sphinxthebibliography sphinxtheindex

fancyhf

c@chapter

waveform_draw Documentation

AVN

Oct 11, 2018

CONTENTS:

CHAPTER

ONE

Intro

1.1 Motivation

Provide a simple scripted interface to produce timing diagrams to facilitate an accurate concise representation of logical design intent.

1.1.1 Species spotted in the wild

Table 1: Comparisons of similar tools

Tool name	License	Notes:
TimingDesigner	yes	• Stood the test of time. • Good for datasheet.
TimingEditor	No	• GUI based
TimingGen	No	• GUI based
TimingAnalyzer	No*	• (Java + Jython scripts) • Remarkably Like TimingDesigner • in beta, opensourced
WaveDrom	No	• browser based • good for simple diagrams.
tikz-timing	No	• Latex Bundle(steep learning), Error prone • extensible with well rendered output
visio	yes	• Templated vector editor • Time consuming, high maintenance overhead • Variable Quality of outputs
word/ppt/excel	yes	• diagram mode based on visio
vector editors	yes/no	• inkscape/omnigraffle etc.

1.1.2 Why another method?

Most tools emulate the familiar look feel and functionality of timing designer. They are great for producing data sheets or capturing characterized I/O timing, However they some what lack when it comes to capturing and communication logic design intend. Advanced features offered by TimingDesigner seem excessive for pure clock referenced digital design tasks were the meaning is captured best by way of dependencies, flow graphs and annotations. WaveDrom comes close to this objective with a simple clean intuitive method to produce consistent renders. However it soon hits the limit for more complicated waveforms

1.1.3 How is waveform_draw different?

Waveform_draw started with the objective of drawing consistent timing diagrams from a seemingly crude, but quick mark-up description of the logic function in a manner similar to a value change dump. Under the hood it invokes Latex with appropriate libraries, and tries to conceal the required heavy latex mark-up from the user. The diagram is defined with simple intuitive mark-ups capable of abstracting information typically in a timing diagram, with an easy template, allowing diagrams to be rendered from a tabular form in Excel. The original source xlsx is a .zip of several xml files which are interpreted by excel. So ideally this unzipped version can be maintained under version control thus avoiding the need for versionaing a binary file. There might even be automation available within a versioning system suchas Git which could make this process transparent.

1.1.4 Why use Excel or a similar utility?

Using such an application provides a simple tabular interface which can often be intuitive for waveform entry. Further, simple formulas and rudimentary intelligence provided by *office can be used advantageously to generate a consistent maintainable and reproducible waveform. A drawing application can be used, but the main issue with those is maintainability, as well as effort required. Waveform_draw provides a work flow with human readable intermediate files at many stages. Python and TeX should provide easy extensibility to the basic feature set:

```
.xlsx ---> .csv ---> .tex ---> pdf,svg,png
      (py)       (py)      (TeX)
```

As an example, the Excel step can be overstepped and a csv generated directly, for example by parsing a VCD. This can then be back annotated if required. At the moment this flow is still conceptual but python/tcl/perl scripts could be deployed to parse plain vcds and provide the required translation. The method still would not be straightforward as vcd's capture value change by time units, not active clock edges.

In terms of the Dreaded USP

- provides a consitent less fiddly method for annotation.
- build on what was lying around in the shed.
- a method to deploy tikz-timing within TeX.
- nothing proprietary. Implent/improve/enhance some scripting and modify to your hearts content. It was all done in about 2 weeks. The documentation dragged along for months!

CHAPTER

TWO

Getting Started

2.1 Pre requisites

2.1.1 Excel

- Some version of Excel capable of saving an XLSX file recommended
- Open Office and Libre Office should also work but not exhaustively tested.
- Recommended to start from the diagramming template provided within the package

2.1.2 python 2,7

- Current implementation is in python 2.7 , This will be migrated to 3 sometime in the future.
- python requires the following packages
  - xls2csv
  - openpyxl

```
$ python -c 'import pkgutil; print(1 if pkgutil.find_loader("xls2csv") else 0)'
1
$ python -c 'import pkgutil; print(1 if pkgutil.find_loader("openpyxl") else 0)'
1
```

2.1.3 Latex

- TexLive or MikTex for latex. These are usually found bundled in linux distributions. MikTex can be installed on windows/Mac etc.
  - texlive/2016 is known good

2.1.4 Get the scripts

Clone from

```
$ git clone /work/scratch3/nairajay/waveform_draw
```

Snap shot of directory:

```
7867  2018-04-11 17:08 README
2743  2018-04-09 17:01 read_xlsx_val.py
26677 2018-04-10 13:28 draw_wave_tex.py
59733 2018-04-10 14:19 waveforms_template.pdf
27277 2018-04-10 12:57 waveforms__template.xlsx
6071  2018-04-10 14:21 run.sh
```

Versions

v1.1: Changes to the intermediate CSV to use ';' instead of '.' as the delimiter. This has the knock on effect that the Annotation type can no longer have B|C:c:r This was anyway not used, so changed to B|C:r where B,C still works as chained i or baseline mode and :r selects the annotation colour. :r is the default and cannot be over overridden, it is not Christmas yet!.

2.2 Putting it to work

- Usage with the wrapper script
- Understanding Common Errors
- MISC set-up info that might be useful

tcsh command line:

```
./run.sh -wb *<workbook.xlsx>* [ -ws *<sheet_name>* | -all | -active ] -disp
```

- -wb : workbook file name with xlsx extension
  - -ws : Name of the sheet within the worksheet
  - -all: All available sheets within the specified workbook
  - -active : The active worksheet, which is the worksheet in focus when the file was saved.
- -disp : display the rendered output with xpdf

1. Generate a waveform description in the XL file by filling in clocks and signals. The description is similar to a value change dump where only a change/transition needs to be recorded

   Details of this can be found embedded in the example *waveform_template.xlsx* file. This file has 3 sheets, a file can have as many sheets as required. Each sheet can be converted individually or all sheets can be converted in batch mode as described below. Sheets can be exempt by adding the suffix '_nt' to its name. If not, when the sheet is not empty a conversion will be attempted. Failing to comply with the template some random Tex error will be generated. Please note tex errors are difficult to debug.

   When converted in batch mode, by **-all** an option step will collate all the individual pages into one pdf file which is named <workbook_name>.pdf

   The recommended approach to start a new waveform is by making a copy of the template sheet by right clicking on the tab and choosing copy. Once the copy is made, and suitably renamed, the contents of the cells in the waveform area, NOTES. CLK_MARKS, ANNOTATE can all be cleared (select rows and delete). The waveforms_template sheet can additionally be renamed with _nt suffix so that it is never converted but available as reference. This sheet is protected with no password. The recommended methods for efficiently creating a waveform is detailed as cell comments.

   Both Microsoft Excel and LibreOfficei can generate a compatible xlsx file.

2. Save this file. **Saving is important** as XL will generate values from formulas. Also the sheet that is in focus at the time of save will become the active sheet.

Note:

Q: Can the file be Directly saved to the H drive?

The file can be saved directly to the H drive, However, sometimes Excel would report the file to be read only and refuse to save. In such instances, Save with a different name and save as with the old name rectifies the problem

---

1. For batch conversions of all sheets from a workbook use

```
./run.sh -wb waveforms_all.xlsx -all
```

To convert all sheets that are non empty. A sheet is empty if it has no valid cell. A sheet might unintentionally be classified as non empty, in such case delete the sheet or add _nt' to the end of the sheet name. This can be useful with sheets used to capture additional info.

