Title : Test results for CASE study on the SHAMROC DAC (Californium-252 Assessment of Single-event Effects)

Prepared by	:	Rob Wolfs	**Date**	:	11-03-2009
Checked by	:	Anne Stellinga Jan-Harm Nieland	**Date**	:	07-04-2009 10-04-2009
PA agreed by	:	P. Kohsiek	**Date**	:	17-06-2009
Authorised by	:	P. Laubert	**Date**	:	17-06-2009

Distribution

SRON: Phillip Laubert
Martijn Smit
Jan-Harm Nieland
Jan-Rutger Schrader
Anne Stellinga
Rob de la Rie
Jelle Talsma
Rob Wolfs
Ed de Vries
Bart van Kuik
Joris van Rantwijk
Chang Shu
Foppe Hoekstra
Peter Kohsiek
Herman Jacobs
Jan Veenendaal
Martin Frericks

Axiom IC:
Daniel Schinkel
Vincent Arkesteijn

ESA/ESTEC:
Bob Nickson
Reno Harboe Sorensen

Document Change Record

Issue	Date	Changed Section	Description of Change
0.1	11-03-2009		New document
1.0	17-06-2009		First official release

Table of contents

List of figures:

List of tables:

Abbreviations and acronyms

Item	Meaning
ADC	Analog to Digital Converter
CASE	Californium-252 Assessment of Single-event Effects
DAC	Digital to Analog Converter
DUT	Device Under Test
ESPAX	Exomars asic SPAce qualification of miXed signal asics
FPGA	Field Programmable Gate Array
LET	Linear Energy Transfer
LET_{th}	Linear Energy Transfer Threshold
SEE	Single Event Effect
SEL	Single Event Latchup
SET	Single Event Transient
SEU	Single Event Upset
SHAMROC	SEIS High Accuracy Mixed-signal Read-out Chip
SRIM	the Stopping and Range of Ions in Matter

Applicable Documents

[AD#]	Doc. Reference	Issue	Title
[AD1]	SRON-SHAMROC-PL-2009-001	1	CASE testplan
[AD2]	ESCC-25100	1	Single Event Effects Test Method and Guidelines
[AD3]	SRON-SHAMROC-RS-2007-001	3	SHAMROC Design Specification
[AD4]	SRON-SHAMROC-PL-2009-001	1	Test Plan for CASE study on the SHAMROC DAC
[AD5]			
[AD6]			

Reference Documents

[RD#]	Doc. Reference	Issue	Title
[RD1]	SRON-ESPAX-PL-2008-001	1.0	ESPAX project plan
[RD2]			
[RD3]			
[RD4]			
[RD5]			
[RD6]			

1 Introduction

CASE stands for Californium-252 Assessment of Single-event Effects. In order to do a SEE test on an accelerator a CASE is necessary to verify if the test setup is capable of detecting SEE. The CASE setup is also a good instrument to get a rough estimate of the sensitivity of the chip for SEE. The test is performed within the ESPAX project [RD1] to get this estimate of the sensitivity of the SHAMROC DAC.

A CASE test is performed using radioactive Californium-252. Californium-252 (^{252}Cf) is a neutron transmitter which decays for 96.9% by alpha emission and for 3.1% in spontaneous fission. The typical activity of ^{252}Cf is between 1 and 3µCi. 1µCi emits 4400neutrons/second. The source has an activity on the silicon between 2500 and 3500counts/cm²/minute at 2cm distance between source and DUT. The average Linear Energy Transfer (LET) of the fission products is 43MeV/mg/cm². The average penetration depth of ^{252}Cf into silicon is 14.2µm (see also [AD2]) where the active layer of the SHAMROC DAC is 10.9µm below the die surface. More detailed simulations can be found in chapter 4.

Since this field is quite new for the ASIC development at SRON, the main question at the start of the CASE test was: How many events can be expected when testing a DAC under a radioactive source? The sensitivity can range from 1 event per second, per minute, per hour or even per day. The results of the CASE test are described in this report. The test was performed according to the plan described in [AD4].

For this first test the DAC is chosen due to the availability of the test setup, personnel and testability. The test board needed some adaption to fit with the vacuum jar of the CASE setup. The test setup as used during the test period is described in chapter 2.

Chapter 3 describes the test results as measured during the test period from 5 March 2009 until 20 March 2009. Because a lid of the package will act as a shield to the radiation, the lid is removed from two of the devices. The devices used during the CASE test have the following serial numbers 6210D-1025 and 6210D-1026.

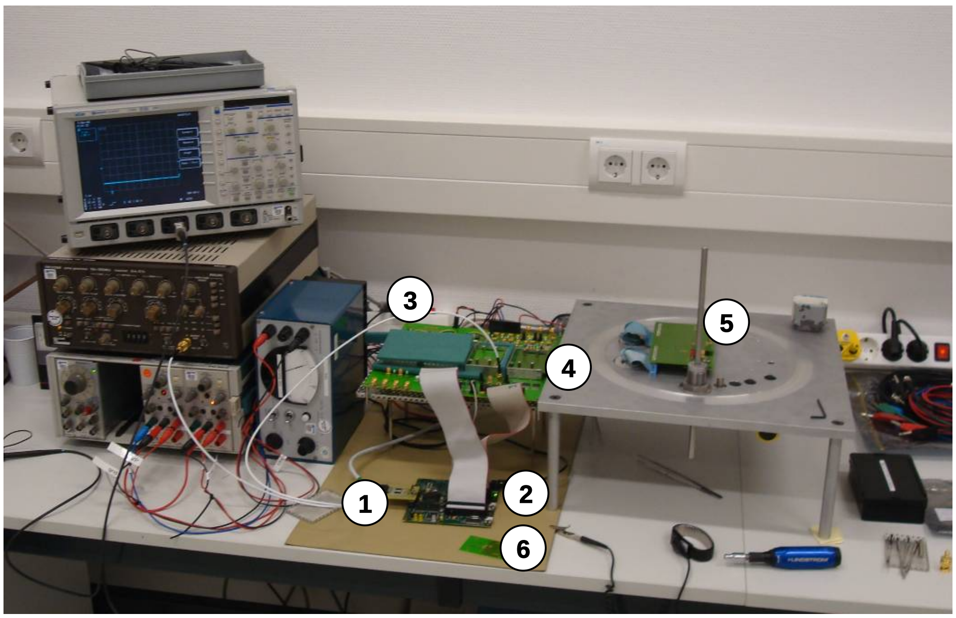
Chapter 5 describes the conclusions made from the results.

The test log can be found in Appendix A. In this log all actions performed during this the CASE test period are described here. Appendix B gives the final version of the test file CASE.pl.

2 Test setup

2.1 Hardware

The test plan [AD1] describes the connections between each individual board. The complete test setup contains six boards which are numbered in Figure 1 and Figure 2. The test setup is powered by three power supplies, SRON-barcode 0733 for 6V digital supply, 0495 for 10Vp and 8Vn analog supply and 0952 for the heaters. The 40MHz clock is provided via an external pulse generator (0214).



Used boards:
①: RS422 interface (328-N-3101)
②: FPGA evaluation board
③: DAC testboard (328-N-3100)
④: Extender 'outside' (328-N-3109)
⑤: Extender 'inside' (328-N-3109)
⑥: Stack PCB with DAC (328-N-3102)

Figure 1: Picture of the complete test setup.

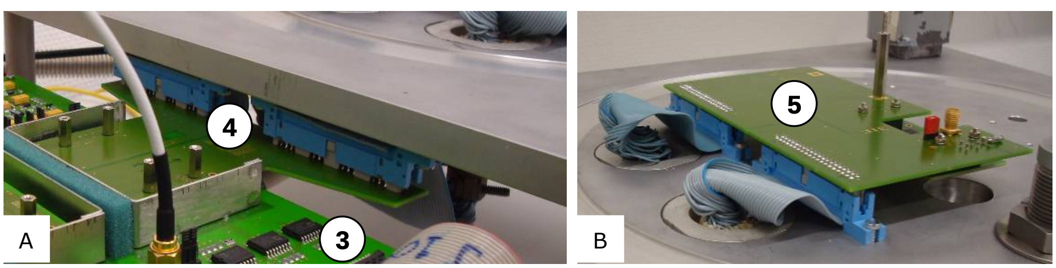


Figure 2: Detailed photos of the connectors towards (A) and inside (B) the vacuum.

Due to the fact that all connections from board to board are made without cables, except for the digital signals towards the FPGA and the PC, the position of each board is fixed. This will result in a very stable test environment. When a sample is changed, the new sample will be in exactly the same position as the previous one.

Since air and the device lid will block the radiation partials, the test is performed in a vacuum and the lid is removed from the package. The position of the DAC with respect to the ^{252}Cf source can be seen in Figure 3.

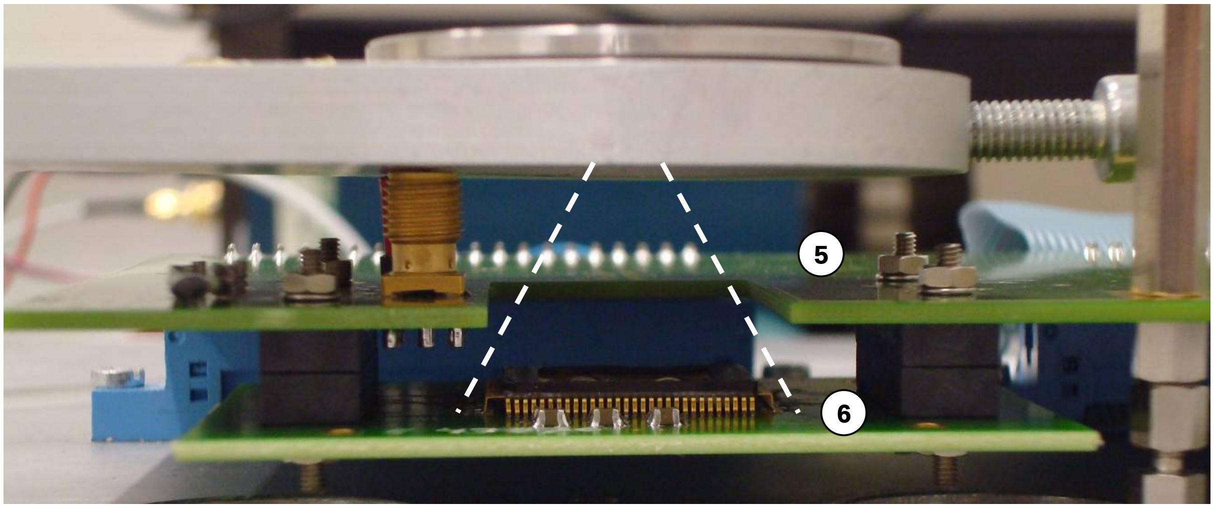


Figure 3: DAC without a lid under ^{252}Cf source inside the vacuum.

2.2 Software

During the CASE test several parameters are read, checked and written via the software scripts as described in the CASE test plan [AD1]. The script used is called CASE.pl (final version can be found in Appendix B) and can be started via the kickoff menu inside the EGSE server, or via a command line inside a terminal. When the script is started from the command line logging of the screen output is possible and multiple starts of the script are possible. The parameters read and written via the CASE.pl script are listed in Table 1. These parameters are grouped inside a block command. Therefore such a block command consists of several read commands from the digital part of the DAC, a read command of the science ADC and several write commands towards the digital part of the DAC. The time diagram of such a communication is illustrated in Figure 4.

Name	Default script setting	Read	Write	Info
DAC Data_in	0x555555	✓	✓	Approximately ¼ Fs range.
DAC PWD	0x000000	✓	✓	DAC on (def.)
DAC TestD 19-0	0x0F0F0F	✓	✓	Inputs for analog section to overwrite digital settings
DAC TestD 34-20	0x0070F0	✓	✓	
DAC TestGrp	0x000000	✓	✓	Select group 0 (def.)
DAC TestSel	0x000000	✓	✓	Select bitstream 0 (def.)
DAC CodeOverr	0x000000	✓	✓	Normal operation, no overwrite with TestD (def.)
DAC Misc	0x000000	✓	✓	ClkRatio = 50kHz, dither disabled. (def.)
ADC data	-	✓	×	Measured analog signal
SEL count ref	-	✓	×	Number of SEL detections on reference current
SEL count dig	-	✓	×	Number of SEL detections on digital current
SEL count I/O	-	✓	×	Number of SEL detections on I/O current
SEL restart	-	✓	×	Zero when DAC is restarting

Table 1: Default list of parameters read and written within each block.

