Reconfigurable electronics at the IOTA interferometer

*Pedretti E.[abc], Millan-Gabet R.[a] Monnier J.D.[a], Morel S.[a]
Traub W.A. [a], Carleton N.[a], Berger J.P.[ea], Schloerb P.[d]
Brewer M.[d], Ragland S.[a] and Lacasse M.[a]

[a]Harvard Smithsonian Center for Astrophysics, 60 Garden street, Cambridge MA, USA.
[b]College de France, Place Marcelin Berthelot, Paris, France.
[c]LISE, Observatoire de Haute Provence, Saint Michel l' Observatoire, France.
[d]University of Massachusetts at Amherst, Amherst MA, USA.
[e]LAOG, Observatoire de Grenoble, Grenoble, France.

ABSTRACT

We describe the new control system for the PICNIC near-infrared camera and the visible star tracker, implemented at the IOTA interferometer, based on the ALTERA Complex Programmable Logic Device (CPLD) technology. These digital components provide an adaptive interface between the control system and the cameras used at IOTA, allowing flexibility when connecting very different devices. In particular the clocking and processing circuits used for the PICNIC camera can be changed in milliseconds during normal operation. The camera can then switch between full quadrant readout mode used for alignment and diagnostics, and a N pixel readout mode used for science operation.

Keywords: infrared interferometry, detector control, data acquisition, CPLDs, instrumentation.

1. INTRODUCTION

A digital circuit is built by connecting several devices which perform logic functions. Such devices usually provide standard functions of different complexities, starting from simple logic functions such as Small Scale Integration (SSI) and Medium Scale Integration (MSI) Transistor-Transistor Logic (TTL) device and ending in high performance full custom designs such as microprocessors and random access memories. Full custom design requires years of development and test and is used when high volume production is required.

There are intermediate solutions for building prototypes and for small volume production. The so called Programmable Logic Device (PLD) permits full construction of logic circuits by internally connecting basic logic functions such as logic ports and flip-flops by burning fuses or using electronic switches.

CPLDs and FPGAs (Field Programmable Gate Arrays) permit the implementation of complex digital circuits on a single chip. CPLDs tend to have faster and more predictable timing properties while FPGAs offer the highest gate density[1]. The number of gates in the chip is on the order of hundreds of thousands and this number is increasing rapidly while the technology improves. In the Altera Flex-10K70, used in the PICNIC IR-camera at IOTA, gates can be interconnected to form complex networks using electronic switches; the switch interconnections are stored on a static RAM, so the interconnection among elements can be changed in just milliseconds. This means that a single component can host several circuits performing very different functions at different times.

CPLDs can be programmed using different tools:

- CAD entry, using standard electronics symbols interconnected by drawing wires.
- AHDL (Altera High level logic Description Language): C-like Altera proprietary language which describes the interconnection of various components into a logic circuit.

*Smithsonian Predoctoral Fellow.

- VHDL (Very High level logic Description Language): Pascal-like language which describes interconnection among logic blocks.

There is no particular advantage in using VHDL with respect to AHDL or the CAD entry but we decided to use VHDL because it is a widely used industrial standard.

The work presented in this paper concerns the design of several circuits loadable in the two CPLD interfaces used at IOTA and of the control software built for them. In particular:

- A circuit for fast readout of a visible CCD matrix used for the detection of star centroid error in the tip-tilt system, which feeds the telescopes' light to the beam combiner.
- A circuit capable of reading a whole 128×128 PICNIC matrix for future use as a IR wavefront sensor for the tip-tilt correction system and for the beam combiner alignment.
- A circuit for the fast readout of the IONIC[2] integrated optics beam combiner or the IOTA free-space combiner (or any future beam combiner to be implemented at IOTA) through the readout of sparse pixels on the PICNIC matrix.
- Software libraries built to interface the CPUs to the various circuits.

2. THE DATA ACQUISITION AND CONTROL SUBSYSTEM

Driven by the requirements of the 3-telescope upgrade at the IOTA interferometer[3], the data acquisition and control system at IOTA has undergone through major changes. The distributed computing network once composed of separate computers running different operating systems are now replaced by a system based on the VME bus and the real time development system *VxWorks* in order to cope with the increased data processing needs introduced by having two new baselines and an additional telescope. A block diagram of the real time system is shown in Figure 1

The new control system is composed of three Central Processing Units (CPUs) based on the Motorola Power-PC microprocessor architecture. This hardware runs on the VxWorks real time operating system. The logic functions programmed inside the CPLDs[1] are implemented using a high level description language called *VHDL*. The circuit can change according to the application and is downloaded by the CPU to the CPLD board permitting very high flexibility in interconnecting dramatically different digital hardware. This means, for example, that a new IR star tracker could be plugged into the system in place of the visible star tracker without changing the interface. Only a different circuit is loaded and a different adapter plugged in. The adapter is simply a passive electrical adaptation from the Eurocard connector on the backplane of the VME rack to the electronics to be controlled.

2.1. The VxWorks real-time system

The control system for the IOTA interferometer is centralised in a VME type computer. VME is an industrial standard defining the mechanical and electrical characteristics of a computer bus, oriented to instrumentation control. CPUs based on different processors can be plugged into the bus: the most widely diffused architectures are Power-PC (a Motorola - IBM joint venture), SPARC (SUN based architecture) and Intel architecture.

Real-time means that the different tasks are executed at very precise intervals in time, synchronised by a "real time clock" which defines the pace of the system. This is crucial for instrument control (in our case controlling visible and IR cameras, telescopes, delay lines, fringe trackers and data storage).

