Sump3
Mixed Signal Logic Analyzer
For ASICs and FPGAs
2023.08.30

Kevin M. Hubbard



Part-I : The Hardware

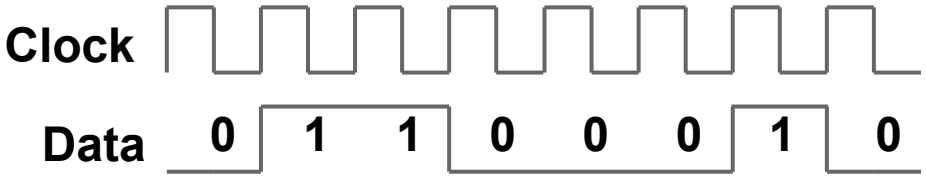
Sump vs traditional Logic Analyzers

Traditional ILA

Capture Length = RAM Depth x Clock Period

1K x 100 MHz = 10uS capture.

10mS capture would requires 1M of RAM.

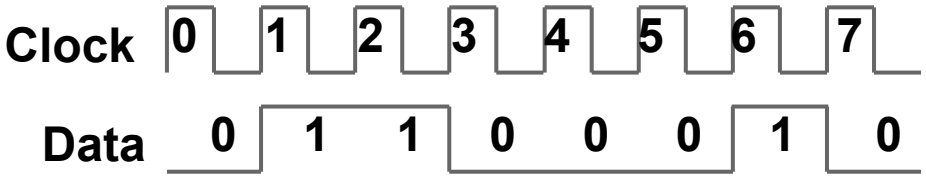


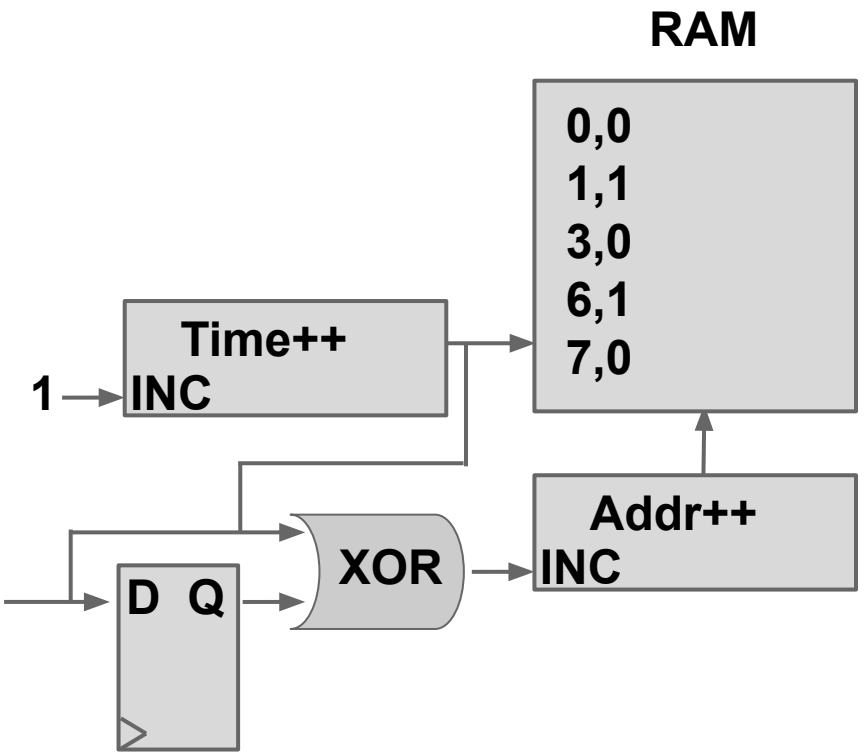
Sump uses real-time hardware compression and off-line software decompression.

From: https://en.wikipedia.org/wiki/Run-length_encoding
Run-length encoding (**RLE**) is a form of lossless data compression in which *runs* of data (sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most efficient on data that contains many such runs

Sump with Run Length Encoding

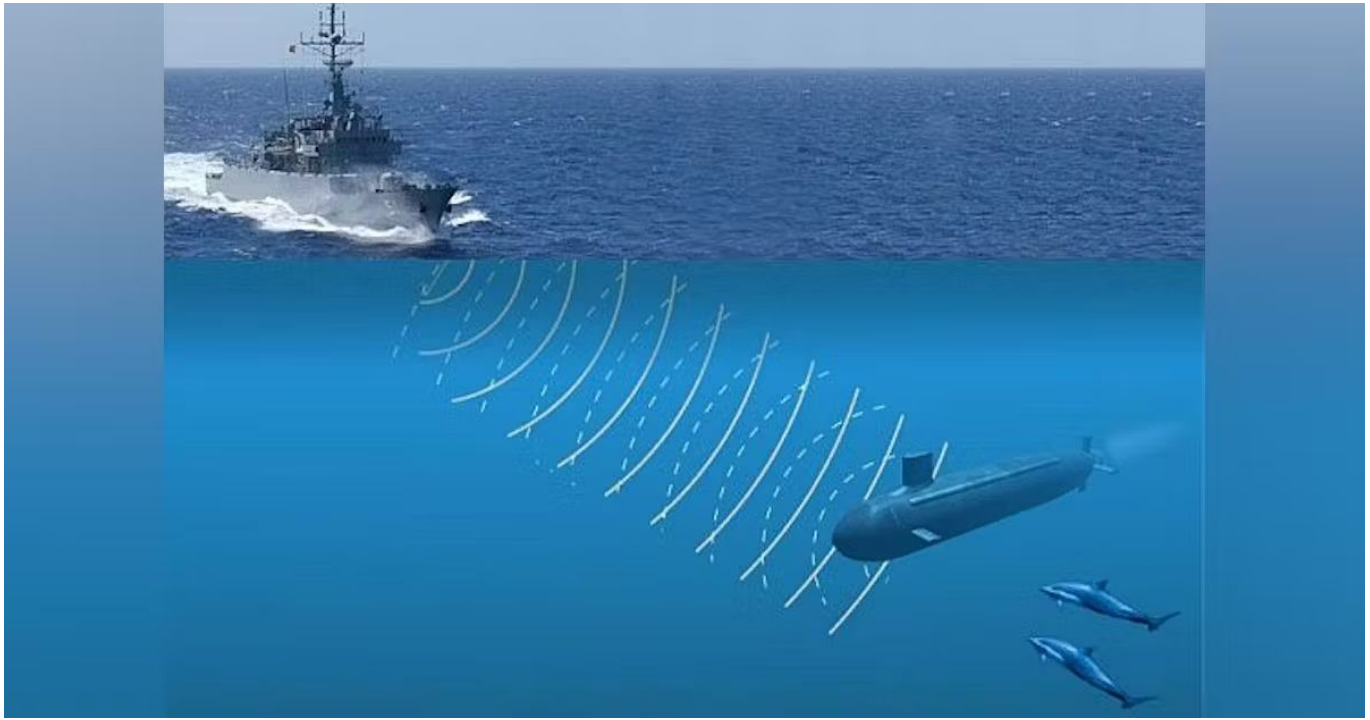
Capture Length = RAM Depth / (Number of Signals x Rate of Change)





RLE can provide 100x - 10,000x compression in some applications.

With 1K RAM at 100 MHz sampling, instead of just 10uS captures, 1 to 100 mS is possible thanks to hardware compression.



“I’ve been using Sump2 since 2016. It’s GREAT. Why Sump3? Why now? What the heck is wrong with Sump2? Don’t move my cheese”

- Anonymous Sump2 User

Sump2 was designed in 2016 for 90 nm FPGAs.

RAM was scarce in the 2010's. The AMD/Xilinx Spartan3 XC3S700A for example has only 20 1Kx36 BRAMs. Sump2 was designed to use a single BRAM for capturing ~1 mS of 32 signals. It does what it was designed to do, but it was never designed to scale up.

EETimes

MY CART

HOME NEWS PERSPECTIVES DESIGNLINES PODCASTS EDUCATION STORE SPECIAL RE

DESIGNLINES | PROGRAMMABLE LOGIC DESIGNLINE

$22 FPGA-Based Logic Analyzer Packs a Punch

By Max Maxfield 10.27.2016 0

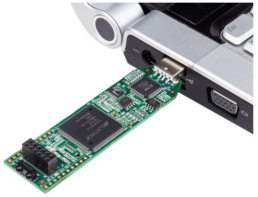
Share Post Share on Facebook Share on Twitter

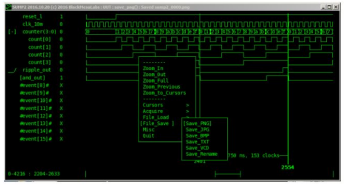
A couple of months ago, I wrote a column about how a team of Ultrasound Engineers recently sent an FPGA to 103,000 feet in a high-altitude balloon and just barely recovered it after three days in the wild (see *Madcap Ultrasound Engineers Send FPGA to 105,000 Feet*).

