

# Waves Timing Diagram Editor

Waves is a simple timing diagram editor distributed as a Windows binary and copylefted (GNU GPL v3) Tcl/Tk source. Waves has been inspired by Wavedrom, the JSON-based timing diagrammer. However, advanced features such as cascaded arrows, timing diagram annotation, flexible text positioning, etc. are much simpler in *Waves*. © Anirban Banerjee 2015

## 1. Waves Window

Waves windows are arranged as shown in Figure 1:

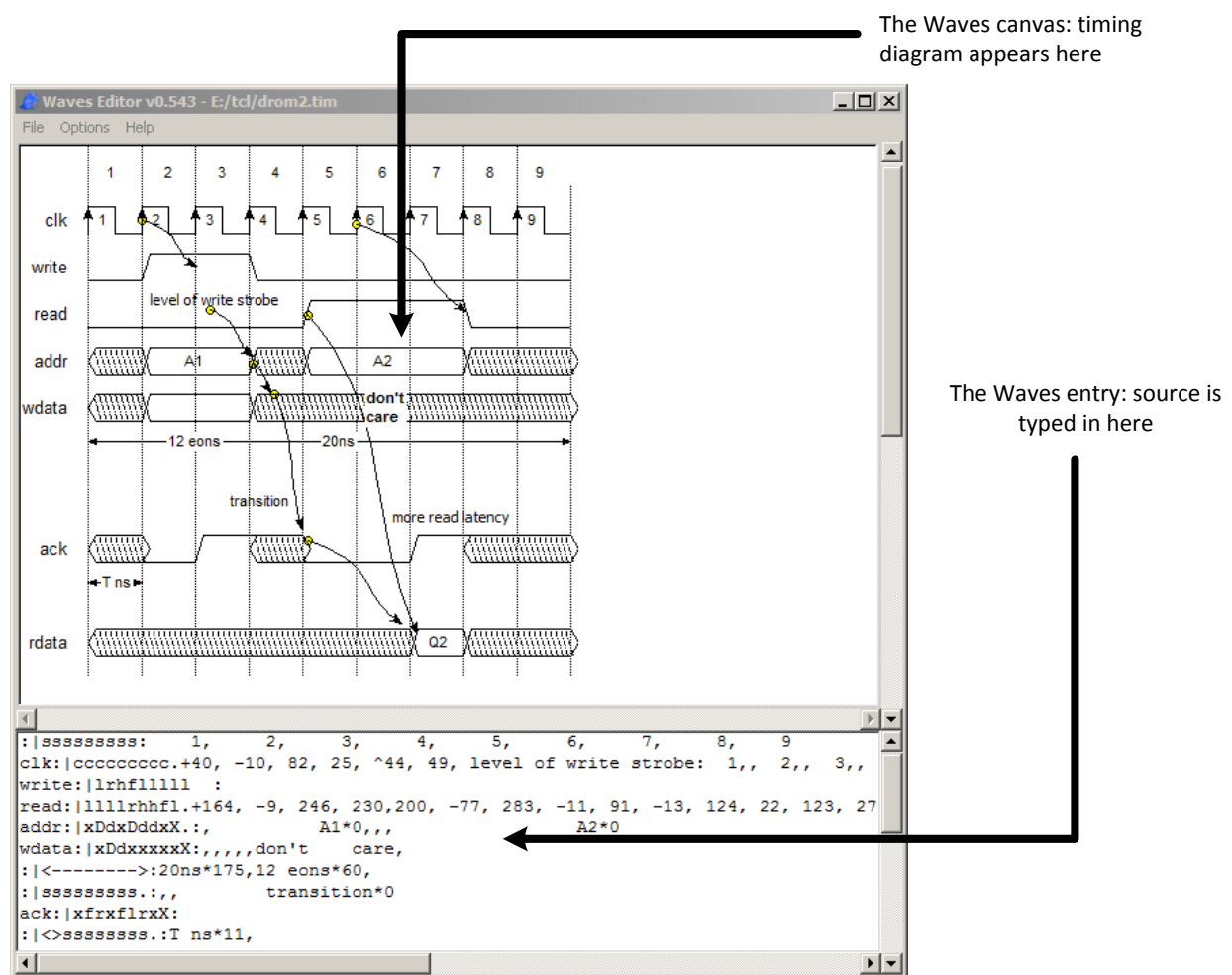


Figure 1: Waves

## 2. Waves Canvas

The Waves canvas is divided into equal-sized vertical grid that can be thought of as a clock period; see Figure 2. Each signal will be described by a new line of command in the source entry window.