The VxWorks system is UNIX based, meaning that all the development can be done on a UNIX computer, using standard cross-compilers (GNU) editors, etc. UNIX is not per-se a real time system (RT-LINUX is a notable and very interesting exception in the open-source real time world). The UNIX computer host (in our case a SUN station) is connected to the VME rack through a fast optical link and communicates with three VME-based CPU targets. The SUN host loads the compiled programs on the targets but these programs (tasks) are executed using another operating system (VxWorks) which is basic and streamlined, but allows us to use standard UNIX sockets to communicate with UNIX based workstations, file-servers etc on a standard Ethernet connection.

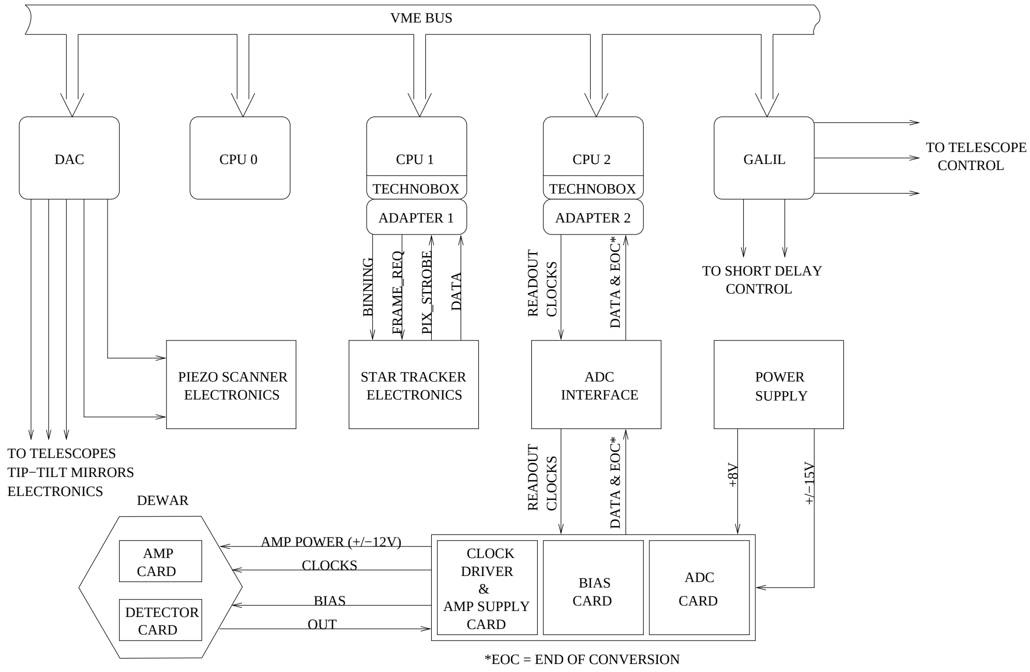

Figure 1. Control system block diagram at IOTA. The three CPUs manage different functions of IOTA. CPU-0 controls the telescopes and the short delay lines through the Galil interface. The long delay lines are controlled through a standard Ethernet socket. Components, like shutters and remotely operated mirrors, are operated through a serial interface board. CPU-1 is entirely dedicated to the wavefront tip-tilt sensing and correction. The interface functions for the wavefront sensors are implemented through the Technobox re-configurable interface, which uses an Altera CPLD. An adapter was built to drive a differential cable which runs from the Technobox interface to the remote star-tracker camera. CPU-2 is dedicated to data acquisition and fringe-tracking. The CPLD on the Technobox interface performs all the tasks necessary to the PICNIC camera operation.

2.2. Reconfigurable electronics versus programmed I/O

Programmed I/O using microprocessors, microcontrollers or computers connected to hardware implemented with parallel interfaces is widely used to control equipment. The advantage of these systems is that their operating mode can be easily changed by modifying the control program, compared to hard-wired digital circuits which once established can normally not be changed. Digital circuits have the advantage of being usually faster and more predictable in real time applications because they are tailored to the specific task, while microprocessors are built for general applications. CPLDs get advantages from both worlds.

- The circuit can be changed in real time like a program in a processor based machine.
- The digital circuit is tailored to the application.
- It is advantageous to use the same interface for controlling different hardware, for maintaining spare boards and for debugging purposes.

2.3. Disadvantages of reconfigurable hardware

Programming FPGAs or CPLDs is not as straightforward as programming a micro-controller. The application is still a digital circuit which, even if it has been implemented correctly from the digital logic point of view, could still not work, or would work at slower speed than expected, due to internal propagation delays of the clock tree and/or of specific signals. CPLDs are usually more predictable than FPGAs from the internal delay point of view[1].

2.4. Circuit implementation

We developed our circuits on a commercially available PCI-PMC card which contained an Altera 10K70 CPLD. This card plugs directly into the PMC slot of our VME CPUs. The advantage of this configuration is being able to load a new circuit in the CPLD from the CPU memory or from the disk during operation. The disadvantage of this configuration, we found out, is in forcing outputs to be be assigned to specific components and I/O pins. The compiler prefers that the pins are freely assigned the first time it builds the circuit, in order to optimise the routing of signals. Fixed output pins likely add delays to the circuit which makes it failure prone at higher speed. Another disadvantage of using a PMC interface is the intrinsic difficulty in using an oscilloscope probe on the component; also the real time debug tools present in the parallel port programming interface cannot be used, complicating the debugging process.

