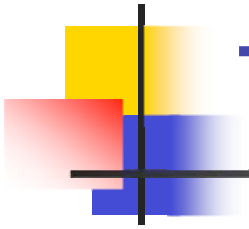
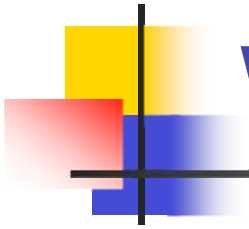
Htwish in Preproduction

John Freeman, 7/28/03



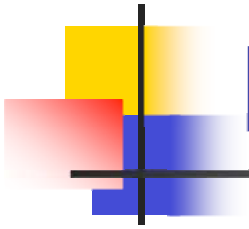
Talk Outline

- A brief description of Htwish
- How we use Htwish
- Recent improvements to the program
- Ongoing and future improvements: discussion encouraged!



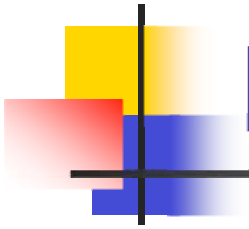
What is Htwish?

- Originally written by Igor Volobouev for testing Svx3 chips/hybrids
- Runs tests on chips/hybrids, storing their output in database files – “Htest” mode
- Analyzes the data in these files, both producing histograms and pass / failing the hybrid by cutting on the data



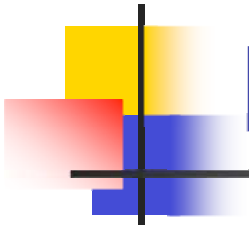
Htest Mode

- Htwish, as of now, can run 16 tests. Some are used only for the histograms they produce, others are involved in official hybrid evaluation
- Running the full set of tests takes ~ 11.5 minutes – this will need to be reduced!



Hybrid evaluation

- Pass/Fail mechanism works in a hierarchical manner: hybrids evaluated in terms of chips, chips from cells and channels, etc.
- Each component is evaluated in terms of (A) its subcomponents, and (B) its own relevant set of tests

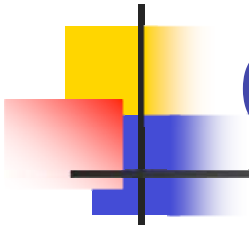


Hybrid evaluation (cont'd)

- Some tests are themselves Pass/Fail (the pipeline cell ID check, e.g.) . Others involve cutting on the measured data: the gain in the chip's pipeline capacitors, for example

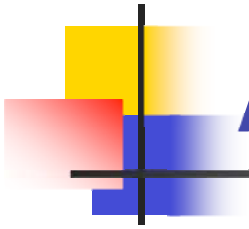
Htwish You Were Here

- In May, William Wester at FNAL used Htwish to produce data for 2 wafers of new Svx4's
- Here at LBL, we've used Htwish to analyze this data, along with our own chips
- In August and September, Htwish will be used during hybrid irradiation



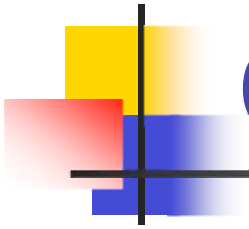
Coming up…

- Htwish will be used at all stages of Svx4 testing: wafers at FNAL, hybrids at LBL, burn-in at Davis, and final approval at FNAL
- Htwish must accommodate all labs: must be transparent to the DAQ hardware, be able to produce flat-ascii format data, etc.



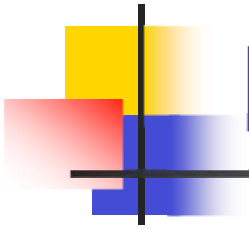
A kinder, gentler Htwish

- Htwish is being improved:
  - Easier to install (less path references, better Makefile, etc.)
  - Easier to run (convenient script passes parameters to Htwish)
  - Easier to change tests used during evaluation
  - Initialization stream data to be recorded



On a helpful Note…

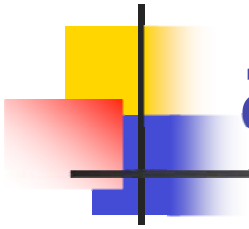
- Htwish manual should ease the learning curve
- Covers many subjects: installation, addition / removal of tests (both in Htest and evaluation mode), code overview, troubleshooting, etc.
- About 8000 words / 20 pages, it will soon be published as a CDF note



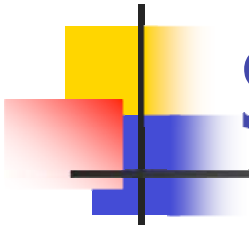
Htwish at FNAL

- Htwish will be used extensively on the PTA/PMC system
- Tom Junk had to recode his Htwish due to the system's different readback scheme
- We've agreed Htwish should have a standard code => recompilation should be all that's necessary

Svx4Gui: Satyajit's ROOT analysis tool

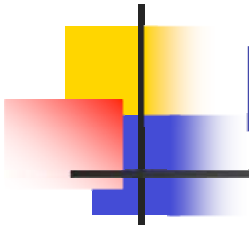


- Satyajit Behari has written a set of ROOT CINT-usable classes designed to histogram hybrid data; they are also in his standalone Svx4Gui program
- These classes handle data in flat ascii format
- Consequently, Htwish can now output data in this format