One member of that team was Kevin Hubbard, an Electrical Engineer with 20+ years designing ASICs and FPGAs for embedded systems. In addition to being an Ultrasound Engineer by day, Kevin founded Black Mesa Labs to develop open source hardware and software to explore new technologies and support the Maker Movement.

BLACK MESA LABS

SUMP2 – 96 MSPS LOGIC ANALYZER FOR $22





2016.22.22 : SUMP2 now also available for RaspPi+icoBoard here.

2016.12.13 : Great video on SUMP2 from Bil Herd at Hackaday.

2016.10.31: Thanx to SteveDC, a 3D printed case is now available for SUMP2 here.

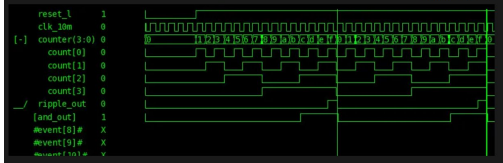
HACKADAY

HOME BLOG HACKADAY.IO TINDIE HACKADAY PRIZE SUBMIT ABOUT

AN OPEN SOURCE 96 MSPS LOGIC ANALYZER FOR $22

by: Jenny List 66 Comments

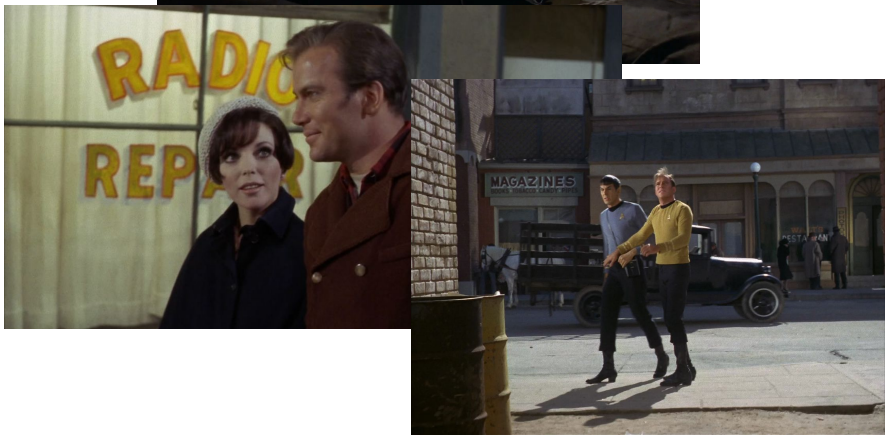
October 26, 2016



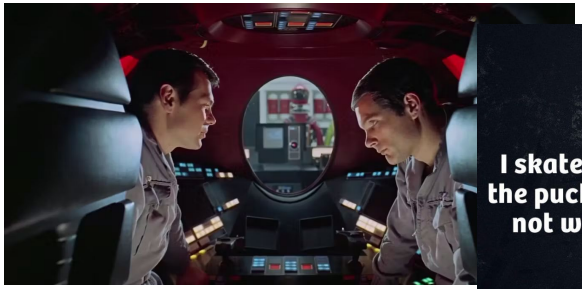
If you are in the market for an inexpensive USB logic analyser you have a several choices, but few of them deliver much in the way of performance. There are kits from China for a few dollars using microcontrollers at their heart, but they fail to deliver significant sample rates. If you require more, you will have to pay for it.

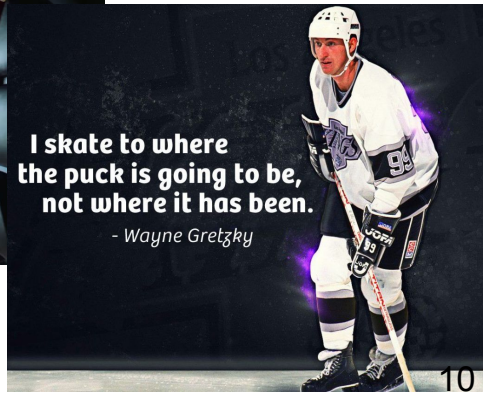
It is therefore rather interesting to see [kevinhub88]'s SUMP2 project, an open source logic analyser with a claimed 96 MSPS sample rate using an off-the-shelf Lattice iCEstick FPGA evaluation board that only costs about $20. ~~It talks to a host computer via USB using the established SUMP protocol, so its software front-end comes from the sump.org logic analyser project.~~ Edit: Since this post was published [Kevin] has contacted us to inform us that the project's capabilities have now moved beyond SUMP's capabilities and in fact it now uses his own software.

Sump2 is from an era of crude 90nm transistors drawn with sticks.



Sump3 is designed to scale to the future powered by 2nm FPGAs.





Sump3 is designed for 2nm FPGAs with abundant RAM.

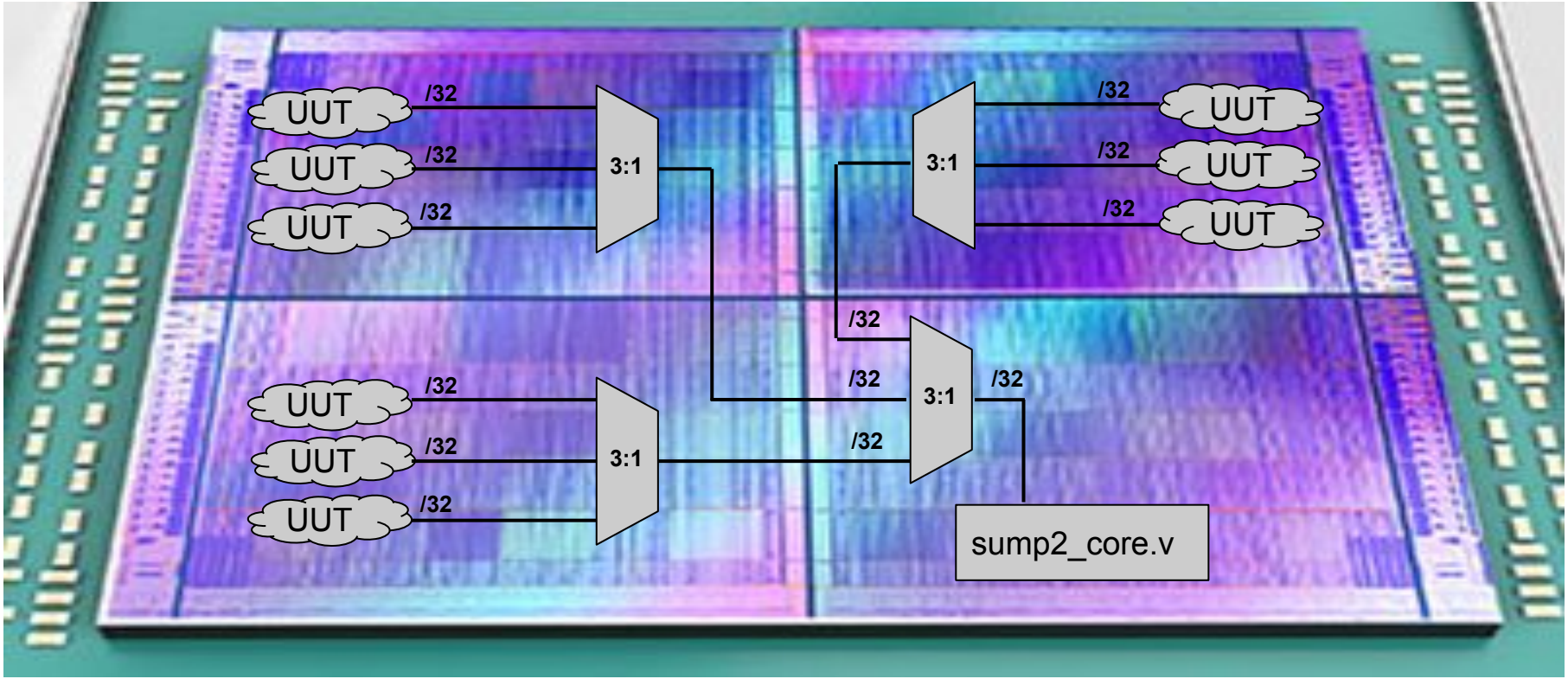
Even today's 16nm AMD/Xilinx VU9P with 3 chiplets has over 2000 1Kx36 BRAMs, 1000 8Kx36 UltraRAMs and 2 million D flip-flops.

FPGAs are no longer single pieces of silicon. New Stacked Silicon Interconnect (SSI) technology combines multiple FPGA "chiplets" onto a single interposer. Gate and RAM resources grow while inter-chiplet metal routing resources shrink. Timing closure and routing congestion are more critical than a handful of BRAMs.

Sump2 can only scale beyond 32 signals using muxes.

