Operation and Test of Hybridized Silicon p-i-n Arrays Using Open-source Array Control Hardware and Software

Andrew C. Moore,[a,b] Zoran Ninkov,[a] Greg S. Burley,[c] William J. Forrest,[b]
Craig W. Mc Murtry,[b] and Lars E. Avery[a]

[a]Rochester Institute of Technology, Rochester, NY, USA
[b]University of Rochester, Rochester, NY, USA
[c]Observatories of the Carnegie Institute of Washington, Pasadena, CA, USA

ABSTRACT

A system for controlling and testing high-resolution non-destructive astronomical imagers was constructed using open-source components, both hardware and software. The open-source electronics design, originated by Carnegie Observatories (OCIW) for CCD cameras, was modified, assembled, and augmented with new circuitry which facilitates monitoring of voltages and currents. The electronics was run from Python user interface software based on a design from the University of Rochester. This new software utilized the Numarray and pyFITS modules developed at the Space Telescope Science Institute (STScI). Interfacing to the "dv" FITS image analysis package from the NASA IRTF was also implemented. Python (the STScI language of choice) was used as the primary language for systems integration, scripts for data acquisition, and scripts for data analysis. The DSP clocking software was a mixture of C and Motorola 56303 assembly. An interrupt-driven kernel-mode PCI device driver for Red Hat Linux was written in C, and used the PC processor and memory for image processing and acquisition. Two $1K \times 1K$ Raytheon SB226-based hybridized silicon p-i-n arrays were operated and tested with the new system at temperatures as low as 10K. Signal path gain, node capacitance, well depth, dark current, and MTF measurements were made and are presented here.

Keywords: Astronomical Imagers, CMOS Multiplexers, p-i-n array, Array Control, Silicon photodetectors, interpixel capacitance

1. INTRODUCTION

The Laboratory for Astronomical Instrumentation, Research, and Education (LAIRE) at Rochester Institute of Technology, led by Zoran Ninkov, and the Infrared Astronomy Group at the University of Rochester, led by William Forrest and Judith Pipher, have performed substantial research in the area of astronomical imager technology. In a joint effort by both institutions, new research and development in the systems that aid this research has been done, building upon components offered by other institutions, most notably Carnegie Observatories (OCIW) and the Space Telescope Science Institute (STScI.) An overview of this system, called "Pysys," is presented here along with some results obtained with it testing silicon p-i-n arrays.

1.1. The Challenge of Developing High Performance Imagers

High performance imagers are essential in many astronomical, medical, and remote sensing applications. Challenges in scientific detector development are reducing read noise and dark current and increasing quantum efficiency while maintaining good MTF. Development of improved devices is an iterative process; characterization of the latest devices is compared with expected performance, then adjustments are made in the next devices. The array control system plays a critical part in closing this development cycle.

Further author information:
E-mail: andrew.moore@rochester.edu

1.2. Systems to Operate Non-Destructive Imagers

High performance imagers require precise low-noise control to operate and characterize the devices in a laboratory. A simple to understand, flexible, and easy to use system is beneficial. A compact control system facilitates imager operation in a non-laboratory environment, such as a telescope. Since replication and/or modification of the system may be required, it is helpful if the design details are completely open and there are few licensing issues or proprietary modules in the system. A low-cost control system is of great benefit for researchers that have limited budgets or a need for high volume testing, and can also serve as a permanent system in an application environment. Pysys provides low-noise clocks and biases controllable to 2 millivolt resolution over a ±8.192 volt range. Its software is mostly C and Python. The electronics enclosure is 10"×5"×4", including the power supply. The Pysys electronics was built at a total component cost of under $3000.*

Low noise acquisition from the non-destructive readouts considered in this paper requires multiple reads of the detectors, with signal processing of these reads to produce the final image. Typically, a difference ("Correlated Double Sampling" or CDS) or a difference of averages ("Fowler sampling") is performed.[1,2] Uncertainty in the reset levels requires pre-integration, or "pedestal" reads to measure these reset levels accurately. After integration, "signal" reads reveal what has changed since reset. The simplest signal processing, CDS, subtracts the pedestal from the signal. Electronic read noise is often significant (and non-white) and the extra reads in Fowler-sampling reduce its effect. Least squares curve fitting, or "line fitting", theoretically best to use when read noise is dominant, is sometimes employed. Line-fitting's advantage over optimal Fowler-sampling is small[3]; the role it plays in cosmic ray rejection algorithms[4] is more significant.

A system to reset the array, read the pixels before, during, and after integration, process the multiple reads into images, and store and analyze the data is required to do the testing. Pysys provides a high degree of control over the voltages and waveforms applied to the imager, and was specifically tailored to control and test non-destructive source follower per detector (SFD) arrays, such as the hybridized focal plane arrays produced by Raytheon Vision Systems. Pysys digitizes four 400 millivolt range analog outputs to 16 bit resolution, with data rates near half a megapixel per second. These specifications were chosen as appropriate for controlling and testing long-integration time high-resolution imagers for use in low-light applications such as ground and space based astronomy. Faster rates, in the neighborhood of five megapixels per second should be possible with modification.

2. BACKGROUND

2.1. Overview of Previous Systems

Pysys is a new hybrid of previously existing technologies. The electronics package was originally designed (by Greg Burley) for OCIW to operate a CCD mosaic. OCIW offers the design for free to any who are interested.[5] In Rochester it was found to be suitable, as it was very similar to a completely custom approach proposed here, and was employed in Pysys with some modifications. (The design details of Pysys will be made available from OCIW as well.) The electronics of Pysys are described in detail later in this paper.

The Pysys software architecture and user interface was modeled after a system developed at the University of Rochester's Near Infrared Astronomy (NIR) lab. Some background on the NIR lab system is presented here, as it is relevent to the development of Pysys. NIR lab's system for array testing consisted of a "silver box" (suite of custom circuit boards for analog control and signal conditioning) and a Linux host PC (containing several ISA cards for digital control and data acqisition.) The "silver box" provided clock and bias voltages to the array and amplified the video signal back. Three of the ISA cards in the host PC were DSP cards, one for control and a pair for data acquisition and signal processing. Two additional custom boards digitized the analog outputs to 16 bit resolution and low noise. A custom software system written in Forth, called "dspsys", allowed the operator to control the system and acquire images with a simple command-line interface. Various peripheral items to monitor and control temperature and illumination completed the system. The dspsys/silver box system was instrumental in much IR detector research, notably the development and test of InSb detectors for SIRTF.[6,7,8,9,10] Its noise performance has proven difficult to match.

*excluding the cost of the Analogic A/D converters. A four channel video board using lower cost converters is in devlopment.

