A flexible stand-alone testbench for characterizing the front-end electronics for the CMS Preshower detector under LHC-like timing conditions

Paul Aspell[1], **David Barney[1]**, Yves Beaumont[1], Suhas Borkar[2], Aruna Borkar[2], Jacques Domeniconi[1], David Futyan[1], Apollo Go[3], Suresh Lalwani[2], Carmen Palomares[1], Remi Prunier[4], Serge Reynaud[1]

[1]CERN, 1211 Geneva 23, Switzerland; [2]Bhabha Atomic Research Centre, Mumbai, India;
[3]National Central University, Chung-Li, Taiwan, [4]French Cooperant at CERN
David.Barney@cern.ch

Abstract

The physics requirements of the CMS Preshower detector [1] demand front-end electronics with large dynamic range, to cope with the range of energies of incident electrons/photons, and low noise, to allow inter-strip calibration with minimum ionising particles (mips). Data from the Preshower are only read-out upon reception of a 1st level trigger, necessitating the inclusion of a pipeline memory. The complexity of the Preshower front-end ("PACE") [2] thus rivals that of the SCT and APV chips used respectively for the ATLAS and CMS trackers, and the testing of the chip, both in terms of design verification and volume production is a complicated task.

A flexible testbench has thus been designed in order to facilitate a large range of tests of PACE, both digital and analogue. This paper describes the implementation of the testbench and its use to validate the design of the first full version of PACE, called PACE-2.

I. PACE-2

PACE-2 is an assembly of two chips, called Delta and PACE-AM, designed in DMILL 0.8µm BiCMOS technology. The separation into two chips minimizes crosstalk of digital signals (in the PACE-AM) into the sensitive pre-amp stage (Delta).

The Delta chip is a 32-channel pre-amplifier and shaper, DC-coupled to a 32-strip silicon sensor. It incorporates leakage current compensation and a switched-gain shaper allowing two modes of operation: low-gain for normal physics running (dynamic range 0-400 mips); high gain for calibration purposes (dynamic range 0-50 mips). Programmable biasing and the amplitude of an internal calibration pulse are provided by means of on-chip DACs.

The PACE-AM is essentially a 32-channel analogue pipeline with 160 cells per channel (i.e. the memory depth is a maximum of 4µsec). An interface compatible with the Philips I^{2}C standard[3] is used for the programmable biasing of the PACE-AM via DACs, and an interface between PACE-AM and Delta allows the Delta DACs to be programmed. LVDS inputs provide the 40 MHz clock and several other timing signals, including the LV1 (1st level trigger) and ReSynch (re-synchronizes the pointers in PACE-AM). Upon reception of an LV1, 3 consecutive columns in the PACE-AM memory are blocked and the column addresses written to a 24-deep FIFO, the analogue data being multiplexed out at 20 MHz.

The basic architecture of PACE-2 is shown in Figure 1.

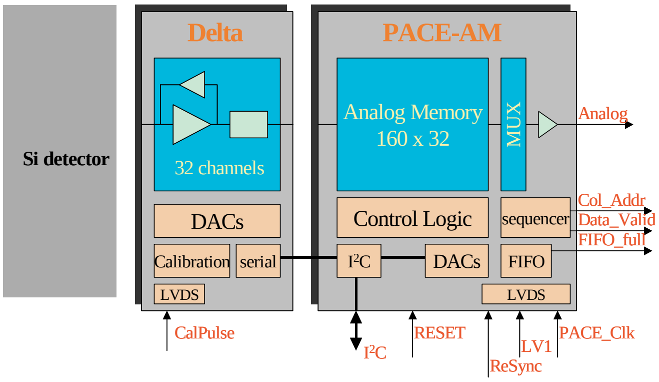


Figure 1: Block diagram showing the main features of the PACE-2

Samples of PACE-2 arrived at CERN in the summer of 2001 and were bonded to PCB hybrids for testing. Figure 2 shows a photograph of a PACE-2 hybrid attached to a silicon sensor.