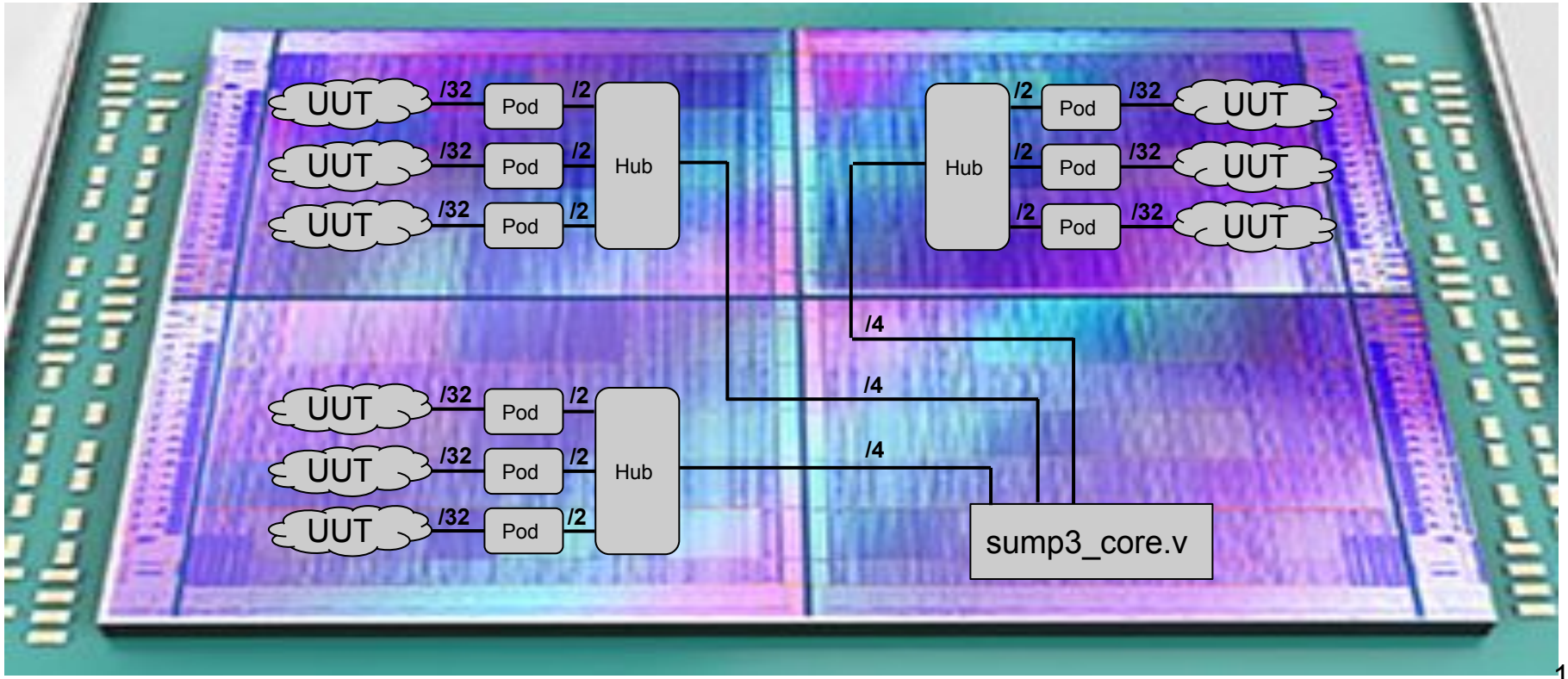
Multiple muxes results in cross chip/chiplet routing congestion.

Sump2 can only ever capture 32 RLE signals at once.



Sump3 uses distributed localized RLE Pods and Hubs.

Localized capture minimizes global routing impacts, improves timing closure and maximizes number of RLE signals captured at once and supports multiple clocks.



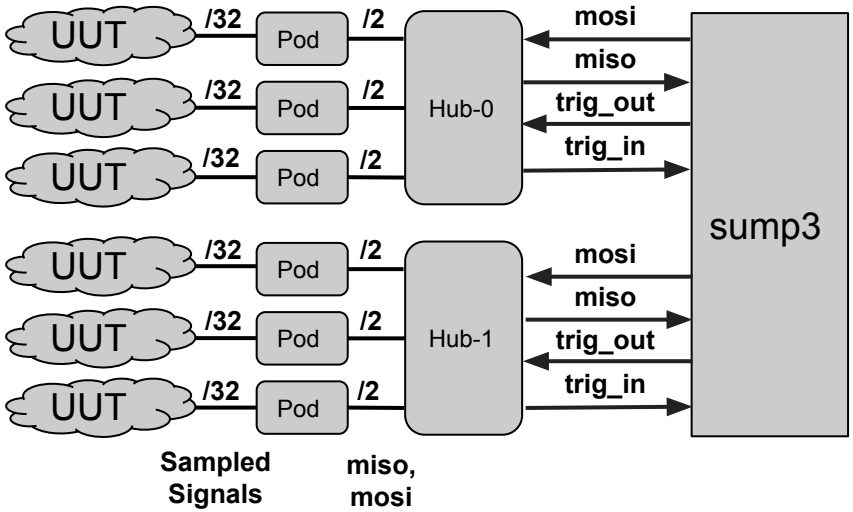
Sump3 may have 1-256 RLE Hubs.

Each RLE Hub may have their own clock domain.

A single RLE Hub may interface with 1-256 RLE Pods.

A single pod may have 4 to 8K signals.

Note: Only 1st 32 signals of each pod may be triggers.



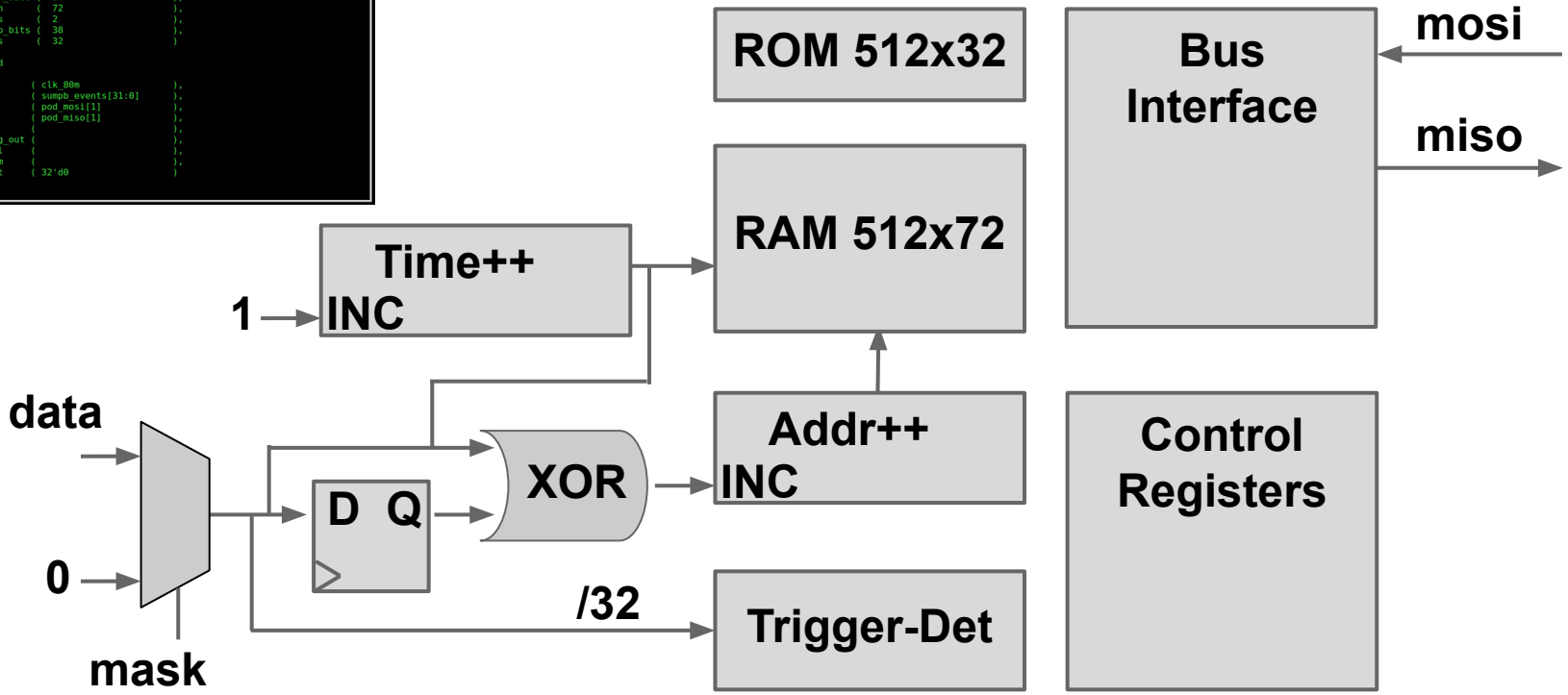
“No FPGA needs 256 clock domains!” - George Kennedy

Today’s FPGAs like the VU9P have more than a hundred SERDES transceivers. Each receiver CDR recovers a different clock, requiring its own unique regional clock tree. Although these clocks are often the same frequency, they all have different phase.