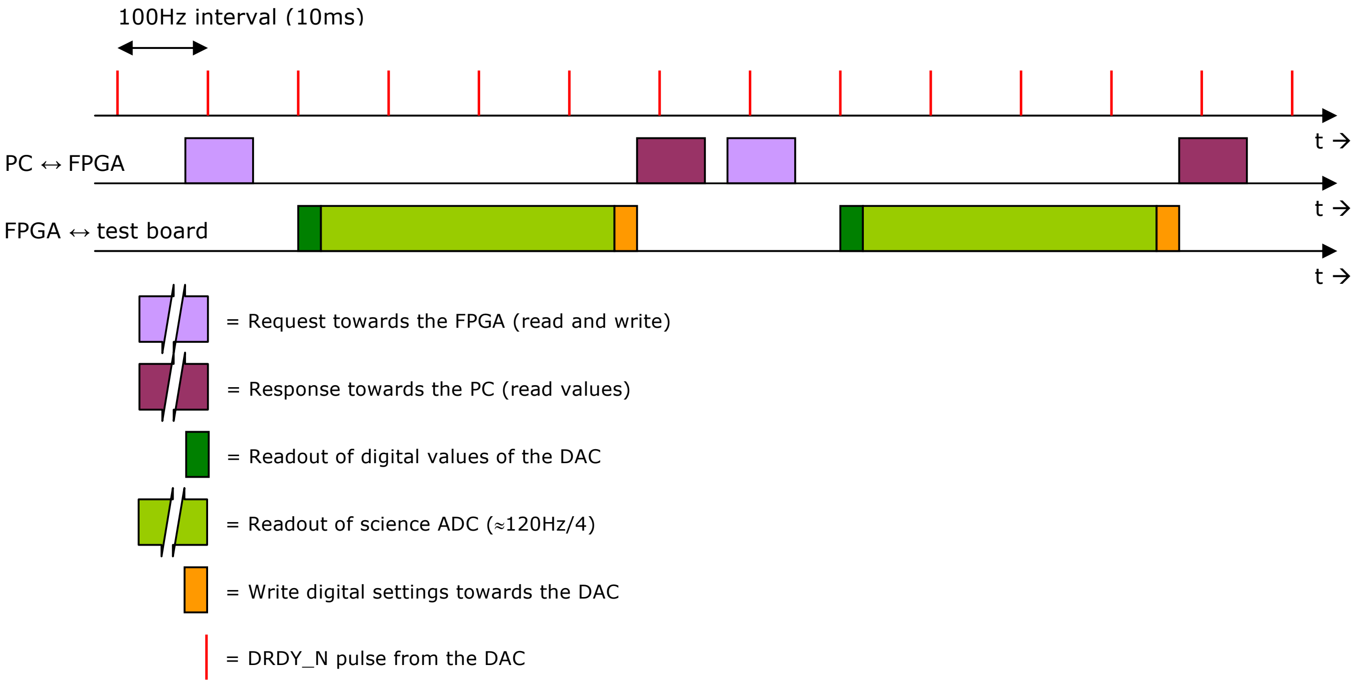


Figure 4: Timing diagram of a single block read.

The soft- and hardware are designed to detect all possible SET, SEU and SEL. The expectation is that the DAC is not very sensitive for these effects due to a number of precautions implemented inside the DAC. For the digital part the DARE kit protects against SEU. The analog part is designed with guard rings to protect against SEL. The real test in practice will prove if these precautions are protecting the device for where they are designed for.

3 Test results

If a SEE occurs (SET, SEL or SEU) this can most likely be seen as a distortion of the output of the DAC. An ADC on the testboard measures the output of the DAC. The software automatically detects if the ADC readout is outside the expected limits. These limits are dependent of the ADC measurements of the last 10 samples. This calculation is done inside the CASE.pl script with a moving average filter. This way slow drifts which are possibly inside a measurement period are cancelled out since an SEE on the DAC output is a rapid change on the ADC readout value. As said, the limit is set to be the mean value of the last 10 ADC samples plus or minus half of a given spread. In formula this looks like Equation 1. This spread value is the default peak-to-peak noise on the DAC output. In case the spread value is set to low too much noise is detected instead of a real SEE. If the spread is set too large it may be possible to miss a SEE. However, since all data is logged into a file it is possible to verify the data for SEE detections with another limit (spread) value even after the measurement.

$$hLim = \frac{\sum_{i=10}^{i=0} ADC\_read_i}{10} + \frac{ADC\_spread}{2}$$

$$lLim = \frac{\sum_{i=10}^{i=0} ADC\_read_i}{10} - \frac{ADC\_spread}{2}$$

Equation 1: Calculation of ADC detection limits

To detect SEE in the digital domain (like SEU), the read data from the registers is compared to the written value in that same register. Each measurement period the same data is written to each register, therefore each digital limit is also a fixed value.

Each SEE detection is logged into its own output file. In each output file several values are stored on a row separated by a tab. In the first column the time from start of the measurement is stored. This time value has the unit seconds. The second column lists the duration time of each individual measurement. The third column stores the SEE counter for that particular register or value. The fourth column lists the actual read value. The fifth and sixth column lists the used high and low limits which are the calculated values for the ADC detection and the written values for all the digital registers.
As said before, the limits of the ADC detection are set to be approximately the same as the peak to peak spread of the ADC measurements. This way the software provides 'focus' points in the data. These detections do not mean that it is indeed a SEE, but only a crossing of the limit. By analysing each individual 'focus' (detection) point it can be determined how often it concerns a real SEE.

3.1 ADC limit detections (SET)

3.1.1 6210D-1026

As said in the previous chapter, limit detections on the ADC value does not mean that it always concerns a real SEE. Table 2 lists the detections measured on sample 6210D-1026 when no ^{252}Cf source was placed over the device. There are however 3 limit crossing detections over the readout period of 1E6 blocks. 1E6 blocks correspond to a readout time of approximately 16.67 h (0.06 s per block). Of course, those limit crossings can not be a SEE since there is no SEE-source. All detections are therefore marked as not likely to be a SEE (–).

Date_time	Log file	Measurement time Min (sec.)	Measurement time Max (sec.)	Limit detection Time (sec.)	Limit detection Value (LSB)	Possible SEE	REF
20090305_153513…	none	0.058	0.14	13259.7	16.1	–	1
				20249.7	-16.5	–	2
20090305_235350…	none	0.058	0.16	12222.1	-16.8	–	3

SEE estimation: –: not likely to be a SEE, +: possible SEE detection

Table 2: Summary results of sample 6210D-1026 over 1E6 readouts <u>without</u> ^{252}Cf.

The time plot of each of these events (Figure 5 and Figure 6) shows that the limit is rarely crossed. When the limit is crossed the detection is hardly above the applied limit.

In the first file (Figure 5 , 500000 values) two detections are found. Each detection is marked (♦) and plotted in the subplot below (zoom plot). Each zoom plot contains three lines, a red line which is the ADC output and two blue lines which represent the limits. If the signal (red line) increases to a value outside these limits, this will be detected as a possible SEE. Looking at the zoomed data values, the detections can be classified as SEE or just noise. SEE detections just above the limit are hard to distinguish from the noise-detections, because both patterns look very similar. Each detection is marked with a REF-number. This number can also be found inside the results table.

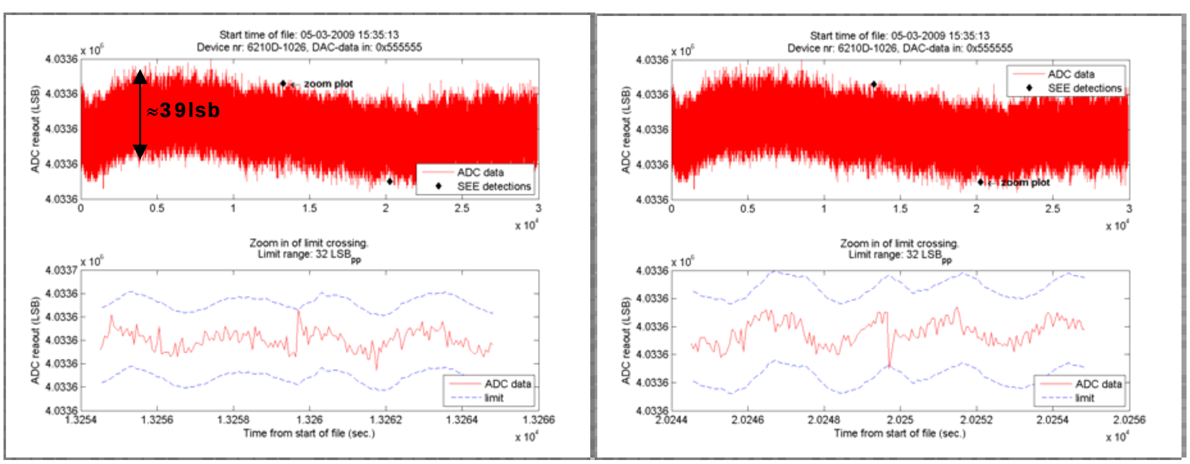


Figure 5: Detections on 6210D-1026 without ^{252}Cf source (left: REF 1 and right: REF 2).

In the next plot (Figure 6) also 500000 ADC values are read but only one limit detection has occurred during this period.

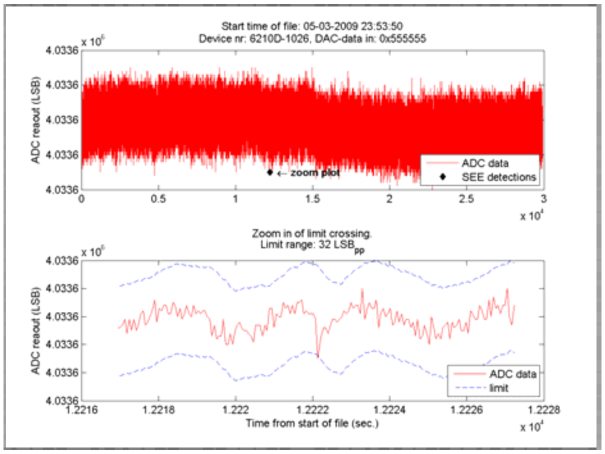


Figure 6: Detections on 6210D-1026 without ^{252}Cf source. (REF 3)

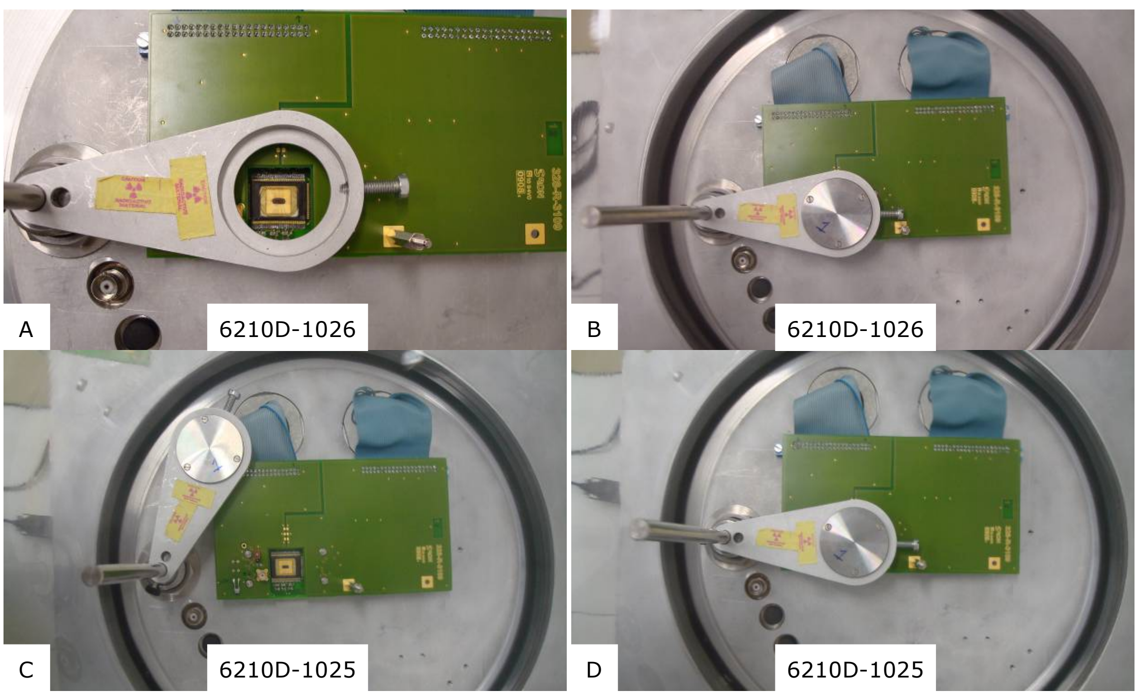


Figure 7: Testing sample without (A, C) and with (B, D) a ^{252}Cf source.

The CASE setup can easily be changed to a setup with or without a source. The source is placed inside the metal socket which is connected to the vertical rod. This rod can be turned from outside the vacuum. It is therefore possible to leave the DUT inside the vacuum while tests are being performed with and without a source just by moving the source over the DUT. The test on sample 6210D-1025 is performed in such a way.

Sample 6210D-1026 is tested with a source placed over the device for a period of 4E6 readouts. The test took approximately 2 days and 18.67 hours and is split up into 8 tests of 500000 readouts. The results of the ADC limit detections are listed in Table 3.

Date_time	Log file	Measurement time		Limit detection		Possible SEE	REF
	CASE_20090306	Min (sec.)	Max (sec.)	Time (sec.)	Value (LSB)		
20090306_131819…	_log4_1.log	0.057	0.14	4497.18	-16.5	+−	4
				4882.97	17	−	5
20090306_213558…	_log4_2.log	0.057	0.12	23644.1	-17	−	6
20090307_055339…	_log4_3.log	0.057	0.13	4332.94	-17.5	+−	7
				10092.9	-16	−	8
				17030.9	16.4	−	9
				22155.1	16.7	−	10
20090307_141125…	_log4_4.log	0.057	0.12	10068.7	-19	+−	11
20090307_222909…	_log4_5.log	0.057	0.6 !	3002.81	16.5	−	12
				9492.72	-16.5	−	13
				20458.4	16	−	14
				28102.7	-16	−	15
20090308_064721…	_log4_6.log	0.057	0.12	4930.75	-16	−	16
				9343.28	-40.5	+	17
				9343.33	-37		
				9343.39	-38		
				9343.47	-31		
				9343.52	-28.6		
				9343.58	-19		
				11939.3	16	−	18
				16519.5	16	−	19
20090308_150507…	_log4_7.log	0.057	0.12	4636.02	-16	−	20
				13875.5	-16	−	21
20090308_232254…	_log4_8.log	0.057	0.12	none	none	none	

SEE estimation: −: not likely to be a SEE, +: possible SEE detection

Table 3: Summary results of sample 6210D-1026 over 4E6 readouts <u>with</u> ^{252}Cf.