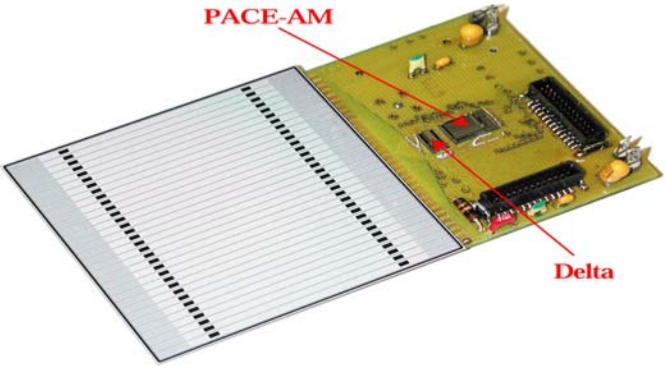


Figure 2: Photograph showing a silicon sensor (32 strips) bonded to a hybrid containing the PACE-2 assembly

One thing to note is that PACE-2 was designed with testability in mind, including the possibility of scan-chain operations. In fact a majority of the outputs from PACE-AM are for test purposes only.

II. TESTBENCH

A. Requirements

The primary objective of the testbench was to perform design verification of PACE-2 so the emphasis was placed on flexibility. There are many digital functions to test, as well as the analogue performance. The testbench must be capable of tracing any possible problems. In addition, the system must simulate the LHC running conditions – i.e. the 40 MHz clock and fast control signals. An important consideration was the possibility to send programmable bursts of triggers in order to test different functionalities and also to examine the response of PACE-2 to various trigger rates – including pseudo-random triggers generated with an LHC-like Poisson distribution.

Interfaces to PACE-2, both LVDS and I^{2}C must be incorporated, and the system should also incorporate its own DAQ system, with subsequent data output to a PC via a simple interface (that should also be used for user programming of the Delta/PACE-AM etc.).

Multiple copies of the testbench were to be produced (for different laboratories, beam tests etc.). The system was therefore designed to be cheap and relatively simple in design. We did not want to use “crate” electronics, but produce a self-contained system requiring simply a PC for user interaction and data storage via a single RS232 link and an oscilloscope for analogue signal observation.

The final requirement was that the system should be viewed as a prototype for a full production test system for the evaluation of ~7000 PACE assemblies.

B. Implementation

The testbench [4] is based around two principle components: an FPGA (Altera Flex 10k [5]) providing the fast timing and control signals, and a microcontroller (Mitsubishi M16C [6]) to provide the user interface to the PACE-2 and the FPGA. The principle characteristics of the M16C are:

- 20k bytes RAM (for stack)
- 250k bytes FLASH RAM (for programs)
- 8 channels of 10-bit ADCs
- 5 serial I/O channels (inc. RS232 and I^{2}C)
- 87 programmable I/O pins
- Multifunction 16-bit timers (6 input, 5 output)

The FPGA was mounted on a custom-built PCB (the “motherboard”) together with some auxiliary components. These included an Analog Devices AD9042 12-bit 40 MHz ADC [7] (for continuous sampling of the analogue signals from the PACE-AM), ADC FIFOs to read the digital data from the ADC (controlled by the FPGA), and delay line chips to allow fine-tuning of the ADC sampling point and the timing of the calibration pulse sent by the Delta. A piggyback board (the “M16C board”) carried the microcontroller and an RS232 interface for PC communication. The piggyback arrangement facilitated testing of the motherboard (and the program on the M16C) using an emulator, and also meant that the M16C board could be used for other purposes[1]. Figure 3 shows a photograph of the motherboard, complete with the M16C board and a PACE-2 hybrid plugged-in. The floating connectors seen on the upper-right of the picture are routed to the ADCs on the M16C. They connect to the PACE-2 hybrid to allow, amongst other things, measurements of the output of the on-board DACs (that provide biasing etc. for the Delta and PACE-AM).