1. Rendering only the active sheets are useful in closing the description-render cycle

```
./run.sh -wb waveforms_all.xlsx -active -disp
```

or explicitly specifying the -ws sheet_name

```
./run.sh -wb waveforms_all.xlsx -ws <sheet_name> -disp
```

Use **-disp**, to open the rendered result. -disp can also be avoided, but a previously opened pdf reloaded. However, if the pdf is open from windows, tex will generate an ERROR. Please Refer ERROR section.

2.3 Common ERRORS:

1. Nature of Error when the CLK_MARKS section is enabled but no clock is defined i.e the clock column is '0' or empty. Ideally this should be the exact copy of the clock for which the timing cycles are to be drawn, reference in the cell as =<cell_containing_the_name_of_the_clk>.

```
Traceback (most recent call last):
  File "./draw_wave_tex.py", line 565, in <module>
    tex_blk_drawedges = draw_edge_lines(signal_array, clock_edges,clk_filter, indent_
↪level, marked_edges, tex_blk_drawedges)
  ...
  ...
sre_constants.error: nothing to repeat
ERROR: waveforms_template.tex convesion failed
```

1. Error when the pdf is open by another application, normally from windows.

```
ERROR:!I can't write on file `waveforms_template.pdf'.
       (Press Enter to retry, or Control-D to exit; default file extension is `.pdf')
       Please type another file name for output
       ! Emergency stop.
```

1. Nature of the error when '...' get replaced with the Unicode equivalent.

```
Traceback (most recent call last):
  File "read_xlsx_val.py", line 68, in <module>
    result = convert_to_csv(ws_active)
  File "read_xlsx_val.py", line 24, in convert_to_csv
    csv_f.writerow([cell.value for cell in row])
UnicodeEncodeError: 'ascii' codec can't encode character u'\u2026' in position 6:␣
↪ordinal not in range(128)
```

2.4 MISC Notes

Script uses the following packages

- xls2csv
- openpyxl

Note: There is a specific version check for python, at the moment this is hardcoded to 2.7.10, you may override this in the script.

The following is needed for xlstocsv conversion from command line

```
mkdir -p /home/nairajay/local/lib/python2.6/site-packages/
```

The required packages for python may not be available on the host or a managed system. Python allows mechanisms to install them locally. Creating virtual env is another option. With both pip available and access to the outside world, pip_install –user should suffice for majority of the cases. This should default to ~/.local/ and python would search this path by default.

```
mkdir -p /home/nairajay/local/lib/python2.6/site-packages/

pip_install --user <package>
```

A messy way is to use easy_install or using the set-up.py from a tarball. Both these can lead to problems.

```
echo $PYTHONPATH

# append if not empty
# Note: python version specific
setenv PYTHONPATH /home/<user>/local/lib/<python_version>/site-packages
# run once
easy_install --prefix=$HOME/local xlsx2csv
```

Example commands

```
module use /opt/ipython/modulefiles
module load ipyhton

module load texlive/2016
# sometimes it might complain about the tikz-timing library, just use what is
# available, Seem to work

python ./draw_wave_tex.py waveforms_cancel_sane.csv waveforms_cancel_sane.tex
pdflatex waveforms_cancel_sane.tex
pdflatex -interaction=nonstopmode waveforms_cancel_sane.tex
```

```
inkscape -z -f waveforms_cancel_sane.pdf -l waveforms_cancel_sane.svg

# Push button script
./run.sh -wb waveforms_all.xlsx -all -disp
-disp : open xpdf after every render
-all  : process all non empty sheets in xlsx, A sheet is considerd non empty if␣
↪atleast one cell has a value.
        Check for validity of a sheet for parsing to produce waveforms is not␣
↪considered.
-active: render only the active sheet. Along with display can be used for development.
<-ws sheet_name> : provide the explicit sheet name.

svg: By default svg and png are generated. scg's are generally large files and hence␣
↪the default dfeature will be turned off in the future.
```

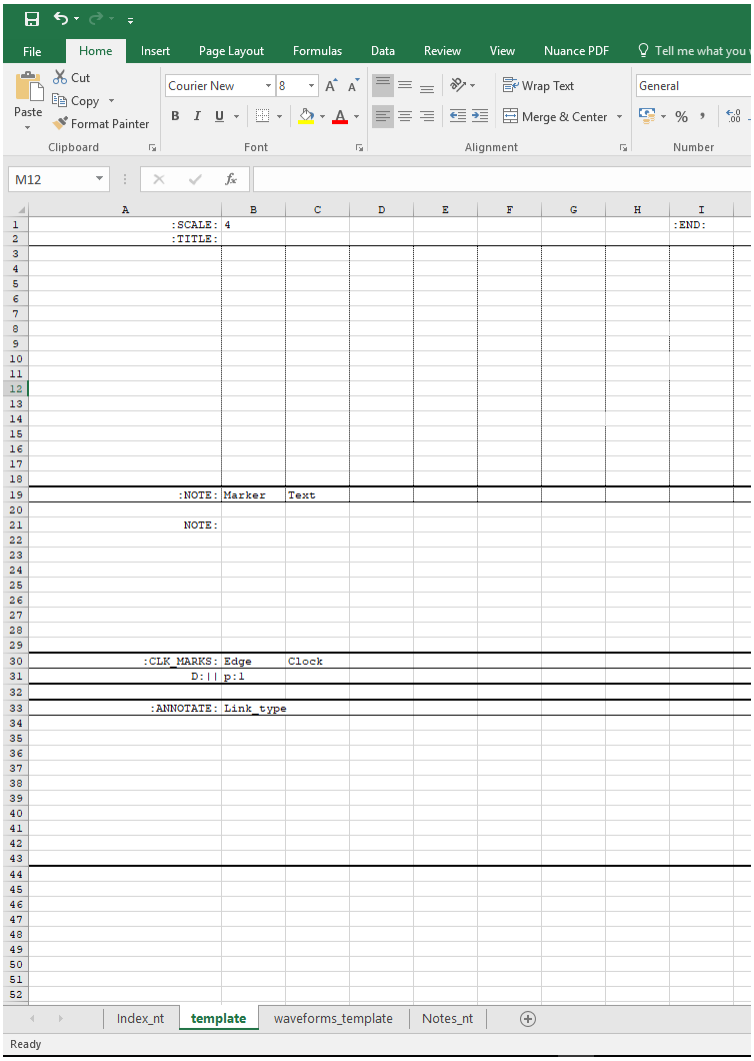


Fig. 1: The starting excel template.
The cell boundaries allow easy segmentation of various areas. This empty template will not compile or generate a waveform.

CHAPTER

THREE

Template and Markup

A worked out example with a recommended flow and other quirks

3.1 Organising the .xlsx file

An .xlsx file may contain more than one sheet. The sheets are to be uniquely named.

Tip: **.xlsx** files is a zip archive which include xml descriptions for the various sheets and some other anciliary information. Although they can be placed under version control, zips are binary files and hence may not be the best way to feed git. An unzipped folder might be better suited for version control

---

In addition, we encourage the user to add two sheets named index_nt and Notes_nt. These will not be rendered but is a good mechanism to capture information regarding the contents of the workbook as well as allow easy navigation. Navigating between the sheets can be done with the sheets tab, but maintaining this info within index and providing hyperlinks to sheets with a short descriptive summary is highly recommended as a method of navigation.

