

Burn-In of Hybrids for the CDF RunIIb Silicon Detector

Maxwell Chertok, Cynthia Frei, Britt Holbrook, Richard Lander, Tiffany Landry,
David Pellett, Aron Soha, Guangmao Xing, Wajohn Yao

University of California, Davis, CA 95616 USA

Abstract

This note describes the runIIb silicon vertex detector hybrid burn-in project carried out at UC Davis, primarily between 2002 and the beginning of 2004. These hybrids contain the SVX4 readout chips that were intended for use in the next generation vertex detector at CDF. The project has included recommissioning hardware that had been used to burn-in earlier versions of the readout chip, development of new hardware to expand the capability and capacity of the burn-in operation, extension of control and analysis software, development and use of quality assurance and quality control procedures, and the implementation of a comprehensive web-based hybrid tracking database. These aspects of the project are discussed in what follows.

Contents

1	Introduction	4
1.1	Context	4
1.2	Outline of the Burn-In Process	5
1.3	Schedule	5
2	Hardware for Burn-In	6
2.1	Overview of Hardware for Burn-In	6
2.2	Linux PC with PCI Cards	6
2.3	Buffer Card	9
2.4	Switch Boards	9
2.5	Pattern Board	10
2.6	Multiplexed Scrambler Regulator Board	12
2.7	Hybrid	13
2.8	SVX4 Chip	14
3	Software for Burn-In	16
3.1	Introduction to Hybrid Monitor	16
3.2	Hybrid Monitor GUI	16
3.2.1	Configuration	17
3.2.2	Voltages and Currents	18
3.2.3	Pedestals and Gains	19
3.2.4	Noise and Dnoise	19
3.2.5	Temperature	19
3.2.6	Messages	19
3.3	Svxscope & Htest	19
3.3.1	Svxscope	20
3.3.2	Htest (also known as Htwish)	20
3.4	Burn-in Scheduler	22
3.5	Data Acquisition	22
3.6	Important Files Generated During Burn-in	22
3.7	Testdisplay	23
4	Quality Assurance and Quality Control	25
4.1	Motivation	25
4.2	Logistics	25
4.3	Tools for Quality Assurance and Quality Control	26
4.3.1	Quality Tree	26
4.3.2	Traveler Plots	27
4.3.3	Before-and-After Plots	27
4.3.4	Htest Log File	38
4.3.5	Raw Htest Data	39
4.3.6	Current and Voltage Plots	39

5 Hybrid Über Tracker Database	39
5.1 Environment	41
5.2 Users	42
5.3 Data	42
5.4 Usage	42
5.4.1 Production Statistics	43
5.4.2 Hybrid Information	43
5.4.3 Hybrid Repairs	43
5.4.4 Hybrid Summary	44
5.4.5 Burn Summary	44
5.4.6 Cables	44
5.4.7 Wafer Probing	44
5.4.8 Substrate & Assembly	45
5.4.9 Chip Quality	45
5.4.10 Hybrid Stuffing	45
5.4.11 Hybrid Testing	45
5.4.12 Hybrid Burn In	47
5.4.13 Hybrid Repair Options	47
5.4.14 FNAL check off/Modules	48
5.4.15 Change Your Password	48
5.4.16 LogOut	48
5.5 How to Add a New User	48
6 Results and Conclusions	48
6.1 Results	48
6.2 Concluding Remarks	50
A Data From Svxscope	51

1 Introduction

1.1 Context

The current silicon vertex detector (SVXII) was not originally intended to withstand the high radiation dose accompanying the second stage of the Tevatron runII program and was slated for replacement by an upgraded silicon detector (SVXIIb) [1] as part of a broader upgrade plan [2]. The SVXIIb detector features a highly modular design where the sensors, voltage lines, control lines, data lines, readout chips, and support structures are all grouped into what are called *staves*, and staves are arranged in six staggered layers to provide complete coverage in ϕ . Each stave contains six *modules*, three on one side and three on the other, where each module consists of two connected sensors (sensors are approximately 96 mm x 41 mm) read out by an associated *hybrid* mounted near the end of each sensor pair. For layers 2-4, one side of each stave has axial strips with a 75 μm pitch while the other side has small angle stereo strips at 1.2 degrees and 80 μm pitch. Layers 1 and 5 are made up entirely of axial strips. An additional inner-most layer, layer 0, would replace the current L00. The hybrids consist of a beryllia substrate, the radiation-hard SVX4 readout chips [3], additional surface mounted components, and conductive traces. Some features of the hybrids, and the associated SVX4 chips, are detailed in section 2.

An important phase in the preparation of a detector such as the SVXIIb is the exercising and testing of the hybrids – a procedure commonly referred to as burn-in. The primary purpose of this procedure is to prevent the inclusion of faulty hardware in the final detector. It can also aid in optimizing parameters for operation and characterizing performance. To understand where the burn-in fits into the detector production, the following provides an outline of the work flow.

The wafers of SVX4 chips are produced using a radiation hard commercial 0.25 μm process. These wafers are shipped to Fermilab for testing, and then sent to a company for dicing. Meanwhile beryllia substrates are received at LBNL from a company called CPT and are inspected and tested. Once the SVX4 chips arrive at LBNL from dicing, they are tested and grouped before being sent to a company called Advanced Assembly for mounting on the beryllia substrate, along with the other surface mounted components. The hybrids are then tested again at LBNL before being sent to one of two companies, Amtech or Promex, for wire bond connections between the SVX4 chips and the substrate (the wire bonding for layer 0 hybrids is done at LBNL). Finally, the hybrids are tested once more at LBNL before being sent to UC Davis for burn-in. After the burn-in is complete, the hybrids are typically sent to Fermilab where each one is connected to a pair of silicon sensors for the formation of a module. These modules are then tested and included on a stave assembly. In turn, the staves are tested and mounted onto a barrel fixture.

The following (section 1.2) introduces the basic steps carried out during the burn-in process, the details of which are provided in the rest of this document. That is followed (section 1.3) by a discussion of the schedule of the project. The results and conclusions, including the production statistics, for the burn-in project are in the final section of this document (section 6).

1.2 Outline of the Burn-In Process

Upon being received at UC Davis from LBNL, each hybrid is placed in a stand-by area until space is available on the burn-in stand. During different phases of the project, the burn-in stand has had different capacities, and this is summarized in section 1.3 below, but typically a hybrid would sit around for no more than a few days. The burn-in proceeds for at least 72 hours, the majority of which consists of controlling each hybrid so that it goes through a repetitive cycle of acquire, digitize, and readout. The burn-in also includes extensive testing of the SVX4 chip features at the beginning and end of the burn-in period. There are also periodic data integrity checks, as well as current and voltage measurements. During the commissioning of the burn-in stand, hardware or software problems occasionally surfaced, in which case, after the problem was identified and solved, the relevant hybrids participated in an additional burn-in. After the burn-in is complete, the hybrid is kept at UC Davis long enough to process the data and complete a performance evaluation (see section 4). Hybrids that have passed the burn-in stage are then sent to Fermilab. Some hybrids that arrived for burn-in, particularly early in the project, were known to have failed portions of initial testing at LBNL and were sent to burn-in only for study. Hybrids in this category have been returned to LBNL for further testing or repair, and are not failures due to burn-in. Finally, a handful of hybrids have been sent to other collaborations for consideration for use in their detectors.

1.3 Schedule

The schedule for the burn-in project has evolved over time, with the largest change being due to the fact that the silicon detector upgrade was canceled as of September 3, 2003. Prior to that decision, the schedule as it stood in the Summer of 2003 included a preproduction phase to be completed in the Fall of 2003 and production occurring during 2004. The full production was to include 1080 4-chip hybrids for layers 1–5 of the SVXIIb and 72 2-chip hybrids for layer 0. The expected average load for burn-in was to be about 40 hybrids per week. For full production, the burn-in stand was to have ports for 64 hybrids, with the capability of doing 2 batches per week if needed, which was well above the expected requirements. After the cancellation of the SVXIIb component of the runIIb upgrade, the silicon detector project has entered a phase called close-out, which includes the construction of up to 15 staves, with a chance to demonstrate detector performance, as well as production techniques and throughput capabilities.

The schedule for the Fall of 2003 has thus changed from a pre-production phase to the close-out phase, with the processing of 90 4-chip hybrids plus at least 10–20 spares and approximately 15 2-chip layer 0 hybrids. For burn-in, this means that the requirement for the number of hybrids to burn-in at one time has been lowered, but it remains important to demonstrate the full production throughput. On October 3, 2003, the capacity of the burn-in stand reached 16 ports using prototype hardware, and on December 18, 2004 it reached 32 ports using a mixture of modified prototypes and components with the final design changes incorporated. With each burn-in

lasting 72 hours, two could be accomplished each week, for a total final capacity of 64 hybrids per week, again above the anticipated requirement of 40 per week for the full production.

A complete summary of the number of hybrids passing through the burn-in stage, and their categorization, is available in the concluding section (section 6) of this document.

2 Hardware for Burn-In

2.1 Overview of Hardware for Burn-In

To exercise and test the hybrids as outlined in section 1.2, and to accommodate the originally intended throughput, the burn-in stand is a complicated ensemble of electronic components, some inherited from previous burn-in projects, and some developed for this specific purpose. Initialization and control signals, as well as low voltages, must be directed to the appropriate hybrids through central switching systems, and data and status readings must be multiplexed back to a computer.

This section provides information about each component and how it interacts with the rest of the burn-in stand. Figure 1 shows a sketch of the complete burn-in stand, at maximum design capacity (64 hybrids), and serves as a reference throughout this section. Figure 2 shows the burn-in stand, as utilized for the close-out of the runIIb silicon project, with a capacity of 32 hybrids.

2.2 Linux PC with PCI Cards

The personal computer (PC) that controls and reads out the burn-in stand uses the Linux operating system and runs a program called Hybrid Monitor, which is thoroughly described in section 3. Figure 3 shows the PC, with the Hybrid Monitor software running. The PC is equipped with three PCI (Peripheral Component Interconnect) cards to provide input and output (I/O) signals, calibration voltage, and current and voltage monitoring. The following describes each of these PCI cards:

Input/Output and Strobe The card used to generate all of the control and pattern signals for the burn-in stand is the Keithley KPCI-PIO96 Parallel Digital I/O board. It also reads out signals that originate from the hybrids and propagate back to the PC. This board has 96 I/O lines which are TTL compatible. 41 of the 96 available lines are used. The motivating factor for using this board was that it was already in use by the LBNL PATT Chip Test Setup system for operating SVX4 chips. We use a device driver written specifically for the PATT Chip Test Setup [4].

Calibration Voltage (VCAL) Measurement Computing's CIO-DAC02/16 dual-channel 16-bit analog output board is used to supply the analog VCAL voltage for injecting charge into the SVX4 preamplifiers. This board has two analog outputs with 16-bit resolution and several output ranges. The burn-in stand uses one of the

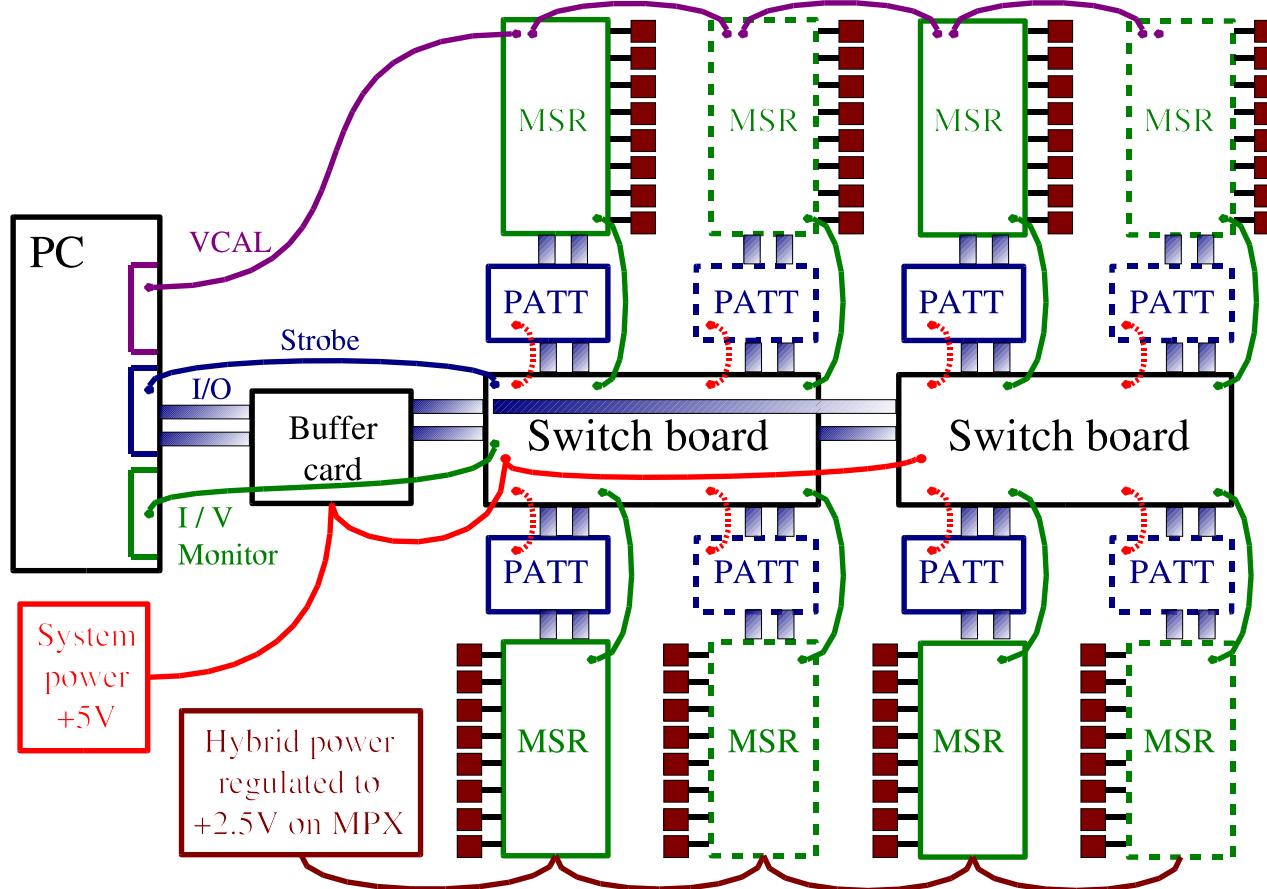


Figure 1: This sketch shows the hardware and connections for the full capacity burn-in stand. The components are described in the text of this section. The pattern (PATT) boards (blue) and multiplexed scrambler regulator (MSR) boards (green) that are outlined by solid lines represent those used in the close-out burn-in project, while those outlined by dashed lines represent additional boards that were not constructed or installed. For simplicity, the calibration voltage (VCAL) distribution is only shown for the top half and the hybrid power distribution is only shown for the bottom half.