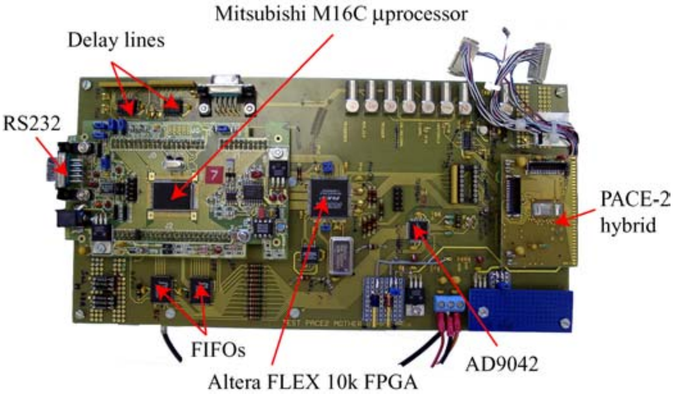


Figure 3: Photograph of the testbench, comprising the motherboard, M16C piggyback and the PACE-2 hybrid

C. Programming environment

A large part of the flexibility of the system comes from the rather low-level tasks programmed into the M16C. Essentially the user sends a simple command to the M16C via the PC in the form of a text string. The format of the text string is:

s<task ID><bytecount><parameters>

Currently there are 20 tasks available, summarised in Table 1. A user interface has been built using the LabView 6i application, but a low-level text interface also exists for debugging purposes.

Task ID	Function	Task ID	Function
0	Scan I^{2}C address	11	Read ADC FIFO status
1	Write to I^{2}C	12	Read 1 trigger block
2	Read from I^{2}C	13	Write to ADC
3	Write to FPGA	14	Read software version
4	Read from FPGA	15	Toggle echo
5	Set one bit in FPGA	16	Set echo to ON
6	Clear one bit in FPGA	17	Set echo to OFF
7	Toggle one bit in FPGA	18	Clear M16C circular buffer
8	Write to delay line	19	Toggle offset to ASCII
9	Read from delay line	20	Set RS232 baud rate
10	General reset	21-25	Various tasks for Poisson-distributed random trigger generation

Table 1: Programmable tasks in the M16C

---

[1] For example, the M16C board has been used by the CMS Preshower group to test several components of the CMS Tracker control system, making use of its I^{2}C interface.

D. Modes of Operation

There are currently three different modes of operation for the testbench:

- System as a “Master” with internal (inside FPGA) trigger generation
    - This is the normal mode of operation for design evaluation
- System as a “Master” with external trigger generation
    - Useful for testing the system with a silicon sensor attached, stimulated by an IR laser or radioactive source, or indeed in a beam test
- System as a “Slave” with external trigger generation
    - Useful to synchronize two motherboards together (e.g. in a beam test)

E. Internal Triggering Modes

The Flex FPGA can be programmed to generate bursts of up to 16 triggers (“CalBurst”) by specifying the time, expressed as a number of clocks (up to 65535), between consecutive triggers. Each of these so-called “triggers” is in fact a sequence of two pulses:

- Calib – telling the Delta to generate an internal electronic injection signal, sent to a selection of channels specified by the user
- LV1 – sent <latency> clocks after Calib to tell PACE-AM to block a sequence of 3 columns in memory where the signal is being held.

In fact these bursts of triggers may occur either at a specified time after a ReSynch signal, allowing the user to select a particular region of the memory, or “on demand” by the user. Indeed the ReSynch signals may also be either single-shot or free running, with a user-defined period. Table 2 summarises the modes of triggering.