The table shows a lot of detections just above or just below the limit level. These detections are most of the time not likely to be caused by SEE because they have a similar profile as the detections without ^{252}Cf. All these detections are marked with a '−' inside the table. Some of these detections show a slightly higher possibility to be a SEE because these have a rapid, short and high change in value. These detections are marked with a '+−' sign. During the readout period there is one period of 6 detections exactly after each other which can be regarded as a possible SEE. This period is therefore marked with a '+' sign. Also the level of this detection is a much larger than all the other detections during this complete period. In Figure 8 to Figure 16, the plots of the detected levels can be found. Again each plot is linked to the table with a REF number.

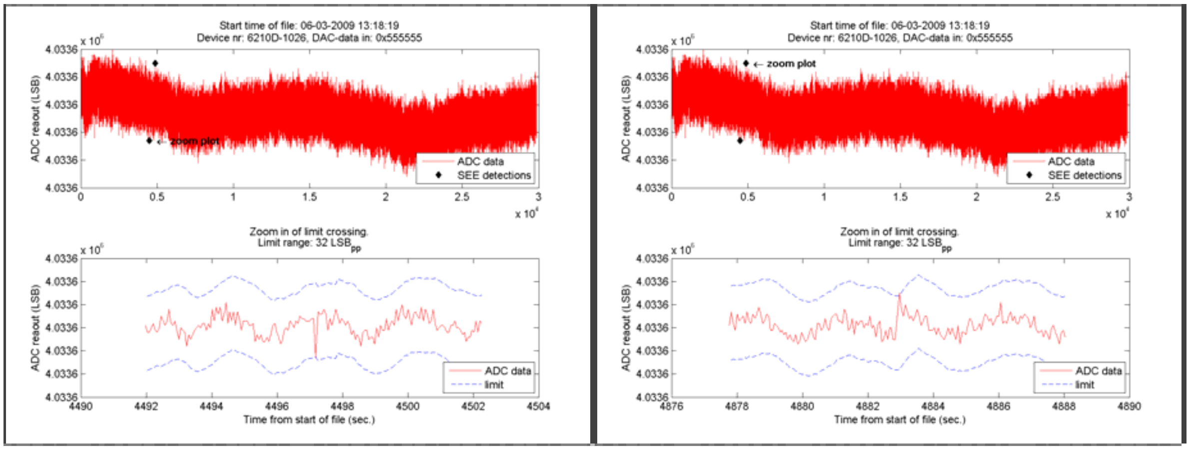


Figure 8: Detections on 6210D-1026 with ^{252}Cf source (left: REF 4 and right: REF 5).

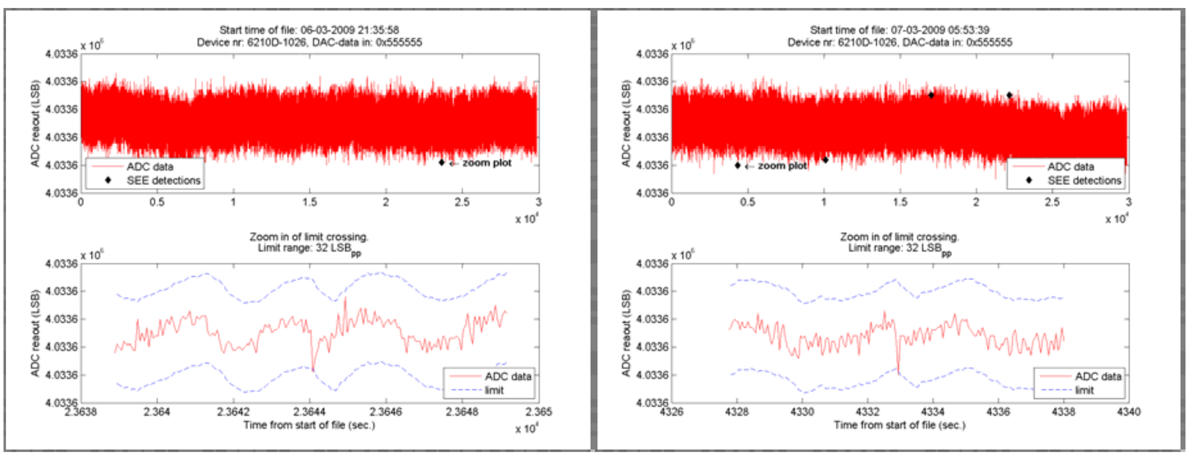


Figure 9: Detections on 6210D-1026 with ^{252}Cf source (left: REF 6 and right: REF 7).

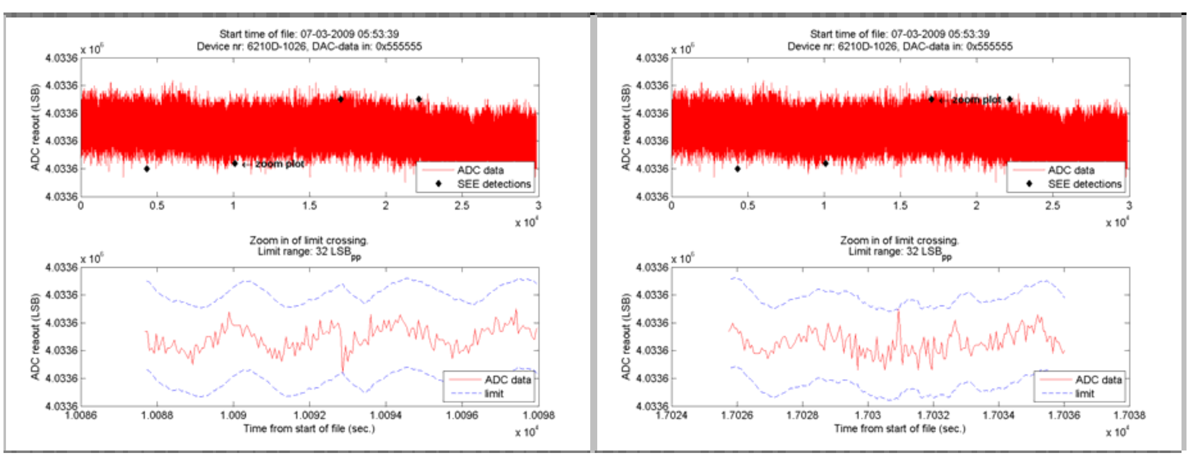


Figure 10: Detections on 6210D-1026 with ^{252}Cf source (left: REF 8 and right: REF 9).

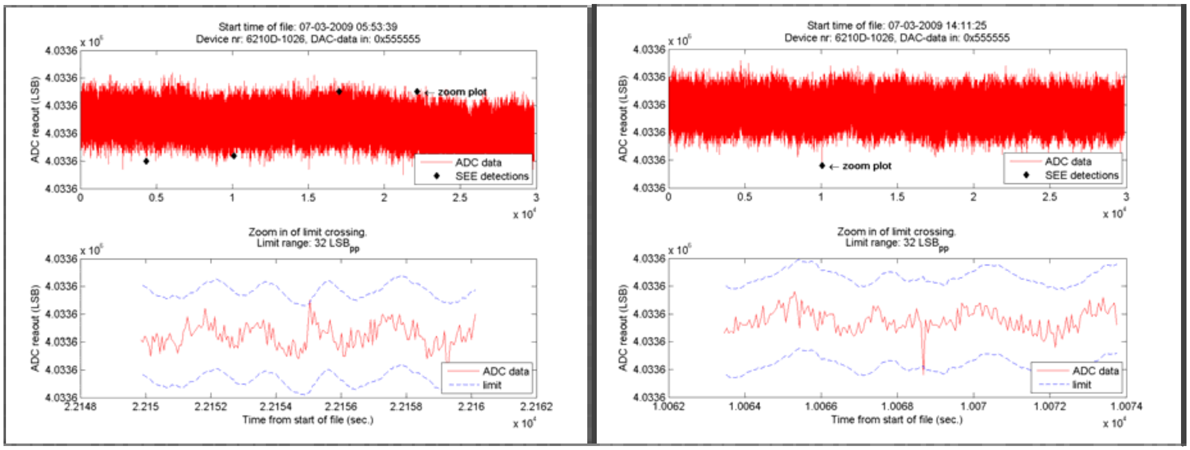


Figure 11: Detections on 6210D-1026 with ^{252}Cf source (left: REF 10 and right: REF 11).

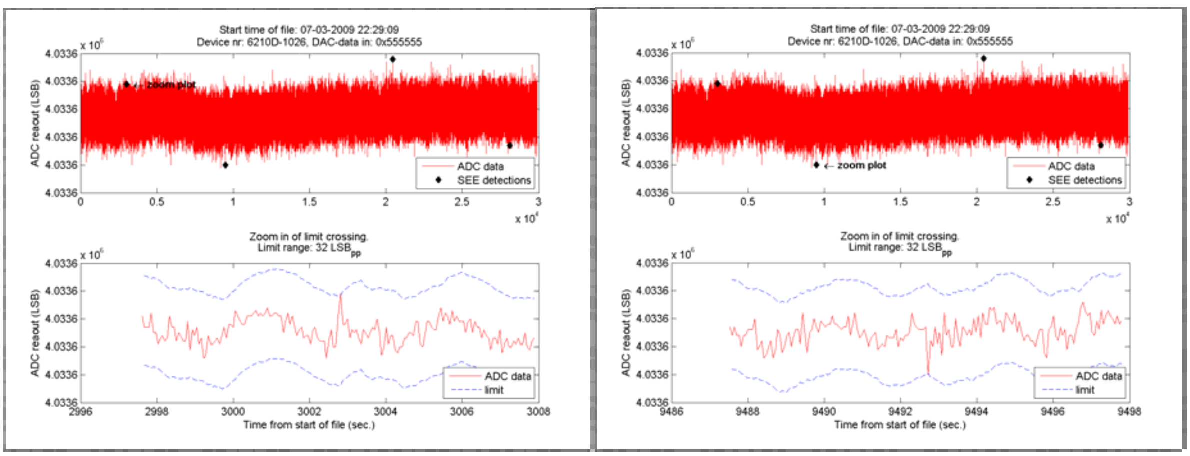


Figure 12: Detections on 6210D-1026 with ^{252}Cf source (left: REF 12 and right: REF 13).

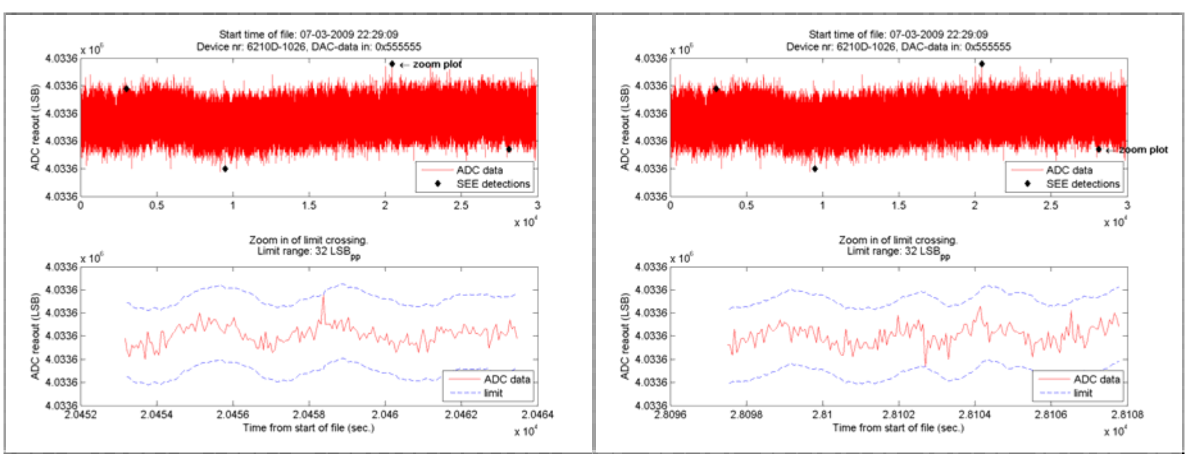


Figure 13: Detections on 6210D-1026 with ^{252}Cf source (left: REF 14 and right: REF 15).

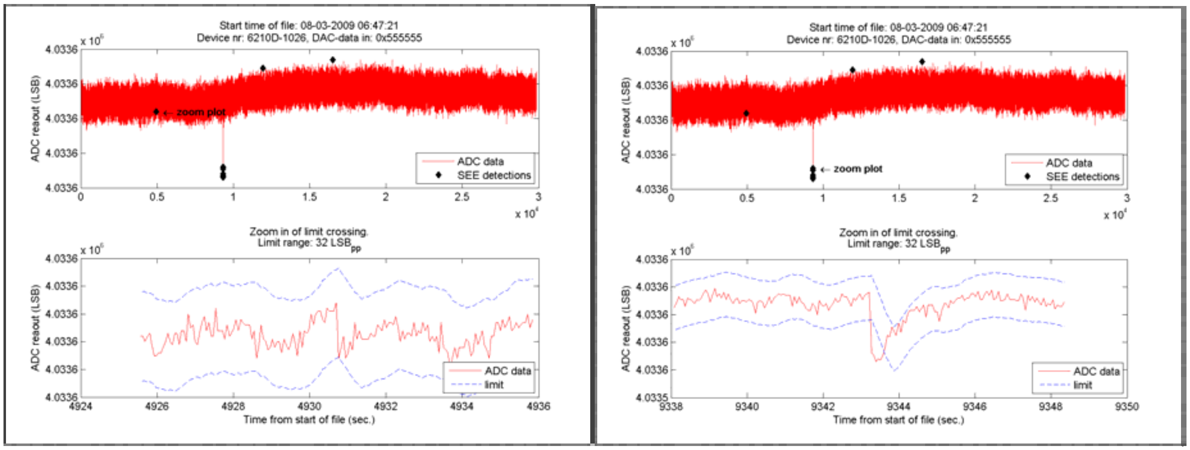


Figure 14: Detections on 6210D-1026 with ^{252}Cf source (left: REF 16 and right: REF 17).