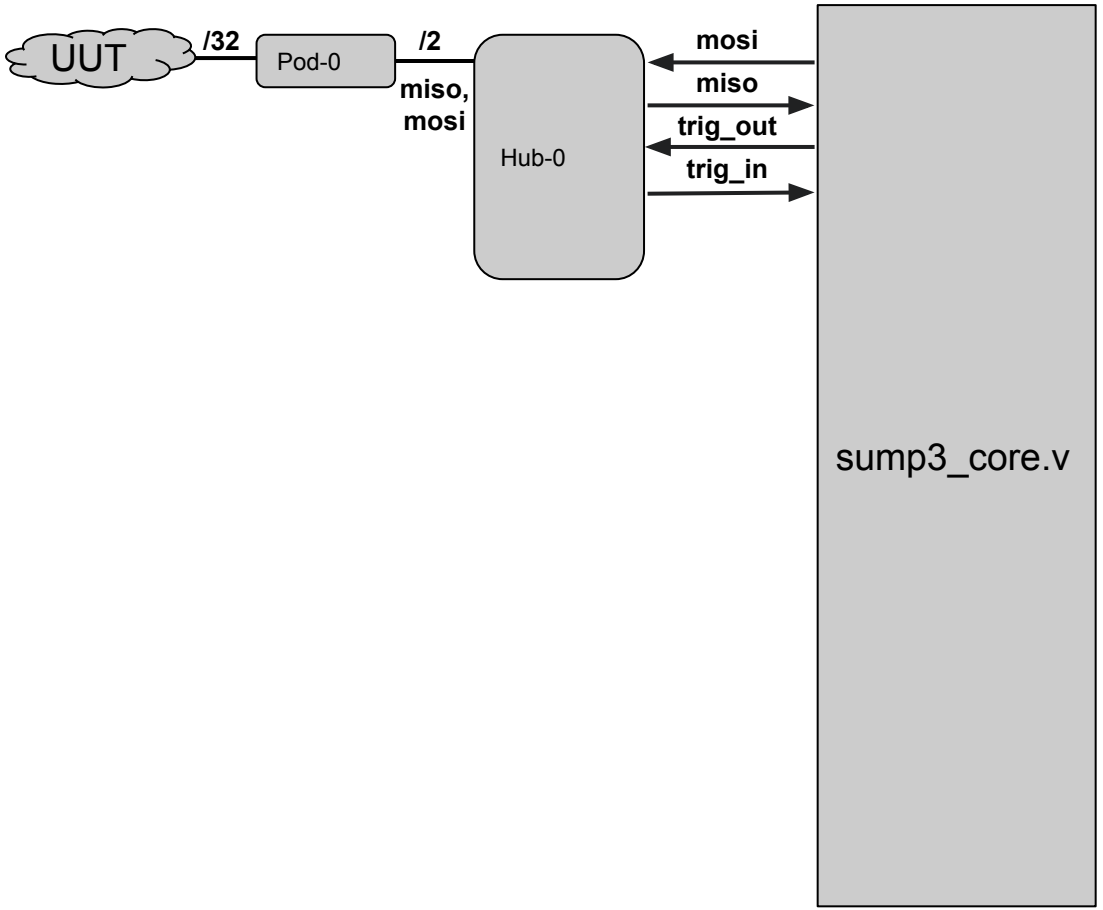
RLE Pod - Localized Capture

A small self contained RLE sample engine

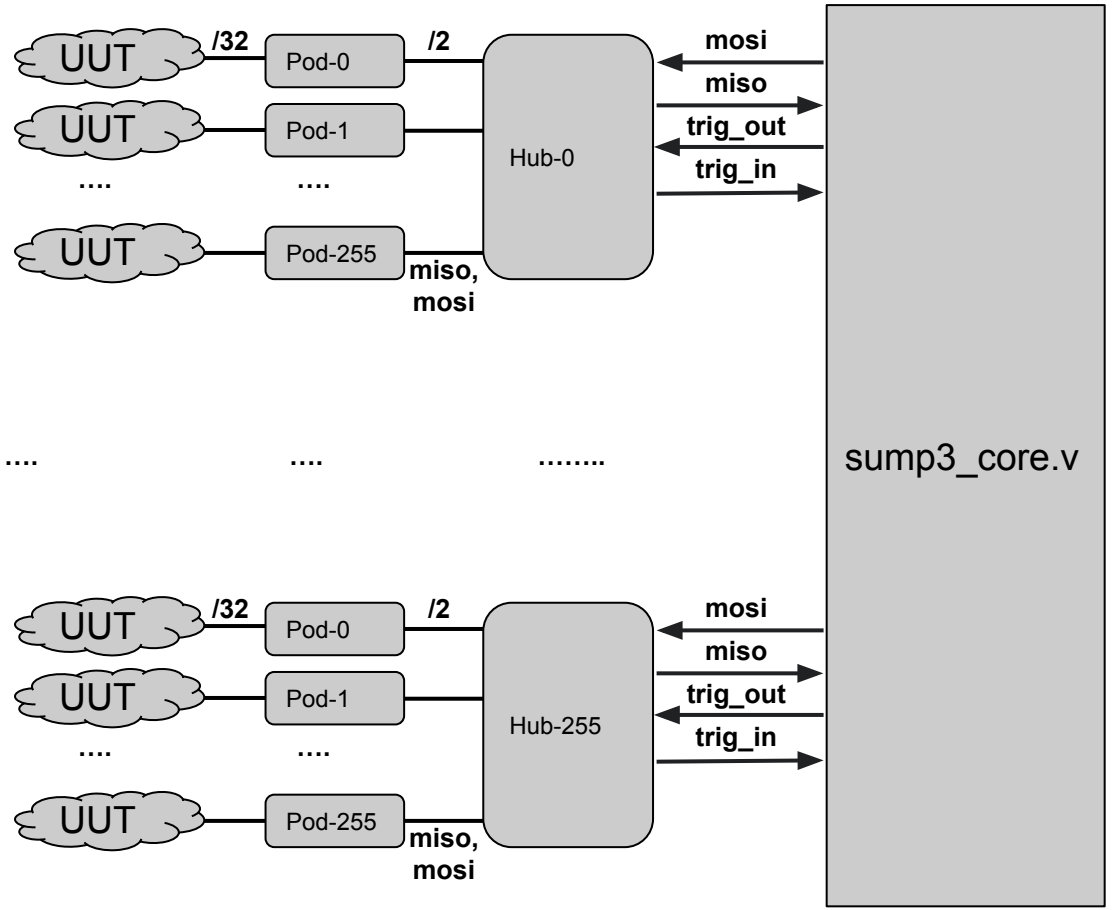
```
//------------------------------------------------------------------------------
// Single SUMP3 RLE Pod
//------------------------------------------------------------------------------
sump3_rle_pod
#
(
  .pod_name           ( "u1_pod       "         ),// 12 chars only
  .rle_ram_depth_len  (  512                    ),
  .rle_ram_depth_bits (  9                      ),
  .rle_ram_width      (  72                     ),
  .rle_code_bits      (  2                      ),
  .rle_timestamp_bits (  38                     ),
  .rle_data_bits      (  32                     )
)
u1_sump3_rle_pod
(
  .clk_cap            ( clk_80m                 ),
  .events             ( sumpb_events[31:0]      ),
  .pod_mosi           ( pod_mosi[1]             ),
  .pod_miso           ( pod_miso[1]             ),
  .pod_is_armed       (                         ),
  .pod_user_trig_out  (                         ),
  .pod_user_ctrl      (                         ),
  .pod_user_stim      (                         ),
  .pod_user_stat      ( 32'd0                   )
);
```



For small FPGAs, Sump3 can be as small as Sump2 ….



… yet still scale up to giant future FPGA devices.

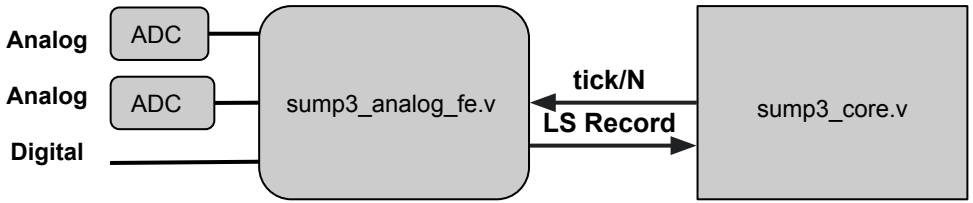


But wait - there's more.
Sump3 isn't just an RLE improvement over Sump2.

Sump3 is a Mixed Signal Analyzer.
It has slow analog and digital capture capabilities for capturing seconds to minutes of data.

Sump3 Low Speed (LS) Analog and Digital capture.

