

Standalone Waveform Unfolding for the PDOM

John Kelley

PINGU Hardware Call

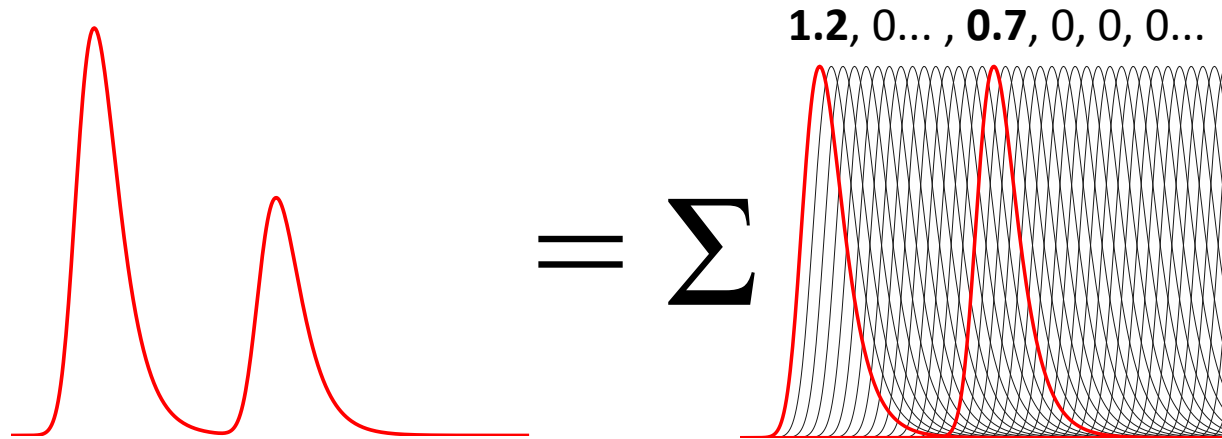
June 5, 2014

Overview

- IceCube runs feature extraction to convert raw waveforms to a pulse series
 - pairs of (charge, time) with given pulse templates
 - filtering clients calibrate data and run on all events
- This is the [WaveDeform](#) project in IceTray
- Goal: pull this code out of IceTray and get it running on a SOCKit dev kit
 - benchmarking / profiling... can we unfold SPEs at > 1 kHz?
 - power consumption
 - (eventually) hand off to a prototype DAQ

Waveform Unfolding

- Use time-shifted SPE pulse templates as a set of basis functions for a PMT waveform



- Non-negative least squares solution to matrix equation $\mathbf{y} = \mathbf{A}\mathbf{x}$
 - \mathbf{y} is the binned waveform as a column vector
 - \mathbf{A} is the basis matrix (pulse templates as columns)
 - \mathbf{x} is the best-fit unfolding (pulse amplitudes as column vector)
- Minimization code (Lawson-Hanson) can be used as-is from IceCube
- Some other tweaks
 - input waveforms are “road-graded” (bins below noise threshold set to zero)
 - saturated bins are not included

