Chronopixe first prototype tests

J. E. Brau, N. B. Sinev, D. M. Strom
University of Oregon, Eugene

C. Baltay, H. Neal, D. Rabinowitz
Yale University, New Haven

EE work is contracted to Sarnoff Corporation

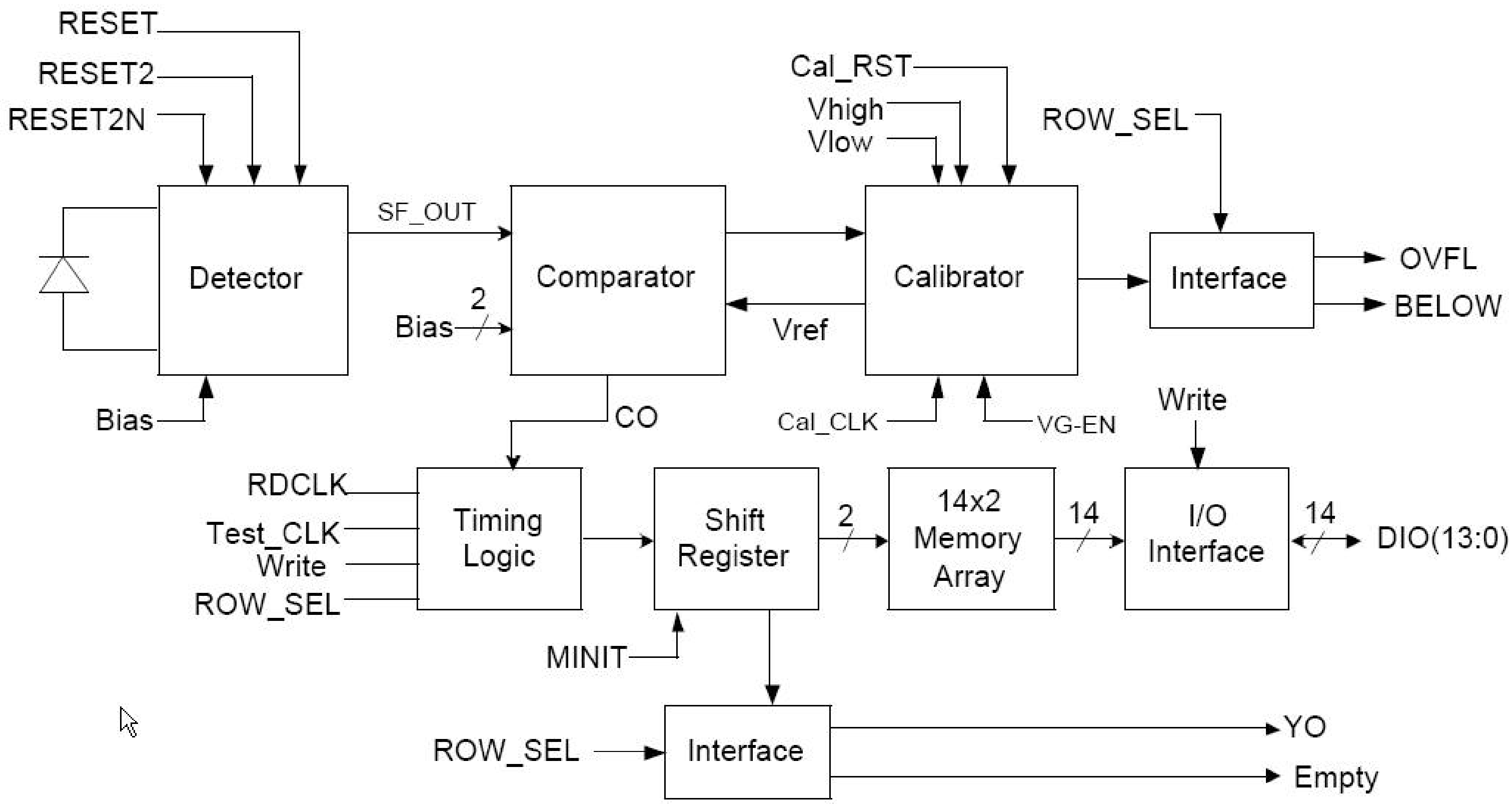
Outline of the talk

- Recall how chronopixel works
- Milestones
- Test stand design
- Test Stand software
- Test plans
- Test results
- Next steps
- Conclusions

How Chronopixel works

- When signal generated by particle crossing sensitive layer exceeds threshold, snapshot of the time stamp, provided by 14 bits bus is recorded into pixel memory, and memory pointer is advanced.
- If another particle hits the same pixel before device readout was completed, second memory cell is used for this event time stamp.
- During readout, pixels which do not have any time stamp records, generate EMPTY signal, which advances IO-MUX circuit to next pixel without wasting any time. This speeds up readout by factor of about 100.
- Comparator offsets of individual pixels are determined in the calibration cycle, and reference voltage, which sets the comparator threshold, is shifted to adjust thresholds in all pixels to the same signal level.
- To achieve required noise level (about 25 e r.m.s.) special reset circuit (soft reset with feedback) was developed by Sarnoff designers. They claim it reduces reset noise by factor of 2.

Simplified Chronopixel Schematic



Essential features: Calibrator, special reset circuit

Calibration procedure

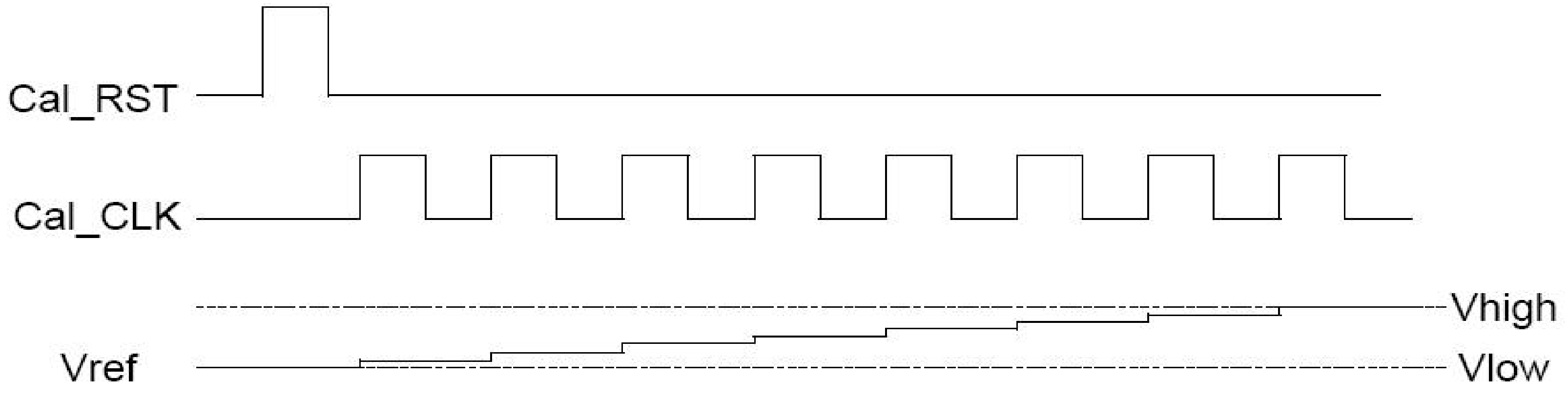


Figure 10.3 Timing diagram showing the calibrator operation

- During calibration, comparator **reference** voltage **changes** from Vlow to Vhigh **in 8 steps**, controlled by Cal_CLK clock pulses. As soon as it **reaches** the value when **comparator flips**, state of the clock counter is **recorded** into **calibration register** – individual for each pixel. During normal operation this **register** is **used** to set comparator **offset** for a given pixel.