A PMC interface, electrically identical to a standard PCI bus used for on personal computers, connects the CPLD to the CPU. There are several components present on this board. The components used for the star-tracker and IR cameras are:

- The PLX-9050 PCI bridge, which is connected to the PCI bus; its main function is the bidirectional data transfer between the PCI bus and the internal bus of the board. An internal bus composed of 16 data lines, 18 address lines and read-write and enable signals, interconnects the components present on the board. It can generate interrupts on the PCI bus in response of events generated by the CPLD.
- A $128K \times 16$ static RAM connected to the internal bus. The CPU has access to this RAM through the PCI bridge. The RAM is shared by the CPLD using arbitration signals.
- A 33 MHz clock line (PCI clock) used as master clock for the CPLD.
- An Altera 10K70 CPLD. This component is programmed through the PLX-9050.

3. VISIBLE STAR TRACKER READ-OUT USING A CPLD

Unlike the previous version of star tracker at IOTA, which used two CCD detectors (one for each telescope), the new star tracker uses only one detector for acquiring stellar images from three telescopes. The software running on CPU-1 is responsible for bringing the star to the right position on the detector and sending corrections to the tip-tilt mirrors which feed the beam combiner. The circuit built on the CPLD interfaces the existing CCD electronics to CPU-1 without building any extra hardware, with the exclusion of interconnection cables and differential line drivers. The star tracker camera transfers data on a 12-bit parallel data bus. Three control lines are used:

- Frame_req; this line is asserted by the interface to ask for a new frame.
- Pix_strobe; this line is asserted by the camera after a new word is put on the data bus.
- Fine/coarse; this line is asserted by the interface to select the coarse or the fine mode for the readout.

The transferred image is 8×8 pixels wide (64 data words are transferred), in coarse or fine pixel mode. In coarse mode the full 32×32pixel CCD image is binned in blocks of 4×4 to generate an 8×8 output. this 8×8 image has four 4×4 quadrants; three of the quadrants are used for the star images from our three telescopes. In fine mode only the central 8×8 pixels are read out, and the rest ignored; this mode cannot be used for three stars. The camera is connected to the interface with a long parallel cable using differential line drivers and receivers. Figure 2 shows the star tracker implementation on the reconfigurable interface. The CPLD (shaded box) is controlled through the PCI bridge which can write into a control register defined in the CPLD. The acquisition starts when the CPU asserts the start bit in the control register. This bit is hard-wired to the control line "Frame_req" of the CCD star-tracker and also connected to a state machine inside the CPLD. The state machine changes state and asks the CPU through the PCI bridge for the control of the internal bus; the PCI bridge asserts a line to inform the state machine that the CPU has released the bus.

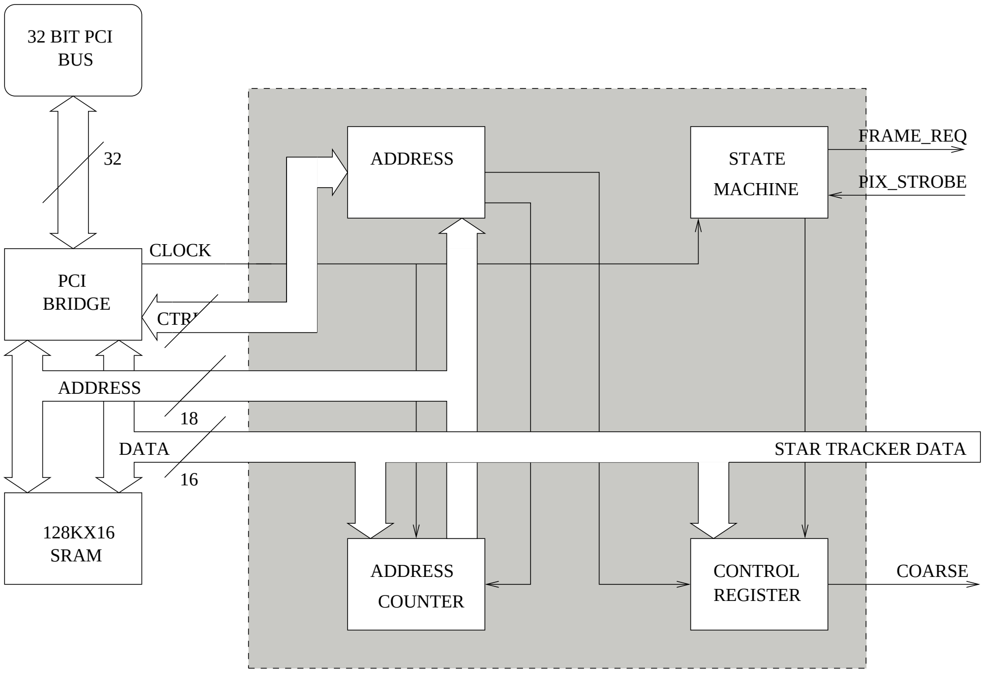

Figure 2. The visible star tracker data block diagram. The state machine implemented here is mostly responsible for the arbitration of the internal bus with the PCI-bridge. It also sends a frame request to the star-tracker camera. The star tracker then clocks data words out, one by one and increments the address counter until 16 data words are transfered.

The address counter is enabled; this counter is connected to the address lines of the static RAM and it is incremented directly (after being synchronised with the 33MHz system clock) by the star tracker through the signal "Pix_strobe". The data bus is bufferised by D-type flip-flops (to synchronise it with the system clock) and connected to the RAM buffer through tri-state ports, to avoid conflicts with the CPU signals. Bus arbitration is performed by the CPLD and the PCI bridge. After the 64 words have been transferred, the CPLD asserts the INTA line of the PCI bridge. This line is an interrupt which is serviced by the CPU as a high priority task. When received, the CPU transfers the block of 64 words from the static RAM to a buffer and processes it to calculate the centroid of the star and correct the star-tracker tip-tilt mirror.