The NIR lab silver box system had several areas where improvement was desired. Voltages were set manually with 20-turn potentiometers. The DSP boards for data acquisition had insufficient memory to store an entire image of modern arrays; a pair of data boards were limited to 512×512 images. In addition, much of the system is obsolete and cannot be purchased today. Measuring most voltages and currents was a manual procedure. Since the host computer contained the clocking and acquisition cards, it was large and could not be located a long distance from the silver box. As a result, the computer was bolted to the telescope and run remotely over ethernet via a second computer in some circumstances. This was somewhat unwieldy, mostly given the size of the host computer.

The architecture and execution of "dspsys" was exemplary. Its powerful collection of well-designed modules and terse command line interface affords the user easy manual and scriptable control of the electronics. Forth, however, (although elegant and powerful in many ways) is generally considered obsolete, and the elegance of dspsys is invisible to most programmers, who are not inclined to learn it. The clocking and data acquisition programs, written purely in Motorola 56000 assembly, also challenge the typical programmer.

Pysys was designed with a goal to increase capability beyond NIR lab's silver box system while reducing system complexity. The increased capability would benefit the user, and the reduced complexity would benefit the engineer and programmer responsible for maintaining and enhancing the system.

2.2. Overview of Pysys

With reduced complexity in mind, Pysys was designed with only one processor (in addition to the host computer.) This processor, a Motorola DSP56303, controls the electronics box and handles both writing the clocking patterns and reading the A/D converters. The user interface software, "pydsp," was designed as a Python clone of NIR lab's dspsys. Python,[11] an open-source interpreted object-oriented scripting language developed by Guido Van Rossum, has proved to be a superb replacement language for Forth. Often perceived as a "scripting" language, Python operates well here as a "systems glue" language, suited for a wide variety of tasks from hardware interfacing to user interfacing.

Development of Pysys was gradual. The PCI card and device driver were integrated into University of Rochester's NIR lab setup. This expanded the capabilities of their "silver box" electronics system from 512×512 images to $2K \times 2K$, easily expandable further with more memory in the host computer. In addition, the individual images acquired during "Fowler-sampling" or "sampling up the ramp" were made available. Access speed to the image data improved as well. Simultaneously at RIT, a new system was coming together, with the modified OCIW electronics taking the place of the University of Rochester "silver box." This system was operated for several months using "dspsys" for the user interface software, but "pydsp" replaced it when it was capable enough.

3. SYSTEM DESCRIPTION

3.1. Pysys Electronics Overview

The main components of the Pysys electronics are shown on the left side of Figure 1. A boxed backplane of four cards, arranged from noisiest (power supply, then DSP) to most noise sensitive (clocking card, then digitizer) attaches via a "header board" to the dewar and device, and via a high-speed serial communication link to a PCI card in the host computer. The DSP generates the clocking sequence patterns. The clocking card receives the sequence patterns, and level-shifts the digital pattern to the voltages required by the array. The array outputs are digitized by the video board and sent back to the host PC.

3.2. Current Sense Header board

The header board, a small $4.25" \times 3.5"$ circuit board, fits between the backplane and one end of the electronics box. It performs interfacing between the array control electronics and the device, providing "RC" filtering for 16 clocks and 16 biases, and preamplification for the returning video signals. Four array outputs are amplified; the board used here was configured with a gain of 25 and a programmable ± 1.2 volt offset. The video signals are then passed to the video card. Additional diagnostic circuitry was also added to sense the voltages and currents in the clock and bias lines and buffer those signals to "monitor" outputs without incurring risk of static

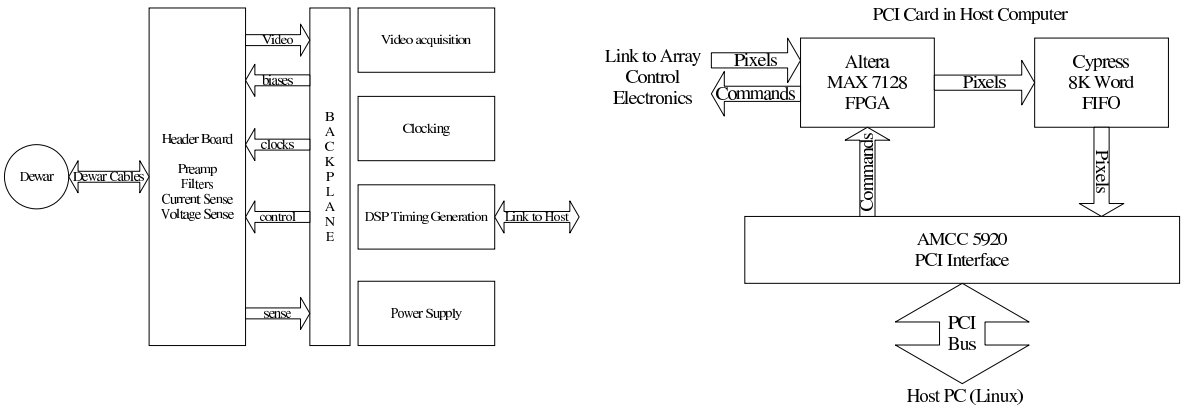

Figure 1. The Pysys electronics setup: The camera array, in the dewar, connected to a header board and ultimately to a filled four-slot backplane. This interfaced via a high speed serial link to a simple PCI interface. The PCI card had no processor and only 8K words of FIFO memory.

discharge damage to the device. Analog multiplexers allow 16 signals to share the same monitor output. The voltage monitor is unity gain. The current monitor senses the drop across the resistor in the RC filter with an instrumentation amp. The header board used here was implemented such that the sensed current is presented with a conversion factor of 10 millivolts per microamp. OCIW provides design details for a header board and preamp tailored to their dewar and CCD mosaic. The header board just described, more appropriate for the Rochester detectors and laboratory environments, was designed and constructed from scratch.

3.3. DSP Card and Clock Program

The DSP56303 processor (DSP) can be run at a software programmable clock rate. 100 MIPS is the maximum speed of the DSP; the programs used here were run at 50 MIPS. A field programmable gate array (FPGA) on the DSP card latches 20-bit clocking patterns. Special addressing modes to simplify the clocking programs were added to the sequence logic, allowing bits to be set, cleared, and toggled independently. A timer module in the DSP was used for integration time control, simplifying timing software and providing millisecond resolution with high stability. C or assembly can be used for clocking programs; the clock program used here was written in C with inlined assembly language. The C compiler (for the Motorola 56300 family) was obtained free from Motorola; wine (a Windows emulation package) allows the (Windows) C compiler to run on the Linux host PC.

3.4. Clocking and Video Cards

The clocking card accepts 16 bits of the sequence pattern, level shifts these bits, and outputs the control signals to the array. Four octal 13 bit D/A converters provide 32 programmable voltage levels for clock rails. Some voltages and sequence bits are shared among outputs; the clocking card provides 20 clock outputs from 16 sequence bits and 32 voltages. Three of the four remaining sequence bits go to the video card, one bit for conversion timing and two for analog switching. The OCIW two-channel video card design supports correlated double sampling circuitry for CCD control. Since the SB226 multiplexer has 4 outputs and cannot use CCD-style correlated double sampling, the CCD CDS circuitry was modified to multiplex a pair of video inputs into each of the two 16 bit Analogic A/D converters using one of the bits. Oscilloscope synchronization was provided with one bit, and a shutter was operated with the remaining bit. Digitization at 16 bits with a range at the converter of ± 5 volts and a header board gain of 25 gave 6.1 microvolt resolution of the array's output voltage. The video card design also provides 8 programmable biases, but a layout error on the (OCIW prototype) boards used hindered their implementation; unused clock lines were used for biases instead.