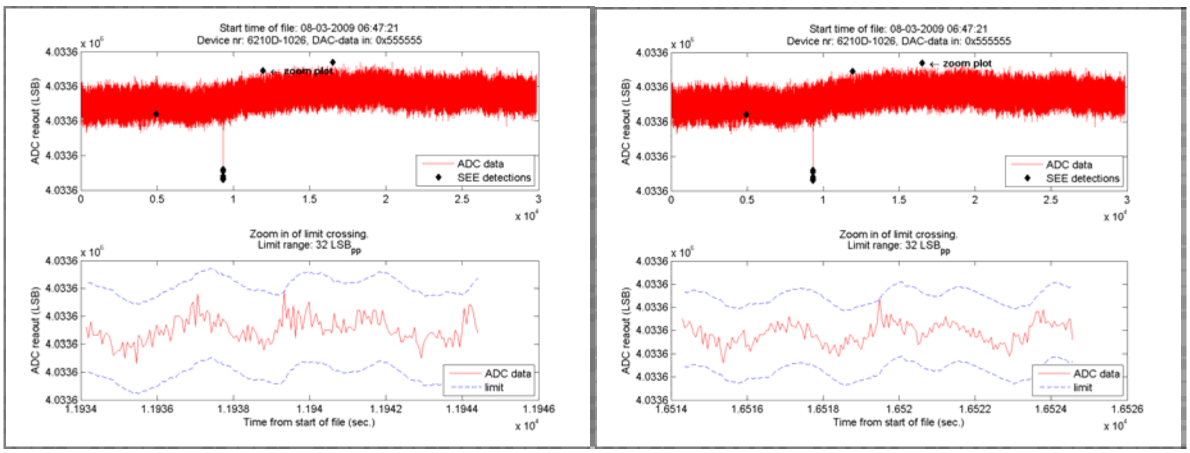


Figure 15: Detections on 6210D-1026 with ^{252}Cf source (left: REF 18 and right: REF 19).

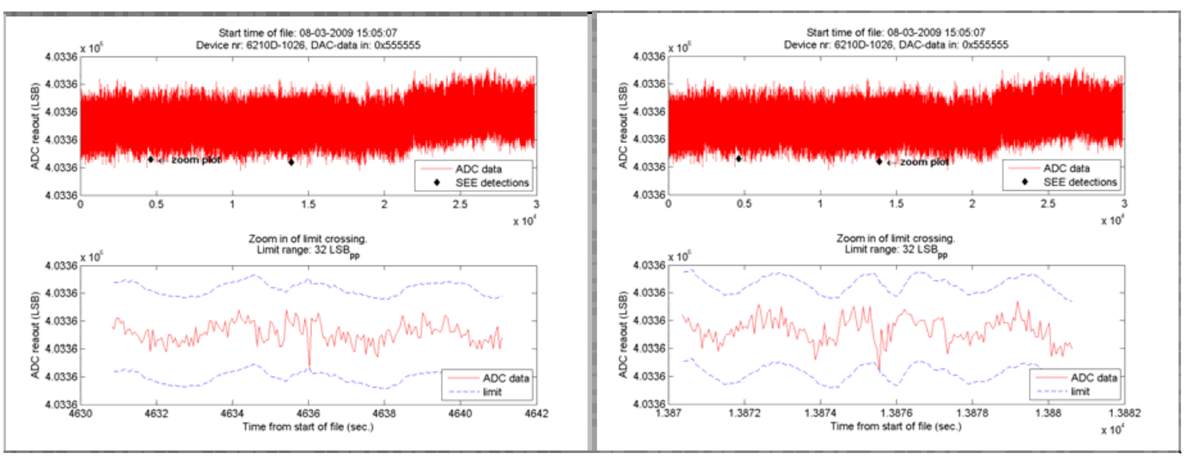


Figure 16: Detections on 6210D-1026 with ^{252}Cf source (left: REF 20 and right: REF 21).

3.1.2 6210D-1025

Device 6210D-1025 is also measured for single event effects. Also from this device the lid is removed and the device is positioned inside the vacuum jar. Device 6210D-1025 is measured with and without a ^{252}Cf source above the die. These two tests are done within one continuous readout period of approximately 4.9E6 readouts. The test started 9 March at 11:45. The test started without the source placed over the device (Figure 7C). The next day (10 March) at 9:00 h, the source was turned to the position over the device (Figure 7D). This was done from outside the vacuum jar by turning the rod. This movement causes some interference due to a metal contact with the test board. The results from these two periods, including the interference of the movement (REF 27), are listed in Table 4 and Table 5.

Date_time	Log file	Measurement time		Limit detection		Possible	R
		Min	Max	Time	Value	SEE	E
	CASE_20090309	(sec.)	(sec.)	(sec.)	(LSB)		F
20090309_114503…	_log5_1.log	0.058	0.14	9630.05	-21	+−	22
20090309_200716…	_log5_2.log	0.058	0.14	7536.83	-17.7	+−−	23
				12246.9	-16.5	+−−	24
				19095.0	-18.8	+−−	25
20090310_042931…	_log5_3.log	0.058	0.15	3052.0	-16.9	+−−	26
				16042.4	74 (9x)	Move ^{252}Cf	27
				16043.9	-85.8 (9x)		

SEE estimation: −: not likely to be a SEE, +: possible SEE detection

Table 4: Summary results of sample 6210D-1025 over 1.266.128 readouts <u>without</u> ^{252}Cf.

Date_time	Log file	Measurement time		Limit detection		Possible	R
		Min	Max	Time	Value	SEE	E
	CASE_20090309	(sec.)	(sec.)	(sec.)	(LSB)		F
20090310_042931…	_log5_3.log	0.058	0.15	none	none	none	
20090310_125203…	_log5_4.log	0.058	0.17	11999.7	16.5	−	28
20090310_211444…	_log5_5.log	0.058	0.18	11822.5	-18	+−	29
				28459.0	-18	−	30
20090311_053736…	_log5_6.log	0.058	0.17	12507.2	-16.9	+−	31
20090311_140047…	_log5_7.log	0.058	0.17	20205.7	-16.6	−	32
				28045.6	16.2	−	33
20090311_222409…	_log5_8.log	0.058	0.19	9144.23	-16.7	−	34
				24664.4	17.4	−	35
20090312_064747…	_log5_9.log	0.058	0.32	7683.74	-19	++−	36
				15377.0	Readout aborted		

SEE estimation: −: not likely to be a SEE, +: possible SEE detection

Table 5: Summary results of sample 6210D-1025 over 3.564.658 readouts <u>with</u> ^{252}Cf.

The results are corresponding with the results found on device 6210D-1026. Again several detections can be seen when there wasn't a source placed above the device. When the source is moved to the position over the device, the number of detections stayed about the same. In each file zero to 4 detections can be seen. All the detected limit crossings are plotted in Figure 17 to Figure 24.

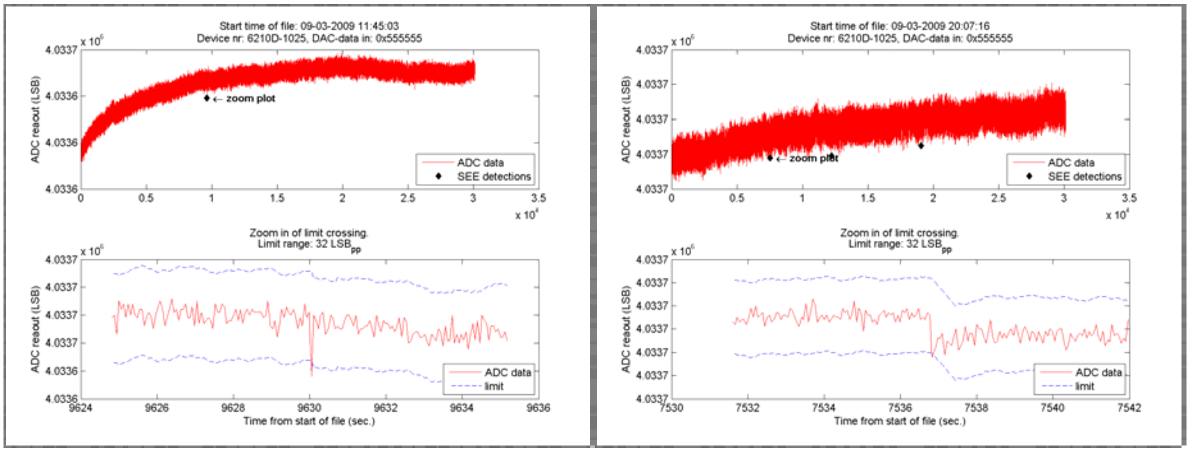


Figure 17: Detections on 6210D-1025 without ^{252}Cf source (left: REF 22 and right: REF 23).

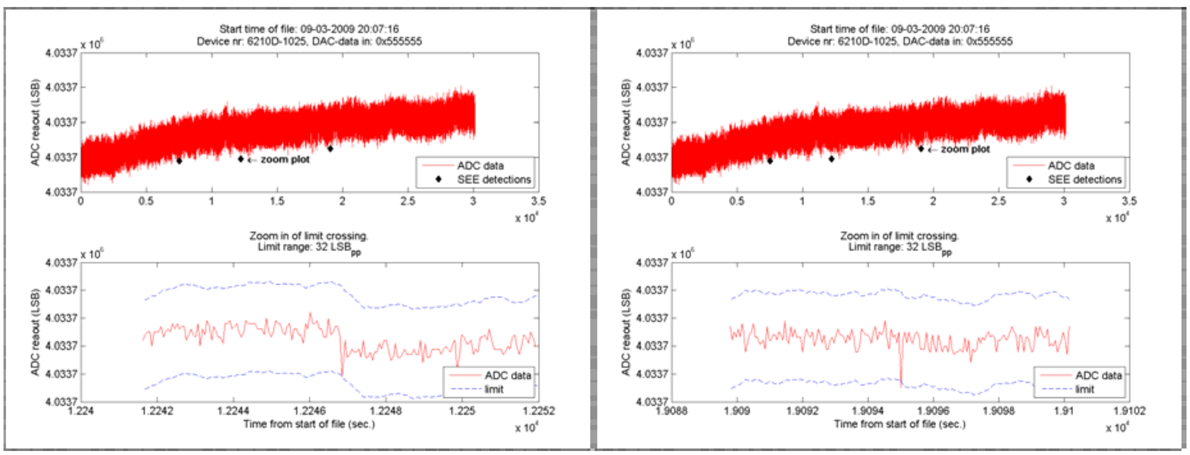


Figure 18: Detections on 6210D-1025 without ^{252}Cf source (left: REF 24 and right: REF 25).

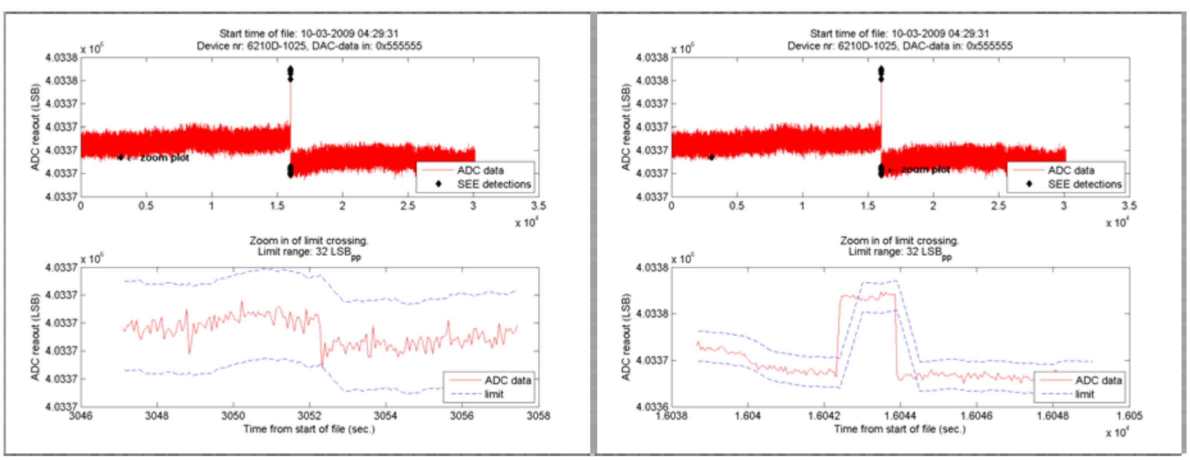


Figure 19: Detections on 6210D-1025 without ^{252}Cf source (left: REF26 and right: REF 27).

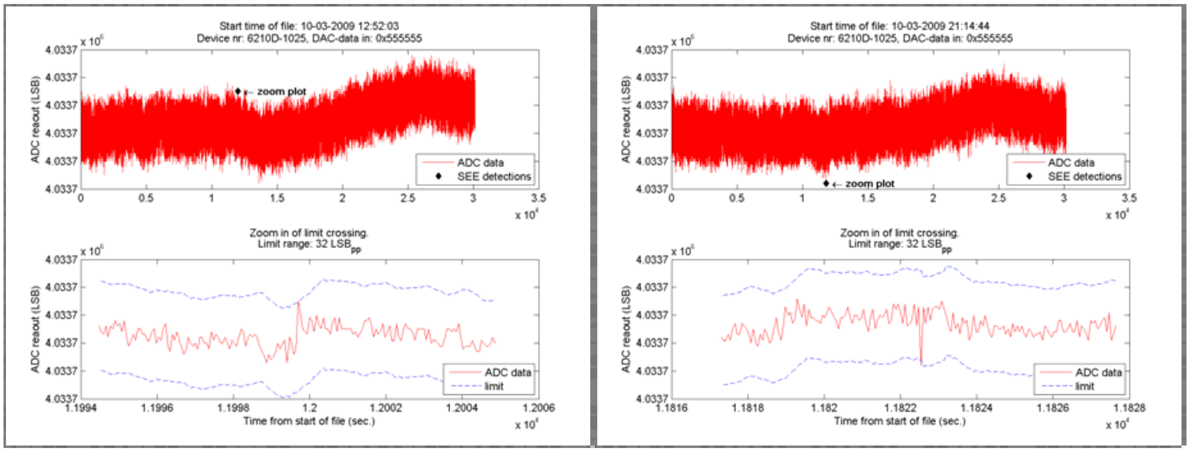


Figure 20: Detections on 6210D-1025 with ^{252}Cf source (left: REF 28 and right: REF 29).

