

Application Manual
TimingTool Version 3
September 2008

TimingTool version 3

Document revision 1.10

Copyright © 1999 – 2008 MOHC Ltd. All rights reserved.

This document contains trade secrets and other proprietary information, which belong to MOHC Ltd. or others. Information in this document is subject to change without notice and does not represent a commitment on the part of MOHC Ltd. The software described in this document is furnished under a license agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of those agreements. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use or by any method conveyed or disclosed to a third party or parties without the written permission of MOHC Ltd.

Trademarks

Java™ is a registered trademarks of the SUN Microsystems Corporation. Windows™ and Word™ are registered trademarks of the Microsoft Corporation.

Contact Information

Support Email support@timingtool.com
Sales Enquiries sales@timingtool.com

1 Contents

<u>1</u>	<u>Contents</u>	<u>3</u>
<u>2</u>	<u>Introduction</u>	<u>11</u>
2.1.1	Timing Diagrams for Documentation	11
2.1.2	Timing Analysis	11
2.1.3	Automated HDL Testbench Generation	11
2.1.4	Timing Diagrams on the Web	11
2.1.5	Si2 TDML Standard	11
2.2	About this Manual	12
2.3	Assumptions	12
<u>3</u>	<u>Installation</u>	<u>13</u>
3.1	Downloading	13
3.2	Starting TimingTool	14
3.3	Licensing	14
<u>4</u>	<u>Timing Diagram Editor</u>	<u>15</u>
4.1	General overview	15
4.2	General Editing	18
4.2.1	Selecting	18
4.3	Edges	19
4.3.1	New Edge	19
4.3.2	Selecting edges	19
4.3.3	Moving an Edge	19
4.3.4	Changing an edge state	20
4.3.5	Edge pop-up menu	20
4.3.6	The edge attribute panel	21
4.3.7	Snap to Grid	24
4.4	Waveforms	25
4.4.1	New Waveform	25
4.4.2	Selecting a Waveform	25
4.4.3	Moving a Waveform	25
4.4.4	Waveform Pop-up menu	25
4.4.5	Waveform Attributes Panels	26
4.4.5.1	Signal Attributes	26
4.4.5.2	Bus Attributes	27

4.4.5.3	Clock Attributes.....	28
4.4.6	Grid Attributes.....	29
4.4.7	Hiding Waveforms.....	30
4.4.8	Waveform Colour.....	30
4.5	Relationships.....	31
4.5.1	New Relationship.....	32
4.5.2	Moving a Relationship.....	33
4.5.3	Auto Position Relationship.....	33
4.5.4	Associating Parameters to Relationships.....	34
4.5.5	Hiding a relationship.....	34
4.5.6	Delay Relationships.....	35
4.5.7	Label Types.....	35
4.6	Text Annotations.....	36
4.7	Events.....	39
4.7.1	Time Breaks.....	39
4.7.2	Markers and the End Marker.....	40
4.8	Document Details.....	41
4.8.1	Base Time Units.....	41
4.9	Import.....	42
4.9.1	HDL.....	42
4.9.2	External TDML Source.....	42
4.10	Loading and Saving.....	42
4.11	Export.....	44
4.11.1	Graphic Bitmap Output.....	45
4.11.2	Graphic Vector Output.....	46
4.11.3	Document Output.....	46
4.12	HTML Output.....	47
4.12.1	View Tab.....	47
4.12.1.1	HTML Output Show Selections.....	48
4.12.1.2	Logo.....	48
4.12.1.3	Diagram.....	48
4.12.1.4	Parameter Tables.....	48
4.12.1.5	Colours.....	49
4.12.2	Settings Tab.....	49
4.13	Printing.....	50
4.13.1	Print Preview.....	50

4.13.1.1View Tab.....	50
4.13.1.1.1Print Preview Show Selections.....	50
4.13.1.1.2Logo.....	51
4.13.1.1.3Diagram.....	51
4.13.1.1.4Parameter Tables.....	51
4.13.1.1.5Colours.....	51
4.13.1.2Print Preview Settings Tab.....	51
4.13.2Print Dialog.....	52
4.14Keywords.....	53
4.14.1System.....	53
4.14.2Parameter Data.....	53
4.14.3Edge Data.....	54
4.14.4Signal Data.....	54
<i>5Parameter Tables.....</i>	<i>55</i>
5.1Creating and editing complete tables.....	56
5.2Selecting and moving around a parameter table.....	56
5.3Adding and Deleting a parameter.....	57
5.4Moving parameters.....	57
5.5Editing the parameters.....	57
5.6Parameter expressions.....	58
5.7What the parameters in the parameter table mean.....	60
5.8Customising the Table Layout.....	61
5.8.1Basic Layout.....	61
5.8.2Advanced Layout.....	61
5.8.3Saving your new parameter display format.....	62
5.9Exporting data from a parameter table.....	63
5.10Parameter table notes.....	63
<i>6Derived Waveforms.....</i>	<i>64</i>
6.1How To Define Signals.....	65
6.2Equation Syntax.....	66
6.2.1Waveform reference.....	66
6.2.2Logical operators.....	66
6.2.3Fixed States.....	66
6.2.4Parenthesis.....	66
<i>7Timing Analysis.....</i>	<i>67</i>
7.1Delays and Delay Chains.....	67

7.1.1	How delays are calculated.....	67
7.1.2	Delay Chains.....	68
7.1.3	Delay Loops.....	68
7.1.4	Delay Target Type.....	68
7.1.4.1	First.....	69
7.1.4.2	Last.....	69
7.1.4.3	Inside.....	70
7.1.4.4	Outside.....	70
7.1.5	Delay examples.....	70
7.1.5.1	Using delays to model an OR gate.....	71
7.1.5.2	Using delays to model an AND gate.....	71
7.2	Setup and Hold Constraints.....	72
7.2.1	Min / Max parameter values.....	72
7.2.2	Calculated Margins.....	72
7.3	Guarantees.....	73
7.4	Measure.....	73
8	HDL Testbench Generation.....	75
8.1	Import of HDL.....	75
8.2	Generation and Tolerance.....	76
8.3	Structure of output HDL.....	77
8.4	VHDL Specifics.....	78
8.5	Verilog Specifics.....	78
8.6	Saving HDL.....	79
9	Macros	80
9.1	Running Macros.....	80
9.2	Writing Macros.....	80
9.2.1	Introducing BeanShell.....	81
9.2.2	The Editing Interface.....	81
9.2.3	Macro Housekeeping.....	81
9.2.4	Macro File Format.....	81
9.2.5	The Mandatory First Example.....	82
9.2.6	Menu Icons.....	83
9.2.7	Examples.....	84
10	TimingTool Online.....	85
10.1	Personal Workspace.....	85
10.2	Export to HTML.....	86

10.3TimingTool Lite.....	86
11Preferences.....	87
11.1File Preferences.....	87
11.2Display Preferences.....	87
11.3Macros Preferences.....	89
11.4Proxy Preferences.....	90
Appendix A – Tutorials	91
A.1Drawing Interface Timings.....	91
A.1.1Describing a block or circuits timing.....	91
A.1.2Describing timings external to a block or circuit.....	91
A.2Modelling a Data Bus.....	92
Appendix B - HDL Reserved Words.....	93
B.1A list of VHDL Keywords is shown below:.....	93
B.2A List of Verilog Keywords is shown below:	93
Appendix C – TDML.....	95
Overview of the Timing Diagram Markup Language (TDML).....	95
Appendix D – The BeanShell Macro Language.....	96
Appendix E – TimingTool Terms and Conditions.....	97
Appendix F – Third Party Contributors.....	99
JEdit Syntax Highlighting Package.....	99
License.....	99
Copyright.....	99
Batik.....	100
Apache Batik Public License.....	100

Figures

<i>Figure 1 Installer First Dialogue</i>	13
<i>Figure 2 TimingTool desktop</i>	15
<i>Figure 3 Creating a new diagram</i>	16
<i>Figure 4 Inserting a new waveform</i>	17
<i>Figure 5 Timing Diagram Editor Window</i>	17
<i>Figure 6 Attribute Panel Divider</i>	18
<i>Figure 7 Selecting LHS Edge</i>	18
<i>Figure 8 Selecting RHS Edge</i>	18
<i>Figure 9 Edge menu</i>	20
<i>Figure 10 Edge attribute panel</i>	21
<i>Figure 11 Bus edge panel</i>	22
<i>Figure 12 Clock edge panel</i>	23
<i>Figure 13 Delay Rules</i>	23
<i>Figure 14 Snap to Grid</i>	24
<i>Figure 15 Waveform popup</i>	25
<i>Figure 16 Signal attributes</i>	26
<i>Figure 17 Bus Attributes</i>	27
<i>Figure 18 Clock attributes</i>	28
<i>Figure 19 Grid attributes</i>	29
<i>Figure 20 Grid Lines Example</i>	29
<i>Figure 21 Hide Check Box</i>	30
<i>Figure 22 Colour Button</i>	30
<i>Figure 23 Creating a new relationship</i>	32
<i>Figure 24 New relationship attribute panel</i>	32
<i>Figure 25 Dragging a relationship label</i>	33
<i>Figure 26 Auto Position Relationship</i>	33
<i>Figure 27 New relationship parameter</i>	34
<i>Figure 28 Edge Target Type</i>	35
<i>Figure 29 Relationship View Panel</i>	35
<i>Figure 30 Text annotation attribute panel</i>	36
<i>Figure 31 Text Annotation Drag Box</i>	37
<i>Figure 32 Diagram with Breaks</i>	39
<i>Figure 33 Break Attributes</i>	39

<i>Figure 34 Marker Attributes.....</i>	<i>40</i>
<i>Figure 35 Document Details.....</i>	<i>41</i>
<i>Figure 36 Open file dialog.....</i>	<i>43</i>
<i>Figure 37 Export Diagram.....</i>	<i>44</i>
<i>Figure 38 Export diagram settings.....</i>	<i>45</i>
<i>Figure 39 Export HTML.....</i>	<i>47</i>
<i>Figure 40 Parameter table exported as HTML.....</i>	<i>49</i>
<i>Figure 41 Print Preview.....</i>	<i>50</i>
<i>Figure 42 Print Preview Parameter Table.....</i>	<i>51</i>
<i>Figure 43 Print Preview Settings.....</i>	<i>52</i>
<i>Figure 44 Print Dialog.....</i>	<i>52</i>
<i>Figure 45 Tables Menu.....</i>	<i>55</i>
<i>Figure 46 General Data Parameter Table.....</i>	<i>55</i>
<i>Figure 47 Parameter table cell selection.....</i>	<i>56</i>
<i>Figure 48 Parameter table ascending sort.....</i>	<i>61</i>
<i>Figure 49 Parameter table format.....</i>	<i>62</i>
<i>Figure 50 Parameter Table Notes.....</i>	<i>63</i>
<i>Figure 51 Delay Example.....</i>	<i>68</i>
<i>Figure 52 Delay Chain.....</i>	<i>68</i>
<i>Figure 53 Delay Loop.....</i>	<i>68</i>
<i>Figure 54 Delay Target Type (First).....</i>	<i>69</i>
<i>Figure 55 Delay Target Type (Last).....</i>	<i>69</i>
<i>Figure 56 Delay Target Type (Inside).....</i>	<i>70</i>
<i>Figure 57 Delay Target Type (Outside).....</i>	<i>70</i>
<i>Figure 58 Delay OR gate example.....</i>	<i>71</i>
<i>Figure 59 Delay AND gate example.....</i>	<i>71</i>
<i>Figure 60 Setup and Hold example.....</i>	<i>73</i>
<i>Figure 61 Guarantee example timing diagram.....</i>	<i>73</i>
<i>Figure 62 Inside and Outside Measures.....</i>	<i>74</i>
<i>Figure 63 Select Entity / Module.....</i>	<i>75</i>
<i>Figure 64 New diagram dialog.....</i>	<i>75</i>
<i>Figure 65 HDL Generate Dialog.....</i>	<i>76</i>
<i>Figure 66 Tolerance Before.....</i>	<i>76</i>
<i>Figure 67 Tolerance After.....</i>	<i>77</i>
<i>Figure 68 HDL Structure.....</i>	<i>77</i>
<i>Figure 69 Save HDL popup dialog.....</i>	<i>79</i>

Figure 70 Online workarea.....85

2 Introduction

Welcome to the TimingTool application. TimingTool is designed to give electronics engineers an easy to use graphical interface for entering and editing timing diagrams commonly seen in digital electronic design.

2.1.1 Timing Diagrams for Documentation

Once a diagram has been entered, it can be output in a number of formats for documentation purposes. This allows easy importing into most of the common graphics and word processing packages (including Word, Star Office, Visio and Frame Maker). All the common bitmap outputs are available along with scaleable vector formats (svg, and emf). Further, pdf output is supplied for easy document generation.

2.1.2 Timing Analysis

The parameter driven timing analysis engine automatically calculates delay paths, setup and hold violations, measures and guarantees. This allows complex timing problems to be quickly identified and corrected.

2.1.3 Automated HDL Testbench Generation

One of the many useful features of TimingTool is the capability of translating a timing diagram into HDL (VHDL or Verilog). This aids an engineer's productivity considerable when producing test benches and other forms of timing related HDL code.

TimingTool's HDL generation has been used most successfully when describing bus cycles, inter IP block communication and interfaces to blocks of IP.

HDL (VHDL or Verilog) may be read into TimingTool, which then analyses the Entity/Module for signals and buses. The information that is extracted is then used to start the generation of a new diagram with the signals and buses filled in ready for the engineer to add timing waveforms.

2.1.4 Timing Diagrams on the Web

TimingTool is web enabled; datasheets can be exported directly to HTML for instant publication to the web/intranet. Timing diagrams can be uploaded to a common work area and can be accessed globally online via TimingTool-Lite the online timing diagram editor, enabling development teams to share information between sites and offices.

2.1.5 Si2 TDML Standard

TimingTool documents are completely TDML compliant. TDML is an XML based standard for recording timing diagram information, which has been developed by the Si2 organisation.

http://archives.si2.org/si2_publications/tdml/

Timing diagrams may be imported into TimingTool from other tools that export tdml, allowing migration of existing Timing diagrams into TimingTool.

2.2 About this Manual

This manual covers the use of the TimingTool graphical editor, the automatic generation of HDL, timing analysis calculations and the web enabling features.

2.3 Assumptions

This manual assumes that you are familiar with digital electronics and timing diagram related information. For HDL code generation it is assumed the reader will have a good knowledge of either VHDL or Verilog.

3 Installation

3.1 Downloading

An installer for TimingTool may be downloaded from the download page of our web site <http://www.timingtool.com>

If the platform you are using does not appear in the list please email support@timingtool.com

The latest release is always available for download. Once downloaded, always check that the downloaded file is the expected size. Never download TimingTool from any domain other than <http://www.timingtool.com>.

Once downloaded, the installer may be run; after a short delay whilst the installer unpacks, a dialogue will be displayed:

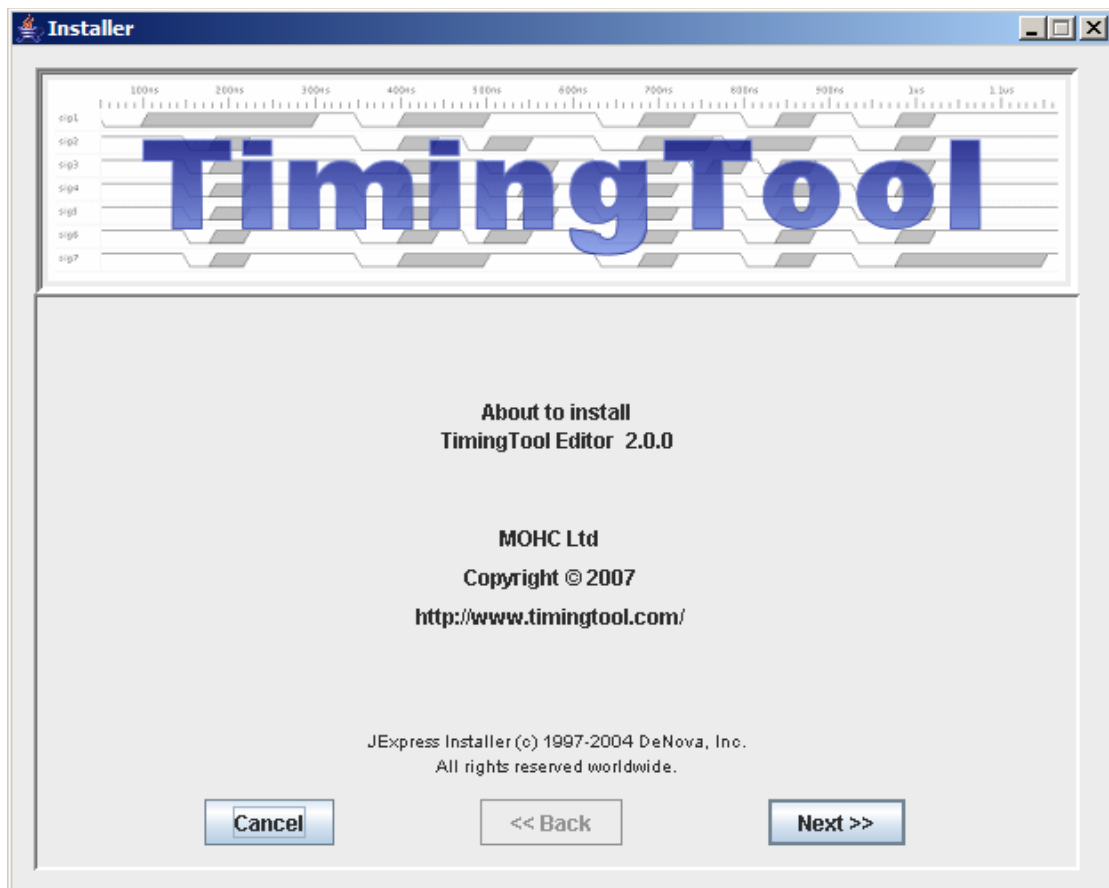


Figure 1 Installer First Dialogue

The version will be displayed - click on 'Next' to proceed.

Please ensure that there is plenty of disk space for the installation. TimingTool occupies approximately 70Mbytes once installed and requires a further 40Mbytes of temporary workspace during the installation process itself.

3.2 Starting TimingTool

To run TimingTool on windows select TimingTool from the start menu.

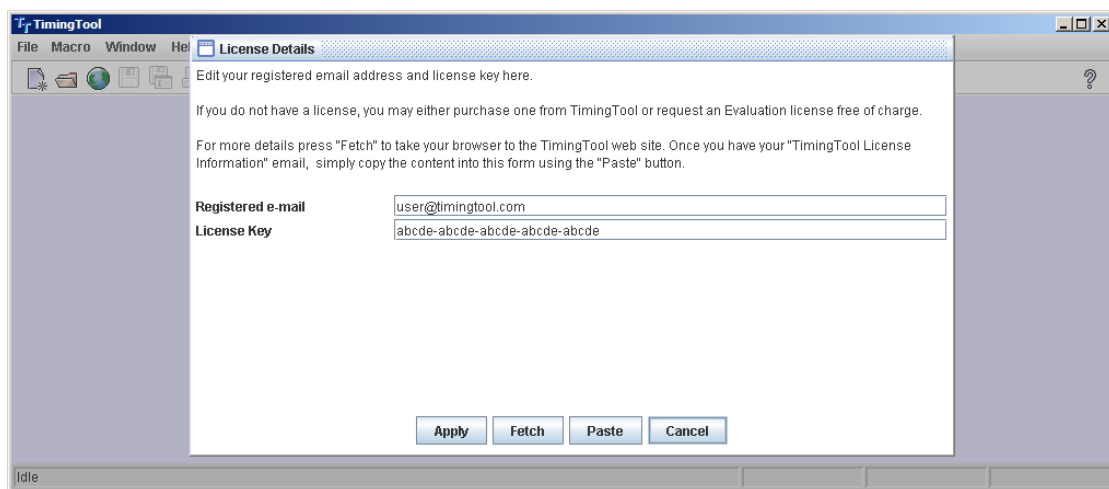
To run TimingTool from Linux, type:

```
> TimingTool
```

A symbolic link to the executable is created in your home directory.

3.3 Licensing

When TimingTool first starts a dialog is displayed requesting a license key, this should have been emailed to you at the time you downloaded the application.



The email address you used to register your account and the license key must both be entered into the dialog in order to unlock TimingTool. If you have an evaluation key, it is valid for 28 days after which the tool will cease to work. If you require a longer evaluation period please contact sales@timingtool.com.

To avoid typing the key into the field you may select all the text in the email from TimingTool and copy it (control-c). Simply click on the "Paste" button in this dialogue and the license information will be automatically filled out for you.

4 Timing Diagram Editor

4.1 General overview

Start TimingTool by running the TimingTool executable in the install directory either directly from the command line or from the shortcuts configured during the installation. For Windows the launch file is **TimingTool.exe** and in Linux the launch file is **timingtool**. When launching, TimingTool may optionally be passed one or more filenames to open. Only one instance of TimingTool application may run at a time on any one machine, subsequent invocations of the launcher will pass any arguments through to the running instance.

When TimingTool starts up, you will see the TimingTool desktop consisting of the main menu and icon bar at the top, the desktop in the middle and the status area at the bottom.

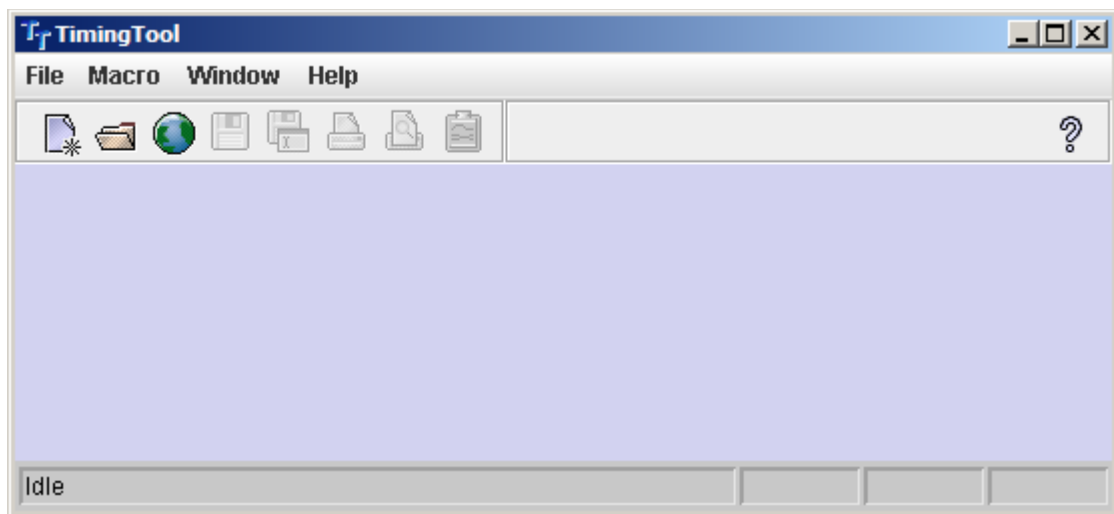


Figure 2 TimingTool desktop

Command Line Directives

The following command line directives are valid:

- new Opens a new, empty diagram prompting for a title.
- export Exports current file to image in default format. Note that file must be loaded before commanding an export; i.e. It is not possible to open a diagram and then export using the same invocation.
- format *EXT* Sets default format for image export. *EXT* may be one of the following:
"jpg", "xbm", "pict", "bmp", "psd", "pcx", "tga", "png", "gif", "svg", "emf", "wmf", "ps" or "pdf"
- close Closes current file. This will prompt to save if file edited.
- execute *MACROFILE* Executes the macro file specified.
- exit Exit timingtool. This will prompt to save edited files.

Starting a new Diagram

From the desktop, you can start a new timing diagram by selecting the 'new' icon, or new from the File menu. Alternatively, load an existing timing diagram using the 'File-open' menu/icon.



The timing diagram editor window will be displayed on the desktop, the 'Document Title' popup is shown, if a new diagram is started or an imported diagram with no title is loaded. Enter the diagram title and hit return. If you do not wish to enter a title, you may select cancel and the title will be set to "untitled"; the diagram will continue to open.