3.1.1 Sheet naming convention

- index_nt : for the cover sheet.
- notes_nt : For the end cover sheet, with any additional info

---

Note: the python parser script will look for the specific character seq **_nt** to ignore sheets that are not to be rendered.

- New sheets can be created by making a copy of template. However, ensure the sheet name is changed and the number removed.
- Keep names short, use the index instead to capture information regarding content.
- Sheet names may or may not be descriptive. For example when using the index as a navigation device it is perfectly acceptable to names the sheets as say set1, set2 etc.

Note: Each sheet will be rendered to a pdf, svg or png file with its **sheet name**. When used with **-all**, in addition to sheets being rendered individually they will also be bundeled into a single pdf with its name as the **workbook** name.

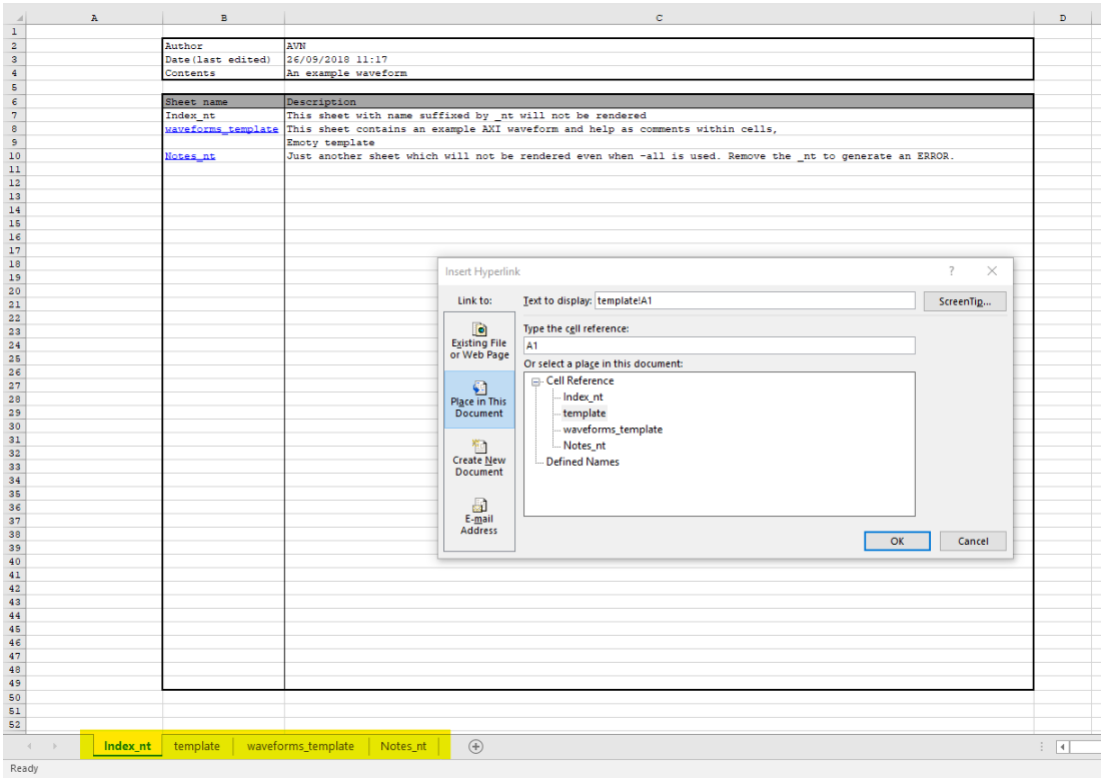
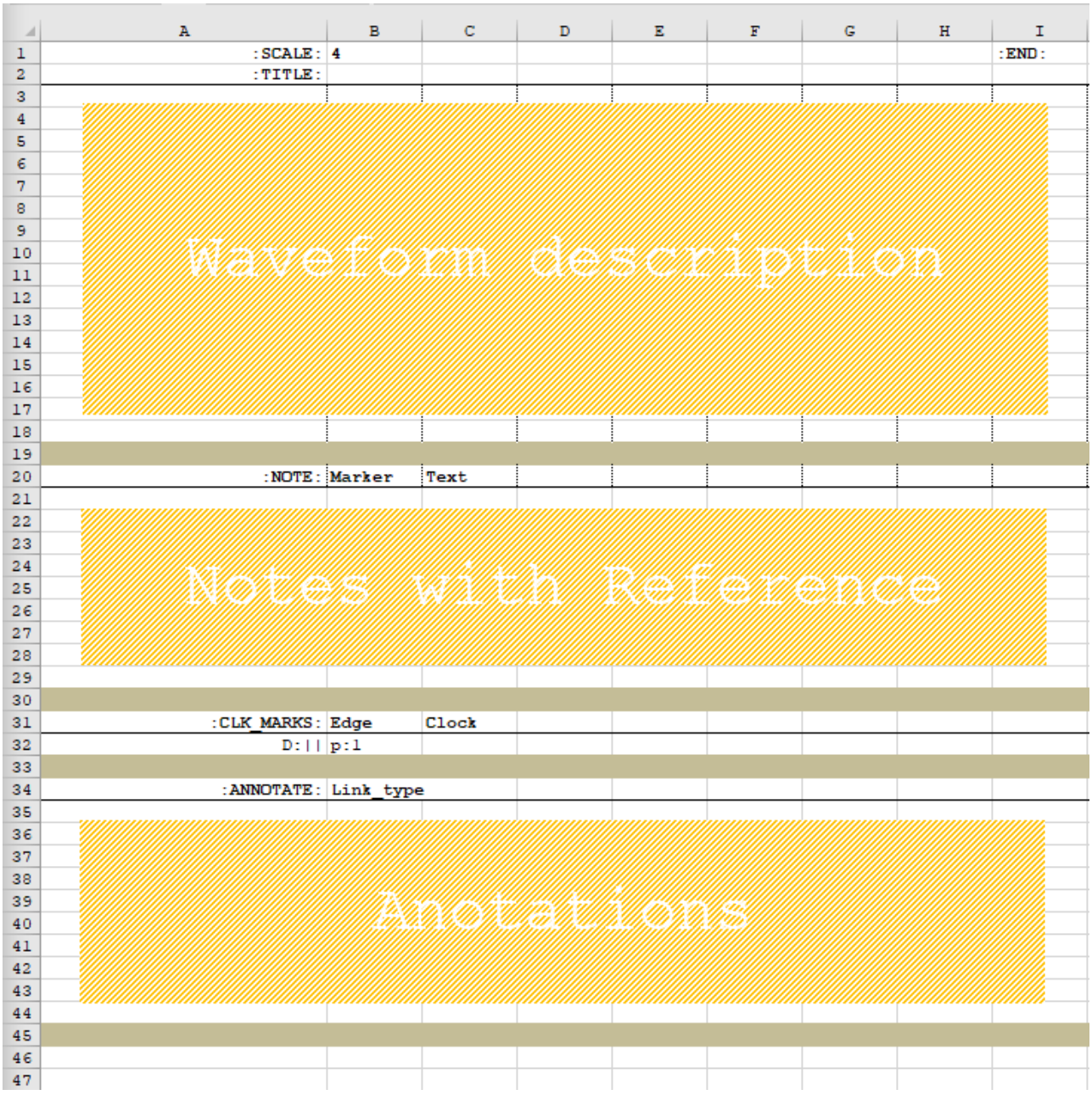


Fig. 1: **Fig**:Recomended workbook layout

Tabs Index_nt, Notes_nt will not be rendered. Tab ‘template’ holds an empty wafeform template whole tab ‘waveform_template’ is a fully working example with hints added as cell comments.