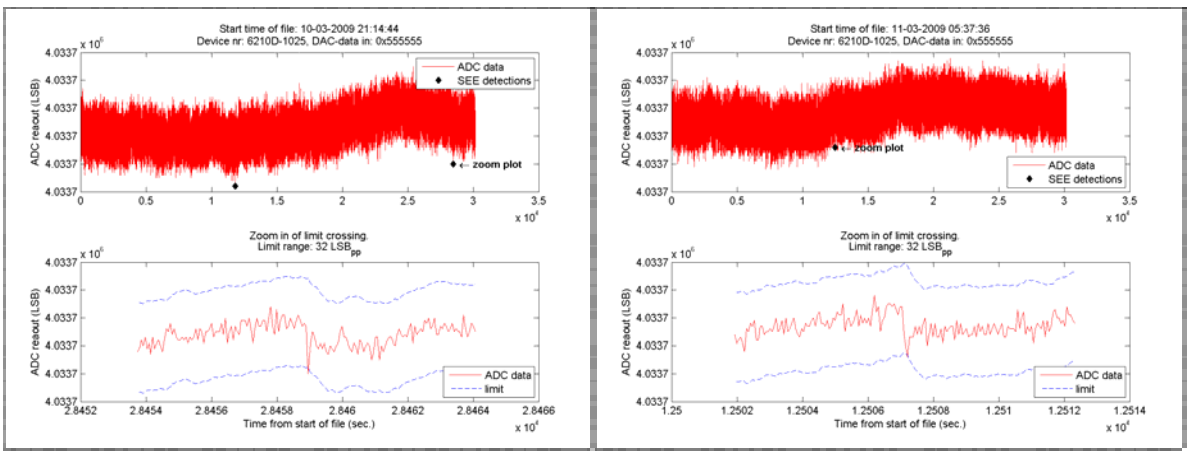


Figure 21: Detections on 6210D-1025 with ^{252}Cf source (left: REF 30 and right: REF 31).

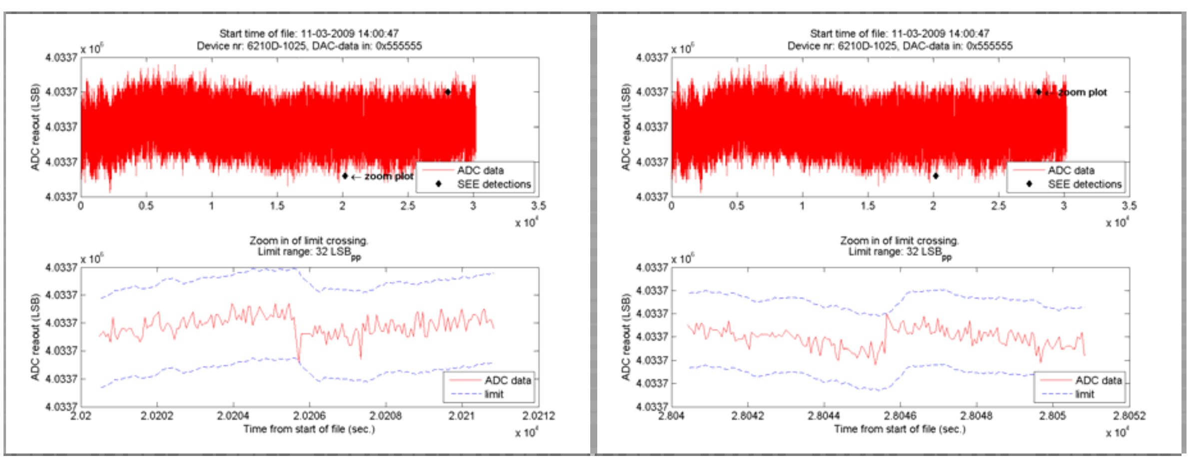


Figure 22: Detections on 6210D-1025 with ^{252}Cf source (left: REF 32 and right: REF 33).

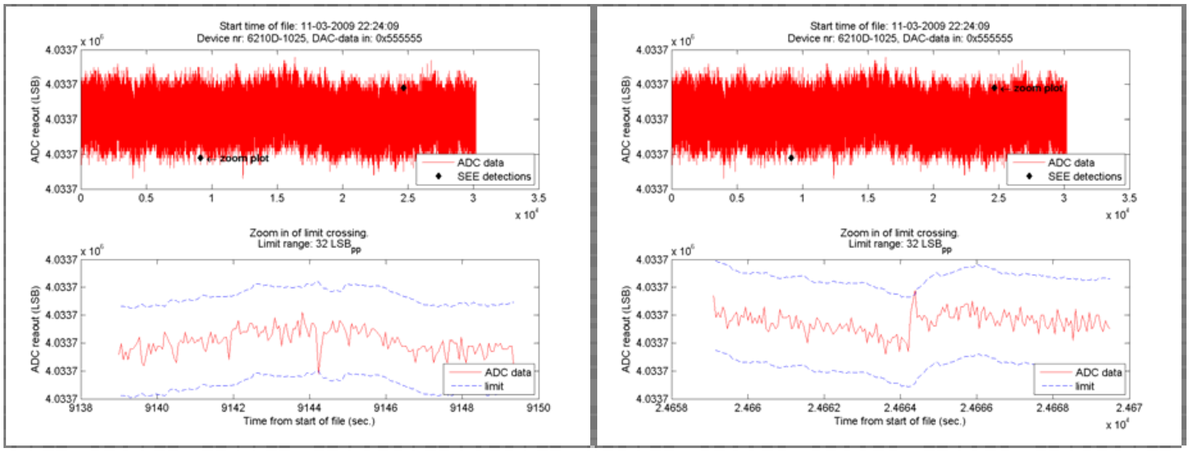


Figure 23: Detections on 6210D-1025 with ^{252}Cf source (left: REF 34 and right: REF 35).

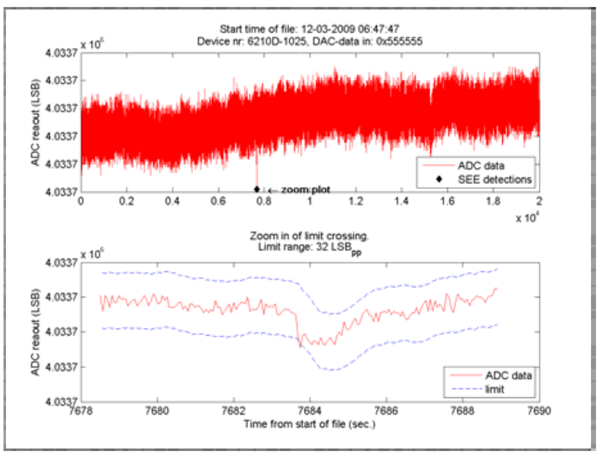


Figure 24: Detections on 6210D-1025 with ^{252}Cf source (REF 36).

Figure 19 with REF 27 shows the detections found when moving the source into position. These detections can therefore be ignored as SEE. The rest of the limit detections are also to not likely to be an actual single event effect, except the detection in Figure 24 (REF 36). This detection has the highest probability to be an actual SEE although the size of this peak is still only 19 LSB.

3.1.3 ADC Measurement before the analog filter

With the DAC test board it is possible to select the measurement path between the DAC and the ADC. The default operation works with a buffer and filter between the DAC and ADC. The results of these measurements are explained in the previous paragraph. This paragraph explains the results found when the ADC measured directly at the output of the DAC so just before the filter. It is possible that the measured results of the previous chapter are influenced by the analog filter. It could be that a large fast SEE at the output of the DAC is filtered to a smaller and slower effect.
The gain of the ADC is also changed from 1x to 64x. The output of the DAC is set therefore to a small value 10LSB. The value of 10 is chosen to have also some ones in the input register of the DAC, these might be set to zero as a result of a SEU. When a SEE occurs at the DAC output, this will not be influenced by the filter which could be the case during the previous measurement. The amplifier inside the ADC could increase the small effect so it is easier to be detected. The results found with these measurements are plotted in Figure 25 for sample 6210D-1025 and Figure 26 for sample 6210D-1025.

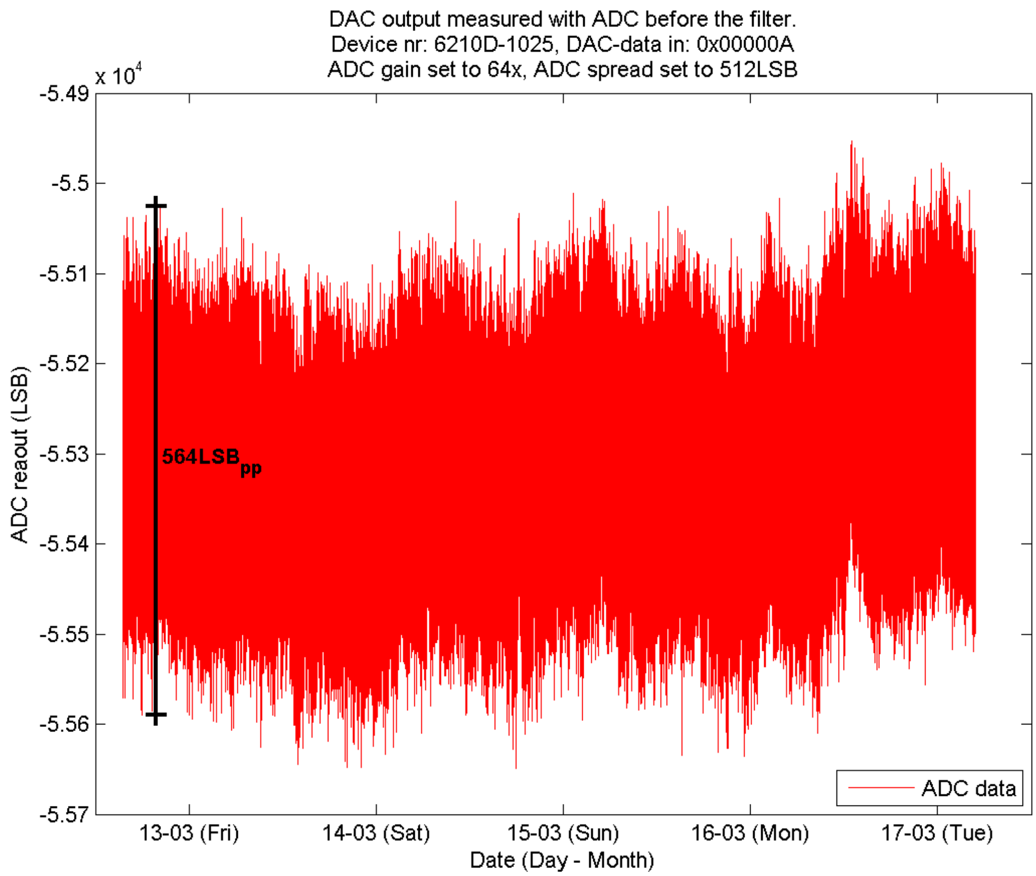


Figure 25: DAC output measured before filter over 4.5 days (6210D-1025).

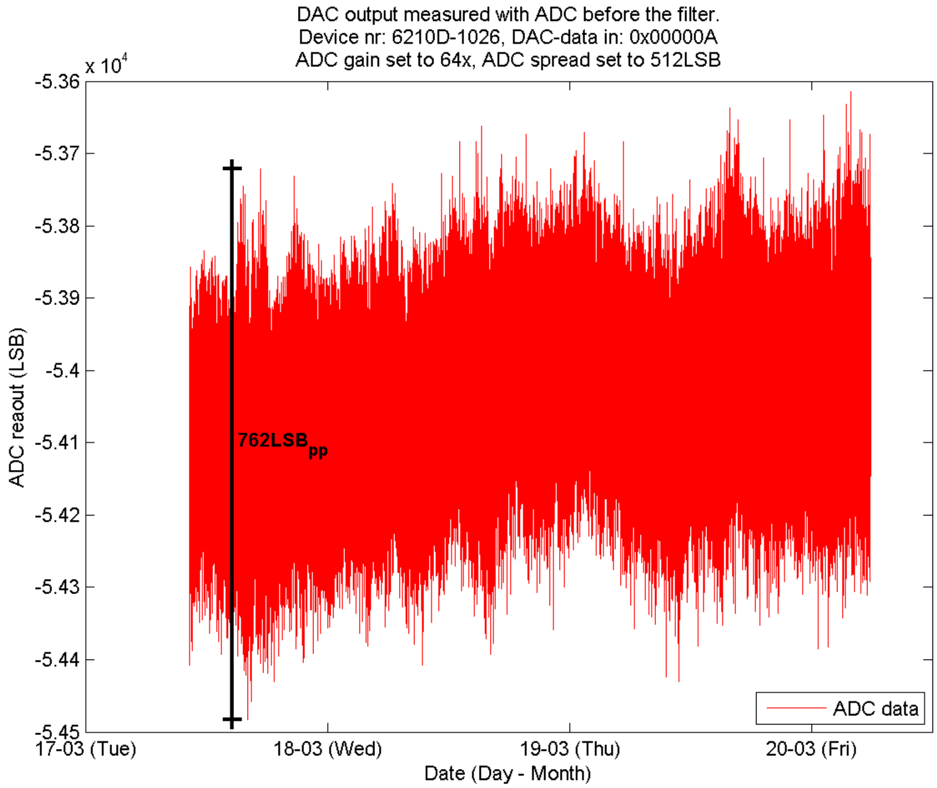


Figure 26: DAC output measured before filter over 3 days (6210D-1026).