If for any reason the number of transferred words is less than 64, the system would hang, waiting for the missing words. To avoid this we implemented a watchdog timer; the timer is reset by the INTA signal every time a data frame of 64 words is received. If data transfer takes more than 4.3 ms (the total frame time is around 3.3 ms) the timer sets the timeout flag in the control register, informing the CPU that a communication problem has occurred.

4. AN INFRARED CAMERA BASED ON THE PICNIC DETECTOR AND CPLD TECHNOLOGY

4.1. The PICNIC detector

The PICNIC focal plane array[4, 5] is composed of four 128 × 128 quadrants of a HgCdTe detector which can be addressed simultaneously and independently. The active area of each pixel is $40\mu m$ square. The PICNIC detector is the evolution of the NICMOS3[4, 5] array. The quantum efficiency (QE) of the detectors is better than 50% for wavelengths in the range of $0.8 - 2.5\mu m$.

The PICNIC detector is a hybrid device where the detectors are built on a small band-gap material (HgCdTe) on an insulator substrate (sapphire) which is interconnected to a silicon substrate, where the clocking and analog

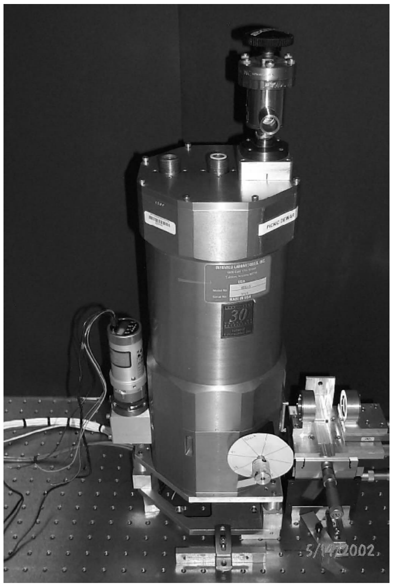

Figure 3. The PICNIC camera dewar and the optics focusing the beams from the IONIC3T beam combiner are shown in this picture. The mechanical assembly is an improved version of the previous NICMOS3 design, but the dewar is taller and contains more liquid nitrogen, one extra filter wheel is implemented, (in order to insert density filters), and both filter wheels are able to be motorised.

circuitry are built and interconnected through indium bumps (a well established technology where indium is deposited on the contacts of the two wafers and the bumps are pressed together to make contact).

The P-N junctions deposited on sapphire are illuminated through this substrate. These junctions are reverse biased during the RESET cycle, when the junctions are connected to a positive potential, so that conduction cannot happen and the width of the depletion zone across the junction is increased. We recall that the depletion zone is a region devoid of free carriers (electrons or holes) which naturally forms at the interface of different conductor or semiconductor materials (eg. copper-constantan thermocouple for metals, or semiconductors doped with acceptors or donor atoms; P and N semiconductors). The junctions can also be considered, in the case of reverse bias, as a capacitor charged by a bias potential. After the bias potential is applied and disconnected, when a photon hits this region it will generate a electron-hole couple (intrinsic carriers) which will recombine in the depletion zone reducing its thickness and also the electric charge deposited on the two sides of the junction. This voltage decrease can be measured through a source follower circuit, a MOS feedback amplifier of unity gain and very high input impedance. Since this discharge follows the reverse bias characteristic I-V curve of a diode it is intrinsically non-linear before the saturation region.

When the photon-generated carriers have depleted all the initial charge separation, the device is saturated. The dynamic range of the detector is then determined by the saturation voltage applied at the beginning of the cycle.

4.1.1. Analog electronics

The analog electronics on the new PICNIC camera is identical to the electronics used in the IOTA NICMOS camera[6] with the exception of new PCB housing the array itself.

4.1.2. Clocking the detector

The detector matrix is addressed through two shift registers: the horizontal shift register and the vertical shift registers. These are controlled respectively by the LINE, FSYNC clocks and the PIXEL, LSYNC clocks. FSYNC and LSYNC act as reset for the shift registers (active low): when they are asserted the respective

