CPEN 311: Digital Systems Design

Introduction to Lab 4

2015/2016 Term 1

Instructor: Steve Wilton

stevew@ece.ubc.ca

Learning Objectives for Lab 4

1. Learning how to read code
2. Learning how to combine datapath and controller in one process
3. Learning how to use SignalTap II embedded logic analyzer
4. Practicing Fixed Point Arithmetic
5. Getting more practice with designs containing a state machine / controller

Task 1: Download and Run the code

Task 2: Understand the Code by Reading

Task 3: Using SignalTap II to observe the circuit in action (2 marks)

Task 4: Easy Changes (1.5 marks)

Task 5: Shrinking the paddle

Task 6: Fixed Point

Task 7: Gravity

Debugging large designs using simulation:

Simulation can be A BILLION TIMES slower than real chip execution

Suppose our design contains a processor which will run Windows.

- Just to boot Windows would require 3000 years in simulation!
- The Romans would have had to start it running….

It is *impossible* to cover all operations in simulation

Intel Pentium 4 Verification/Validation[1]:

- 6,000 CPUs, running simulations, 24/7 for 2 years produced less than 1 minute of real time operation
- 60 person-year formal verification effort found about ~20 high-quality bugs
- 2^{10441} distinct configurations in a "simple" out-of-order X86 processor model, but only 2^{37} were covered in verification
- 10 months of validation from first-silicon to release-to-production (by comparison Intel aims at a new processor released every year…)

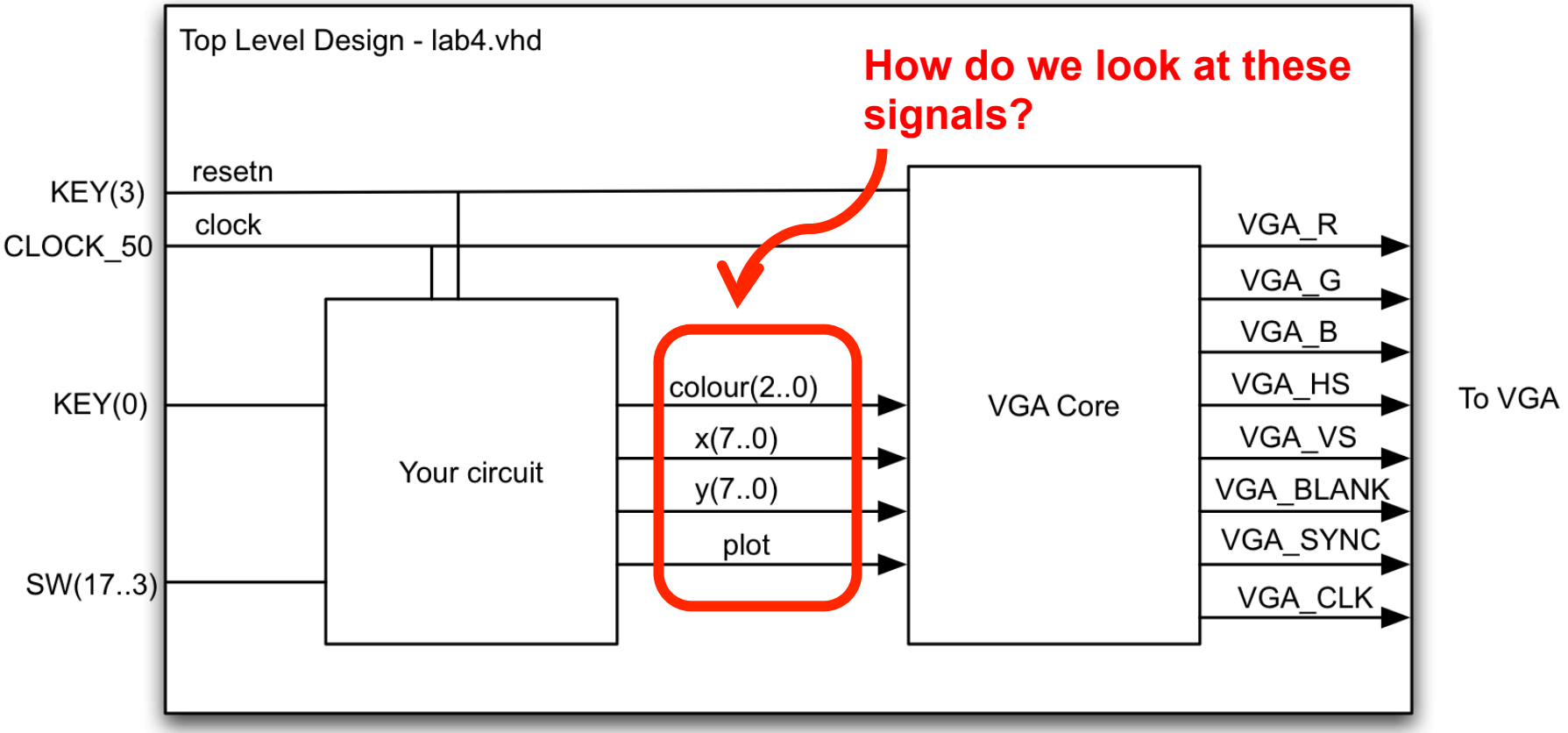
[1]B. Bentley, "Validating a modern microprocessor", Proc. Int. Conf. CAV