Sensor design

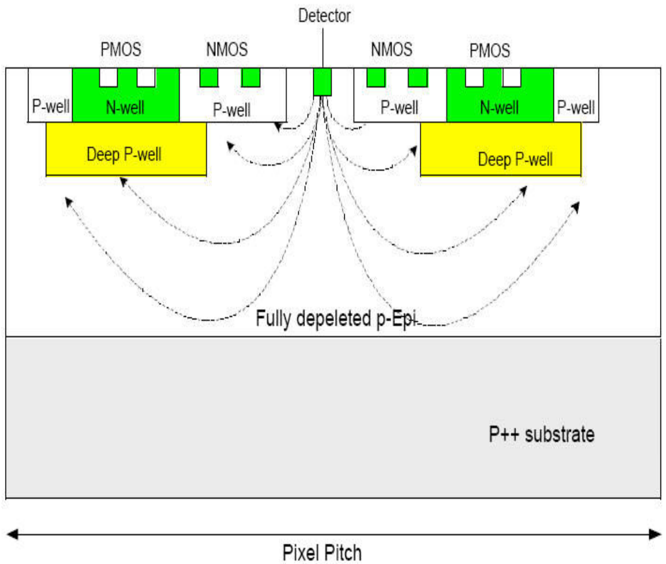


Figure 11.1 Proposed pixel architecture employing the deep p-well layer

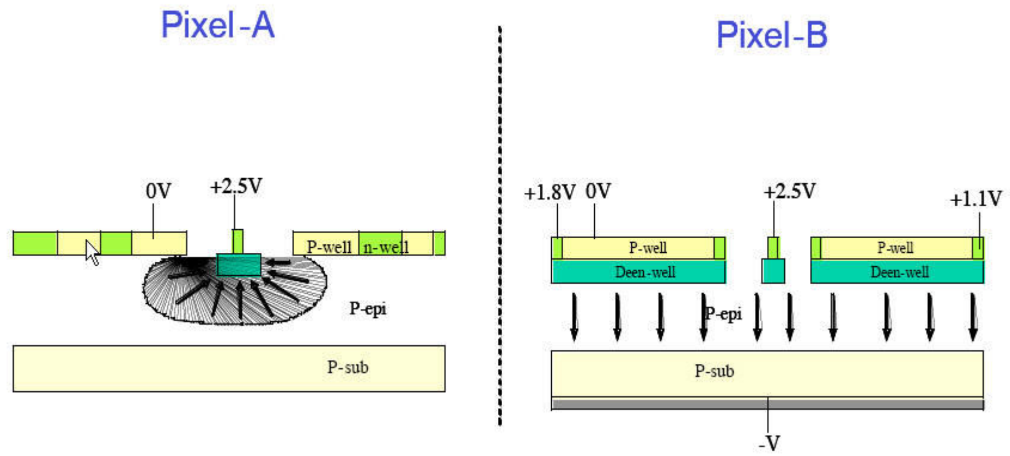


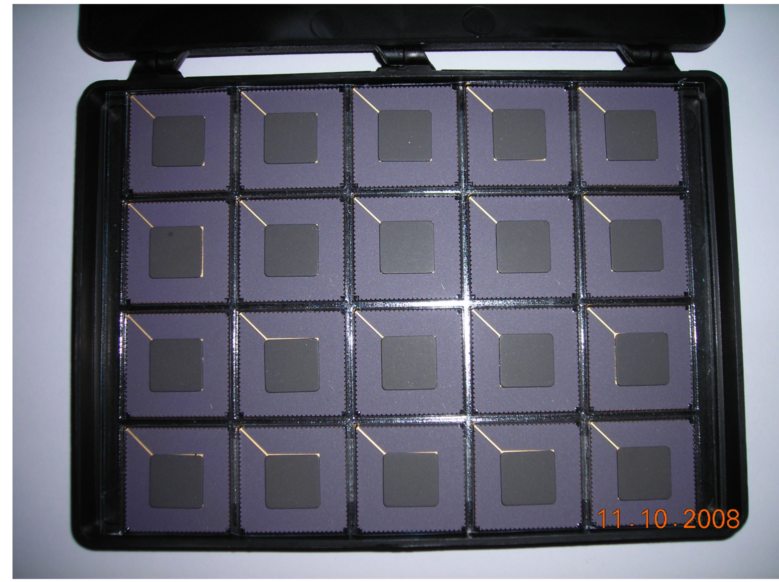
Figure 6.3 Comparison of the vertical cross section views of two pixels

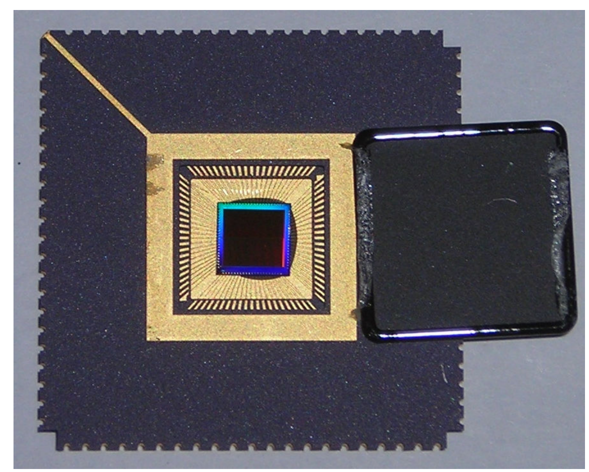
Ultimate design, as envisioned **Two sensor options in the fabricated chips**

- **TSMC process does not allow for creation of deep P-wells. Moreover, the test chronopixel devices were fabricated using low resistivity (~ 10 ohm*cm) epi layer. To be able to achieve comfortable depletion depth, Pixel-B employs deep n-well, encapsulating all p-wells in the NMOS gates. This allow application of negative (up to -10 V) bias on substrate.**