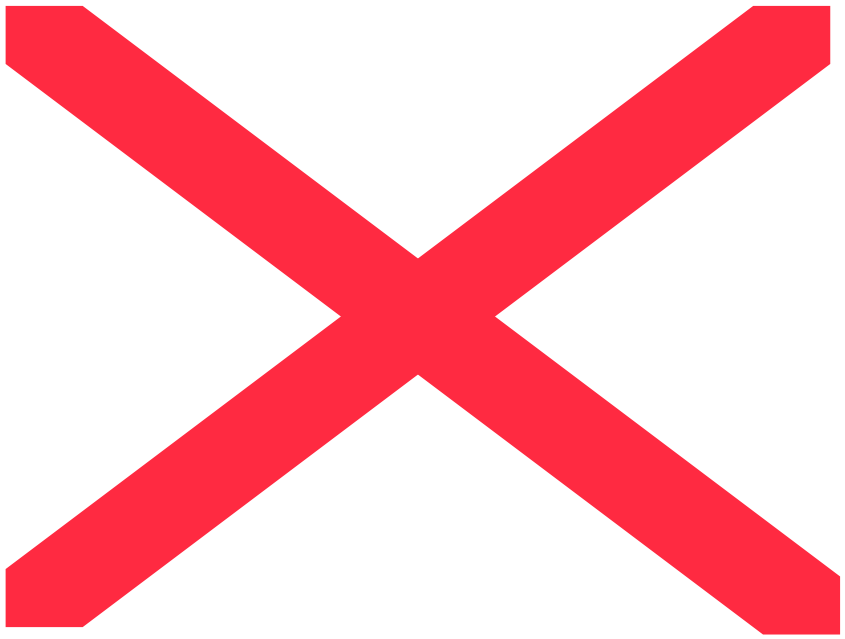
Svx4Gui cont'd

- Open question: how much Htwish data should be written to ascii (space is an issue!) ?
- Less space-consuming, but less-powerful: have Htwish output summary files for LBL-coded ROOT macros
- We wish to plot pipeline pedestal noise and gain, as well as preamp risetimes

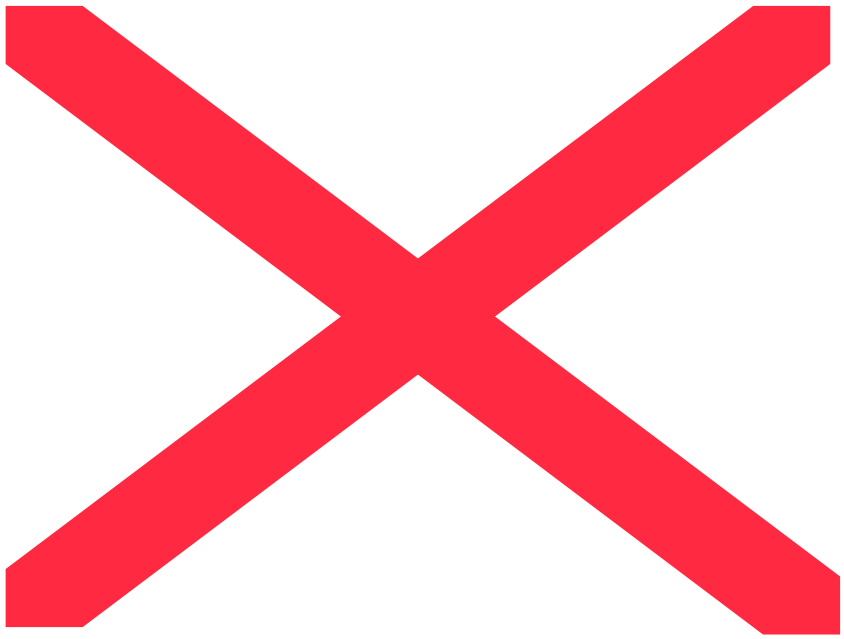


Benchmarking issues

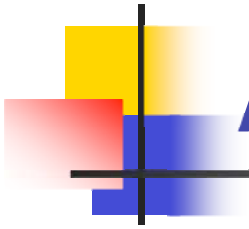
- Longer term issues remain: e.g., how long should it have to take to test a chip or hybrid?
- Currently, the breakdown of test times on our systems is as follows…



Current test-time breakdown: total time = 656 s



Potential test-time breakdown: total time = 286 s



Analysis issues

- Also an open question: what should be our component / data cuts during analysis?
- Let's examine different sets of cuts on a sample of 100 chips from William's data…

The Component Cuts

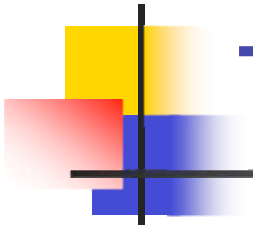
of bad components permitted

CUT SEVERITY	Channels in chip	Cells in chip	Caps in chip	Caps in channel	Caps in cell
Not Strict	2	1	20	2	4
Strict	1	0	10	1	2

Note: 46 caps/channel, 128 caps/cell, 46 x 128 = 5888 caps/chip

The Data Cuts

CUT SEVERITY	Pedestal Median, Residual, Max Noise	Gain Median, Residual, Max Noise	Fast preamp, risetime	Medium preamp, risetime	Slow preamp, risetime
Not Strict	(50,150), (-5,5), 8	(80,130), (-20,20), 8	(0,1.7)	(0.2, 2)	(0.2,3)
Strict	(70,100), (-4,4), 8	(-85,120), (-18,18), 8	(0.1,1.2)	(0.2, 1.5)	(0.2 ,2.2)



The Results

COMPONENT CUT SEVERITY	DATA CUT SEVERITY	PASS # / FAIL #
Not Strict	Not Strict	61 / 39
Not Strict	Strict	55 / 45
Strict	Not Strict	51 / 49
Strict	Strict	40 / 60

The Results, cont'd

Q: How many of the chips had all 5888 pipeline capacitors pass the gain and pedestal noise and residual tests?

A: With the "not strict" data cuts, 43 out of 100.
With the "strict" data cuts, 35 out of 100.