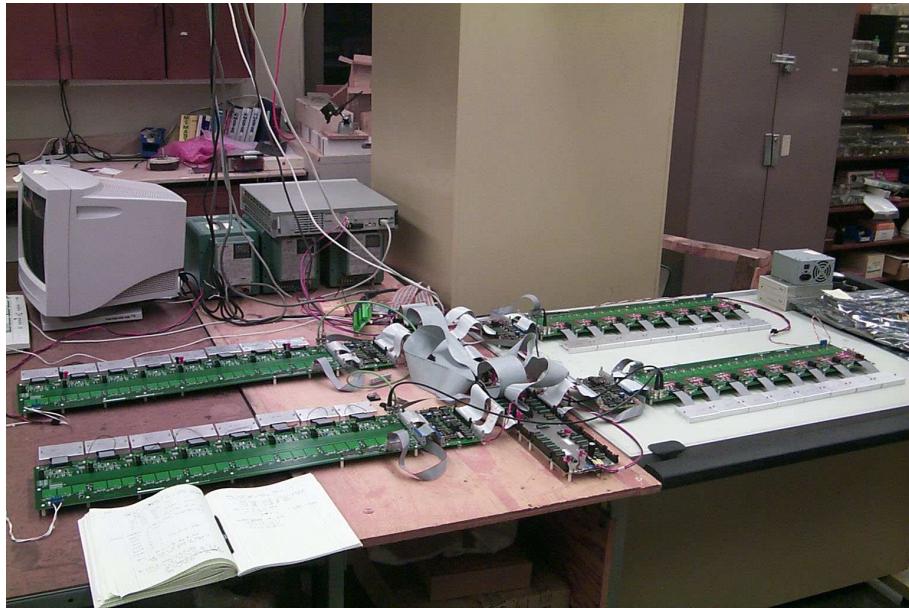


Figure 2: This photograph shows the burn-in stand as used for the close-out portion of the hybrid burn-in project. Here the stand is shown with the maximum capacity of 32 hybrids (the 32 silver rectangles that are lined up in 4 rows).



Figure 3: This Linux PC, which is running the Hybrid Monitor burn-in control program (see section 3), contains the three PCI cards discussed in section 2.2.



Figure 4: The buffer card, pictured here and described in section 2.3, passes signals between the I/O PCI card and the rest of the burn-in stand.

available output signals with an output range of ± 5 V. The motivating factor for using this board was that it was already in use by the LBNL PATT Chip Test Setup system for operating SVX4 chips, and we use a device driver written specifically for the PATT Chip Test Setup [4].

Current and Voltage Monitoring The National Instruments 6036E multifunction I/O board was chosen as the device to read currents drawn by the hybrids on the burn-in stand. It has 16 channels of single-ended 16-bit analog input (8 channels differential) and eight lines of digital I/O. Three of the 16 analog inputs are used to read two signal voltages and one reference voltage. National Instruments does not offer a Linux driver for this board. We chose to use the open source Comedi (Linux control and measurement device interface) driver collection and the Comedilib device interface library [5].

2.3 Buffer Card

The buffer card, pictured in Figure 4, is a wire-wrapped board that was designed and built at UC Davis. As shown in Figure 1, it is connected to the I/O PCI card of the PC by a pair of ribbon cables. It provides pass-through to the rest of the burn-in stand for input pattern and control lines. It also converts returning data from differential to single-ended format.

2.4 Switch Boards

The so-called switch boards consist of two wire-wrapped boards, connected with each other, and to the buffer card, by a set of ribbon cables. The switch board pair is shown in Figure 5, and their position relative to other components of the burn-in stand can be seen in Figure 1. Each board can serve up to four arms of the burn-in stand. These boards were constructed for a previous burn-in endeavor and were



Figure 5: The switch boards primarily provide power switching and I/O routing to and from each of the arms of the burn-in stand. Each board can handle up to four arms of eight hybrids. They are further described in section 2.4.

recommissioned for this project. The primary functions of the switch boards, with control from the PC, are to provide power switching and I/O routing to and from each of the possible eight arms of the burn-in stand. They also serve to select from which arm the current and voltage readings are made. Additionally, they provide a level of hardware protection by resetting if a voltage that is supplied to the hybrids drops below a certain threshold.

The overall function of the switchboard is to multiplex SVX4 operation patterns sent to the eight arms of the burn-in stand. Any combination of the eight arms may be used for burn-in. A common set of signal lines (D0–D23) is used both for configuring the arms to receive signals from the computer and subsequently for passing operation patterns to the arms. A somewhat complex set of strobes is used to coordinate these processes.

There is one main strobe, `WRITE_STROBE`, which in turn generates other strobes (S0–S9), depending on the state of signal lines D16–D18 (D16–D18 are address lines that select which strobe to generate). `WRITE_STROBE` is generated by the computer; S0–S9 are generated on the switchboard. Each of S0–S7, in conjunction with various other signal lines, controls the state of the corresponding switchboard arm (whether or not the arm is powered and whether the bus lines for that arm are enabled or tri-stated). Once all arms have been powered and their buses have been enabled, patterns are sent to the arms. These patterns will step the SVX4 chips through acquire, digitize, and readout exercises. After the pattern has run, data from one hybrid can be read back into the computer for analysis. S8, in conjunction with signal lines D12–D14, selects which arm to read data from, and in conjunction with signal lines D8–D10, selects which hybrid on that arm the data comes from. At various times, the currents drawn by the hybrids need to be checked. S9, in conjunction with address lines D12–D14, selects which arm to read currents from, and in conjunction with address lines D8–D11, controls which hybrid's currents on that arm will be read.

2.5 Pattern Board

There is one pattern board for each arm of the burn-in stand. Figure 1 shows how the pattern boards are connected to the two switch boards. The pattern boards, one of which is pictured in Figure 6, have been designed by LBNL, with some input from the UC Davis group for the latest version.



Figure 6: Each pattern board contains a FIFO for sending patterns to the hybrids and a FIFO for receiving data coming from the hybrids. They also contain the oscillator chip that controls the send and receive rates of these FIFOs, and send the front-end and back-end clock signals to the hybrids. More information about the pattern boards is available in section 2.5.

Each pattern board contains a FIFO for sending patterns to the hybrids. This FIFO is used in either a burst mode, where the bit stream it contains is sent just once (as in the case of initialization), or in a cycle mode, where the stream is repeatedly sent to the hybrids (as in the case of exercising the chips with a repeating pattern of acquire, digitize, and readout). The pattern boards contain a second FIFO for receiving data returning from the hybrids, in the event that a particular hybrid is selected for readout. The output coming from any given hybrid enters the pattern board from ten bus lines, eight of which carry the output *data* signals. Because the switchboard can only handle nine bits, the pattern board contains logic to select between the last two bits, which contain information useful for checking data integrity. One of these is *priority out*, which contains the initialization bit stream that has previously passed into a hybrid (as *priority in*), from chip to chip within that hybrid, and then out of the hybrid. The other is *Odd Byte Data Valid*, which alternates between “1” and “0”. The pattern board allows for the selection of one of these two bits to occupy the ninth bit, referred to as **bit9**, of the readout system.

Each pattern board also contains an oscillator that is used to drive both the output from the pattern board send FIFO and the store timing into the pattern board receive FIFO. The outgoing pattern contains the front-end and back-end clock signals for the hybrids, so these clocks either run at the frequency of the oscillator or at an integer fraction thereof, depending on so-called pattern files on the PC. During the commissioning of the hardware 30 MHz oscillators were used, and for the close-out project 40 MHz oscillators were used, in part to match the initial tests run at LBNL. Once the most advanced version of each hardware component was available, tests were done to verify that it could successfully be run at 50 MHz, which was near the expected operating frequency for use at CDF (53 MHz). One challenge of operating this system at high frequencies involves the selection of **bit9** as described

in the previous paragraph. If, for example, a 50 MHz oscillator is being used, there is a 20 ns period for the store FIFO and just 10 ns during which an incoming signal can change from low to high or high to low and still reliably be recorded. However, on early versions of the pattern board, the logic that selected which signal to use for `bit9` delayed the chosen signal by 12 ns with respect to the eight data bus lines. As a first step to fixing this, the selection circuit was sped up to 5 ns using a faster chip. Ultimately this particular problem was eliminated by putting all of the bus lines through the same set of “selection” logic (this circuit was added to the final version of the Multiplexed Scrambler Regulator board that is described in the next section). More details of the challenges presented by the timing of this system are also presented in the next section.

2.6 Multiplexed Scrambler Regulator Board

A new version of the Multiplexed Scrambler Regulator (MSR) board was designed for this project by UC Davis. Each is a printed circuit board measuring approximately 80.5 cm by 19.0 cm and each has connectors to receive eight hybrids. The functionality of the MSR includes the following: it distributes initialization signals and control patterns to the eight hybrids, it distributes analog and digital voltage to the hybrids, it selects among hybrids for data readout, and it selects among hybrids for current and voltage measurements.

Figure 7 is a photograph of the most advanced version of the MSR. Input and output connectors between the MSR and the associated pattern board are located on one of the short ends of the MSR. Bus lines for the input signals traveling to the hybrids, and for the output signal returning from the hybrids, run the length of the board. These lines are tapped at eight locations along the board for the individual hybrid ports and are terminated at the far end. The circuitry near each hybrid connection is similar for each of the eight ports. It serves to receive the bus line signals, pass signals on to the hybrids, receive signals back from the hybrids, and drive those signal back along the MSR toward the store FIFO on the pattern board. As part of this design, all of the hybrids connected to a given MSR receive the same input signals. However, there is selection circuitry to control which hybrid port drives a signal back up the readout bus lines. Also, analog and digital voltage is distributed to all connected hybrids from on-board regulators, and power is either enabled or disabled for all of the hybrids on a given MSR at the same time. Additional selection circuitry controls which of the hybrid current and voltage monitor signals are sent back through the switch board to the PC. This selection circuitry can quickly scan through all of the hybrids during a periodic current and voltage monitoring step in the burn-in cycle.

The length of the bus lines running along this board introduce two categories of challenges. Both have been addressed by modifications made to the MSR between the prototype version and the final version. The first relates to driving clean signals over this distance. In particular, the chip that drives the readout signal back up the bus line was upgraded to a more powerful model. In addition, the termination was optimized at the far end of the bus lines opposite the input and output connections

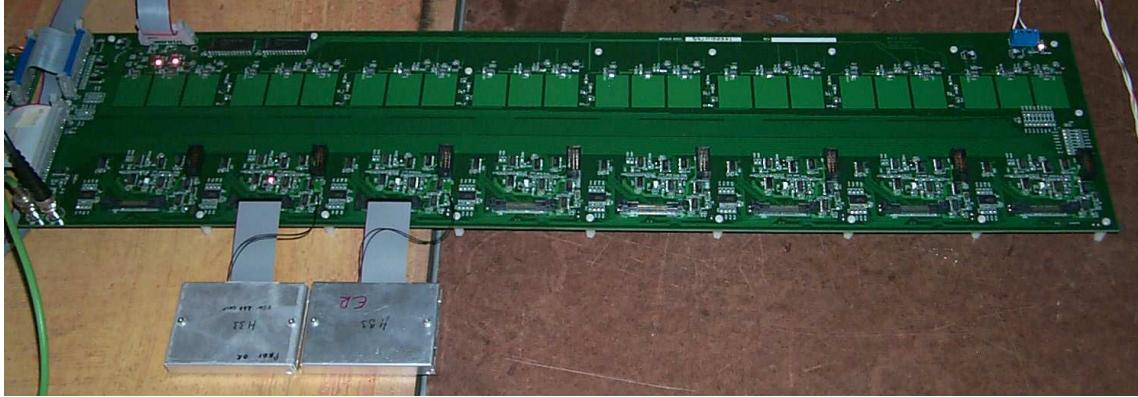


Figure 7: The MSR board distributes initialization and control signals, including the burn-in pattern, to the eight hybrid ports that can be seen along the lower edge of this photograph (only two hybrids are connected at this time). It also distributes analog and digital voltage to the hybrids, selects among hybrids for data readout, and selects among hybrids for current and voltage monitoring.

with the PATT board. The second challenge due to the length of the MSR board relates to timing. As discussed at the end of section 2.5, the logic on the PATT board that selects which signal to use for `bit9` delays the `bit9` signal by 5 ns. To compensate for this, the final version of the MSR board adds this delay to all of the other output lines by sending them through similar logic. A second problem relating to timing is that the signal coming from each successive port along the MSR is delayed by an additional 2 ns, just due to the additional signal propagation distances. For running at the higher clock frequencies of 40 MHz or 50 MHz this becomes an issue due to the difference in the arrival times of the output signals from the extreme ports (the first port and the last port) as compared to the window allowed by the oscillator that drives the store FIFO on the PATT board. This difficulty was overcome by logically splitting the MSR board into two halves. When any of the first four ports are selected for readout, the front-end clock signal that is sent to the hybrids is delayed by several nanoseconds. When any of the last four ports are selected for readout, no delay is added. Thus, the readout signal from any hybrid can arrive within the FIFO window.

2.7 Hybrid

The hybrids for layers 1–5 contain mounting points for four SVX4 chips (see section 2.8 for information about the SVX4 chip), and provide pads for wire-bonding input, output, and power connections to each chip, or between adjacent chips. There are also 23 resistors and 16 capacitors mounted on the beryllia (BeO) substrate. Figure 8 shows a 4-chip hybrid, as it sits mounted on an aluminum holder that is used from the time a hybrid arrives at LBNL to just before the silicon sensors are attached. There is also an aluminum cover, not shown in the photograph, that protects

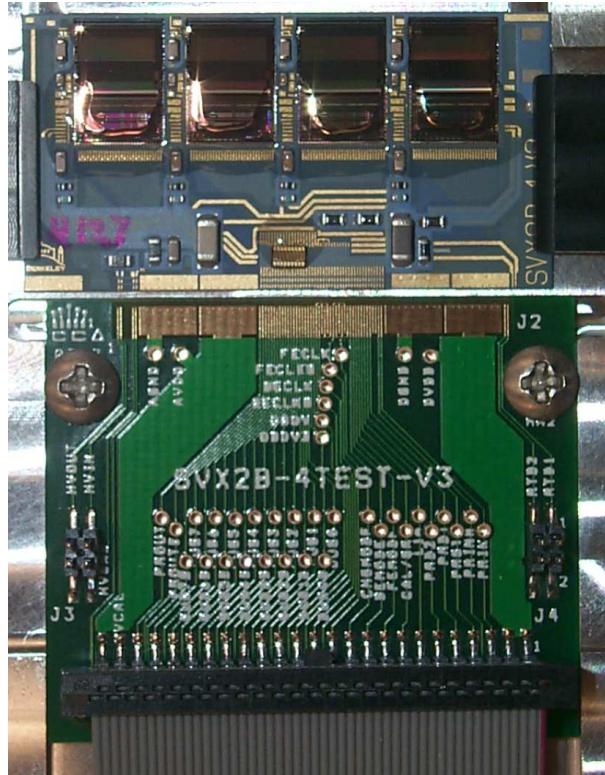


Figure 8: This photograph of a 4-chip hybrid shows the blue beryllia substrate with four SVX4 chips and surface mounted resistors and capacitors. The hybrid also contains wire-bonding pads to provide connections with the chips, connections between the chips, and connections between the beryllia board and the neighboring green board, which serves as an interface to the MSR.

the surface of the chips from accumulating dust and helps prevent accidents that could release potentially dangerous beryllia dust. For the most part, there is no reason to remove the covers during burn-in. The green board in the photograph is a small adapter board that interfaces to the beryllia substrate via wire-bonds on one side and contains a 50-pin header, onto which a connector is placed to link the hybrid to, for example, an MSR board.

A 2-chip hybrid, designed for use as part of layer 0, is pictured in figure 9. The 2-chip hybrids are mounted in a rectangular plastic box.

2.8 SVX4 Chip

The SVX4 chip is rich with features and well documented elsewhere [3]. Here, we wish to draw from existing documentation to list some of the key features of the chip (taken from reference [3]):

- 128 channels per chip
- Die size of 6.3 mm by 8 mm



Figure 9: In this picture of a 2-chip hybrid, the two SVX4 chips are visible, along with yellow connectors that lead to a larger adapter board (not shown) that interfaces with the MSR.