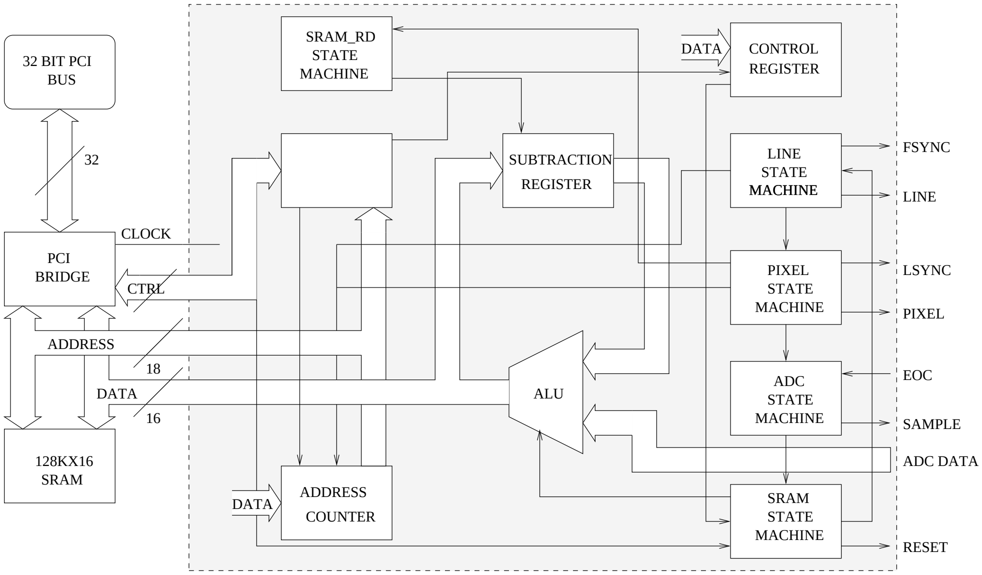

Figure 4. The PICNIC quadrant readout circuit. Five state machines are responsible for the operations of the camera. Complex functions are broken into elementary functions of clocking lines, pixels, resetting the detector, commanding the ADC and transferring data into SRAM, executed by each single state machine.

registers are loaded with all zeroes. The LINE and PIXEL clocks are double edge clocks. The first edge of such clocks will simply connect the line or pixel circuitry to the output of the detector; the second edge will put a zero in the first bit of the clocked shift register. The following clock edges will shift the zero of one position throughout the register; the zero bit will enable an analog switch to read the corresponding line and pixel of the detector. Before reading the detector the pixels must be connected to the bias potential, in order to charge the junction capacitance. The circuit used in the PICNIC detector permits us to reset a whole line at the time, a large improvement in speed and performance with respect to the NICMOS3 detector, where the pixels were reset one at a time. Asserting the RESET signal (active high) together with the LINE clock will reset the whole line.

4.1.3. Implementation of the quadrant readout mode

Although the camera's principal use is reading out a sparse number of pixels illuminated by the beam combiner, it is necessary to read a whole quadrant for alignment purpose, and, because in the future the camera could be used as an IR star tracker, to track sources which are too faint at visible wavelengths for the CCD detector. The mode adopted for quadrant readout is Correlated Double sampling (CDS).

4.1.4. Correlated double sampling

In CDS mode the detector is reset, sampled, allowed to integrate and re-sampled. The difference between the two frames is then recorded. This reduces noise and eliminates detector offsets

CDS is implemented entirely in the CPLD electronics. The CPU receives a buffer containing the final, ready to use, data.

This mode reads a whole quadrant from the PICNIC array; the image is stored in the static RAM on the Technobox card which is shared between the CPU and the CPLD through the PCI bus. The readout starts when the start bit in the control register of the CPLD is asserted by the CPU. The interrupt line (INTA) is asserted when the readout process has finished and a frame is present in the SRAM.

Between these events many operations are handled in parallel by state machines which have been built into the CPLD; these state machines are:

- SRAM state machine which handles communications with the PCI bridge for bus arbitration. This state machine is a slight modification of the state machine used for the same purpose in the star-tracker. It polls the start bit of the control register and sets a semaphore which starts the clock generation state machine. If a reset frame and an image frame have been acquired and subtracted, it releases the PCI bus and asserts INTA.
- LINE state machine, which controls the line clocks generation for the PICNIC array; once a line has been selected it sets a semaphore to start the PIXEL clock generation; lines are counted by the line counter. It will also assert the RESET line, together with FSYNC if the toggle flip flop, which keeps the state of the frame (reset-frame or data-frame), is set on reset-frame.
- PIXEL state machine, which controls the PIXEL clock generations for the PICNIC array; it controls the sequence of clocks and sets a semaphore when the ADC can sample the selected pixel; pixels are counted by the pixel counter.
- ADC state machine: it sends start of conversion to the ADC, waits for the end of conversion signal from the ADC and writes a word of data in the static RAM if the toggle flip-flop is in data-frame state. If it is in reset-frame state, it stores the difference between the data word at the current SRAM address and the data word obtained from the ADC; the SRAM address is obtained by combining the line counter and pixel counter.
- Read SRAM state machine: it reads one word of the reset frame previously written to SRAM, during the ADC conversion. The current value of the ADC is subtracted from this value .

The DAQ scheduler task, running on the CPU when in quadrant state loads the circuit into the CPLD and initialises the readout parameters. After that the start bit is asserted in the CPLD register and the state machines are started.

4.1.5. Readout Parameters

Some readout parameters can be changed by the application running on the CPU, accessing the registers on Altera:

1. PICNIC base clock period. The 33 MHz PCI clock is divided by a programmable counter built into the CPLD, in order to generate the base clock rate used to clock the PICNIC array; the counter can be programmed by the CPU through the PCI bridge.
2. Delay between PICNIC clock and sample assertion; this is necessary to avoid sampling the PICNIC pixel when it is not stable, after it has been addressed, which results in increased readout noise. The delay is generated by a programmable counter and is accessible to the CPU through the PCI bridge.
3. Integration time; this is also a presettable counter which determines the time between the reset frame and the image frame
4. Sub-quadrant corner position; with these variables it is possible to define a sub quadrant to read a small area of the detector. This flexibility will be crucial when implementing an IR wavefront tip-tilt sensor.

4.1.6. Interferogram detection - scan mode

This mode reads a number of pixels at arbitrary coordinates on the camera. For better clocking efficiency the pixels should be closely spaced and, preferably, on the same line. The pixel voltages are read sequentially and stored in SRAM. INTA is asserted at the end of every group of pixels readout. INTA is used to delimit the end of a data frame but it is also used to step the piezo OPD scanner which is synchronous with data acquisition. This simple cycle is performed several times and stored in memory as a time sequence. When one sequence (or scan) is finished a flag is asserted in the control register and the CPU transfers the stored vector to a buffer.

