MS
11/2011

2011 PXL Latchup Tests

Contact persons:
Michal Szelezniak
Leo Greiner
Xiangming Sun

Test goals 1
Tested devices 2
Test setup 3
  Hardware 3
  Ion beams 5
LU test results 7
SEU test results 8
Ultimate sensor test results 9
  Estimated consequences for the PXL detector 10
Notes and observations 11
Appendix A 12
Appendix B 13

Test goals

The goal of these tests was twofold:

- to help assess the need for designing new, latchup resistant digital cells for use in the sensors for the PXL detector,
- to establish latchup sensitivity of the PXL electronics components that are planned to be located in high and moderate radiation areas in STAR (sensor ladders, MTB)

The tests were performed in two runs at the BASE Facility[1] at the 88" cyclotron, LBNL. The first run took place on Nov. 13-14, 2010 and was limited to latchup (LU) tests of Mimosa26, a dedicated latchup test structure from IPHC, and all PXL electronics components. The second run took place on Aug. 26 and 28, 2011, and expanded our testing program by adding a PXL pre-production sensor (Ultimate) and allowing for measurements of single event upset (SEU).

This report focuses on the more extensive set of measurements from the second run. Test results from the first run are presented in Appendix A.

[1] http://user88.lbl.gov/base-rad-effects

Tested devices

We have tested the following MAPS sensors and commercially available ICs:

- Mimosa26 – predecessor of the PXL sensor featuring half-reticle size and on-chip zero suppression
- Mimosa28 (Ultimate) – pre-production prototype of the PXL sensor
- LU test structure – a dedicated chip composed of 4 shift registers of 64 D flip-flops with various NMOS-PMOS spacing (1.5 µm, 3.5 µm, 6.5 µm)

PXL readout components that are planed to be located in the high radiation area (on ladders) include:

- FIN1108 Fairchild LVDS 8 Port Repeater
- SN74LVC126A TI Quad Bus buffer gate
- SN65LVDS2 TI Single LVDS receiver

Components located on the MTB include:

- FIN1108 Fairchild LVDS 8 Port Repeater
- SN65LVDT14 TI Interconnect Extender Chipset w/ LVDS
- SN65LVDT41 TI Interconnect Extender Chipset w/ LVDS
- AD7997 Analog Devices 8-ch, 10-bit I2C ADC
- CY7B9950 Cypress 200MHz PLL Clock Buffer
- TPS75801 Voltage Regulator
- AD626 Analog Devices single supply diff. amplifier
- AD8611 Analog Devices single supply comparator
- polyfuse

AD8611, AD626, TPS75801, and polyfuse, were tested at 5.0 V. All other components were tested at a 3.3 V power supply.

Most of the above devices were studied for latchup cross section and SEU. A small subset of these devices is not compatible with SEU tests or does not need to be studied in this context. This includes AD626, MIC37152, polyfuse, and AD7997 (occasionally corrupted data in the temperature monitoring system is not expected to cause any substantial problems in the PXL detector system).

Test setup

Hardware

We have designed a dedicated test PCB[2] that allowed us to test almost all of our test devices at the same exposure time (Figure 1). The test card has the size of approximately 5"x 8.5" with all DUTs located in the central region, in a circle with a diameter of 1.9". This arrangement allows us to test all of our test devices in parallel and to reduce the number of beam/ion changes required to accomplish our test plan. Auxiliary buffers (mostly SE<->LVDS converters) are located further out, closer to the flat-cable connectors. The interface to power supplies and the LU monitoring system is provided via four, 50-pin headers.

The Ultimate prototype was tested separately, on a test PCB dedicated to individual sensor tests[3].

In order to monitor latchup events on all devices at the same time, we used a set of LU-monitoring power supply modules that were developed for the PXL detector[4]. Each module provides two independently latchup-monitored power supply channels. Ten of such modules were attached to a dedicated PCB that interfaced 20 power supply monitoring nodes to a prototype PXL readout board (see Figure 2).

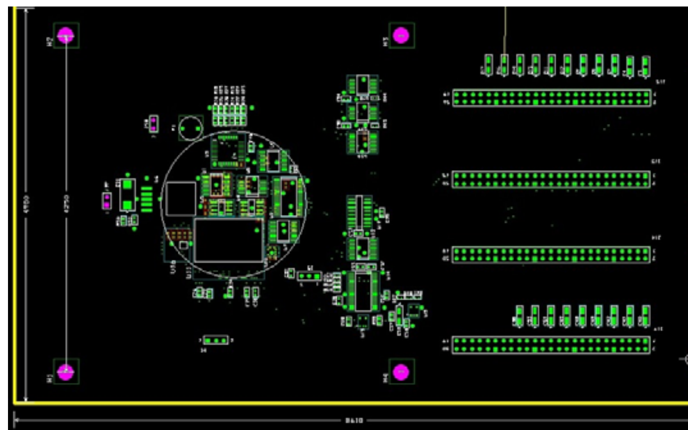

Figure 1 Test PCB designed for LU and SEU measurements of multiple devices.