3.5. PCI Card and Device Driver

The PCI card in the host PC, shown on the right side of Figure 1, is very simple. An AMCC S5920 PCI interface chip, an Altera Max 7128 FPGA, and a Cypress CY4255 8K by 18 bit FIFO compose the core of the card.

Low Voltage Differential Signaling (LVDS) was used for the communication link. A fiber-optic communication link is also avaliable in the design, but was not implemented in Pysys. The small size of the FIFO has proved sufficient at data rates of 400K pixels/sec. FIFO overflow has never been observed at these rates, and no special precautions have been taken regarding what else is running on the host computer. Operation at 2M pixels/sec was reliable virtually all of the time, but FIFO overflows were noticed on rare occasions. No investigation into the source of these overflows was done since that data rate was not required.

OCIW's original PCI hardware only allowed for an interrupt when there was a single pixel in the FIFO. Reading the FIFO would thus entail reading a single pixel, checking the FIFO status, and looping until no more pixels remained. Although this approach would maximize allowable latency, it would leave little time for other processes in the host; the processor would be re-interrupted almost immediately after exiting the ISR and spend excessive time context switching while pixels were arriving. Checking FIFO status every pixel limits the maximum data rate that pixels can be copied from the PCI card. The FIFO provides a signal it can assert when it is filled to some level, with a default level of half full, and a modification to the board allowed this line to serve as the interrupt source. This change increased the amount of time the processor was allowed to execute other processes as the FIFO filled, and eliminated the need to check FIFO status for every pixel, but allowed only 4K pixels (10 millisec interrupt latency) before overflow.

Two special hardware features of the PCI chip were also employed, allowing faster access to the FIFO. The serial eeprom for the PCI chip needed a programmimg change to enable these special features. One feature was the ability to "prefetch" a block of data. A specially defined "region" in the PCI chip, when read by the host, tells the PCI chip to assume that sequential access of the entire block will be performed. The interface chip then fetches as many additional reads as it can hold in a small FIFO of its own. If a DMA operation or higher priority interrupt blocks the FIFO interrupt service routine (ISR), the PCI chip will continue to unload the FIFO. A 4K word read region was defined with all read accesses mapped to the FIFO. When 4K pixels are known to be in the FIFO, they can be read from this region. The PCI chip also provides bus width translation. In this mode, the PCI chip accesses the FIFO in two 16 bit reads and presents the same data on the PCI side as a single 32 bit read. This mode was employed as well, cutting the number of PCI reads in half.

The left side of Figure 2 illustrates the software architecture of the device driver used. An excellent reference for Linux device drivers by Rubini and Corbet[12] was extremely helpful for its development. The device driver has two tasks to accomplish at high priority: emptying the small FIFO into a large circular buffer, and performing signal processing on pixels in the circular buffer before it wraps around. The "bottom half" mechanism described by Rubini and Corbet was appropriate here. An ISR should do the minimum required of it, typically service hardware, and exit. In this case, the minimum requirement was to empty the FIFO and wake up another process (the bottom half) that processes this data at high but interruptible priority. Access to both the circular buffer and the processed data is provided, and was implemented using memory mapping.

After starting an acquisition, the user process polls the driver for how many pixels remain. If a large number remain, the user process is put to sleep for a short time and then informed the number of pixels remaining. When the number remaining falls below 4K, (the number required to create an interrupt,) the ISR disables itself and the query routine called by the user empties the last pixels from the FIFO directly. The query routine also wakes up the bottom half when the last pixel has been emptied and placed in the circular buffer. Since the bottom half runs at high priority, the user process is guaranteed that the bottom half has executed when it receives confirmation that all pixels are in. An error is returned if the FIFO is not empty when the last expected pixel has been read.

3.6. Suggestions for Increased Data Rates

Some of the enancements just described were made in anticipation of higher data rates. Since the PCI bus is capable of 66 megapixels per second, it is expected that further modifications to the system could provide 5 - 10 megapixels per second. For those interested, suggestions are offered here: A 64K word pin-compatible FIFO is available from Cypress that would increase the PCI card's latency tolerance. The PCI chip operated from a 10 MHz master oscillator in Pysys, but its data sheet indicates that it can be run faster. A revised PCI card with a pair of FIFOs side by side could present pixels to the PCI chip without bus width translation overhead. Some motherboards do not support PCI burst reads, and care should be taken to avoid these motherboards. Special

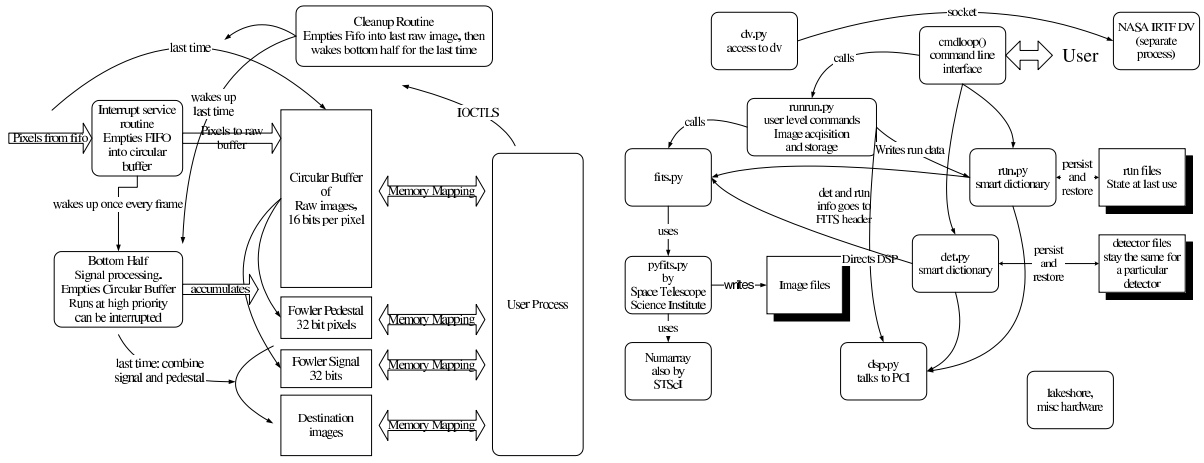

Figure 2. Left: The kernel-mode PCI device driver empties the FIFO at interrupt priority into a large circular buffer. An "immediate priority" task, which preempts any user process, performs signal processing on the pixels in the circular buffer. If the circular buffer does not wrap, all intermediate reads are available to the user process. Right: A simple collection of software modules called "pydsp" brings the power of Pysys to the user.

operating system calls to access the PCI card in burst mode may be required as well. The data rate of the serial link would also have to be increased above the 50 megabits per second used here. (LVDS can operate at 400 megabits per second.)