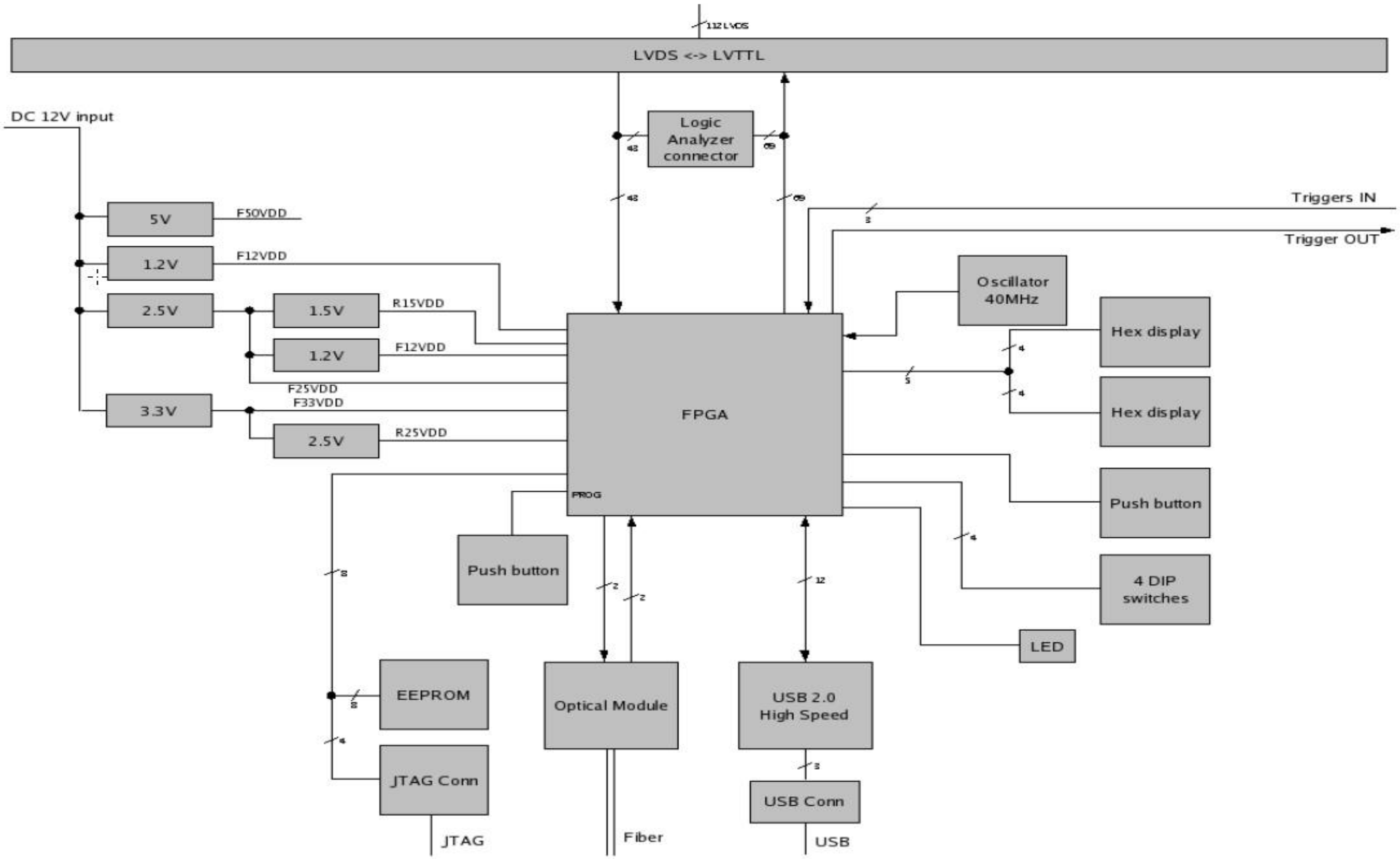
Milestones

- January, 2007
  - Completed design
    - **2 buffers, with calibration**
- May 2008
  - Fabricated 80 5x5 mm chips, containing 80x80 50 μm Chronopixels array (+ 2 single pixels) each
  - TSMC 0.18 μm ⇒ ~50 μm pixel
    - Epi-layer only 7 μm
    - Low resistivity (~10 ohm*cm) silicon
    - Talking to JAZZ (15 μm epi-layer)
- October 2008
  - Design of test boards started at SLAC
- June 2009
  - Test boards fabrication. FPGA code development started.
- August 2009
  - Debugging and calibration of test boards
- September 2009
  - Chronopixel chip tests started

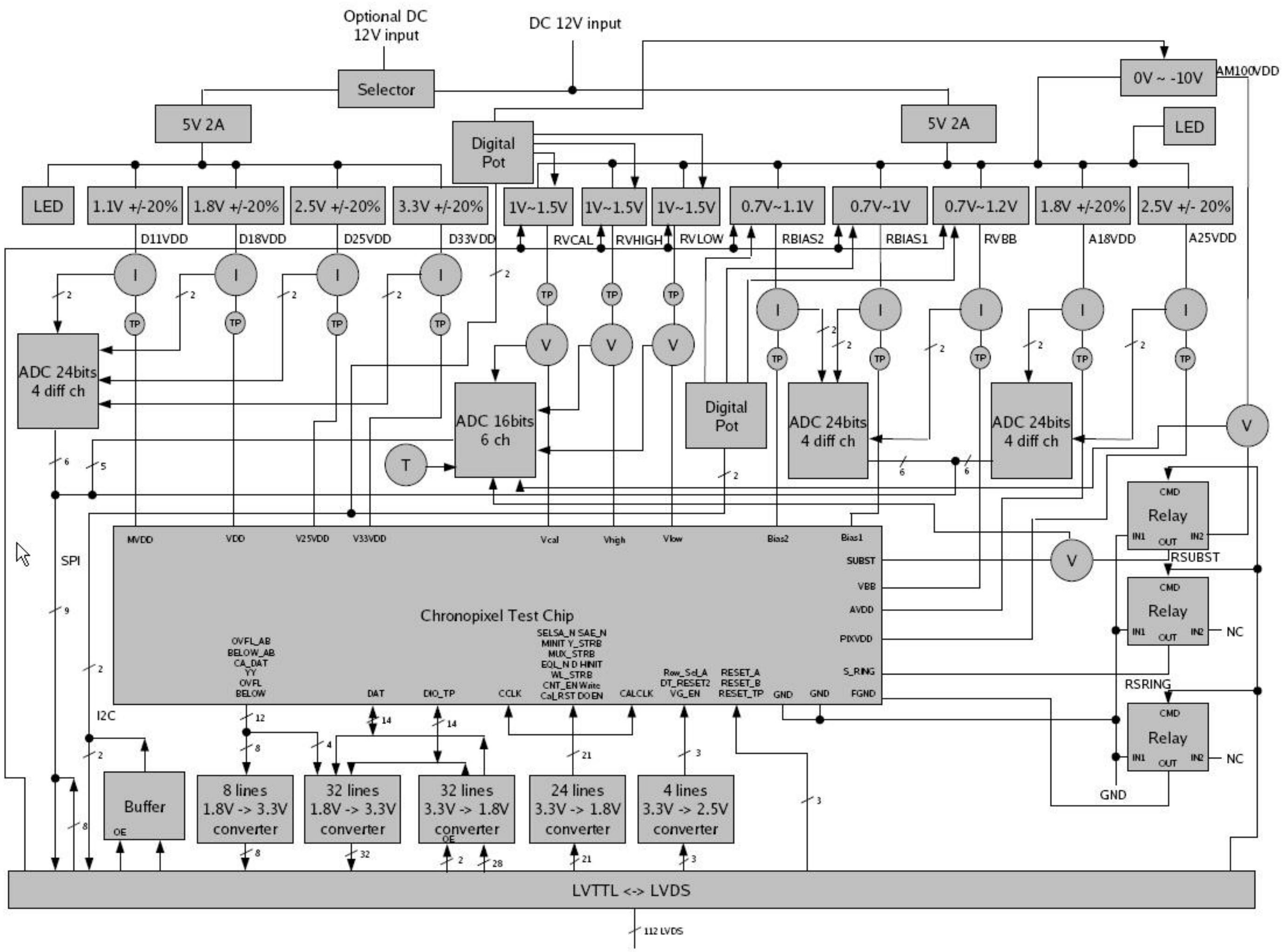




Test stand design. Block-diagram of FPGA board



Block-diagram of chronopixel test board



Test stand software

- Since May 2008 I started development of **test stand software**.
- **Graphical User Interface** was developed on the basis of **Motiff** library for Unix.
- Main **idea** of how to provide large number of **different control signal waveforms** was to use very fast **waveform memory** in the Xilinx FPGA. Memory has 32 bit wide words, and its capacity is 4096 such words. **Each bit of the memory** output register is connected to some of **chronopix control signal** wire. Memory **address** is increasing with **80 MHz rate**. So control signals have time bin width of **12.5 ns**.
- There is another, **larger memory** (24 bits wide 16 kwords) for storing **read back data** from chrono **pixels**.
- **Everything** in test stand software is **configured** by set of **text files** in configuration directory. These text files contain all **voltage settings**, all **waveforms information** and **list** of all **monitored voltages** and **currents** together with **calibration** constants.