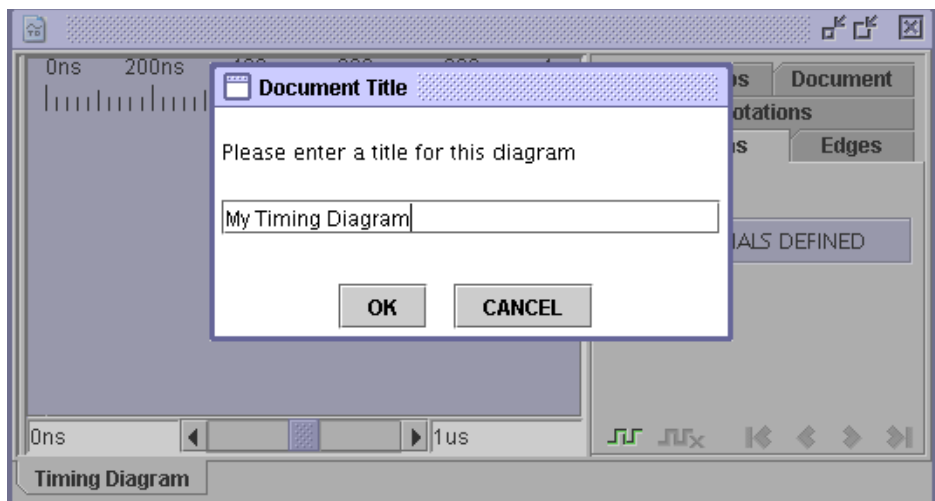


Figure 3 Creating a new diagram

Add some waveforms to the empty diagram by selecting the new waveform icon at the bottom left hand side of the waveform attribute panel, alternatively select one of the three waveform icons from the top left hand side of the diagram (signal, bus or clock), see Figure 4 Inserting a new waveform. These buttons will appear when the mouse is hovered over this area of the window.

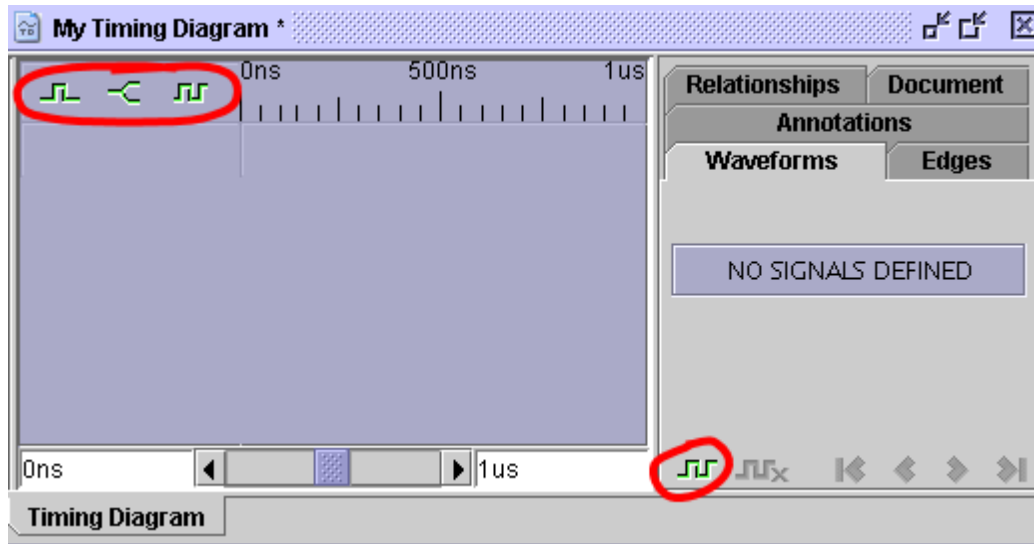


Figure 4 Inserting a new waveform

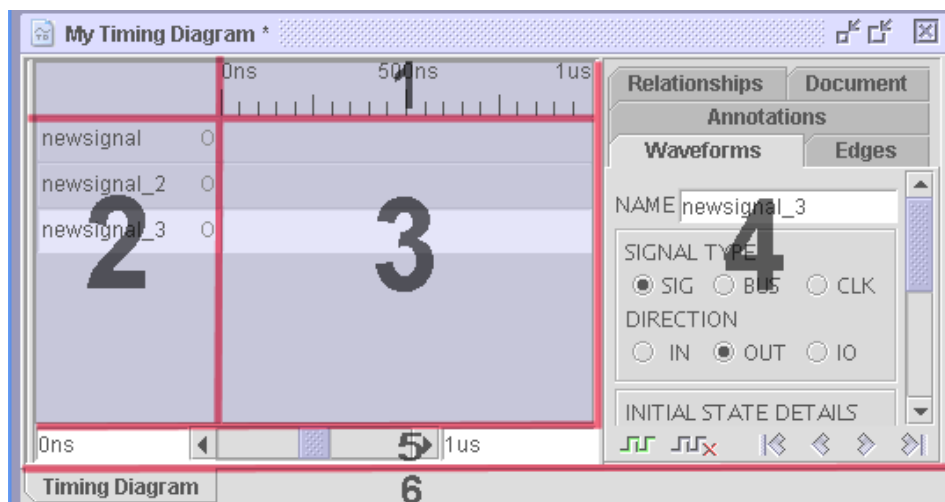


Figure 5 Timing Diagram Editor Window

The timing diagram editor window above is composed of 6 regions:

1. The time line ruler
2. The waveform label area
3. The waveform display area
4. The attribute information panels
5. The time range navigation.
6. The current view tabs. Eg, Timing Diagram, Parameter Tables, Tdml Source

The new waveform labels appear down the left hand side of the diagram (2). The waveform attribute panel on the right hand side of the diagram (4), shows attributes of the currently selected waveform, these are described in detail later. There are currently no edges defined, so the edge area of the diagram is empty (3).

The width of the attribute panel (4) may be changed by dragging the divider (Figure 6) between the waveform display (3) and the attribute panel. The attribute panel may be hidden temporarily; this facilitates viewing of the waveform using the full width of the editor window. Double-click on the divider to hide the panel. To restore, move the mouse over the divider which will now be on the far right of the editor window or click on an annotation or relationship on the waveform display.

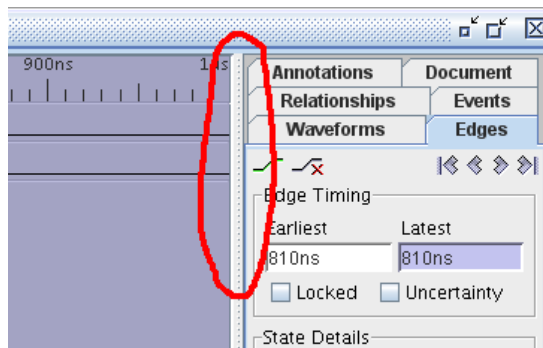


Figure 6 Attribute Panel Divider

4.2 General Editing

4.2.1 Selecting

As you move the mouse around the diagram selectable objects are highlighted.

To select an object click the left mouse button on the object select box. The edit mode will change depending on the type of object you have selected.

The five main object types are:

- Waveforms
- Edges
- Relationships
- Annotations
- Events

To select a waveform click on the waveform label to the left hand side of the diagram, (– zone 2), or click on the waveform attribute tab on the right hand side of the diagram.

To select an edge click on the edge select box that appears as you hover your mouse over the edge graphic, if the edges are close together zoom in to make selection easier and click to the side of the edge you wish to select. The edge select box will appear as the mouse pointer is moved towards it.



Figure 7 Selecting LHS Edge



Figure 8 Selecting RHS Edge

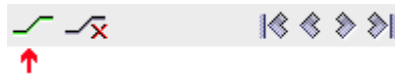
4.3 Edges

Edges represent a change in state of a signal in the timing diagram at a point in time.

4.3.1 New Edge

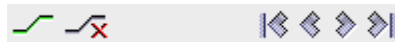
Edges can be inserted in a variety of ways, by:

- Double clicking on the location on the waveform where you wish to insert the new edge.
- Pressing down the left hand mouse button and dragging to insert a pulse.
- Using the pop-up menu: by holding down the right hand mouse button over the point where you wish to insert the edge and selecting from the pop-up 'insert edge' or 'insert pulse'.
- Use the new edge icon at the top or bottom of the edge attribute panel on the right hand side of the diagram.



4.3.2 Selecting edges

To select an edge click with the left hand mouse button on the edge select box. If the mouse is clicked on the state line rather than an edge the nearest edge to the point where the mouse was clicked is selected. Where edges are close together the navigation buttons at the top or bottom of the edge attribute panel can be used to select next and previous edges.



When an edge is selected, its attributes are shown in the right hand side attribute panel, where they can be edited.

Multiple edges may be selected using the mouse by holding down the control key whilst clicking on an edge. Multiple edges are selected to perform specific tasks such as multiple drag, multiple delete or repeating multiple edges.

4.3.3 Moving an Edge

To move an edge, click on the edge to be moved (a select box will appear around the edge). Then by holding down the left mouse button on the edge, the edge can be moved to the left or right as desired using the mouse. The edge cannot be moved outside the times of the surrounding edges. Note: Derived edges and clock edges are locked and cannot be moved. The left and right move arrows turn red if the edge cannot be moved any further. The display area will automatically scroll as the edge is dragged off the left or right hand side of the waveform window.

To accurately position an edge type the new time into the Time Earliest / Time Latest fields on the edge attribute panel.

Multiple edges may be dragged although only the primary selected edge will be used to determine how far all the edges may be dragged. Any secondary

selected edges may not move as far as the primary edge if they are obstructed by another edge.

4.3.4 Changing an edge state

Click the right mouse button over a selected edge or on the edge state line, the state value is changed and if you keep on clicking all the available states are cycled through.. The edge state can also be changed by editing the state option field in the edge attribute panel.

4.3.5 Edge pop-up menu

The pop-up menu allows the most common functions to be applied without the need to use an edges' attribute panel.

The "Insert Edge" places a new edge at the current cursor position.

"Insert Pulse" places two edges onto the waveform at the current cursor position. If for example the waveform is a signal and the state is 0 where the pulse is to be added then the first new edge forces the state to 1 and the second edge forces the state back to 0.

To change an edge's state at the current cursor position simply use the "Change Edge State" option and select the required new edge state.

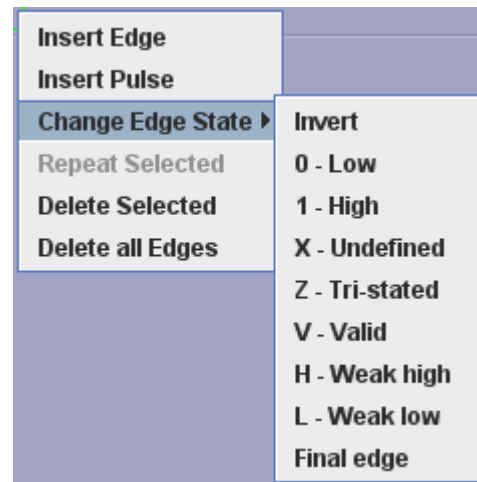


Figure 9 Edge menu

The "Repeat Selected" pops up a small dialogue to determine the repeat period and number of repeats for all selected edges.

The "Delete Selected" removes the edge closest to the current cursor position or any selected edges. The "Delete all Edges" removes all edges on the selected waveform.

4.3.6 The edge attribute panel

The edge attribute panel shows different information dependant on the type of edge you are editing.

The panel in Figure 10 shows the basic fields that are common to all edges.

Starting from the top:

The edge time earliest / latest

The uncertainty check box (the time latest field is only editable if this box is checked)

The locked check box, if an edge is locked its time cannot be altered by dragging the edge on the diagram, (derived and clock edges are always locked).

The edge state choice field, signal edges can have the following states:

- (i) 0 - low
- (ii) 1 – high
- (iii) X – undefined
- (iv) Z – tri state
- (v) V – valid
- (vi) H – weak high
- (vii) L – weak low

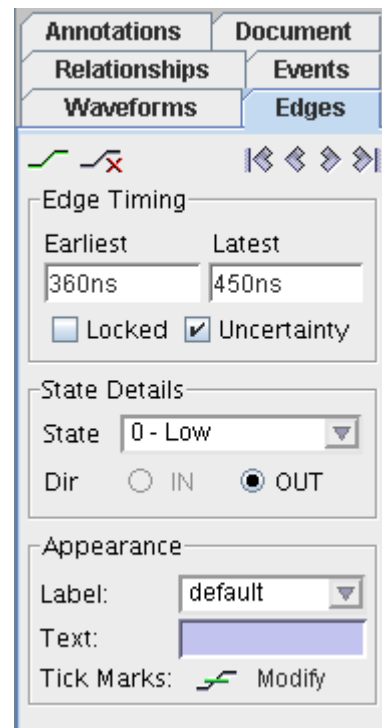


Figure 10 Edge attribute panel

Final Edge (the final edge has a time but no state).

The direction, IN or OUT, this is only editable if the signal is defined as bi-directional (INOOUT).

The label, this can have the following values:

- (i) Default, inherit from as the waveform label setting
- (ii) Text, custom text entered in the text field beneath the label type choice list.
- (iii) Time, the edge time.
- (iv) Uncertainty, the edge uncertainty.
- (v) Blank, nothing.
- (vi) Increment, edge count starting from 1, taking into account any skipped edges.
- (vii) Order, the edge order starting from 0 and not taking into account any skipped edges.
- (viii) Skip, skip an edge when calculating the edge increment.



The Tick Marks' modify button permits the edge's tick marks to be modified for the selected edge.

Bus edges can only be set to Valid / Invalid or Tri-state.

The bus edge attribute panel has an additional field into which the vector state can be entered.

This is entered in vhdl syntax:

Example:

16'h ffff for hex

8'b11011011 for binary

16'd65535 for decimal

The first number indicates the width of the bus in bits, the next character is the format of the number [h,d or b] followed by the number itself.

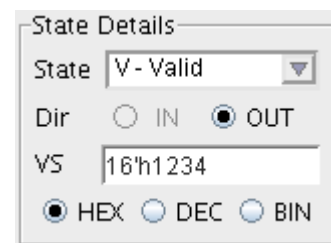


Figure 11 Bus edge panel

Clock edges have a more restricted set of attributes showing the edge time and label.

The label field can have the same setting described in above for a signal edge.

The clock edge time cannot be edited.

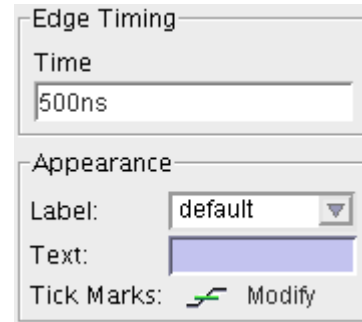


Figure 12 Clock edge panel

Derived edges have their time calculated by the timing analysis process. Derived edges are the targets of one or more delay relationships.

When an edge is derived from a delay additional information is shown in the edge attribute panel.

This panel shows which delay relationships drive the time earliest and time latest of the edge.

To view the associated delay relationship click on the D-TE or D-TL buttons.

The target type of the edge can be set in this panel. The target type controls how delay resolution is calculated when more than one delay relationship feeds into the same edge.

The target type attribute is described in detail in section 4.5.6

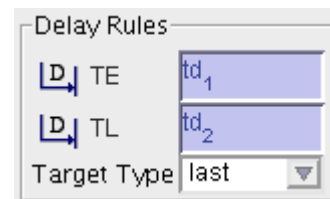


Figure 13 Delay Rules

4.3.7 Snap to Grid

To speed positioning of edges when dragging, by default TimingTool snaps the edge to the nearest time division on the ruler. It is possible to change this behaviour in the display preferences (see section 11.2).

It is sometimes required to manually position edges at fixed times relative to a clock or an edge in another signal.

To make this easier it is possible to set 'Snap to grid' on a signal or clock, see section 4.4.6 below.

All entered edges will then snap to the times of the grid; the grid is highlighted as you drag the edge to indicate the active snap times.

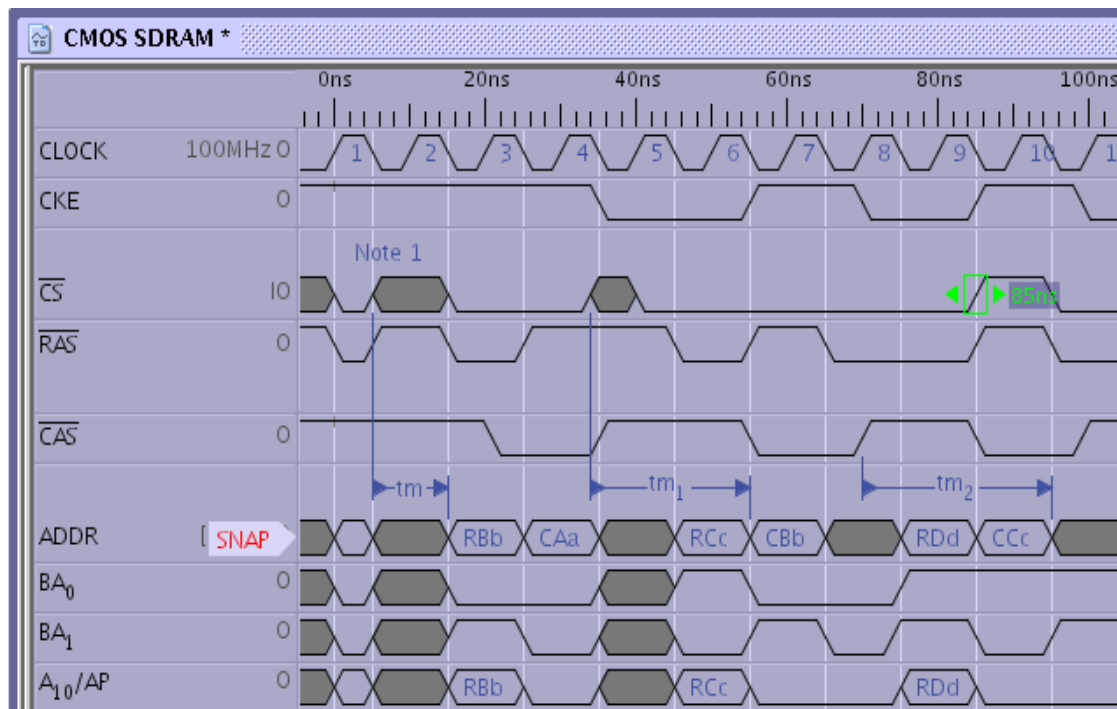


Figure 14 Snap to Grid

4.4 Waveforms

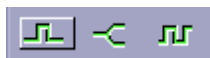
4.4.1 New Waveform

Waveforms can be inserted in a variety of ways, by:

1. Using the pop-up menu: by holding down the right hand mouse button over the selected signal a pop-menu appears select (insert: bus, signal or clock).
2. Selecting the new waveform icon at the top or bottom of the waveform attribute panel on the right hand side of the diagram.



3. By selecting one of the speed buttons at the top left hand corner of the editor window.



4.4.2 Selecting a Waveform

Left click on the waveform label area (left hand side), to select more than one waveform hold down the control key whilst left clicking on each waveform you wish to select.

4.4.3 Moving a Waveform

To move an entire waveform up or down with respect to the other waveforms in the diagram, firstly left click on the waveform label (LHS) that is going to be moved so that it is highlighted. Then click and hold the left mouse button on the waveform area of the selected waveform and simply drag the mouse up or down to where you wish the waveform to be placed. A purple line indicates where the waveform will be inserted when the left mouse button is released. All waveform information can be changed also by editing the fields available in the waveform attribute panel.

4.4.4 Waveform Pop-up menu

Holding down the right mouse button on either the waveform area or the edge area will generate a popup menu that is context sensitive for that area.

From the waveform area (LHS of diagram), the right mouse button produces the following options.

These options allow the most common functions to be applied without the need to edit a waveforms' attribute panel. The Insert options places the required signal type below the selected waveform. The delete selected signals deletes all the currently selected signals.

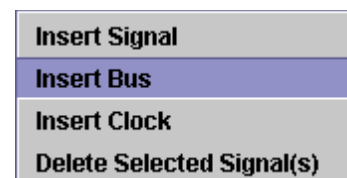


Figure 15 Waveform popup

4.4.5 Waveform Attributes Panels

4.4.5.1 Signal Attributes

A signal carries a single bit of data; its direction can be an input, an output or bi-directional.

Signal edges can have the following states:

- 0 - zero
- 1 - one
- Z – tri-state
- L - low
- H – high
- V - valid
- X – invalid

The initial state and direction of the signal defaults to a state of '0' with direction "output". This can be changed in the 'Initial State Details' section of the attribute panel. Signal edges when inserted will alternate between '1' and '0'.

The 'Default Edge Appearance' refers to the default type of label to display on each edge of the waveform, the edge slope angle and the style of tick marks.

Label types are:

- default – blank
- text – user entered text label
- time – the edge time
- uncertainty – the edge uncertainty, if any
- blank - nothing
- increment – the incremental edge count (starts from 1)
- order – the order of the edges (starts from 0)
- skip – skip over edge when counting

The "VHDL Type" is used by the VHDL generator so that the correct signal type is output to test-benches.

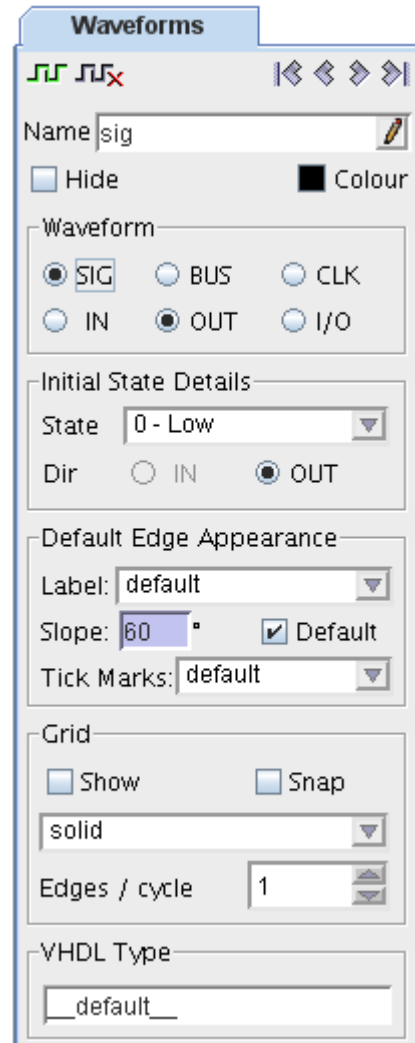


Figure 16 Signal attributes

4.4.5.2 Bus Attributes

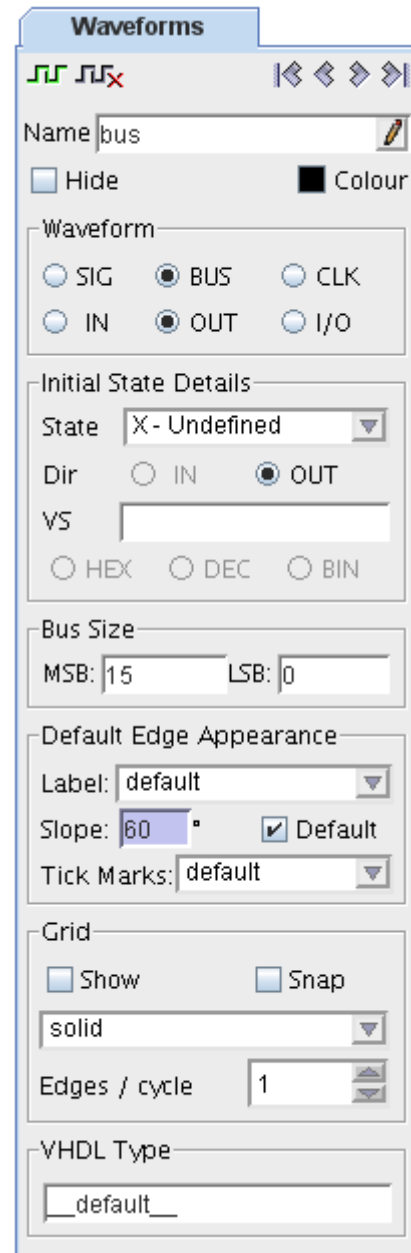
A bus can carry one or more bits of data, the bus width is entered in the 'Bus Size' section of the attribute panel.

The initial bus state defaults to X-invalid output; this can be changed in the 'Initial State Details' section of the attribute panel.

Bus edges can have the following states:

- Z – tri-state
- V - valid
- X – invalid

The initial edge can be given an initial vector state, if the state is set to valid, vector states can be entered in hex/decimal or binary format.



The screenshot shows the 'Waveforms' panel with the following settings:

- Name:** bus
- Hide:**
- Colour:**
- Waveform:**
 - SIG
 - BUS
 - CLK
 - IN
 - OUT
 - I/O
- Initial State Details:**
 - State:** X-Undefined
 - Dir:** IN, OUT
 - VS:**
 - HEX, DEC, BIN
- Bus Size:**
 - MSB:** 15
 - LSB:** 0
- Default Edge Appearance:**
 - Label:** default
 - Slope:** 60° Default
 - Tick Marks:** default
- Grid:**
 - Show, Snap
 - solid**
 - Edges / cycle:** 1
- VHDL Type:**

Figure 17 Bus Attributes

4.4.5.3 Clock Attributes

Clocks are used to synchronise the timing of events that drive a timing function, *e.g.* a memory read cycle.

The clock's edges are automatically derived from the clock's frequency; real edges are only added to the underlying source file when an object is attached to a clock edge, *e.g.* a relationship, annotation or a label.

The clock can be defined by entering its frequency or period into the 'Clock Parameters' section of the attribute panel. Note that if frequency is entered, the period is calculated to the closest value within the resolution of the diagram.