3.7. "Pydsp" Host Computer Python Software

The right side of Figure 2 shows the main components in the architecture of pydsp, the Python based user interface software. Python, like Forth, has its own command line interpreter, but a nice feature of the Forth interpreter is that a single word at the command line can invoke execution of code. In Python, a code object needs to be followed by a pair of parentheses to signify that execution is intended, otherwise a string describing the object is printed. A simple Python command loop was implemented to allow simple words to execute code. This loop prompts the user and then interprets and executes the user's input. It was written to emulate commonly used dspsys commands; new ones were added as well. A "break" command exits the loop back to the Python command prompt. Typing "cmdloop()" in Python brings the command loop back up again.

In both dspsys and pydsp, FITS images produced by the system are stored in an "object" subdirectory of a "night" subdirectory of a main FITS data directory. At a telescope, each observing session is stored in its own "night" and as the telescope moves from object to object, new directories are created for each object. Images of the same object are stored together in object directories. In the laboratory, the night directory can be changed for each new detector, and within that directory, object directories store the images from the various tests that are run. Two commands, "night" and "object", allow the user to create new night and object directories. Commands to take an image and place it in the current object directory were also provided. In astronomy, images of interest often have a "background" image subtracted from them to remove system artifacts, so commands were made to distinguish a "source" image from a "background" image. Either can be set as the default. A distinction is made between a "scan" which takes a temporary image and a "run" which allocates a unique new file name that is not to be overwritten. Commands to add comments to the FITS header were also implemented.

At system startup, an initializing state of the system is searched for. A variable from the OS environment directs pydsp to where a file entitled "lastrun.run" may be. The state of the system can be persisted to this file (or another file) with a "savesetup" command. The "lastrun.run" file directs pydsp to several other files and directories. Given a detector name from "lastrun.run", pydsp looks for a matching "*detname*.map" file that maps names of clock rails and biases to their appropriate DAC numbers. After this file is found and loaded, a "*detname*.bias" file maps these names to their proper voltages.

The names of the bias and clock voltages are added to the pydsp vocabulary when these detector files are loaded. Once in the vocabulary, entering the name of the voltage returns its present value in millivolts and entering a number with the name causes the voltage to be set to that value. Voltages are saved in the FITS image headers, keyed by their name. These words, and several others, are keys in a python "smart dictionary" with overloaded "get" and "set" operations. If setting or getting the value involves an extra operation (such as communication with the DSP) the additional code is invoked automatically by the dictionary access. Acquisition scripts using this mechanism are easily written in Python, and executed using Python's "execfile" capability.

3.8. Numarray, pyFITS, and DV

The Space Telescope Science Institute (STScI) has developed several modules for Python, and offers them without cost.[13,14,15] Their "pyFITS" module was used by pydsp to save image data in the standard FITSimage format. The "DV" FITS image analysis package[16] was used to view and analyze images. Socket-based communication from pydsp triggers DV to automatically load images after acquisition. Python analysis scripts using pyFITS and Numarray were written and used for processing images generated from other Python acquisition scripts for most of the results presented here.

4. SILICON P-I-N ARRAYS

Pysys was used to operate and test a pair of hybridized silicon p-i-n arrays produced by Raytheon Vision Systems under subcontract to RIT. Silicon p-i-n (or "pin") arrays are a promising detector technology. They offer low dark current, very sharp imaging with low detector crosstalk, nodes with very low capacitance, and good response in both the blue and near IR spectral regions. Blue response is enhanced since surface recombination is less of an issue. Near IR response is enhanced because the detector can be made quite thick without loss of MTF to diffusion. Most importantly, multiplexed arrays like these pin arrays are radiation-hard, a critical consideration for space applications and one which plagues CCDs in space. Raytheon's SB226 multiplexers operate quite nicely at temperatures around 35 degrees K. The pin arrays tested here may be the very first visible imagers to operate below pumped liquid nitrogen temperatures.

Pin arrays have a wide biased intrinsic region, unlike the depletion region of a typical p-n junction. This has two advantages. First, released photocarriers are actively swept into detector nodes. MTF degradation due to diffusion is substantially reduced. Second, the capacitance of the detector junction is a negligible part of the total detector node capacitance of a hybridized device. This is due to the large distance between the p and n regions. A small node capacitance is important in an imager since it results in increased detector sensitivity.

The detectors tested here are 185 μm thick wafers of high purity silicon, N doped on the illuminated side and P doped on the bonding side. The N doping is one big thin implant, conductive but transparent; this side was biased to a large positive voltage. Each pixel is a separate P implant; the pixels have a 27 micron pitch. The pin detectors were "bump-bonded" by Raytheon to two of their best SB226 1024 $\times$ 1024 cryogenic CMOS multiplexers.

Zoran Ninkov proposed the NASA-funded SB226/pin research initiative. A team led by Dr. Paul Hickson at the University of British Columbia, after hearing of Zoran Ninkov's planned Si pin array prototypes, proposed the use of silicon pin arrays similar to the ones tested here as visible imagers on the James Webb Space Telescope (JWST, previously NGST). The University of British Columbia and the Canadian Space Agency supported this concept.[17,18,19,20] Unfortunately, this concept is not included in the current plans for JWST. Testing of these pin arrays was done at 35K to see how such an imager might perform at the planned JWST focal plane temperature.

5. THEORETICAL MTF OF PIN DEVICES

As shown in Figure 3, three components create the overall point spread function of the pin device: transverse diffusion in the detector bulk, capture by square pixels, and inter-pixel capacitive effects in the multiplexer.