Programmable tick/N supports trade off between sample rate and total sample time.



Example: 256 points can either be:

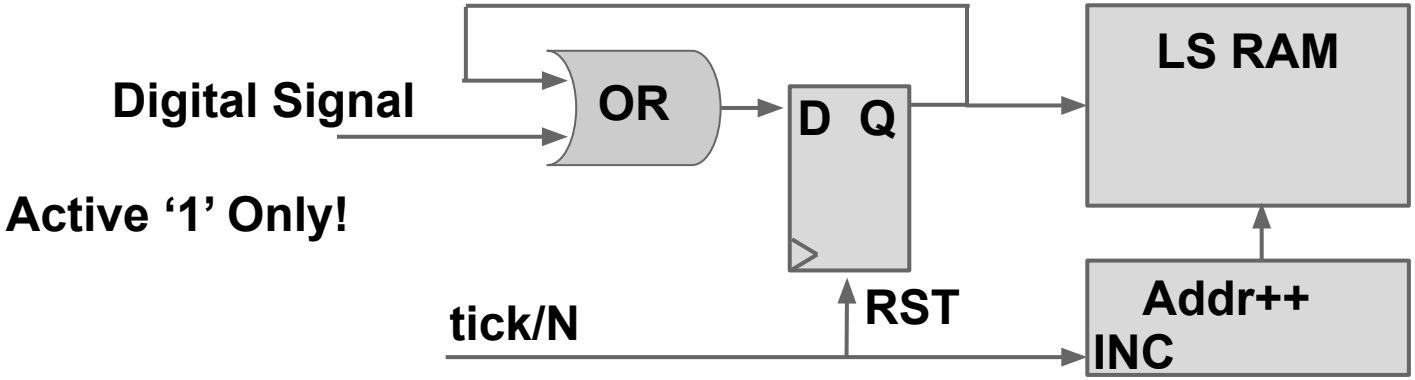
5 mS @ 20 uS Sample Intervals

25 mS @ 100 uS Sample Intervals

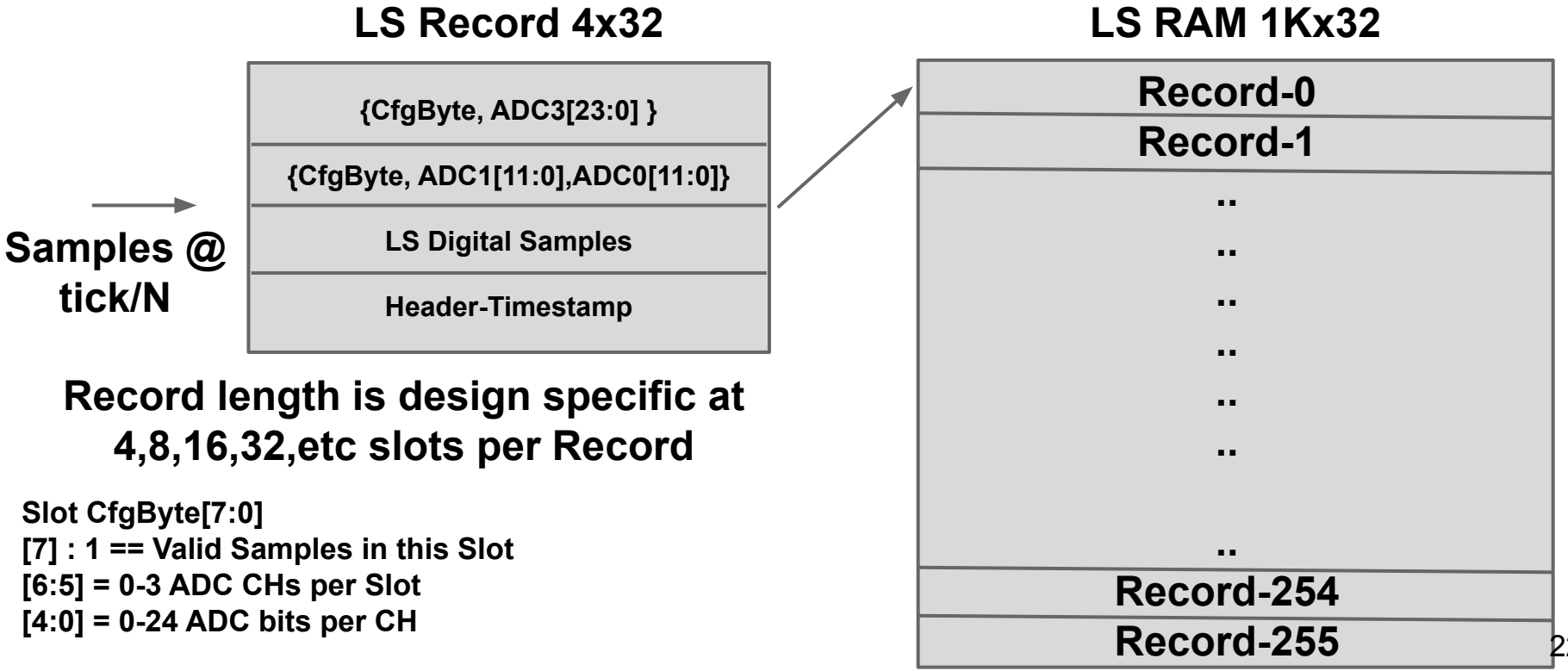
5 Seconds @ 20 mS Sample Intervals

25 Seconds @ 100 mS Sample Intervals

Low Speed (LS) Digital Sample and Hold



Low Speed (LS) Sample Record
Single RAM for Digital and MANY ADC Channels



Record length is design specific at 4,8,16,32,etc slots per Record

Slot CfgByte[7:0]
[7] : 1 == Valid Samples in this Slot
[6:5] = 0-3 ADC CHs per Slot
[4:0] = 0-24 ADC bits per CH

Part-I : The Hardware Questions and Answers

Part-II : The Software

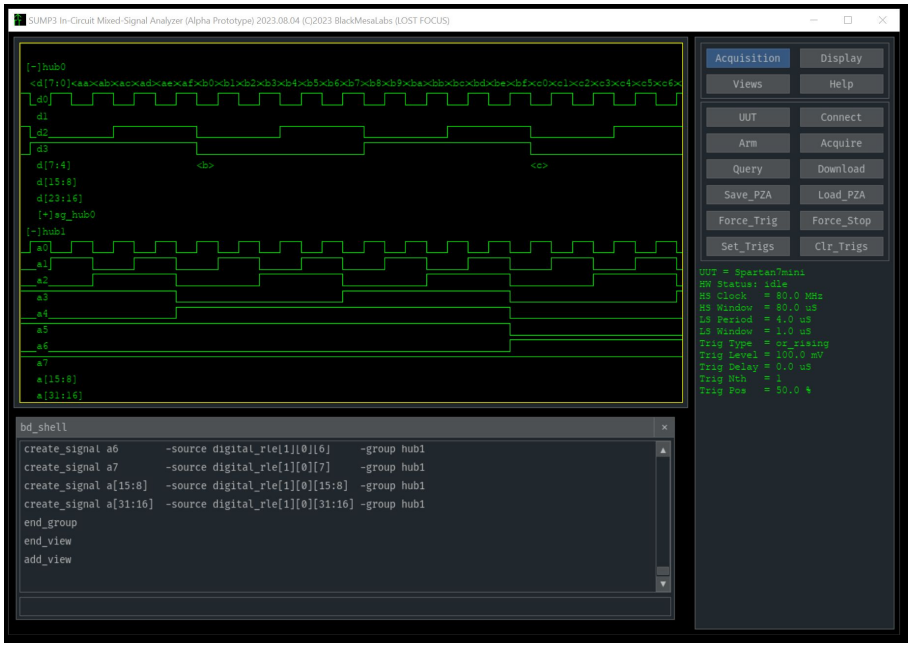
An entirely new GUI - yet still very much Sump

Written entirely in Object Oriented Python3

PyGame module still used for very fast GPU access.

PyGame-GUI module now used for button widgets.

pip.exe download pygame-gui

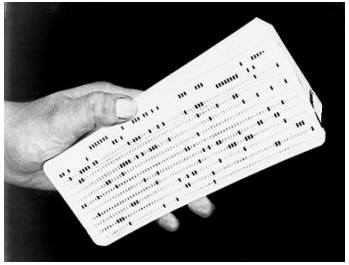


Sump3 is a UNIX EDA tool with a Command Line Interface (CLI) at its core.