- Power supply voltage of 2.5 V
- Digitization speed up to 500 MHz
- Designed operating speed of up to 53 MHz
- Maximum interaction rate (i.e. acquisition period) of 132 ns
- Optimized for capacitive loads of 10–25 pF
- Channel mask for either charge injection for calibration/testing or masking channels with excessive DC current
- Built in charge injection capability
- Choice of operation in either CDF or D0 mode using external pad as selector (CDF mode includes dead-timeless operation)
- Selectable input bandwidth
- Large dynamic range on input integrator to minimize dead time due to pre-amplifier resets
- Programmable 46-cell-deep analog pipeline
- Digitization of analog signals with 8-bit resolution using Wilkinson type ADC
- Dynamic pedestal subtraction
- Data sparsification (i.e. zero suppression)
- Neighbor channel readout selection
- Low noise, with a S/N of between 10:1 and 20:1 for an input charge equal to a minimum ionizing particle, or about 4 fC
- Low power consumption of about 2 mW per channel
- Pre-amplifier gain of 5 mV/fC
- Overall gain of about 0.15 fC/bit

- Compatible with single-sided AC coupled silicon strip detectors
- Capable of daisy chain operation
- Parallel bus data readout
- Built in data valid strobe signal in the data bus lines
- Implemented in the radiation hard 0.25 micron process
- Radiation hardness of at least 20 MRad
- Single event upset register cross section less than $6 \times 10^{-17} \text{ cm}^2$

3 Software for Burn-In

3.1 Introduction to Hybrid Monitor

Hybrid Monitor is a program that automates the burn-in process for SVX4 hybrids.¹ The front-end GUI is written in Tcl 8.0.5 and Tk 8.0.5, with extension packages Tix 8.0.2 and BLT 2.4, which communicate to C function calls that control the hardware and data acquisition database I/O. This program exercises the wire-wrapped switchboard which is essentially a state-machine controlling power to the eight different arms of the burn-in stand as well as enabling and disabling flip-flops and tri-stating bus lines. See section 2 for more details on this. The data acquisition database is BerkeleyDB 2.6.4, which is an older version of the BerkeleyDB developed by Sleepycat Software.

The burn-in tasks that Hybrid Monitor performs are straight forward. The hybrids are burned-in for approximately 72 hours. At the beginning and end of the burn-in process, a program called Htest is run to characterize the hybrid; the data gathered from Htest are stored in the BerkeleyDB database and are later used to analyze the hybrids. In between these Htest characterizations, the hybrid is exercised continuously and put through the states of acquire, digitize, and readout. A bias check is performed every minute and a simple readout check is performed every ten minutes to ensure the hybrids are operating properly. Should any measured bias parameter be outside of normal operational levels, Hybrid Monitor will shut down power to prevent any damage to the hybrids or related hardware.

3.2 Hybrid Monitor GUI

The graphical user interface to hybrid monitor is organized as a notebook with six tabs; they are: Configuration, Voltage and Currents, Pedestals and Gains, Noise and Dnoise, Temperatures, and Messages. Of these tabs, the Temperature tab is the only one that is an addition to the original GUI. Thus, much of the following discussion builds upon the original documentation of the features provided by the Hybrid Monitor GUI tabs [6]. The organization of the GUI is very natural and intuitive, so extensive instructions are not essential, but a detailed explanation follows describing

¹Hybrid Monitor was originally written by Igor Volobouev (ivolobouev@lbl.gov).

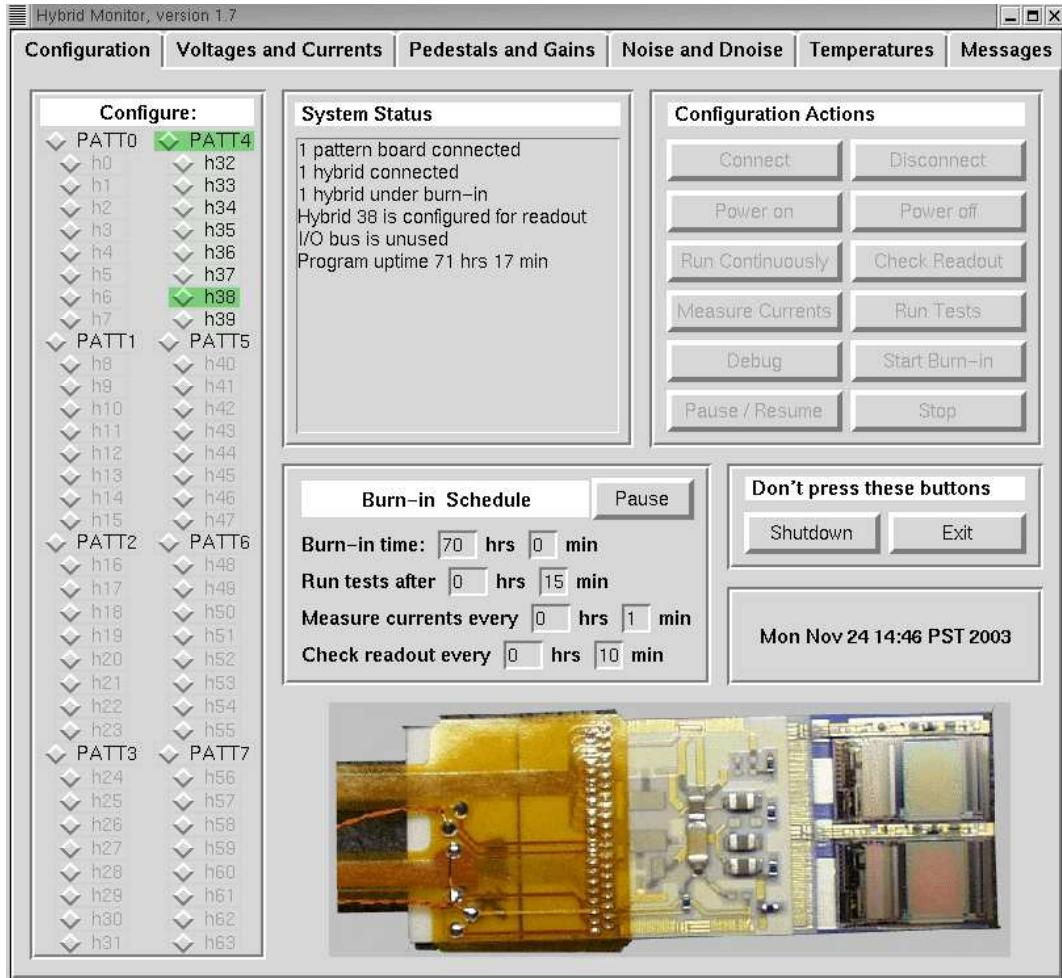


Figure 10: Screen capture showing the configuration screen of the Hybrid Monitor GUI. The functionality of the different regions of this main screen are explained in the text. The tabs to access the other portions of the hybrid monitor program, also described in the text, are visible along the top of this screen.

the various tabs and their functions. A screen capture showing the Hybrid Monitor GUI is shown in Figure 10.

3.2.1 Configuration

All hardware interactions are accessed on this page. One can connect, disconnect, and power hybrids and patterns boards; measure currents and voltages; run debug software; check hybrid readout; start, stop, or pause the burn-in process.

Hardware Selection Frame This is the leftmost frame labeled “Configure.” This frame is used to select hybrids and pattern boards where configuration/measurement actions are to be performed. The PATT0–PATT7 entries designate the pattern board

and the h0–h63 designate hybrids that might be connected downstream of those pattern boards. The color of each entry cell indicates the five possible states the hardware can be in. The color legend is located just below the hardware selection frame, and gray, which is not included in the legend, is implicitly understood as the disconnected state. A hybrid is not selectable until a pattern board has been connected.

Status Frame This is the top middle frame that displays various types of status information of the selected hardware component. The system-wide status can be viewed by clicking the top “configure” button on the hardware selection frame.

Configuration Actions Frame This is the top right frame. Logically, one cannot burn-in a hybrid without first connecting and powering it, so buttons here are enabled or disabled depending on the state of the selected hardware components. The meaning of the actions should be evident by the labeling.

Burn-in Schedule This is the lower middle frame that displays the burn-in schedule information (i.e. wait time before the first Htest, length of burn-in, etc.) for all hybrids on the system. The fields in this frame cannot be altered via the GUI, but can be altered by editing the `globals.tcl` file and changing the appropriate tcl-tk parameters. For more information on altering these parameters see section 3.4.

Don’t Press These Buttons This oddly named frame is used only when it is necessary to power down the whole system or exit the program. If Htest is running while one of these buttons is pressed, then Htest will try to finish up what it is doing and exit quietly. The “Exit” button actually exits the program, while the “Shutdown” button just shuts off power to all arms on the wire-wrapped switchboard.

3.2.2 Voltages and Currents

Voltage and current measurements take place one time when you power a hybrid, periodically throughout the burn-in process, and whenever the user clicks the “Measure now” button located at the bottom right of the Hybrid Monitor Voltages and Currents screen. When these measurements take place, all powered hybrids are measured. Since, over time, there can be multiple bias measurements performed on a given hybrid, Hybrid Monitor only displays the most recent measurement.

There are two types of user imposed bias voltage and current limits: “Warn” and “Turn off.” These can be set by the user in the lower right box. If the particular hybrid bias parameter exceeds the “warn” parameter then a warning beep is generated and a message is displayed in the Messages tab. If a “Turn off” parameter is exceeded, then AVDD & DVDD are turned off for the entire arm that the hybrid is connected to, as well as the pattern board. This has the side effect of shutting down all hybrids connected to that particular MSR board, and serves to protect all of the hybrids on that arm from possible damage.

3.2.3 Pedestals and Gains

When a readout check is performed once every ten minutes (or however often the user has selected), the results for each hybrid are displayed on this page. The plots shows ADC counts versus channel number. In addition to being performed at regular intervals during burn-in, readout checks can be initiated by selecting a hybrid and clicking on the “Check Readout” button on the main Hybrid Monitor configurations screen. Only the plots from the most recent readout check are displayed.

3.2.4 Noise and Dnoise

Noise is measured for each channel by calculating the standard deviation of all pedestal readings for 230 events. Dnoise is measured by finding the standard deviation of the difference between two neighboring channels for 230 events. The plots on this page are generated when readout checks are performed and display information from the most recent readout check.

3.2.5 Temperature

This page displays the temperature for each hybrid, if available. It also displays the currents of the hybrid as a convenience to see if there is any correlation between a high temperature and current draw. The temperature monitoring of the hybrid is performed by measuring the voltage across an RTD (Resistance Temperature Detector) resistor and then looking up that voltage to find the corresponding temperature. The precision of the measurement is to half a degree Celsius. Since this feature was not in the original design, but was instead added later, the hardware can take temperature measurements on the seven ports of each MSR board that are closest to the pattern board but not on the eighth port.

3.2.6 Messages

All warnings and error messages that arise through the use of Hybrid Monitor are displayed on this page. Debugging information can also be displayed on this page if a particular Tcl variable is set correctly. To see debugging information, set the tcl variable `proc_sequence`, contained in `hybmon.tcl` from “0” to “1”. Tcl-Tk process communication messages are also displayed here as well as simple status messages about Hybrid Monitor. For instance, if Hybrid Monitor needed to send a message to Htest, the message would be displayed in this window. Finally, an indexed help feature is accessible from this page.

3.3 Svxscope & Htest

Svxscope and Htest are two independent programs that Hybrid Monitor uses to facilitate the burn-in of hybrids. When the need arises to use these programs, Hybrid Monitor forks and runs these programs as a child processes using `execv()` (a C system call). The reason why Svxscope and Htest are separate programs from

Hybrid Monitor is for robustness. If for some reason the child process (Svxscope or Htest) terminates prematurely, the rest of the burn-in can still proceed, as the parent process (Hybrid Monitor) continues to run. If these processes were a part of Hybrid Monitor and a burn-in terminated prematurely, the entire burn-in process would halt and would be left in an incomplete state, with no simple way to resume. With the parent/child arrangement, much of the burn-in can continue through to completion. The next two sections describe the Svxscope and Htest programs in more detail.

3.3.1 Svxscope

The user of Hybrid Monitor has the ability to debug a particular Hybrid in the midst of burn-in by selecting a hybrid and clicking “debug” located on the Configuration Tab. The debugging tool is Svxscope, a screen capture of which is shown in Figure 11. It has the necessary features to communicate with Hybrid Monitor (via special internal messages) in order to operate without bus contentions between the two programs.

Svxscope displays real time plots of the readout data coming from a hybrid, including the pedestal and noise. It is limited by the fact that it can only run the same readout pattern over and over again. However, a useful feature of Svxscope is that it allows one to see how the hybrid performs with various initialization settings. The user may manually alter these initialization settings and then reload the pattern by clicking the “Reload Pattern” button. As a result, Svxscope is mostly used as a debugging tool to check that everything is alright with the hybrid and the hardware related to hybrid readout. It is one of the best tools for observing the data returning from a hybrid while, for example, adjusting the timing of the readout circuitry of the pattern boards. Svxscope also provides a tool to dump the raw data to the screen for inspection (an example is shown in Appendix A). This dump includes, for each chip of the hybrid, the chip ID, the memory cell ID, and a series of pairs consisting of the channel ID and ADC value for each channel. It also includes `bit9`, which in this case carries *odd byte data valid*.

3.3.2 Htest (also known as Htwish)

Htest was also originally written for the SVX3, but was modified to characterize SVX4 chips.² The Htwish Manual [7] provides a description of the tests and how they are implemented and executed. These Htests range from testing the functionality of certain registers to advanced features of the hybrid such as sparsification, or characterizing rise-times and gains. The data from the tests are also analyzed by the analysis part of Htest, which is run separately after a burn-in is complete, and can be used to generate some useful performance and diagnostic plots. In short, this is the main testing and analysis program for hybrids.

²This modification was performed by John Freeman, Paul Lujan, and Marc Weber at LBNL.

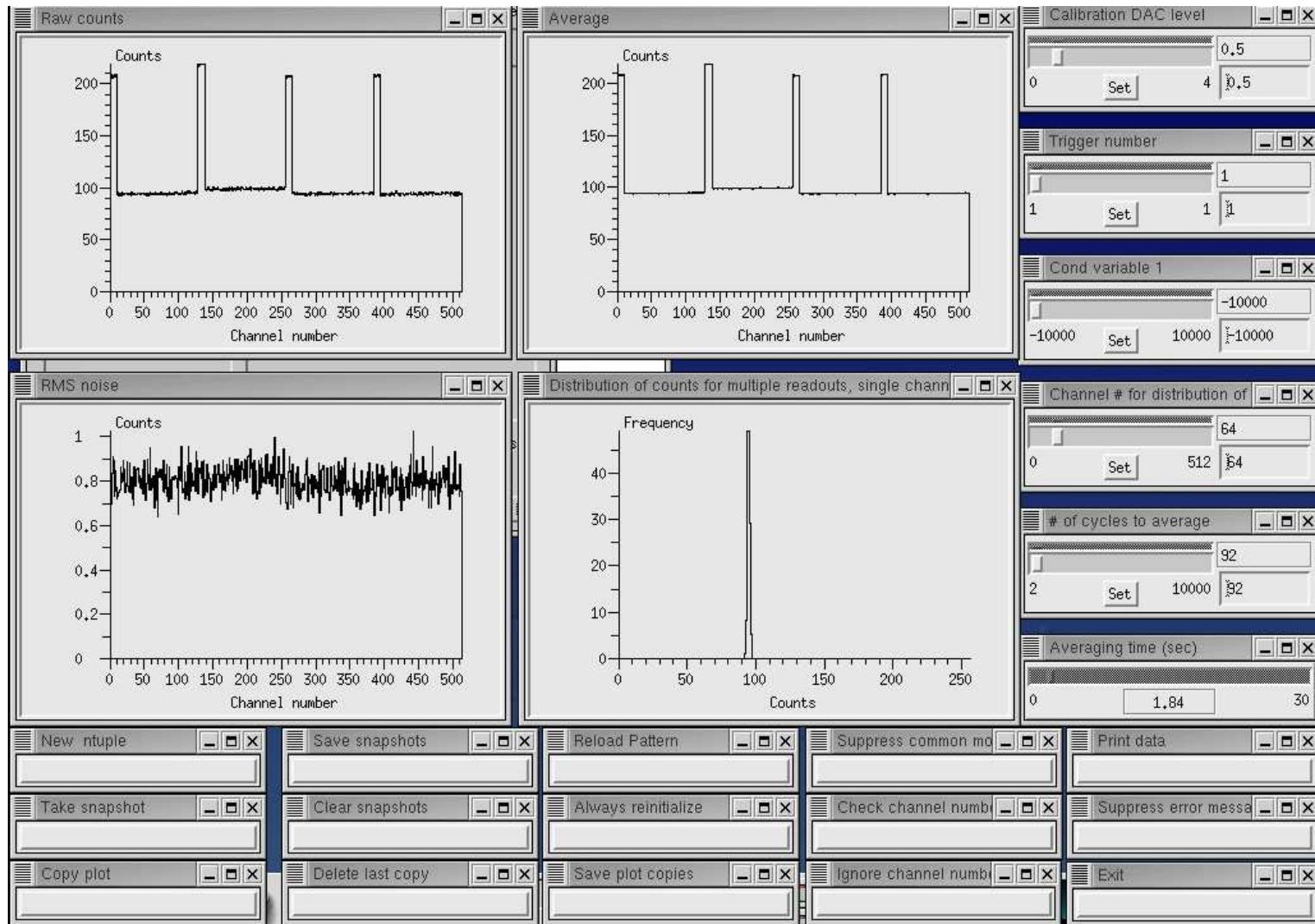


Figure 11: The graphical user interface for the Svxscope program. Svxscope allows for real time debugging of hybrid performance by monitoring the data structure and values returning from a hybrid that is initialized and then repeatedly cycled through the stages of acquire, digitize, and readout. The plot on the upper left (upper right) shows the instantaneous (time averaged) pedestal levels for all 512 channels of a hybrid. Note that some of the channels are charge-injected. The plot on the lower left shows the RMS noise and the plot on the lower right shows the noise distribution for one selectable channel.