5.1. Transverse Diffusion

Virtually all of the bulk region of pin detectors is intrinsic with a powerful electric field in it when biased to full depletion. The field strength used was 33.6 volts over 185 microns, an average of 1800 volts per centimeter. At this field strength the hole drift velocity in silicon is limited by the themal velocity. Ottaviani *et al*[21] measured $v_{drift} = 6 \times 10^6$cm/s at 35K, which predicts a hole lifetime of $\tau = 0.0185\text{cm}/6 \times 10^6(\text{cm/s}) = 3 \times 10^{-9}$s before the hole reaches a detector node. The diffusion length can be found from $L = \sqrt{D_p \tau}$. Under high fields, the diffusion coefficient D_p splits into two components. $D_{//}$ denotes the "longitudinal" component, (along the field) and $D_\perp$ denotes the "transverse" component (at right angles to the field). $D_\perp$ is the component of interest for modeling point spread, but data on $D_{//}$, which is important in high-speed circuitry, seems to be more abundant. Canali *et al*[22] cite Persky and Bartelink, who measured the longitudinal diffusion coefficient of holes in high fields. Their data indicates $D_{//} = 20 - 30\text{cm}^2/s$ at 1800V/cm and 77K. This value increases with the increased mobility at lower temperatures. Transverse diffusion of holes $D_\perp$ in silicon appears to be equal to $D_{//}$ at moderately high fields, and is higher than $D_{//}$ only under very high fields. $D_\perp = 40 - 60\text{cm}^2/s$ is an estimate for 35K, assuming an inverse relationship with temperature. This model predicts a lateral diffusion of 3.4 - 4.2 microns, with a gaussian spread.† Since the field strength varies in the intrinsic region, and the drift and diffusion are both dependent upon the field strength, a more accurate equation which can be used with more detailed transverse diffusion data is:

$$L^2 = \int_0^{0.0185} \frac{D_\perp\left(E\left(x\right)\right)}{v_{drift}\left(E\left(x\right)\right)} dx. \tag{1}$$

The gaussian definition preferred by Gaskill[23] and Bracewell[24] was used in the model:

$$gaus\left(x, y; b\right) = \frac{1}{b^2} \exp\left(-\pi \frac{x^2 + y^2}{b^2}\right) \tag{2}$$

The scaling parameter $b > 0$ changes the spread of the gaussian.

5.2. Collection by pixels

An ideal square pixel is modeled with the "*rect*" function; the 2D pixel is the product of a *rect* in x and a *rect* in y. The *rect*, as defined by Bracewell and Gaskill, is:

$$rect(x) = \begin{cases} 0, |x| > \frac{1}{2} \\ \frac{1}{2}, |x| = \frac{1}{2} \\ 1, |x| < \frac{1}{2} \end{cases} \tag{3}$$

The model used the pixel spacing as the base dimensional unit, so the *rect* did not need a scaling parameter. The fact that the pin array pixels have a gap between them was not accounted for in the model. Given the powerful field near the P implants, there is probably no need to include it.

†Typical detectors; ones that have unbiased bulk and rely on diffusion, are more properly modeled with a spread that exponentially decays, until carrier lifetime becomes significant. The gaussian drops off much more rapidly than the exponential. Thus, nearby detector nodes are theoretically much less likely to collect charge generated by cosmic events in p-i-n detectors.

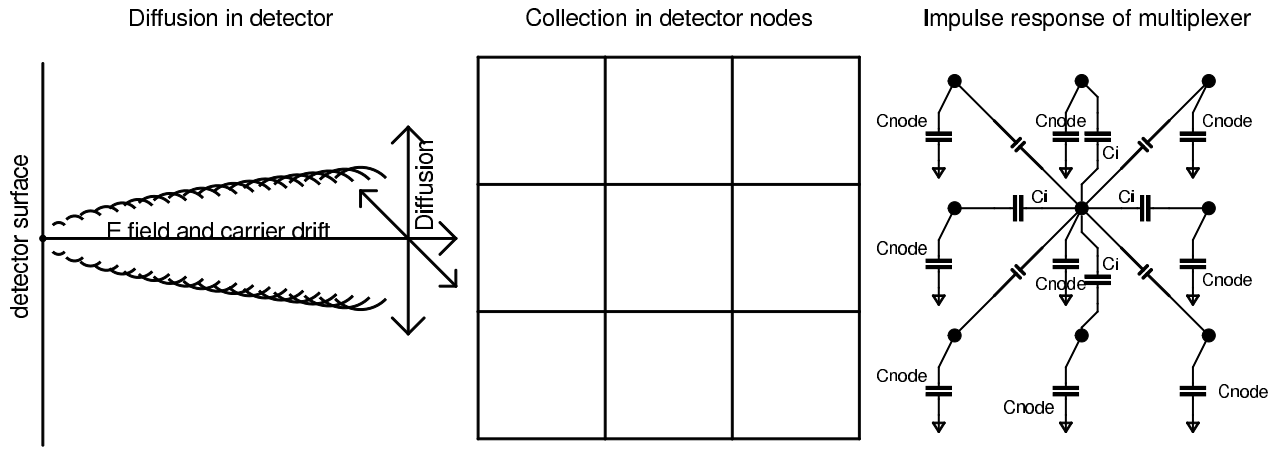

Figure 3. Overall point spread function of the device is from three effects in sequence: diffusion in the detector - gaussian for p-i-n arrays, collection by square pixels, and interpixel capacitance in the multiplexer.

5.3. Inter-pixel Capacitive Effects

The detector nodes are not electrically independent; a coupling capacitance exists between adjacent pixels as shown on the right side of Figure 3. Thus, the multiplexer also has an impulse response. If the multiplexer impulse response is assumed to be an even function with non-negative coefficients, it can be obtained from the autocorrelation of a white distribution of charge in the detector nodes.[25]

The impulse response of the multiplexer was measured by performing spatial autocorrelation of the poissonian noise in 1800 multiply sampled difference patches, 50 by 50 pixels in size. The 2D Fourier Transform of the autocorrelation was taken, then each term in the transform was replaced by its square root. Taking the inverse transform yielded an estimate of the multiplexer's impulse response.

The central value was estimated at 0.871, the four nearest neighbors were 0.027, and the four diagonal neighbors were 0.00525.

6. MODEL OF EDGE SPREAD FUNCTION

A common MTF analysis uses the Edge Spread Function (ESF.) The ESF is the integral of the line spread function (LSF.) The LSF is the convolution of a line impulse with the point spread function. For circularly symmetrical point spread, the process of deriving LSF from PSF is called the Abel transform.[26] Separability of the *gaus* makes the Abel transform trivial here; gaussian line spread is expected.

Only vertical and horizontal edges were used, thus the expected ESF was the convolution of the multiplexer's 1D response, the 1D *rect*, and an integrated gaussian:

$$esf_{pixel}(x;b) = (0.0375rect(x+1) + 0.925rect(x) + 0.0375rect(x-1)) * \int_0^x gaus(x';b)dx'. \quad (4)$$

7. TESTING OF PIN ARRAY MTF

A small "bullseye" patterned reticle, the center of which is shown on the left side of Figure 4, was gently laid pattern side down on the detector surface. A "Bessel V" filter (visible, $\lambda = 550$nm) removed the deeply pentrating red wavelengths, so that carriers would be released near the surface of the detector. Illumination was limited by a 200 micron pinhole approximately 1.8 inches from the array surface. The outer edge of the last (fifth) ring of the bullseye target was used to provide the edge, as it was nearly straight compared to the dimensions of a pixel. Only pixels that were a small angular displacement from horizontal and vertical were used. Each pixel's intensity