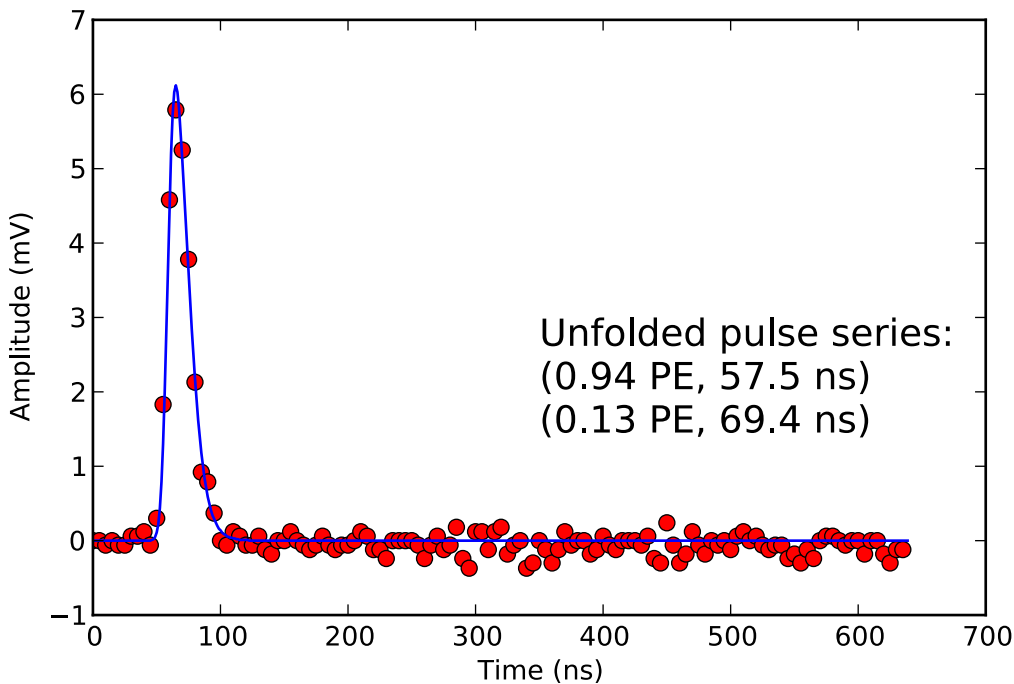
domwave

- <http://github.com/WIPACrepo/domwave>
- Standalone C99 port of WaveDeform
 - pull out all IceTray hooks
 - PDOM simplifications (single-channel front end)
 - add prototype pulse shape (courtesy K. Jero)
 - add command-line interface for testing
- Not done: cross-compilation for ARM

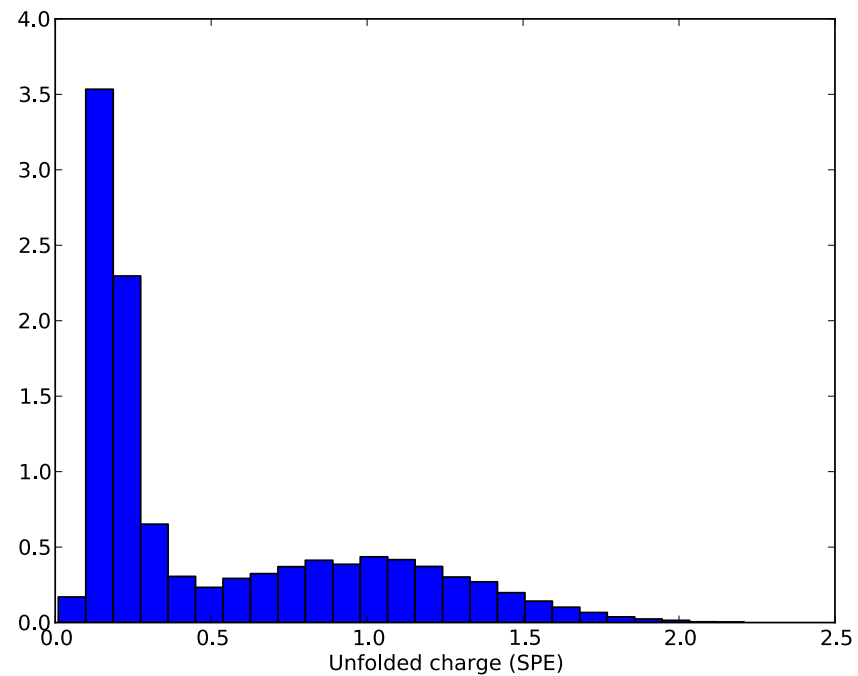
Sample Results

SPE data from early K. Jero lab setup (PMT + TI ADC + prototype firmware)