3.2 Working from the template



Template is divided into 3 sections row wise.

- Waveform description
- Notes,
- Annotations.

The other markers in the template are required by the python parser and is used as a mark-up Mark-up usually follows the convention :*markup_name*: , i.e. a reserved mark-up_name bracketed by colons

---

Note: Column 1 or A in the excel sheet cannot be empty. The parser interprets most markers when placed in the first

column. The only exception is :END:

SCALE The number placed in the next column is used to control the scale of the waveform. Scale impact how many clocks are rendered and is an important parameter to fit the required number of clocks after allowing for margins on a ISO:A4 paper in landscape mode.

Default scale is 4, allowing unto 32 clocks (or xl columns) to be rendered

END When **:END:** is specified in any column within the same row as **:SCALE:**, the parser limits time to that column. See Example TODO

TITLE Text following title will be placed as the title of the waveform: See example TODO

NOTE The notes area has three logical columns. The left most column just carries the text NOTE: This is provided so as to allow placement of multiline notes to form say a bulleted list.

:NOTE:	Marker	Text
NOTE:	cell_id>	Note accompanying the marker
NOTE:		This is a continuation line of note above if 'Marker' is empty

CLK_MARKS Control the edge and the clock on which the clock boundary is drawn.

- D:‖ is the marker for edge and
- p:1 identifies rising or posedge.

:CLK_MARKS:	Edge	Clock
D:‖	p:1	• name_of_the_clk_from_waveform_section • Do not type name, instead use reference • type '=' in the cell • click on the name of the clock in the waveform window

---

Note: For edges to be drawn it is important that each column apart from any containing a break 'I' should be numbered.

ANNOTATE Mark-up to render edges, constraints etc. Mark-up and link_type together decide the intended style of annotation

***Annotate* can specify two types of relations**

- E:<> specifies annotation drawn between edges
- L:<> specifies annotation drawn between levels. These can be used to link sampling conditions or combinatorial results.
- <> the exact type of edge and how they are interpreted varies. See table

Arcs/arrows are drawn with the form <start_type>-<endtype> where start and end are from the list below.

- > arrow head
- * filled circle

- o small caps ‘o’, open circle
- | dimension line

Examples: o->, *-*, *->, |-|

Link_type **can take on two values to specify the type of link**

- C:r - Specifies a **C**hained link. i.e a string of arcs connected back to back.
- B:r - specified a **B**aseline link. i.e the arcs all have the same start point, but multiple end points.

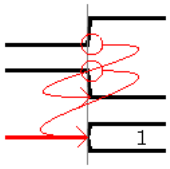
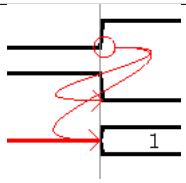
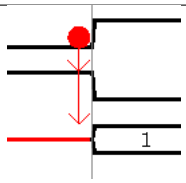
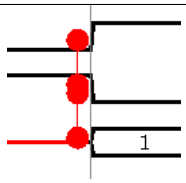
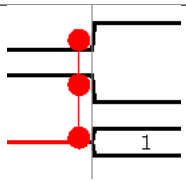
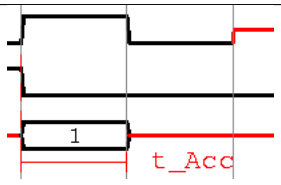
Markers **column is composed of all remaining columns where each column links to a marker.**

- For a **chained (C:r)** link arcs are drawn between pairs of markers, left to right.
- For **baseline (B:r)** links, arcs are drawn between the first specified mark and every successive edge.
- When *:ANNOTATE:* is of type E:|-|, a constraint is drawn.

This is different from the above modes in that it takes exactly two marks and the next column specifies a text to be placed following the annotation. As an example, consider annotating access time. Specify an edge of type E:|-|, C:r between two markers M1, M2 with text as ‘t_Acc’. See specific examples in table

---

Note: When more than one constraint ends on the same destination, they are drawn one below other. However, this might cause them to overlap another waveform. When such conditions are detected the script emits an error of the form:: **[WARNING 554] add_arrows(): Multiple dimensions drawn may overlap a waveform, Add a spacer in excel if required R24> W24>**

:ANNOTATE:	Link Type	Markers/Action.	Render
E:o->	**C:r**	• C18> C19> C20> • chained edge links	
E:o->	**B:r**	• C18> C19> C20> • baseline edge links	
L:*->	**B:r**	• C18> C19> C20> • basline level links • useful for combinatorial dependencies, co-sampling etc	
L:*-*	**C:r**	• C18> C19> • two co-samples signals	
L:*-*	**B:r**	• C18> C19> • two co-samples signals • **preferred method**	
E:|-|	**B:r**	• C18> C19> t_Acc • baseline edge links	

3.3 The empty template

Table 1: Source –> render

source

	A	B	C	D	E	F	G	H	I
1	:SCALE:	4							:END:
2	:TITLE:								
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20	:NOTE:	Marker	Text						
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31	:CLK_MARKS:	Edge	Clock						
32	D:||	p:1							
33									
34	:ANNOTATE:	Link_type							
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									

Waveform description

Notes with Reference

Anotations

Rendered ouptut

CHAPTER

FOUR

Markup for signal names

4.1 Clock

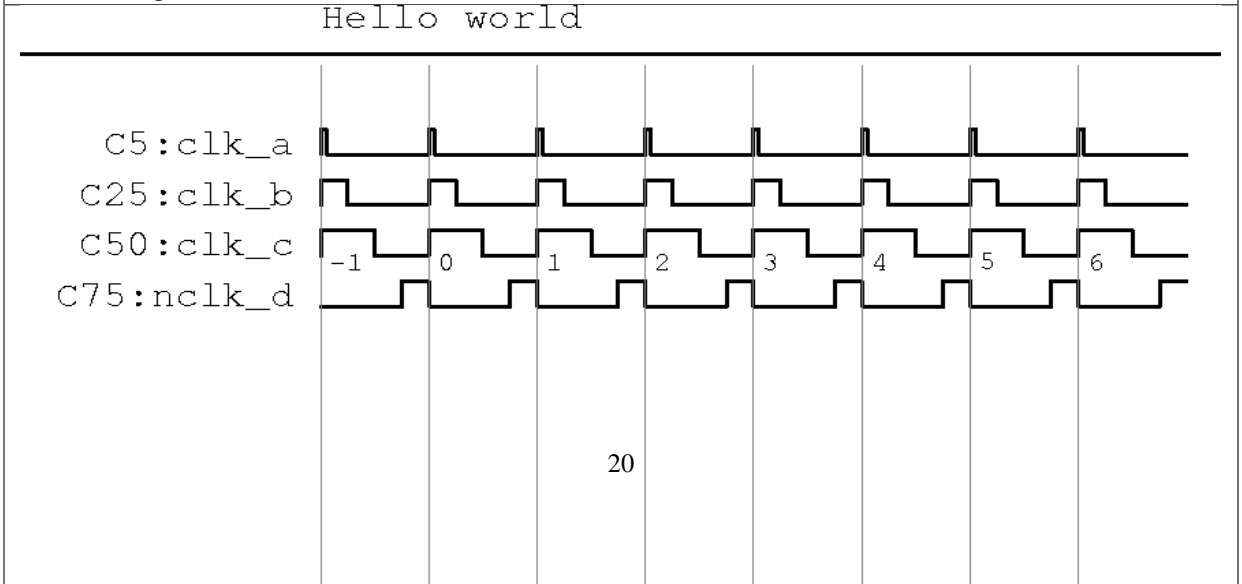
- add the cycle numbers, not doing so at least till **:END** leads to errors
- Add one clock to the CLK_MARKS section.
- The clock name shall have **clk_xxx** in it
- Mark-up for clock is **C<duty>:** where duty is the duty cycle of the clock.
- If the clock name contains **nclk_** an inverted clock will be drawn