GUI

bd_shell CLI

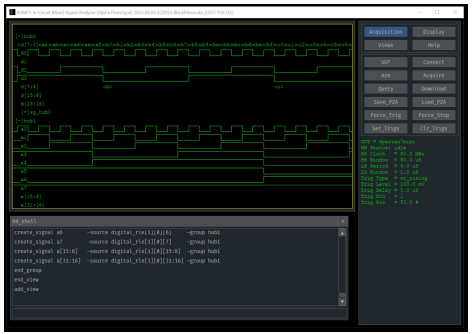
scripts



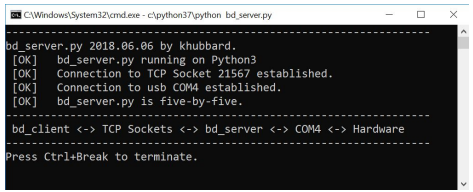
Under the hood Software Engine

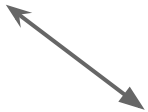
bd_server.py device driver to hardware

sump3.py GUI



bd_server.py



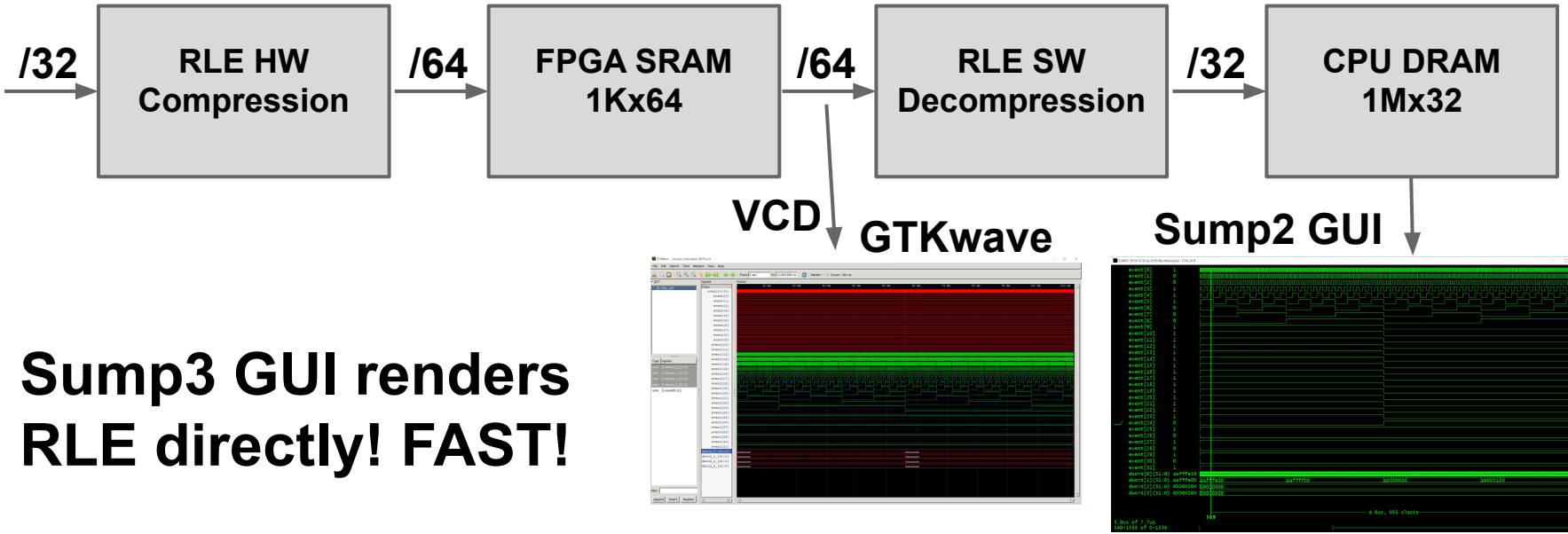


PHY

FPGA with sump3_core.v

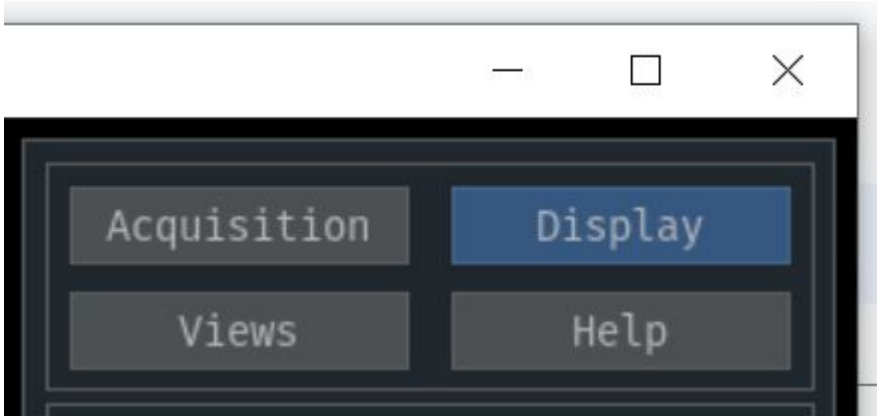
Berkeley Sockets (TCP/IP Ethernet,etc)

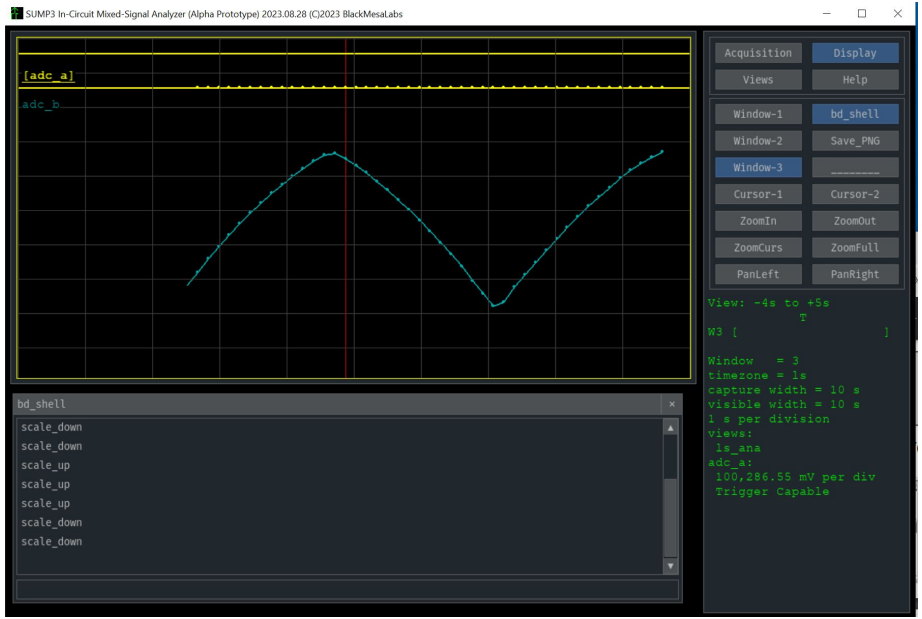
Foundation of the Sump2 GUI software was to download RLE samples and decompress them into discrete time samples. This required large amounts of CPU memory and GPU cycles. Above 10mS, rendering is VERY slow. Exporting VCD to GTKwave was a punt.



Sump3 GUI renders RLE directly! FAST!