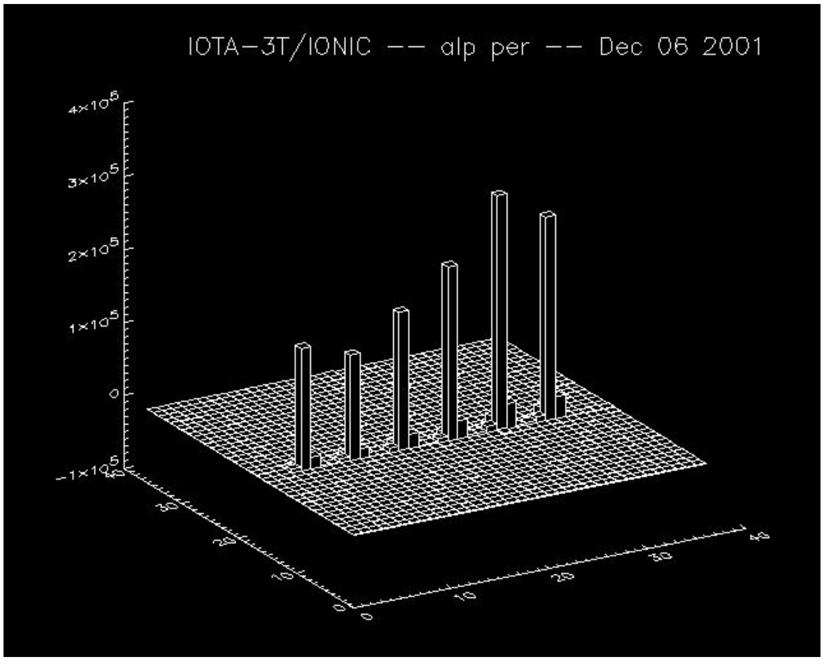

Figure 5. Image of the pixels illuminated by the IONIC3T beam combiner on the PICNIC camera in quadrant readout mode. Two consecutive channels contain information about the same baseline and they are subtracted one from the other to improve SNR and calibration.

4.1.7. Differential sampling

With this method the array is reset only once, then sampled for the pixels of interest and the difference between a sample and the following sample in time is taken; avoiding the array reset for every frame readout improves the SNR and the readout speed but reduces the dynamic range of the detector.

4.1.8. Circuit implementation

We use here the same methods implemented in the NICMOS3[6] camera previously employed at IOTA: each pixel can be sampled several times and the values added up and stored in a register; we call this *reads*. It is also possible to read the whole group of pixels and integrate the values separately; we call this *loops*. In this way values are integrated and the readout noise is divided by $\sqrt{N}$ where $N = loops \times reads$. Using many reads achieves a better SNR but increases the latency between sampling different pixels; using many loops degrades slightly the SNR since we are clocking the detector, which adds noise, but the time delay among different pixels is reduced.

This was implemented in the readout software for the NICMOS3 camera. The hardware implementation of this function for the PICNIC was more complex than the software implementation for the NICMOS3 camera. Data for each pixel must be stored in separate registers and added there, then data is transferred to SRAM and INTA is sent to the CPU through the PCI-bridge. The interrupt must be cleared by the CPU at the end of this process. To limit access to the SRAM and avoid conflict with the CPU we opted to do processing in the internal registers rather than in the SRAM as was done for the quadrant readout. This unfortunately limits the number of pixels that can be read to nine, since generating internal memory in the CPLD is intrinsically inefficient. This limitation will only become important when spectral dispersion is implemented[7].

The interrupt generated from the CPLD, other than merely informing the CPU that a new data point is available in the SRAM, is also used for incrementing the voltage ramp generated by a DAC card installed on the VME bus. The pixel registers, the adders used to integrate the value of the pixels, the counters which count