From these results is can be seen that noise is the only thing measured. No single events are found when measuring directly at the output of the DAC while the californium source was all the time over the device. Therefore it can be concluded that if the filter has some effect on the SEE that it makes the ability to detect a SEE even better. The filter is located outside the vacuum jar so the filter itself could not be influenced by the radiation source. A possible explanation is illustrated in Figure 27. When a SEE is a very short effect with respect to the sample speed of the ADC a filter reduces the height of the pulse but increases the length. The chance of being detected is therefore with a filter much higher then without a filter. The relative slow sample rate of the DAC, and the ADC, makes this device more SEE tolerant.

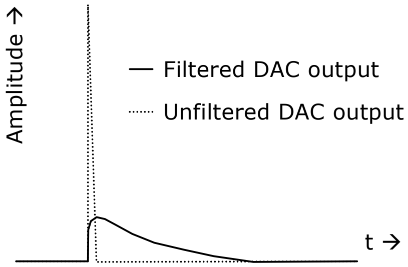


Figure 27: Possible impact of the filter between the DAC and ADC.

3.2 Digital register detections (SEU)

The digital registers are also read back via the software script. In case the read value is not as expected it could be explained as a SEU. However, during all readouts with a ^{252}Cf source and in vacuum none of the read digital values is read back false. In total more than 18E6 blocks were read. Each block contained 8 digital parameters (see also Table 1). This means that more than 144E6 values were read without being effected by the californium source.

3.3 Latch-up detections (SEL)

During almost the complete period the latch up detection circuit was active inside the FPGA. Each millisecond, the latch up detection measures the level of the supply current. When the measured value was 1.5 times the nominal value this should have been detected as a SEL and the device should have been reset. The complete SEE testing period in which a californium source was placed over the device lasted for 13 days. Within this period SEL was never detected on each of the three power supplies. This means that each supply current was checked for 1E9 times within this period and none of those values was outside the limits.

4 Penetration depth

Metal layers inside a chip are known to have the ability to act as a shielding. In order get a feeling about the shielding power of those metal layers, some simulations are performed. The simulations are done with the SRIM tool (can be downloaded from SRIM.com). This tool calculates the penetration depth of a specific element into a given matter or into a stack of layers from different matter. The number of layers inside the simulation tool is limited. Therefore the chip layers of equal matter are grouped together. This can cause some minor effects in the direction of the penetration curves.
The theoretical penetration depth of ^{252}Cf in the CASE setup is approximately 14.2µm into Si (Silicon) (see also [AD2]). The spontaneous fission products of ^{252}Cf are of main interest for SEE. For the simulation Rb (Rubidium) is used for the ion penetrating the silicon. The atomic weight is set to 100amu with an energy level of 80MeV. The angle of incidence is set to 0°. These settings are likely to be a real product from fission of ^{252}Cf. The simulation uses the stack listed in Table 6 to calculate the penetration depth without metal shielding the sensitive layer of the chip.

Layer name	Width (µm)	Density (g/cm³)	Symbol	Name	Atomic number	Weight (amu)	Atom stoichiometric (%)
Si3N4 (pass3)	0.7	1.581	Si	Silicon	14	28.08	42.8571
			N	Nitrogen	7	14.00	57.1429
SiO2 (pass 2 + 1)	1.5	1.7243	Si	Silicon	14	28.08	33.3333
			O	Oxygen	8	15.99	66.6666
SiO2 (no metal)	7.8	1.7243	Si	Silicon	14	28.08	33.3333
			O	Oxygen	8	15.99	66.6666
---- Sensitive layer ----							
Si	>10	2.3212	Si	Silicon	14	28.08	33.3333

Table 6: Stack without metallisation layers.

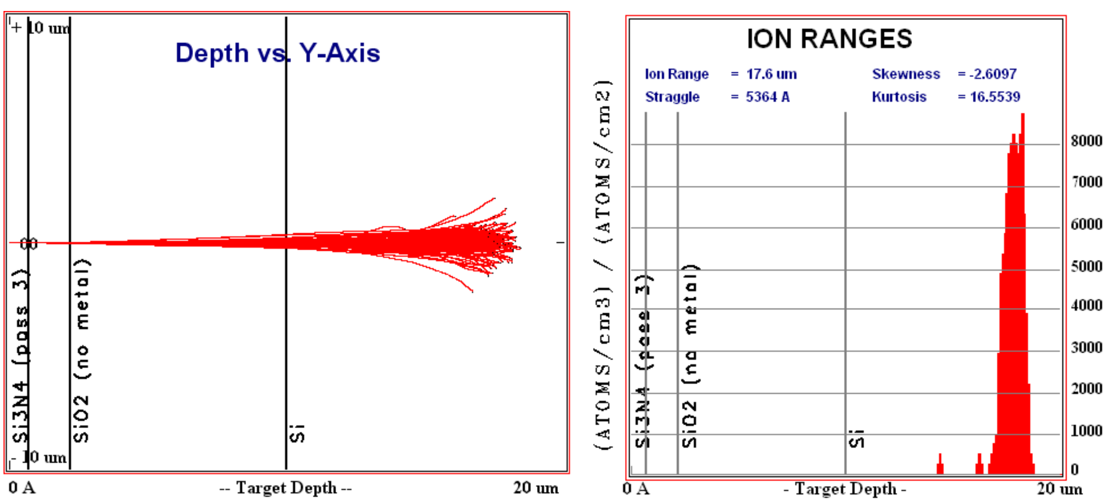


Figure 28: Penetration simulation with Rb (100amu, 80Mev) without metallisation layers.

The simulation result can be seen in Figure 28. The left picture shows the depth of the ion inside the silicon vs. the Y axis. The sensitive layer of the chip is located in the silicon layer just next to the silicon oxide layer. The figure at the right shows a histogram of the simulation. In this plot it can be clearly seen that the average penetration depth with these settings is about 17.6µm. This simulated depth is slightly larger than the depth listed in documents (for example [AD2]). This could be explained if the chosen simulation settings differ slightly from reality. For this experiment this is however not a major problem due to the fact that the simulation is used to determine if metallisation layers can indeed cause a significant shied inside a CASE test environment.
The chip has also several metal layers made of aluminium. The maximum number of layers inside the SHAMROC DAC is 6. The chance that the sensitive layer is blocked by 6 layers at the same position is very small. However, to get a feeling about the worst case blocking power of these layers, the simulation is setup with a metal thickness equal to those 6 layers. The SiO2 layer is decreased in thickness due to these metal layers. Table 7 shows the stack of the simulation when 6 metal layers are blocking the sensitive layer.

Layer name	Width (µm)	Density (g/cm³)	Symbol	Name	Atomic number	Weight (amu)	Atom stoichiometric (%)
Si3N4 (pass3)	0.7	1.5810	Si	Silicon	14	28.08	42.8571
			N	Nitrogen	7	14.00	57.1429
SiO2 (pass 2 + 1)	1.5	1.7243	Si	Silicon	14	28.08	33.3333
			O	Oxygen	8	15.99	66.6666
6 metal layers	3.66	2.702	Al	Aluminium	13	26.98	100.00
SiO2 (no metal)	5.00	1.7243	Si	Silicon	14	28.08	33.3333
			O	Oxygen	8	15.99	66.6666
---- Sensitive layer ----							
Si	>10	2.3212	Si	Silicon	14	28.08	33.3333

Table 7: Stack with 6 aluminium metallisation layers above SiO2.

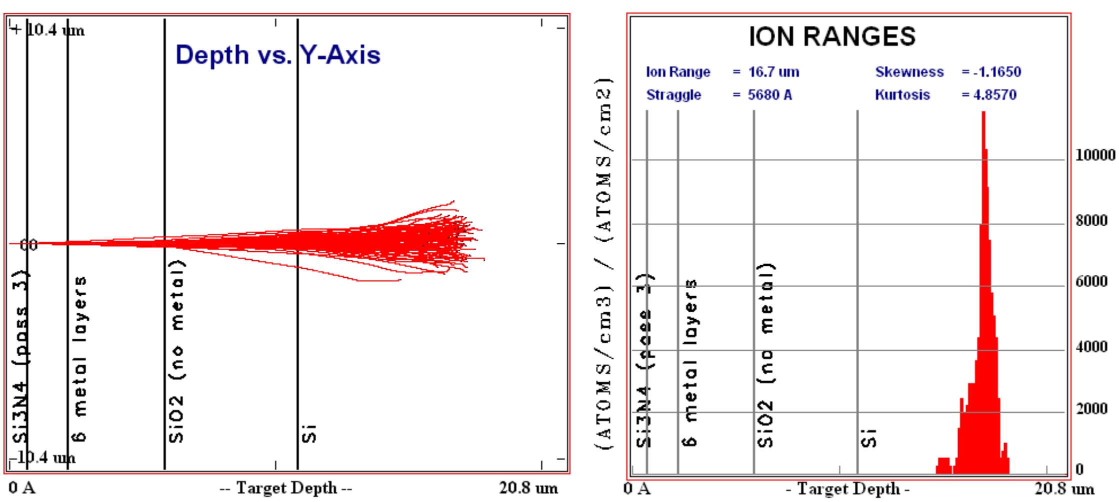


Figure 29: Penetration simulation with Rb (100amu, 80Mev) with aluminium metallisation layers.

Figure 29 shows the results from the simulation. It can be seen that these results do not differ much from the results found with the simulation without the aluminium layer. This simulation does not prove the theory that metal layers act as a shield within a SEE test. Some chip production processes (for example ≤90nm) have copper layers instead of aluminium. The metallisation layer can also be simulated as if it is manufactured in copper. The new stack used for this simulation is listed in Table 8. A clear difference between copper and aluminium is the density. For copper the density is 8.92 g/cm³ while aluminium is only 2.7 g/cm³.

Layer name	Width (µm)	Density (g/cm³)	Symbol	Name	Atomic number	Weight (amu)	Atom stoichiometric (%)
Si3N4 (pass3)	0.7	1.5810	Si	Silicon	14	28.08	42.8571
			N	Nitrogen	7	14.00	57.1429
SiO2 (pass 2 + 1)	1.5	1.7243	Si	Silicon	14	28.08	33.3333
			O	Oxygen	8	15.99	66.6666
6 metal layers	3.66	8.9200	Cu	Copper	29	63.54	100.00
SiO2 (no metal)	5.0	1.7243	Si	Silicon	14	28.08	33.3333
			O	Oxygen	8	15.99	66.6666
---- Sensitive layer ----							
Si	>10	2.3212	Si	Silicon	14	28.08	33.3333

Table 8: Stack with 6 copper metallisation layers above SiO2.

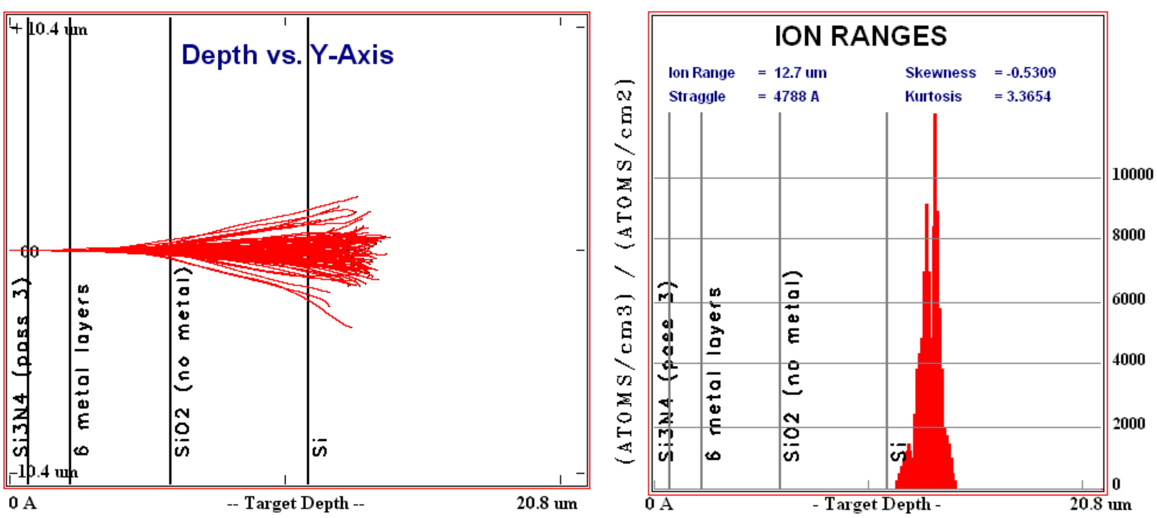


Figure 30: Penetration simulation with Rb (100amu, 80Mev) with copper metallisation layers.

The simulation results in Figure 30 show the blocking power of copper metallisation. This is much higher than the blocking power of aluminium. Although the simulated penetration depth (12.7µm) is still slightly higher then the sensitive layer (10.86µm) it will become a significant problem when using such fabrication processes. Since also the penetration depth simulated without metallisation layers is higher than the theoretical depth.

This simulation proves that metal layers can, in some cases, indeed act as a shielding for SEE (CASE) tests. The influence of the (copper) metal layers is for major concern inside the CASE environment because the energy of the radiation particles is limited. Inside an accelerator the energy levels are much higher and the blocking effect is much less significant because of the higher penetration depth. In space the effect is even negligible due to the enormous energy levels where particles even penetrate trough the entire space craft. The use of copper layers will therefore not improve the real sensitivity of the chip, but only worsen the ability to determine the sensitivity at the test facilities on earth.