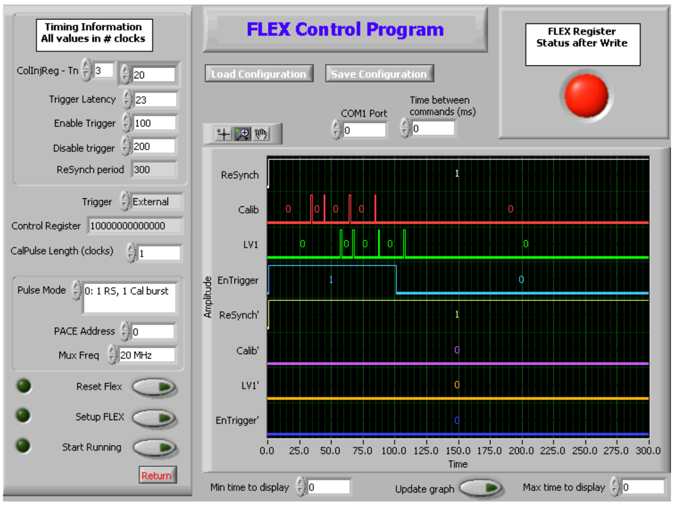
Mode	Description
0	Single ReSynch, single CalBurst
1	Free running ReSynch (user-specified period), each followed by a single CalBurst
2	Single ReSynch, free running CalBursts (user-specified CalBurst period)
3	Free running ReSynch, free running CalBursts
4	Single ReSynch, CalBurst on demand (by toggling one bit in Flex control register)
5	Free running ReSynch, CalBurst on demand

Table 2: Internal triggering modes

III. USER INTERFACE

User interfaces to control the Flex FPGA parameters (via the M16C), Delta and PACE-AM, were provided with LabView™. For the Flex FPGA it was found useful to include a graphical display of the basic timing signals sent to PACE-2: the ReSynch, Calib, LV1 and EnableTrigger signals. The display is updated on-the-fly so it is easy to see potential problems (LV1s arriving outside of the EnableTrigger etc.). When the parameters are sent to the Flex[2], they are automatically read back and verified. Figure 4 shows the front panel of the user interface to the Flex FPGA.

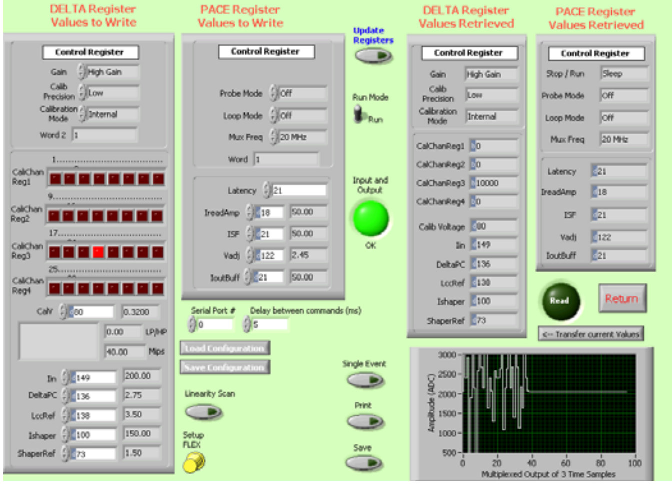
Control of the Delta and PACE-AM is achieved through registers on the chips, programmed via I^{2}C from the M16C. These registers control the modes of operation, set the biasing conditions, set-up the channels for the charge injection and also switch on/off probe pads for testing purposes. As with the Flex FPGA, after sending parameters to the Delta/PACE-AM, the data are automatically read-back from the registers and verified.



ColInjReg - Tn 3 20
Trigger Latency 23
Enable Trigger 100
Disable trigger 200
ReSynch period 300
Trigger External
Control Register 10000000000000
CalPulse Length (clocks) 1
Pulse Mode 0: 1 RS, 1 Cal burst
PACE Address 0
Mux Freq 20 MHz
COM1 Port 0
Time between commands (ms) 0
Min time to display 0
Max time to display 0

Figure 4: Front panel used to control the Flex. The user can change the basic timing signals, the mode of operation etc.

The front-panel for the control of the Delta/PACE-AM is shown in figure 5.