3.4 Burn-in Scheduler

The burn-in scheduler is the heart of Hybrid Monitor. The Tcl procedure is called `burnin_scheduler` and is located in the file `burnin_scheduler.tcl`. This procedure is essentially a loop that constantly checks the state of hybrids to see if they are in the “burned” state. The state of a hybrid is a software construct to keep track of what the hybrid is doing. For example, a hybrid that is not doing anything but is powered is in the “powered” state, and a hybrid running the burn-in pattern is in the “burned” state. So, until the hybrids are in the burned state, the burn-in scheduler does nothing. To transition the hybrids into the burned state, one just needs to start the burn-in. Once the hybrids are in the burned (or burn-in) state, the burn-in scheduler starts a timer to keep track of when to perform three main tasks: measuring bias parameters, checking the readout, and performing Htests on the hybrid. The burn-in scheduler can be paused by simply clicking on the “Pause” button on the Configurations page and can be resumed by toggling the same “Pause” button that would now be labeled “Run.”

As mentioned before the bias measurements are performed every minute, a readout check is typically performed every ten minutes, and an Htest is done at the beginning and end of each 72 hour burn-in to each hybrid. The burn-in scheduler is responsible for ensuring that all of this is completed. It is important to note that each Htest takes about 25 minutes to run per hybrid. Although this time is considered burn-in time, Hybrid Monitor does not account for it. Hybrid Monitor only counts the time when the hybrid is not doing readout checks, bias measurement check, or Htests. For the purpose of recording how long a hybrid has been exercised, we include the time during which there have been bias measurements, readout checks, Htests, and the basic burn-in patterns, which dominate the total burn-in time.

3.5 Data Acquisition

During the entire burn-in process, Hybrid Monitor is constantly acquiring information about the hybrids and their performance. Most of this information is stored in the BerkeleyDB database [8], but text files are also generated. These files are briefly described in the following section.

3.6 Important Files Generated During Burn-in

DB Files There is one db file for each hybrid and it contains the raw data generated from the Htests run at the beginning and end of the burn-in. The data in these db files are used by the program called Testdisplay to analyze the performance of the hybrid (see section 3.7).

Htest Log Files There are two Htest log files generated for each hybrid during the course of a burn-in, one at the beginning and one at the end. These files contain pass or fail ratings resulting from each of the sub-tests that we run as part of the

Htest program. The following example shows the summary located at the end of one of these files:

```
12:54:09-W-run\tests: Hybrid h0081 test results summary:
12:54:09-W-run_tests: serial line          passed
12:54:09-W-run_tests: pipeline pedestal map  passed
12:54:09-W-run_tests: pipeline gain map    passed
12:54:09-W-run_tests: pipeline gain 2      passed
12:54:09-W-run_tests: linearity            passed
12:54:09-W-run_tests: capnoise             passed
12:54:09-W-run_tests: pipeline delay       passed
12:54:09-W-run_tests: adc limits           passed
12:54:09-W-run_tests: common noise suppression  passed
12:54:09-W-run_tests: sparse readout       passed
12:54:09-W-run_tests: sparse with neighbors  passed
12:54:10-W-run_tests: Test results summary completed
12:54:10-W-hyptest: Finished tests of hybrid h0081 on Fri Oct 31 12:54:10 2003
```

Current and Voltage File A file called `c_and_v.txt` contains a list of all of the bias measurements for all of the hybrids during the burn-in. This includes the analog voltage, the analog current, the digital voltage, and the ground level, as well as the temperature for hybrids that are on ports that have a temperature readout circuit. The date and time of every measurement are also recorded in this file.

3.7 Testdisplay

Testdisplay is the analysis portion of Htest. See Figure 12 for a screen capture of the Testdisplay graphical user interface. It is a very powerful program because it allows one to look directly at the raw data to determine, interactively, how the hybrid behaved during testing, and it also contains components that run an analysis to determine the quality of the hybrid. Testdisplay determines the quality of the hybrid by assessing whether or not measured parameters corresponding to each of the various features of the hybrid fall within acceptable ranges (cuts). Appropriate ranges have been selected for operating the hybrids at 40 MHz.³ It is important to have ranges set for the particular frequency at which Htest is run because different frequencies affect the rise-times and pedestals in different ways. For example, higher frequencies make for a faster back-end clock (`be_clock`), which has the effect of pushing pedestals higher, so if the cuts are not properly set, the Testdisplay analysis might indicate a failure where none actually exists.

Testdisplay also generates a useful debugging, quality assurance, and quality control tool called the quality tree. This quality tree is a way to get an overall picture of the performance of the hybrid. It displays color-coded results for many of the hybrid features that the Testdisplay program probes, and it organizes these results into a tree structure where information about a failure at a low level (“leaf”) propagates up to a higher level (“branch”), providing a very quick visual indication of the hybrid performance.

There are three more files that Testdisplay creates for analysis purposes. They are files that contain summarized results of the Htests that measure linearities, rise-times, and pedestal levels. All three of these files are human readable text files,

³Many of these determinations were made by Paul Lujan and Marc Weber at LBNL.

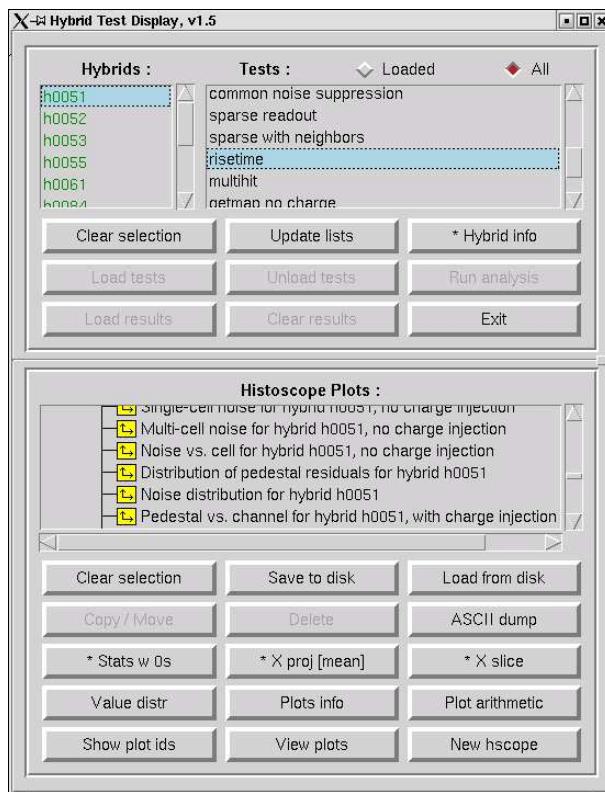


Figure 12: Screen capture showing the user interface of the Testdisplay program that handles the post-burn-in analysis portion of Htest.

but they are used, in an additional processing step, to create various plots. The linearities and rise-times files are used by another program to create plots that can be used to compare these two features of the chips before and after the burn-in (see section 4 for more information on these plots). The pedestal information in the third file is used to generate plots that can be compared with similar information produced during an initial testing phase at LBNL, effectively also providing a comparison of performance before and after burn-in.

4 Quality Assurance and Quality Control

4.1 Motivation

A central component of the burn-in task is to determine whether each hybrid meets a high level of performance requirements during and after burn-in. In particular, any hybrid that shows a statistically significant degradation in performance during the burn-in should not be included in a final detector. The performance assessments can be divided into two categories: (1) comparisons to references or predetermined specification tolerances and (2) stability or reproducibility over time. Of course the two categories are overlapping, since any performance that satisfies category 2 must also satisfy category 1 if the hybrid is to be deemed acceptable. Tests run at the end of burn-in are sufficient to address category 1, while addressing category 2 requires tests at multiple times, and this is one reason why measurements are made throughout assembly at LBNL and before and after burn-in at UC Davis.

Section 4.2 discusses the logistics of data handling and analysis, as used to facilitate our assessment of the quality of the hybrids. Section 4.3 then documents the various tools and pieces of information that are available in order to decide whether a hybrid passes the burn-in phase.

4.2 Logistics

The primary hybrid performance test suite, called Htest (already discussed in section 3.3.2), produces BerkeleyDB files, log files, and additional ASCII formatted summary files when it performs the tests at the start and end of burn-in (as discussed in sections 3.6 and 3.7). Current and voltage measurement results are also periodically appended to a file for later analysis. When the burn-in completes, all of these files are resident on the computer that runs the Hybrid Monitor control program. The next step is to run a shell script that carries out the following: moves the files to a backup location on the burn-in PC to reduce any chance that they get overwritten; copies those files that are ready to be used in performance evaluations or simply kept for archival purposes to the computer that hosts the Hybrid Über Tracker Database (to be discussed in section 5); and copies those data files that need further analysis to the database host computer where they are accessible to the computer on which the additional analysis procedures are run.

The Testdisplay component of Htest, as introduced in section 3.7, is used to load

in the test results, run an analysis of the data (e.g. organizing data by chip, channel, cell, or parameter setting, and comparing performance to expected values), produce a quality tree of the test results (see section 4.3.1), and produce an additional set of ASCII files for use in generating plots. After the test results have been loaded and analyzed for all of the hybrids participating in a given burn-in, a second shell script is run to carry out the following actions: use MINUIT to generate a set of plots based on the reorganized data for comparison to similar plots generated during initial testing at LBNL (these so-called “traveler plots” are described in section 4.3.2); use ROOT to generate a set of plots that show performance characteristics both at the start and end of the burn-in at UC Davis on the same plots (these so-called “before-and-after plots” are described in section 4.3.3); convert all of these plots into a format suitable for viewing on the web; move files and plots to the computer that hosts the database; and remove temporary files.

As another step in preparation for evaluating the hybrids, we run a ROOT macro to parse the ASCII file that contains time stamps, currents, and voltages in order to produce plots of the currents and voltages as functions of time. These plots are also transferred to the PC that hosts the database and are linked in appropriately.

Finally, information about the burn-in, such as which hybrids participated, start time, end time, and hybrid locations on the burn-in stand, is entered into the Hybrid Über Tracker Database (see section 5) at various times throughout the process.

4.3 Tools for Quality Assurance and Quality Control

4.3.1 Quality Tree

The quality trees, which use Htest results, provide a quick way to judge the overall performance of a hybrid and help in identifying a marginal or defective hybrid or chip component. Figure 13 shows an example quality tree, generated from the second set of Htests for hybrid h0123. A quality tree classifies a hybrid and each of its chips based upon pre-defined cuts on measurements such as risetime (using various bandwidth settings), pedestal residuals, pedestal noise, gain residuals, noise, pipeline delay, and common noise suppression, among others. The classification categories and corresponding color codes are good (green), fair (yellow), and bad (red). A chip receives the good rating if it has no bad capacitors, the fair rating if it has exactly one bad capacitor, or the bad rating if it has more than one bad capacitor, a bad channel, a bad cell, or fails one of the other Htest requirements. A colored (green, yellow, or red) circle appears next to each Htest comparison result. If there is a parameter outside of tolerance, there is text to indicate the position of the error (which might be a specific channel or cell of a specific chip). For some tests there is also a set of ratings for each of the individual chips. An example would be “good good fair good”, which would indicate that the third chip on this four-chip hybrid received the fair rating. In this example, there would also be a yellow circle next to this line of the quality tree. Any error shown for a “leaf” of the quality tree propagates up the “branches” of the tree, so that, with a glance at the top lines of the quality tree, one can see that there is a potential problem with the hybrid. This

simplifies the comparison of the three quality trees we use for each hybrid: one from the initial testing at LBNL, one from the first Htest during burn-in at UC Davis, and one from the final Htest at the end of burn-in at UC Davis.

4.3.2 Traveler Plots

Another important tool we use in evaluating hybrid performance is a set of plots that are collectively referred to as *traveler plots*. In previous hybrid production projects, forms similar to these were moved with the hybrids during each phase of production (hence the name traveler plots). However, we now typically keep only electronic versions of these documents. The travelers are produced from data collected during the initial tests performed at LBNL and the second of the two sets of Htests that are run as part of burn-in at UC Davis. Comparing these two travelers, one can observe changes in any of the plotted quantities (which are discussed below) due to the burn-in process.

The traveler consists of three pages. The first, as shown in Figure 14, contains some information that overlaps with that shown at the top of the quality tree. In addition to the hybrid serial number and traveler creation date and time, this includes the overall hybrid quality rating, the individual chip ratings, and lists of any bad channels, bad cells, or bad capacitors. It also shows the levels for dynamic common mode noise suppression. The second page of the traveler is shown in Figure 15 and contains two plots. The top plot shows the pedestal level (in units of ADC counts) for each channel, both with and without charge injection. The bottom plot shows the average (over channels within each chip) of the pedestal level for each pipeline cell. This quantity for the 46 pipeline cells is shown consecutively for each chip along the x-axis of the plot, both with and without charge injection. Figure 16 shows the third page of the traveler. The first pair of plots on this page show the single-cell and multi-cell noise for the hybrid, while the second pair of plots show the pipeline capacitor pedestal residual (relative to the chip average) and the capacitor RMS noise distribution.

4.3.3 Before-and-After Plots

Another tool employed in the evaluation of hybrid performance is a set of plots that we call before-and-after plots. These were originally developed for use in the Single Event Upset and radiation hardness testing,⁴ as a means of tracking the hybrid performance over time. With some minor adaptations these plots are used to compare the following hybrid attributes at the beginning and end of burn-in at UC Davis: pedestals, noise, gain slope (or linearity), risetime, and gain. The before-and-after plots are one of the most useful ways to check for any change or deterioration in hybrid performance due to burn-in. In what follows, each of these plots is explained, and figures are shown as examples.

Figure 17 shows the before-and-after plots that contain pedestal information for hybrid h0123. The upper-left plot displays the means of Gaussian fits to pedestal

⁴Paul Lujan of LBNL wrote much of the code to create these plots.