The offset attribute can be used to delay the start of the first clock pulse. Clocks without an offset start at minus infinity with the start of a pulse cycle beginning at time 0s.

The clock duty cycle may be set as a percentage with a resolution of 0.001%.

A clock can be set as inverted, if a clock is inverted the pulse at 0's or at the offset time will start as a 1 state.

The screenshot shows the 'Waveforms' attribute panel for a clock signal named 'clk'. The panel is divided into several sections:

- Name:** clk
- Hide:**
- Colour:**
- Waveform:**
 - SIG
 - BUS
 - CLK
 - IN
 - OUT
 - I/O
- Clock Parameters:**
 - Frequency: 10MHz
 - Period: 100ns
 - Offset: 0ps
 - Jitter Rising: 0ps
 - Jitter Falling: 0ps
 - Duty Cycle: 50.0
 - Inverted:
- Default Edge Appearance:**
 - Label: default
 - Slope: 60° Default
 - Tick Marks: default
- Grid:**
 - Show: Snap:
 - Line Style: solid
 - Edges / cycle: 1
- VHDL Type:**
 - __default__

Figure 18 Clock attributes

4.4.6 Grid Attributes

The grid attributes are used to create a vertical grid of lines that are positioned at the edges of the clock/signal.

There are the following grid styles available:

- Solid
- Dot
- Dash
- Dash-dot
- Dash-dot-dot

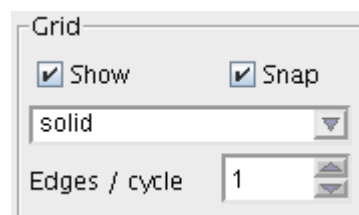


Figure 19 Grid attributes

Clock cycles / edges can be skipped, *e.g.* To show the grid line on every falling clock edge set the 'edge/cycle' attribute to 2.

Snap to grid, will effect the positioning of new edges and snap them to the times of the grid snap waveform. *E.g.* if snap to grid was set for a clock then all edges entered for other waveforms would be forced to the position of the clock edges.

The example below shows dotted grid lines generated for the selected clock

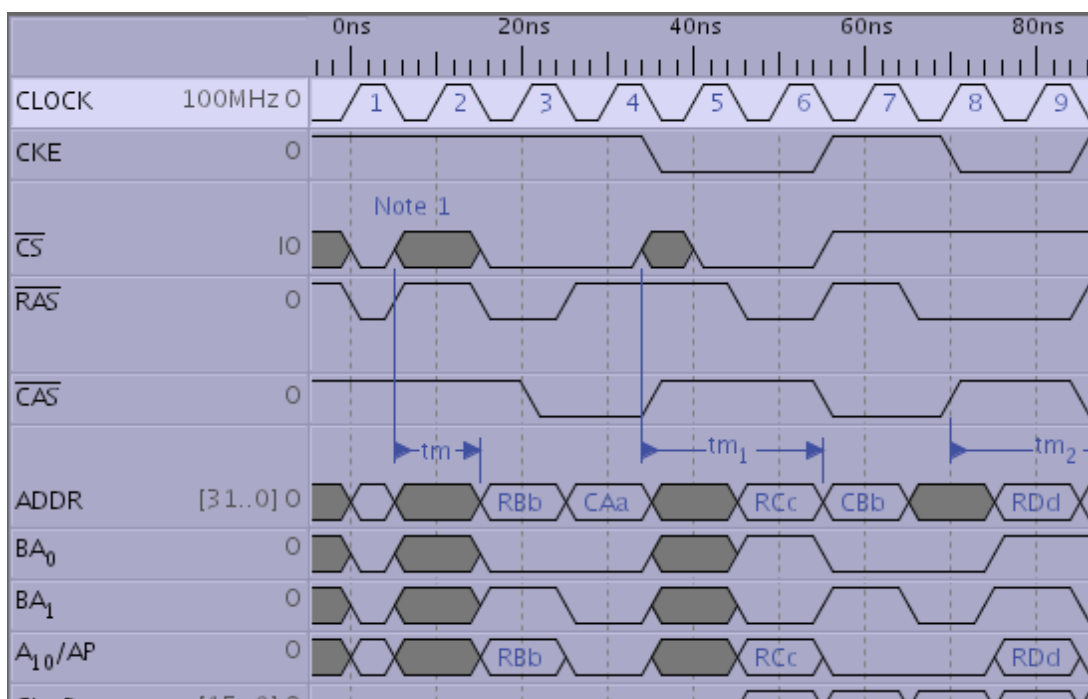


Figure 20 Grid Lines Example

4.4.7 Hiding Waveforms

Waveforms may be hidden from view. You may wish to do this for example if they have been used to derive other waveforms. To hide a waveform, click on the “hide” check box shown in Figure 21.

To globally view all hidden waveforms, click on the “View Hidden Waveforms” check box on the main menu. This permits simple navigation to hidden waveforms for editing or setting visible again.

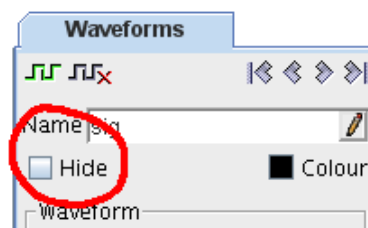
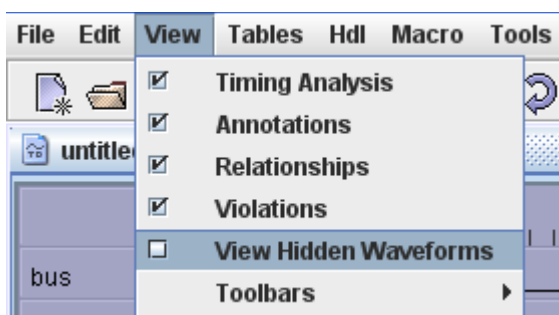


Figure 21 Hide Check Box



4.4.8 Waveform Colour

Waveforms are rendered in the default colour which may be globally modified in the user preferences. Individual waveforms may be coloured differently by clicking on the colour button, see Figure 22. A colour chooser dialogue will be displayed which may be used to either select a new colour or reset the waveform to use default colour.

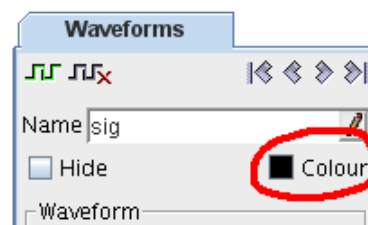


Figure 22 Colour Button

4.5 Relationships

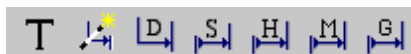
Relationships describe how one edge affects another; there are 6 relationship categories:

- Delay, represents the min/max time between two edges. When one edge is moved the other will follow it to maintain the specified delay. Delays are the only relationships that effect the time on the target edge.
- Setup, is a requirement for the target edge to occur a fixed min/max length of time before the source edge. Violations are shown in red.
- Hold, is a requirement for the target edge to occur a fixed min/max length of time after the source edge. Violations are shown in red.
- Guarantee, this is a guarantee that the time between two edges will always be between the min/max values.
- Constraint, is a requirement for the target edge to occur a fixed min/max length of time after the source edge. Violations are shown in red.
- Measure, is a measurement of the time between two edges, any linked parameter in the parameter table is automatically updated.

4.5.1 New Relationship

Relationships can be inserted in a variety of ways:


- Selecting one of the relationship icons in the main icon bar.




- Selecting the new relationship icon at the bottom of the relationship attribute panel on the right hand side of the diagram.



When a new relationship is inserted, the relationship panel changes to guide you through the required steps.

 The cursor shape will change indicating that an edge selection is required.

Select the source edge with the mouse left hand button.

 The cursor shape will now change to indicate that a target edge is now required.

Select the target edge with the left hand mouse button.

Note: A delay cannot have a clock edge as a target, since clock edges cannot be moved.

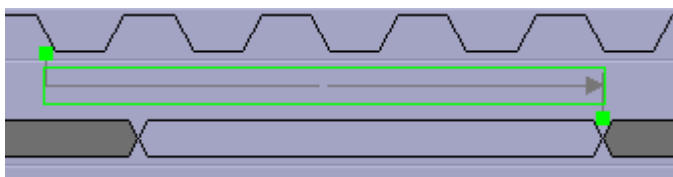


Figure 23 Creating a new relationship

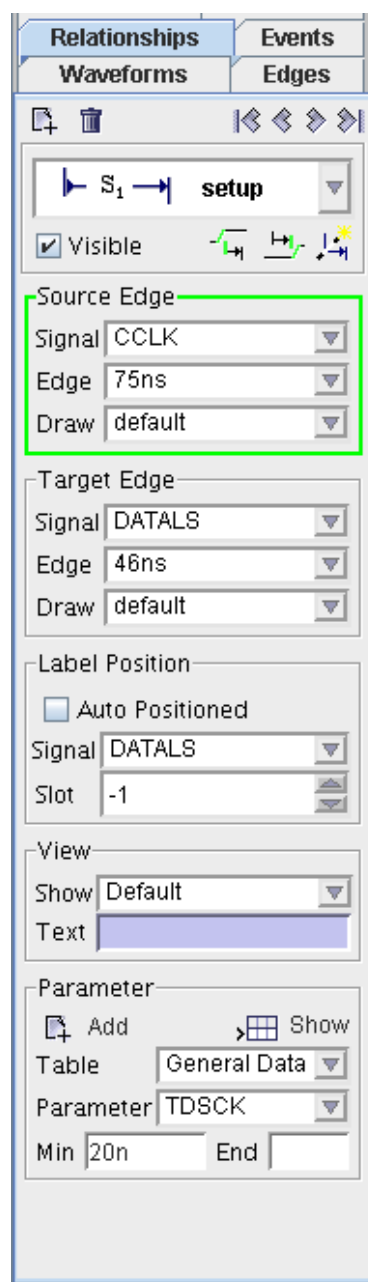


Figure 24 New relationship attribute panel

The newly created relationship will auto layout to the best position, attempting to avoid overlap with existing relationships where possible.

4.5.2 Moving a Relationship

The relationship label position can be manually placed either by selecting and dragging the label or by selecting a new position in the “Label Position” section of the attribute panel.

Relationship labels are positioned relative to a specified signal (defaulting to the lowest position on the source or target). Areas of space above and below the signals are referred to as slots (each slot is the same height as the waveform graphic), slots above the waveform are positive and below are negative.

As the relationship label is dragged up and down the slot number is shown on the right hand side of the drag box. To create additional space for the relationship, drag it to a slot above or below the top/bottom of the waveform and release, additional space is automatically allocated.

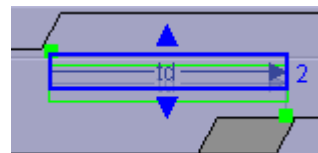


Figure 25 Dragging a relationship label

The source and target edges can be changed by selecting the small green square handles located at the source and target edges and dragging to the new location.

4.5.3 Auto Position Relationship

To automatically layout a relationship label, select the relationship with the left hand mouse button and click on the “Label Position” button on the RHS attribute panel.

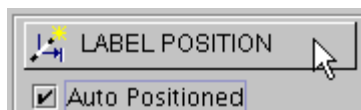



Figure 26 Auto Position Relationship

The selected relationship will auto position, attempting to avoid overlap with other relationships where possible.

To auto position all relationships that have not been manually repositioned, select the ‘Auto Layout’ icon  from the main menu bar. The relationships will be re-positioned automatically.

To turn off manual positioning of a relationship uncheck the ‘Auto Positioned’ check box, the relationship will now be positioned automatically.

Note: Relationships are not re-positioned as the diagram is scrolled or the scale changed, this has to be done as a manual action by selecting the ‘Auto Layout’ icon.

4.5.4 Associating Parameters to Relationships

For relationships to perform a function in your design, they need to be associated with parameters in the parameter table.

You can either create a new parameter or assign an existing parameter to a relationship.

To create a new parameter select the “Add” button on the parameter panel of the relationship RHS attribute panel.

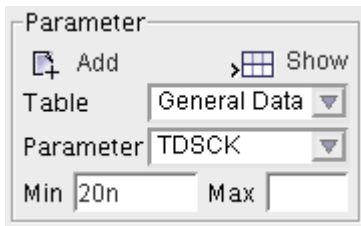


Figure 27 New relationship parameter

A parameter will be added to the parameter table.

Enter a minimum and/or a maximum value for the parameter.

Note: for a measure the min and max are calculated and filled in for you (these values will now track the measure).

To view the parameter select the “show” button, the parameter will be highlighted in the parameter table where additional attributes can be specified. See section 5, Parameter Tables.

To assign an existing parameter to a relationship select the table and parameter from the drop down list. The same parameters can be re-assigned to as many different relationships as you like.

Note: a measure cannot share a parameter with an existing relationship.

For information on using relationships to verify a design, see: Section ,

4.5.5 Hiding a relationship

Relationships can be hidden to de-clutter a diagram, hidden relationships are still active in the Timing Analysis calculations but they don't appear graphically on the diagram.

To hide a relationship de-selected the “visible” check box on the relationship RHS attribute panel. To hide all relationships select the: ‘View→ Relationships’ menu option to toggle relationship visibility.

4.5.6 Delay Relationships

Delay relationships are used to represent propagation delays from a source edge to a target edge.

TimingTool automatically runs a timing analysis algorithm every time there is a change in the diagram; this repositions the delay target edges according to a set of rules (see section 6.1 How To Define Signals).

The delay target edge attribute panel shows additional information when the target edge is selected; see Figure 28 Edge Target Type.

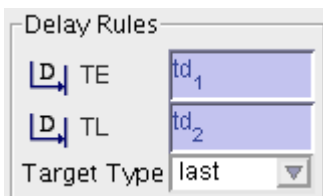


Figure 28 Edge Target Type

The information in this panel indicates which delays drive the edges te and tl. The 'target type' attribute of the edge defines the rule used to determine the critical path of the delays that drive the edge.

4.5.7 Label Types

The label displayed for a relationship is selected via the "View" panel.

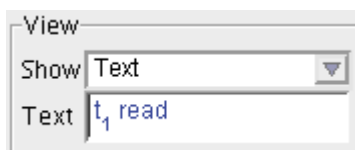


Figure 29 Relationship View Panel

The following options are available:

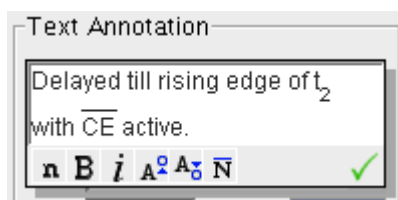
Value	Meaning
blank	No label is displayed.
default	Label type is determined on a diagram wide basis. The default is to show the symbol of the associated parameter. (blank if there is no parameter).
name	Displays relationship name.
min/max	Displays relationship min and max value
formula	Displays relationship formula
margin	Displays slack time for relationship. (Problems are highlighted in red).
comment	Displays text comment associated with relationship. This is read from the parameter description in the parameter table. (blank if there is no associated parameter)
measure	Displays time between two edges of the relationship.
text	Displays expanded custom text string.

4.6 Text Annotations

Text annotations are used to place text descriptions within the timing diagram. The text annotation can be either free floating text, anchored to a signal, or anchored to an edge. To create a new text annotation, click on the **T** in the main icon toolbar, or the new text annotation icon at the bottom of the text annotation attribute panel.

The attribute panel for text annotation consists of 3 main areas. The actual text for the text annotation, the anchor point for the text and the x, y offset position from the anchor point.

The text entry in the text annotation box can be of any length and can have text emphasis applied using the text emphasis icons that pop up during edit.



These control the text from left to right as follows: Normal, Bold, Italic, Superscript, Subscript, and Invert (bar). To finish editing, click either on the green tick or anywhere outside the edit box.

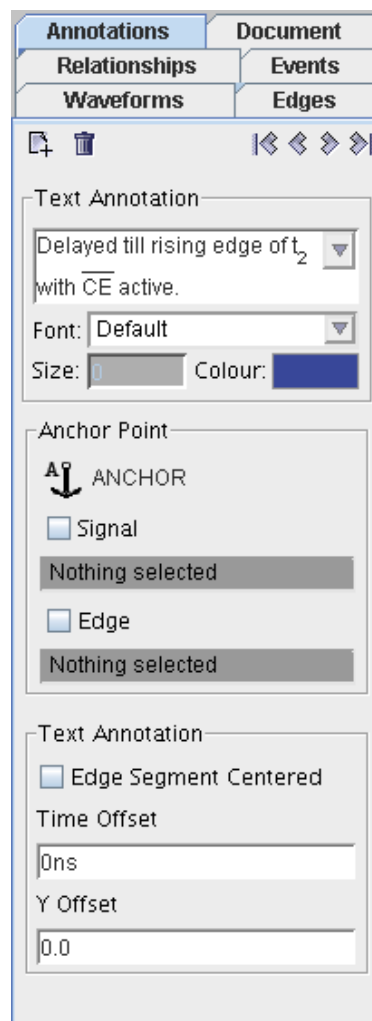


Figure 30 Text annotation attribute panel

The text font may be selected using the font selection chooser that lists all fonts available on your system. Fonts in use within the diagram are listed first. Font size may be set between 1pt and 48pt; this is independent of zoom level. The text colour may also be adjusted by clicking on the colour chooser button.

The anchor point can easily be adjusted so that the text is attached to a specific edge within a specific signal. The anchor point can be changed either by selecting the edge time or signal name. If the edge box is deselected, the anchor applies to the complete signal. If the edge box and the signal box are deselected, then the text is completely free floating; it may then be placed anywhere within the diagram.

The offsets show the time offset from the anchor point, and the 'Y' offset shows the vertical offset from the anchor point. Negative is above and positive below.

(Note: Unfortunately, some vendors have misinterpreted the TDML standard and inverted the 'Y' offset, TimingTool attempts to detect and correct this on file import).

If Edge Segment Centred box is selected, the time offset is fixed to the mid point of the anchoring edge and the next edge. The text can still be moved vertically as desired.

The icons at the bottom of the text annotation attribute panel from left to right are: new text annotation, delete annotation, first annotation, previous annotation, next annotation and last annotation.

Selecting the annotation in the diagram is the easiest way to move the text. The following shows a selected text annotation.

To move the text, left click and hold and move the text to the desired location. The diagram automatically moves signals apart to make room for the text.

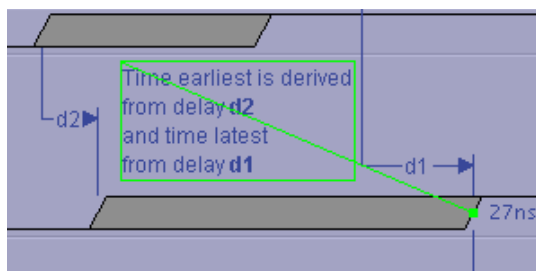
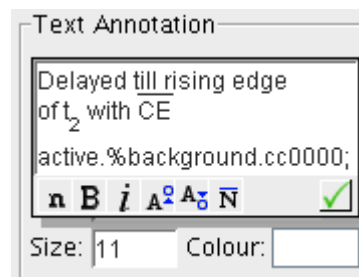


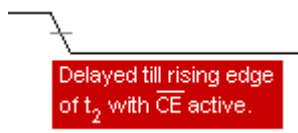
Figure 31 Text Annotation Drag Box

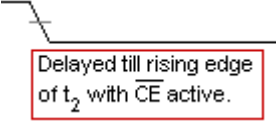
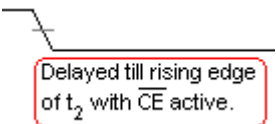
Annotations may be dragged above or below a signal to create extra space between waveforms. Further, dragging an annotation above the first waveform or, conversely, dragging an annotation below the last waveform will create space above or below the diagram respectively.

Annotations may be decorated using pre-defined keywords within the text. Further keywords may be found in section 4.14.



Keyword	Description	Example
<code>%background.COLOUR</code>	<p>Set background colour where COLOUR is:</p> <ul style="list-style-type: none"> A 24 bit hexadecimal number representing the colour as three, 8 bit RGB components. A 24 bit hexadecimal number representing the colour as 4, eight bit ARGB components. "A" is the alpha (transparency) value from 01h to FFh (1 to 255). Named colours: red, green, blue, white, black or grey. 	<pre>%background.cc0000;</pre> <pre>%background.80202020;</pre> <pre>%background.red;</pre>



Keyword	Description	Example
<p><code>%border.</code>COLOUR;</p> 	<p>Set a border around the text where the colour is as described with the “background” keyword.</p>	<p><code>%border.black;</code></p>
<p><code>%shape.</code>SHAPE;</p> 	<p>Sets the shape of the background and/or border. As default it is square but either “square” or “round” may be specified. Note that this value may be appended to the “background” or “border” keywords.</p>	<p><code>%shape.round;</code> <code>%background.red.round;</code></p>

4.7 Events

TimingTool provides a number of graphical markings that illustrate the occurrence of time related events.

- Marker – to illustrate when a significant event occurs
- Break – to mark the start and end of a time period where waveform display is not required.
- End – to mark the end of a diagram beyond which waveform display is not required.

4.7.1 Time Breaks

Time breaks are used to hide time periods containing uninteresting timing information.

An example of an ADC conversion is shown to the right; the microprocessor instruction cycles are short in comparison to the conversion itself however using time breaks, all parts of the cycle can be seen clearly in one diagram.

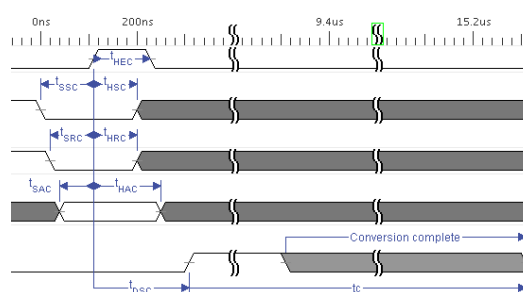


Figure 32 Diagram with Breaks

Time break attributes may be added, modified or deleted in the “Events” attribute panel.

Select “break” from the event type drop down list. From here a new break may be added or existing breaks modified.

Breaks may also be added by right clicking over the diagram timescale and select “insert break”. A break created in this way places a break at the cursor.

Breaks may be relative to an edge or placed at an absolute time. To attach to an edge, use the “Source Edge” panel drop down lists to select the desired edge.

Alternatively use the source edge button to select a visible edge using the mouse.

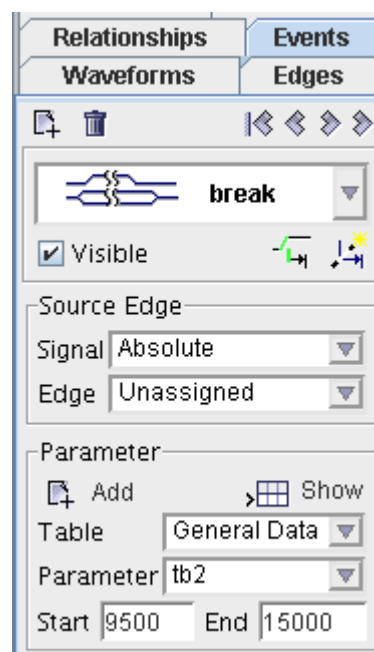


Figure 33 Break Attributes

The start and end times of a break are configured using a parameter. The parameter may be adjusted in the “Parameter” panel or in the parameter tables. Note that the start time equates to the “MIN” time and the end time equates to the “MAX” time. You may omit the end time or set the start time to equal the end time to show a break diagrammatically without breaking the timescale.

Background checks are made to ensure time break entries are valid. End times, where defined, must always be greater than or equal to the start time. Further, multiple time breaks may not overlap. If an error is found, then one or more breaks may be automatically hidden and an error message will be displayed. Should this occur, use the attribute panel to navigate through each break event and correct as necessary; setting the breaks visible again as required.

4.7.2 Markers and the End Marker

A marker is a vertical line with an optional text comment. A variation on a standard marker is the end of diagram marker; this is identical to a standard marker with the exception that waveforms are not drawn beyond it. A marker’s attributes may be added, modified or deleted in the “Events” attribute panel.

Select “marker” (or “end”) from the event type drop down list. From here a new marker may be added or existing markers modified.

Markers may be relative to an edge or placed at an absolute time. To attach to an edge, use the “Source Edge” panel drop down lists to select the desired edge.

The offset of a marker from an edge (or time zero) is configured using a parameter. The parameter may be adjusted in the “Parameter” panel or in the parameter tables. Note that the offset equates to the “MIN” time or to the “MAX” time. If you enter both “MIN” and “MAX” then the mid-time will be calculated.



Figure 34 Marker Attributes

4.8 Document Details

Select the Document tab from the attribute panel. This will display the panel as shown in Figure 35.

The filename is for display purposes only and is not editable.

The Title field is a single line free text string.

The Version field is a single line free text string.

The Description field is a multi line free text string. This can be any length.