[2] PCB schematic: http://rnc.lbl.gov/hft/hardware/docs/latchup/LATCHUP_2010_schematic.pdf
[3] Design files: http://rnc.lbl.gov/hft/hardware/docs/ultimate/ultimate_individual_test_board.zip
[4] PCB schematic: http://rnc.lbl.gov/hft/hardware/docs/latchup/LU_POWER_PCB.pdf

Figure 2 A set of 10 LU-monitoring power supply modules that were built for the PXL detector development is shown attached to a dedicated support PCB; there are five modules on each side of the PCB. Each module provides two latchup-monitored, independent power supply channels.

SEU measurements were performed by providing a CLK-like test signal at 40 MHz to each DUT and comparing it with the output signals from the tested devices. The sequence monitoring and error counting were implemented in the prototype PXL readout board based on a Virtex-5 FPGA.

The PXL readout board operated with a dedicated firmware that allowed for automated monitoring and recording of latchup events, SEU events, and JTAG programming for MAPS sensors. The complete suite of measurements was controlled with a PC-run script, attached in Appendix B.

The system was designed to allow for masking frequently latching channels so that studies on more latchup resistant devices could be performed.

An example of the test setup at BASE is shown in **Error! Reference source not found.**.

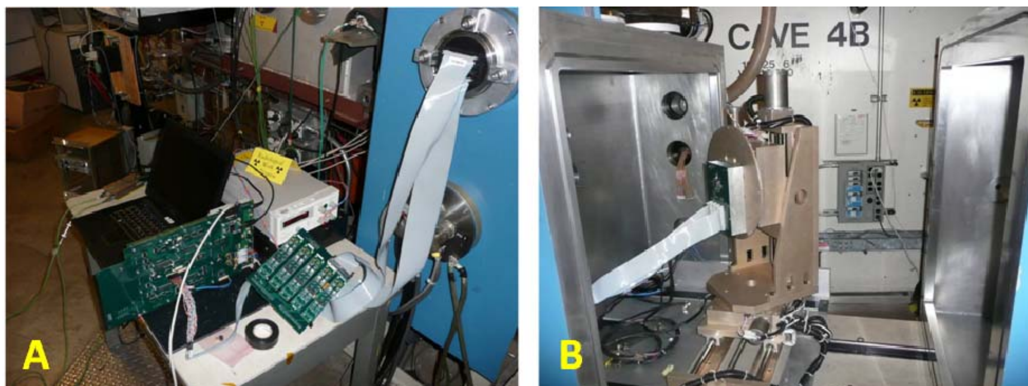

Figure 3 LU and SEU test setup in the configuration used in the first run at BASE (2010). (A) - the PXL RDO board and the 20-channel power-supply LU-monitoring unit. (B) - test PCB mounted to a fixture in the vacuum chamber.

Ion beams

The features of the BASE facility at LBNL include:

- Switching between beams typically takes 5-10 minutes
- Changing cocktails takes approximately 4 hrs.
- The operating vacuum in the test chamber is reached in about 5 min.

Throughout the tests described in this report we used two standard cocktails with E/A = 10, 16 MeV/nucleon.
Table 1 summarizes the test beams used for LU and SEU measurements. To reach LET of 50 MeV/mg/cm^2, we used the Kr beam with an incident angle of 60°.

Ion	Cocktail	LET 0º	Range in Si
	(MeV/nuc)	(MeV/(mg/cm2))	(µm)
B	**10**	**0.89**	305.7
O	**10**	**2.19**	226.4
Ne	**10**	**3.49**	174.6
Ar	**10**	**9.74**	130.1
N	**16**	**1.16**	505.9
O	**16**	**1.54**	462.4
Ne	**16**	**2.39**	347.9
Si	**16**	**4.56**	274.3
Cl	**16**	**6.61**	233.6
V	**16**	**10.9**	225.8
Kr	**16**	**24.98**	165.4

Table 1 Test beams with different LET used for LU and SEU measurements described in this report. More ions and cocktail energies are available at the BASE Facility.

Our preferred cocktail is 16 MeV/mg/cm^2 due to the larger penetration range in silicon, making it more suitable for testing of the PXL standard IC devices that are embedded in standard packages. However, we also took data at 10 MeV/mg/cm^2 since this cocktail was available at the beginning of our test.

To estimate the thickness of the IC packages, we have milled off the top layers of the IC packages of the devices included in our tests, as shown in Figure 4. We have learned that in many cases the silicon chip is buried 10 mils (250 um) under the package surface. The exceptions here were AD7997, AD626, AD8611, LVC126A with package thicknesses of 15 mils, and the TPS75801 at 118 mils.

We have been unable to find detailed information about the composition of IC packaging, but it appears reasonable to assume that the material density is approximately half of that of silicon

We have decided to keep the thickness of the IC package layer above the silicon to approximately 10 mils (125 um silicon equivalent), this should allow us to explore a reasonable range of LET with the ion species listed above. For this reason we have thinned the thicker packages to 10 mils, except for the regulator, which apparently has internal wire bonds extending into the whole volume the package.