Test stand software - continue

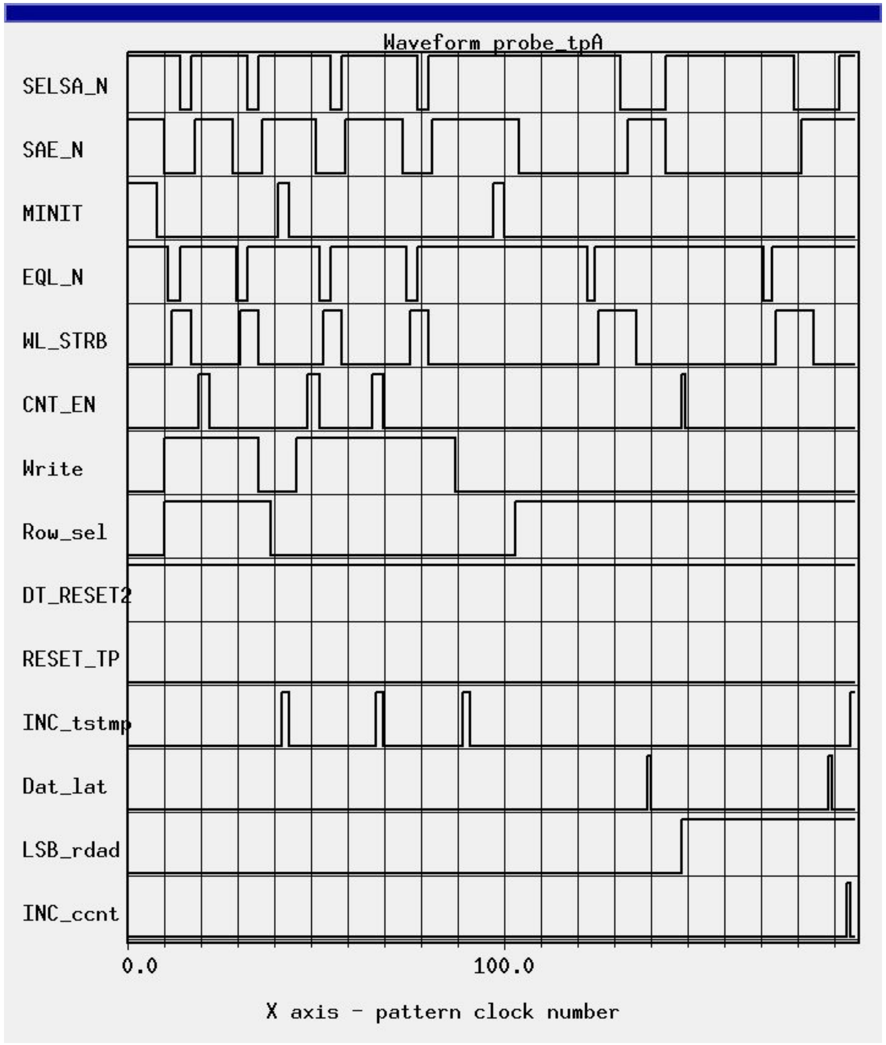
- Configuration directory also contains list of commands, which are assigned to buttons on the GUI. This list can be changed, and command buttons number and assigned commands will change without need of any code recompiling.
- All commands (performed by button clicks), are automatically saved in log files. Log files are named by current date, and each day will have only one log file, even if you restart GUI many times. Log file contains also all voltage settings.
- Another file, automatically created and filled is the monitored values records.
- Yet another files, created automatically by some of the tests – KUMAC files for use with PAW
- For every KUMAC file, record in the testcond.txt file is created, containing specific settings for the test and operator comments.

Test Stand GUI

Waveform Display Example

Test Stand GUI has a button for displaying waveforms. You can select (from drop down menus) which waveforms to display, the mode of operation to display, and clocks range – as some waveform may be too long to be shown on the display from start to the end.

Example at right – waveforms used in most noise measurements with Test Pixel A. They provide initial reset memory to 0, then manipulations to record comparator status, and then reading out results. As soon as file, describing WF is modified, new waveforms immediately can be displayed, no need to restart GUI. New waveform display always create new window, so you can easily compare different waveforms by keeping old windows open.