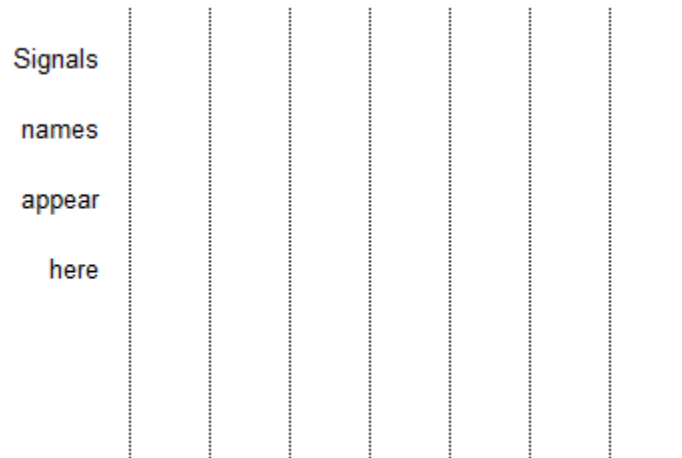


Figure 2: Waves Grid

A Waves command consists of three sections, each separated by a colon. This is shown in Figure 3.

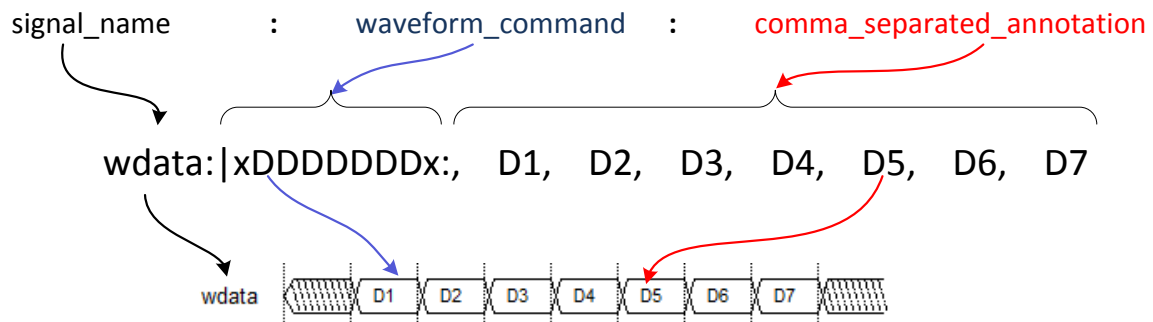


Figure 3: Command Format

## 3. Waves Signal Names

The signal name can be of any length but wraps around after 14 characters and is right justified. All characters other than colon ':' are allowed. Future versions will remove this limitation. Signal names are optional.

## 4. Waveform Commands

The basic Waves commands are shown in Figure 4.

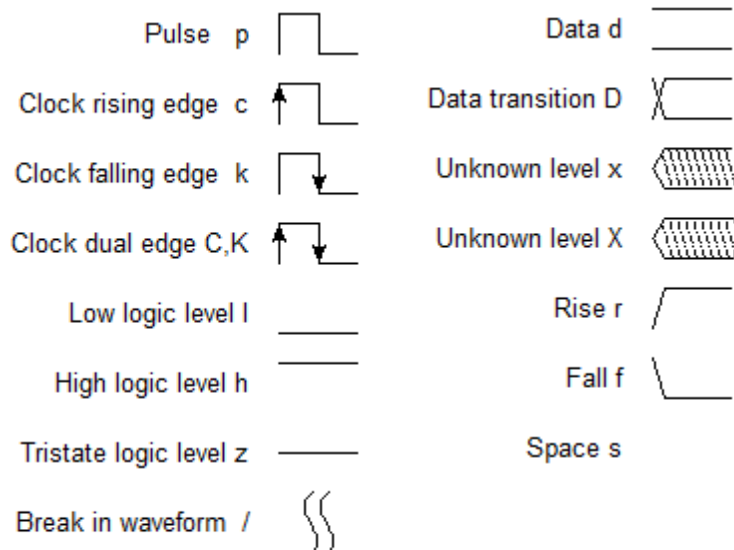


Figure 4: Waves Basic Commands

## 5. Waveform Metacommands

Waves has these metacommands that can appear in the waveform command shown in Table 1:

	This turns on the vertical grids. The last gridline can be turned on by a '.' or ';'.
!	This turns off the vertical grid.
~	This inverts clocks or pulses (commands p, c, k, C and K).
#	This indicates a comment till the end of the line. If this is the first character of the Waves command, an empty line will be inserted.
. or ;	Adds grid line to the end.