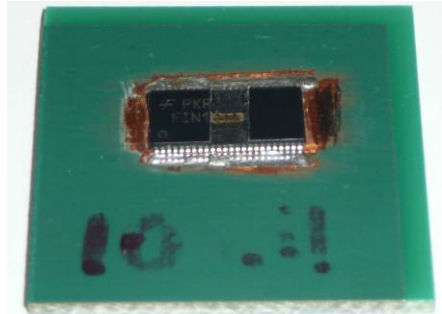

Figure 4 FIN1108 package thinned to the silicon die. We had to remove approximately 10 mils of package before we reached the die.

LU test results

The summary of LU measurements for Mimosa26 and the PXL standard ICs is presented in Figure 5. Latchup in the Ultimate sensor is discussed later. The plot is limited to devices that exhibited LU during measurements. Other devices did not show any latchup sensitivity in the range of LET values used. For measurements at high LET values, monitoring of frequently latching devices was disabled.

Multiple measurements that were performed at the same LET values are shown in Figure 5 individually to highlight run-to-run variations.

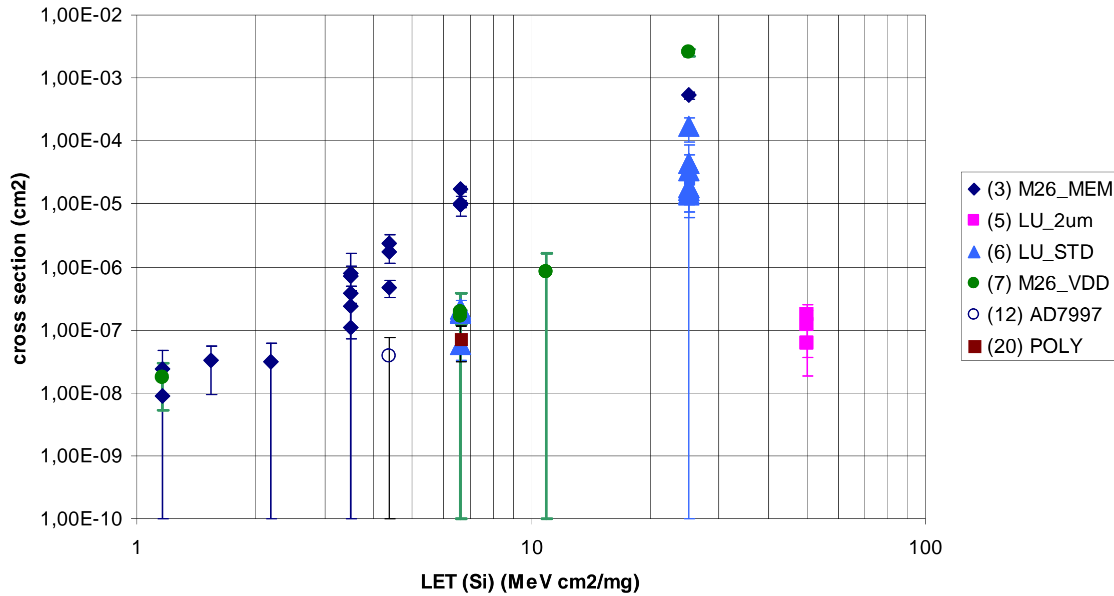

Figure 5 Summary of LU measurements for Mimosa26 and standard ICs to be used in the PXL detector.

Observations:

- Latchup in Mimosa26 is dominated by events in the memory section of the chip.
- Single LU occurrence in AD7997 at 4.35 MeV cm^2/mg, but no latchup events at higher LET values.
- Single event in the polyfuse (monitored for open circuit instead of short).
- 2 µm separation of NMOS and PMOS transistors in the IPHC LU-test structure significantly decreases LU sensitivity.
- The onset of LU events in Mimosa26 is at above 1 MeV cm^2/mg for cross sections larger than 1e-8.

SEU test results

Two Mimosa26 chips with different thickness of the epitaxial layer (15 and 20 µm) were used in the presented studies. Figure 6 shows measured cross sections for two Mimosa26 and two Ultimate prototypes. Not all of these devices were tested with the same ion beams.

The limited overlap in measurements of the two Mimosa26 prototypes is not sufficient to clearly distinguish possible differences in the measured cross-sections. However, it should be noticed that the data points align well on the same line.
SEU cross sections measured in Mimosa26 are noticeably larger than LU cross sections in this chip.