Table 1: Step 1

source

	A	B	C	D	E	F	G	H	I	J
1	:SCALE:	4							:END:	
2	:TITLE:	Hello world								
3										
4	C5:clk_b	-1	0	1	2	3	4	5	6	
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24	:NOTE:	Marker	Text							
25										
26	NOTE:									
27										
28										
29										
30										
31										
32										
33										
34										
35	:CLK_MARKS:	Edge	Clock							
36	D:||	p:1	C5:clk_b							
37										
38	:ANNOTATE:	Link_type								
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										

Rendered ouptut



4.2 Signal

- Signals have no markers and represent a single bit
- The first columns is used for the initial condition, which is 1, 0 or x.
- When the first row is empty, parser emits a warning and insert an x
- For a transition, place a 1 in the corresponding clock column. This specifies the state of the signal following the active clock edge at the start of the cell. i.e the clock edge at its left boundary.
- To transition back place a 0 in the immediate clock.
- A seq 010 draws a pulse
- Only value changes are required to be captured.

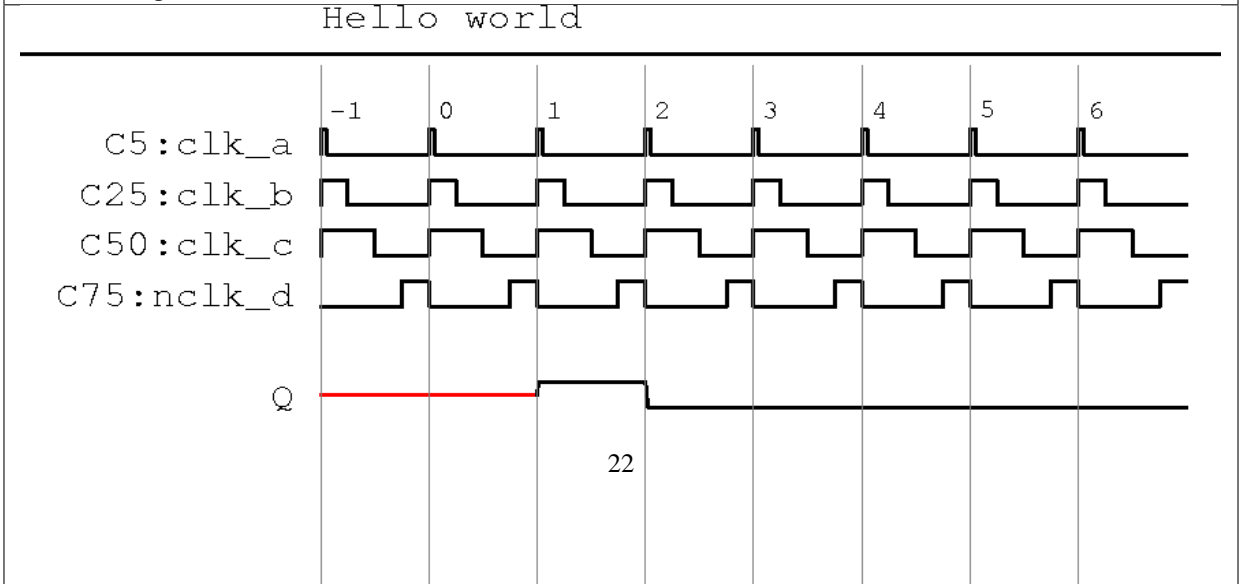
Table 2: Step 2

source

	A	B	C	D	E	F	G	H	I	J
1	:SCALE:	4							:END:	
2	:TITLE:	Hello world								
3										
4	C5:clk_a	-1	0	1	2	3	4	5	6	
5	C25:clk_b	-1	0	1	2	3	4	5	6	
6	C50:clk_c	-1	0	1	2	3	4	5	6	
7	C75:nclk_d	-1	0	1	2	3	4	5	6	
8										
9	Q			1	0	1				
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24	:NOTE:	Marker	Text							
25										
26	NOTE:									
27										
28										
29										
30										
31										
32										
33										
34										
35	:CLK_MARKS:	Edge	Clock							
36	D:||	p:1	C5:clk_a							
37										
38	:ANNOTATE:	Link_type								
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										
52										
53										

Index_nt | template | waveforms_template | Notes_nt

Rendered ouptut



4.3 Bus

- Buses use the mark-up **B:<name>**
- Buses take on the additional state ‘u’ in addition to ‘x’
- Ulu is rendered shaded to represent a don’t care value.
- x is used interchangeably az HiZ on the bus.
- Buses also follow value change representations.
- A valid value is interpreted when a non u|x character is present by itself in the cell.
  - Bus values can be coloured with one of four colours by appending [c:o|r|g|b]
  - o - orange, r- red . . . get the picture
  - use colours sparingly, only if you absolutely need to.
- characters from the set A-Z a-z0-9_+-:*() with or without space are supported

---

Note:

- Reccomended to use atmost 8 char for values of data. More characters may be used but they tend to overflow the space allocated for a clock when used with scale 4.

OuterFrameSep

Caution: While colours are provided their grey scale weights have not been chosen appropriately. Thus use of colours may be confusing in grey scale prints.

In the example below, bus illustrates use of colouring. Bus2 has no initial condition defined and the annotation although starts from cell C13 overflows into adjacent cells within excel. Since there is no value change in subsequent cells they are rendered cleanly. The example demonstrates characters that can be used for the annotation, as well as the use of x to render Hiz