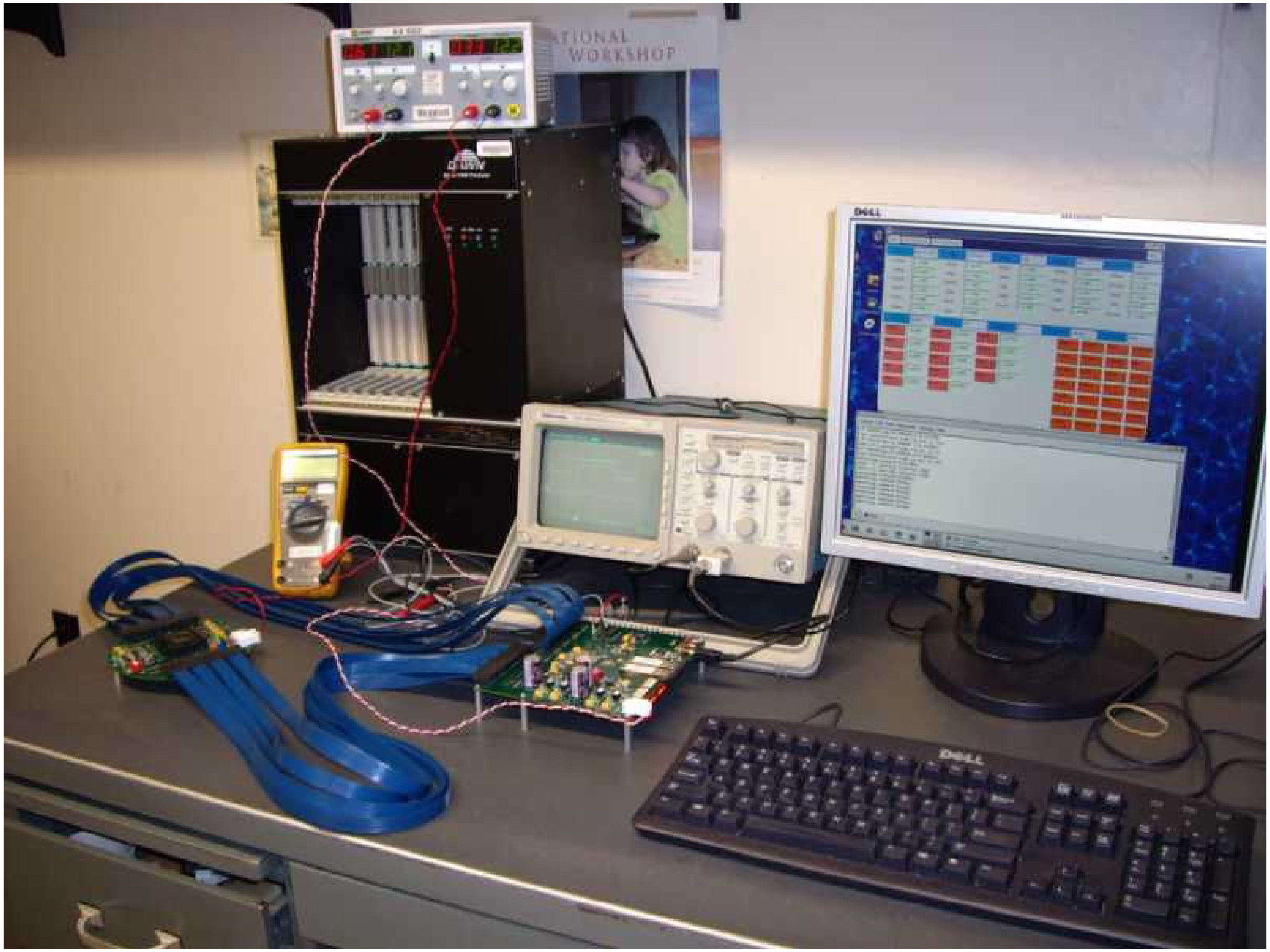
Example of waveform describing file

File Edit Search Preferences Shell Macro Windows Help

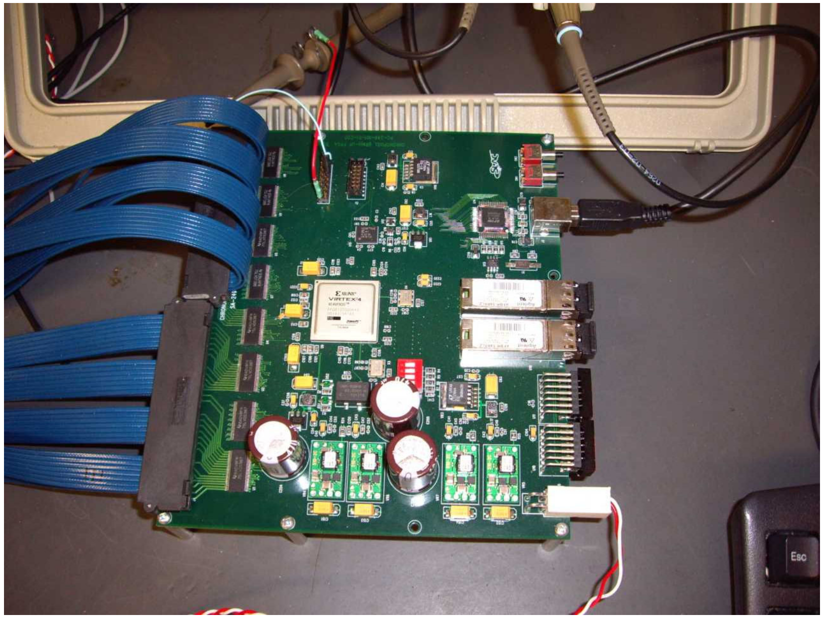
/afs/slac.stanford.edu/u/ey/sinev/CCDTest/chronotest/config/chrono_wf_memwr.dat 1041 bytes L: 1 C: 0

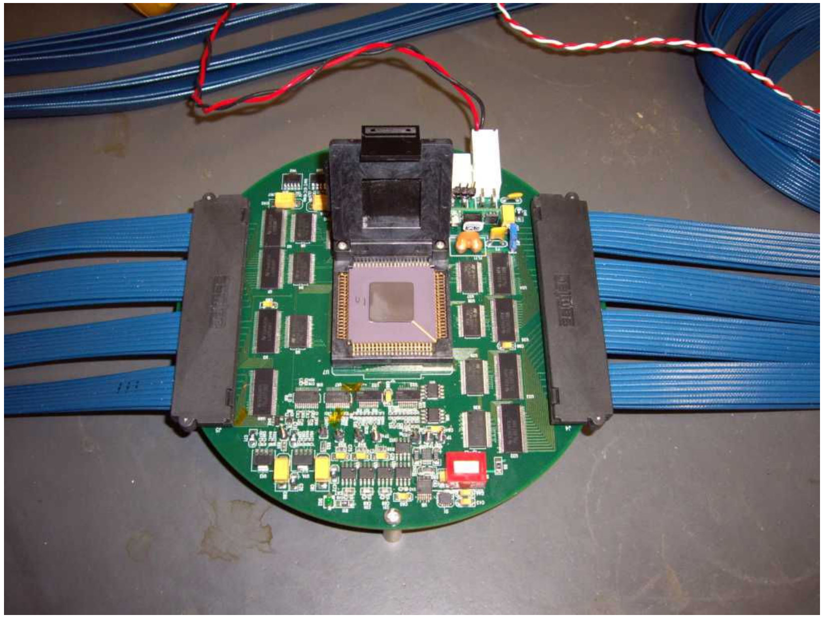
```
{80.} [55]
SELSA_N      !1 [15] (1) [35] (0) [5] (1) !1
SAE_N        !1 [15] (1) [5] (0) [5] (1) [10] (0) [5] (1) [10] (0) [5] (1) !1
MINIT        !0 [8] (1) [43] (0) 0 0 0 0 !0
Y_STRB       !1 [55] (1) !1
MUX          !1 [55] (1) !1
EQL_N        !1 [55] (1) !1
HINIT        !0 [55] (0) !0
WL_STRB      !0 [15] (0) [33] (1) [7] (0) !0
CNT_EN       !0 [30] (0) [4] (1) [18] (0) 1 1 1 1  !0
Write        !0 [15] (0) [35] (1) [5] (0) !0
Cal_RST      !0 [55] (0) !0
DOEN         !1 [8] (1) [44] (0) [3] (1) !1
CCLK         !0 [55] (0) !0
Row_sel      !0 [55] (0) !0
DT_RESET2    !1 [55] (1) !1
VG_EN        !1 [55] (1) !1
CALCL        !0 [55] (0) !0
RESET_A      !0 [55] (0) !0
RESET_B      !1 [55] (1) !1
RESET_TP     !0 [55] (0) !0
Vcal_EN      !1 [55] (0) !1
INC_tstmp    !0 [55] (0) !0
Clk_SEL      !1 [55] (1) !1
Dat_lat      !0 [55] (0) !0
LSB_rdad     !0 [34] (0) [21] (1) !1
RST_rdad     !0 [55] (0) !0
INC_rdad     !0 [55] (0) !0
INC_ccnt     !0 [51] (0) 0 1 1 0 !0
RST_tstmp    !0 [55] (0) !0
```

Teststand is working !