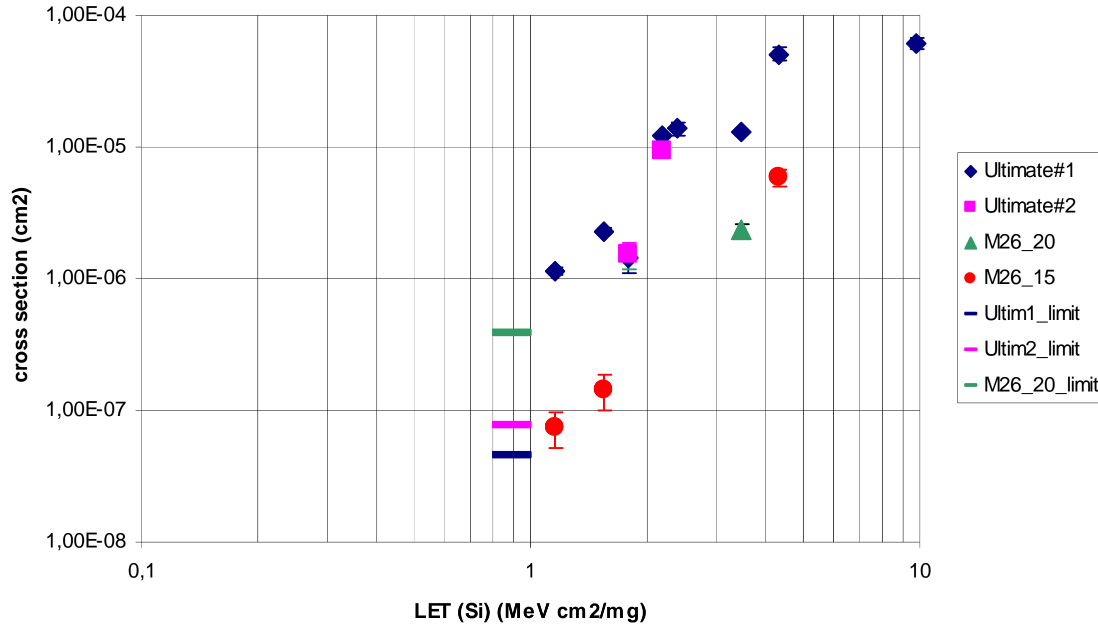

Figure 6 SEU cross section measured on two Mimosa26 and two Ultimate prototypes. Horizontal bars represent measurement limits.

SEU cross sections for other tested devices are shown in Figure 7. The figure contains data for all of the devices that exhibited data corruption. This is limited to the LU-test structure and 2 channels of the 4-channel PLL Clock buffer.

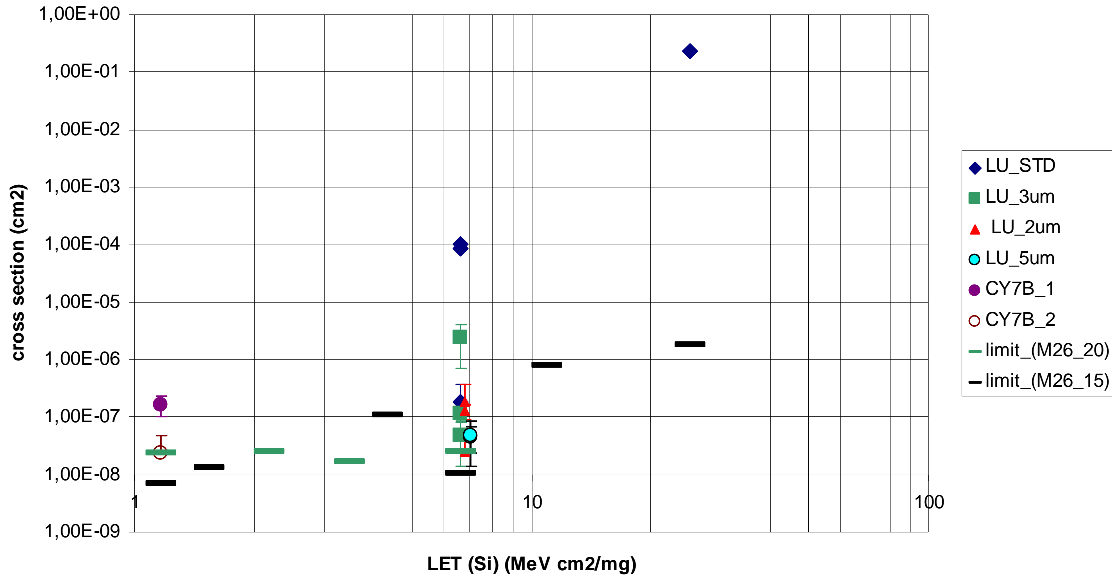

Figure 7 SEU cross section measured for the IPHC LU-test structure and the CY7B9950 PLL clock buffer. Other devices did not exhibit any SEU effects. Horizontal bars represent measurement limits and allow to distinguish between two PCBs hosting different Mimosa26 chips.

Observations:

- A small number of data corruptions in the PLL occurred just above 1 MeV cm^2/mg, but has not been observed for other ion beams. These errors were observed only on one of the two PCBs that were used during tests (M26_20).
- Presence of data errors in all registers of the LU-test structure clearly demonstrates that the mechanism for SEU is different than for LU.

Ultimate sensor test results

The Ultimate sensor was tested on the individual sensor test card. In this configuration, the sensor had two power supply lines that were monitored for latchup: VDD and VDA. Our data acquisition system registered no latchup events on the VDA power supply. The measured latchup cross section must be dominated by events in the memory section of the chip, as it is indicated in test results of the Mimosa26 chip.