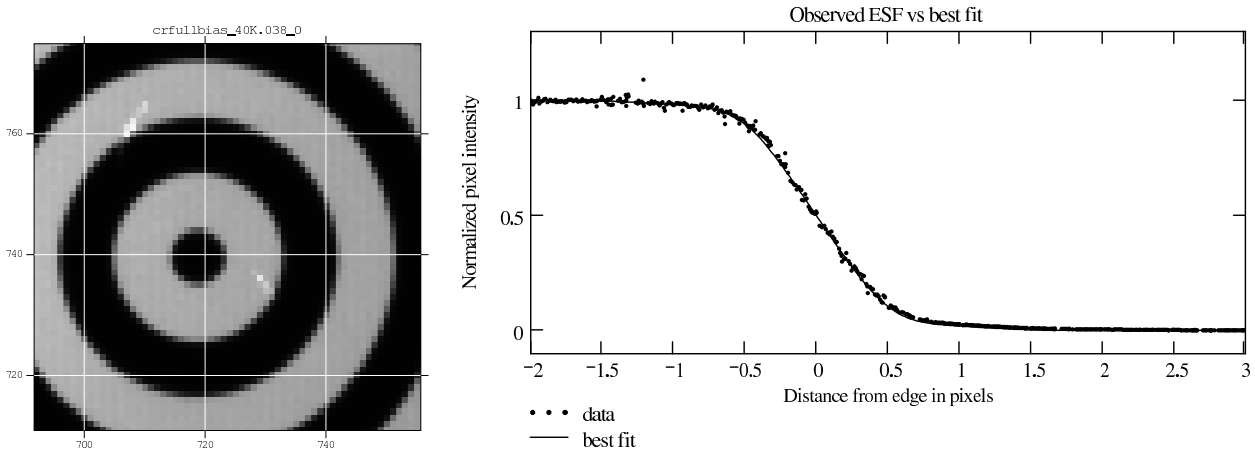

Figure 4. Left: Image of bullseye target. (White streaks are muon hits.) Right: Comparison of observed ESF and best fit model.

was normalized to the bright and dark levels of the image, and plotted against its distance from the center of the bullseye, shifted by the radius of the ring. Vibration from the closed-loop refrigeration unit occasionally moved the reticle, so it was necessary to re-center the edge for each image and reject images when there was evidence of motion. Many runs of this were averaged into bins 0.01 pixel in width. The gaussian shape parameter was varied and the mean square error between the model and the binned edge data was tallied. The shape parameter that provided the lowest mean square error, $b = 0.475$, was used as the estimate of spreading. The right side of Figure 4 shows the best fit model plotted against the normalized and averaged data. Converting this into the frequency domain results in the MTF curves shown in Figure 5.

A shape parameter of $b = 0.475$ corresponds to a σ of $0.475/\sqrt{2\pi} = 0.19$ pixels, or roughly 5.1 micron spread in the detector. Thus, 98.4 percent of the carriers produced by a point source centered on a pixel should remain in that pixel's detection node; each immediate neighbor node should only get 0.4 percent.

8. TESTING OF READ NOISE AT 35K

The read noise of the pin array was measured in RMS electrons with two experiments. In one experiment, a pair of short integration time CDS (Fowler-1) images were subtracted from each other, and the noise in the difference image was analyzed. The RMS noise in Analog to Digital Units (ADU) was observed to be 6 ADU RMS. Dividing by the square root of two yielded 4.24 ADU in a single image. The other experiment computed the electron conversion factor by using the noise variance versus signal method.[27] Pairs of images with uniform illumination were taken using successively longer integration times, and the variance of the difference images was plotted against the mean value of the summed images. Since the noise due to photon quantization is poissonian (where the variance is equal to the mean) the slope of this line is the inverse detector conversion factor. This plot, showing $2.4e^-$/ADU is shown in Figure 6. However, the multiplexer impulse response causes the noise squared measurements to be low by a factor of $0.871^2 + 4(0.027^2 + 0.00525^2) = 0.762$. Correcting the slope by this effect results in an estimate of 1.83 electrons per ADU. Combining the results of these experiments yields a read noise of $7.77e^-$ at 35K.

9. GAIN, NODE CAPACITANCE, AND WELL DEPTH MEASUREMENTS

The source follower gain of the main output FET was measured by turning on the reset and varying the reset voltage while reading out frames. To be sure that there were no effects from current flowing during reset, the back bias was disconnected and left floating and the reset FET gate voltage was kept a constant 1.8 volts below the reset drain voltage. A succession of images were taken, and the pedestal images were saved. The right side of Figure 6 shows the resulting measurement: a gain of 0.902 with a slight reduction as the output voltage rose

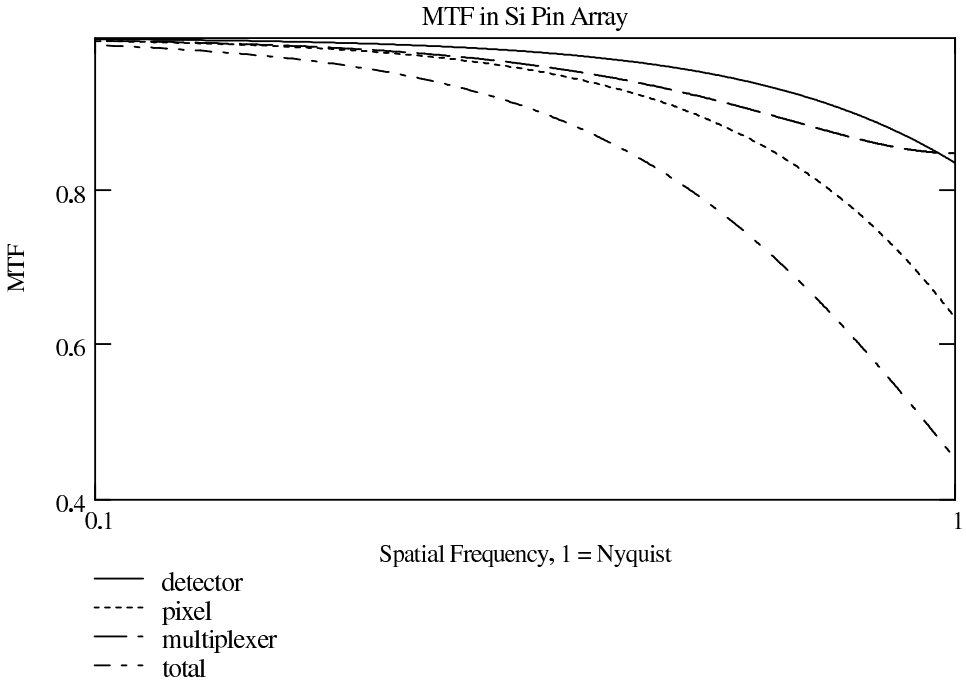

Figure 5. The various components that make up the MTF, plotted relative to Nyquist frequency of one half-cycle per pixel. The detector itself is superb. The overall MTF is dominated by pixel size and inter-pixel capacitance. Frequency is log scale.