Table 1: Waves Metacommands

## 6. Advanced Waveform Commands

- Data annotation
- Time arrows
- Cause-effect arrows
- Flexible text

## 6.1. Data Annotation

The data annotation section starts after second colon in the command. Each comma-separated string is associated with a grid. The text is aligned to the left of the grid ([Figure 5: Data Annotation](#)), and the data string will wrap around on each grid ([Figure 6: Data Annotation Wrap](#)).

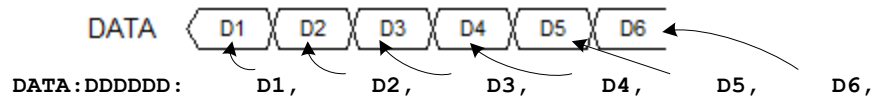


Figure 5: Data Annotation

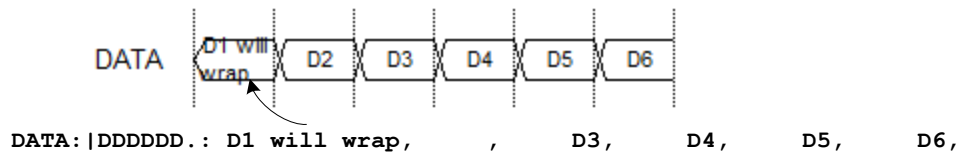


Figure 6: Data Annotation Wrap

In order to set the wrap at  $n$  characters, the string should have a “ $*n$ ” suffix. This is shown in [Figure 7](#) showing wrapping after three grids. In order to to disable all wrapping, use a “ $*0$ ” suffix.

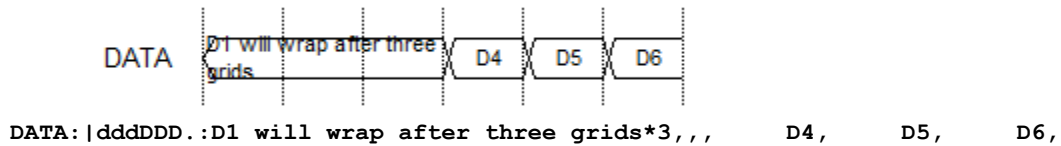


Figure 7: Data wrap after three grid

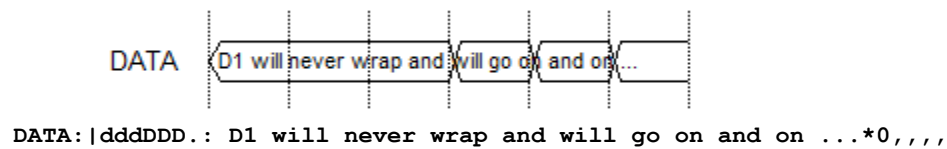


Figure 8: Data Annotation Without Wrap

## 6.2. Time Arrows

Waves supports arrows on the time axis as shown in Figure 9 that can be optionally annotated (see section on Data Annotation). The position of the annotation is from the first gridline on the left. Please note that since the annotation is associated with the gridline where the left arrowhead starts, it must be after the second comma.

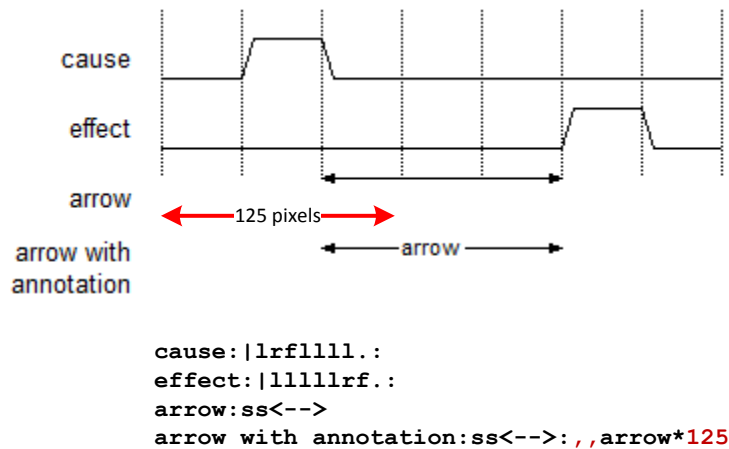


Figure 9:Waves Time Arrows

The time arrow commands are shown in Figure 10.

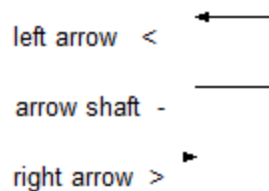


Figure 10: Time Arrow Commands

## 6.3. Cause-effect Arrows

Waves provides an easy and flexible way of inserting causality arrows.