3 Tabs of Operation : Acquisition, Views and Display

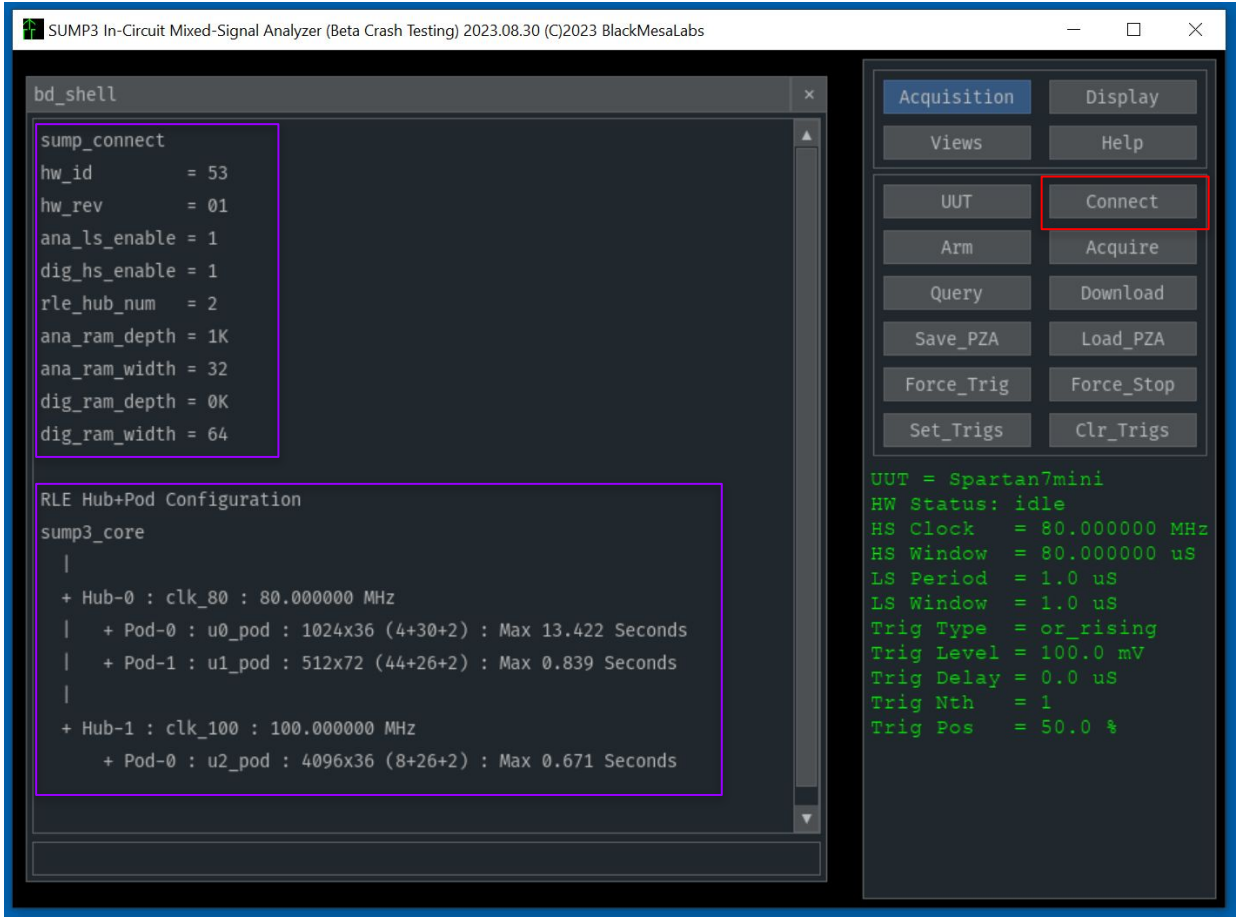




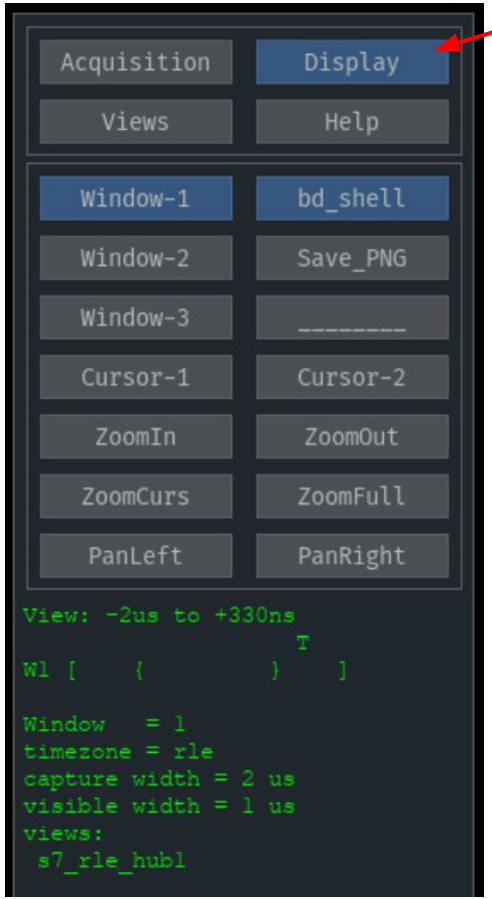
3 Tabs of Operation : Acquisition

[UUT] : Specify the Unit Under Test
[Connect] : Connect to UUT
[Arm] : Place UUT in Arm state
[Acquire] : Place UUT in Acquire state
[Query] : Query status of UUT
[Download] : Download RAM from UUT
[Save_PZA] : Save acquisition to PZA file
[Load_PZA] : Load acquisition from PZA file
[Force_Trig] : Force a software trigger
[Force_Stop] : Abandon Arm / Acquire
[Set_Trigs] : Specify Trigger signals
[Clr_Trigs] : Clear existing Triggers

sump_connect : Software queries the hardware



3 Tabs of Operation : Display



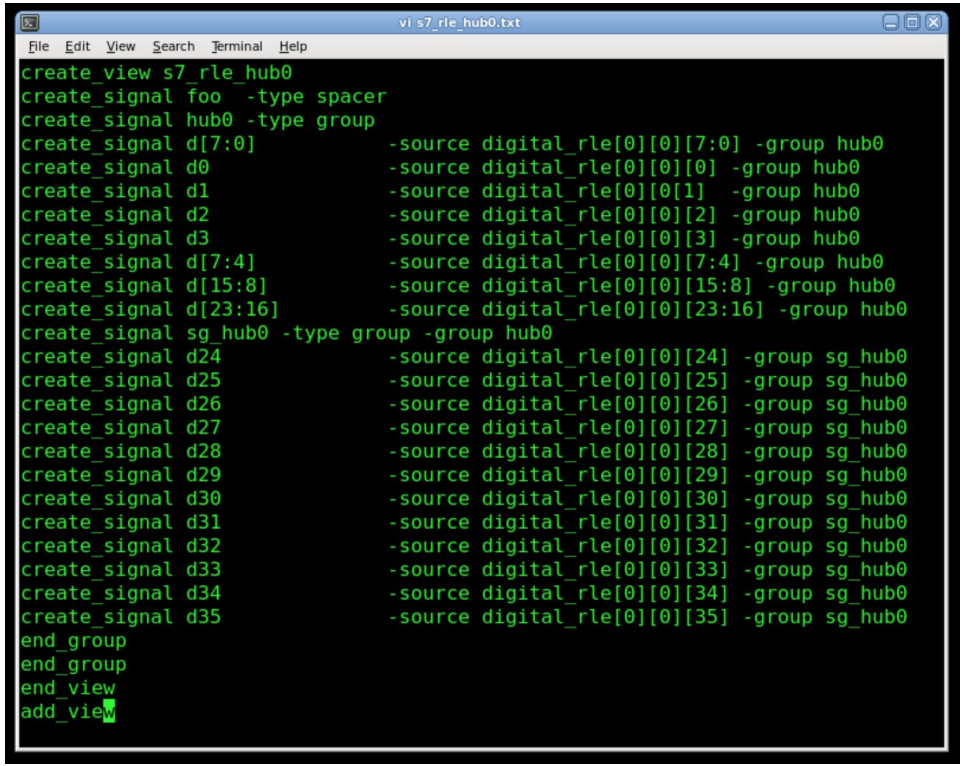
[Window-1] : Toggle Window-1 ON/OFF
[Window-2] : Toggle Window-2 ON/OFF
[Window-3] : Toggle Window-3 ON/OFF
[bd_shell] : Toggle bd_shell ON/OFF
[Save_PNG] : Save display to PNG file
[Cursor-1] : Toggle Cursor-1 ON/OFF
[Cursor-2] : Toggle Cursor-2 ON/OFF
[ZoomIn] : Zoom in
[ZoomOut] : Zoom out
[ZoomCurs] : Zoom to Cur-1 + Cur-2 regior
[ZoomFull] : Zoom full out
[PanLeft] : Pan Left in time
[PanRight] : Pan Right in time

3 Tabs of Operation : Views

[Apply] : Apply selected view from bottom available views list to selected Window.

[Remove] : Remove selected view from top applied views list.

View Files



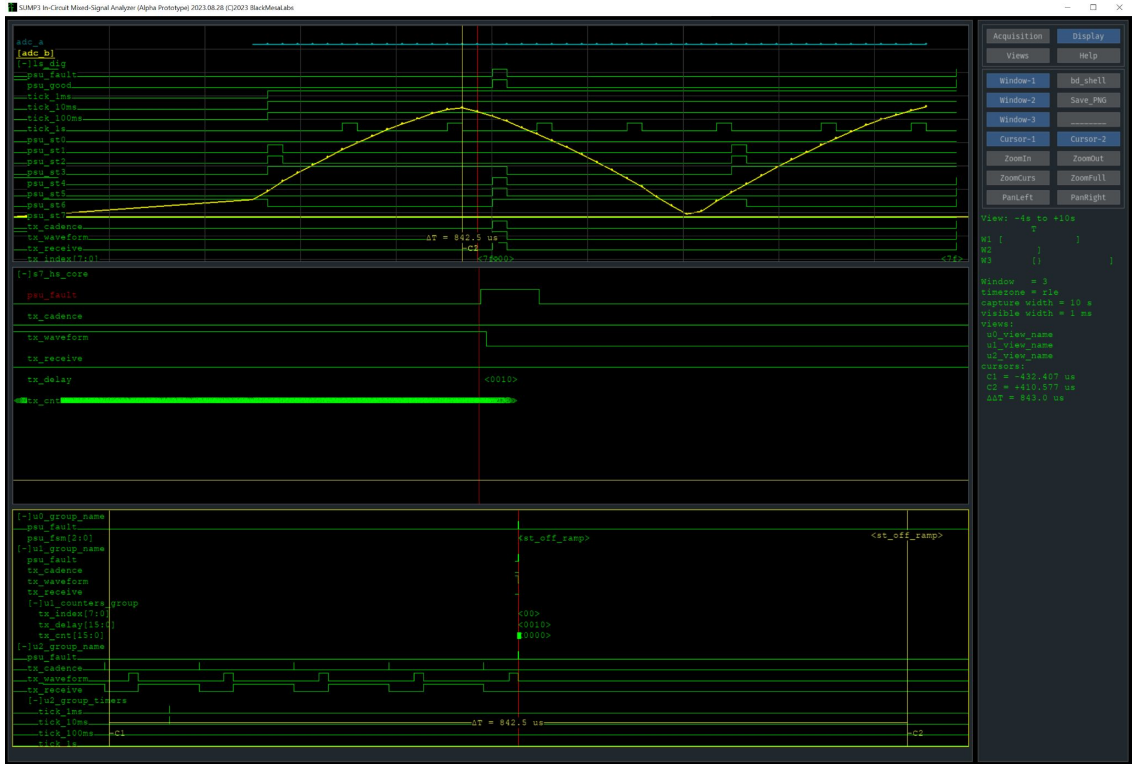
Sump3 can capture thousands to millions of signals, but computer display resolution limits how many binary nodes can be displayed at once. The sump3 “view file” is used for managing what is displayed and where. Similar to a sump2 “wave.txt” file, Sump3 supports rapidly applying and removing dozens of view files at once.