Hybrid h0123 Test 2 Quality Tree

```

Hybrid classification: good
Chip quality: {good good good good}
Bad channels: {}
Bad cells: {}
Bad capacitors: {}
DCMS location: {38.3 38.4 37.7 36.7}

```



Hybrid h0123 Traveler

December 22, 2003

1 Data Analysis Summary

The following summarizes the findings of the htwish analysis program.

```
% Hybrid h0123 classification info. Created 14:49:43 12/22/03
```

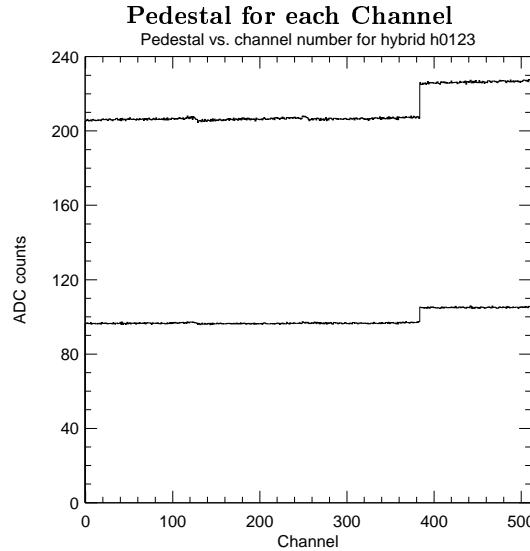
```
Hybrid h0123 quality: good
Chip classification: good good good good
Bad channels: none
Bad cells: none
Random bad capacitors: none
Channels at pedestal needed for DCMS: 38.3 38.4 37.7 36.7
```

Note: Chips are numbered from 1 (first in readout, on the right of the hybrid) to 4 (last in readout, on the left of the hybrid).

Figure 14: An example first page of a traveler, in this case summarizing hybrid and chip quality based on tests performed at the end of the burn-in of hybrid h0123.

2 Pedestal

The following plot shows the pedestals vs. channel number with and without charge injection:



The following plot shows the average pedestal vs. pipeline cell ID with and without charge injection. The x-axis is cell ID offset by 46 for each chip. Thus cell ID 5 for chip 1 is at $x=5$, chip 2 at $x=51$, chip 3 at $x=97$, etc.

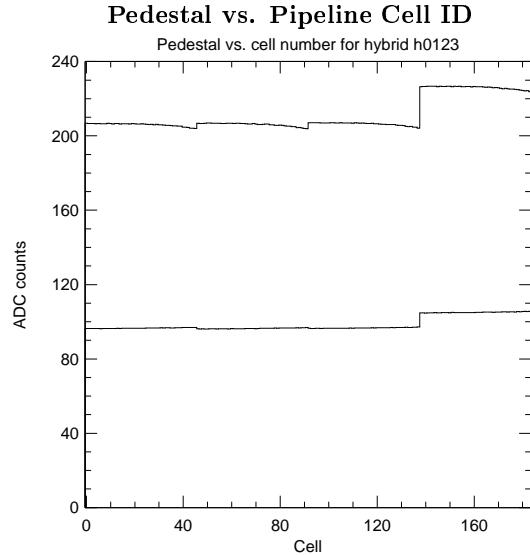
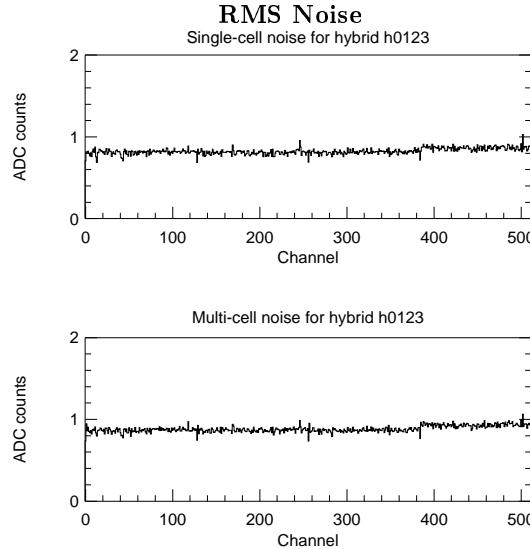


Figure 15: Second page of the traveler for hybrid h0123, with data from the tests performed at the end of burn-in. As shown on the traveler, and explained in the text, these plots show the pedestal levels versus channel and versus pipeline cell, both with and without charge injection.

3 Noise

The following plots show the single-cell and multi-cell RMS noise:



The following are histograms of the pedestal of each pipeline capacitor relative to the chip average (top); and the RMS noise of each capacitor (bottom):

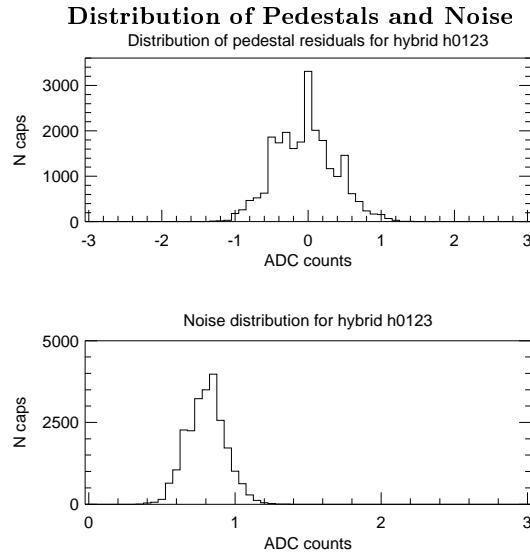


Figure 16: Last page of the traveler for hybrid h0123, with data from the tests performed at the end of burn-in. The top plots show the single-cell and multi-cell noise, while the bottom plots show the pipeline capacitor pedestal residual (relative to the chip average) and the capacitor RMS noise distribution.

levels (in units of ADC counts) at the beginning of burn-in (black) and at the end of burn-in (red) as functions of channel number. The upper-right plot shows the pedestal level for channel 64 at the start and end of burn-in, labeled as run 1 and run 2 respectively, along the x-axis. In the lower-left is a 3-dimensional plot of pedestal level, channel, and run.

Figure 18 shows the before-and-after plots that include noise distributions for hybrid h0123. The upper-left plot displays the means of Gaussian fits to noise levels (in units of ADC counts) at the beginning of burn-in (black) and at the end of burn-in (red) as functions of channel number. The upper-right plot shows the noise level for channel 64 at the start and end of burn-in, labeled as run 1 and run 2 respectively, along the x-axis. In the lower-left is a 3-dimensional plot of noise level, channel, and run.

Figure 19 shows the before-and-after plots that display information about the gain slope, or linearity, for hybrid h0123. The upper-left plot shows the gain slope (in units of ADC counts per volt) at the beginning of burn-in (black) and at the end of burn-in (red) as functions of channel number. The upper-right plot shows the gain slope for channel 64 at the start and end of burn-in, labeled as run 1 and run 2 respectively, along the x-axis. The lower-left plot is a 3-dimensional plot of gain slope, channel, and run. Finally, the lower-right plot shows the χ^2 per degree of freedom, as a function of channel number, for the fit that is used to extract the gain slope.

Figure 20 shows the before-and-after plots that contain information about risetime performance for hybrid h0123. The upper-left plot shows the risetime (in units of clock cycles), as a function of channel number, for a band-width setting of 1, where the black curve is for the beginning of burn-in and the red curve is for the end of burn-in. Note that the risetime of the SVX4 chip with a band-width setting of 1 is often too fast for the fitting algorithm that is used to measure the risetime. Therefore, results of this plot are only alarming if they are much above a risetime of 1 clock cycle or exhibit some other pathological behavior. The upper-right plot shows the same thing, except for a band-width setting of 8, where the fitter consistently does a better job. The lower-left plot is a 3-dimensional plot of risetime (in this case for a band-width setting of 15), channel, and run. The lower-right 3-dimensional plot shows the risetime for channel 64, for three different bandwidth settings, for the two different runs.

Figure 21 shows the before-and-after plots relating to gain for hybrid h0123. The upper-left plot displays the gain (in units of ADC counts per volt) at the beginning of burn-in (black) and at the end of burn-in (red) as functions of channel number. The upper-right plot shows the gain for channel 64 at the start and end of burn-in, labeled as run 1 and run 2 respectively, along the x-axis. In the lower-left is a 3-dimensional plot of gain, channel, and run.

Experience has shown that changes in the quantities represented in the before-and-after plots have nearly always been traced to a software or configuration problem with the burn-in stand rather than the hybrids themselves. The issue with the band-width setting 1 risetime fitting algorithm was mentioned above. Another example occurred when we found that when a given hybrid was included in more than one

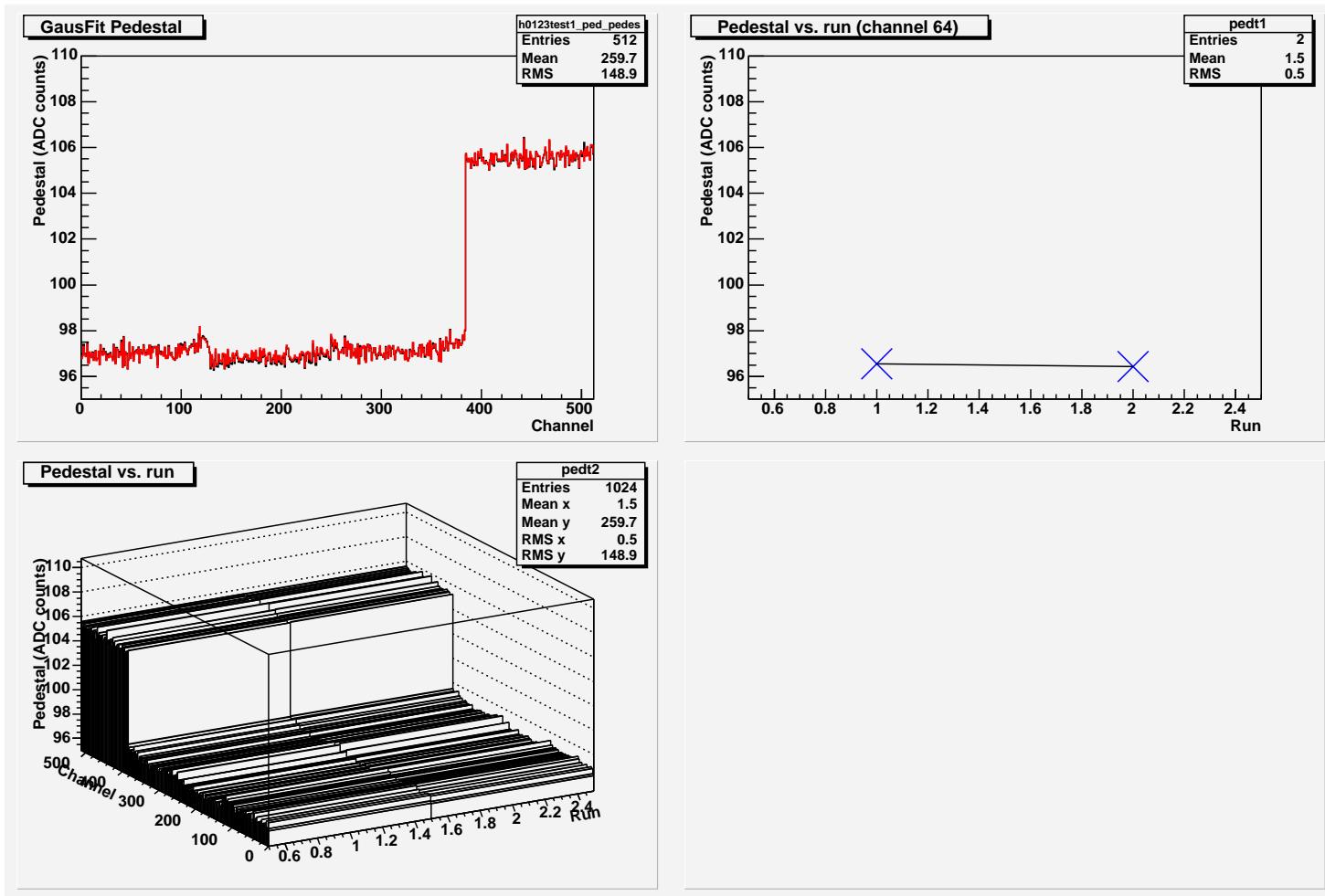


Figure 17: Before-and-After plots of pedestal levels for hybrid h0123. Upper-left: means of Gaussian fits to pedestal levels at the beginning (black) and end (red) of burn-in as functions of channel number. Upper-right: pedestal level for channel 64 at the start (run 1) and end (run 2) of burn-in. Lower-left: 3-dimensional plot of pedestal level, channel, and run. The plots on the left indicate that chip 4 of this hybrid has a higher pedestal than the other 3 chips. There are no significant changes to the pedestal due to burn-in.

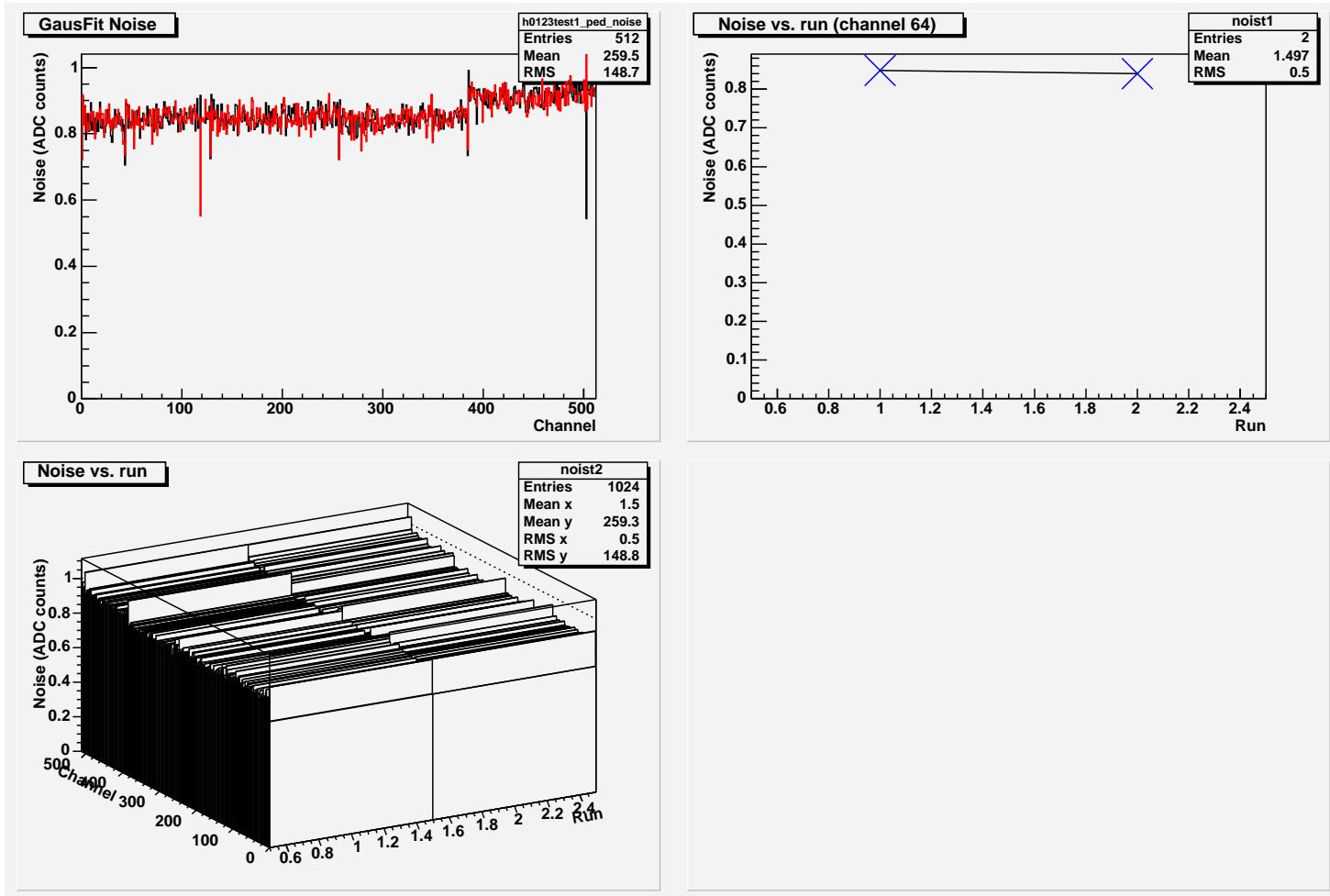


Figure 18: Before-and-After plots of noise levels for hybrid h0123. Upper-left: means of Gaussian fits to noise levels at the beginning (black) and end (red) of burn-in as functions of channel number. Upper-right: noise level for channel 64 at the start (run 1) and end (run 2) of burn-in. Lower-left: 3-dimensional plot of noise level, channel, and run.