Pulse series and reconstructed waveform

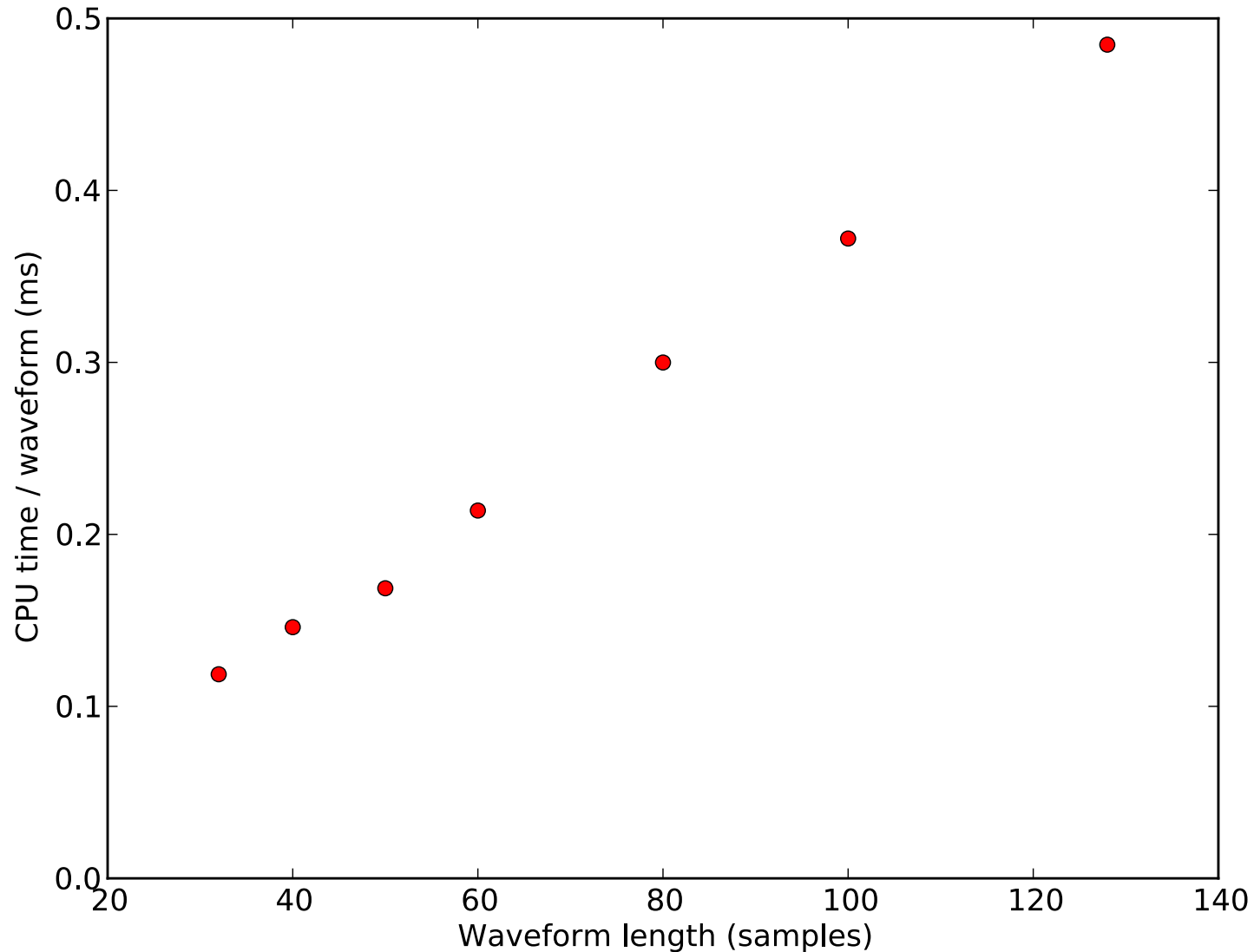


Charge histogram (SPE tuned to 1.0)