4.8.1 Base Time Units

The Base Time Units are used whenever a time value is entered (in this example ns). The base time can be set to sec, ms, us, ns, ps, or fs. If a time value is entered into a time field and no units are specified then the base time unit is always used. If a time is entered with units (*e.g.* 6ps) then this overrides the base time units as long as there is resolution available. The table below shows different base time units and the effect they have on time resolution.

Time Required	Base Time Unit	Diagram Time
1.1 ns	ns	1 ns
1.1 ns	ps	1100 ps
6 ps	ns	0 ns
6 ps	ps	6 ps

The best approach is to make the base time units of sufficient resolution as to allow for any time that you require.

i.e. If you require precision to 0.001ns then select the base time units as ps.

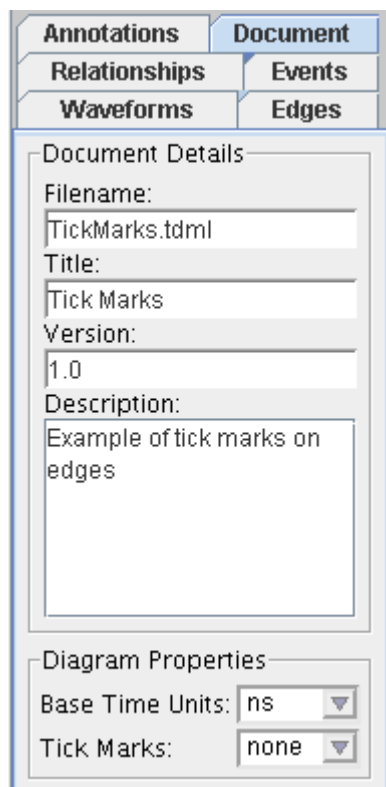


Figure 35 Document Details

4.9 Import

TimingTool has the capability of importing waveforms from many different sources.

4.9.1 HDL

A full description of the HDL (VHDL, Verilog) import capability is given in section 8.1.

4.9.2 External TDML Source

TimingTool has the capability of importing TDML generated from other tools. When importing, the TDML is rigorously checked so that any discrepancies from other tools TDML can be automatically corrected. This allows for easy and safe migration from legacy TDML into TimingTool.

4.10 Loading and Saving



A timing diagram file may be opened by selecting File → Open from the menu or by clicking on the file open button.

When opening, a file open popup dialog is displayed as shown in Figure 36. This dialog has many of the common features that can be expected for browsing around the file system. There are five icons in the top right of this dialog:



Select level of detail displayed for each file entry



Move up to the parent directory



Select home directory



Create a new directory



Refresh current view

In Windows based machines the home directory is placed in:

My Documents\TimingTool Files\work

In Unix / Linux based machines the home directory is placed in:

`/home/username/timingtool/work`

File Types default to *.tdml, *.tdm, but HDL file types can also be opened. When opening an HDL file TimingTool looks for a timing diagram in the same directory with the same file name (with the .tdml extension) and opens this. If there is no associated timing diagram, the HDL file is parsed and a new diagram is automatically generated (see section 8.1).

If you are using a Windows based machine you may also wish to map to a network drive. For best results map a network drive to a drive letter and then

the drive letter is displayed in the open file dialog. To access a file quickly on a network drive it is also possible by typing into the filename field in the following format: `\\network_drive\directory`.

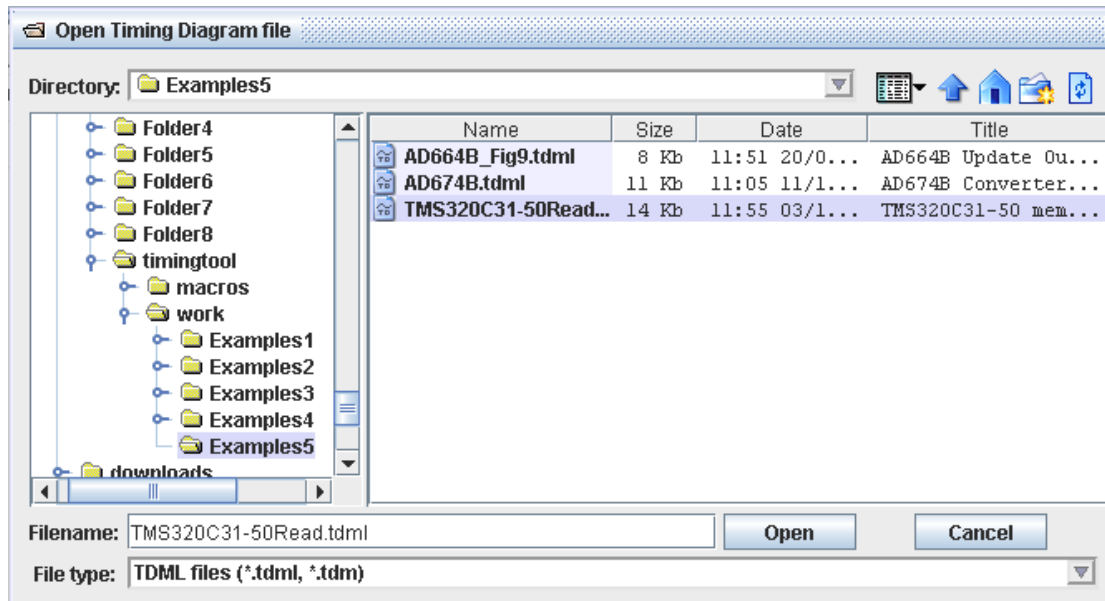


Figure 36 Open file dialog

When saving the save file popup dialog is almost identical to the open dialog. The main differences are that HDL is not supplied as a file type and there is a Save button. The network drive mechanism for windows based machine is also the same as with the file open dialog.

4.11 Export

From the main menu select File → Export Diagram, this opens the popup dialog as show in Figure 37. The exported graphic can be manipulated to suit almost any look and feel. Within the view panel it is possible to enter your company logo which will be displayed on the bottom right of the exported graphic. The default setting places a small TimingTool logo in the logo position and none switches logos off completely.

The timescale along the top of the exported graphic can be switched off. The signal names down the left of the exported graphic can be switched off. Colours can be altered for the diagram background, waveforms, signal labels, timescale, relationships, and violations.

The four buttons at the bottom (Export, Apply, Save and Close) have the following functions.

- | | |
|--------|---|
| Export | Generates a save popup dialog that allows you to save the exported diagram. |
| Apply | Makes your new export diagram settings live within the display area. |
| Save | Saves your export diagram settings so that the next diagram that is exported will have a standard style |
| Close | Closes the dialog. |

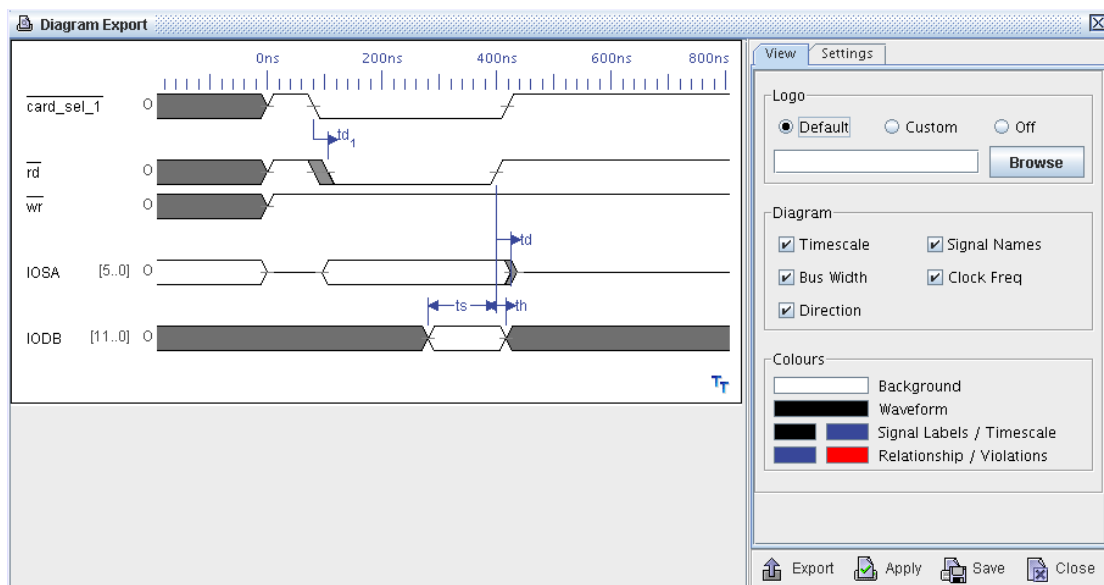


Figure 37 Export Diagram

The Settings tab shown in Figure 38 allows the user to change the width and height of the exported diagram bitmap size. Within the bitmap size the diagram can be scaled to get the best fit. The image format may also be chosen (see section 4.11.1 and section 4.11.2).

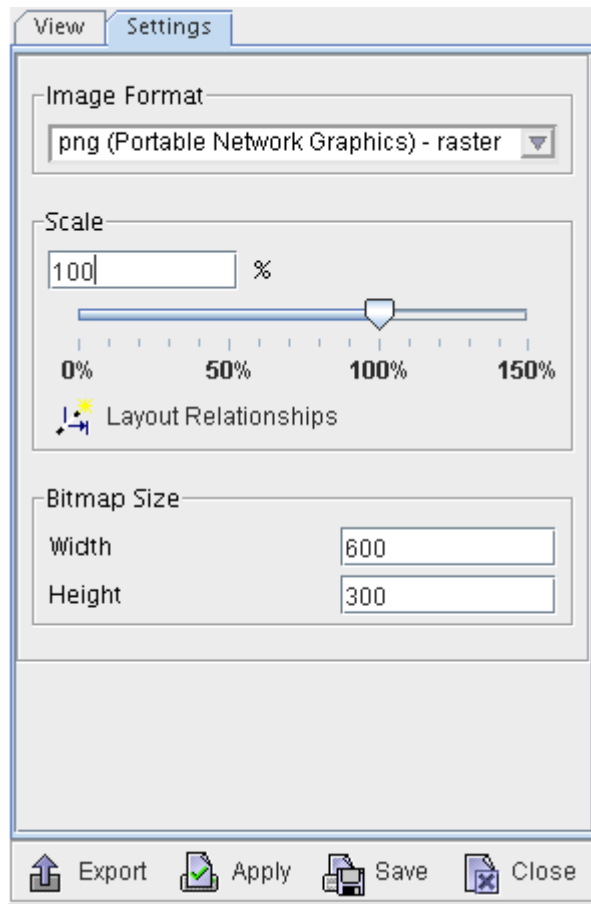


Figure 38 Export diagram settings

4.11.1 Graphic Bitmap Output

Bitmap output formats supported are as follows.

- xbm – X BitMap
- jpg – Joint Picture Expert Group
- pict – Picture Macintosh graphics
- bmp – Bitmap graphics
- psd – Adobe Photoshop
- pcx – Picture image
- tga – TarGA image format
- png – Portable Network Graphics
- gif – Graphics Interchange Format

4.11.2 Graphic Vector Output

Vector output formats supported are as follows.

- svg – Scalable Vector Graphic (XML Based internet standard)
- emf – Enhanced Meta File (Windows compatible format)
- ps – Postscript

4.11.3 Document Output

For easy documentation and web publishing the PDF format is supplied.

4.12 HTML Output

From the main menu select File → Export HTML. A popup dialog will be generated that is very similar to Export Diagram. The Export HTML dialog is shown in Figure 39. The HTML export dialog is split into different areas. The main graphic area shows a preview of the HTML that will be exported or the HTML source code itself. To select what is displayed, use the Datasheet or Source tabs at the bottom left of this dialog.

Changes can be made to either Datasheet text (not the graphics of the timing diagram) or the HTML source and then exported to a file. The changes made to the exported HTML file will be lost if the Apply button is pressed, or the Export HTML dialog is closed and then re-opened.

The four buttons at the bottom (Export, Apply, Browser and Close) have the following functions.

- Export** Generates a save popup dialog that allows you to save the exported HTML.
- Apply** Makes your new export HTML settings live within the preview area.
- Browser** Requests the default system browser to display the exported HTML within it. The browser may be configured in the preferences.
- Close** Closes the dialog.

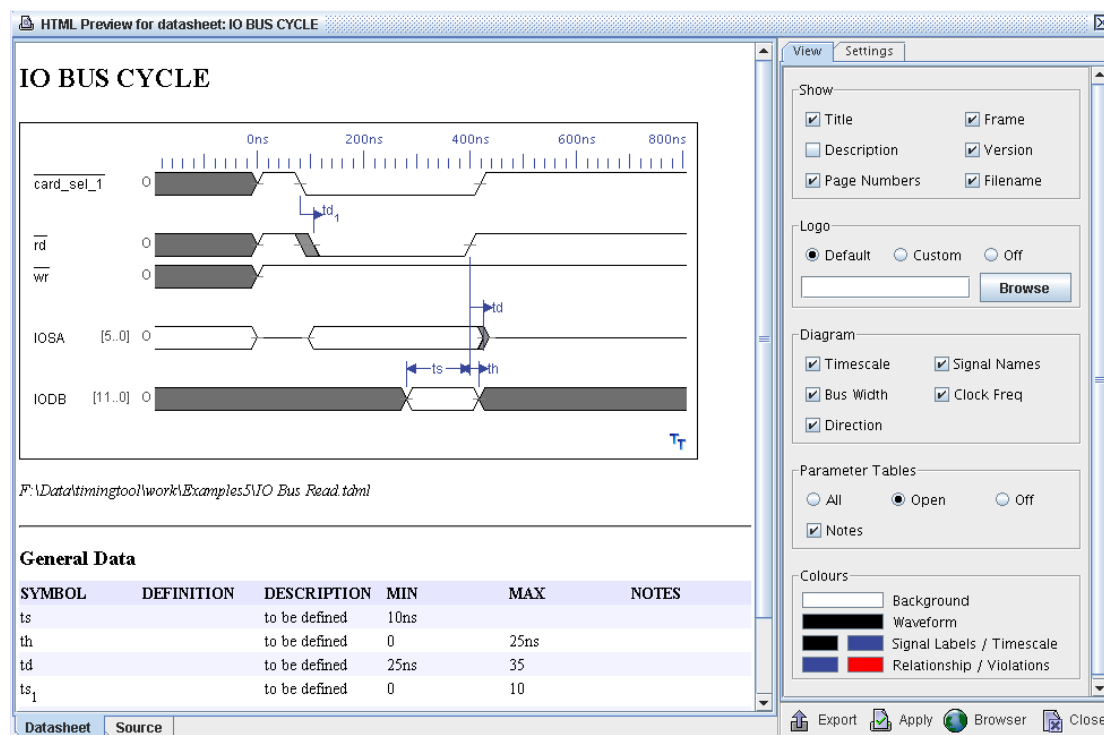


Figure 39 Export HTML

4.12.1 View Tab

The following sections refer to the control panel on the right of the Export HTML popup dialog when the View tab is selected from the top right.

4.12.1.1HTML Output Show Selections

The Show selections on the top right panel are used to select the following within the output graphic within the html page:

Title	Diagram title, in this example its set to "IO BUS CYCLE".
Frame	The box surrounding the diagram.
Description	The description defined within the Document panel (see section 4.8).
Version	The version defined within the Document panel (see section 4.8).
Page Numbers	Places a page number at the bottom of each HTML page.
Filename	Places the path and filename to the original TDML under the diagram.

Note: To update the preview click "Apply".

4.12.1.2Logo

The default logo is a small TimingTool logo that is placed in the bottom right of the timing diagram. The "Custom" options allow for company logos to be imported into the diagram (only png, gif, and jpg image types are accepted). The "Off" option removes the logo completely. For any changes to take effect with the HTML preview area, the Apply button must be pressed at the bottom of the right panel.

4.12.1.3Diagram

Within the timing diagram in the Preview HTML area, both the Timescale along the top and the signal names down the left can be displayed or removed as required. For any changes to take effect the Apply button must be pressed.

4.12.1.4Parameter Tables

Parameter tables can be output into the HTML as shown in Figure 40; either all parameter tables are displayed, only the currently open or none at all (off). The open parameter tables are made open by selecting Table from the main menu and then selecting which parameter tables you wish to display.

General Data

SYMBOL	DESCRIPTION	MIN	MAX
t_C	Conversion time 8 - Bit Cycle	6us	10us
t_C	Conversion time 12 - Bit Cycle	9us	15us
t_{DSC}	STS Delay from CE		200
t_{HEC}	CE Pulse Width	50	
t_{SSC}	CS to CE Setup	50	
t_{HSC}	CS Low During CE High	50	
t_{SRC}	R/C to CE Setup	50	
t_{HRC}	R/C LOW During CE High	50	
t_{SAC}	A_0 to CE Setup	0	
t_{HAC}	A_0 Valid During CE High	50	

Figure 40 Parameter table exported as HTML

The layout for each parameter table is based on the currently saved layout for the current diagram; if you have adjusted a table's layout you must save the changes to take effect (see 5.8 Customising the Table Layout).

4.12.1.5 Colours

Colours can be altered for the diagram background, waveforms, signal labels, timescale, relationships, and violations.

4.12.2 Settings Tab

The following refers to the control panel on the right of the Export HTML popup dialog when the Settings tab is selected from the top right. This control panels selects the output image as when exporting a diagram (see Figure 38). Care should be taken when choosing what image format is to be used for the timing diagram graphic. Not all browsers support all the image formats that TimingTool can generate, and usually if the diagram type is not supported, the HTML page is displayed as blank. For a universal image type that is compact, try using png. Vector formats such as svg are not supported by most of the common Internet browsers yet.

Note: If Save is pressed in Export Diagram, the image format is saved along with sizes, and colours of an image. These settings are automatically picked up by the Export HTML dialog.

4.13 Printing

TimingTool when printing loads the current printers that are set in the operating system.

4.13.1 Print Preview

From the main menu, select File → Print Preview. This generates the print preview popup dialog as shown in Figure 41. The 4 buttons at the bottom (Print, Apply, Save and Close) have the following functions.

- | | |
|-------|---|
| Print | Prints the current page with the currently applied settings. |
| Apply | Makes your new print preview settings live within the preview area. |
| Save | Saves your current print preview settings as default. |
| Close | Closes the dialog. |

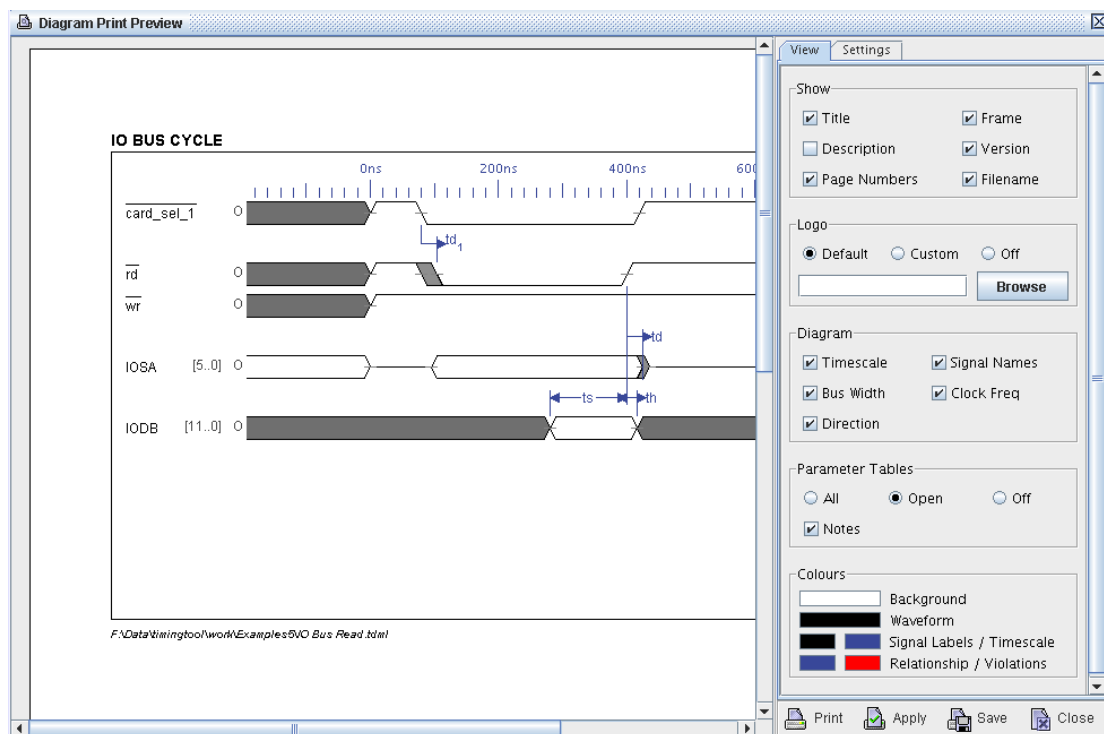


Figure 41 Print Preview

4.13.1.1 View Tab

The following sections refer to the control panel on the right of the Print Preview popup dialog when the View tab is selected from the top right.

4.13.1.1.1 Print Preview Show Selections

The Show selections on the top right panel are used to switch on and off the following within the output graphic within the preview page (you must 'Apply' each change in selection for the changes to be seen in the preview pane).

- | | |
|-------|---|
| Title | Diagram title, in this example its set to "IO BUS CYCLE". |
| Frame | The box surrounding the diagram. |

Description	The description defined within the Document panel (see section 4.8).
Version	The version defined within the Document panel (see section 4.8).
Page Numbers	Places a page number at the bottom of each page.
Filename	Places the path and filename to the original TDML under the diagram.

4.13.1.1.2 Logo

The default logo is a small TimingTool logo placed in the bottom right of the timing diagram. The “Custom” option allows company logos to be imported into the diagram (only png, gif, and jpg image types are accepted). The “Off” option removes the logo completely. For any changes to take effect with the preview area, the “Apply” button must be pressed at the bottom of the right panel.

4.13.1.1.3 Diagram

Within the timing diagram in the Preview area, both the Timescale along the top and the signal names down the left can be displayed or removed as required. For any changes to take effect the Apply button must be pressed.

4.13.1.1.4 Parameter Tables

Parameter tables can be output into the preview page as shown in Figure 42; either all parameter tables are displayed, only the currently open or none at all (off). The open parameter tables are made open by selecting Table from the main menu and then selecting which parameter tables you wish to display.

General Data					
SYMBOL	DEFINITION	DESCRIPTION	MIN	MAX	NOTES
d1		to be defined	5	10	
d2		to be defined	5	10	
d3		to be defined	2	4	
d4		to be defined	2	4	
d5		to be defined	5	5	
t1		to be defined	6	40	
measure		to be defined	6	40	
measure1		to be defined	6	60	
measure d1 d2		to be defined	6	40	

Figure 42 Print Preview Parameter Table

4.13.1.1.5 Colours

Colours can be altered for the diagram background, waveforms, signal labels, timescale, relationships, and violations.

4.13.1.2 Print Preview Settings Tab

The print preview settings are show in Figure 43. The printer settings allow the user to select between different printers (network or local) and the paper size being used by the selected printer. The orientation can also be selected between landscape and portrait.

The scale is used to alter the scaling so that more or less appears on a page when printing. This is useful for forcing the page break positions on large diagrams that roll over multiple pages.

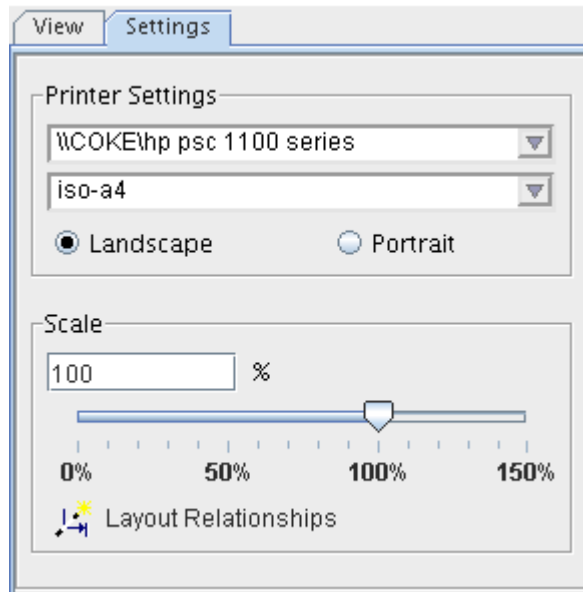


Figure 43 Print Preview Settings

4.13.2 Print Dialog

Selecting File → Print from the main menu generates the Print popup dialog as shown in Figure 44. This allows the user to change any of the printer settings for their printer such as paper size, number of copies etc.