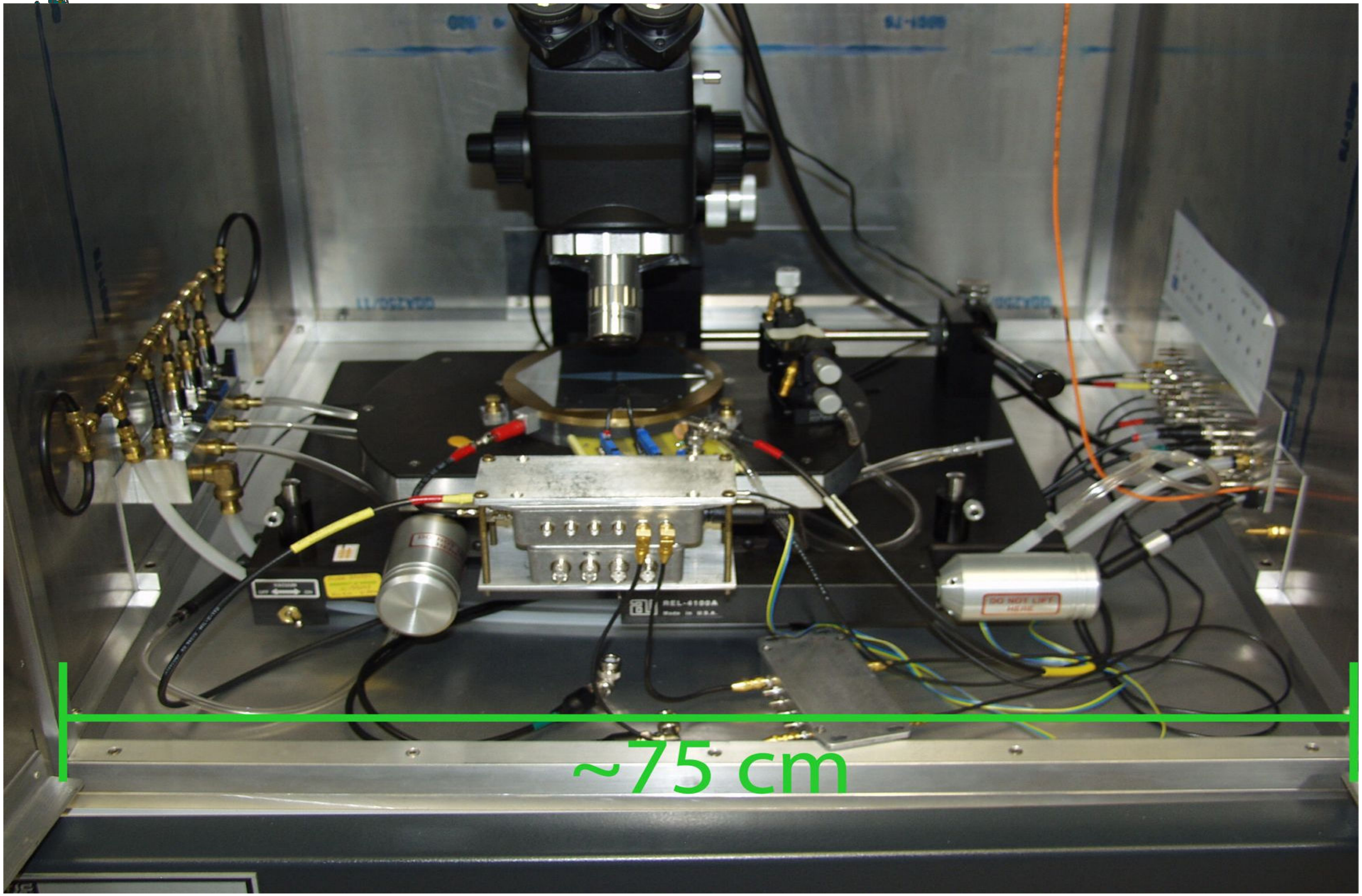
More photos





FPGA board **Chronopixel test board**

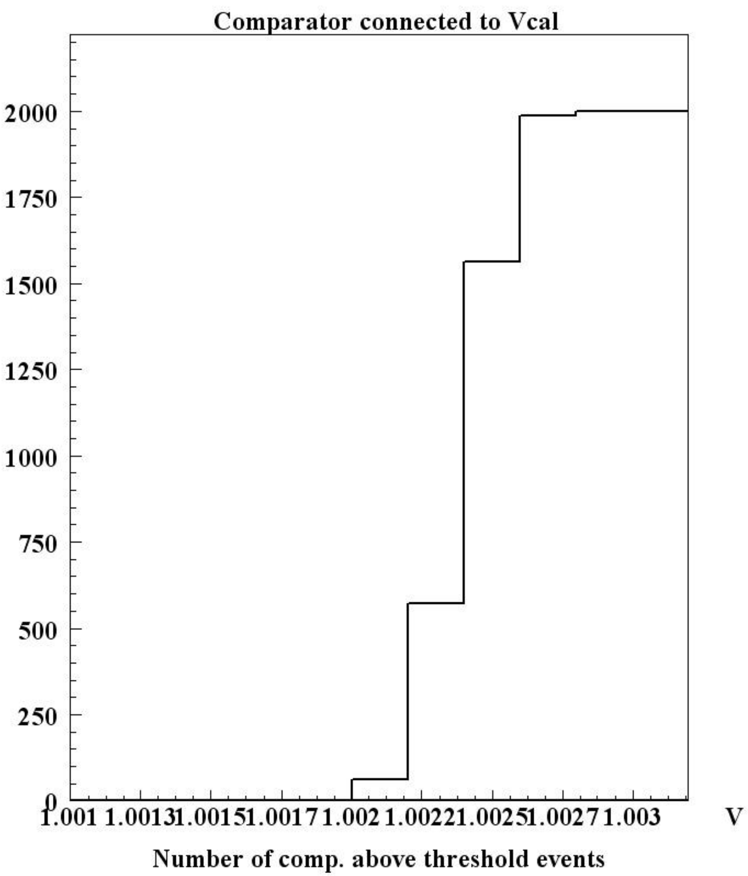
IR laser with microscope at UO



Tests plan

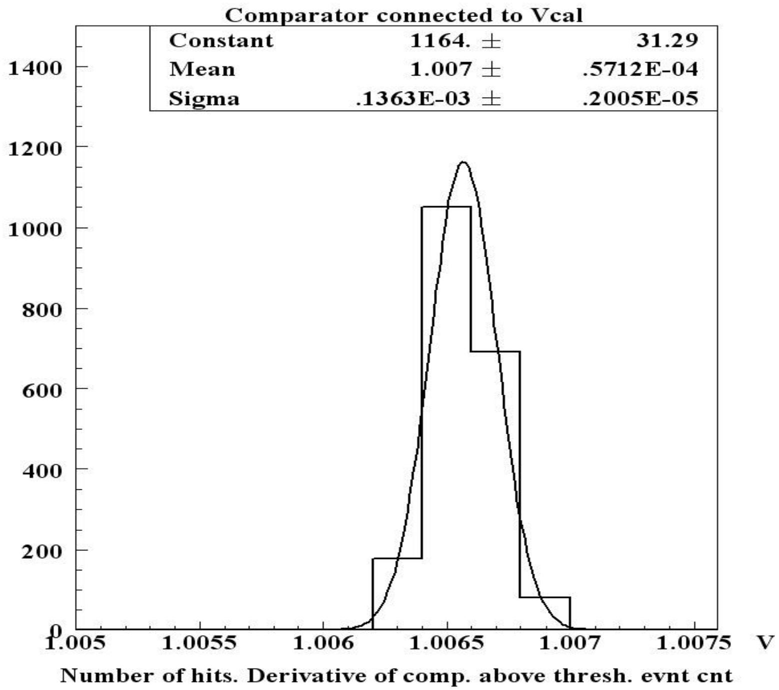
- First tests will be done with single pixels (Test pixel A and B) to learn how everything works.
- The most important part of the tests is to check, if calibration procedure works, and is 2 mV range enough to cover offsets in all pixels.
- Next test will be to check memory operations. In principle, writing into time stamps memory is only done by pixel comparator, sensing signal. But for testing of memory proper operation, external write signal can be used to record any value into all memory cells simultaneously and when read it back cell by cell.
- If everything goes smooth, even for some part of the pixels, Fe55 source can be used to determine sensitivity (expected 10 μV/e) and noise level (by the width of Fe55 peak).
- After that tests with IR laser will follow to check time stamping operations.
- Of course, power consumption, and all questions concerning 3MHz time stamp bus (crosstalk, recording errors) operation should be investigated.