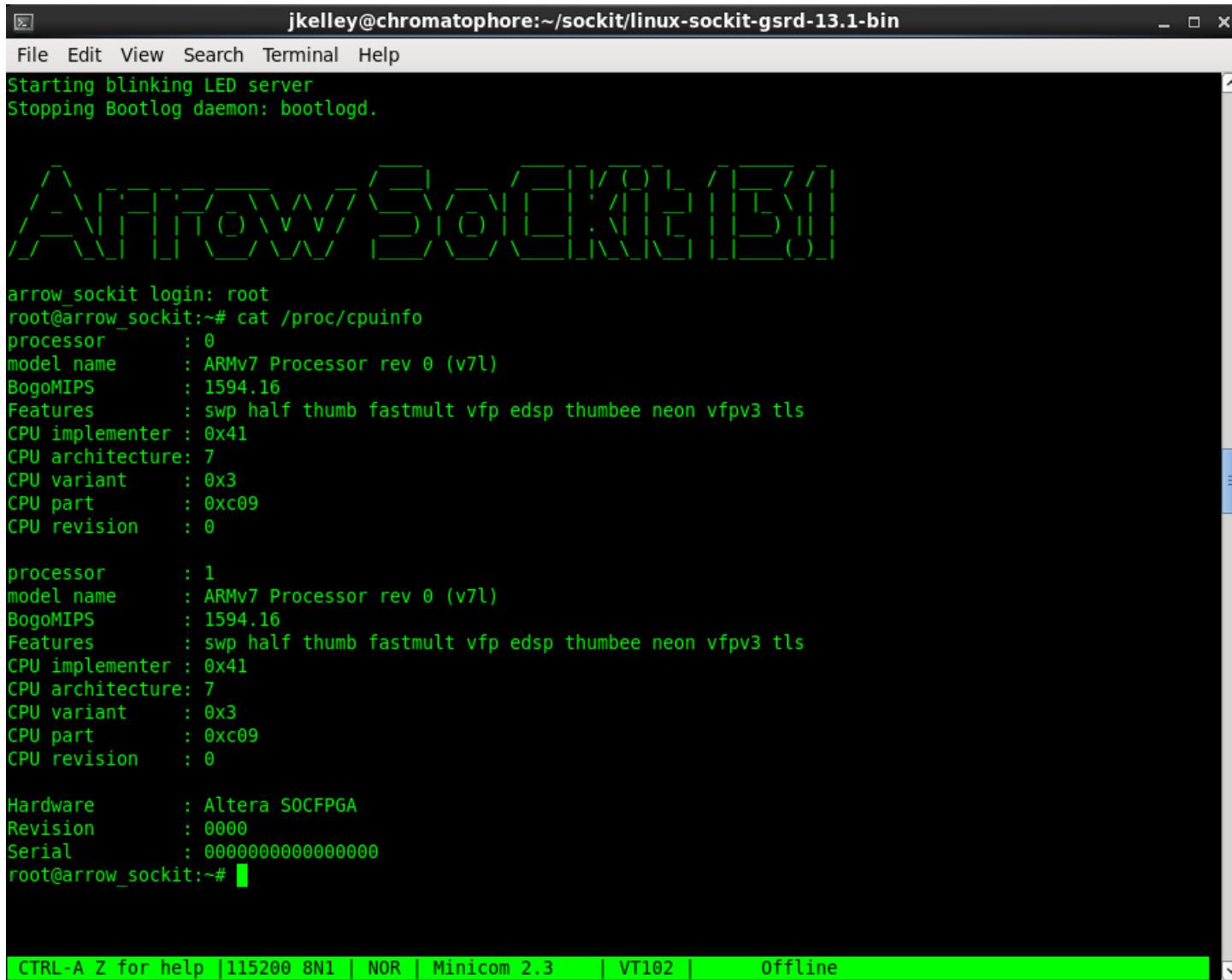
Initial Benchmarking (desktop)

Unfolding CPU time vs. waveform length (1.9 GHz AMD A6-5400)



Linux on the SOCKit (pre-built)

Excellent instructions at rocketboards.org



The screenshot shows a terminal window titled 'jkelley@chromatophore:~/socket/linux-socket-gsrd-13.1-bin'. The terminal output includes the following text:

```
Starting blinking LED server
Stopping Bootlog daemon: bootlogd.

Arrow SoC Kit

arrow_socket login: root
root@arrow_socket:~# cat /proc/cpuinfo
processor       : 0
model name     : ARMv7 Processor rev 0 (v7l)
BogoMIPS      : 1594.16
Features      : swp half thumb fastmult vfp edsp thumbee neon vfpv3 tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x3
CPU part       : 0xc09
CPU revision   : 0

processor       : 1
model name     : ARMv7 Processor rev 0 (v7l)
BogoMIPS      : 1594.16
Features      : swp half thumb fastmult vfp edsp thumbee neon vfpv3 tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant    : 0x3
CPU part       : 0xc09
CPU revision   : 0

Hardware      : Altera SOCFPGA
Revision      : 0000
Serial        : 0000000000000000
root@arrow_socket:~#
```

The terminal window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The status bar at the bottom shows 'CTRL-A Z for help', '115200 8N1', 'NOR', 'Minicom 2.3', 'VT102', and 'Offline'.

Cross-compiler Toolchains

- ARM compiler toolchains tried so far:
 - [Sourcery CodeBench Lite](#) (from Mentor; reasonably easy to get going, good instructions)
 - [Linaro toolchain](#) (worse documentation, but comes with libgfortran; using this now)
 - horrible licensing issues with ARM DS-5 compiler, ultimately unsuccessful
- To do: get domwave running on the dev kit!
 - I have OpenBLAS and LAPACK compiled...