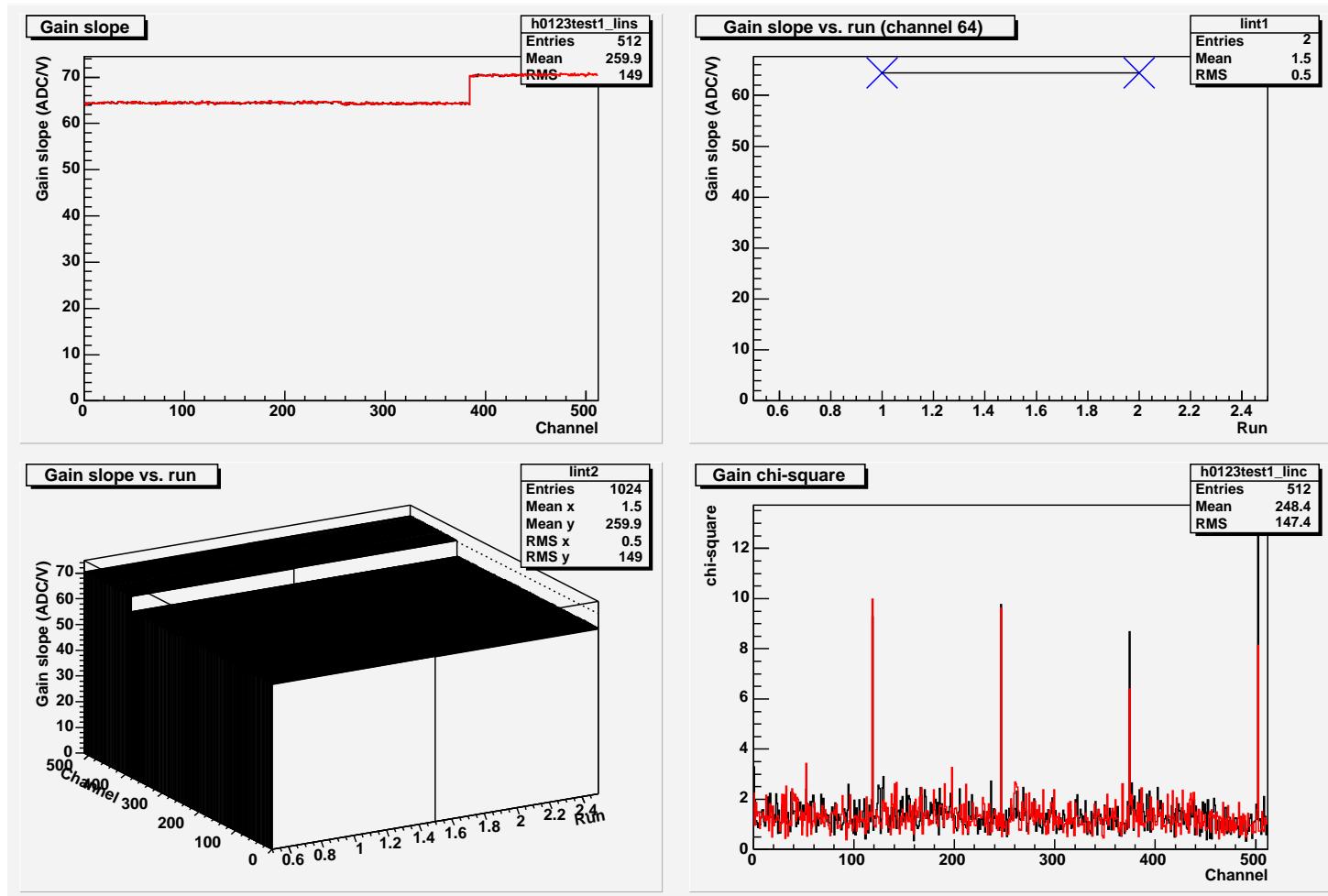


Figure 19: Before-and-After plots of gain slope for hybrid h0123. Upper-left: gain slope at the beginning (black) and end (red) of burn-in as functions of channel number. Upper-right: gain slope for channel 64 at the start (run 1) and end (run 2) of burn-in. Lower-left: 3-dimensional plot of gain slope, channel, and run. Lower-right: χ^2 per degree of freedom versus channel for gain slope fit.

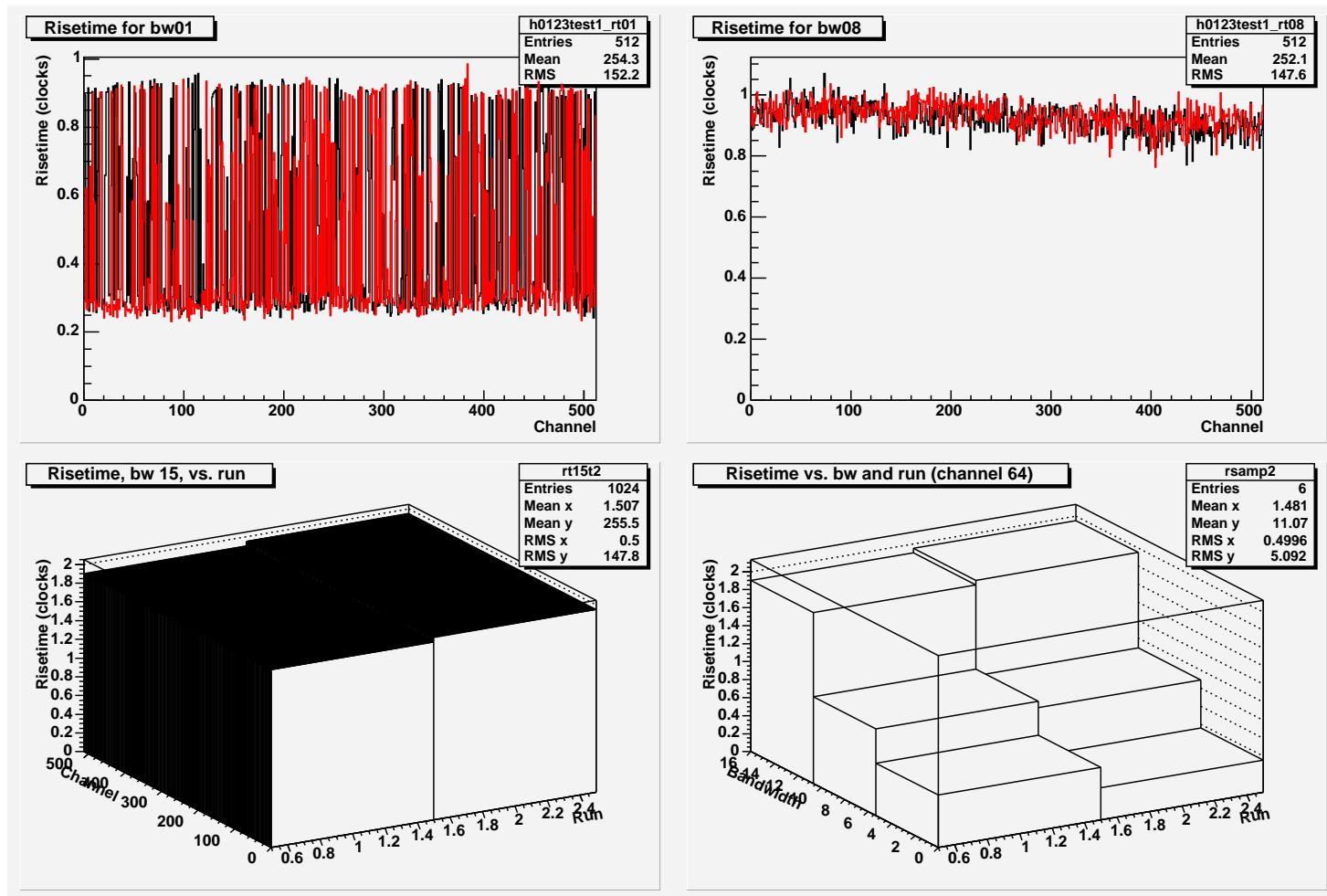


Figure 20: Before-and-After plots of risetime for hybrid h0123. Upper-left: risetime for band-width setting 1 at the beginning (black) and end (red) of burn-in as functions of channel number. Upper-right: risetime for band-width setting 8. Lower-left: 3-dimensional plot of risetime (band-width setting 15), channel, and run. Lower-right: 3-dimensional plot of risetime for channel 64, three band-width settings, and two runs.

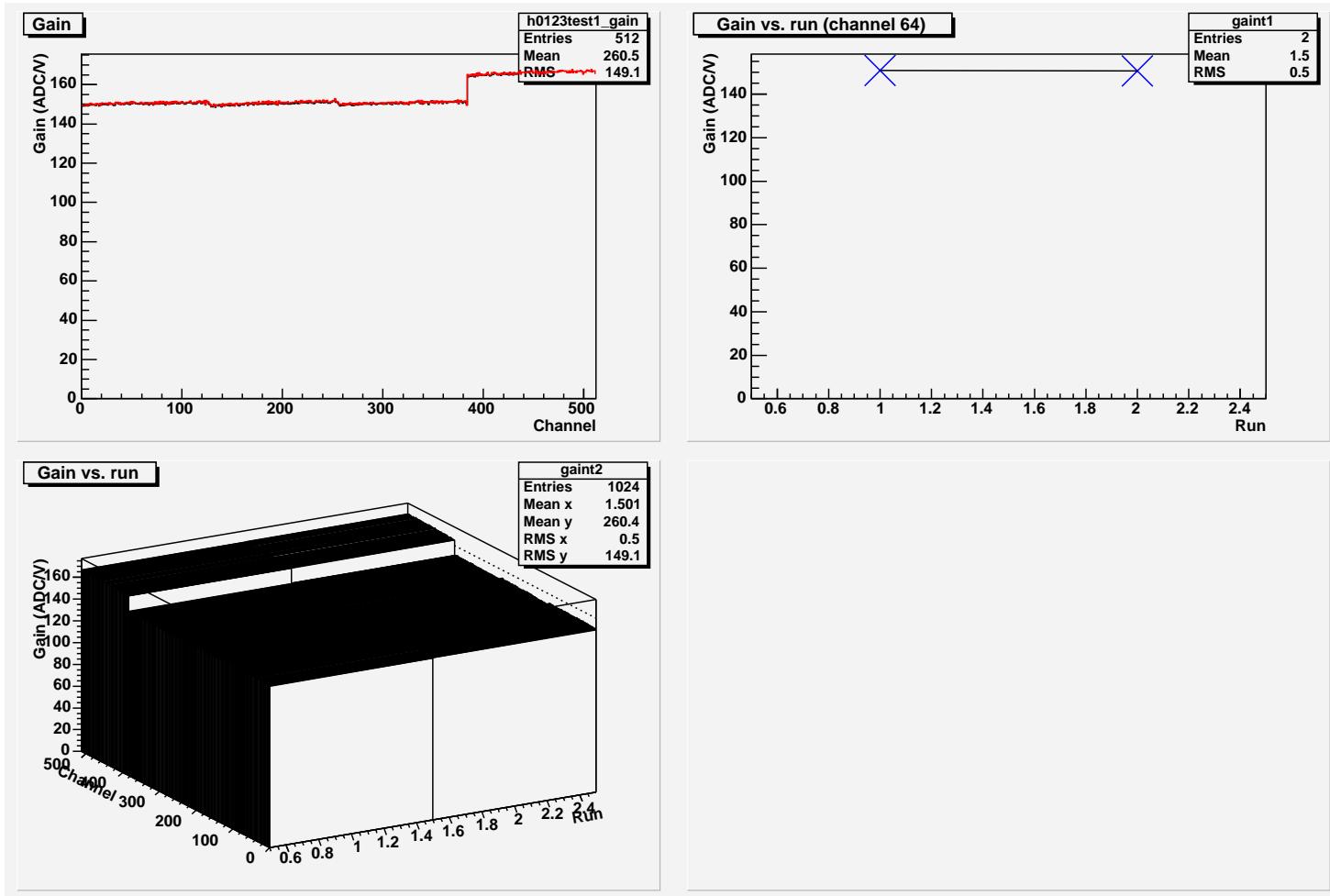


Figure 21: Before-and-After plots of gain for hybrid h0123. Upper-left: gain at the beginning (black) and end (red) of burn-in as functions of channel number. Upper-right: gain for channel 64 at the start (run 1) and end (run 2) of burn-in. Lower-left: 3-dimensional plot of gain, channel, and run.

burn-in some of the ASCII files that were used to generate the before-and-after plots were being appended to, rather than replaced, from one burn-in to the next. This became apparent after doing a test at 50 MHz and then switching back to the usual operating frequency of 40 MHz. The subsequent burn-ins of the hybrids that participated in this study had, for example, shifted pedestals from the start of burn-in to the end of burn-in because the start of burn-in curves erroneously included information from the 50 MHz operation tests. After discoveries such as these, the problem is corrected and the hybrid is put through an additional burn-in, to verify that indeed there is no change or degradation in performance. As of the writing of this document, there is one case still under investigation, but it is most likely that, as with a few other hybrids, the burn-in uncovered a pre-existing problem with the hybrid rather than damaged the hybrid.

4.3.4 Htest Log File

The Htest log file has already been discussed in section 3.6, and a portion of an example log file is shown there. One of these files is produced for each run of the Htest suite of tests, and lines near the end of the file give a pass/fail summary rating for each of the sub-tests that are part of the suite. These binary ratings do not reflect the sort of detailed cut-based performance requirements that are used in other areas of the evaluation, such as in the construction of the quality tree. Instead, these pass/fail ratings are largely based upon the structure of the returned raw data, where these data integrity checks are tailored for the particular test types.

These log files were useful while commissioning the burn-in stand hardware, such as the MSR boards. The Htest suite includes a file where one can control which subset of the available tests are run. By starting with the simplest test, the serial line test, where configuration bits are sent into the chip registers and then read back out for comparison, and adding more tests on top of that, we were able to check that the control signals for increasingly complex features of the chip were being properly sent through the multiplexing hardware. Later in the burn-in project, these log files continue to be useful indicators of problems in the setup or configured of a particular burn-in. For example, all tests fail if the connector between a hybrid and the MSR is not securely seated. Problems can thus be quickly tracked down, and the relevant hybrids can be included in additional burn-ins (for a given hybrid, there are database entries for each burn-in in which it participated, and the first one or more might have failed Htest results).

The quality assurance and quality control procedures that we have adopted include looking at these log files and making sure that either all of the Htests have passed, or we understanding the cause of any failures. If the cause for the failure is not serious, meaning that it is an indication of performance that is still within pre-defined specifications, this aspect of the overall performance evaluation does not keep the hybrid from passing burn-in. Examples of this case are the handful of hybrids that fail the “sparse readout” and “sparse with neighbors” tests. These are 2-chip hybrids made from a combination of one “A” type chip and one “B” type chip, where the pedestal and charge injected levels for the “A” chips can be as much

as 30 and 40 ADC counts higher than for “B” chips. These sparsification tests use threshold levels for including or excluding readout data, from any given channel, that are set to be the same for all chips of the hybrid, which means the large difference between “A” and “B” chips can have an effect. As part of a detector readout system, each chip would be given an independent threshold setting, and so the chip to chip variation would not be a problem. This is a case where the Htest is too stringent.