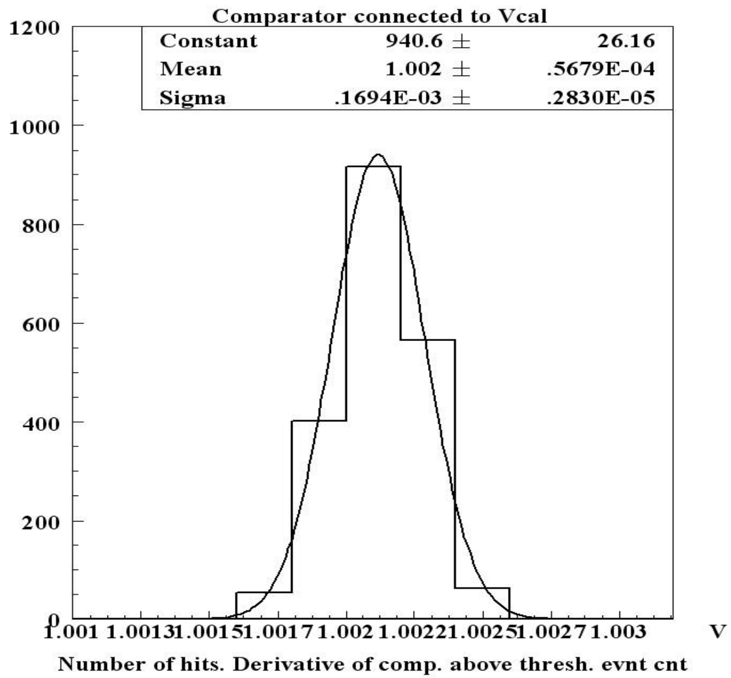
First results-noise measurements



- Horizontal axis on the plot at left shows comparator threshold (set by connecting reference input of comparator to calibration selected tap of resistive ladder and setting voltages Vlow and Vhigh, on the ends of the ladder. (These voltages differ only by 2 mV !). Values shown on x axis are Vhigh. Vlow is always by 2 mV lower.
- Vertical axis shows number of cases than comparator at the sampling moment appeared fired (which means it sensed input voltage as lower than reference – remember we expect negative signals). Notice, entire range of x axis values is only 2 mV on this plot!

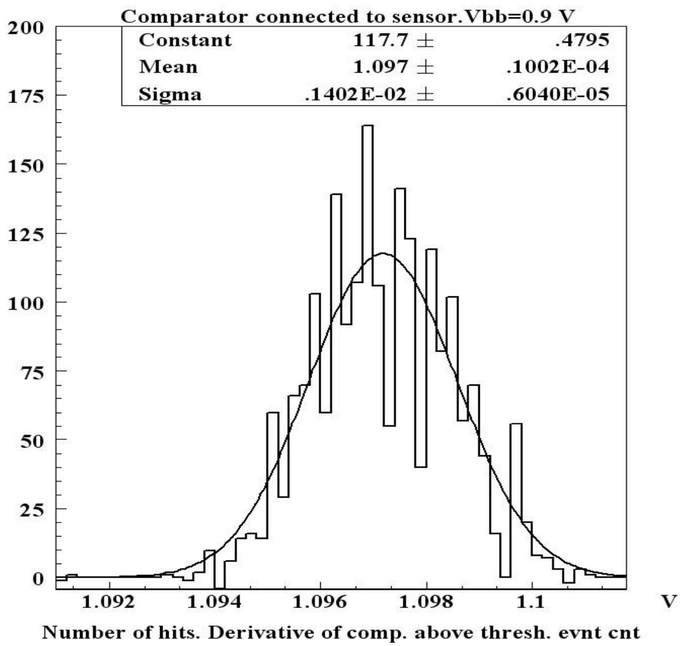
Test results - continue

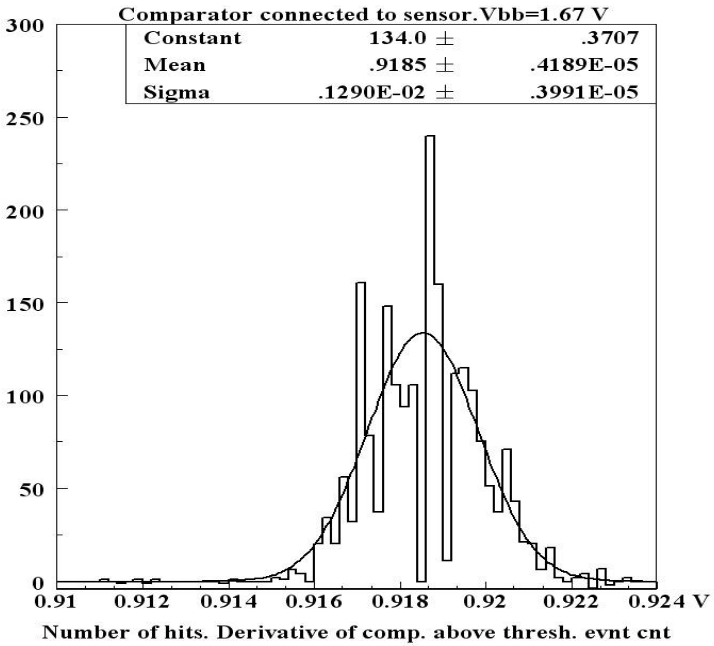




- **Derivative of curve shown on previous page gives us comparator noise distribution. Such curves are shown here. Keep in mind, that subtracting two random values gives large fluctuations, so errors in fitted parameters are, probably, wrong – fitter underestimate statistical errors in bin contens.**
- **This results looks good (noise level of order 150 µV, and if responsivity, as expected, is really 10 µV/e, that corresponds to 15 e). But this is comparator alone, no sensor connected.**

Noise with sensor connected





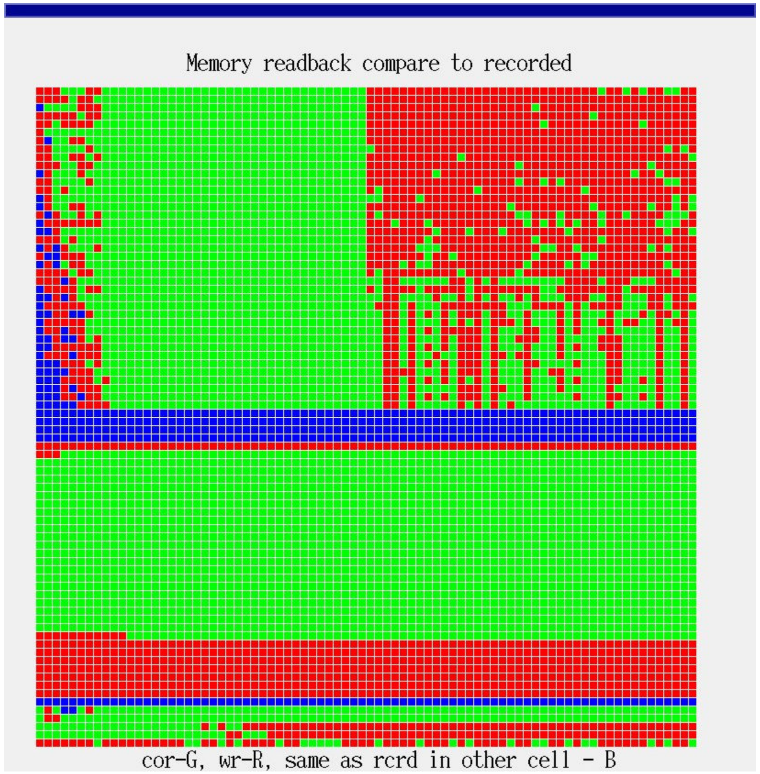
- **When sensor is connected, noise is much worse! It was expected, though not so much. Test pixels do not have sophisticated "soft reset" circuit, which, as Sarnoff engineers claim, can reduce reset noise by factor of 2. So, more important will be results with pixels in arrays, where such reset is implemented. Results on the pictures above are obtained with 2 different values of source follower current - ~0.7µA at left and ~1.5µA at right. Difference in noise levels (1.4 mV and 1.3 mV) has expected sign, but may be just accidental.**