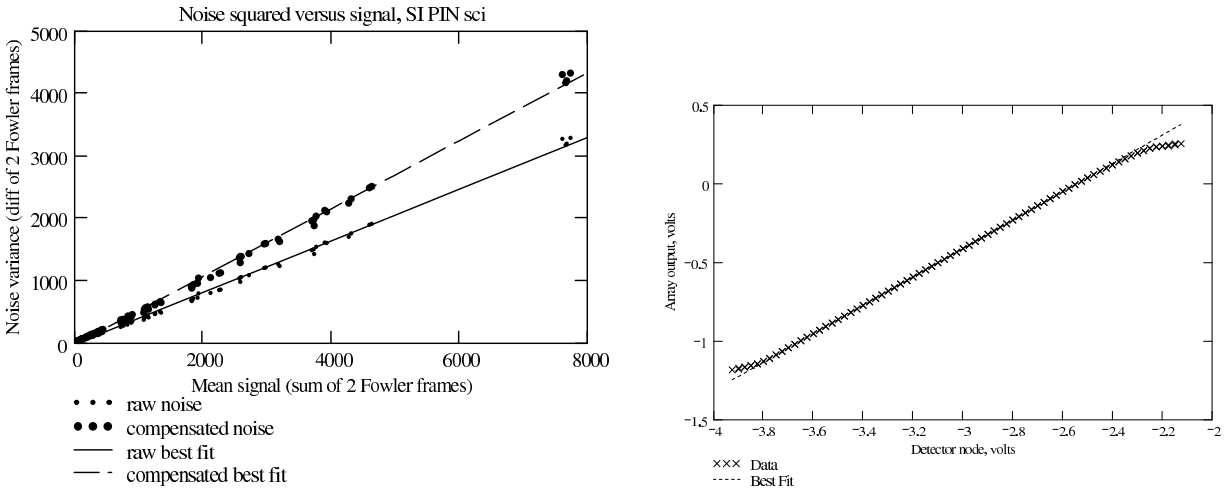

Figure 6. Left: Plot of noise squared versus signal in silicon pin array. Right: Source-follower gain plot, obtained by varying the node voltage with continual reset and observing the output voltage.

above 0 volts. Unity gain of the unit cell output source followers is assumed. Combining the above results yields a capacitance measurement of 43 fF:

$$C = 1.6 \cdot 10^{-19}(\text{coul}/e^-) \cdot 1.83(e^-/\text{ADU}) \cdot (1\text{ADU}/6.1\mu\text{v}) \cdot 0.9 = 43\text{fF}. \quad (5)$$

Interpixel capacitance does not contribute to well depth; it is not included in the node capacitance figure.

The capacitance on a bare mux from the same wafer was measured by the NIR lab at 28 fF. (The same bare mux measured about 30 fF on Pysys.) Interpixel capacitance was not evident on the bare mux. It was expected that the detector would add negligible additional capacitance over the bare mux, so the extra 13 to 15 fF raised a question of its origin. The back bias, connected to the n-doped detector surface on the far left in Figure 3, was varied by 5 volts during an integration. This shifted the resulting image by only 30 millivolts, confirming that there is very little additional detector capacitance (0.6 percent of total or approximately 0.25 fF) added by the pin junction. Varying a field control bias grid, however, indicated a 16.5 percent coupling to this bias as measured at the output. Thus, roughly 18.3 percent (16.5/0.9) of the 43 fF (˜8 fF) is to the grid. Raytheon estimates that the indium bump adds 4 fF, which leaves 3 fF unaccounted for. (If the Pysys 30 fF bare mux estimate is used, only 1 fF remains.)

Images were obtained that used the full range of the converters, (0.4 volts, or 65535 ADU) so the well depth is at least $1.83e^-/\text{ADU} * 65535 = 120,000e^-$. Since the detector is a current source, the well depth is effectively limited only by the multiplexer; thus $300,000e^-$ appears possible given the 1 volt swing measured in the gain test.

9.1. Dark current

It was expected that the dark current would be immeasurable below 150K, and this proved to be the case. Thermal fluctuations of tenths of a degree Kelvin from temperature controller resolution caused dc shifts in the output that made direct measurement of dark current impossible. Measuring the noise due to dark current was more feasible, and in 1000 second integrations, this noise did not rise significantly until 160K. Above 200K, an increase in noise became noticeable at longer integrations. Dark current was not uniform, and "hot pixels" became apparent.

At lower temperatures, dark current was estimated by looking at the noise in the difference of two long integrations with no illumination. The sigma, in ADU, was converted to a sigma in electrons by the conversion factor. Squaring this value gave a variance in electrons squared which, assuming poissonian statistics for the dark current, should be equal to the mean accumulated dark current in electrons. Interpixel capacitance compensation is required for this measurement. Dividing by the effective integration time (twice the integration time of each frame) yields a dark current estimate in electrons per second. When the dc signal was significant enough in the images, direct measurement of dark current was done instead. At room temperature, the dark current filled the wells before even a small portion of the pedestal was read.

10. DISCUSSION

The system seemed to run quite well. The noise result however, even without interpixel capacitance compensation, seemed a little low. Noise of $9.5e^-$ was measured at the NIR lab on a bare SB226 multiplexer from the same wafer. Given the 50 percent increase in capacitance over a bare multiplexer, read noise in electrons should have increased by 50 percent as well. Since the video card had only two converters, it was run at a slower speed to allow the multiplexing; this slower rate may have reduced noise somewhat.

Since the SB226 multiplexer is optimized for cryogenic temperatures, it can be safely said that these arrays have no dark current to speak of. The biases were set for operation at 35K; at temperatures where dark current became apparent, the output waveforms started to look strange. Noise measurements may not be trustworthy at these temperatures.

The pin arrays exhibited crisp imaging, but not as crisp as expected. Interpixel capacitance was quite significant in its effect, both in the edge spread and the compensations required for practically all other measurements. The detector is theoretically capable of much sharper imaging, and discovering interpixel capacitance's effect

Table 1. Noise due to dark current at various temperatures. The measurements below 200K were on the "engineering" array, and the measurements 200K and above were on the "science" array.

Temp K	DC Shift ADU	Noise ADU	Time s	dark current e^-/s
100	–	3.2	1000	0.03
130	–	4.6	1000	0.06
160	–	8.0	1000	0.18
190	–	20.6	1000	1.22
200	–	10.8	300	1.11
210	1900	27	300	7.0
220	10400	114	300	125
240	23000	134	35	1477

upon MTF was somewhat disappointing in this regard. The effects of interpixel capacitance should be correctable with an inverse filter algorithm, limited only by read noise.

A thinner detector (such as a 40 micron thickness) will give a slight advantage over the detector's already superb mtf. Given the same field strength, lateral diffusion length should drop with the square root of thickness. Crosstalk, being gaussian in pin detectors, is predicted to drop linearly with thickness. However, crosstalk is practically nonexistant in the pin detector at a 27 micron pixel pitch. The number of pixels directly affected by cosmic events drops with thickness as well; cosmic events pass through the detector in all directions. A project to thin these devices and reduce cosmic ray susceptibility is underway now.