Gain High Gain
Calib Precision Low
Calibration Mode Internal
Word 2 1
CalV 80 0.3200
0.00 LP/HP
40.00 Mips
Iin 149 200.00
DeltaPC 136 2.75
LccRef 138 3.50
Ishaper 100 150.00
ShaperRef 73 1.50

Probe Mode Off
Loop Mode Off
Mux Freq 20 MHz
Word 1
Latency 21
IreadAmp 18 50.00
ISF 21 50.00
Vadj 122 2.45
IoutBuff 21 50.00
Serial Port # 0
Delay between commands (ms) 5

Gain High Gain
Calib Precision Low
Calibration Mode Internal
CalChanReg1 0
CalChanReg2 0
CalChanReg3 10000
CalChanReg4 0
Calib Voltage 80
Iin 149
DeltaPC 136
LccRef 138
Ishaper 100
ShaperRef 73

Stop / Run Sleep
Probe Mode Off
Loop Mode Off
Mux Freq 20 MHz
Latency 21
IreadAmp 18
ISF 21
Vadj 122
IoutBuff 21

Figure 5: Front panel used to set the registers in the Delta and PACE-AM. The user may also send a single trigger and read the output from PACE to see the effect of changing conditions etc.

---

[2] In fact the parameters are sent to the M16C on the RS232 line, using commands as mentioned in section II.C, and then sent by the M16C to the Flex FPGA.

IV. DESIGN VERIFICATION

The first part of the design verification involved testing the basic digital functionality, such as controlling the Delta/PACE-AM registers via I^{2}C. Indeed at power-on, the registers are automatically set to default values, and the user can simply read these values into LabView. After communication and control was verified, the digital outputs from PACE-AM were switched-on (“probe mode”) and triggers sent. An oscilloscope was then used to view these output digital signals – such as the ADC FIFO empty flag, the data valid signal, various multiplexer signals, and loop signals (signifying the number of columns in the memory that are currently blocked). It is also possible to switch-on a “scan mode” in PACE to test the digital blocks.

An example of the usefulness of the ability to send multiple triggers with our test system was the fact that we could extensively test the skipping mechanism in the memory of PACE-AM with the following procedure:

- Send *N* consecutive triggers to block a complete region in memory
- Send one further trigger that should straddle the blocked region
- Test to see if the memory addresses read-out are ok

This process was repeated for values of *N* between 1 and 7, before and after irradiation with X-rays. Simulations had shown that the skipping mechanism should cope with skipping 18 consecutive cells before irradiation, and this was indeed verified.

After the basic digital functionality had been verified, the on-chip DACs were calibrated. Each DAC has an analogue output pad, for testing purposes, attached to connectors on the PACE-2 hybrid. These outputs were fed into the M16C on-board ADCs and the subsequent digital values acquired with a LabView program. The appropriate DAC values for biasing could then be determined and set.

The analogue output was then tested, by switching-on the internal charge generation in the Delta and sending a trigger from the Flex. The multiplexed analogue output of PACE-AM was initially examined using an oscilloscope. This analogue output is sampled by the AD9042 and the resulting digital values sent to the FIFOs on the motherboard and subsequently to the M16C where they can be read by another LabView routine. Figure 6 shows a typical multiplexed digital signal seen with LabView, after a charge was sent into channel 11.

After verifying that the charge injection was seen in each of the 32 channels, the analogue performance was then examined, in terms of gain uniformity between channels, pedestal uniformity (and noise) through the memory and between channels, and the dynamic range (in both gains).

The analogue signal shape could also be measured, using the delay lines on the motherboard to fine-tune (in steps of 250ps) the time at which the charge injection was sent. As the PACE-AM produces three samples at 25ns intervals for each trigger received, all three samples could be used to reproduce the signal shape, as illustrated in Figure 7.