4.3.5 Raw Htest Data

While looking directly at the raw Htest data is not part of our usual performance evaluation procedure, it is worth noting that it has occasionally been a useful tool in diagnosing puzzling hardware or software behavior. For example, the problem described at the end of section 4.3.3 involving appending to ASCII data files, rather than replacing them as desired, was uncovered by studying the raw data. Furthermore, the raw data is kept on the PC that hosts the hybrid burn-in database (section 5), and is copied on a regular basis to a backup system, in case there is ever a reason to reprocess the data in any way.

4.3.6 Current and Voltage Plots

Another step in the quality assurance and quality control procedures is to examine the current and voltage behavior of each hybrid during burn-in. The associated tool for this is a set of plots, an example of which is shown in figure 22 for hybrid h0123. The page of plots includes a title bar with information about the name of the burn-in (so that it can be distinguished from others if the hybrid participated in multiple burn-ins), and the start and end times of the burn-in. The top plot shows the analog and digital currents as a function of time, with linear fits and corresponding slope parameters displayed. Our quality evaluation includes looking at these fits and verifying that the slope is zero within the statistical significance of the fit. The middle left plot is a histogram of the analog current measurements, providing an alternative view of the spread in values and a quick way to check for overflow or underflow outliers. The middle right plot is the equivalent histogram of the digital current measurements. Here, the structure is due to the different stages of operation of the SVX4 chip, as it cycles through acquire, digitize, and readout. The plot on the lower left is of the analog voltage measurements as a function of time and the plot on the lower right is of the hybrid ground voltage level as a function of time. These last two plots are checked for their overall level and stability.

5 Hybrid Über Tracker Database

Members of the UC Davis group wanted to avoid the proliferation of so-called paper “travelers” that were used during previous hybrid production projects, and collaborators on the CDF runIIb silicon upgrade agreed on the benefits of having a web accessible database containing all the information about hybrid testing and burn-in.

Currents and Voltages for Hybrid h0123

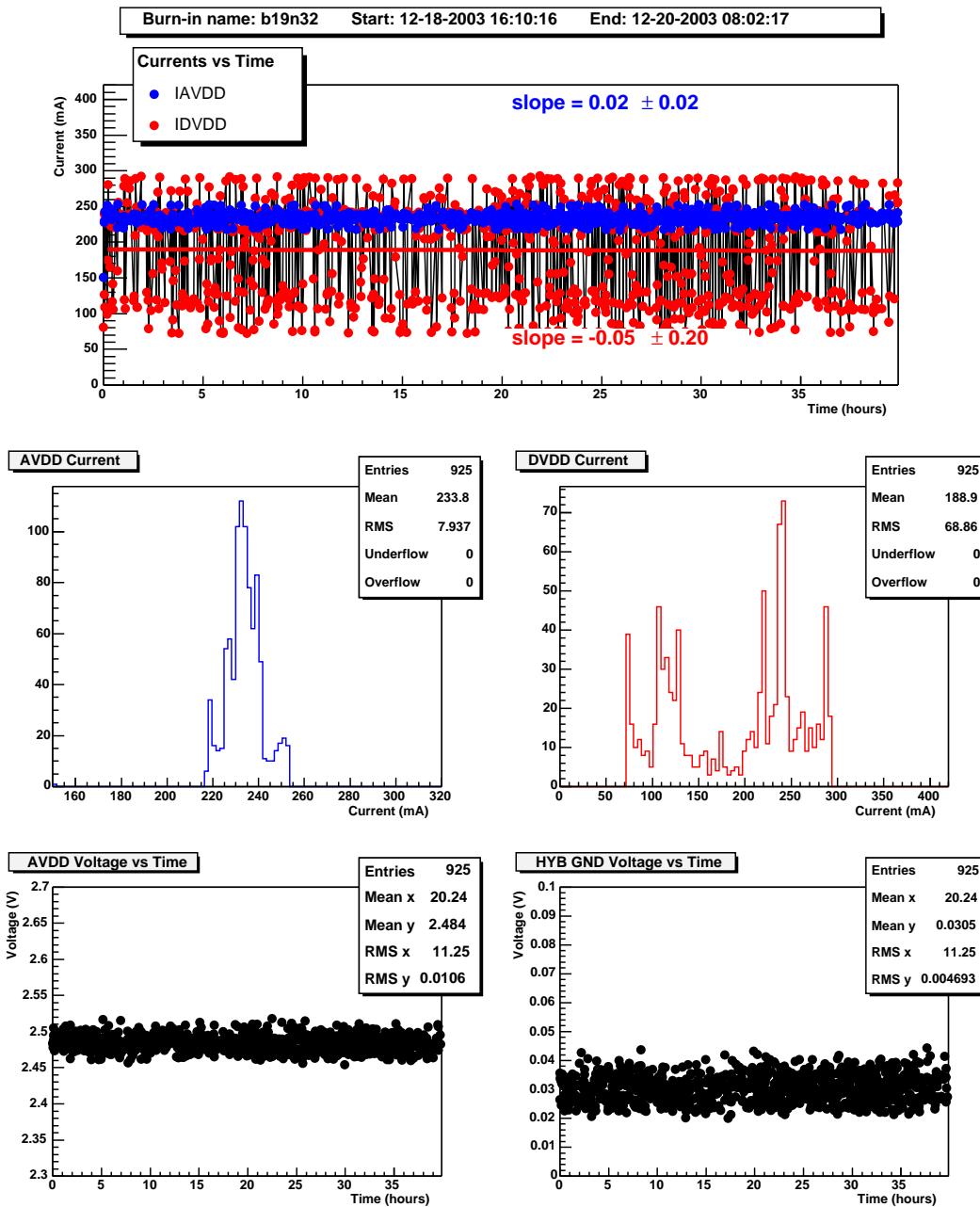


Figure 22: Current and voltage plots for hybrid h0123. Top: analog (blue) and digital (red) currents as a function of time, with linear fits and slope parameters. Middle-left: analog current. Middle-right: digital current. Lower-left: analog voltage as a function of time. Lower-right: hybrid ground voltage as a function of time.

UC Davis took on the challenge and created “RunIIb Über Tracker.” The database contains a wide range of information, starting with the components that are used in constructing the hybrids, including test results and identification information for the wafers of chips, the individual chips, and the substrates. There is also one section on prototype stave bus cables. Furthermore, the database contains details about hybrid assembly, mounting of components on the hybrids, hybrid initial testing results, and burn-in results. For these last two, the database contains links to plots and data files of the test results, useful for both performance evaluation and archival purposes. Throughout the database there is information relevant to tracking and logistics, such as the date on which a procedure is performed and transfer dates from site to site. It also holds information about repairs made to hybrids and some summarized production statistics. The contents of the database are detailed in the following sections, after a discussion of the implementation.

5.1 Environment

RunIIb Über Tracker database is distinct from any other databases in use for RunII. The database itself is physically located at UC Davis and is maintained by the UC Davis group. The components that make up RunIIb Über Tracker are the following:

- Apache Web Server 1.3.27
- MySQL Database 2.23.56-1
- PHP: Hypertext Preprocessor 4.3.2

All of these are open source packages and can be downloaded from the Internet, thus avoiding the costs of commercial software licenses for this project. Apache Web Server is the most commonly used web server, is well tested, and has security components. MySQL is a relational database management system that is fast, easy to use, and suitable for applications of any size. It stores data in separate smaller tables, rather than one big table, which adds flexibility. PHP (Hypertext Preprocessor) is a programming language that is embedded into HTML (HyperText Markup Language) to create dynamic web pages. A desirable feature of PHP is that it contains built-in functions for connecting to MySQL databases. PHP is widely used, and there are many tutorials and help documents on the Internet. A most useful reference during the development of this database was a recent book on combining these open-source tools [9].

The actual database part (MySQL) of RunIIb Über Tracker is made up of several tables, which are the constructs within the database that contain the actual information. All tables in the hybrid tracking database are linked by relations so it is easy to combine data from multiple tables with one request. For most stages of production (e.g. substrate assembly or hybrid testing) there are two or more corresponding tables in the database.

The following description shows how the three components combine to make RunIIb Über Tracker work. A RunIIb Über Tracker user requests a particular file

(the RunIIb Über Tracker login page, for example) by entering the URL (Uniform Resource Locator) in his or her browser or by clicking on a link that is already displayed. The Apache web server grabs the file and generates HTML based on the PHP code embedded in the file; the resulting HTML is sent to the user's browser and displayed.

5.2 Users

There are users at UC Davis, LBNL, and Fermilab who use RunIIb Über Tracker to keep track of work. All users of the database are given unique login names and passwords, and this information is stored in a MySQL table called **Accounts**. A MySQL database is equipped with built-in privilege control at the database, table, and table-column levels. RunIIb Über Tracker uses both database level (for selecting items to view) and table level (for inserting and updating items) privileges. All users, including a “guest” user, are allowed database level select privileges. This allows all users to view data from all tables. Collaborators at different institutions are allowed insert and update privileges only on tables that correspond to the work they do. For example, only UC Davis collaborators are given the ability to update information in the burn-in tables.

Besides the built-in privilege features of MySQL, we have implemented a web-page level privilege mechanism which displays some items in the web pages as active links for some users and as plain text to other users. There is a table called **Privs** which contains all users of the database and a corresponding privilege code for each of them. The **php** scripts access this table and then display the clickable links appropriately.

5.3 Data

In general, all data are entered into the database through the web page interface. (Exceptions have been made when large amounts of data needed to be entered at once. In those cases, simple text files were created, and a script read the text files and inserted the data into the database.) Data are inserted through text boxes, check boxes, and pull down menu options in the form of ASCII text or integers. After items have been entered into the database, they can also be updated in the same manner. The removal of data is very rare and not currently supported by the **php** scripts, although this can be done directly through the MySQL command line.

5.4 Usage

The RunIIb Über Tracker user utilizes any web browser and navigates to the URL <https://cdfw09.ucdavis.edu/Login.php>. At this page, the user enters his or her login name and password and submits the request. If the login name and password are valid, the main page is then presented in the browser window. This is comprised of a main menu on the left hand side of the page (see Figure 23), an action bar at the bottom (which may or may not display any options, depending on which user



Figure 23: The options available in the RunIIb Über Tracker main menu.

is accessing the page and which page is being accessed). The major section of the window is empty and will fill in once a menu item is selected

The left-hand menu items are self-explanatory and basically correspond to the production stages. The “Information” section of the main menu contains information about the hybrids for viewing only. No inserting, changing, or updating of information occurs here. The items in the “Production Stage” section contain pages for inserting and updating information and more detailed information than the summaries presented in the “Information” pages. Below is a brief description of each of these menu items.

5.4.1 Production Statistics

This section summarizes wafer processing by listing how many wafers have been probed and diced. Also visible is a listing of how many hybrids have been through each stage of production. The user can also select one particular hybrid and view its progress through the stages of production. As of the writing of this document, the Production Statistics page is still a work-in-progress.

5.4.2 Hybrid Information

This page lists all hybrids that have been entered into the database and provides information about the assembly of each hybrid (which chips are installed on the hybrids and what assembly company was used).

5.4.3 Hybrid Repairs

This page lists all hybrid repairs that have been made. Each repair is considered separate from other repairs, so there may be multiple entries for a particular hybrid if it has had multiple repairs. Most repairs fall under the categories of fixing a shorted power line or replacing a broken wire bond. If there was a short in the power lines, an entry under the Shorts column will be seen (AV-AG or DG-AG or DG-DV or AV-DV). If a bond was repaired, an entry in the Wire Bonds column will be seen. This entry is coded (H/C1/C2/C3/C4)-(L/R/B)-(#), which describes if the bond was associated with the hybrid in general or with a specific chip; if the bond was at the left, right, or bottom; and a number describing the position of the bond.

Cable	Delivery Date	Company	Pass/Fail	Clamp Mark Grade	HV Separation (mils)	Short/Open	Current Location	Shipping Date To FNAL	Comments
1	2003-07-14	Altaflex	Pass	B	25	none	FNAL	2003-07-21	
2	2003-07-14	Altaflex	not done	C	12	none	LBNL	not sent	Mika is using it for testing (marked "test cable")
3	2003-07-14	Altaflex	Pass	B	15	none	FNAL	2003-08-08	
4	2003-07-14	Altaflex	Pass	C	15	none	FNAL	2003-08-08	

Figure 24: Screen capture showing what information about stave bus cables is available in the RunIIb Über Tracker database. Note that throughout the database, table headings that are links can be used to sort the table, with the order determined by the elements of that column (both forward and reverse sorting are available, with multiple clicks).

5.4.4 Hybrid Summary

This is the main summary page for each hybrid. It allows the user to view summary details from each production stage. This includes information on assembly (which chips are on the hybrid, which substrate batch the beryllia came from, if there is a termination resistor chip bonded, etc.), testing (whether the hybrid passed visual inspection, power test, Svxscope and Htest; and LBNL quality tree and traveler), burn-in (summary of Htest results and readout errors, various plots generated during burn-in, UC Davis quality tree), and tracking dates. There is access to a printable version of this information which does not display the menu items on the left side and which includes plots in-line rather than as links. The hybrid summary section of the database is used extensively while evaluating how a hybrid performed during burn-in, as described in section 4.

5.4.5 Burn Summary

This page lists statistics for all the burn-in runs. Information included are the burn ID, number of hybrids burned-in, how many Htest passes and fails there were, how many readout errors there were, and general comments for the burn-in run.

5.4.6 Cables

This page lists testing results for the prototype stave bus cables. For LBNL users only, an “Add new Cable” link is visible in the action bar. Figure 24, a screen capture of a portion of the Cables page, shows what information is available in this table.

5.4.7 Wafer Probing

This page lists the wafers that have been entered into the database. A list including the chip version (A or B) and chip number are available for each wafer. There is also the option (using the “Alternate View of Chips on Wafer” link in the action bar) of viewing a diagram of the wafer with chip number and version labeled for each chip. This diagram was intended to display a 16-level color code result from

the wafer probing tests. For appropriate Fermilab users only, an “Add new Wafer to Database” link is visible in the action bar.

5.4.8 Substrate & Assembly

This page has information pertaining to the beryllia substrates used in hybrids. The number of substrates in each batch is listed. For each batch, information on each substrate is listed. This includes visual and power test results of the substrates, the current location of the substrate, the stuffing company, and the date the substrate was sent to be stuffed. For LBNL users only, links for adding and editing batch and substrate information are available in the action bar.

5.4.9 Chip Quality

This page is an alternative version of the Wafer Probing page. It displays a schematic picture of the wafer with the chip numbers and versions labeled. Each chip is colored according to its quality and the color code is provided. A chip is rated as good (if it has no bad capacitors), fair (if it has exactly one bad capacitor), or bad (if it has more than one bad capacitor, a bad channel, a bad cell, or fails one of the other Htest requirements). An example of this view of the chip quality information for a particular wafer in the Über Tracker database is shown in Figure 25.

5.4.10 Hybrid Stuffing

This page has information about the stuffing of the hybrid. Figure 26 shows a screen capture of what is displayed in a table that makes up a portion of this page. It keeps track of the substrate used in each hybrid, the date LBNL received it from stuffing, the visual and power test results after stuffing but before bonding, the bonding company, and the date it was sent to be bonded. It also displays the hybrid’s current location. For LBNL users only, an “Add New Stuffed Hybrid to Database” link is visible in the action bar. It is important to note that this is the step where actual hybrids, with hybrid ID numbers, are entered into the database. When a hybrid is entered into the database, it is entered into all necessary MySQL tables at once.