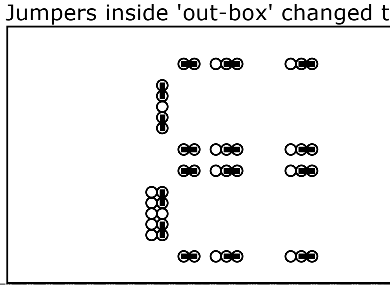
5 Conclusions

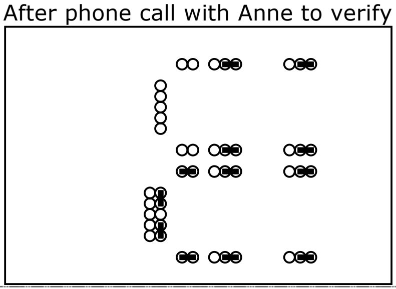
From the measurements done at the CASE setup several conclusions can be drawn. The conclusions are based on two samples, with a high number of readouts per sample. This provides a good confidence in the outcome of this test.

- SEL:
  Single Event Latchup has not been detected during the complete test. All supply currents were individually measured for $1 \cdot 10^9$ times over a period of 13 days.
- SEU:
  Single Event Upset has also not been detected during the CASE test. In total $144 \cdot 10^6$ register values are checked. None of those measurements is found to be different with respect to the desired value.
- SET:
  Single Event Transients are found but only two detections can be categorized as possible SET. During such a transient the output of the DAC changed only a little bit (< 40 LSB) from the mean output value. The fact that we do not detect a lot of transients can be explained by the fact that the DAC and the ADC are working at a low sample rate. A single event effect like a transient is a very fast phenomenon and can easily be spread out over the sample period. This seems one of the possible explanations why such little effect is observed during the high number of readouts.
- Penetration depth:
  A possible explanation why the number of SEE is such low could be the blocking power of metal layers. But this blocking power is very small when using aluminium. This metal is used for the metal layers in the SHAMROC ASIC. The SHAMROC ASIC is made in a 0.18 micron CMOS technology. Processes which have a minimum feature size of 90nm or smaller use copper for the metal layers, and in that case the metal will reduce the penetration depth significantly.

Appendix A

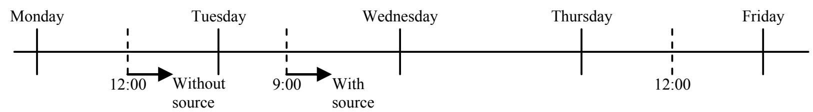
CASE test action log.

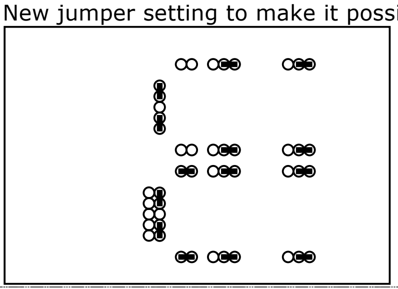
Date / time	Action
05-03-2009	(Rob Wolfs)
8:00	Started packing of DAC test setup at SRON
10:00	Transport to ESA-ESTEC
11:30	Arrival at ESA-ESTEC radiation lab
12:30	Test setup ready for initial tests. - Power supply -> oke - Clock -> oke - Dev1 communication -> oke - Dev2 communication -> FAIL Reason: jumper placement at barrier was wrong. Due to altered settings for offset test by Anne and Yijun. After correction of jumper placement, dev2 communication oke.
12:40	Started first time CASE test script trough 'kickoff' CASE.pl --devicenr: 6210D-1003 --pack_nr: 30000 (30 minutes) --block_nr: 1
13:15	Test finished without SEE detections. - Δt between 0.06sec and 0.09sec (column 2 of full file) - ADC mean readout -4176LSB (column 28 of full file) - Max_pp = 19LSB (difference between max and min value in ADC readout) Readout value does not match the expected value -> check jumper settings towards ADC!
13:30	Pause controllers Connect scope to output of ASIC 1 (three input jumper outside 'out-box') Toggle signal with 'SET_register'. -> chip seems to operate oke Jumpers inside 'out-box' changed to default position: 
13:45	Kickoff CASE.pl (6210D-1003, 500, 1) No heaters active. 1 SEE detection found on ADC readout. - Δt between 0.06sec ±0.002sec (column 2 of full file) - ADC mean readout 3.98947E6LSB (column 28 of full file)

Date / time	Action
	Readout seems to be oke with these new settings. SEE is detected on first readout values (settling problems, no issue!)
13:50	Started controllers, and housekeeping. CASE.pl script adjusted so all information necessary is logged inside the header of the file. - FPGA version - Ambient temp. - Voltages dev1 - Currents dev1 - SEL pwr down time - SEL Reset time - SEL Inhibit time - Device_nr - Nr pack - Nr block - Coll names - Coll unit - Max nr lines inside full file - Start time test - Socket selection - ADC gain value - ADC freq. select - ADC filt. select - SEE ADC spread - SEE ADC filt. length - SEL ref limit - SEL 1v8 limit - SEL 3v3 limit - SEL cnt_set.
14:30	Started new script with kickoff. CASE.pl (6210D-1003, 30000, 1)
15:02	Finished. 1 SEE detected on first ADC readout.
15:06	Kickoff CASE.pl to 'improve' file headers (some small mistakes) (6210D-1003, 50, 1)
15:07	Kickoff CASE.pl (6210D-1003, 50, 1)
15:10	Files and readout looks oke now Changed device 6210D-1003 to 6210D-1026 Kickoff: boardpower.pl (select: power off)
15:14	Kickoff: boardpower.pl (select: power on)
15:15	Kickoff CASE.pl (6210D-1026, 3000, 1) (approx.: 1000 readouts per minute) Output seems to be oke for this new sample. After phone call with Anne to verify the jumper functions, they are changed: 
15:35	Started measurement without source (dry run). Test from 15:30 until 9:30 will take 18 hours. 1000000 samples. Test outside EGSE terminal. > perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 500000 --block_nr = 1; perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 500000 --block_nr = 1

Date / time	Action
06-03-2009	(Rob Wolfs and Bart van Kuik)
9:46	Arrival at ESTEC No errors logged. Checking files: - File 20090305_153513_000_... - Column 2: between 0.06 and 0.14 sec. -> oke - Column 28: mean: 4.03363E6 LSB -> oke pp_noise: 20 LSB -> oke - File 20090305_235350_000_... - Column 2: between 0.06 and 0.16 sec. -> oke - Column 28: mean: 4.03363E6 LSB -> oke pp_noise: 20 LSB -> oke Terminal did not log the script output properly. Will be solved by logging screen output to a file.
10:01	Start script from terminal including file logging: > perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 50 --block_nr = 1 | tee CASE_20090306_log1.log; perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 50 --block_nr = 1 | tee CASE_20090306_log2.log
10:55	Removed LID from DAC 6210D-1026 Changed CASE.pl: turn OFF SEL_detection at the end of script removed. Reason: If SEL detection is turned OFF and a SEL occurs after the end of the script, this might damage the device.
10:56	Kickoff CASE.pl (6210D-1026, 50, 1) Note: No lid on the device
10:56	Put source (^{252}Cf) over device with a device source distance of approximately 2.5cm. According to Bob Nickson this will result in approximately 1000 partials per minute.
10:57	Start to pump the vacuum jar Start readout via kickoff CASE.pl (6210D-1026, 30000, 1) 1 SEE detected on ADC readout. - 4033597 (limit 4033597.7) at readout nr 19050 (1140.09 sec.) Most likely no SEE. (But cannot be ignored as well.)
11:43	Added two extra header lines to CASE.pl Digital write registers and their digital settings.
11:46	Tested script via kickoff CASE.pl (6210D-1026, 50, 1) Modification -> oke
11:58	Start for 1 hour (60000 readouts) with source and vacuum at $4 \cdot 10^{-2}$ mBar. Start from terminal: > perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 60000 --block_nr = 1 | tee CASE_20090306_log3.log
13:10	No errors detected. Only some minor fluctuations in ADC read after 360 sec. up to 600 sec.
13:18	Start of a test which lasts over the weekend. Start 13:30 (06-03-2009)

Date / time	Action
	End 10:00 (09-03-2009) Total: 69Hrs, 4140000 readouts. -> 8 x 500000 = 4000000 readouts. Expected end time: 07:30 09-03-2009 Started from terminal: > For i in $(seq 8); do perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 500000 --block_nr = 1 \| tee CASE_20090306_log4_$i.log;done
09-03-2009	(Rob Wolfs)
9:20	Arrival at ESTEC, test is finished. Files generated by the script: 20090306_131819_000_... -> CASE_20090306_log4_1.log -> 2 ADC detections 20090306_213558_000_... -> CASE_20090306_log4_2.log -> 1 ADC detections 20090307_055339_000_... -> CASE_20090306_log4_3.log -> 4 ADC detections 20090307_141125_000_... -> CASE_20090306_log4_4.log -> 1 ADC detections 20090307_222909_000_... -> CASE_20090306_log4_5.log -> 4 ADC detections 20090308_064721_000_... -> CASE_20090306_log4_6.log -> 9 ADC detections 20090308_150507_000_... -> CASE_20090306_log4_7.log -> 2 ADC detections 20090308_232254_000_... -> CASE_20090306_log4_8.log -> 0 ADC detections -> conclusion from these files: If an SEE had taken place it was only during CASE_20090306_log 4_6.log. Therefore 1 detection within 4000000 readouts.
10:15	Vacuum jar removed. Turn off SEL detection Kickoff boardpower.pl (select power OFF) Turn off controllers. Do NOT take out the device with bare hands. It is slightly radioactive.
10:30	Packed datafiles of last weekend into a gzip file.
10:38	Changed from 6210D-1026 to 6210D-1025 Board power on and start controllers
10:42	Kickoff CASE.pl (6210D-1025, 500, 1) Chip works oke, without SEE detections. Timing -> oke ADC value -> oke
10:50	Cleared housekeeping plots inside EGSE server.
10:58	Start readout (without vacuum jar) for 60000 reads via terminal. Measure data towards the chip over the SPI interface (CASE_20090309_log5.log) > perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1025 --pack_nr = 60000 --block_nr = 1 | tee CASE_20090309_log5.log Measurements show that first data is read from the registers, second the ADC is read, and third the registers are written. This means that what the read write sequence is according to the desired approach.

Date / time	Action
	Test plan for new sample:  If Bob Nickson can 'move' the source over the device at 9:00 tomorrow the test can run for three days (72 hours) The results can be analysed over 21 hours without a source but inside vacuum, and 51 hours with source.
11:39	Source at side of PCB, therefore device is free of radiation! Started to pump the vacuum jar Started measurement for 72 hours (4320000 reads) > changed number to 4500000 samples to have a multiple of 500000. The total measurement time will therefore be 75 hours.
11:45	Start within a terminal: > For i in $(seq 9); do perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1025 --pack_nr = 500000 --block_nr = 1 \| tee CASE_20090309_log5_$i.log;done
10-03-2009	(No SRON personnel, only Bob Nickson)
9:00	Source moved over device.
12-03-2009	(Rob Wolfs)
11:00	Arrived at ESTEC
11:04	Test still running. Current readout at 253900 samples. Files generated by the script: 20090309_114503_000_... -> CASE_20090309_log5_1.log -> 1 ADC detections 20090309_200716_000_... -> CASE_20090309_log5_2.log -> 3 ADC detections 20090310_042931_000_... -> CASE_20090309_log5_3.log -> 19 ADC detections 20090310_125203_000_... -> CASE_20090309_log5_4.log -> 1 ADC detections 20090310_211444_000_... -> CASE_20090309_log5_5.log -> 2 ADC detections 20090311_053736_000_... -> CASE_20090309_log5_6.log -> 1 ADC detections 20090311_140047_000_... -> CASE_20090309_log5_7.log -> 2 ADC detections 20090311_222409_000_... -> CASE_20090309_log5_8.log -> 2 ADC detections 20090312_064747_000_... -> CASE_20090309_log5_8.log -> 1 ADC detections -> conclusion from these files: In file 20090310_042931_000_... the source was placed over the device. The movement of the source has an influence on the measured analog signal. Besides these movement detections, no major SEE detections took place.
	Next options for measurements: - Measure with negative digital setting (data_in = 0xAAAAAA) - Measure without the analog filter (opt_filter = 0)
12:21	Stopped the measurement manually (ctrl-C) after 330812 samples. Reason: No real errors found in first files, better to change the setup and start a new test. Changed CASE.pl -> write value towards dev_data_in to 0xAAAAAA (was 0x555555) Opt_filter = 0 (was 1).