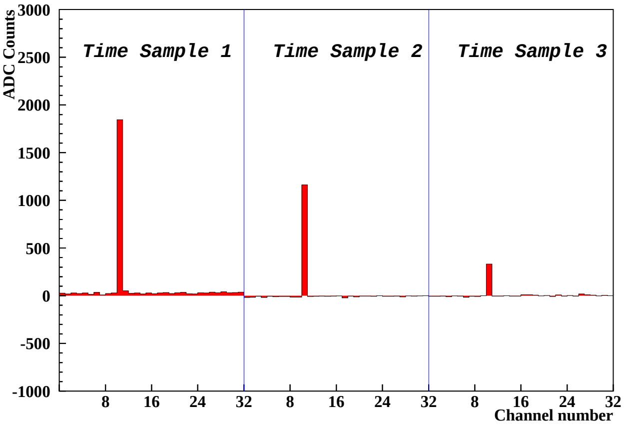


Figure 6: Multiplexed output from PACE, showing the three time samples, with a charge injection sent to channel 11. Pedestals have been subtracted. Note that in “real life” the latency will be adjusted so that the first sample measures the baseline, whilst the second and third measure the signal.

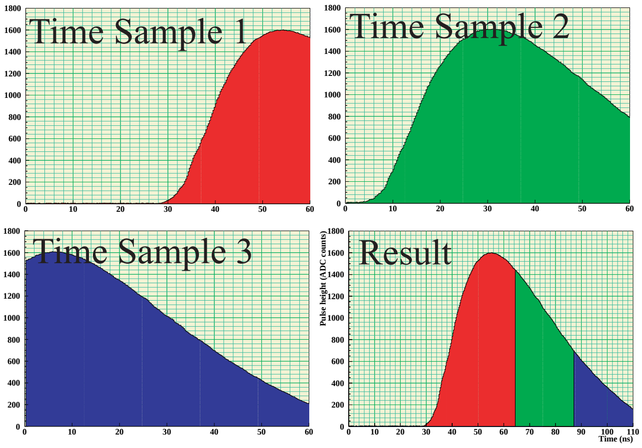


Figure 7: The “timing scan” used to reproduce the signal shape from the Delta.

V. PRODUCTION TESTING

On the basis of the measurements necessary for the design verification we developed a systematic testing procedure suitable for non-expert users. Each chip is assigned a bar code, as are the hybrids containing the Delta and PACE-AM. The hybrid code is entered by the user when starting to test a PACE-2 assembly, and is used to create a directory on the PC and subsequent data files. The user is then led through a series of tests, each of which produces a file in spreadsheet format and an indication as to whether the test was successful or not. Clearly during production testing if the result of any functionality test is bad, the remaining tests are skipped. The tests performed systematically are:

- Startup: load registers in Delta and PACE-AM with default values
- Scan mode
- Check all possible I^{2}C addresses on the PACE-AM

- Calibrate the DACs, then load optimum values
- Check charge injection into each channel
- Perform a dynamic range scan (two gains, two calibration precisions)
- Measure the pulse shape from the Delta, in both gains
- Make a pedestal scan through the memory (also measures noise in each cell etc.)

When all tests are finished, the user can send the data files to the main database (based upon the CRISTAL [8] package). The full testing of a single assembly takes approximately ten minutes and as of August 2002 we have measured approximately 100 PACE-2 assemblies. The limiting factors for the testing time are the RS232 communication speed (115kbits/sec) and the settling time necessary for some tests. A future version of the testbench may use USB communication and we estimate that in this case the full set of measurements will take approximately two minutes.

VI. OTHER USES OF THE TESTBENCH

In addition to the verification of the design of PACE-2, the testbench has been used to look at signals from Preshower silicon sensors bonded to PACE-2 assemblies, stimulating the sensors with an infrared laser or a radioactive source.