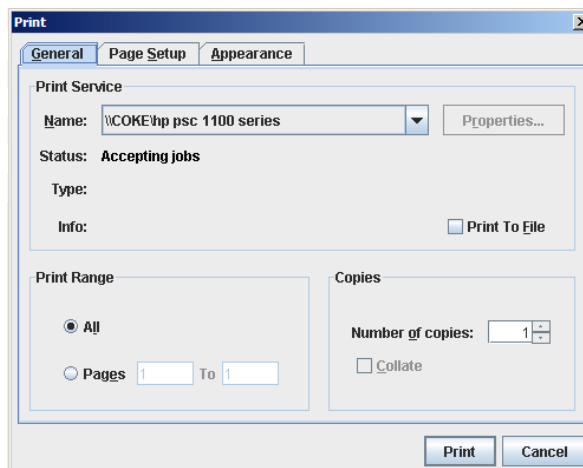


Figure 44 Print Dialog

4.14 Keywords

TimingTool includes a number of keywords used within certain text fields for substituting dynamic data. The following sub-sections describe the use of these keywords and where they may be used. Keywords always start with a '%' character and may be terminated using a non-alpha character. To clearly indicate the end of a keyword, use a semi-colon ";" – the semi-colon will not print but any other terminator will. Where further arguments are required these are separated from the main keyword and any other arguments by a period "." character.

4.14.1 System

The system keywords may be used in annotations, edge labels, relationship labels and event labels.

Keyword	Description
%file;	The current file name.
%title;	The title of the diagram
%path;	The file path to this document
%time;	The time of the last save.
%date;	Time and date stamp in the format configured in the preferences.

4.14.2 Parameter Data

Parameter data keywords may be used in relationship labels and event labels.

Keyword	Description
%sym;	The symbol in plain text.
%max;	The maximum time value.
%min;	The minimum time value.
%value;	The min and max in standard format "[min,max]"
%desc;	The description in plain text.
%param. <i>PARAM</i> ;	The time value of the parameter passed where <i>PARAM</i> is "min", "max", "typ", "nom" etc.

4.14.3 Edge Data

Edge data keywords may be used in edge labels only.

Keyword	Description
%early;	The edge's time earliest.
%late;	The edge's time latest.
%state;	The state after this edge.
%vector;	The vector state after this edge (for a bus waveform).

4.14.4 Signal Data

Signal data keywords may be used in edge labels and annotations.

Keyword	Description
%signal;	The signal's name in plain text.
%frequency;	The signal's frequency (if a clock).
%period;	The signal's period (if a clock).
%offset;	The signal's offset (if a clock).

5 Parameter Tables

Parameter tables hold any number of pieces of information that are required to correctly specify the timing characteristics for a timing diagram. TimingTool creates a general data table for each new diagram. Parameters for setups, holds and delays etc may be entered into this table whilst drawing a timing diagram. By default, all parameter tables are opened when a diagram is opened. You may close or re-open tables from the tables menu (as shown in Figure 45).

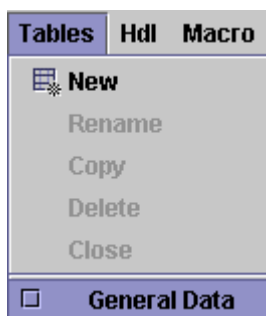


Figure 45 Tables Menu

The user can select between the timing diagram and any open parameter table using the tabs at the bottom left of the diagram. Once a parameter table is selected for viewing it is displayed in the main TimingTool window. An example general data parameter table is shown in Figure 46.

SYMBOL	DESCRIPTION	MIN	MAX
t_{C1}	Conversion time 8 - Bit Cycle	8us	10us
t_c	Conversion time 12 - Bit Cycle	9us	15us
t_{DSC}	STS Delay from CE		200
t_{HEC}	CE Pulse Width	50	
t_{SSC}	\overline{CS} to CE Setup	50	
t_{HSC}	\overline{CS} Low During CE High	50	
t_{SRC}	R/\overline{C} to CE Setup	50	
t_{HRC}	R/\overline{C} LOW During CE High	50	
t_{SAC}	A_0 to CE Setup	0	
t_{HAC}	A_0 Valid During CE High	50	
t_{b1}	to be defined	400	9200
t_{b2}	to be defined	9500	15000
t_m	Marker offset from leading time earliest	6000	f_n

Notes: Show

Timing Diagram | General Data

Figure 46 General Data Parameter Table

5.1 Creating and editing complete tables

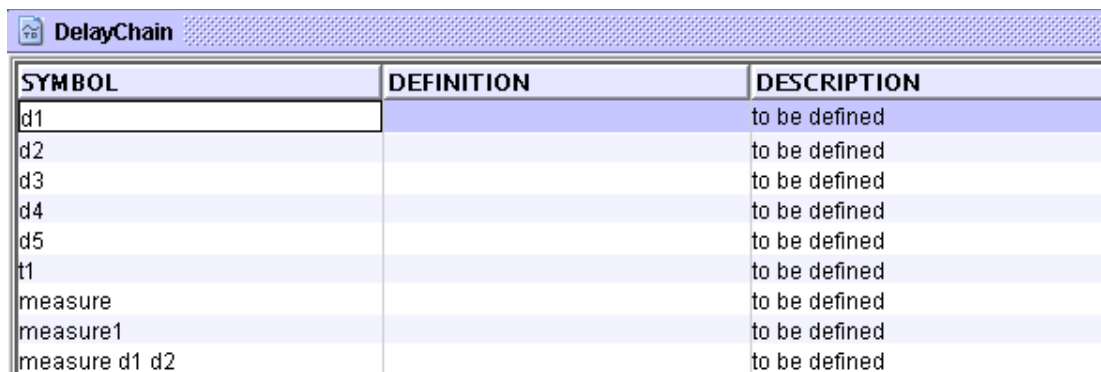
Any number of parameter tables can be added to a timing diagram. To create a new table, select from the menu bar: 'Tables → New', (see Figure 45). Once a table has been created it may be Renamed, Deleted, or Closed. Closing a table removes it from the main TimingTool window, but does not delete it from the timing diagram.

One very useful feature of the tables menu is the Copy table capability. This allows the user to make a copy of the currently active parameter table into a new table. When describing components timings there may be multiple speed grades of the component. Hence making a copy of one parameter table (for one speed of component) to another and then making the timing changes greatly speeds up parameter entry.

5.2 Selecting and moving around a parameter table

To select a cell within a parameter table simply click on it and the cell will open for editing. Figure 47 shows a selected cell (row 1 of symbol – d1) within a parameter table. To move around the cells the cursors keys or the tab are used. To move the caret within a cell, use the mouse and click on the exact position that is required for editing, or use the numeric keypad cursor keys. To go to the beginning of the current line use the home key, and to go to the end of the current line use the end key. To go to the first line of a multi line cell use the page up key. To go to the last line of a multi line cell, use the page down key.

Multiple characters within a cell can be selected by left click and hold, and then drag.



SYMBOL	DEFINITION	DESCRIPTION
d1		to be defined
d2		to be defined
d3		to be defined
d4		to be defined
d5		to be defined
t1		to be defined
measure		to be defined
measure1		to be defined
measure d1 d2		to be defined

Figure 47 Parameter table cell selection

Selecting a whole row and moving up and down a row can also be performed using the arrow icons at the bottom of the parameter table.



From left to right these icons perform select first row in table, select next row up, select next row down, and select the last row in the parameter table.

5.3 Adding and Deleting a parameter

To add a complete parameter entry to the table simply click on the “add new row” icon or type ALT I. The new row is added after the current selection or if no rows are selected then at the bottom of the list of parameters.

To delete a row, first select the row that is to be deleted as described in section 5.2. To delete the row simply click on the “delete selected row” icon or type ALT D.

To delete multiple rows, select multiple rows using the shift key whilst selecting then click on the “delete selected row” icon or type ALT D. The “add” and “delete” icons shown left to right are shown below.



5.4 Moving parameters

Parameters may be moved either by sorting (see section 5.8.1 below) or by using the “Row move” icons.

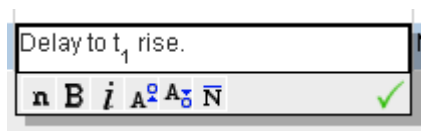


5.5 Editing the parameters

Once a cell is selected for editing, values can be typed directly in as free text. Two column types that are not editable are: id, and margin, these parameters are derived from within the timing diagram. In addition, a measure from within the diagram when displayed in the parameter table will show the min, max times. These times are also derived from the diagram, and so cannot be edited.

When entering time into a cell, text can be entered but obviously when performing calculations it will have no effect. Numbers can be entered with their time units (sec, ms, us, ps, fs) or if no time unit is specified then the diagram base time unit is used. The base time unit can be set on the timing diagram document details panel (see section 4.8).

Text descriptions can be modified using the text control icons on the main icon bar.



These icons allow text to be made bold, italic, superscript, subscript, and with an inversion bar. Multiple characters can be affected if selected prior to applying the text control. The lowercase ‘n’ will revert the selected text back to normal.

5.6 Parameter expressions

The cells that represent time (MIN, MAX, TYP, NOM, NTOL, PTOL and RAW) may contain mathematical expressions. To enter an expression simply prefix the expression with an equals symbol '='. After entering an expression, it will be automatically evaluated and the result will be displayed. On re-editing the cell, the expression will be displayed to permit modification.

All expression within the table may be evaluated by clicking on the calculator icon in the tool tray.



The 'Auto' checkbox when selected will cause all expressions to be recalculated every time an edit is made to the parameter table. This always defaults to off whenever TimingTool is started.

Expressions may be simple mathematical expressions for example:

$$= (3 * 4) - 2$$

This simple expression will always display as '10'.

Reference may be made to other cells using built in functions named after the column names for example:

$$= \text{MAX}() - 2$$

This expression takes the value in the MAX cell on this row and subtracts 2.

Other parameters may be referenced thus:

$$= \text{MAX}(t_1) / 3$$

This expression takes the maximum value for parameter 't₁' and divides it by three. The plain text version of the symbol name is used to match with the argument specified.

Parameters in other tables may be referenced by including an extra argument:

$$= \text{MAX}(\text{Fast Timings}, t_1) / 3$$

This expression takes the maximum value for parameter 't₁' from a table named "Fast Timings" and divides it by three.

Handy hint: The string parameter for the table name could be taken from another field:

```
= MAX(DEF($Speed), t1)
```

Many expressions can then reference a single field as a "pointer" to the table you wish to use. This is an easy way to try different sets of timings *e.g.* Change device for a faster/slower part easily. Remember, tables may be copied and then timings in the copies may be adjusted.

Should you wish to display the result in engineering units then use the function 'ENG()' thus:

```
= ENG(TYP(t2) / 2)
```

This expression would result in something like '1.5ns'.

Multiple statements may be entered. Statements must be separated by semi-colons. The result of the last statement is returned as the cell value. For example:

```
= x = 2; y = 3; x + y
```

This expression will display '5'.

The expression engine is a loose Java interpreter so any valid Java style statements may be processed. For more information, see the Macro section.

5.7 What the parameters in the parameter table mean

The following table lists the parameters that are available and a description of their purpose.

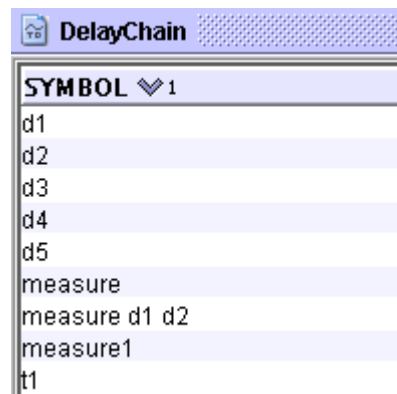
Parameter	Description
SYMBOL	Symbolic description of the parameter e.g. T _{SU}
DEFINITION	Multi line definition of the parameter: This parameter is only valid when parameter x y z is within certain tolerances. This parameter is only valid for the following power supply conditions This parameter is only valid over these environmental conditions.
DESCRIPTION	Single line description of the parameter – e.g. Data Time Set Up
MIN	(Minimum Time) The smallest permitted or observed value for a property. (zero tolerance)
MAX	(Maximum Time) The upper limit at which the provider of the device said the device will operate.
MARGIN	Shows how close a timing constraint is to failing or by how much it has failed.
ID	Non editable – shows the internal TDML assigned ID
TYP	(Typical Time) A value suitable for operation, usually determined by some form of statistical averaging measured at mid-range. This is derived from measured values.
NOM	(Nominal Time) An optimised value suitable for operations of all devices in a population. Nominal is the point around which the device is designed. Nominal is the same as Optimal and is not always measured.
NTOL	(Negative tolerance) Relates only to nominal values, not to typical ones.
PTOL	(Positive tolerance) Relates only to nominal values, not to typical ones.
RAW	Direct measured value.
TEST INFO	General test information.
NOTES	General notes text field.

5.8 Customising the Table Layout

5.8.1 Basic Layout

The displaying of parameter tables can be modified in many different ways. The column widths can easily be adjusted and the order of the columns can be changed simply by clicking and holding the column title bar. If you left click and hold in the middle of the column title bar and then move left or right, the column is repositioned when the left mouse button is released. If you left click and hold at the edge of a column in the column title bar and move the cursor left or right, the width of the column can be adjusted.

Right clicking on the column title bar will add a sort to the column so that each row is in alphabetical order. The first right click makes the sort ascending, and the second right click makes the sort descending. A third right click turns the sort back off. Figure 48 shows the ascending sort applied to the SYMBOL column.



SYMBOL ▾ 1
d1
d2
d3
d4
d5
measure
measure d1 d2
measure1
t1

Figure 48 Parameter table ascending sort

5.8.2 Advanced Layout

Selecting the advanced layout table format from the icon bar at the bottom of the parameter table will generate the table format popup dialog (see Figure 49).



The above icons from left to right allow the user run a table sort and start the table format tool.

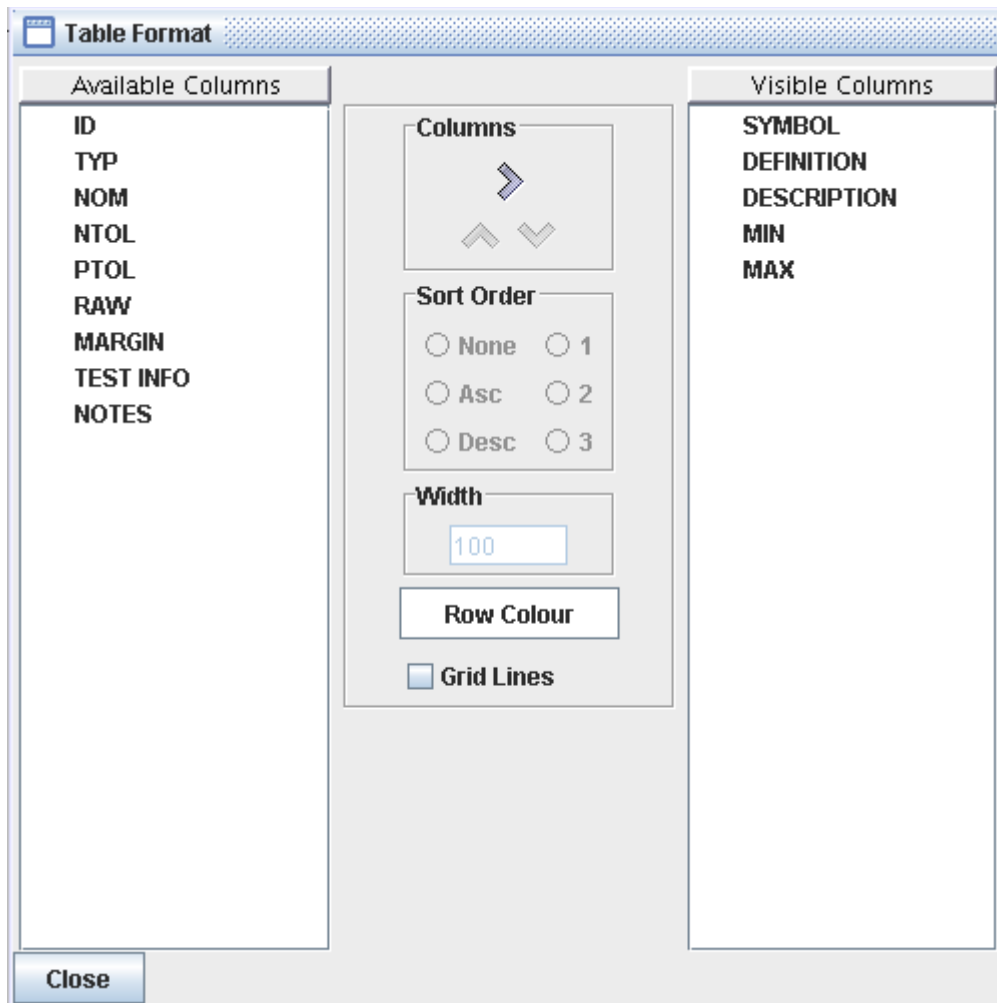


Figure 49 Parameter table format

The table format dialog allows the user to select which columns are displayed, which order the columns are displayed in, and the column widths. In addition, complex sorts can be applied to one column followed by another.

Selecting the Row Colour generates a further popup dialog with a full colour chooser.

Selecting grid lines places a black line between each cell throughout the whole table.

5.8.3 Saving your new parameter display format

Once the parameter table format has been changed it is possible to save the new format for all future use. The icons on the bottom of the parameter table that are shown below are for saving parameter table display formats.



From left to right these icons perform as:

Restore saved table layout.

Restore default table layout.

Save the current table layout in this document.

Save the current layout as the default layout.

Note: You will need to save your new table layout before printing, saving or exporting your diagram.

5.9 Exporting data from a parameter table

All parameters held in a parameter table can be exported. Below shows the icons that are used for export and are located at the bottom of the parameter table icon bar.



There are formats for exporting:

Export to clipboard suitable for pasting into other applications such as a Microsoft Excel spreadsheet.

Export to a Comma Separated Variable (CSV) file.

Export to a HTML file.

Exports to clipboard or CSV file may be selective. Select two or more rows and only these will be exported.

5.10 Parameter table notes

At the bottom left of the parameter table there is a notes button. This allows users to enter any general notes they require to make about the parameters. Figure 50 is an example of parameter notes.

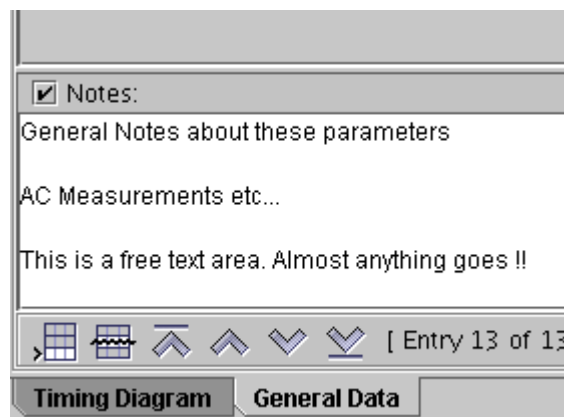


Figure 50 Parameter Table Notes

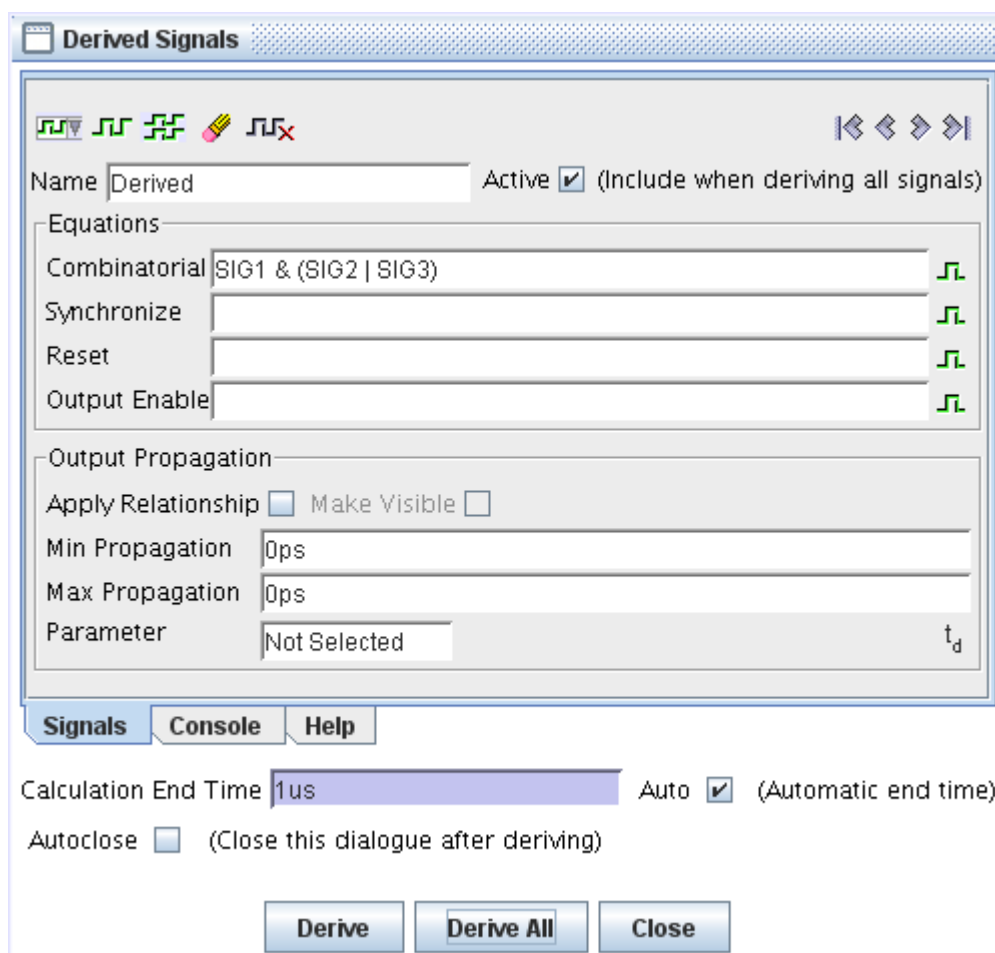
6 Derived Waveforms

TimingTool provides the means to derive waveforms from logical combinations of others.

To specify how a waveform is to be derived:

- use the main menu Tools → Edit Derived Signals
- right click over the left hand signal name of a waveform and select Edit Derived Signals
- double click on the left hand signal name

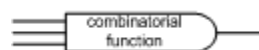
This will pop up the derived waveform parameter panel shown:



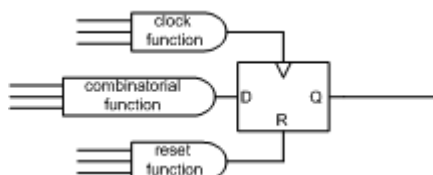
The default tab view on this panel is the configuration for the selected signal. The console tab provides an output for warnings or errors during the calculation of derived waveforms. The help tab provides a brief overview of the derived waveform functionality for quick reference.

6.1 How To Define Signals

In its simplest form, the derived waveform is calculated using an equation relating one or more waveforms.



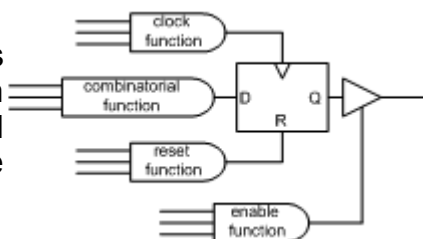
For example “SIG1 & SIG2” will give the result of logically **ANDing** waveforms “SIG1” and “SIG2”.



Synchronised waveforms may be derived using an additional to define a clock. The clock equation may specify a single waveform or may be a logical combination of more than one; a transition from logic 0 to logic 1 will clock the combinational result to the derived waveform output.

To ensure the initial state of your derived signal is predictable, a reset equation may be defined. The reset equation provides an asynchronous input to the calculated waveform; a logic 1 result from the equation will force the derived waveform to logic 0.

Asynchronous and synchronous waveforms may be further manipulated with an asynchronous enable equation. The derived waveform becomes hi-Z when the enable function results in a logic 1.



Derived edges may be optionally delayed from their causal edges by configuring minimum and maximum delay parameters. Further, the derived signal calculator can apply delay relationships to each edge from the causal edge or edges. This permits TimingTool’s timing analyser (see section 7) to make adjustments to the derived signals as the source waveforms or timing parameters are modified.

Note:

- A derived signal calculation is always instigated manually.
- All edges (and any attached relationships) for a derived waveform are automatically deleted prior to re-calculation of the derived signal.

6.2 Equation Syntax

The equations consist of waveform references, logical operators, fixed states and parenthesis, for example:

```
SIG1 & (SIG2 | !SIG3)
```

6.2.1 Waveform reference

Waveforms are identified by their unformatted text name; waveform names containing spaces or reserved characters must be placed within quotes. A helper button is provided to aid insertion of the correct signal name into the equation.

6.2.2 Logical operators

The following symbols represent logical operations:

&	Logical AND
	Logical OR
^	Logical XOR (Exclusive OR)
!	Invert

Note: Where these symbols occur within waveform names, any references to such waveforms must have the names placed within quotes.

E.g. "A&B" or "SIG!".

6.2.3 Fixed States

Fixed states "0", "1", "V", "X" and "Z" may be used within equations.

6.2.4 Parenthesis

Use to change the order in which logical operations are applied. For example:

```
SIG1 & (SIG2 | SIG3)
```

In this equation, "SIG2" and "SIG3" are logically **OR**ed. The result is then **AND**ed with "SIG1".