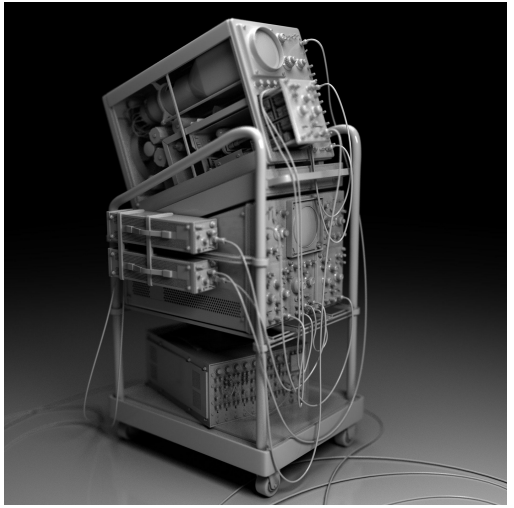
Plug-and-Play View ROM

Sump3's View ROMs solve Sump2's wave file FPGA revision issue. Now all signal name information is stored inside the FPGA and defined with a simple ASCII markup language in Verilog.

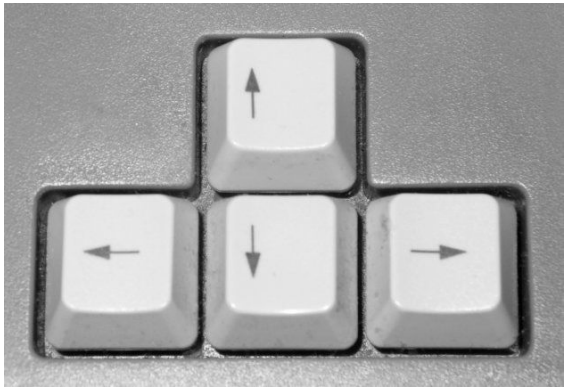
```
  .view_rom_txt       (
    {
     64'd0,                                    // Required 2-DWORD postamble
     8'hF0,                                    // ROM Start
     8'hF1, "u0_view_name",                    // Name for this view
     8'hF2,                                    // Signal source is THIS Pod
     8'hF5, "u0_group_name",                   // Make a top level group
       8'hF6, 16'd0,          "psu_fault",     // Bit
       8'hF7, 16'd3, 16'd1, "psu_fsm[2:0]",    // Vector
         8'hF8, 8'h00,        "st_reset",      // State Definitions
         8'hF8, 8'h01,        "st_idle",
         8'hF8, 8'h02,        "st_on_req",
         8'hF8, 8'h03,        "st_on_ramp",
         8'hF8, 8'h04,        "st_on_full",
         8'hF8, 8'h05,        "st_fault",
         8'hF8, 8'h06,        "st_off_ramp",
         8'hF8, 8'h07,        "st_unknown",
     8'hE5,8'hE2,8'hE1,8'hE0                   // End Group,Pod,View,ROM
    })
)
u0_sump3_rle_pod
(
  .clk_cap            ( clk_80m                 ),
```

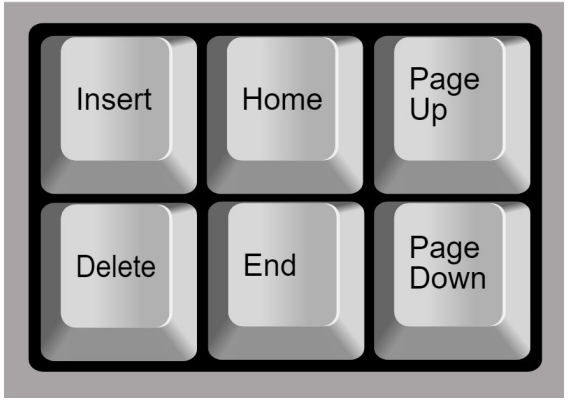
Sump3 GUI has 3 Windows as Sump3 HW is like three oscilloscopes that share a single BNC trigger. LS+HS+RLE.





Sump3 GUI Quick Keys :
Rapidly display only the signals you want.





Arrow Up/Down : Zoom In/Out
Arrow Left/Right : Pan Left/Right

Delete : Delete signal (push to delete stack)
Insert : Undo last deleted signal (pop stack)
Home : Undo ALL deleted signals (pop stack)
End : Toggle signal hidden attribute

Page-Up : Rotate Up Windows 1,2,3
Page-Down : Rotate Down Windows 1,2,3

Part-II : The Software Questions and Answers

Part-III : Conclusion and Demonstration

THE END

A Decade of Sump RLE Logic Analyzers

2014 : Sump1

- 1 clock domain
- 16 Triggers
- 16 HW Compressed Signals with 20 bit RLE
- 88 non-Compressed Signals
- Configurable RAM depth
- Software - C# Windows .NET based GUI

2016 : Sump2

- 1 clock domain
- 32 Triggers
- 32 HW Compressed Signals with 24/32 bit RLE
- Up to 512 non-Compressed Signals
- Configurable RAM depth
- Text file configuration
- Software - Python PyGame based GUI

2018 : DeepSump for Sump2

- 2nd RLE compression block for Sump2 with spurious signal detection
- VCD Exportation to GTKwave/Modelsim for captures > 1mS

2023 : Sump3

- 1-256 clock domains
- 1 to 2 Million Triggers
- 4 to 500 Million HW Compressed Signals with N-bit RLE
- Up to 4,000 non-Compressed Signals
- Configurable RAM depth
- Plug-and-Play configuration
- Analog ADC and Digital Sample and Hold at Tick/N rate
- Software - Python PyGame-Gui based GUI

Hardware Comparison	Sump2	Sump3
Target FPGA Technology	90nm	2nm
Inferrable Verilog Generic-IP	Yes	Yes
Nth,Delayed, Multi-Trig	Yes	Yes
RLE Compression	Yes	Yes
Analog Capture	No	Yes
Digital Slow Sample+Hold	No	Yes
Clk/N Slow Capture of seconds	No	Yes
Small	Yes	Scalable
Scalable	No	Yes
Non-Intrusive Routing Impact	No	Yes
Maximum Clock Domains	1	256
Maximum RLE Bits	32	Millions
Maximum non-RLE Bits	512	4096
Two DWORD Virtual PCIe Xface	Yes	Yes
Plug-N-Play SW Configuration	No	Yes*

Software Comparison	Sump2	Sump3
PC+Linux Multi-Platform	Yes	Yes
Python-FOSS	Yes	Yes
GPU Interface Library	PyGame	PyGame-GUI
Direct RLE Fast Rendering	No	Yes
Multi-TimeZone Windows	No	1-3
Analog Waveform Display	No	Yes
Integrated BD_Shell	No	Yes
Rapid View Changes	No	Yes
Hardware Interface	bd_server	bd_server
Remote TCP/IP Access	Yes	Yes
External Software Control	No	Yes
Multi-Chip System Access	No	Yes
VCD Export	Yes	No*
Offline Mode	No	Yes
Golden Capture Compare	No	Yes*
On-Demand Download	No	Yes*
Plug-N-Play HW Configuration	No	Yes*

Development Schedule

1) Feature Complete Hardware.
2) Fully test new RLE HW+SW. Fix any bugs.
3) Fully test depopulated LS,HS,RLE blocks.
4) Verify Multi-Clock Domain Trigger alignment.
5) GitHub upload.
6) Add new software features:
   - 6a) View ROM to View File Generation
   - 6b) FSM State Name rendering
   - 6c) On-Demand RLE Downloads
   - 6d) Golden Capture Compare (PZA Source)
   - 6e) /,? Signal Transition Search
   - 6f) VCD Export

EOF