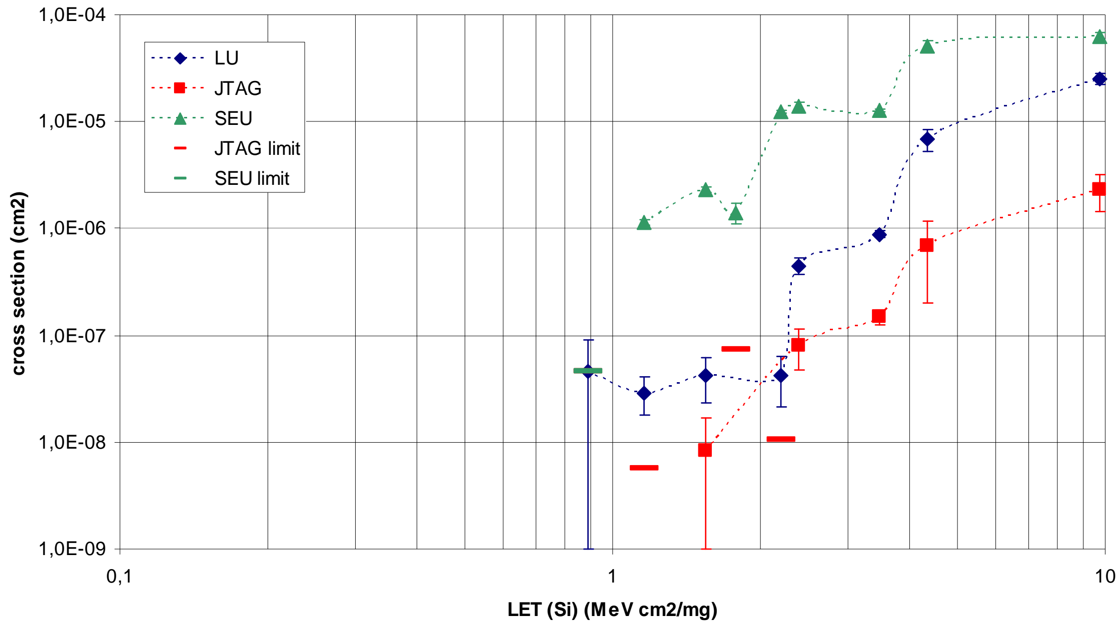

Figure 8 Summary of the Ultimate sensor test results: cross sections for latchup, JTAG errors, and single event upsets in the readout section of the chip. Horizontal bars represent measurement limits when no events were registered.

Test results presented in Figure 8 show that the onset of all single event effects is near the LET value of 1 MeV cm^2/mg. The SEU cross section is approximately 30 times larger than the latchup cross section. The cross section for SEU errors in the sensor's configuration registers (JTAG) is approximately 20% of the LU cross section.

Estimated consequences for the PXL detector

Based on the above results, for every latchup in the PXL detector, we can expect 30 corrupted words in the sensor output stream. At the estimated latchup frequency in the PXL detector of approximately one LU event every 2-3 hrs[5], we can expect this data corruption rate to have a negligible effect on the detection efficiency of the detector.

At the same time, we can expect a corruption in the sensor JTAG settings to occur on average every 10 hrs. The impact of such errors on the detector performance needs to be evaluated and we need to consider implementing an appropriate configuration refreshing cycle in the detector system.

[5] Summary of LU measurements in the STAR environment:
http://rnc.lbl.gov/hft/hardware/docs/latchup/Latchup_in_PXL_DRAFT.doc

Notes and observations

- Our initial intention was to fully log data corruption patterns from the Mimosa26 and Ultimate sensors. We expected that single bit corruptions would indicate SEU effects in the memory block; while possible multi-bit/multi-word corruptions would originate in the zero suppression circuitry. Unfortunately, we did not discover on time a bug in the vhdl code that prevented proper acquisition of corrupted words. This bug did not affect SEU counting, and SEU cross sections for Mimosa26 and Ultimate can be calculated from the acquired data.

- When testing with the Kr ions; we started registering latchup events associated with the VD33 power supply that powered all auxiliary IC on the test PCB (buffers and SE-LVDS translators for certain signals). After a short investigation and shifting the center of the beam spot; we concluded that the LU events were originating in the 1-to-4 LVDS fanout buffer (SN65LVDS104). This buffer is the only device on the test PCB that is not among devices tested for LU effects, all of which showed no LU sensitivity in the tested LET range.

- LVC126 on one of the test boards was tested in a full thickness package.

- Running at the BASE fqcility:

    - Ethernet cable installed in the cave did not work; we had to bring a spare one.
    - If the "close" button on the door to cave B doesn't' work, it is possible that the door has been opened too wide. In this case, one can press the "open" button on the door to cave A.

Appendix A

Summary of latchup measurements: LBL BASE Nov.13, 14, 2010

Figure 9 shows a raw-data summary of the first series of tests at the BASE facility at LBL. Single LU occurrences are circled in red.

In this test, three sections of the Mimosa26 chip were independently powered and monitored for LU: memory, VDA, VDD. The lutchup events registered in the VDA channel are false – they exhibited false triggering and continuous latchup condition that could not be recovered from at a low LU-detection threshold. This problem was later recreated and studied in the lab environment. The problem was resolved by increasing the LU detection threshold on the VDA power supply.

LU events were also registered in two sections of the IPHC LU-test structure: the shift-register based on standard cells and the shift-register based on cells with an additional 2 μm separation between NMOS and PMOS transistors.

A single LU occurrence was registered in the AD7997.

The results agree with expectations based on the history of earlier LU measurements[6]. There is a very good agreement between these results and the results from the second run described in this write-up.