7 Timing Analysis

TimingTool automatically performs advanced Timing Analysis on your timing diagrams. Timing Analysis calculations are performed on your diagram automatically with every change you make. The timing analysis process traces through all the delays in the diagram and calculates the critical paths; edges derived from delays are repositioned. All measures and timing constraints are then recalculated and violations are highlighted in red on the diagram, timing margins and measures are updated in the parameter tables.

7.1 Delays and Delay Chains

Delay relationships specify the minimum / maximum length of time between source and target edges. The target edge is automatically positioned according to the delay parameters.

A delay is typically used to model propagation delays through the logic gates in a design, when delays are chained the uncertainty of the derived edges increases as you progress down the chain.

Delays are the only relationships that reposition edges.

Details on how to add delays to your diagram are given in Section 4.5.1 New Relationship.

7.1.1 How delays are calculated

Delays can effect the time earliest (te) or time latest (tl) or both of the target edges depending on whether the min or max parameter values have been specified.

In Figure 51 below, the following rules apply:

- td, has both a min and a max value specified and drives both the te and tl of the target edge.
- td₁, has no parameter values set and does not drive the target edge time at all.
- td₂, has only the min parameter value set and drives only the te of the target edge (it is drawn to the te of the target).
- td₃, has only the max parameter set and drives only the tl of the target edge (it is drawn to the tl of the target).

SYMBOL	MIN	MAX
td	20	40
td ₁		
td ₂	50	
td ₃		40

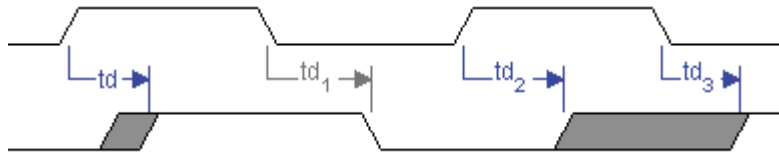


Figure 51 Delay Example

Delay target edges are locked and cannot be moved manually in the diagram.

7.1.2 Delay Chains

Where delays are chained together, the delays propagate as the timing analysis progresses down the chain, see in the example below Figure 52.

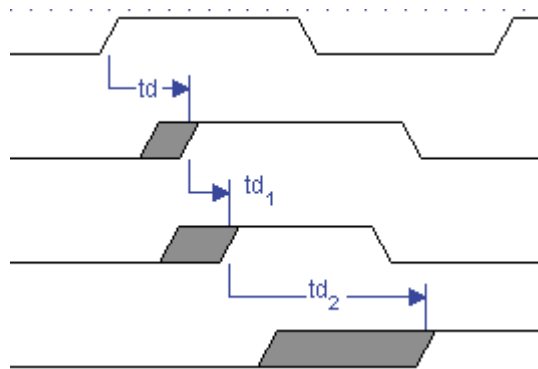


Figure 52 Delay Chain

7.1.3 Delay Loops

If you inadvertently create a delay loop in your timing diagram, the timing analysis will not be able to progress any further until the error is corrected. Delay loops are highlighted in orange on your diagram.

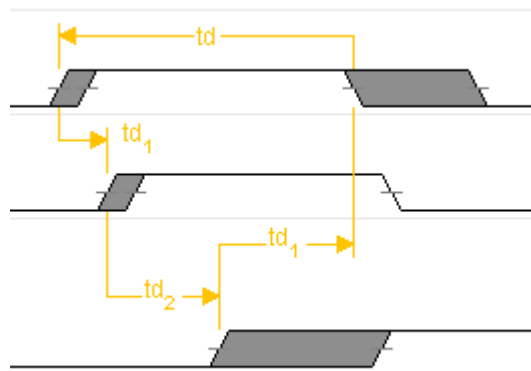


Figure 53 Delay Loop

To correct the error either reposition or delete one of the delays forming the loop.

7.1.4 Delay Target Type

It is possible to have several delays all referring to the same delay target, when this situation occurs the timing analysis engine needs to determine the critical path, how this is done will depend on the target type of the target edge.

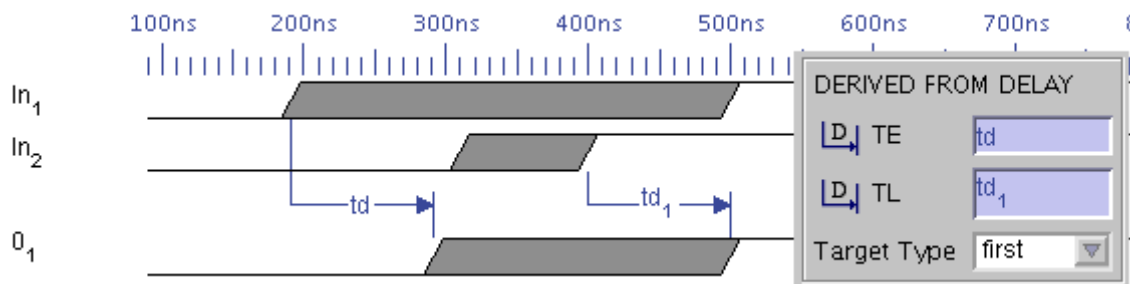
The delay analysis starts by calculating the earliest te and the earliest tl then the latest te and the latest tl of all the delays that drive the target edge.

The critical path for the min and max values of the target edge are then calculated in accordance with the following table:

Target Edge Type	Target te	Target tl
First	Earliest te	Earliest tl
Last	Latest te	Latest tl
Inside	Latest te	Earliest tl
Outside	Earliest te	Latest tl

7.1.4.1 First

When the target edge has the 'first target type rule', the critical path is calculated using the earliest te and earliest tl of all the delays feeding into the target edge.



SYMBOL	MIN	MAX
td	100	100
td ₁	100	100

Figure 54 Delay Target Type (First)

The time earliest of the target edge is derived from (td) and the time latest from (td₁), this information is displayed in the edge attribute panel, when the target edge is selected.

7.1.4.2 Last

When the target edge has the 'last target type rule', the critical path is calculated using the latest te and latest tl of all the delays feeding into the target edge.

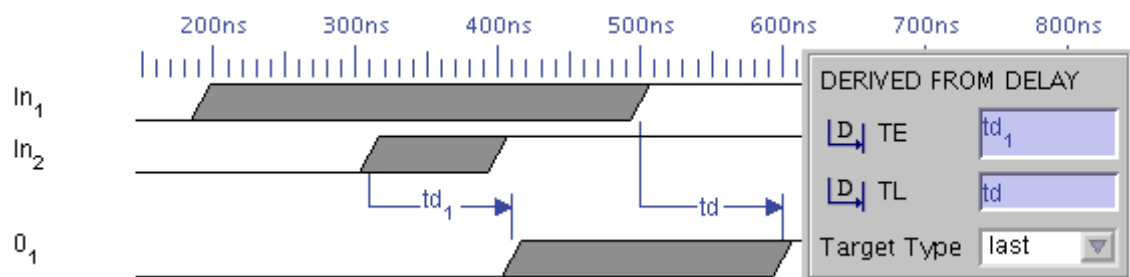


Figure 55 Delay Target Type (Last)

The time earliest of the target edge is derived from (td_1) and the time latest from (td), this information is displayed in the edge attribute panel, when the target edge is selected.

7.1.4.3 Inside

When the target edge has the 'inside target type rule', the critical path is calculated using the latest te and earliest tl of all the delays feeding into the target edge.

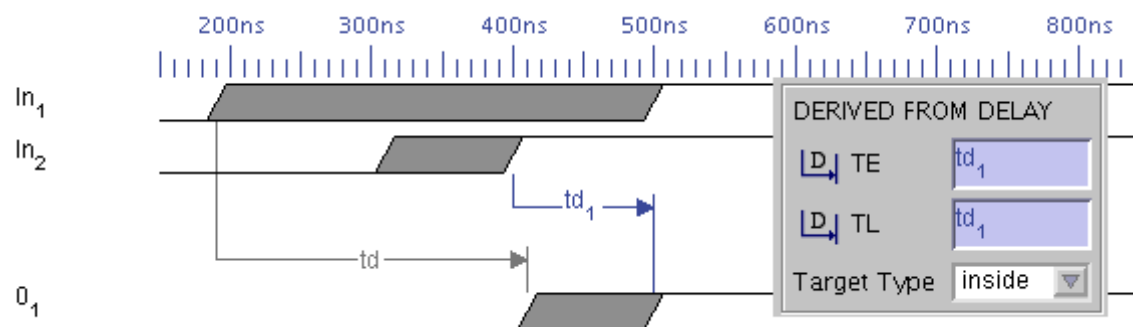


Figure 56 Delay Target Type (Inside)

The time earliest of the target edge is derived from (td_1) and the time latest from (td_1), this information is displayed in the edge attribute panel, when the target edge is selected.

Delays that have no effect on the timing of the target edge (in this case td) are shown in grey.

7.1.4.4 Outside

When the target edge has the 'outside target type rule', the critical path is calculated using the earliest te and latest tl of all the delays feeding into the target edge.

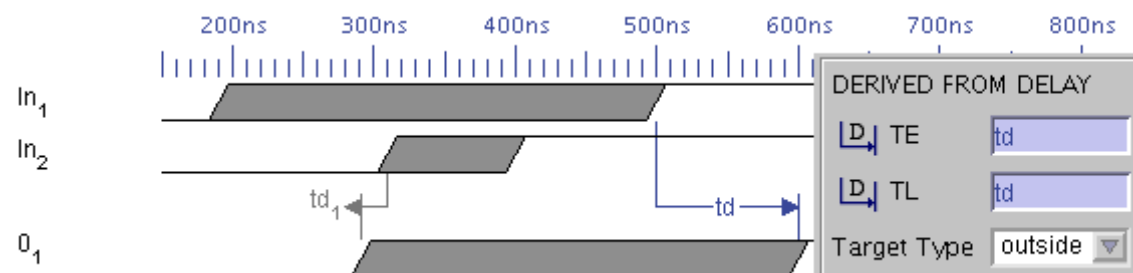


Figure 57 Delay Target Type (Outside)

The time earliest of the target edge is derived from (td) and the time latest from (td), this information is displayed in the edge attribute panel, when the target edge is selected.

Delays that have no effect on the timing of the target edge (in this case td_1) are shown in grey.

7.1.5 Delay examples

The two examples below demonstrate how the first and last target types are used to model the behaviour of an OR and AND logic gate.

7.1.5.1 Using delays to model an OR gate

In this simple example, signal O_1 models a logical OR of signals In_1 and In_2 .

The first input edge on the inputs to go high will drive the output high, so the *first* 'target type rule' is set for the first edge on O_1 .

The last edge on the inputs to go low will drive the output low, so the *last* 'target type rule' is set for the second edge on O_1 .

The propagation delay of the OR gate is set as the min/max in the parameter table for td_1 and td_2 and is shown as uncertainty on the target edge.

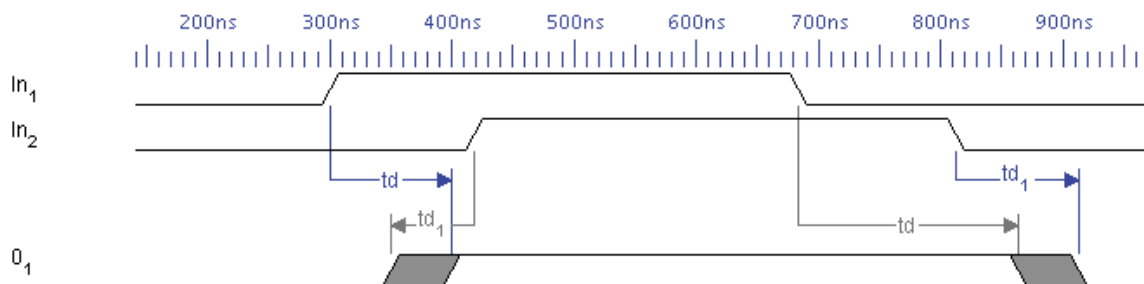


Figure 58 Delay OR gate example

7.1.5.2 Using delays to model an AND gate

In this simple example signal O_1 models a logical AND of signals In_1 and In_2 .

The last input edge on the inputs to go high will drive the output high, so the *last* 'target type rule' is set for the first edge on O_1 .

The first edge to on the inputs to go low will drive the output low, so the *first* 'target type rule' is set for the second edge on O_1 .

The propagation delay of the AND gate is set as the min/max in the parameter table for td_1 and td_2 and is shown as uncertainty on the target edge.

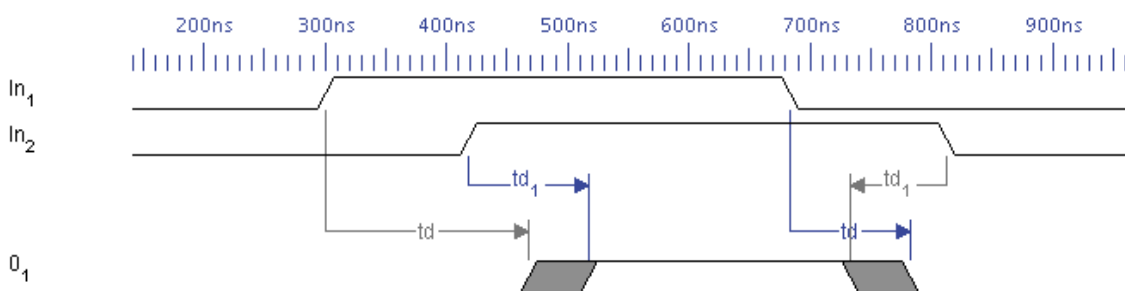


Figure 59 Delay AND gate example

7.2 Setup and Hold Constraints

Constraints are used to automatically check critical timing requirements in the diagram. Violated constraints are highlighted in red, and margins are calculated to show how close a constraint is to being violated.

In TimingTool, there are three types of constraint:

- Set-up, the min/max time that a target edge should be set-up ahead of the source edge.
- Hold, the min/max time that the target edge should be held after the source edge.
- Constraint, a general constraint that the target should occur within the min/max time of the source edge (same as a hold).

Note: Some timing diagram editors do not differentiate between the three types, so that all setup/hold relationships will import into TimingTool as general constraints. This will not effect timing calculations, but you may wish to update the diagram so that setups and holds use the more specific relationship type. If you change an imported constraint to a setup, you should be aware that TimingTool calculates setups backwards in time from the source to the target (ie. the target would normally occur before the source).

7.2.1 Min / Max parameter values

When adding a constraint (see Section: 4.5 Relationships), the associated parameter determines the min and max values permitted by the constraint. The table below shows how a constraint is calculated, where (tm) is the time between the source and that target. Note: (tm) applies backwards in time for a setup relationship.

Min	Max	Timing Analysis
-	-	No calculation performed
10	-	$tm \geq 10$
-	20	$tm \leq 20$
10	20	$10 < tm \leq 20$

7.2.2 Calculated Margins

Margin times are calculated and shown between angled brackets e.g. <5ns,-5ns> would show that the timing is within 5ns either side of the constraint. To view the margin time of a constraint in the timing diagram, select 'Margin' from the drop down menu on the Label area of the Relationship attribute RHS panel.

The parameter table is automatically updated to show calculated margins (you may need to add this column to your saved/default view, see Section: 5.8 Customising the Table Layout). Where more than one relationship is linked to the same parameter, the margins are listed one after another.

Violated margins are shown in red e.g. <-8ns, ->, the violated margin is highlighted in bold, in the example this is the minimum value, it can be seen

from this example that the edge time occurred 8ns less than the minimum value required, see below.

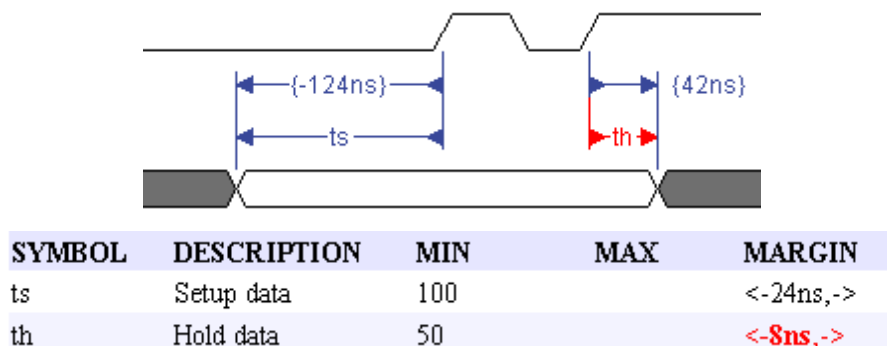


Figure 60 Setup and Hold example

In the example above the setup constraint is met by a 24ns margin, but the hold is violated by a margin of 8ns.

- For t_s , the margin is <-24,->, the target edge is setup 24ns before the required minimum, so the condition is met. Remember setups apply backwards in time, so that a setup target edge should occur a minimum length of time before the source edge.

For t_h , the margin is <-8ns,->, the target edge occurs 8ns before the required minimum and is violated.

7.3 Guarantees

Guarantees specify a time between two edges that is guaranteed to be maintained. An example of a guarantee would be the guaranteed time between two edges derived from a clock pulse by a common delay.

Example:

SYMBOL	DESCRIPTION	MIN	MAX
td_1	Delay	100	100
tg	Guaranteed min/max pulse width	380	420
tm	Measure (without guarantee)	350	450

Figure 61 Guarantee example timing diagram

The measured time between the two derived edges is (350,450).

The clock has a 50ns jitter, this results in each delay-derived edge having an uncertainty of 50ns and a measured gap of 350-450ns. However, the manufacturer's datasheet specifies that the clock pulses are guaranteed to be between 380-420ns, so a guarantee is added to the two delayed edges to indicate that they will be a minimum of 380 and a maximum of 420ns apart.

Guaranteed times are shown between square brackets in the timing diagram e.g. [380,420].

7.4 Measure

Measures are used to measure the time between the source and the target edges. Measures can help you measure time down delay chains, and can be

used to populate the parameter table with useful parameters showing key timings automatically extracted from the diagram.

A measure can be associated with a parameter so that the parameters min and max values are automatically updated as the diagram is changed.

Measure times are shown between curly brackets in the timing diagram, e.g. {350,450}.

The measure in: Figure 61, shows the min and max time between the source and the target edges. The measure in the example is linked to the parameter 'tm', this parameters min and max values will automatically track the associated edges.

The minimum time is calculated from the inside measure and the maximum time from the outside measure.

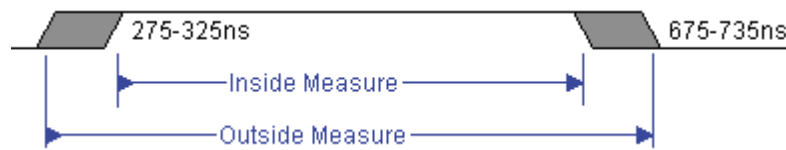


Figure 62 Inside and Outside Measures

8 HDL Testbench Generation

8.1 Import of HDL

If a VHDL or Verilog file is opened from either the local file system or from the online workarea, the HDL file is initially analysed to find entities or modules. If there is more than one entity or module within the file, a popup is generated so that the user can select which entity or module to use for generating the initial TDML. With an entity or module selected, the inputs, outputs, and IO are used to setup an initial TDML file which contains the signal and bus names, the signal and bus directions, and the bus widths. For VHDL the signal and bus types are also recorded within the TDML for later use by the HDL generators. The directions are automatically reversed so that the generated HDL code from TimingTool interfaces correctly to the original HDL source.

An example of the Entity/Module selection dialog is shown in Figure 63.

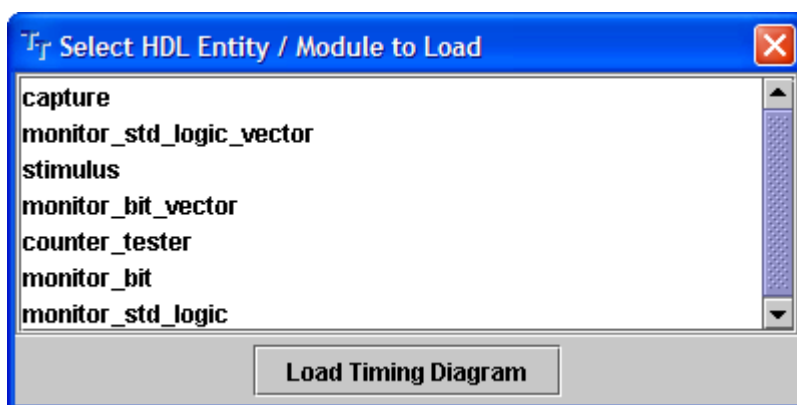


Figure 63 Select Entity / Module

If a TDML file has been created (by saving a timing diagram generated by TimingTool) from a HDL import, the next time the HDL file is imported, a new diagram is not generated and the existing TDML is loaded automatically. This only occurs when both the HDL and the TDML coexist within the same directory and have the same filenames with different extensions. When a new diagram is generated from the HDL, TimingTool generates a warning dialog as shown in Figure 64.

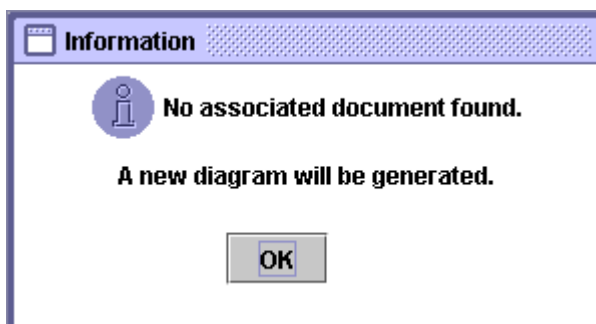


Figure 64 New diagram dialog

If the input HDL source contains generics, constants or defines then these values are placed into the HDL parameter table (viewed from the HDL menu bar).

8.2 Generation and Tolerance

From the menu bar select HDL and then either VHDL from Timing Diagram, Verilog from Timing Diagram, VHDL Top Level, or Verilog Top Level. TimingTool generates two HDL output files. The HDL top Level instantiates the imported original HDL code (if imported by the user) and the HDL code generated by the TimingTool Timing Diagram. The HDL from Timing Diagram generates the stimulus / capture HDL code. The following describes the code generated for HDL Timing Diagram menu items.

If HDL timing diagram is selected a popup window is then generated that allows you to enter a tolerance value and the end of simulation time. This popup dialog is shown in Figure 65.

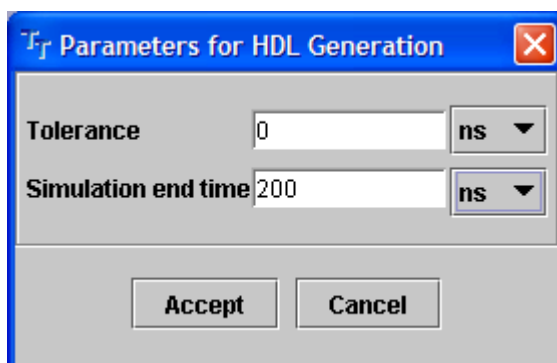


Figure 65 HDL Generate Dialog

The 'Tolerance' applies to VHDL and Verilog; it is used to make an allowance for time differences between RTL and gate-level (with real delay times). The tolerance applies only when the HDL from TimingTool has an input that is being checked against what was drawn in TimingTool (not used for stimulus).

The diagrams below show where the tolerance (set as 4 ns) applies.

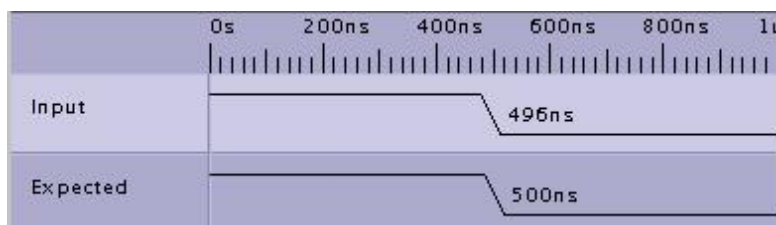


Figure 66 Tolerance Before

If the Input waveform occurs up to 4 ns before the drawn (expected) waveform with a tolerance set to 4 ns, then the HDL simulation will continue and not generate an error.

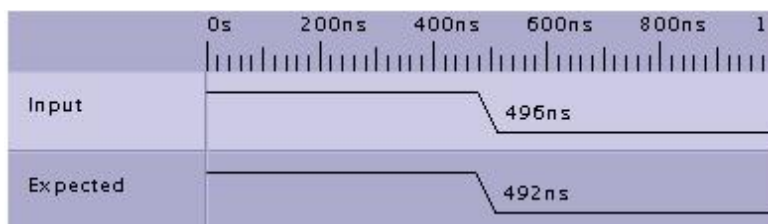


Figure 67 Tolerance After

If the Input waveform occurs up to 4 ns after the drawn (expected) waveform with a tolerance set to 4 ns, then the HDL simulation will continue and not generate an error.