To effectively debug, we need to run chip at-speed.

- Allows for much more complete tests
- Allows for testing connected to real-world I/O

Problem, how do we look at internal signals?

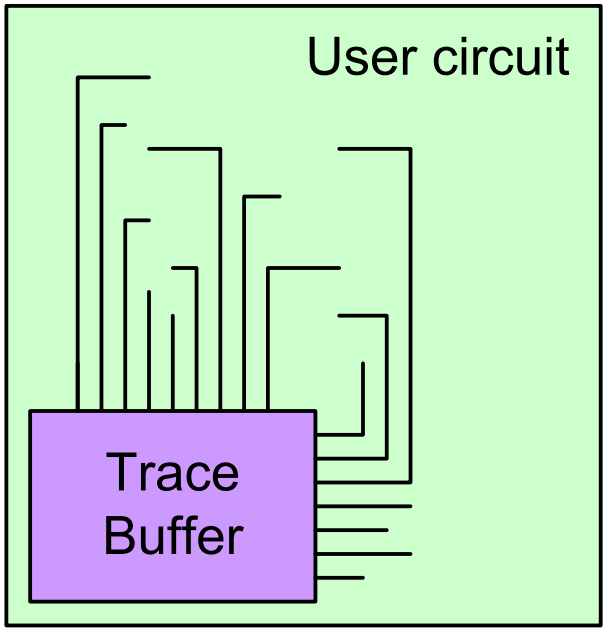
- Bringing them out to pins is a lot of work, not necessarily effective



Trace Buffers: Embedded Logic Analyzers

Store "key" signals in an on-chip memory

- This allows the chip to run at speed
- After the chip runs, you can read out from the trace buffer and analyze off-line



Some issues:

- Need to decide which signals to monitor
- Need a mechanism to "trigger" (stop recording)
- Could use compression to store more data
- What about multiple clock domains?

These sorts of embedded logic analyzers are included as part of FPGA design tools:

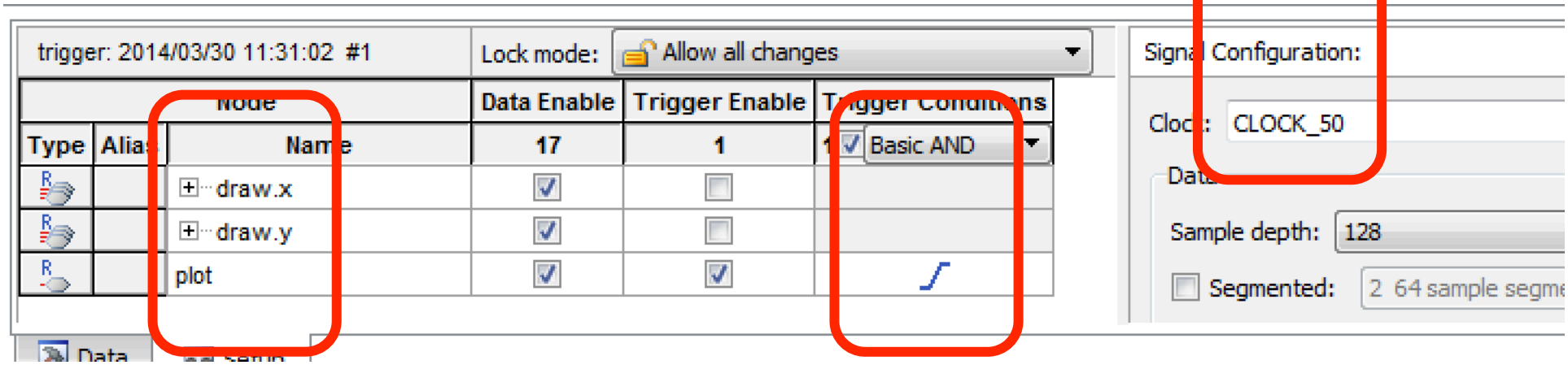
Altera -> SignalTap II
Xilinx -> ChipScope

There are third party tools as well (Certus from Mentor)

Altera SignalTap II:

Clock to align samples to

Set signals you want to look at, triggers, and clock to sample on:

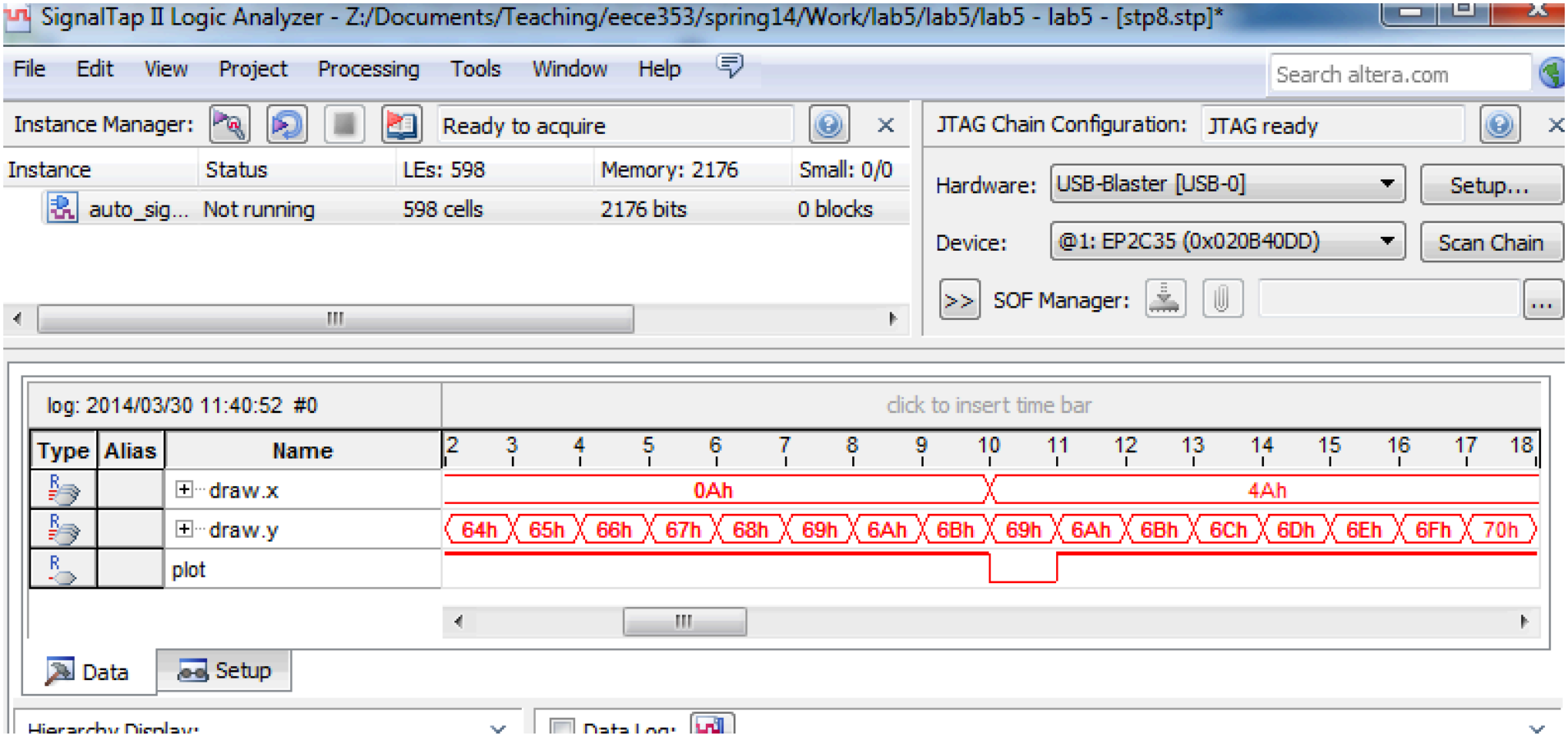


Signals I want to trace

Trigger Conditions (when to record)

Then recompile, and logic analyzer will be included in your design

- Uses logic elements and memory blocks just as any other circuit


SignalTap II Logic Analyzer - Z:/Documents/Teaching/eece353/spring14/Work/lab5/lab5/lab5 - lab5 - [stp8.stp]*
File Edit View Project Processing Tools Window Help
Search altera.com
Instance Manager:
Ready to acquire
Instance Status LEs: 598 Memory: 2176 Small: 0/0
auto_sig... Not running 598 cells 2176 bits 0 blocks
JTAG Chain Configuration:
JTAG ready
Hardware: USB-Blaster [USB-0]
Setup...
Device: @1: EP2C35 (0x020B40DD)
Scan Chain
SOF Manager:
log: 2014/03/30 11:40:52 #0
click to insert time bar
Type Alias Name
draw.x
draw.y
plot
0Ah
4Ah
64h 65h 66h 67h 68h 69h 6Ah 6Bh 69h 6Ah 6Bh 6Ch 6Dh 6Eh 6Fh 70h
Data
Setup

Aside: Commercialization: Veridae / Tektronix / Mentor

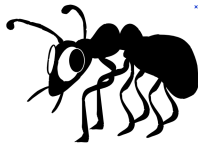
Dec 2005: First Publication:

B.R. Quinton, S.J.E. Wilton, "Post-Silicon Debug Using Programmable Logic Cores"

Dec 2009: Brad graduates, we decide to form Veridae Systems to develop "logic analyzer on a chip"