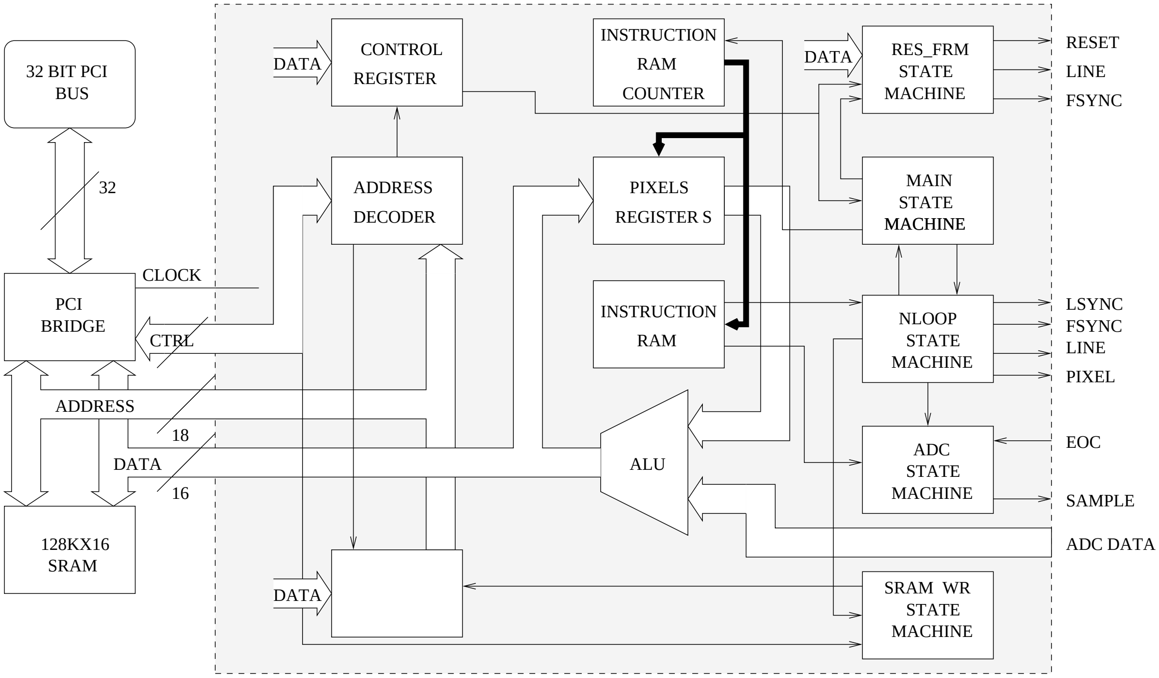

Figure 6. The block diagram of the PICNIC camera in scan mode. The basic functions are implemented with five state machines. The same functions of the NICMOS3 camera are available for integrating pixel values. The pixel readout sequence is stored in the instructions RAM, internal to the CPLD. This RAM stores micro-coded instructions specialised to generate clocks sequences for the PICNIC array.

the number of data points transferred, the SRAM address, the internal RAM address and clock generation are controlled by five separate state machines:

- RES_FRM_PROCESS state machine: this state machine resets the whole PICNIC array continuously, clocking lines from 0 to 127 and back to zero. This process is stopped when the start flag is asserted and the main process is idle.

- main state machine: schedules the N_LOOP_PROCESS state machine and increments the address of the instruction internal RAM which contains the readout sequence transfered from the CPU. It starts when the start bit of the control register is asserted; it clears all registers and starts the loop state machine. It ends when the loop_tc semaphore is asserted.

- N_LOOP_PROCESS state machine: it is scheduled by the main process and decodes part of the instructions written in the internal RAM. The internal RAM is loaded by the CPU via the DAQ scheduler program. It schedules the SAMPLE_PROCESS and the SRAM_WR_PROCESS.

- SAMPLE_PROCESS: it sends start of conversion to the ADC, waits for the end of conversion signal from the ADC and writes a word of data to the internal register corresponding to the pixel sampled; if the requested number of reads is greater than one the state machine reads the ADC several times and the values are accumulated in the corresponding pixel register.

- SRAM_WR_PROCESS: this state machine executes the data transfer to SRAM from the internal registers when the requested loops and reads are completed. When the sequence starts again to sample a new data point it executes in parallel, transfers the data from the pixel registers to SRAM and sends an interrupt when the process is completed.

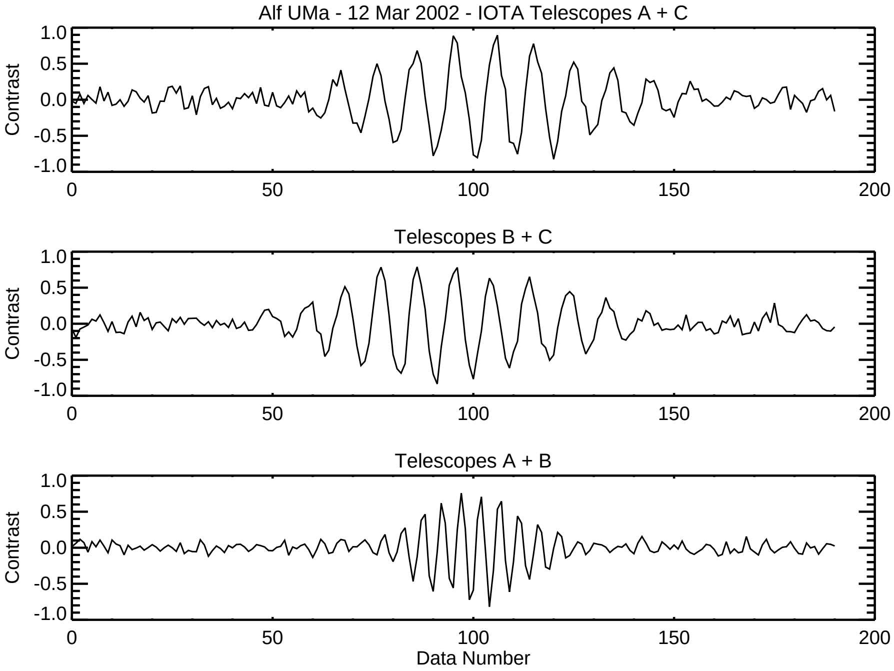

Figure 7. Fringes from the IOTA IONIC3T combiner on the PICNIC camera obtained using the CPLD sparse pixels readout circuit. Complementary fringes for each telescope pair have been subtract to enhance the SNR.

Instruction	Hex Value	Action
fsync	02	FSYNC pulse generated
lsync	03	LSYNC pulse generated
line	00	LINE clock transition n-times
pixel	01	PIXEL clock transition n-times
jump	04	Jump to program location

Table 1. CPLD micro-coded instructions

4.1.9. Micro-coded instructions

It is highly desirable to have the flexibility of choosing which pixels are sampled in order to use different combiners. For this reason the scanning sequence has to be alterable. This was solved by allowing the state machines to change state according to a program written into a RAM internal to the CPLD. Five instructions can be executed by the state machines as shown in Table: 1 These instructions permit to generate a clock sequence; the LINE clock will change level n-times according to the value of the following byte in the program. The same happens with the PIXEL clock, but in this case every pixel is read n-times, according to the value written in the "Nreads" register. Loops are achieved through the jump instruction which permits to branch back to a specific instruction in the program. The loop is executed n-times according to the content of register "Nloops".