In May 2002 we used two such silicon sensors, each bonded to a PACE-2 assembly and read-out/controlled with an individual testbench, in a pion/proton beam at the Paul Scherrer Institut. The ability of the testbench to operate either as a master or a slave, coupled with the capability to accept external triggers (from scintillators) facilitated this beam test, even though the readout rate was limited to about 20 Hz due to the RS232 communication speed and the size of the ADC FIFOs on the motherboard. Figure 8 shows a photograph of the setup.

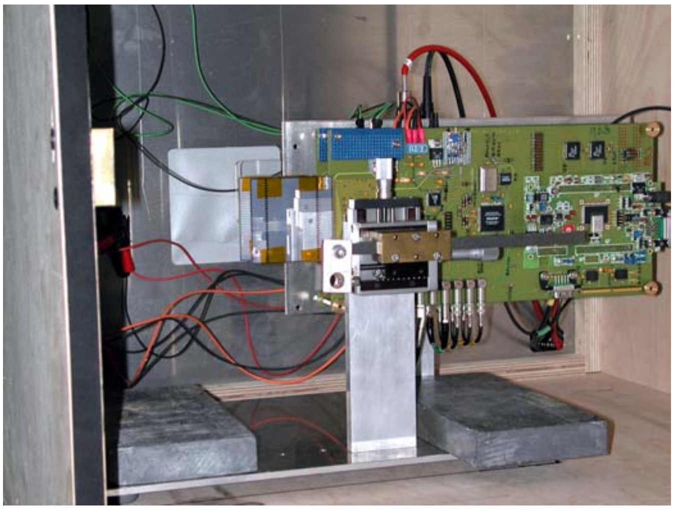


Figure 8: Photograph of two testbenches connected together, each equipped with a PACE-2 assembly bonded to a preshower silicon sensor.

The small size of the testbench meant that we could also place it directly in an X-ray irradiation facility and monitor the performance of PACE-2 in-situ.

The flexibility of our system means that it could, in principle, be adapted in order to characterize/evaluate other mixed-signal chips. It would simply require some program adaptation and possibly a change in some connectors.

VII. SUMMARY AND CONCLUSIONS

A low-cost standalone testbench has been developed for the evaluation of the CMS Preshower front-end electronics (PACE). The use of a microcontroller coupled to an FPGA results in an extremely flexible system that has allowed us to test all of the digital functionality of PACE-2 and measure its analogue performance. A user interface and systematic testing procedure based upon LabView 6i has been developed and used to characterize approximately 100 PACE-2 assemblies, allowing us to see chip-to-chip variations etc. So far three copies of the testbench have been produced, allowing us to characterize chips before and after irradiation, with and without silicon sensors attached, and even in a particle beam.

We envisage that the next version of the testbench will utilize a higher-speed communication to the PC, allowing it to form the basis of the production test facility for the final version of PACE.

VIII. REFERENCES

[1] P. Wertelaers *et al*, "*ECAL Preshower Engineering Design Review*", CMS ECAL EDR-4 (see http://cmsdoc.cern.ch/cms/ECAL/preshower/Documents/EDR2000/FullEDR.pdf)

[2] P. Aspell, "*The Design and Development of the Front-end Electronics for the CMS Preshower Detector*", PhD thesis (see http://cmsdoc.cern.ch/cms/ECAL/preshower/Documents/internal/PACEIIthesis/PACEIIthesis.pdf)

[3] Philips Semiconductors, "*The I^2C Bus Specification, Version 2.0*", Philips publication December 1998

[4] A user manual and schematics for the testbench may be found on the Preshower web site (http://cmsdoc.cern.ch/cms/ECAL/preshower) under the "electronics→Chips→PACE II" section.

[5] For details concerning the Flex 10k FPGA, consult: http://www.altera.com/products/devices/flex10k/f10-index.html

[6] See, for example, http://www.mitsubishichips.com/products/mcu/products/m16c

[7] See, for example, http://products.analog.com/products/info.asp?product=AD9042

[8] J-M. Le Goff *et al*, "*Detector Construction Management and Quality Control: Establishing and Using a CRISTAL System*", CMS Note-1998/033