5.4.11 Hybrid Testing

This page displays information about the hybrid after it is bonded. It gives visual and power test results as well as the results of the Svxscope and Htest tests performed at LBNL. It also lists the date the hybrid was received at LBNL from bonding and to what location the hybrid is sent. Examples of this information can be seen in the screen capture of the table shown in Figure 27. On the web-site, by clicking on the hybrid ID, a user can view all test results up to this stage and repairs for a hybrid. In addition, from this sub-page, LBNL users can modify the test results and enter a repair for the hybrid.

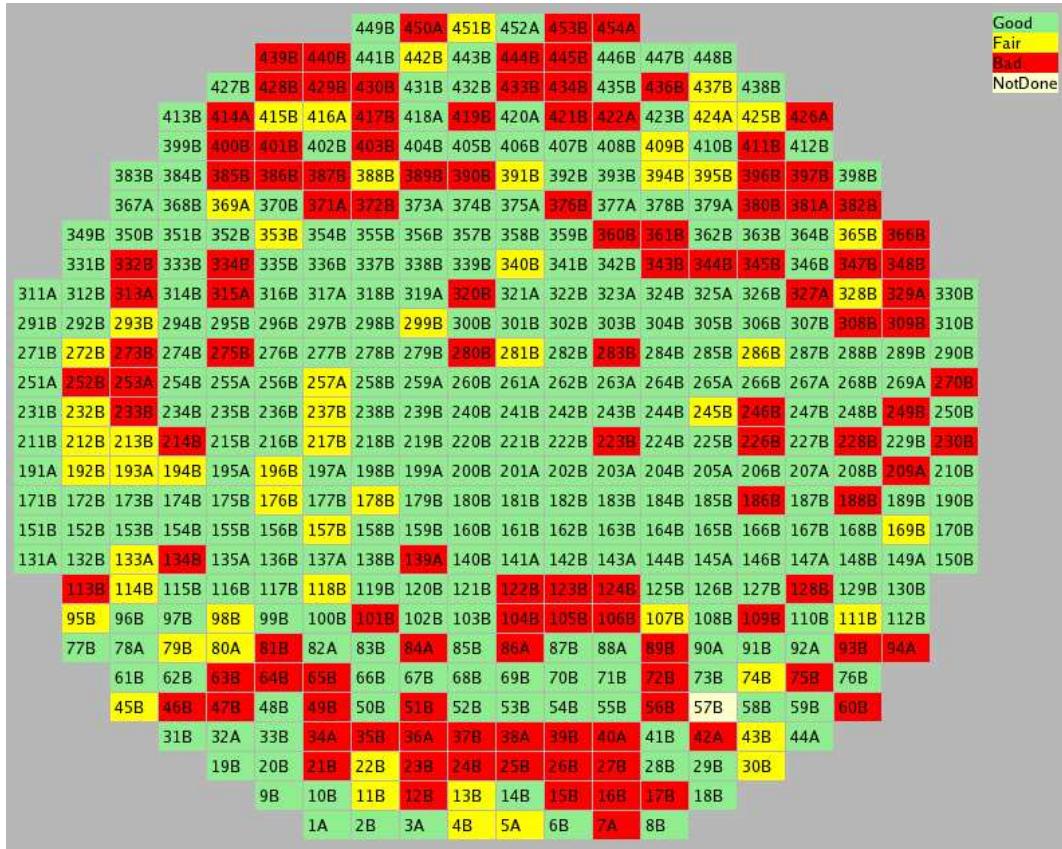


Figure 25: Screen capture showing the chip quality information for a particular wafer. This graphical representation includes the chip number, version (A or B), and a quality rating color code (explained in the text) for each chip.

Hybrid ID	Batch ID	Substr. ID	Date Rec'd from Stuffing	Visual	Power Test	Date Sent to Bonding	Bonding Company	Current Location	Comment
h0014	1	17	2003-09-26	pass	----	2003-09-26	Promex	FNAL	
h0015	1	18	2003-09-26	pass	----	2003-09-26	Promex	FNAL	
h0016	1	19	2003-09-26	pass	----	2003-09-26	Promex	FNAL	
h0017	1	20	2003-09-26	pass	----	2003-09-26	Promex	FNAL	
h0018	1	21	2003-09-26	pass	----	2003-09-26	Promex	FNAL	

Figure 26: Information available relevant to the hybrid component stuffing and bonding stages.

Hyb. ID	Bonding	Rec. From Bonding	Visual	Power Test	Svxscope	Htest	Encap.	Sent To	Current Location	Comment
h0216	done	2003-09-23	pass	pass	2003-09-24	pass	2003-09-24	2003-10-06	UCDavis 2003-09-25	FNAL
h0215	done	2003-09-24	pass	pass	2003-09-24	pass	2003-09-24	not done	not sent	UCDavis
h0214	done	2003-09-24	pass	pass	2003-09-29	pass	2003-09-25	not done	UCDavis 2003-09-25	FNAL
h0213	done	2003-09-24	pass	pass	2003-09-24	pass	2003-09-24	not done	not sent	UCDavis 2003-09-25 FNAL
h0212	done	2003-09-24	pass	pass	pass	pass	not done	not sent	not sent	UCDavis

Figure 27: Screen capture of a table that is included on the Hybrid Testing page.

Hyb. ID	Burn ID	Start Time	End Time	Htest1	Htest2	Hours Burned	Readout Errors	Burnin Port	Comment
h0051	b13n12	2003-11-14 13:21:00	2003-11-18 11:17:00	pass	pass	93.9	0/359	MPX 2 Chan1	add
h0052	b13n12	2003-11-14 13:21:00	2003-11-18 11:17:00	fail	fail	93.9	0/359	MPX 2 Chan5	add
h0053	b13n12	2003-11-14 13:21:00	2003-11-18 11:17:00	pass	pass	93.9	0/362	MPX 3 Chan1	add

Figure 28: This part of the Über Tracker burn-in page contains selection boxes for burn-in name, information type, and, in this case, specific information about the burn-in.

Hybrid ID	Burn ID	Arrived	Sent to FNAL	Current Location	Comment
h0017	b5n9	2003-10-03	2003-10-10	FNAL	add
h0030	b5n9	2003-10-03	2003-10-08	FNAL	add
h0031	b5n9	2003-10-03	2003-10-10	FNAL	add

Figure 29: As already shown in Figure 28, this part of the Über Tracker burn-in page contains selection boxes for burn-in name and information type, but, in the case shown here, the selection has been made to display the tracking information.

5.4.12 Hybrid Burn In

This page has the option of displaying two types of tables (using the “Info Type” pull-down menu). The “Burn Info” table (see Figure 28) lists all hybrids and displays which burn-in runs the hybrid was included in, the time that each hybrid was burned-in for that run, and where it was located on the burn-in stand. It also lists the results of the two Htests run during burn-in and the number of readout checks that hybrid failed compared to the total number performed. This information can be viewed for all hybrids or for a particular burn-in (using the “View Hybrid” pull-down menu). The “Tracking” table (see Figure 29) shows the arrival at UC Davis and sent to FNAL dates as well as the hybrid’s current location. Again, this table can show all hybrids or just hybrids for a particular burn-in. For UC Davis users only, links for adding burn-in IDs to the database and editing information are visible in the action bar.

5.4.13 Hybrid Repair Options

This menu link is visible only to LBNL users because they are thus far the only people who repair hybrids. This page gives the option of adding a repair, editing an existing repair, or deleting a repair (if an error had been made by entering the repair in the database). As noted earlier, other database users can read these repair

notes.

5.4.14 FNAL check off/Modules

This page displays all hybrids that are entered into the database with an arrival date at Fermilab, the quality of the hybrid when it is received, and whether or not the hybrid is used in a module. For most users, all items in the table appear as plain text. For the appropriate users, the items in the table are links for updating the information.

5.4.15 Change Your Password

This section of the main menu provides a way for users, with the exception of the guest user, to change their login passwords. No feature is available to change login names. This has to be done by the database administrators in MySQL.

5.4.16 LogOut

This link provides the user a method for logging out of the database and resetting all session variables. In addition, users are periodically logged out at 12:00 pm and 12:00 am Pacific time.

5.5 How to Add a New User

There are several steps that the administrator must complete to successfully add a new user. Each of the following steps consists of a number of commands issued through the MySQL command line before a new user can log-in and use RunIIb Über Tracker:

1. Enter the user into a pre-existing internal MySQL user table.
2. Grant all appropriate privileges - for the production stage tables and also for the **Accounts** table.
3. Make an entry for the user in the **Accounts** table.
4. Make an entry for the user in the **Prvs** table.

6 Results and Conclusions

6.1 Results

This section summarizes the results of the burn-in project. Table 1 contains the summary information pertaining to the 4-chip hybrids. The first section of the table contains numbers that have been compiled by our colleagues at LBNL and relate to the first part of production at LBNL and not directly to burn-in. There were 127 4-chip hybrids produced, but the first 10 are excluded from most of the accounting as they were used for a variety of special purposes, such as irradiation tests (at a

Table 1: Summary of burn-in results for 4-chip hybrids.

	Number of hybrids
Production	
Produced (not including first 10 used for studies)	117
Failed LBNL initial testing	9
Passed LBNL initial testing	108
Damaged during wire-bond encapsulation	5
Remaining good hybrids	103
Good hybrids kept at LBNL	2
Burn-in	
Good hybrids received at UC Davis for burn-in	101
Hybrids received for burn-in, from first batch of 10 produced	2
Failed LBNL initial testing but sent for burn-in anyway	1
Total burned-in	104
Total that failed due to burn-in	0

reactor) and single event upset tests (at a cyclotron beam). However, 2 of these first 10 were burned-in and are included in the lower portion of the table. Of the 117 hybrids included in the initial tests at LBNL, 108 passed and 9 failed. Of those 108 that passed, 5 were damaged during the procedure used to encapsulate the wire bonds. Of the 103 remaining good hybrids, 2 were kept at LBNL and 101 were sent to UC Davis for burn-in. As shown in the lower portion of Table 1, in addition to the 101 hybrids that were known to perform well, 2 hybrids of the initial batch of 10, and 1 hybrid that had failed initial testing at LBNL, were sent to UC Davis for burn-in. Therefore, the total number of 4-chip hybrids that have gone through burn-in is 104. None of these failed burn-in. This means that all 104 of these hybrids passed the requirements for matching the initial tests that were carried out elsewhere and for not degrading in performance before and after burn-in. This 100% pass rate is a testament to the robustness of the SVX4 chip and the hybrid, and also demonstrates the quality of the inspection, testing, and screening that takes place before the hybrids reach the burn-in stage.

The summary of the burn-in results for the layer 0 hybrids is shown in Table 2. There were 10 layer 0 hybrids received at UC Davis for burn-in. Of these, 9 passed the burn-in and were sent to FNAL, while 1 remains under investigation and has been sent back to LBNL for further examination. This last hybrid has exhibited behavior that may indicate damage during burn-in. In particular, between the Htest at the beginning of burn-in and the Htest at the end of burn-in, the pedestal dropped by about 5 ADC counts, the Gaussian fit to the noise distribution changed from about 1.4 to about 1.0, and the gain dropped by about 5 ADC/V, among other small changes.

Table 2: Summary of burn-in results for 2-chip (layer 0) hybrids.

	Number of hybrids
Burn-in	
Good layer 0 hybrids received at UC Davis for burn-in	10
Passed burn-in	9
Sent to FNAL	9
Sent back to LBNL for further study	1
Total burned-in	10
Total that failed due to burn-in (still under investigation)	≤ 1

6.2 Concluding Remarks

The majority of the time dedicated to the burn-in project came in the preparation stages, with modifications to existing hardware and software, as well as design, implementation, testing, and debugging of new hardware and software. The other primary aspects of the project were the establishment of the tools and procedures for quality assurance and quality control, and the design and implementation of an open source hybrid database.

The first burn-in started on September 9, 2003, and apart from special long-term studies, the last burn-in completed on January 19, 2004. The final capacity of the burn-in stand is 32 hybrids, twice per week, for a total of 64, which surpasses the original estimated load of 40 per week. The burn-in pass rate for 4-chip hybrids is 100% (104 out of 104) and the pass rate for layer 0 hybrids is $\geq 90\%$ (≥ 9 out of 10). As of February 2004, two hybrids remain at UC Davis on the burn-in stand for long-term studies.

A Data From Svxscope

The following shows an example data dump from Svxscope, and has been edited to include the output from just the first 28 channels of the first chip and the first 14 channels of the second chip. For each chip, the format includes a preliminary line with the chip ID (158–161), and the cell ID (1–46), followed by a series of pairs consisting of the channel ID and ADC value for each channel. The last two numbers in each line show the value of `bit9`, which, for the timing configuration we use, should be “1” when the channel number is read out and “0” when the ADC data is read out. Note that the first ten channels of each chip are charge injected in this example.

```
Event dump 1, printing 1032 words
bit#  bit1 bit2 | bit9[1] bit9[2]
 0: 158   1   1   0
 2: 0 [254]   1   0
 4: 1 [253]   1   0
 6: 2 [255]   1   0
 8: 3 [254]   1   0
10: 4 [254]   1   0
12: 5 [254]   1   0
14: 6 [253]   1   0
16: 7 [251]   1   0
18: 8 [255]   1   0
20: 9 [252]   1   0
22: 10 [143]  1   0
24: 11 [143]  1   0
26: 12 [142]  1   0
28: 13 [140]  1   0
30: 14 [139]  1   0
32: 15 [141]  1   0
34: 16 [143]  1   0
36: 17 [143]  1   0
38: 18 [143]  1   0
40: 19 [143]  1   0
42: 20 [144]  1   0
44: 21 [140]  1   0
46: 22 [141]  1   0
48: 23 [141]  1   0
50: 24 [144]  1   0
52: 25 [141]  1   0
54: 26 [142]  1   0
56: 27 [141]  1   0
:
:
258: 159   1   1   0
260: 0 [253]  1   0
262: 1 [253]  1   0
264: 2 [252]  1   0
266: 3 [250]  1   0
268: 4 [251]  1   0
270: 5 [251]  1   0
272: 6 [252]  1   0
274: 7 [252]  1   0
276: 8 [250]  1   0
278: 9 [252]  1   0
280: 10 [141] 1   0
282: 11 [142] 1   0
284: 12 [141] 1   0
286: 13 [141] 1   0
:
:
```

References

- [1] M. Aoki *et al.*, “The CDF RunIIb Silicon Detector”, CDF Note 6750, to appear in the proceedings of Frontier Detectors for Frontier Physics, Isola d’Elba, Italy, May 25-31, 2003.
- [2] CDF Collaboration, “The RunIIb Technical Design Report”, CDF Note 6261.
- [3] B. Krieger *et al.*, “SVX4: A New Deep Submicron Readout IC for the Tevatron Collider at Fermilab”, to appear in the proceedings of IEEE Nuclear Science Symposium 2003, Portland, Oregon, U.S.A., Oct. 19-25, 2003.
L. Christofek *et al.*, “SVX4 User’s Manual”, in preparation as a D0 Note.
- [4] Device drivers written by I. Volobouev are available at <http://www-cdf.lbl.gov/~igv/Drivers/>.
- [5] More information about comedi, the driver collection, and the device interface library are available at <http://www.comedi.org>.
- [6] Documentation to the original implementation of the Hybrid Monitor program was written by I. Volobouev, and can be found in a file named `svx_help.txt`, which is available at http://willamette.ucdavis.edu/~soha/hybmon/svx_help.txt and is also distributed as part of the Hybrid Monitor package.
- [7] J. Freeman, P. Lujan, M. Weber, “Htwish – A Program for Automated Testing and Analysis of SVX4 Chips and Hybrids”, CDF Note 6851.
Draft documentation by J. Freeman on the Htwish test suit (Htest) is available at http://willamette.ucdavis.edu/~soha/hybmon/Htwish_manual.txt.
- [8] For details about the BerkeleyDB data files, see documentation by I. Volobouev at http://willamette.ucdavis.edu/~soha/hybmon/database_structure.txt.
- [9] L. James, B. Ware, *Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP*, Addison-Wesley Publishing Company (2003).