Veridae

May 2010: Run-down office infested with *real* bugs, 4 developers



Nov 2010: First suite of tools released

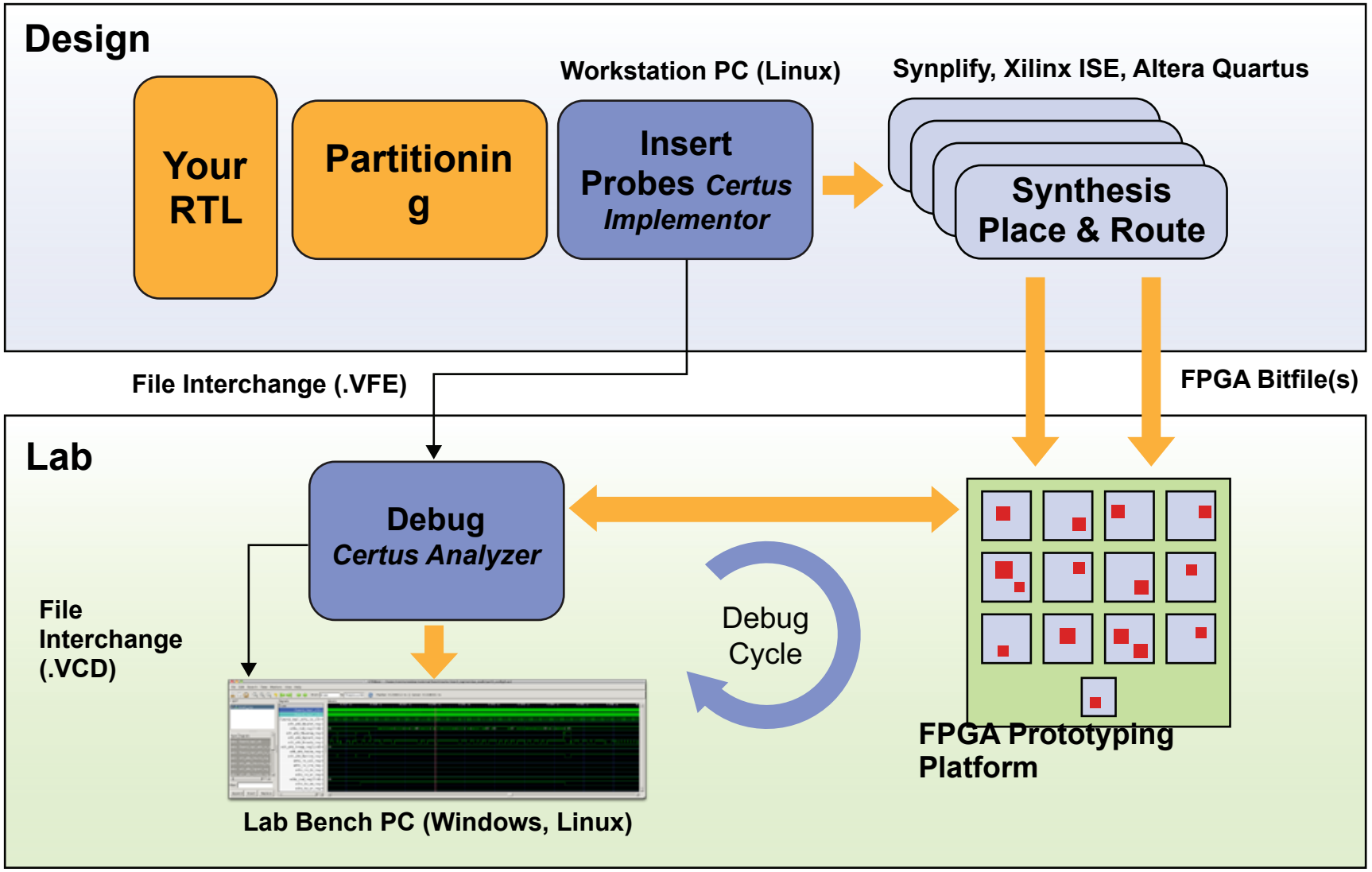
July 2011: Acquired by Tektronix, team stays in Vancouver

