sequence—if the Shift key is released prematurely it will not result in the desired action.

Zooming-out is conceptually a bit trickier than zooming-in or panning, but the following scheme has the effect of zooming-out in a reasonably predictable manner. The new width is set to width * (width / abs(press - release)), i.e. the smaller the difference between press and release the greater the zooming-out, and the new time is set to time + (time - press) / (width / abs(press - release)), i.e. not very easy to explain! By far the easiest way to understand how zooming-out works is by trying it, as it is a simple and highly-intuitive concept.

10 Signal Trace Files

Another (optional, but nearly always used) file type input by the simulator is the trace-signal file, which has the default filename extension '`.sig`'.

The trace-signal file defines a set of signals to be traced, the total simulation time, and the displayed time width, i.e. the period of simulation time on view in the simulator's trace window at any one time.

Trace-signal files may either be created by the user prior to simulation, or created from a previous run of the simulator using the 'Save' button in the navigate window.

Normally, it is easiest to select the set of signals to be traced from within the simulator's navigate window (see section 6 above).

Once the signals to be traced have been selected in the navigate window, the trace window's displayed time scale and the total simulation period are defined using the 'Width' and 'Time' buttons respectively inside the trace window. Setting of the 'Time' period triggers simulator execution, and provided sensible signals and parameters are chosen the resulting simulation waveforms should appear in the trace window. Signals may then be untraced, other signals traced, different 'Width' and 'Time' parameters specified, and the simulation process repeated. When the user is happy with the signals and parameters chosen these selections may be saved to a trace-signal file for subsequent use, using the 'Save' button in the navigate window. In fact, it is a good idea to save frequently, since CSIM is absolutely full of bugs.

Trace-signal files should always be named with the filename extension '.sig', otherwise they will not subsequently appear in the list of available trace-signal files displayed when the 'Restore' button is selected (see below). A number of such trace-signal files may be set up to select different sets of signals and display their associated simulation waveforms.

To reuse a previously 'Save'd, or manually created, trace-signal file, select the 'Restore' button in the navigate window. A list of files with the default trace-signal file extension '.sig' will be displayed. Chose one of the displayed list of files, or enter the name of the required file if it is not named '.sig' or is not located in the current directory.

11 Other CSIM Command-line Options

CSIM has several command-line options, as follows:

The `-listmods` command line option causes CSIM to simply list the names of all modules supplied in the source Verilog files and then exit.

The `-f argfile` command line option causes the file given to be read as though it contained further command line options. This is handy when supplying a lot of files to CSIM.

The `-root rootname` command line option sets the top level simulation module name. The default is 'SIMSYS'.

The `-nox` command line option runs the simaultor without X windows. The only output will come from Verilog's metacommands such as '`$display`'.

The `-I<path>` command line option overrides the CVPATH in the environment with the given command line search path.

The `-ncd` command line option skips port rule checks and so allows simulation of misconnected modules.

The `-verbose` command line option adds detail to error messages and so on.

The `-events n` command line option sets the event count limit before which events from the past are reclaimed. (Not implemented.)