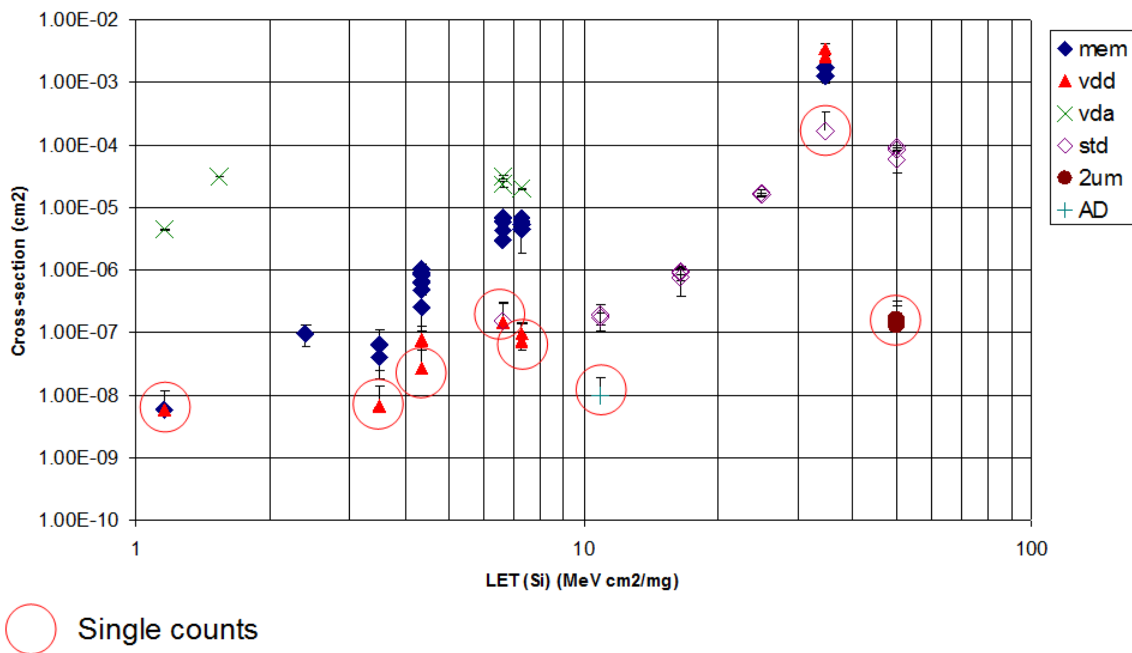

Figure 9 LU cross sections calculated based on raw data acquired during the first run at the BASE facility at LBL.