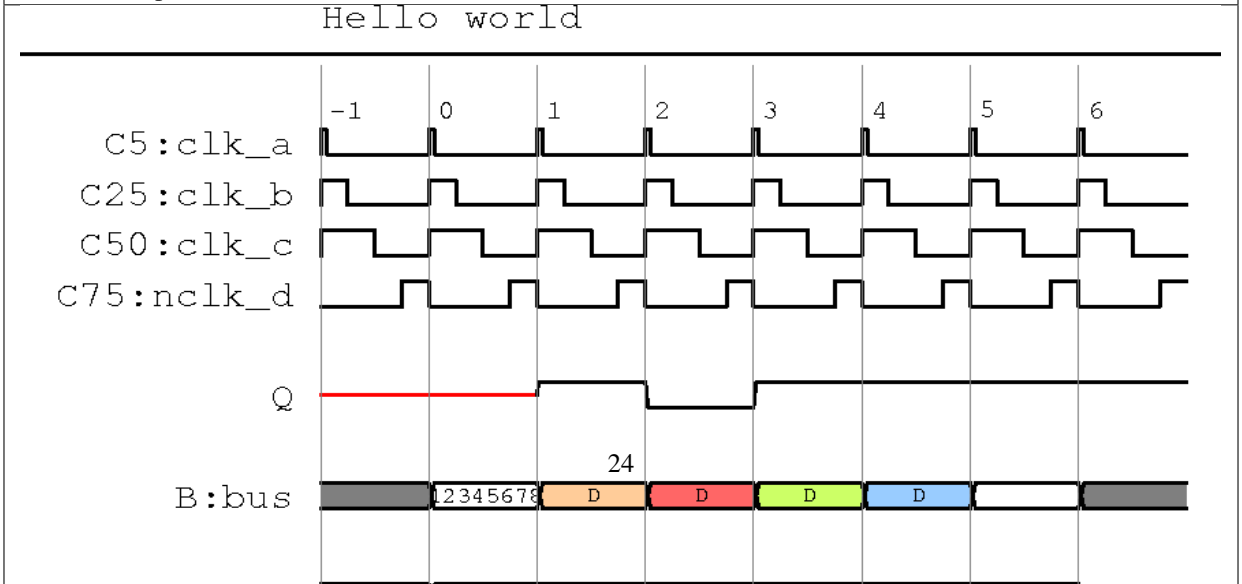
Table 3: Step 3

source

	A	B	C	D	E	F	G	H	I	J
1	:SCALE:	4							:END:	
2	:TITLE:	Hello world								
3										
4	C5:clk_a	-1	0	1	2	3	4	5	6	
5	C25:clk_b	-1	0	1	2	3	4	5	6	
6	C50:clk_c	-1	0	1	2	3	4	5	6	
7	C75:nclk_d	-1	0	1	2	3	4	5	6	
8										
9	Q			1	0	1				
10										
11	B:bus	u	12345678	D[c:o]	D[c:r]	D[c:g]	D[c:b]		u	
12										
13	B:bus2		A-Z a-z 0-9 + - : * ( )						x	
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24	:NOTE:	Marker	Text							
25										
26	NOTE:									
27										
28										
29										
30										
31										
32										
33										
34										
35	:CLK_MARKS:	Edge	Clock							
36	D:||	p:1	C5:clk_a							
37										
38	:ANNOTATE:	Link_type								
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51										
52										
53										

Index_nt | template | waveforms_template | Notes_nt

Rendered ouptut



4.3.1 Breaks

The text decoration on a signal name can be additionally controlled by appending any of the following modifiers to the name. o <i> Italics

o <b> Bold o <b><i> bold italics

The motivation to include such modifiers is to enhance communication. For example a signal of interest may be picked out in bold, but a place holder (ie signal name used is not the exact name in design, or a group of signals used for capturing design intend) may be optionally marked in italics. As an example <addr_phase> or <intr_packet>, which collects all co samples signals which have the same timing during an address phase, or the output of the interrupt router. It is often convenient and effective to abstract away trivial detail such as individual signal names when drafting a conceptual idea.

4.4 Other Enhancements

4.4.1 Breaks

- Waveform breaks may be added by placing a | in the columns representing break.
- It is recommended that all rows of the column within the waveform window carry '|'
- Marker rows are automatically exempted. This allows free insertion of marker rows and reducing the chance of parser errors.
- While rendering a break, the state before the break is extended across the break. Thus a text annotation on a bus will continue across the break.

OuterFrameSep

Caution: some gotchas around this yet to be resolved. The parser does not break, but if it does not produce result as expected it is reccomended to flank the break and avoid toggles.

4.4.2 Gated Clock

- To render a gated clock use G in the cell where the clock is to be gated off.
- These mark-ups can also be used to create divided clocks.

Tip:

Instead of manually placing 'G' in cells to create gated clocks, a reccomended method would be to pick a row that lies after the

- as an example to generate clk_c below.
- =IF((MOD(B51,3)),"G",B51) will replace all except every 3rd clock with G rendeinrg a div by 3 clock. Row 51 just contains an ascending count

---

4.4.3 Glitches

- A mechanism is provided to draw a glitch. A Glitch will be drawn just following the clock edge assuming the Cell contents represents 'Q' as opposed to 'D' The above is only for interpretation as Q values don't glitch.
- the utility of a glitch is in representing a combo signal, for example a req that can be taken away w/o an ack. So here the glitch demonstrates intent and behaviour rather than a purely physical signal glitch.
- Mark-up used is **G** by itself on a signal row.
- Direction of glitch is inferred from the bounding signal levels. A glitch does not make much sense unless both the pre and post states are logically opposed to the glitch state.

4.4.4 Combinatorial, Late arriving signals.

- A class is added to draw a combinatorial or late arriving signal. This provides a mechanism to accurately communicate intent, and a hint as to how the signal is to be implemented. An example of such a signal would be a ready, or ack.
- Mark-up for rendering such a transition is to replace 1 with **0.75** to produce a delayed transition to 1
- Mark-up for rendering such a transition is to replace 0 with **-0.75** to produce a delayed transition to 0

Note: When 0.xx or -0.xx is used to add an uncertainity to a combo signal the actual fraction has no relevance. the transition region always defaults to 25% of the clock.

The example below illustrates, glitches, gated clocks(clk_c), combo uncertainty and breaks

Table 4: Step 4

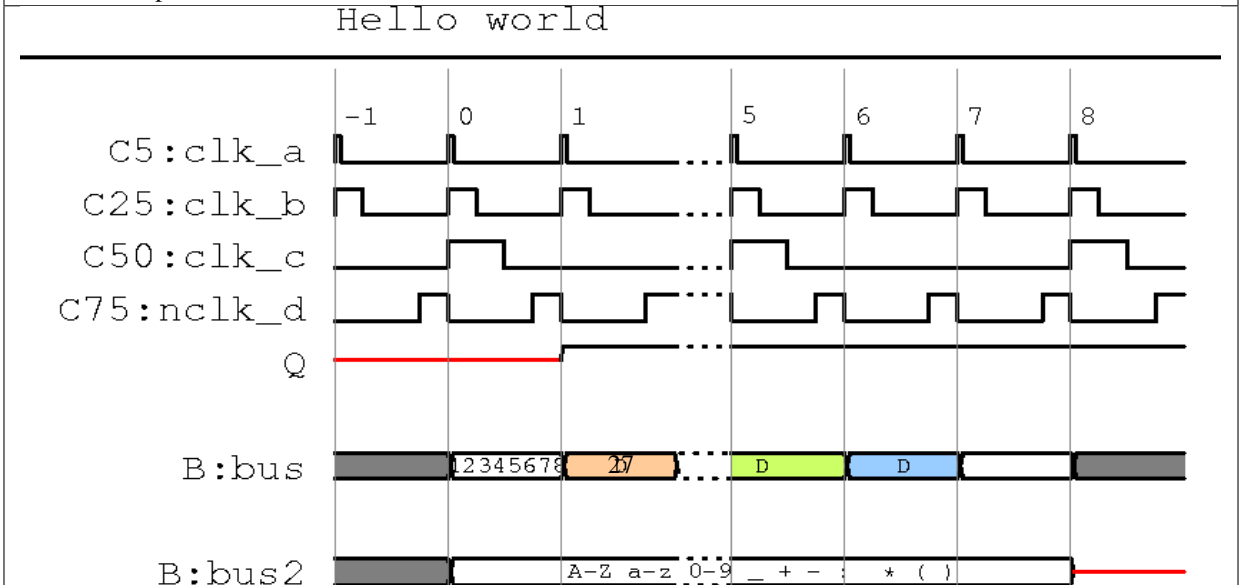
source

	A	B	C	D	E	F	G	H	I	J
1	:SCALE:	4							:END:	
2	:TITLE:	Hello world								
3										
4	C5:clk_a	-1	0	1		5	6	7	8	
5	C25:clk_b	-1	0	1		5	6	7	8	
6	C50:clk_c	G	0	G	G	3	G	G	6	
7	C75:nclk_d	-1	0	1		5	6		8	
8	M:					F9>				
9	Q			1		1				
10										
11	B:bus	u	12345678	D[c:o]		D[c:g]	D[c:b]		u	
12										
13	B:bus2		A-Z a-z 0-9 _ + -						x	
14										
15	Combo	x	G	0		1	G	0.5	-0.75	
16										
17										
18										
19										
20										
21										
22										
23										
24	:NOTE:	Marker	Text							
25										
26	NOTE:									
27										
28										
29										
30										
31										
32										
33										
34										
35	:CLK_MARKS:	Edge	Clock							
36	D:||	p:1	C5:clk_a							
37										
38	:ANNOTATE:	Link_type								
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										
50										
51	M:	-1	0	1	2	3	4	5	6	
52										
53										