4.1.10. Readout parameters

In this readout mode some of the parameters, which can be changed, are the same as in the quadrant readout circuit. Some registers are only used in the pixel readout mode.

1. PICNIC base clock period: same as in Section: 4.1.5 for quadrant mode.
2. Delay: same as in Section: 4.1.5 for quadrant mode.
3. Nloops: reads the specified pixels group in a loop several times, the value specified in the Nloops register. The values read are integrated separately in the pixels registers.
4. Nreads: reads the specified pixel several times the value specified in the Nreads register. It integrates the values for that pixel only and puts the value in the corresponding pixel register.
5. Readout sequence: vector containing the readout code sequence used to sample the pixels: the vector contains instructions which assert the lines LSYNC or FSYNC for the detector as well as the LINE and PIXEL lines. Line and pixel instructions have arguments which specify the number of lines or pixels to clock the detector
6. Nsamples: number of npixels sample acquired and stored into memory.

4.1.11. The C libraries

The PPC-CPU is interfaced to the Altera CPLD using a PMC bus, as described earlier. The kernel of this library is given by the manufacturer of the card (Technobox). This is only an example which can be modified by the user according to the CPLD programming. The modules dealing with the PCI bridge and the remote programming of the CPLD do not need any modification.

Two basic function are associated with the library:

- Writing to the control register to modify the behavior of the CPLD.
- Read the static RAM and transfer its contents to a data buffer.

These functions are common to the CCD and the PICNIC camera readout circuits. .

5. FUTURE DEVELOPMENTS

Based on this positive first experience we envisage including other instruments under the control of the CPLD reconfigurable interface. The first application in order of importance will be the interfacing of the avalanche photo-diodes (APDs)[8] with the VME rack. These detectors are used for the visible beam combiner. This will entail implementing another circuit on the CPLD which will count the photo-events from the detectors. The rest of the electronics which deal with transferring data to the SRAM will probably stay untouched.

It has also been proposed to install an IR star tracker at IOTA using the previous science camera based on the NICMOS3 array. This requires some slight modifications for the clocking circuits used for the PICNIC camera but essentially most of the circuit used there for the quadrant readout could be ported untouched.

More sophisticated processing could also be ported to the CPLDs: in the case of the star-tracker circuits could be implemented to calculate the star centroids, unloading the CPU from processing tasks. Likewise, the piezo OPD scanner could be controlled directly by the CPLD by means of DAC circuits connected to its outputs, increasing the resolution of the waveform generated to scan the piezo mirrors.

6. CONCLUSIONS

We have shown that CPLDs are a mature technology which can handle the many digital circuit necessary to interferometer subsystems. At IOTA we were able to implement the complex functions necessary to read our science detector for different modes of operations, entirely on the CPLD. We have also shown the advantage of being able to change digital circuits on the fly and of using a standard interface for controlling very different electronics.

ACKNOWLEDGMENTS

We thank the Altera corporation for providing the CPLD development software through the Altera University Program.

REFERENCES

1. J. O. Hamblen and M. D. Furnan, *Rapid prototyping of digital system*, Kluwer Academic Publishers, 2000.
2. J. P. Berger and M. R. Kern, P., "The IONIC3T Integrated Optics Beam Combiner," in *Proc. SPIE in preparation*, Aug. 2002.
3. W. A. Traub and M. R. Carleton, N., "The IOTA Interferometer Three-Telescope Upgrade," in *Proc. SPIE in preparation*, Aug. 2002.
4. L. J. Kozlowski, J. T. Montroy, C. A. Cabelli, D. E. Cooper, A. C. Chen, G. L. Bostrup, Y. Bai, K. Vural, K. Hodapp, and D. N. Hall, "Visible and infrared detectors at Rockwell Science Center," in *Proc. SPIE Vol. 4008, p. 1240-1253, Optical and IR Telescope Instrumentation and Detectors, Masanori Iye; Alan F. Moorwood; Eds.*, **4008**, pp. 1240–1253, Aug. 2000.
5. C. A. Cabelli, D. E. Cooper, A. K. Haas, L. J. Kozlowski, G. L. Bostrup, A. C. Chen, J. D. Blackwell, J. T. Montroy, K. Vural, W. E. Kleinhans, K. Hodapp, and D. N. Hall, "Latest results on HgCdTe 2048x2048 and silicon focal plane arrays," in *Proc. SPIE Vol. 4028, p. 331-342, Infrared Detectors and Focal Plane Arrays VI, Eustace L. Dereniak; Robert E. Sampson; Eds.*, **4028**, pp. 331–342, July 2000.
6. R. Millan-Gabet, F. P. Schloerb, W. A. Traub, and N. P. Carleton, "A NICMOS3 Camera for Fringe Detection at the IOTA Interferometer," *PASP.* **111**, pp. 238–245, Feb. 1999.
7. S. Ragland, W. A. Traub, and R. Millan-Gabet, "A Low Resolution Spectrograph for the IOTA Interferometer," in *Proc. SPIE in preparation*, Aug. 2002.
8. C. M. Coldwell, C. D. Papaliolios, and W. A. Traub, "First visible-light measurements from the IOTA interferometer," in *Proc. SPIE Vol. 3350, p. 424-431, Astronomical Interferometry, Robert D. Reasenberg; Ed.*, **3350**, pp. 424–431, July 1998.