The tolerance value is extended with a time unit (ps, ns, us etc.) for VHDL generation. Verilog uses the time precision value set on the 'timescale directive. The 'timescale precision value is set via the options panel resolution.

8.3 Structure of output HDL

The following diagram shows the structure of the output HDL code generated by the HDL from Timing Diagram menu items. There are 3 main sections to the HDL code (Top_level, Stimulus, and Capture). The top level code connects up the stimulus and capture blocks and also instantiates tri-state buffers for I/O signals. The stimulus block outputs signals, clocks and buses as drawn in the TimingTool diagram. The capture block inputs signals, clocks, and buses and compares them against what is drawn in the TimingTool diagram.

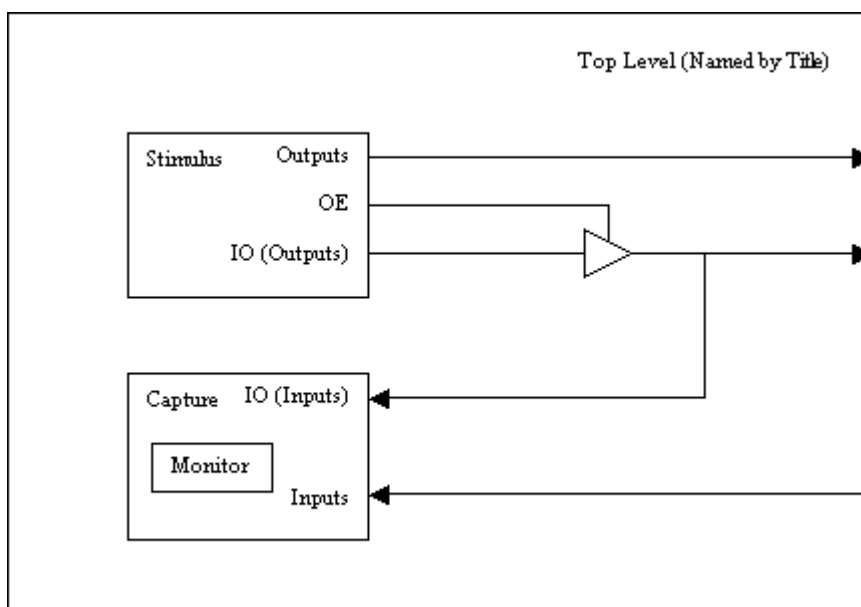


Figure 68 HDL Structure

Inputs in the TimingTool diagram are generated as inputs in the HDL code. Outputs in the TimingTool diagram are generated as outputs in the HDL code. I/O's are handled in the HDL translation with the code swapping between input and output as drawn in the TimingTool diagram.

Connect the TimingTool HDL to other HDL blocks in the design using the code produced by the HDL top menu, compile and simulate.

During simulation TimingTool outputs generate stimulus and TimingTool Inputs are checked against the drawn waveforms. The Monitor block within the capture block is used to compare a drawn input with the actual input signal. See the section on Generation and Tolerance for further information regarding the effect of the Tolerance on the monitor block. If an 'X' is drawn on a signal, or bus in a TimingTool diagram, then the signal or bus is not checked during simulation when the 'X' is present as the expected result. The tolerance is also not applicable when the 'X' is present. When entering or exiting the 'X' state, a check is performed to verify that the simulation input signal matches the drawn input.

When drawing a timing diagram using TimingTool the user should note that any VHDL or Verilog keywords for signals, clocks and buses should not be used. The signal names used with TimingTool are translated directly into signals within the HDL output.

The popup window that is generated when selecting HDL from Timing Diagram also allows the user to enter an "End of Simulation Time". This time is used to stop a simulation and also to check to make sure there has been activity on either the simulation or drawn signals. If there has been no activity over the complete simulation then a warning is generated.

8.4 VHDL Specifics

If VHDL is imported into TimingTool, libraries, generics, and signal types are also captured and stored into the TDML as user defined parameter table TDML tags. All VHDL that is then generated by TimingTool uses the same libraries, generics, and signal types throughout without the need for any type conversion functions. All types must be defined within the libraries specified though and also have overloaded '=' and '/=' functions. Also if a signal or bus is drawn with a state that does not exist for a VHDL type then this will cause an error (e.g. Setting a state value to H on a bit vector type will cause an error). The only special case is the 'X' state, which can be drawn on any input signal (or IO when in input mode) that can have any VHDL type. This is because the 'X' causes the monitor to stop checking and ignore the simulation states while the 'X' is present. The VHDL types can be edited using the entry field provided.

8.5 Verilog Specifics

If Verilog is imported into TimingTool, only parameters and 'defines are stored as user defined parameter table TDML tags. These are used to define bus widths. It is a requirement that all bus widths that have parameter or 'define values as the MSB or LSB should be defined in the HDL parameter table (actually given a value). This is so the Verilog Timing Diagram generator can connect monitors across these buses of variable bus width.

8.6 Saving HDL

Once the HDL has been generated, it can then be saved via a standard file save popup dialog. An example of this popup dialog is shown in Figure 69.

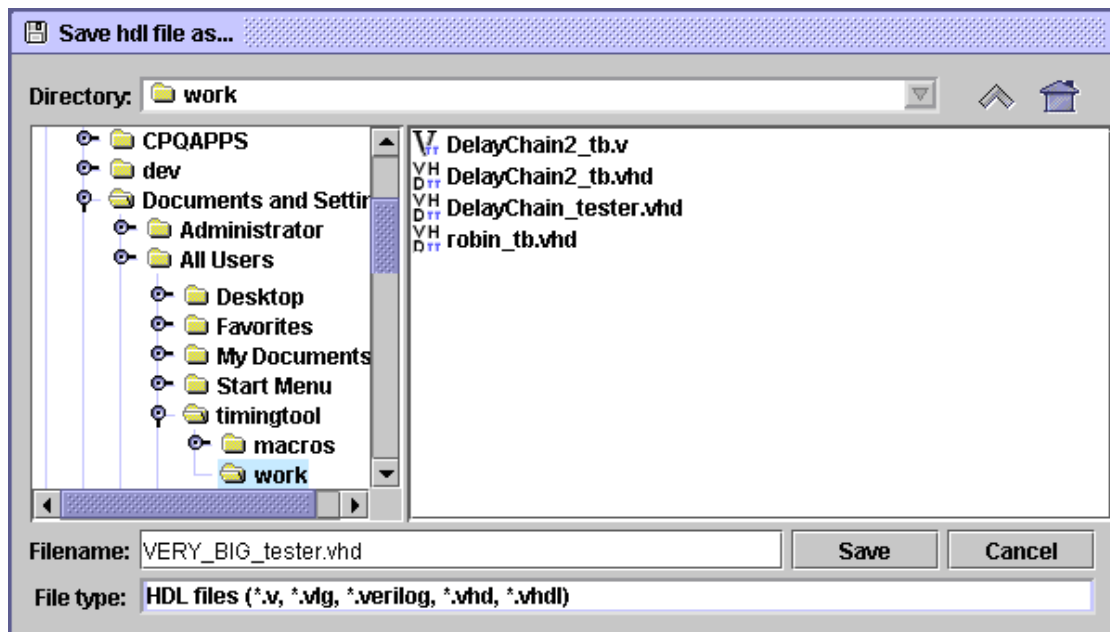
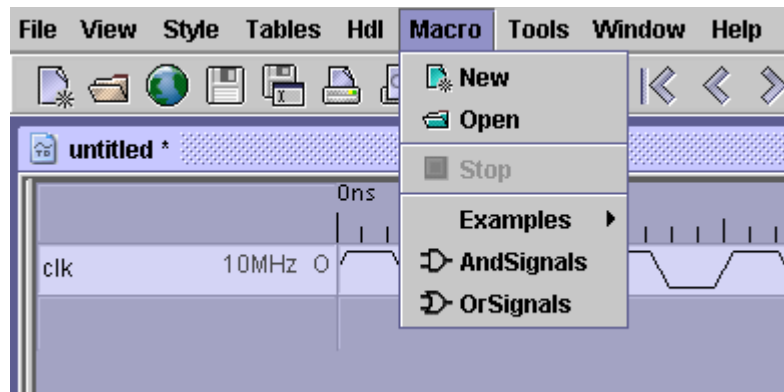


Figure 69 Save HDL popup dialog

9 Macros

9.1 Running Macros

The Macro menu provides a drop-down list of all the macros you have in your personal and system macro directories. Macros provided within your installation are shown on this menu as “Global macros”, macros you have written and placed in your macro directory are shown on this menu as “My macros”. To run an existing macro, simply click on the entry in the drop-down list.



Only one macro may be run at one time. If a macro is left running or gets stuck in a loop, you must press Macro|Stop before running another.

9.2 Writing Macros

The TimingTool editor macro language uses BeanShell, a lightweight scripting language for Java. The macro scripts are therefore written in a loose form of Java.

An interface is provided to enable scripts to control TimingTool enabling access to editor documents and their components e.g. signals and edges. Also included are some helper classes to aid user input and output. These features are provided in the `com.timingtool.ei` package which is automatically imported ready for use.

WARNING:

Java is a powerful language; macro scripts in TimingTool potentially have as much access to your machines files and networks as any other applications. **Only use macro scripts from reliable sources.** TimingTool does not accept any responsibility for any damage caused by running Macro scripts. All macro scripts distributed by TimingTool are examined carefully to reduce the risk of undesirable effects to a minimum.

9.2.1 Introducing BeanShell

The TimingTool editor macro language uses BeanShell, a lightweight scripting language for Java. BeanShell is kindly provided by www.beanshell.org under the LGPL (the *Lesser Gnu Public License*).

BeanShell is rapidly being adopted by modern Java based application as a common, cross-platform and even cross-application technique of facilitating macro/script control.

You do not have to know anything about Java to begin writing your own TimingTool macros. But if you know how to program in Java, you already know how to write BeanShell scripts. This section focuses on using BeanShell in macros within TimingTool. If you are interested in learning more about BeanShell generally, consult the [BeanShell](http://www.beanshell.org) web site. Information on how to run and organize macros, whether included with the TimingTool installation or written by you, can be found within this section.

9.2.2 The Editing Interface

To start writing a new macro select Macro|New from the main menu. To edit an existing macro select Macro|Open which will bring up a file open dialog. Navigate to the macro file you wish to view/edit and select "open".

When starting a new macro script, a dialog appears asking you for a title. The title for a macro script is synonymous with its filename. The filename/title is used to form the menu item should it be displayed on the macro menu. For this reason it is best to keep the name reasonably concise.

9.2.3 Macro Housekeeping

If you have several macros, you may wish to arrange them in groups. This is easily achieved by creating directories for each group and place the macros for each group within these directories. The macro menu will then permit navigation through any such directories (and any further sub-directories) to reach macros within the groups.

If you wish to share certain macros with more than one user, simply store them in the macros directory where TimingTool is installed. Any macros in this directory (or groups of macros) are displayed under the heading "Global macros".

9.2.4 Macro File Format

The format of a TimingTool macro script is simply plain text. You may, if you wish, write lines of code from line 1 using the syntax described in the BeanShell manual. However, when you create a new macro from within TimingTool using Macro|New, a standard header will be automatically inserted. This header is not mandatory but is required any macros submitted to TimingTool for public use. Future releases of TimingTool editor may make use of the fields within this header if available.

9.2.5 The Mandatory First Example

```
Dialog.information("Hello World!");
```

Running this one line script causes TimingTool to display a message with the traditional text.

This statement calls a static method (or function) named “information” in TimingTool's “Dialog” class. If you don't know anything about classes or static methods or Java (or C++, which employs the same concept), you will need to gain some understanding of a few terms. Obviously, this is not the place for academic precision, but if you are entirely new to object-oriented programming, here are a few skeleton ideas to help you.

- An *object* is a collection of data that can be initialised, accessed and manipulated in certain defined ways.
- A *class* is a specification of what data an object contains and what methods can be used to work with the data. A Java application consists of one or more classes written by the programmer that defines the application's behaviour. A BeanShell macro uses these classes, along with built-in classes that are supplied with the Java platform, to define its own behaviour.
- A *subclass* (or child class) is a class which uses (or “inherits”) the data and methods of its parent class along with additions or modifications that alter the subclass's behaviour. Classes are typically organized in hierarchies of parent and child classes to organize program code, to define common behaviour in shared parent class code, and to specify the types of similar behaviour that child classes will perform in their own specific ways.
- A *method* (or function) is a procedure that works with data in a particular object, other data (including other objects) supplied as parameters, or both. Methods typically are applied to a particular object which is an instance of the class to which the method belongs.
- A *static* method differs from other methods in that it does not deal with the data in a particular object but is included within a class for the sake of convenience.

Java has a rich set of classes defined as part of the Java platform. Like all Java applications, TimingTool is organized as a set of classes that are themselves derived from the Java platform's classes. We will refer to Java classes and TimingTool classes to make this distinction.

In this one line script, the static method Dialog.information() has a single parameter because that is the way the method is defined in the Dialog class. You must specify this parameter when you call the function. The parameter, which appears to be quoted text, is a string literal - a sequence of characters of fixed length and content. Behind the scenes, BeanShell and Java take this string literal and use it to create a String object. Normally, if you want to create an object in Java or BeanShell, you must construct the object using the new keyword and a constructor method that is part of the object's class. We'll show an example of that later. However, both Java and BeanShell let you use a string literal anytime a method's parameter calls for a String.

If you are a Java programmer, you might wonder about a few things missing from this one line program. There is no class definition, for example. You can think of a BeanShell script as an implicit definition of a main() method in an anonymous class. That is in fact how BeanShell macros are implemented; the class is derived from a BeanShell class called XThis. If you don't find that helpful, just think of a script as one or more blocks of procedural statements conforming to Java syntax rules. You will also get along fine (for the most part) with C or C++ syntax if you leave out anything to do with pointers or memory management - Java and BeanShell do not have pointers and deal with memory management automatically.

Another missing item from a Java perspective is a package statement. In Java, such a statement is used to bundle together a number of files so that their classes become visible to one another. Packages are not part of BeanShell, and you don't need to know anything about them to write BeanShell macros.

Finally, there are no import statements in this script. In Java, an import statement makes public classes from other packages visible within the file in which the statement occurs without having to specify a fully qualified class name. Without an import statement or a fully qualified name, Java cannot identify most classes using a single name as an identifier.

TimingTool automatically imports a number of commonly-used packages into the namespace of every BeanShell macro script. Because of this, most BeanShell scripts you write will not require import statements.

Java requires import statement to be located at the beginning of a source file. BeanShell allows you to place import statements anywhere in a script, including inside a block of statements. The import statement will cover all names used following the statement in the enclosing block.

If you try to use a class that is not imported without its fully-qualified name, the BeanShell macro interpreter will complain with an error message relating to the offending line of code.

9.2.6 Menu Icons


As a visual aid, it is possible to assign icons to macros. Icons must be 16x16 pixels and have the following name format:

*Macroname*16.gif

Where *Macroname* is the body of the filename (or title) of the macro to which its associated.

For example, a macro named MyMacro.ttm would need an image named MyMacro16.gif.

The image must be placed in the same directory as that of the macro itself.

If such a file exists, the menu builder will automatically use the image. If it doesn't, the default image  will be used instead.

9.2.7 Examples

You will find some example macros packaged with your installation. These macros are stored in the global area i.e. in the macro directory within the installation directory; by default this is

C:\Program Files\timingtool\macros\Examples

in Microsoft Windows or

/usr/local/bin/timingtool/macros/Examples

in Linux. To open an example macro, use the Macro|Open menu item and navigate to the above location. Choose the macro to view/edit and select "open".

Note that the examples are for demonstration purposes only and may not have any other practical purpose.

The Examples directory may be safely deleted if not required.

10 TimingTool Online

10.1 Personal Workspace

TimingTool has the ability to transfer timing information and HDL code over the Internet as the user desires. This can be very useful if a project is being developed across multiple sites / countries.

TimingTool can output an HTML web page showing the timing diagram and parameter tables in a typical data-sheet style format.

Set-up of an online workarea

Firstly, if you have downloaded TimingTool you will have had to set up an account which can be used for file transfers. If extra accounts are required for file transfers it is necessary to go to the TimingTool web site:

<http://www.timingtool.com>

Next click on: [TimingTool Lite](#) followed by [\[New Users\]](#) on the left menu. The new user or company account can then be set-up by simply filling out the online forms. When the new online account is set-up, the username and password can also be used by the web based file browser when using the TimingTool application.

Transfer of information to and from the online work area.

The online file browser allows users to load, and save TDML to / from the online workarea. The workarea is web based which enables users to access files stored in the workarea from any location that has a working internet connection. Figure 70 shows the file browser window:

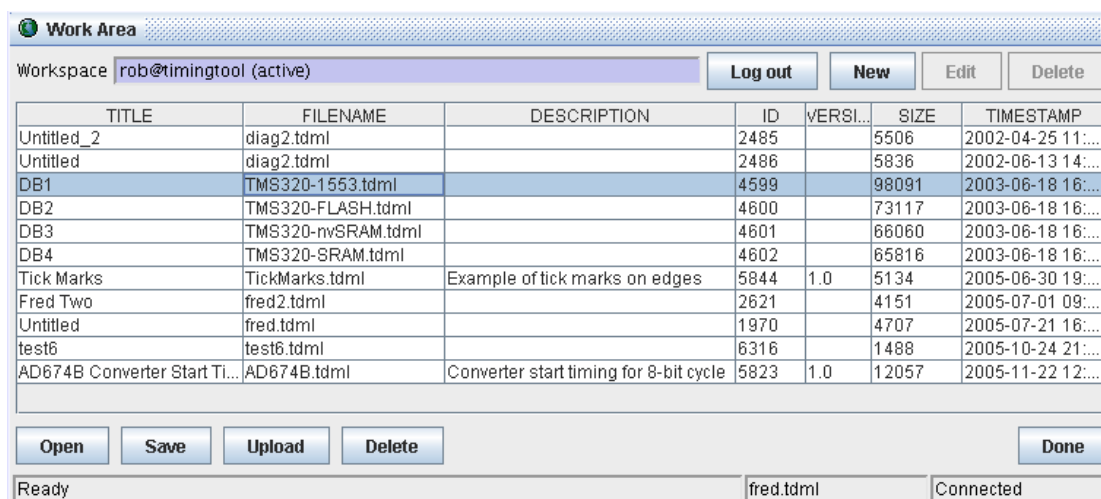


Figure 70 Online workarea

The connection defaults to a secure ssl connection with a direct connection to the Internet. If you have a proxy server you may need to adjust the connection settings found in “Preferences|Proxy” from the main “File” menu (See section 11.4).

10.2 Export to HTML

Timing diagram datasheets in HTML are very easily generated using TimingTool. Refer to section 4.11.3.

10.3 TimingTool Lite

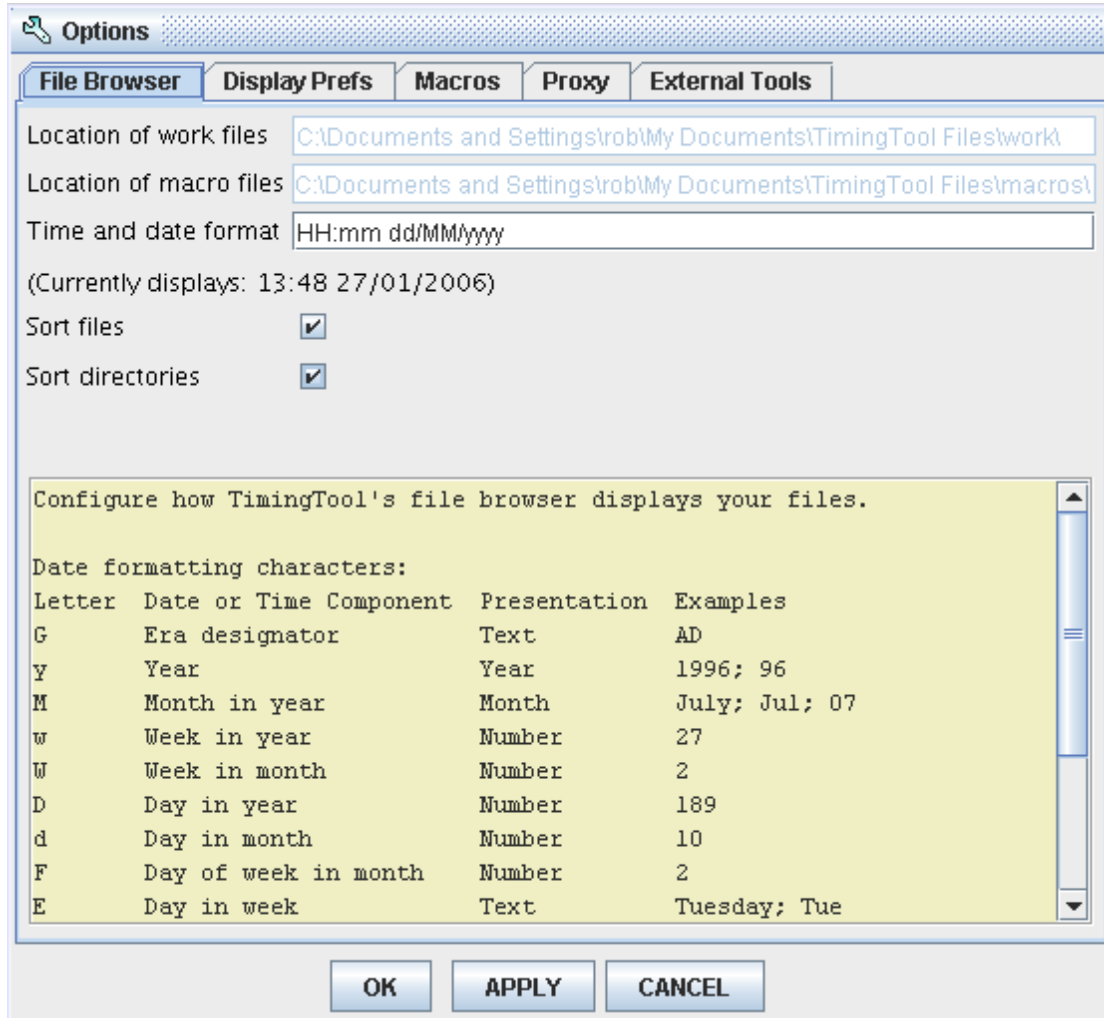
TimingTool Lite is a free to use version of TimingTool that runs as an applet within the user web browser. To use TimingTool Lite go to www.timingtool.com. Note that not all features available to TimingTool Editor are available in the applet version.

Timing diagrams generated in TimingTool can be uploaded and viewed in TimingTool Lite.

11 Preferences

11.1 File Preferences

The file browser preferences tab allows you to define the date format and sort order of files and directories in the TimingTool file browser.



11.2 Display Preferences

The display preferences tab allows you to define the colour of the diagram, labels, annotations, relationships and edges.

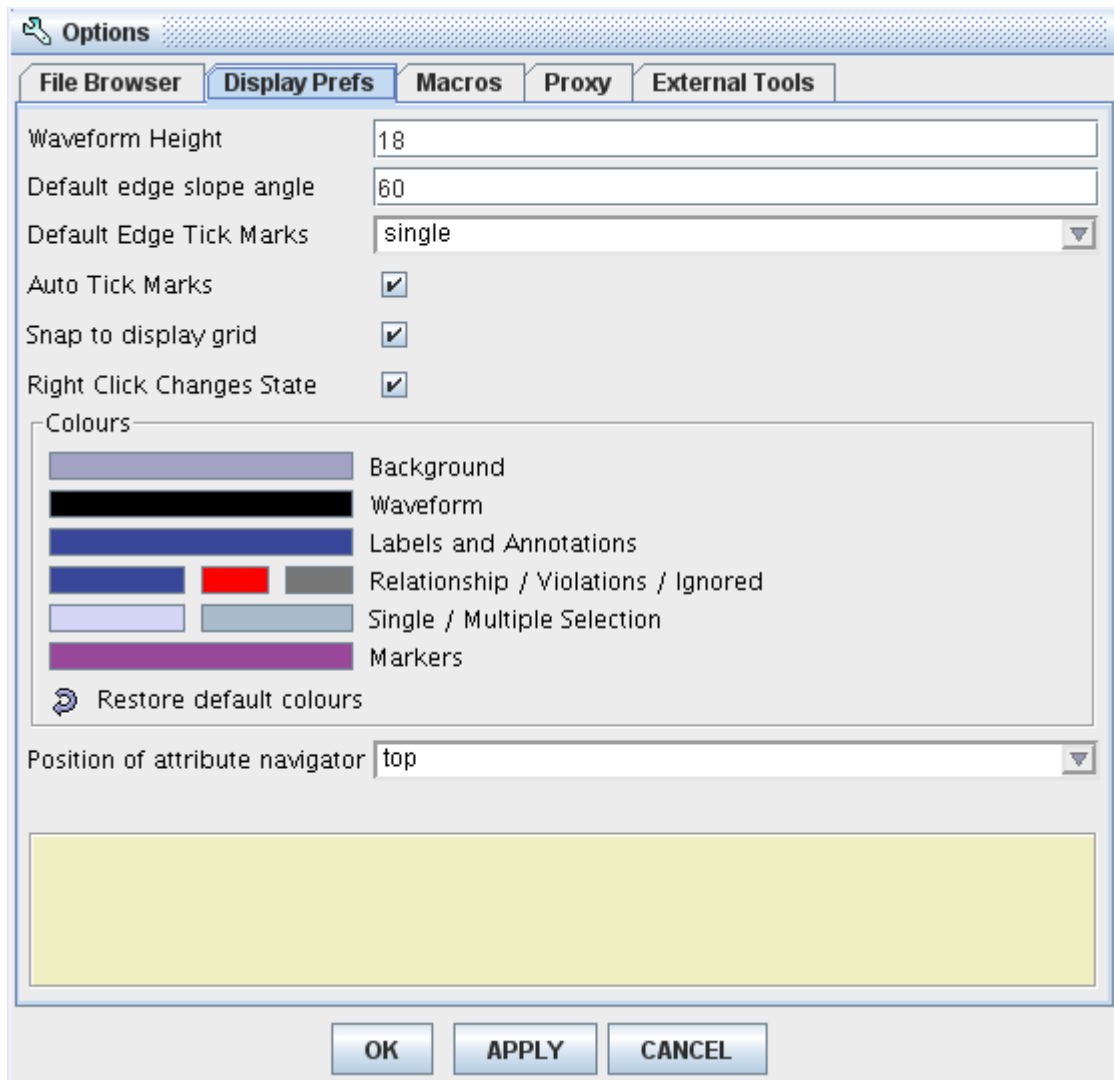
You can also specify the edge slope in degrees, eg. To draw a diagram with straight edges set the edge slope to 90 degrees.