=IF((MOD(B51,3)),"G",B51)

Index_nt | template | waveforms_template | Notes_nt

Rendered ouptut



4.5 Markup for annotations.

4.5.1 Creating a markup row.

In order to add annotations, the parser relies on marked points in the timing diagram. The markers can only be placed on a cycle boundary, and this cycle boundary coincides with the grid on the excel template, which in turn times the fastest clock.

- Markers are created on a special row placed under the signal of interest with mark-up **M:** in the signal name column.
- The marker points to the edge flanking the right of the cell where it is placed. This is visually encoded with '>', as in example.
- To place a marker, it is best to use a formula. The formula puts the position of the cell ie column and row along with >, such as 'H31>'
- =CONCATENATE(SUBSTITUTE(CELL("address"),"$",""),">") will make the naming consistent when row names exceed Z

Note:

- *=CONCATENATE(CHAR(COLUMN()+64)&ROW(),">")*, was the excel formula used previously.
- This has been replaced with the robust method above. The simple formula existed in earlier versions where the scale was fixed and consequently the number of columns was contained. With a relaxed scale the number of columns can grow.
- If the note does not appear explicitely marked, change to formula above.
- When using the template, adding the mark-up M: will conditionally format the entire excel row to a hatched pattern. Although the parser does not see or use this information it is of great benefit at design entry.

4.5.2 Annotation.

Once the markers are placed, they can be used in annotations. Refer section for mark-up.

There are two categories of annotations

1. A label with a note to give explicit information a,r,t to a specific location in the waveform.
2. Relationship and flow patterns.

Adding a note

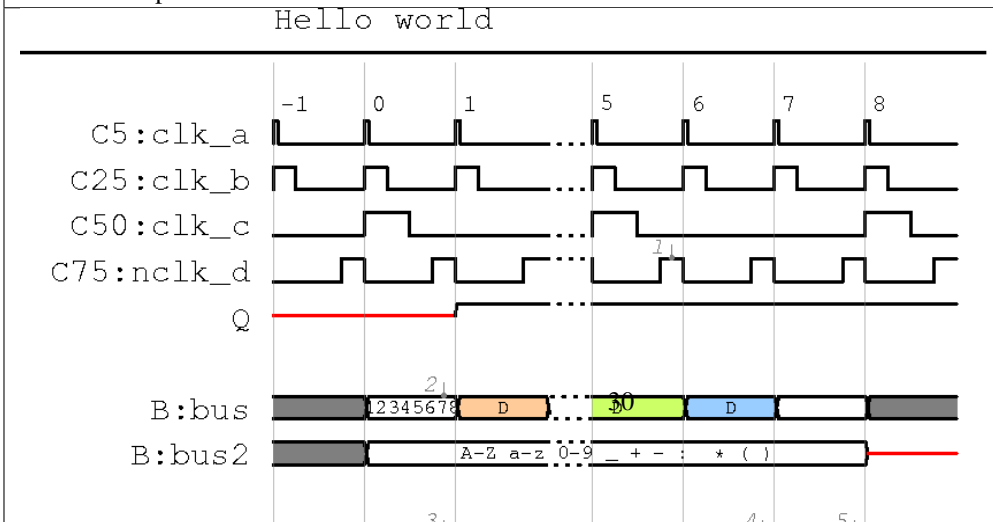
Table 5: Step 5

source

	A	B	C	D	E	F	G	H	I	J
1	:SCALE:	4							:END:	
2	:TITLE:	Hello world								
3										
4	C5:clk_a	-1	0	1	|	5	6	7	8	
5	C25:clk_b	-1	0	1	|	5	6	7	8	
6	C50:clk_c	G	0	G	G	3	G	G	6	
7	C75:nclk_d	-1	0	1	|	5	6	7	8	
8	M:				1	F8>				
9	Q			1	|	1				
10					|					
11	B:bus	u	12345678	D[c:o]	|	D[c:g]	D[c:b]		u	
12	M:		C12>		1					
13	B:bus2		A-Z a-z 0-9 _ + -		|				x	
14					|					
15	Combo	x	G	0	|	1	G	0.5	-0.75	
16	M:		C16>		1		G16>	H16>		
17					|					
18										
19	:NOTE:	Marker	Text							
20										
21	NOTE:	F8>	Mark something on a clock cycle.							
22	NOTE:	C12>	The text just fits							
23	NOTE:	C16>	A glitch at the beginning of the clock							
24	NOTE:	G16>	Negative glitch							
25	NOTE:	H16>	Transition to 1							
26	NOTE:		Next cycle transition to 0							
27	NOTE:		This is a multi line note with							
28										
29										
30	:CLK_MARKS:	Edge	Clock							
31	D:||	p:1	C5:clk_a							
32										
33	:ANNOTATE:	Link_type								
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46	M:	-1	0	1	2	3	4	5	6	
47										
48										
49										
50										
51										
52										
53										

Index_nt | async | template | waveforms_template | Notes_nt

Rendered ouptut



The example below illustrates, glitches, gated clocks(clk_c), combo uncertainty and breaks