[6] http://rnc.lbl.gov/hft/hardware/docs/latchup/index.html

Appendix B

DAQ script

```
@echo off
setlocal EnableDelayedExpansion

IF "%1"=="" (
        echo ERROR: missing input parameter - run number
        goto end
)
REM                         1) seu_setup
REM                         2) latchup_setup
REM                         3) check_seu
REM                         4) seu_setup
REM                         5) check_seu
REM                         6) init_M26
REM                         7) :jtag2
REM                         8) :read
REM                                 - log M26 SEU errors if they are valid (not from LU or corrupted
JTAG)
REM                                 - if CNT<10 -> :read
REM                                 - if CNT>10 -> :read_latch
REM                                 - if SEU errors -> :read_latch
REM                         9) :read_latch
REM                                 - log latch data
REM                         - if latchup in M26 -> (log LU induced SEU errors) -> :jtag2
REM                         - if no erors in M26 -> :read_jtag
REM                     10) :read_jtag
REM                                 - if JTAG errors: log JTAG sequence -> (log JTAG induced SEU
errors) -> :jtag2
REM                                 - if no errors -> :read



REM seu_setup ^

command.exe -seu_start 0 -seu_mode 0 -no_default -no_reset_dcm
usb.exe -usb_send_file command.sun

command.exe -delay_file_generate 128 41 -reset_iodelay  -no_default
usb.exe -usb_send_file command.sun
command.exe -sleep 2

command.exe -delay_file delay_data.txt  -no_default
usb.exe -usb_send_file command.sun
command.exe -delay_file_generate 192 28   -no_default
usb.exe -usb_send_file command.sun
command.exe -delay_file delay_data.txt  -no_default
usb.exe -usb_send_file command.sun

command.exe -delay_file_generate 200 19   -no_default
usb.exe -usb_send_file command.sun
command.exe -delay_file delay_data.txt  -no_default
usb.exe -usb_send_file command.sun

command.exe -delay_file_generate 192 15   -no_default
usb.exe -usb_send_file command.sun
command.exe -delay_file delay_data.txt  -no_default
usb.exe -usb_send_file command.sun

command.exe -delay 20 9   -no_default
usb.exe -usb_send_file command.sun

command.exe -delay 0 1   -no_default
```

```
usb.exe -usb_send_file command.sun

command.exe -delay 0 2   -no_default
usb.exe -usb_send_file command.sun

command.exe -delay 0 3   -no_default
usb.exe -usb_send_file command.sun

command.exe -delay 20 4   -no_default
usb.exe -usb_send_file command.sun

command.exe -delay 32 5   -no_default
usb.exe -usb_send_file command.sun

command.exe -seu_start 1 -seu_mode 2 -led_page 1 -no_default  -seu_mask 0
REM -seu_mask 36 channels = ffffffff (MSB-LSB)
usb.exe -usb_send_file command.sun

command.exe -reset_frame -reset_usb  -reset_sram -m26_data 0 -m26_error 0 -m26_ref_update 0 -m26_ref_read 0 -no_default
usb.exe -usb_send_file command.sun
REM seu_setup v


REM latchup_setup ^

REM command.exe -usb_frame  -no_upload  -reset_usb -reset_frame -no_default
REM usb.exe -usb_send_file command.sun
REM command.exe -usb_frame -request_frame_latch_seu 2 -no_upload -no_reset_usb -measure_time_in_ms 10000000 -pull_time_in_clock 50 -latch_mask 0 -no_default
REM usb.exe -usb_send_file command.sun
REM command.exe -sleep 1 -no_default
REM command.exe -phase1_start 0 -no_upload -no_default -no_default
REM usb.exe -usb_send_file command.sun
REM command.exe -phase1_start 1 -no_upload -no_default
REM usb.exe -usb_send_file command.sun
REM command.exe -sleep 1 -no_default

command.exe -usb_sram  -no_upload  -reset_usb   -no_default
usb.exe -usb_send_file command.sun

command.exe -usb_sram -request_frame_latch_seu -no_upload -no_reset_usb -measure_time_in_ms 100000000 -pull_time_in_clock 50  -no_default
usb.exe -usb_send_file command.sun

command.exe -sleep 0.1

command.exe -phase1_start 0 -no_upload -no_default
usb.exe -usb_send_file command.sun
command.exe -phase1_start 1 -no_upload -no_default
usb.exe -usb_send_file command.sun

command.exe -sleep 1  -no_default

REM command.exe -sleep 60
REM latchup_setup v

set CNT=0
set UCNT=0
set run=run%1


echo 0x00000000 > error\!run!_latch.txt
echo 0x00000000 > error\!run!_m26_error.txt
echo 0x00000000 > error\!run!_m26_latchup_error.txt
echo 0x00000000 > error\!run!_m26_jtag_error.txt
echo 0x00000000 > error\!run!_jtag.txt

:check_seu
command.exe -no_default  -sleep 2
command.exe -no_default  -usb_frame
```

```
usb.exe -usb_send_file command.sun
usb.exe -usb_frame_recv 10000 frame.sun
command.exe -compare_file_size frame.sun  400

echo check_seu error_level %errorlevel%
        IF  errorlevel=2 (
                ECHO ERROR: SEU filled up the FIFO
                copy frame.sun error\!run!_seu_error.txt
                goto end
        )
        IF not errorlevel=2 (
                ECHO SEU is good
        )


:jtag
REM jtag_M26 ^
set root_file="M26_settings_2.txt"
JTAG_chain_programmer_M26.exe -jtag_trans %root_file%

call jtag_upload.bat
call jtag_start.bat
call jtag_upload.bat
call jtag_start.bat
call chip_start_usb_M26.bat

copy jtag.dat error\!run!_jtag.dat
REM jtag_M26 v

:init
REM init_M26 ^
command.exe -usb_sram  -request_sram_m26  -no_default -ddl_frame
usb.exe -usb_send_file command.sun
command.exe   -m26_data 1 -m26_error 1 -m26_ref_update 1 -m26_ref_read 1 -no_default
usb.exe -usb_send_file command.sun
command.exe  -reset_sram  -reset_usb
usb.exe -usb_send_file command.sun
command.exe -sleep 0.02 -no_default
usb.exe -usb_frame_recv 40000 rec.sun

copy rec.sun error\!run!_M26_init.dat

echo CNT=!CNT!
set /a CNT+=1

command.exe -check_m26_data rec.sun error.txt 40000

echo error_level %errorlevel%