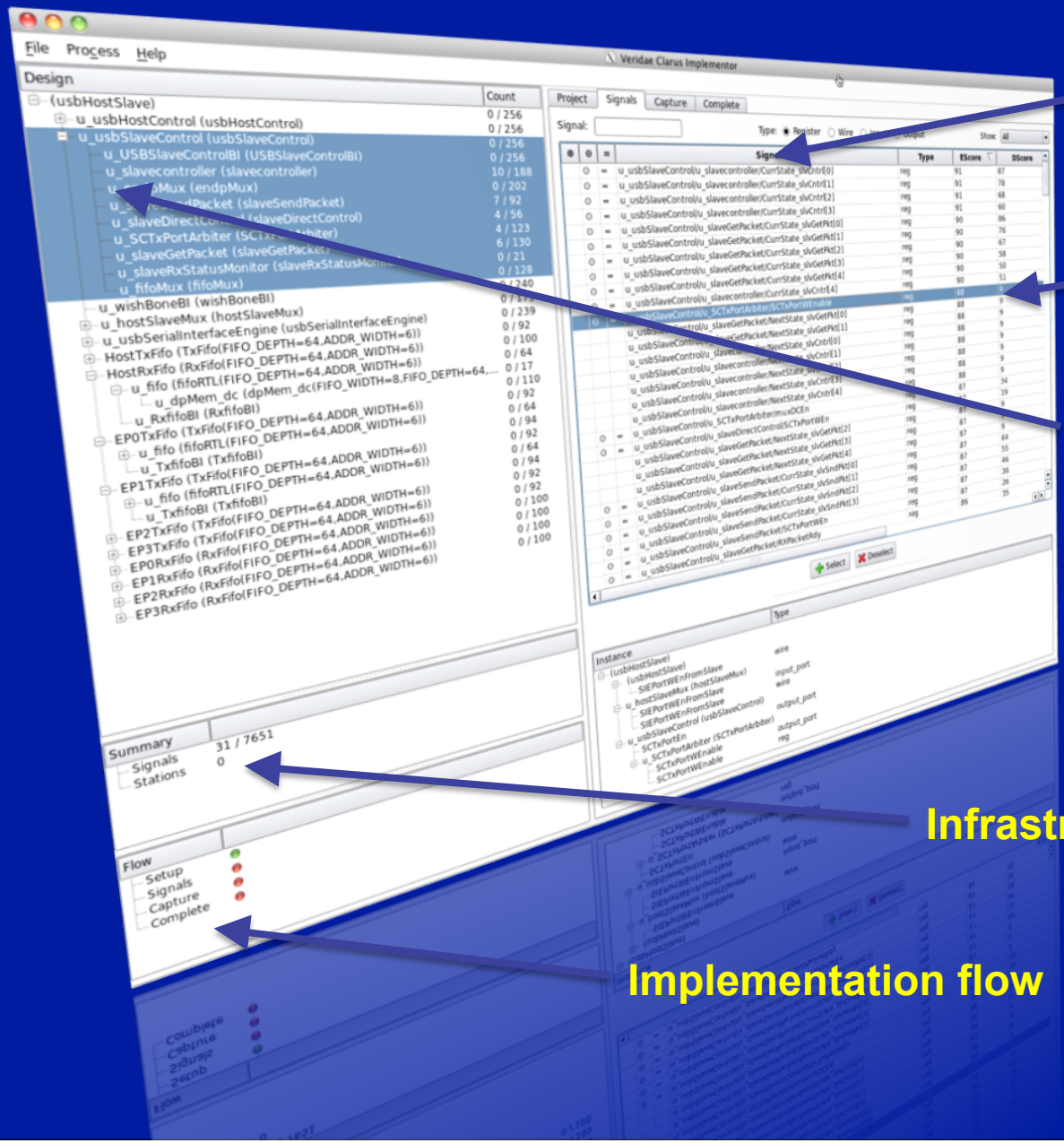
Tektronix®

Nov 2012: Version 2 of Certus tool released

Spring 2013 : Technology transferred to Mentor Graphics

Commercialization: Veridae / Tektronix / Mentor





Choose your signals

OptiScore

Your design hierarchy

Infrastructure summary

Implementation flow

"An engineer is as only as good as his/her tools…"

Should be:

"An engineer is only as good as the tools he or she knows how to use…"

There are a lot of advanced features of design tools, and they are worth getting to know. In this slide set, we saw two, but there are a lot more. Lots of information and tutorials on the web.