The height of the waveforms can be specified in pixels.

The default edge tick marks settings can also be defined.

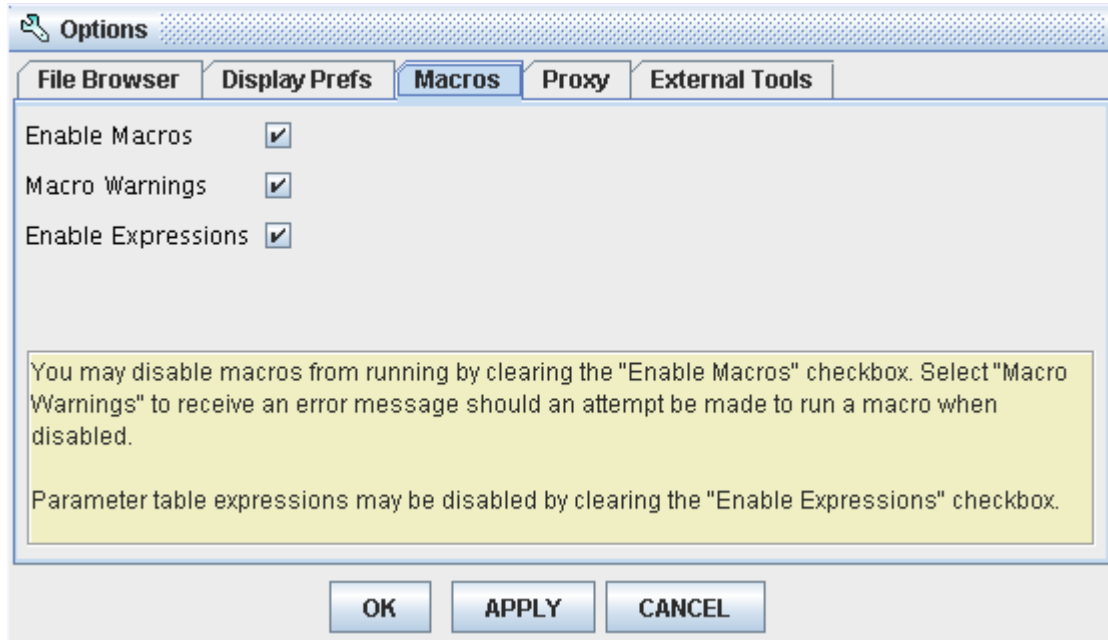
Snap to grid when dragging edges.

Behaviour of the right click over edges and steady states.



11.3 Macros Preferences

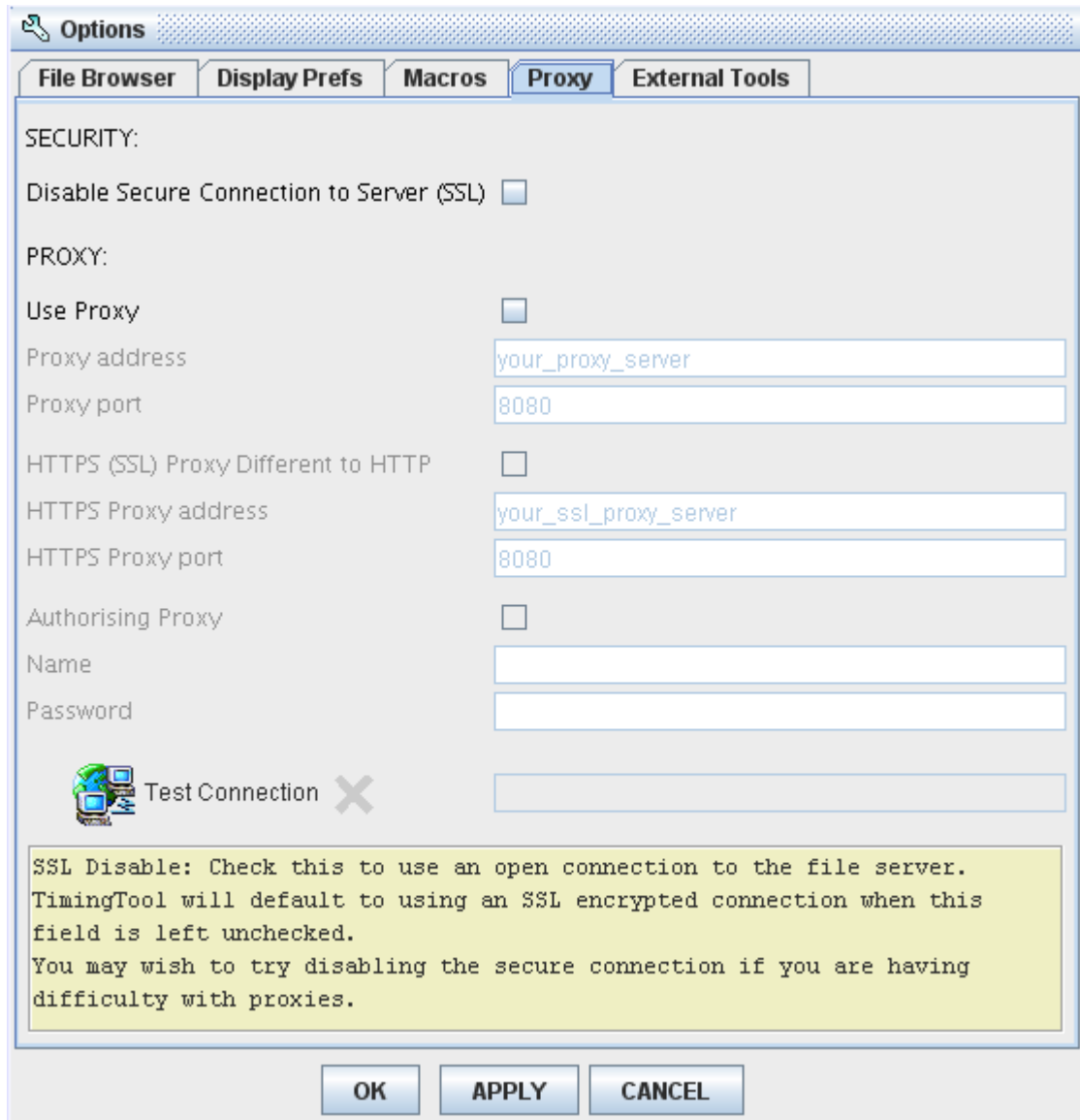
The macros preferences tab allows you to enable or disable the running of macros, and of the expressions defined in the parameter tables.



11.4 Proxy Preferences

The Proxy preferences tab allows you to enter details about your proxy server.

If your Internet connection is via a proxy you will need to configure the proxy settings before you can access the online workarea to load and save diagrams created using TimingTool Lite.



The screenshot shows the 'Options' dialog box with the 'Proxy' tab selected. The dialog has a title bar with a question mark icon and the text 'Options'. Below the title bar are five tabs: 'File Browser', 'Display Prefs', 'Macros', 'Proxy', and 'External Tools'. The 'Proxy' tab is active. The main area contains the following settings:

- SECURITY:**
 - Disable Secure Connection to Server (SSL)
- PROXY:**
 - Use Proxy
 - Proxy address:
 - Proxy port:
 - HTTPS (SSL) Proxy Different to HTTP
 - HTTPS Proxy address:
 - HTTPS Proxy port:
 - Authorising Proxy
 - Name:
 - Password:
 - Test Connection

At the bottom of the dialog is a yellow warning box with the following text:

```
SSL Disable: Check this to use an open connection to the file server.
TimingTool will default to using an SSL encrypted connection when this
field is left unchecked.
You may wish to try disabling the secure connection if you are having
difficulty with proxies.
```

At the bottom of the dialog are three buttons: 'OK', 'APPLY', and 'CANCEL'.

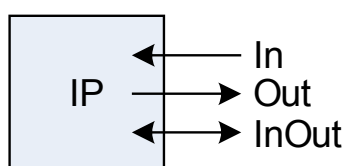
Appendix A – Tutorials

A.1 Drawing Interface Timings

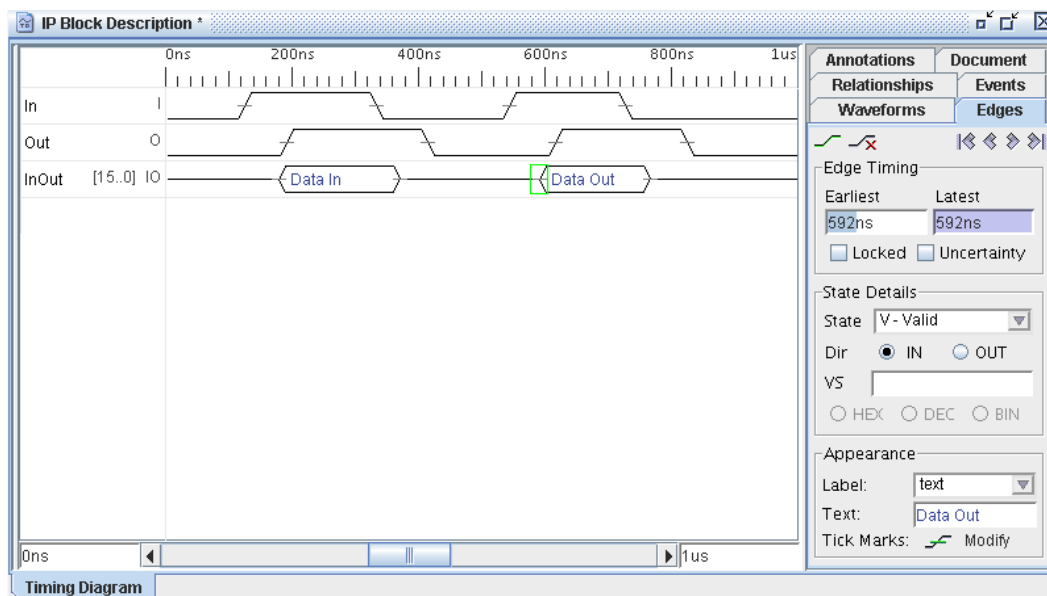
When drawing an interface's timing diagram, the designer should consider whether the timing diagram is describing a block or circuits behaviour, **OR** if the timing diagram is describing external behaviour that is connected to a block or circuit.

A.1.1 Describing a block or circuits timing

The diagram below shows the directions of the signals that the timing diagram will be describing.



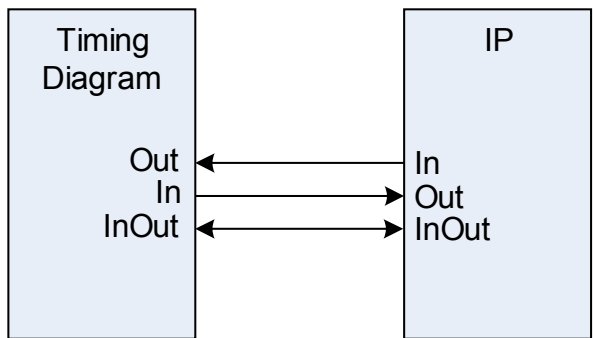
The timing diagram below describes exactly how this block of IP behaves. Note that the directions are set to exactly the same as the pins on the IP block. The InOut has been drawn as a bus (one of the most common types of InOut signal) and for each edge that drives the bus, the direction has been set to either input or output (depending on the direction of the bus e.g. a Read or Write cycle).



When generating HDL from this diagram the signals behave exactly as the IP block.

A.1.2 Describing timings external to a block or circuit

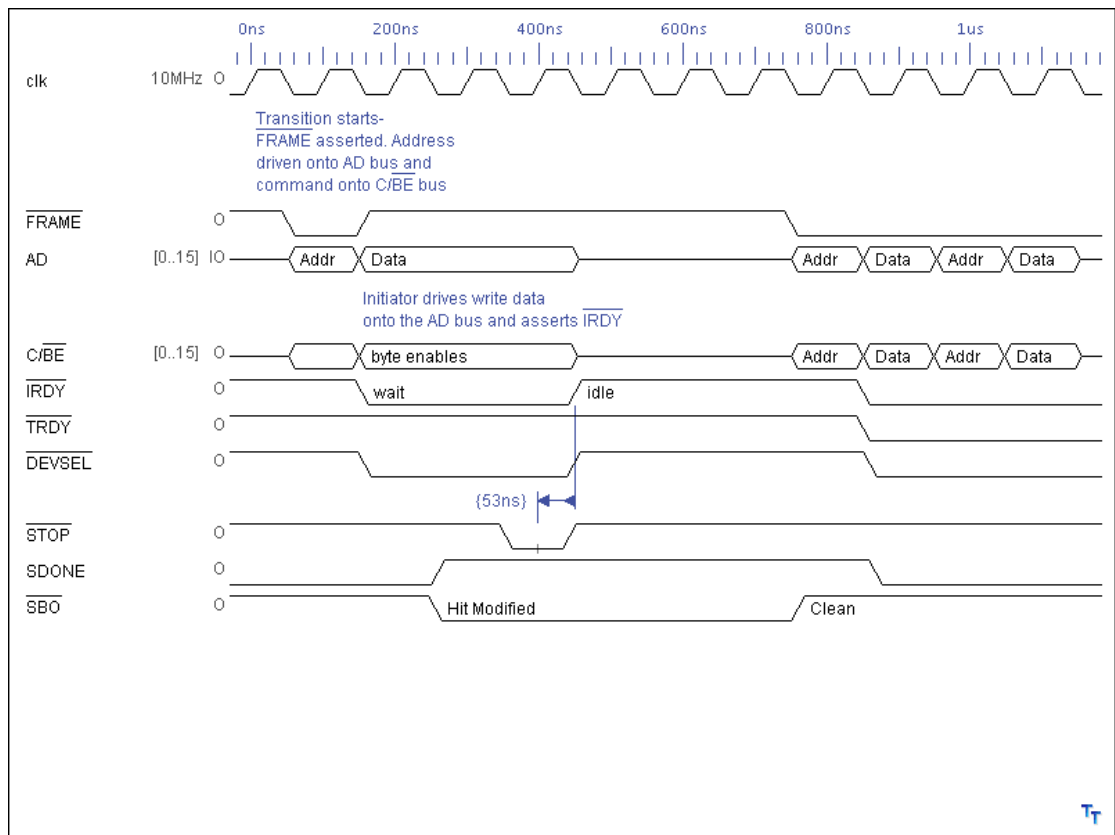
A timing diagram can also be drawn from the perspective of the system (external from the IP block or circuit). The diagram below shows the timing diagram relative to the IP block.



Note that the directions of the timing diagram are reversed. When generating HDL from this timing diagram the HDL can be used as a testbench for the IP block. When importing a HDL IP block into TimingTool, the diagram is set up initially with directions as in this case.

A.2 Modelling a Data Bus

When modelling bus systems, TimingTool is extremely useful. The next example is from a datasheet describing a PCI bus cycle. The diagram has been exported using the TimingTool export diagram feature (see section 4.10).



Appendix B - HDL Reserved Words

B.1 A list of VHDL Keywords is shown below:

Abs	access	after	alias
all	and	architecture	array
assert	attribute	Begin	block
body	buffer	bus	Case
component	configuration	constant	Disconnect
downto	Else	elsif	end
entity	exit	File	for
function	Generate	Generic	guarded
If	in	inout	is
Label	library	Linkage	loop
Map	mod	Nand	new
next	nor	not	null
Of	on	open	or
others	out	Package	port
procedure	process	Range	record
register	rem	report	return
Select	severity	signal	subtype
Then	to	transport	type
Units	until	use	Variable
Wait	when	while	with
Xor			

B.2 A List of Verilog Keywords is shown below:

always	and	assign	attribute
begin	buf	bufif0	bufif1
case	casex	casez	cmos
deassign	default	defparam	disable
edge	else	end	endattribute
endcase	endfunction	endmodule	endprimitive
endspecify	endtable	endtask	event

for	force	forever	fork
function	highz0	highz1	if
ifnone	initial	inout	input
integer	join	medium	module
large	macromodule	nand	negedge
nmos	nor	not	notif0
notif1	or	output	parameter
pmos	posedge	primitive	pull0
pull1	pulldown	pullup	rcmos
real	realtime	reg	release
repeat	rmos	rpmos	rtran
rtranif0	rtranif1	scalared	signed
small	specify	specparam	strength
strong0	strong1	supply0	supply1
table	task	time	tran
tranif0	tranif1	tri	tri0
tri1	triand	trior	trireg
unsigned	vectored	wait	wand
weak0	weak1	while	wire
wor	xnor	xor	

Appendix C – TDML

Overview of the Timing Diagram Markup Language (TDML)

TDML, the Timing Diagram Markup Language is an industry-backed effort to establish an open, industry standard language for the exchange of interactive timing diagrams. The TDML specification enables the exchange of interactive timing diagram information in a standard form. Having a standard form for the information promotes sharing between different organizations and allows interested parties to develop tools for generating, editing, browsing, timing diagrams.

For more information regarding TDML visit:

http://archives.si2.org/si2_publications/tdml/

This site includes documentation, and XML specification (DTD) for the TDML specification.

Appendix D – The BeanShell Macro Language

The TimingTool editor macro language uses BeanShell, a lightweight scripting language for Java. BeanShell is kindly provided by www.beanshell.org under the LGPL (the *Lesser Gnu Public License*).

The Bean Shell user guide is available on the Macro Editor Help Button.

Appendix E – TimingTool Terms and Conditions

TimingTool is the trading name of MOHC Ltd whose head office is at: Nevis House, 14 Iddesleigh Road, Bournemouth, Dorset BH3 7JR

READ THE TERMS OF THIS AGREEMENT CAREFULLY BEFORE PROCEEDING. DO NOT USE THIS SOFTWARE UNTIL YOU HAVE CAREFULLY READ AND AGREED TO THE FOLLOWING TERMS AND CONDITIONS. INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON. IF YOU DO NOT AGREE TO ALL THESE TERMS SELECT "DECLINE" AND PROMPTLY DELETE THE UNUSED SOFTWARE.

License to use: TimingTool grants you a non-exclusive and non-transferable licence for the internal use only of the accompanying software program (which includes any associated media, printed materials and online or electronic documentation) ("Software"). You will not use, copy, modify, rent, sell or transfer the Software or any portion thereof except as provided in this agreement.

You may:

1. install and use one (1) copy of the Software strictly in accordance with any specific terms of the contract of sale ("Sales Order") and with the usage granted by any software authorisation license keys supplied with the Software. For the avoidance of doubt, if the usage of the Software granted by the software authorisation key is more extensive than the authorisations to use the Software set out in the Sales Order, then the provisions of the Sales Order shall prevail;
2. if authorised by the Sales Order to do so, store or install a copy of the Software on a storage device, such as a network server used to install or run the Software on your other computers over an internal network. Unless otherwise stated in the Sales Order, a licence for the Software may not be shared or used concurrently on different computers; and
3. copy the Software only for backup or archival purposes. All restrictions and obligations imposed upon you in relation to the Software shall apply to any backup or archival copies.

Restrictions: You may not:

1. make copies of Software, other than a single copy for archival purposes;
2. modify, decompile or reverse engineer the Software;
3. rent, lease or lend the Software;
4. permit simultaneous use of the Software by more than one user; or
5. transfer the Software to a third party without TimingTool's prior written consent. Any transfer or purported transfer of the Software will terminate your right to use the Software.

Ownership and Copyright of Software: Title to the Software, and all associated intellectual property rights, remains with TimingTool. No right, title or interest in or to any trademark, logo or trade name of TimingTool is granted under this licence. The Software is copyrighted and is protected by UK copyright laws and international treaty provisions. You will not remove copyright notices from the Software. You agree to prevent any un-authorized copying of the Software.

TimingTool may make changes to the Software, or to items referenced therein, at any time without notice, but is not (unless otherwise agreed in writing) obligated to support or update the

Software. Without prejudice to the foregoing, TimingTool may provide you with support services related to the Software such as but not limited to a Maintenance License (Support Services). Use of the Support Services is governed by TimingTool's maintenance agreements from time to time in force. Any supplemental code provided to you as part of the Support Services shall be considered part of the Software and subject to the terms and conditions of this Agreement.

Limited Media Warranty: TimingTool warrants that the media (excluding electronic transfer) on which the Software is furnished will be free from defects in material and workmanship for a period of one (1) year from the date of purchase. Upon return of such defective media, TimingTool's entire liability, and your exclusive remedy, shall be the replacement of the media.

Limited Warranty: TimingTool warrants that:

1. the Software will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt; and
2. any Support Services provided by TimingTool shall be substantially as described in applicable written materials provided to you by TimingTool, and TimingTool's support engineers will make commercially reasonable efforts to solve any problem issues. Local law or jurisdiction may not allow limitations on duration of an implied warranty, so the above limitation may not apply to you. To the extent allowed by applicable law, implied warranties on the Software, if any, are limited to ninety (90) days.

Disclaimer of Warranty. UNLESS SPECIFIED IN THIS AGREEMENT, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT THESE DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Limitation of Liability. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL TimingTool BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF TimingTool HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. In no event will TimingTool's liability to you, whether in contract, tort (including negligence), or otherwise exceed the amount paid by you for Software under this Agreement. This will be zero in the case of Software supplied for evaluation purposes only or in the case of the Software marked as Beta Test Version. Any attempt to use a Beta Test Version for commercial purposes will be entirely at the users risk. Software for The foregoing limitations will apply even if the above stated warranty fails of its essential purpose. In any case, TimingTool's entire liability under any provision of this Agreement shall be limited to the greater of the amount actually paid by you for the Software or 1.00UKP (one UK Pound) provided however, if you have entered into a TimingTool maintenance agreement, TimingTool's entire liability regarding Support Services shall be governed by the terms of that agreement. Local law or jurisdiction may not allow the exclusion or limitation of liability, so the above limitation may not apply to you.

Audit: TimingTool reserves the right to have audits conducted to verify your compliance with this Agreement.

Termination: This agreement is effective until terminated. You may terminate this agreement at any time by destroying all copies of the Software.

Severability: The terms and conditions stated in this Agreement are declared to be severable. If any paragraph, provision, or clause in this Agreement shall be found or be held to be invalid or unenforceable in any jurisdiction in which this Agreement is being performed, the remainder of this Agreement shall be valid and enforceable and the parties shall use good faith to negotiate a substitute, valid, and enforceable provision which most nearly effects the parties' intent in entering into this Agreement.

Appendix F – Third Party Contributors

JEdit Syntax Highlighting Package

<http://syntax.jedit.org>

Provides syntax highlighting for the TDML (view source) window and for generated VHDL and Verilog.

Distributed under the MIT open source copyright allowing unrestricted integration into commercial software.

License

<http://www.opensource.org/licenses/mit-license.php>

The MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Copyright

The jEdit 2.2.1 syntax highlighting package contains code that is

Copyright 1998-1999 Slava Pestov, Artur Biesiadowski, Clancy Malcolm,
Jonathan Revusky, Juha Lindfors and Mike Dillon.

-- Slava Pestov

25 September 2000

<sp@gjt.org>

Batik

Batik is a Java(tm) technology based toolkit for applications or applets that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as viewing, generation or manipulation.

<http://xml.apache.org/batik>

TimingTool uses the Batik library to provide scaleable vector graphics SVG output.

Apache Batik Public License

=====
The Apache Software License, Version 1.1
=====

Copyright (C) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Batik" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact
apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.