set file=error.txt
set search_str=" 0 error.txt"

        IF exist !file! (
                DIR !file! |FIND !search_str!

                IF not errorlevel=1 (
                        ECHO !file! is good in setup ref for M26
                        copy rec.sun error\!run!_init.txt
                )
                IF errorlevel=1 (
                        ECHO !file! is  bad in setup ref for M26
                        copy rec.sun error\!run!_init_error.txt
                        goto end
                )
        )
        IF not exist !file! (
                echo ERROR: No such file! !file!
                goto end
        )
```

```
command.exe -usb_sram  -request_sram_m26  -no_default -ddl_frame
usb.exe -usb_send_file command.sun
command.exe   -m26_data 1 -m26_error 1 -m26_ref_update 0 -m26_ref_read 0 -no_default
REM -m26_ref_read 1  - compare frame cnt vs memory value        => always erors
REM -m26_ref_read 0  - compare frame cnt vs previous frame   => no errors
usb.exe -usb_send_file command.sun
command.exe  -reset_sram  -reset_usb
usb.exe -usb_send_file command.sun
command.exe -sleep 0.020 -no_default
usb.exe -usb_frame_recv 40000 rec.sun

command.exe -check_m26_data rec.sun error.txt 40000
echo error_level %errorlevel%

        IF not errorlevel=1 (
                ECHO !file! is good in init M26
                copy rec.sun error\!run!_init2.txt
        )
        IF errorlevel=1 (
                ECHO ERROR: !file! is  bad in init M26
                copy rec.sun error\!run!_init2_error.txt
                goto end
        )
        IF not exist !file! (
                echo ERROR: No such file! !file!
                goto end
        )
REM init_M26 v


:jtag2
echo IN: jtag2
REM jtag_M26 ^

set root_file="M26_settings_2.txt"
JTAG_chain_programmer_M26.exe -jtag_trans %root_file%
call jtag_upload.bat
call jtag_start.bat
call jtag_upload.bat
call jtag_start.bat
call chip_start_usb_M26.bat

REM jtag_M26 v


REM error readout


:read
echo IN: read
REM read_M26 ^
        IF exist m26_error.sun (
                command.exe -check_file_size m26_error.sun  0
                IF not errorlevel=1 (
                        echo m26_error.sun exist!
                        echo #!UCNT! >> error\!run!_m26_error.txt
                        copy /a error\!run!_m26_error.txt+m26_error.sun error\!run!_m26_error.txt
                )
        )

:mread
echo IN: mread
echo CNT=!CNT! UCNT=!UCNT!
set /a CNT+=1
set /a UCNT+=1

command.exe -usb_sram  -request_sram_m26  -no_default -ddl_frame
usb.exe -usb_send_file command.sun
command.exe   -m26_data 0 -m26_error 1 -m26_ref_update 0 -m26_ref_read 0 -no_default
usb.exe -usb_send_file command.sun
```

```
command.exe  -reset_sram  -reset_usb
usb.exe -usb_send_file command.sun
command.exe -sleep 1.5 -no_default
usb.exe -usb_frame_recv 1000 m26_error.sun
REM "C:\Program Files (x86)\Notepad++\notepad++.exe" rec.sun

REM read_M26 v

REM check_M26 ^

command.exe -check_file_size m26_error.sun  0
set file=rec.sun
set search_str=" 0 rec.sun"
echo  read M26 error_level %errorlevel%
		IF not exist !file! (
			echo ERROR: No such file! !file!
			goto end
		)
		IF !CNT! equ 10 (
			set /a CNT=0
		)
		IF  errorlevel=1 (
			ECHO !file! is good in read M26

			IF !CNT! equ 9 (
				goto read_latch
			)
			goto mread
		)
		IF not errorlevel=1 (
			ECHO !file! is bad in read M26
		)

REM check_M26 v

:read_latch
echo IN: read_latch
REM latchup_readout ^
command.exe -no_default  -request_sram_flag
usb.exe -usb_send_file command.sun
command.exe -no_default  -reset_sram -reset_usb
usb.exe -usb_send_file command.sun
command.exe -no_default  -usb_frame
usb.exe -usb_send_file command.sun
usb.exe -usb_frame_recv 10000 frame.sun

command.exe -check_latchup_channel frame.sun 3 2 6 7
REM 3 is how many channel to be check, followed by the channel number as 2 6 7 (vmem vdd and vda)
echo check_latchup_channel error_level %errorlevel%

		IF  errorlevel=1 (
			ECHO !file! there is latch up in M26
			echo #!UCNT! >> error\!run!_latch.txt
			copy /a error\!run!_latch.txt+frame.sun error\!run!_latch.txt
			REM save latchup corrupted M16 SEU data
			echo #!UCNT! >> error\!run!_m26_latchup_error.txt
			copy /a error\!run!_m26_latchup_error.txt+m26_error.sun error\!run!_m26_latchup_error.txt
			del m26_error.sun
			goto jtag2
		)
		IF not errorlevel=1 (
			ECHO !file! No latch up in M26
			echo #!UCNT! >> error\!run!_latch.txt
			copy /a error\!run!_latch.txt+frame.sun error\!run!_latch.txt
		)

REM latchup_readout v


:read_jtag
```

```
echo IN: read_jtag
call jtag_upload.bat
call jtag_start.bat
call jtag_read_frame1.bat
REM read jtag from frame fifo1 so we don't flush data in frame fifo

seu_test.exe jtag_r.dat jtag.dat out1.txt out2.txt
echo  seu_test error_level %errorlevel%

        IF  not errorlevel=1 (
                ECHO seu_test.exe jtag_r.dat jtag.dat good!
                goto read
        )
        IF  errorlevel=1 (
                ECHO seu_test.exe jtag_r.dat jtag.dat errors
                echo #!UCNT! >> error\!run!_jtag.txt
                copy /a  error\!run!_jtag.txt+jtag_r.dat error\!run!_jtag.txt

                echo #!UCNT! >> error\!run!_m26_jtag_error.txt
                copy /a error\!run!_m26_jtag_error.txt+m26_error.sun error\!run!_m26_jtag_error.txt

                del m26_error.sun
                REM goto jtag2
        )
REM check for losing power
call jtag_upload.bat
call jtag_start.bat
call jtag_read_frame1.bat
REM read jtag from frame fifo1 so we don't flush data in frame fifo

seu_test.exe jtag_r.dat jtag.dat out1.txt out2.txt
        IF  not errorlevel=1 (
                goto read
        )
:end
```