Adding relationsips

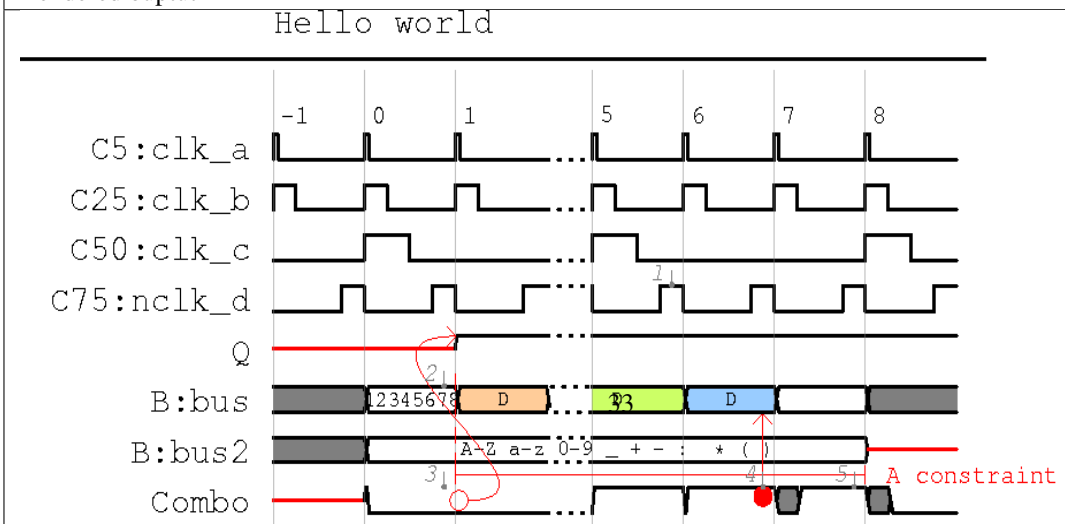
Table 6: Step 6

source

	A	B	C	D	E	F	G	H	I	J
1	:SCALE:	4							:END:	
2	:TITLE:	Hello world								
3										
4	C5:clk_a	-1	0	1	|	5	6	7	8	
5	C25:clk_b	-1	0	1	|	5	6	7	8	
6	C50:clk_c	G	0	G	G	3	G	G	6	
7	C75:nclk_d	-1	0	1	|	5	6	7	8	
8	M:				|	F8>				
9	Q			1	|	1				
10	M:		C10>		|					
11	B:bus	u	12345678	D[c:o]	|	D[c:g]	D[c:b]		u	
12	M:		C12>		|		G12>			
13	B:bus2		A-Z a-z 0-9 _ + -		|				x	
14	M:				|			H14>		
15	Combo	x	G	0	|	1	G	0.5	-0.75	
16	M:		C16>		|		G16>	H16>		
17					|					
18										
19	:NOTE:	Marker	Text							
20										
21	NOTE:	F8>	Mark something on a clock cycle.							
22	NOTE:	C12>	The text just fits							
23	NOTE:	C16>	A glitch at the beginning of the clock							
24	NOTE:	G16>	Negative glitch							
25	NOTE:	H16>	Transition to 1							
26	NOTE:		Next cycle transition to 0							
27	NOTE:		This is a multi line note with							
28										
29										
30	:CLK_MARKS:	Edge	Clock							
31	D:||	p:1	C5:clk_a							
32										
33	:ANNOTATE:	Link_type								
34										
35	E:o->	C:r	C16>	C10>						
36	L:*->	C:r	G16>	G12>						
37	E:|-|	C:r	C10>	H14>	A constraint					
38										
39										
40										
41										
42										
43										
44										
45										
46	M:	-1	0	1	2	3	4	5	6	
47										
48										
49										
50										
51										
52										
53										

Index_nt | async | template | waveforms_template | Notes_nt

Rendered ouptut



CHAPTER

FIVE

Dealing with Asynchronous events

With state of the art digital design, invariable one will run into a requirement to represent an asynchronous event. Although asynchronous events in a standard work flow remains strictly controlled there might exist cases such as interfaces which are to be represented accurately. The current parser does not provide dedicated support for asynchronous events. However, the following tricks can be used to implement asynchronous timing events very effectively. In most cases, the scenario revolves around some kind of clock domain crossover. Even when the clock is not known using a virtual clock to time the signal should still work.

5.1 Tips for modelling async

- Assume we have two async clocks. For modelling purposes, these need not be accurate relationships. The very idea of representing async clocks is that we cannot use a predictable mechanism to go from one domain to another. If such predictable method exist we drop down to a less stringent mechanism for moving between multi-cycle clocks.
- Assume two clocks, say with freq 3 and 5. Use the LCM of these two clocks to define a master clock.
- Create the master clock. This will be rendered but has no relevance apart from acting as a time base for the clocks of interest ie, 3,5. For the same reason, we could specify an extremely low duty cycle, to provide something like an impulse to avoid confusion. for this clock the cycle numbers are done sequentially.
- To create the freq5 and freq 3 clocks, we use the idea of gated clocks. ie a form like 1 G G G G 6 etc.

OuterFrameSep

Caution:

- The clocks rendered cannot achieve 50% duty cycle. If you wish to do so the clock has to be drawn using the signal construct. However, doing so makes maintenance difficult, i.e if we decide to add a cycle inbetween, the entire signal might need readjusting. It might just be prudent to use the defined method, but add a note to draw the readers attention to such irregularity.
- Using this method implies the pulses are numberd non sequentially, ie they assume the clock number from the master clock when active. A new derived clock construct may be used in the future to overcome this inconsistency.

- Rather than hand coding the waveform, a simple trick is to employ an excel formula such as using MOD() to decide where the clock is active; i.e. *mod(master_clk_cnt,5)==0 then master_clk_cnt else G*
- repeating for the other clock with mod(master_clk_cnt,3) now provides two async clocks which will suffice for illustration.

- At this stage, we could use the master clock to mark edges, However this can be distracting. Instead it is recommended to draw edges from both derived clocks. They will still be rendered in grey solid lines, as not option is provided yet to control the type of line. This may be extended in the future.

Table 1: Step 1

source

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	:SCALE:	2													
2	:TITLE:	A simple clock domain crossover with REQ/ACK, and double synchronisation either way													
3															
4	C1:clk_master	-1	0	1	|	5	6	7	8	9	10	11	12	13	14	15
5	C25:clk_b	G	0	G	|	G	6	G	G	9	G	G	12	G	G	15
6	C50:clk_c	G	0	G	|	5	G	G	G	G	10	G	G	G	G	15
7	C75:nclk_d	-1	0	1	|	5	6	7	8	9	10	11	12	13	14	15
8	M:															
9	D			1	|										
10	sig_clk_b<i>	0			|		1									
11	M:						G11>			J11>						
12	B:sig_clk_c<i>										0b01					0b11
13	M:									J13>					O13>	
14	B:sig_ack_clk_b															
15	M:															
16																
17																
18																
19																
20																
21																
22																
23																
24	:NOTE:	Marker	Text													
25																
26	NOTE:	Z13>	Possible metastability on landing flop in clk_c													
27	NOTE:	B1>	Possible Metastability on landing flop in clk_b													
28																
29																
30																
31																
32																
33																
34																
35	:CLK_MARKS:	Edge	Clock													
36	D:||	p:1	C50:clk_c													
37	D:||	p:1	C25:clk_b													
38																
39	:ANNOTATE:	Link_type														
40																
41	E:o->	C:r	J11>	J13>	O13>											
42	E:o->	C:r	R13>	R15>	U15>											
43	E:o->	C:r	X15>	X11>												
44																
45																
46																
47																
48																
49																
50																
51																
52	M:	-1	0	1	2	3	4	5	6							
53																

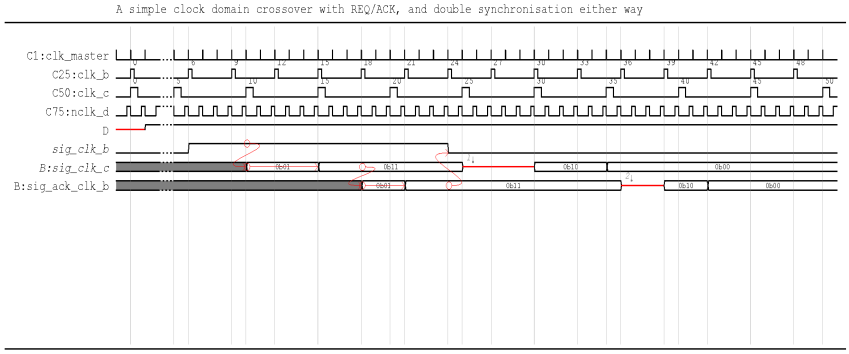
=IF((MOD(B4,3)),"G",B4

=IF((MOD(B4,5)),"G",B4

Index_nt | async | template | waveforms_template | Notes_nt

Ready

Rendered ouptut



NOTE:

1. (1) Possible metastability on landing flop in clk_c
2. (2) Possible Metastability on landing flop in clk_b

CHAPTER

SIX

Indices and tables

- genindex
- modindex
- search