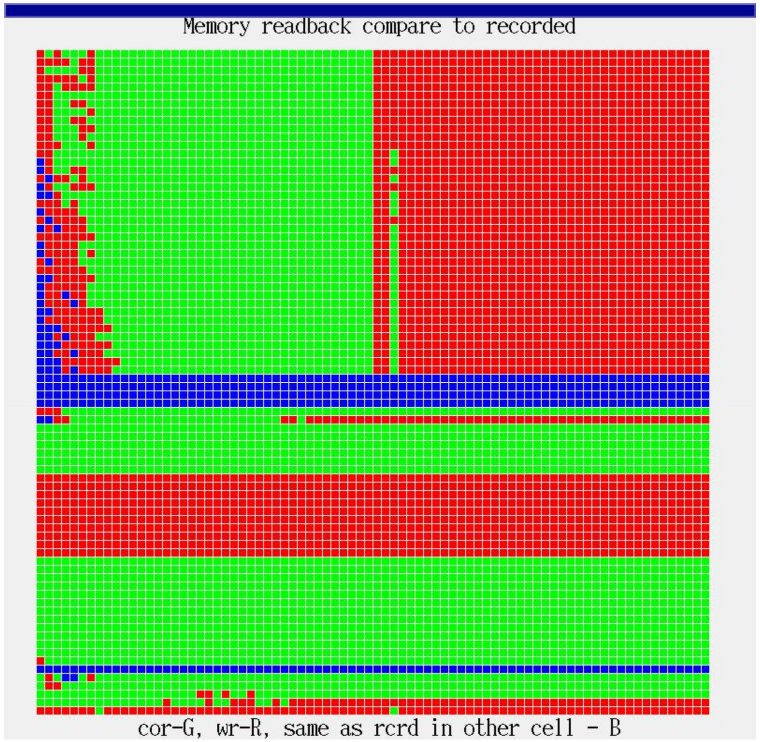
Table of noise measurements

Date	Chip #	Comp in	Temp	Vbb	Peak pos	Sigma
Sept 23	4	Vc=1.000 V	~35 C	1.25 V	1.0066 V	136 µV
Sept 23	4	Sensor	~35 C	1.25 V	1.0198 V	1.44 mV
Sept 24	5	Vc=1.000 V	~35 C	1.25 V	1.0032 V	266 µV
Sept 24	5	Vc=1.000 V	~35 C	1.25 V	1.0032 V	271 µV
Sept 24	5	Vc=1.000 V	~28 C	1.25 V	1.0021 V	170 µV
Sept 25	5	Vc=1.000 V	~35 C	1.25 V	1.0032 V	229 µV
Sept 25	5	Vc=0.999 V	~35 C	1.25 V	1.0022 V	224 µV
Sept 25	5	Vc=1.000 V	~28 C	1.25 V	1.0016 V	170 µV
Sept 26	5	Vc=1.000 V	~35 C	1.25 V	1.0017 V	183 µV
Sept 26	5	Sensor	~35 C	1.67 V	0.9185 V	1.29 mV
Sept 26	5	Sensor	~35 C	0.9 V	1.0970 V	1.40 mV

- You can see for yourself how thing are stable, if there temperature dependence and how worse noise with sensor connected is (in red).

First glance at pixel arrays

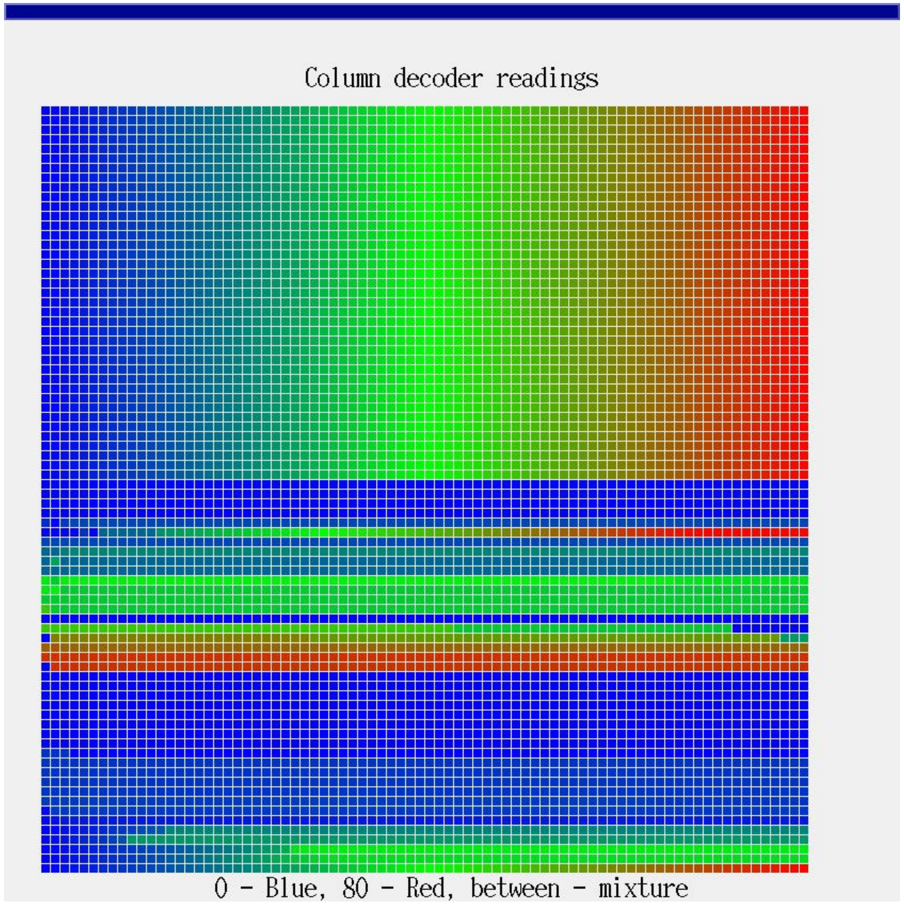




- **Here you can see comparison of code written in individual pixels memory with read back values. Picture at left is for recorded code = 0x3fff, at right for code = 0. Green read back is the same as written, red – different, blue – corresponds to code written into another cell (recall each pixel has 2 memory cells). Pixel array A is in left 40 columns, B – right columns.**

First glance at arrays - continue

- Pictures on **previous slide** do **not** tell **all story**. Here you can see **column decoder** readings. Upper 40 rows behave exactly **as expected** – every next pixel has column number increased by 1, but in **lower 40 rows** readings are **chaotic**. I think something is wrong with **row number decoding** – either in the chip, or in our test board connection to correct pins encoding row number.
- Remember, it is **first try** – I did not have time to debug things. I am showing it, to demonstrate, that **in general pixel array logic is working**. And to brag about my pixels display.



Next Steps

- September – December 2009
  - Test and characterize prototype 1
  - Design prototype 2 (with Sarnoff) and start fabrication
- January 2010 – September 2010
  - Finish fabrication of prototype 2
  - Design and implement modification of test boards
  - Test prototype 2
  - Design prototype 3 with Sarnoff and start fabrication
- October 2010 – September 2011
  - Test prototype 3 – close to real detector for ILC

Conclusions

- First chronopixel prototypes have been **fabricated, packaged delivered** to SLAC and are been tested.
- Tests show that general **concept is working**, but we need to do much more measurements before we can characterize its performance.
- It is obvious now, that at least **some corrections** to design will be **needed**.
- We are looking for the manufacturer of the next prototype implementing **deep P-well**. Depending on how much correction to the design will be needed, **next prototype** may be ready for submission at the **end 2009** – beginning 2010. It still will be 50x50 μm pixels, but completely operational, **100% efficient** device.
- After that accomplished, **scaling to 45 nm** technology may be thought. So, funding depending, we can be ready to start design of final vertex detector sensors in **2010-2011**.