Non-Linear Waveforms for Quick Trace Navigation

Jannis Stoppe[1] Rolf Drechsler[2]
[1,2]University of Bremen, Group of Computer Architecture
[1,2]German Research Center for Artificial Intelligence (DFKI), Research Department for Cyber-Physical Systems
[1]jstoppe@informatik.uni-bremen.de [2]drechsler@uni-bremen.de

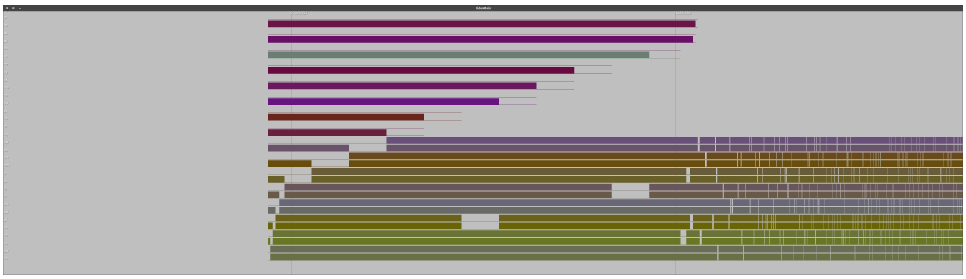

Fig. 3. *Tides'* non-linear scaling of the time axis

System trace analysis is mostly done using waveform viewers – tools that relate signals and their assignments at certain times. While generic hardware design is subject to some innovative visualisation ideas [1], [2] and software visualisation has been a research topic for much longer [3], [4], these classic tools have been part of the design process since the earlier days of hardware design – and have not changed much over the decades. Instead, the currently available programs [5], [6], [7], [8], [9], [10], [11] have evolved to look practically the same, all following a familiar pattern that has not changed since their initial appearance.

We argue that there is still room for innovation beyond the very classic waveform display though. Especially considering how designers are often using waveform viewers on a daily basis, these are tools that should be subject to constant improvement and re-evaluation instead of the current stagnation.

We implemented a proof-of-concept waveform viewer (codenamed *Tides*) that has several unique features that go beyond the standard set of features for waveform viewers.

- First, it uses an OpenGL canvas to draw the waveforms instead of standard operating system APIs. It thus utilizes the graphics unit to draw the given data, thus shifting the drawing tasks from the CPU to the graphics unit. This in turn allows the waveform viewer itself to display a smooth behaviour such as animated zooms and shifts, which in turn keeps the users actions more comprehensible for potential spectators that do not control the UI themselves.
- To the sides of the canvas, there are areas that switch to a logarithmic scale of the time axis (see Fig. 3). This leads to waveforms that are distorted towards the window's borders, appearing more and more compressed as they approach them. This feature again serves a dual purpose:
  1) It allows designers to inspect a given area of interest while at the same time giving them an overview over the other parts of the trace, showing them at a single glance whether they are at the beginning or the end of the trace and where to expect more activity.
  2) It enables them to quickly select the area of interest as the classic interaction of specifying the left and right border of the circuit to be displayed no longer can only be used to zoom in. Instead, selecting an area outside of the centre now allows them to zoom out and selecting an area that is at least partly located to one side of the centre enables them to pan the viewport.

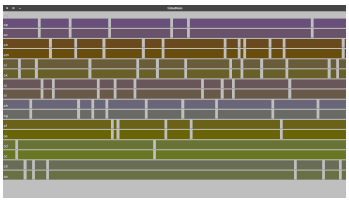

Fig. 4. More classic view, disabling the non-linear scaling

- The usage of more modern languages and frameworks allows the application to be ported easily. Building it for e.g. smartphones or a web application can be done at virtually no additional cost, e.g. allowing engineers to briefly look at data when on the go.

Acknowledgements

This research was supported by the German Federal Ministry of Education and Research (BMBF) under grant 01IW16001 (SELFIE).

References

[1] R. Drechsler and M. Soeken, "Hardware-software co-visualization: Developing systems in the holodeck," in *Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 1–4, IEEE, 2013.
[2] K. Jelemenská, M. Nosál, and P. Čičák, "Visualization of verilog digital systems models," in *Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering*, pp. 805–818, Springer, 2013.
[3] M. Petre, E. de Quincey, *et al.*, "A gentle overview of software visualisation," *PPIG News Letter*, pp. 1–10, 2006.
[4] A. R. Teyseyre and M. R. Campo, "An overview of 3d software visualization," *IEEE transactions on visualization and computer graphics*, vol. 15, no. 1, pp. 87–105, 2009.
[5] A. Gallotta and W. Snyder, "Dinotrace." http://www.veripool.org/wiki/dinotrace, 1998. Accessed: 2016-11-15.
[6] "Styx: the Hades waveform viewer." https://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/styx.html, 2006. Accessed: 2016-11-15.
[7] J. Wilcox, "Scansion." http://www.logicpoet.com/scansion/, 2008. Accessed: 2016-11-16.
[8] T. Bybell, "GTKWave." http://gtkwave.sourceforge.net. Accessed: 2016-11-15.
[9] cadence, "SimVision." https://www.cadence.com/content/cadence-www/global/en_US/home/tools/system-design-and-verification/debug-analysis/simvision-debug.html. Accessed: 2016-11-16.
[10] Mentor Graphics, "EZWave." https://www.mentor.com/products/ic_nanometer_design/analog-mixed-signal-verification/ezwave/. Accessed: 2016-11-16.
[11] synopsis, "WaveView." http://www.synopsys.com/Tools/Verification/AMSVerification/DesignAnalysis/Pages/CustomWaveView.aspx. Accessed: 2016-11-16.