1. In the waveform command, type '+'.
2. Right-click at the arrow start position, and click on “Insert Coords into source”. This will insert the start canvas X and Y coordinates into the command.
3. Right-click at the arrow end position, and click on “Insert Coords into source”. This will insert the end canvas X and Y coordinates into the command, and the arrow will be rendered.
4. There must be a single '+' command in a command line. Any number of cause-effect arrows can be added to sequentially.

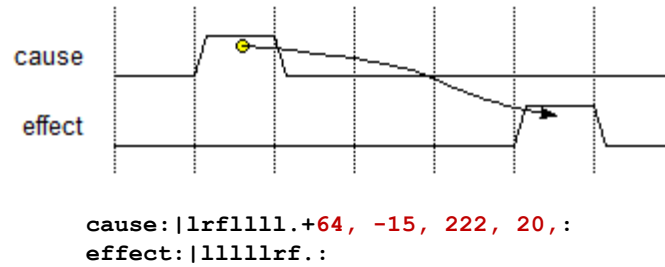


Figure 11: Cause-effect Arrows

## 6.4. Flexible Text Labels

Waves has a simple method of inserting text string at any position in a timing diagram.

1. In the waveform command, type '^'.
2. Right-click at the arrow start position, and click on "Insert Coords into source". This will insert the canvas X and Y coordinates into the command for the left edge of the text.
3. Type in the text in the command.
4. There must be only one '^' command in a line.
5. Arrow '+' commands must always precede any text label '^' command in a command line.

In any line of command, the '+' command and all cause-effect arrow coordinates must appear before all '^' command and coordinates for text.

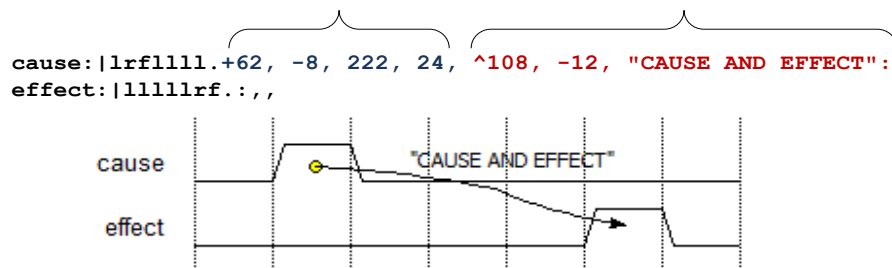


Figure 12: Flexible Text Labels

## 7. Finally

Figure 13 shows the rendering of a moderately complex timing diagram.

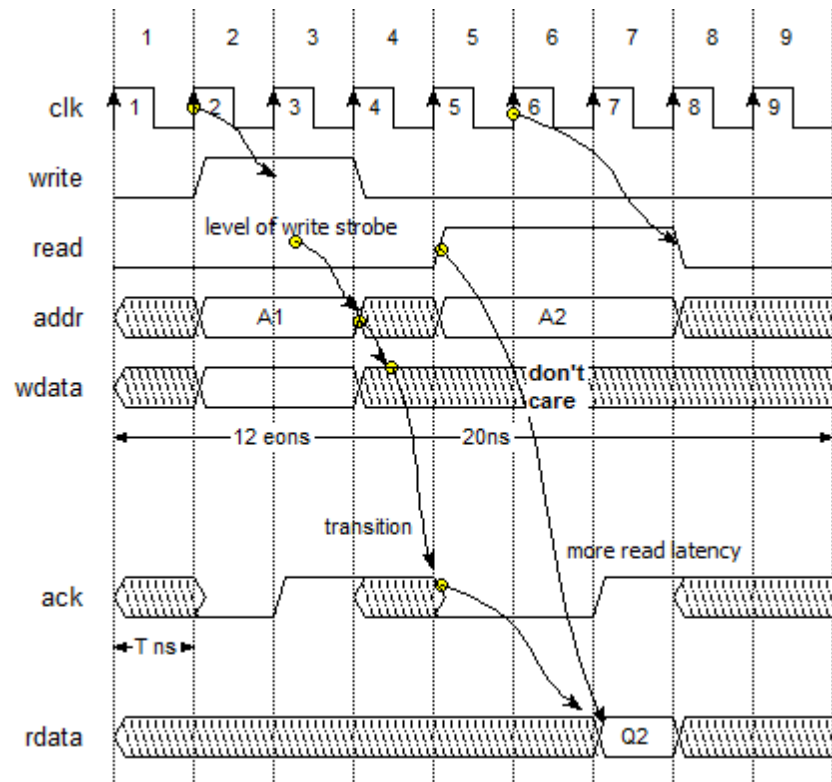


Figure 13: Final Rendering

## 8. Legal

Waves is copylefted software released under the GNU GPL version 3. © 2015 Anirban Banerjee

Waves comes with absolutely no warranty.