Spectral sensitivity was not measured, but there is no reason to expect it to be worse than any other silicon photodetector; only improvements are expected.

11. SUMMARY

A simple low-cost high-performance array control system was constructed using open-source components, both hardware and software. The system was used to test hybridized silicon p-i-n arrays at various temperatures, with special interest in the range of the JWST focal plane range of 30K - 40K. The system performed well, with excellent noise performance and reliable acquisition. The pin arrays tested exhibited excellent MTF, despite their thickness. Their read noise was measured at 7.77 e^- in a single frame, with a detector node capacitance of 43 fF. Spread due to transverse diffusion was estimated at 5.1 microns. More of the loss in MTF was due to interpixel capacitance; which influenced practically all other measurements as well. The detectors exhibited very low sensitivity to the back bias, indicating negligible additional capacitance from the detector junction.

ACKNOWLEDGMENTS

Funding for this SB226 pin array research was provided by a grant from NASA. The authors would like to thank all the team members at Raytheon Vision Systems (formerly Raytheon Infrared Operations) especially Jerry Cripe, for producing fine multiplexers and focal plane arrays. Carnegie Institute of Washington's openness with their hardware and software design is greatly appreciated. The Space Telescope Science Institute's development of pyFITS and numarray for the research community at large is likewise appreciated. Dr Paul Barrett at STScI was a wealth of information and encouragement. Andrea Cisternino, the original author of the AMCC PCI device driver, deserves thanks for sharing that effort as well.

REFERENCES

1. A. M. Fowler and I. Gatley, "Noise reduction strategy for hybrid IR focal-plane arrays," in *Proc. SPIE, Infrared Sensors: Detectors, Electronics, and Signal Processing,*, T. S. Jayadev, ed., **1541**, pp. 127–133, Nov. 1991.

2. A. M. Fowler and I. Gatley, "Demostration of an Algorithm for Read-Noise Reduction in Infrared Arrays," in *The Astrophysical Journal*, **353**, pp. L33–L34, 1990.
3. J. D. Garnett and W. J. Forrest, "Multiply sampled read-limited and background-limited noise performance," in *Proc. SPIE, Infrared Detectors and Instrumentation,*, A. M. Fowler, ed., **1946**, pp. 395–404, Oct. 1993.
4. J. D. Offenberg, "Cosmic Ray Rejection Aboard the Next Generation Space Telescope," *www.ngst.nasa.gov-/doclist/bytitle.html* **Document 231**, 1998.
5. G. Burley, "www.ociw.edu/~burley/ccd/base.html." web site, 2002.
6. W. J. Forrest, H. Chen, J. D. Garnett, S. L. Solomon, and J. L. Pipher, "Near-infrared arrays for SIRTF, the Space Infrared Telescope Facility," in *Proc. SPIE, Infrared Detectors and Instrumentation,*, A. M. Fowler, ed., **1946**, pp. 18–24, Oct. 1993.
7. J. Wu, W. J. Forrest, J. L. Pipher, N. Lum, and A. Hoffman, "Development of infrared focal plane arrays for space," *Review of Scientific Instruments* **68**, pp. 3566–3578, Sept. 1997.
8. J. L. Pipher, W. J. Forrest, and J. Wu, "InSb arrays for SIRTF," in *Proc. SPIE, Infrared Detectors and Instrumentation for Astronomy,*, A. M. Fowler, ed., **2475**, pp. 428–434, June 1995.
9. A. W. Hoffman, K. J. Ando, A. D. Estrada, J. D. Garnett, N. A. Lum, P. J. Love, J. P. Rosbeck, K. P. Sparkman, A. M. Fowler, J. L. Pipher, and W. J. Forrest, "Near IR arrays for ground-based and space-based astronomy," in *Proc. SPIE, Infrared Astronomical Instrumentation,*, A. M. Fowler, ed., **3354**, pp. 24–29, Aug. 1998.
10. G. G. Fazio, J. L. Hora, S. P. Willner, J. R. Stauffer, M. L. Ashby, Z. Wang, E. V. Tollestrup, J. L. Pipher, W. J. Forrest, C. R. McCreight, S. H. Moseley, W. F. Hoffmann, P. Eisenhardt, and E. L. Wright, "Infrared array camera (IRAC) for the Space Infrared Telescope Facility (SIRTF)," in *Proc. SPIE, Infrared Astronomical Instrumentation,*, A. M. Fowler, ed., **3354**, pp. 1024–1031, Aug. 1998.
11. python.org, "www.python.org." web site.
12. A. Rubini and J. Corbet, *Linux Device Drivers, 2nd ed.*, O'Reilly, Sebastapol, CA, 2001.
13. P. Greenfield, J. C. Hsu, and W. Hack, *PyFITS Users Manual*, Space Telescope Science Institute, 2002.
14. R. L. White and P. Greenfield, "Using Python To Modernize Astronomical Software," 1999.
15. P. Barrett and W. Bridgman, "PyFITS, a FITS module for Python," *Astronomical Data Analysis Software and Systems VIII*, 1998.
16. T. Denault and K. M. Hawarden-Ogata, "irtf.ifa.hawaii.edu." web site.
17. P. Hickson, "A Visible Imager for NGST," *www.ngst.nasa.gov/doclist/bytitle.html* **Document 518**, 1999.
18. R. Alexander, "NGST Visible Imager Detector Testing Report," *www.ngst.nasa.gov/doclist/bytitle.html* **Document 937**, 2001.
19. P. Hickson, "Canadian Visible Imager for NGST Final Report," *www.ngst.nasa.gov/doclist/bytitle.html* **Document 936**, 2001.
20. R. Alexander, "Potential CSA Contributions to NGST; Science Instruments and Observatory and Ground Systems," *www.ngst.nasa.gov/doclist/bytitle.html* **Document 542**, 1999.
21. G. Ottaviani, L. Reggiani, C. Canali, F. Nava, and A. A. Quaranta, "Hole drift velocity in silicon," *Physical Review B Solid State* **12(8)**, pp. 3318–3329, 1975.
22. C. Canali, F. Nava, and L. Reggiani, "Hot electron transport in semiconductors," *Topics in Applied Physics* **58**, pp. 107–110, Springer-Verlag, (Berlin), 1985.
23. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, p. 47. John Wiley and Sons, New York, 1978.
24. R. N. Bracewell, *Two-Dimensional Imaging*, p. 149. Prentice-Hall, Englewood Cliffs, NJ, 1995.
25. G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis*, pp. 122–127. Holt, Reinhart, and Winston, 1986.
26. J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, pp. 339–345. John Wiley and Sons, New York, 1978.
27. L. Mortara and A. Fowler, "Evaluations of CCD: Performance for Astronomical Use," *Proc. SPIE, Solid State Imagers for Astronomy* **290**, pp. 28–30, 1981.