Date / time	Action
12:28	Kickoff CASE.pl (6210D-1025, 500, 1) Errors in digital communication. Setting is 'converted' to negative value where the limit is still a positive one. -> update script to try to solve this.
12:42	Kickoff CASE.pl (6210D-1025, 500, 1) Still not oke, also analog value is not measured correct.
12:51	Changed jumper settings and hard coded the limits for the digital part. New jumper setting to make it possible to measure before the analog filter: 
12:53	Short run via kickoff CASE.pl (6210D-1025, 50, 1) -> results as expected.
12:57	Start CASE.pl via terminal. > perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1025 --pack_nr = 60000 --block_nr = 1 | tee CASE_20090312_log6.log
13:57	Found 7detections. Not one of the detections is above 17 LSB. The noise of device 6210D-1025 looks worse in the time plot than the noise measured on device 6210D-1026. In both plots a slow sine wave signal can be seen in the noise pattern. Both devices show jumps inside the noise spectrum, but for device 6210D-1025 these jumps look faster. This is not due to radiation because this also happens when the radiation source is turned away. After phone call with Anne: Try to measure with a large ADC gain and a low output signal of the DAC. It should be possible to detect a low SET better. The gain of the ADC amplifies this SET and LSB upsets to a larger ADC value.
14:33	Changed CASE.pl - value_w[0] = 0x00000A (data_in, was 0xAAAAAA) - ADC_gain = 64 (was 1)
14:43	Kickoff CASE.pl (6210D-1025, 500, 1) -> Due to higher gain, limits also need to be adjusted. Noise smaller than 300 LSBpp. If detection scales with same gain: ±16 x 64 = ±1024. This seems to be a bit large compared to the measured spread. Spread of detection can be adjusted to 1024 (±512)
14:51	Kickoff CASE.pl (6210D-1025, 500, 1)

Date / time	Action
	Modifications up to this point: - measure before filter - ADC gain set to 64 - DAC data set to 0x00000A - ADC spread set to 1024. -> after analysis of the results the spread seem a bit to large.
14:54	Change the spread value to 512. Kickoff CASE.pl (6210D-1025, 500, 1) -> result looks better, if in future this spread still is too large or too small, detection can be 'recalculated' after the test is finished with new limits.
14:58	Start CASE.pl for 30000 readouts from the terminal. > perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1025 --pack_nr = 30000 --block_nr = 1 | tee CASE_20090312_log7.log
	No strange effects measured during this half hour.
15:32	Start a test for a few days via the terminal. > For i in $(seq 13); do perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1025 --pack_nr = 500000 --block_nr = 1 \| tee CASE_20090309_log8_$i.log;done Test will be finished Tuesday 17-03-2009 at 03:50u
15:58	Compress data from previous days and copy to memory stick. 20090309_114503… to 20090312_064747
16:03	Memory stick is full. Moved data from memory stick towards DAC test PC as a backup. Compress and copy 20090305_153513… and 20090305_235350… these files contain data from sample 6210D-1026 without the ^{252}Cf source.
17-03-2009	(Rob Wolfs)
9:15	Arrived at ESTEC Temperature still constant at 38°C over 7 days. Test is finished. Files generated by the script: 20090312_135259_000_….-> CASE_20090312_log8_1.log -> 1 ADC detections 20090312_235641_000_….-> CASE_20090312_log8_2.log -> 1 ADC detections 20090313_082030_000_….-> CASE_20090312_log8_3.log -> 1 ADC detections 20090313_164433_000_….-> CASE_20090312_log8_4.log -> 1 ADC detections 20090314_010853_000_….-> CASE_20090312_log8_5.log -> 1 ADC detections 20090314_093330_000_….-> CASE_20090312_log8_6.log -> 1 ADC detections 20090314_175815_000_….-> CASE_20090312_log8_7.log -> 2 ADC detections 20090315_022312_000_….-> CASE_20090312_log8_8.log -> 1 ADC detections 20090315_104825_000_….-> CASE_20090312_log8_9.log -> 1 ADC detections 20090315_191350_000_….-> CASE_20090312_log8_10.log -> 1 ADC detections 20090316_033923_000_….-> CASE_20090312_log8_11.log -> 1 ADC detections 20090316_120441_000_….-> CASE_20090312_log8_12.log -> 1 ADC detections 20090316_202932_000_….-> CASE_20090312_log8_13.log -> 1 ADC detections

Date / time	Action
	-> conclusion from these files: Each file contains at least 1 SEE detection on the ADC data. This first detection is in all of the files the first readout sample. This detection can therefore be ignored as a possible SEE event. The only detection found is during …log8_7.log
9:47	No errors found in the files. All files look extremely good, even better than with analog filter.
9:48	Turn off SEL detection inside FPGA (set register: sel_sequencer_mode = 0) Turn off boardpower (kickoff Boardpower.pl, select power off)
9:58	Changed sample 6210D-1025 to 6210D-1026 Turn on boardpower (kickoff Boardpower.pl, select power on)
10:03	Saved temperature data. Plot entry 1: Reference, Plot entry 2: Output stage. Saved as: CF252_Temperatureprofile_6210D-1025.csv CF252_Temperatureprofile_6210D-1025.pdf
10:07	Pack data last days and copy to memory stick
10:08	Kickoff CASE.pl, no vacuum and no source (6210D-1026, 500, 1) > no errors found
10:15	Put source over device Start vacuum pump Kickoff CASE.pl (6210D-1026, 500, 1) > working as expected
10:20	Starting long test from terminal: 8x 500000 reads. Test finished 20-03-2009 at 05:00u > For i in $(seq 8); do perl /data/shamroc/scripts/dac/CASE.pl --devicenr = 6210D-1026 --pack_nr = 500000 --block_nr = 1 \| tee CASE_20090312_log9_$i.log;done
20-03-2009	(Rob Wolfs)
10:00	Arrival at ESTEC Arrived at ESTEC Temperature still constant at 38°C over last days, but more noise is measured on this temperature sensor!!! > reason unknown Test is finished. Files generated by the script: 20090317_102021_000_….-> CASE_20090312_log9_1.log -> 44 ADC detections 20090317_184557_000_….-> CASE_20090312_log9_2.log -> 10 ADC detections 20090318_031146_000_….-> CASE_20090312_log9_3.log -> 9 ADC detections 20090318_113747_000_….-> CASE_20090312_log9_4.log -> 35 ADC detections 20090318_200409_000_….-> CASE_20090312_log9_5.log -> 13 ADC detections 20090319_043051_000_….-> CASE_20090312_log9_6.log -> 7 ADC detections 20090319_125751_000_….-> CASE_20090312_log9_7.log -> 43 ADC detections 20090319_212458_000_….-> CASE_20090312_log9_8.log -> 47 ADC detections -> conclusions from these file: It seems that the noise from these measurements is a little bit higher. Therefore more detections are seen. However none of the detections can be recognized as a SEE effect.

Date / time	Action
10:51	Saved temperature data. Plot entry 1: Reference, Plot entry 2: Output stage. Saved as: CF252_Temperatureprofile_6210D-1026.csv CF252_Temperatureprofile_6210D-1026.pdf This data includes the data already stored in the previous file: …_6210D-1025…
10:53	Stopped the CASE experiment
11:33	Packed data. (No memory stick present at the time.)

Appendix B

Final version of CASE.pl

```
#!/usr/bin/env perl
#
#

# Perl modules for EGSE:
use lib "/data/shamroc/scripts";
use strict;
use warnings;
use Time::HiRes qw (usleep time sleep);

use IO::File;

use APID;
use EDB;
use EGSE;
use LOG;
use POSIX;
use DAC;
use UTIL;
use Getopt::Long qw(:config pass_through);

# == sub routines ==================================================================

sub extract_block($$) {
   my ($block_length, $pack_ref) = @_;
   my @block;
   for (my $i = 0; $i < $block_length; $i++) {
     shift @{$pack_ref};
     $block[$i] = shift @{$pack_ref};
   }
   return @block;
}

sub mean(@) { # mean of values in an array
  my $sum = 0 ;
  foreach my $x (@_) {
    $sum += $x ;
  }
  return $sum/@_ ;
}

$| = 1; # Flush STDOUT continuously

# == get commandline options =======================================================
my ($devicenr,
    $nr_pack,
    $nr_blocks);
GetOptions('devicenr=s'    => \$devicenr,
           'nr_pack=i'   => \$nr_pack,
           'nr_blocks=i'   => \$nr_blocks)
    or die("Invalid command line option");
die("Missing option --devicenr") if (!defined($devicenr));
die("Missing option --nr_pack")  if (!defined($nr_pack));
die("Missing option --nr_blocks") if (!defined($nr_blocks));
die("Illegal value for --nr_pack") if $nr_pack < 1;
die("Illegal value for --nr_blocks") if $nr_blocks < 1;

# == variable declaration ==========================================================
my $timestamp = POSIX::strftime('%Y%m%d_%H%M%S', localtime(time()));
my ($start_time, $send_time, $run_time, $meas_time);
  # each block contains 1 wait statement, 1 read packed, 1 write packed
#my $nr_pack = 5;#3300000; #<--RW
```

```
#my $nr_blocks = 1; #<--RW
  # each readout contains a specified number of blocks, so if block is 10 and
  # readout is 20 the total number of readouts per parameter is equal to
  # 10 x 20 = 200
my (@cmdlist_R_dac, @cmdlist_R_others, @cmdlist_W_dac, @cmdlist_rw, @cmdlist);
# header variables
my $fpgaversion;
  #ADC variables
my $opt_adcgain = 64;# analog amplifier inside ADC, gain can be 1 upto 64.
my $opt_adcfreq = 120; # posible values 240, 120, 60, 30, 15, 7.5
# -> ADC freq does not mean sample frequency between two samples
# --> when reading only the ADC the SET frequency is aprox 4 to 5 times higher
#     then the real output frequency!
# --> The Reason for this is that the ADC is opperating in NON-continous mode.
#     It takes four iterations (controlled by ADC itself!) before is reaches
#     its output.
my $opt_filter = 0; # determined by DAC-testboard configuration
# -->(0: before filter, 1: after filter)
my $socket = 1; # socket ID (1: extender postion, 2: heater position)
my ($filt_range, $filt_hLim, $filt_lLim);
   $filt_range = 2E5;
my $length_filt = 10; # length of moving average filter on ADC readout.
my ($ADC_mean, @ADC_old, @ADC_array); #will be determined by the filter mean value
my $ADC_spread = 512;
   $ADC_mean = 4034156; # initial value

my @list_name = ("dev1_data",          "dev1_pwrdwn",      "dev1_testd1",
                 "dev1_testd2",        "dev1_testgrp",     "dev1_testsel",
                 "dev1_codeoverr",     "dev1_dither_clk",
                 "sel_cnt_i0",         "sel_cnt_i1",       "sel_cnt_i2",
                 "sel_sequencer_sts", "madc_conv$socket"                );

my @value_W   = (0x00000A, 0x000000, 0x0f0f0f,
                 0x0070f0, 0x000000, 0x000000,
                 0x000000, 0x000000,
                 undef,    undef,    undef,
                 undef,    undef              );
my @value_R;
my @hLim      = ($value_W[0], $value_W[1], $value_W[2],
                 $value_W[3], $value_W[4], $value_W[5],
                 $value_W[6], $value_W[7],
                 0x000000   , 0x000000   , 0x000000   ,
                 0x000001, $ADC_mean+$ADC_spread/2     );
my @lLim      = ($value_W[0], $value_W[1], $value_W[2],
                 $value_W[3], $value_W[4], $value_W[5],
                 $value_W[6], $value_W[7],
                 0x000000   , 0x000000   , 0x000000   ,
                 0x000001, $ADC_mean-$ADC_spread/2     );
if ($value_W[0] > ((2**23)-1)) {
  $hLim[0] = -5592406;
  $lLim[0] = -5592406;
}
my @SEE_count;

my $SEL_pwrdowntime = 200000;
my $SEL_resettime   = 199995;
my $SEL_inhibittime = 5;
my $SEL_cntset      = 1000;
my $SEL_reflimit    = 1920; #negative gain in path!
my $SEL_1v8limit    = 2816;
my $SEL_3v3limit    = 816;


# maximum nr of lines per file. Each block of a pack is grouped per file!
my $max_lines = 500000;
my $file_index = 0;
my $lines_stored = 0;
```

```
my $filedir = "./dacdata/RW/$timestamp/";
mkdir $filedir;
my $filename_F = sprintf("%s%s_%03D"."_CASE_full_file.txt",
                          $filedir,
                          $timestamp,
                          $file_index);
my @filename_E_list;
my @FH_array;

# == connect to server ==============================================================
EGSE::connect();             # connect script to EGSE
EGSE::set_timeout(3, 4);     # change timeout
EDB::connect();              # connect script to EngDB

# F: Full file;
open(F, '>', $filename_F)
  or die("Can not open output file $filename_F ($!)");
F->autoflush(1);

# == Make Header lines in each file with start information about the test. =========
my @hdrs; # only small capital as header tag!!!

# FPGA version
(undef, $fpgaversion) = EGSE::get('dac:version');

push(@hdrs, 'date_time'        => $timestamp,
            'fpgaversion'      => sprintf('%08x', $fpgaversion),
            'devicenr'         => $devicenr,
            'nr_pack'          => $nr_pack,
            'nr_blocks'        => $nr_blocks,
            'max_line'         => $max_lines,
            'socket'           => $socket,
            'sel_tpwrdown'     => $SEL_pwrdowntime,
            'sel_treset'       => $SEL_resettime,
            'sel_tinhibit'     => $SEL_inhibittime,
            'sel_cntset'       => $SEL_cntset,
            'sel_reflimit'     => $SEL_reflimit,
            'sel_1v8limit'     => $SEL_1v8limit,
            'sel_3v3limit'     => $SEL_3v3limit,
            'adcgain'          => $opt_adcgain,
            'adcfilt'          => $opt_filter,
            'adcfreq'          => $opt_adcfreq,
            'see_adcspread'    => $ADC_spread,
            'see_adcfiltlength'=> $length_filt,
            'digital_w_reg'    => join(', ', @list_name[0..7]),
            'digital_setting'  => join(', ', @value_W[0..7]) );

# Housekeeping data
my $v;
($v) = EGSE::get_eng("dac:hk_tdev${socket}_box");
push(@hdrs, 'tempdut' => $v);
($v) = EGSE::get_eng("dac:hk_tref_box");
push(@hdrs, 'tempvref' => $v);
($v) = EGSE::get_eng("dac:hk_tout_box");
push(@hdrs, 'tempconv' => $v);
($v) = EGSE::get_eng("dac:hk_uref_p${socket}");
push(@hdrs, 'voltref' => $v);
($v) = EGSE::get_eng("dac:hk_1v8_a${socket}");
push(@hdrs, 'voltana' => $v);
($v) = EGSE::get_eng("dac:hk_1v8_d${socket}");
push(@hdrs, 'voltdig' => $v);
($v) = EGSE::get_eng("dac:hk_iref_a${socket}");
push(@hdrs, 'currentref' => $v);
($v) = EGSE::get_eng("dac:hk_i1v8_d${socket}");
push(@hdrs, 'currentdig' => $v);
($v) = EGSE::get_eng("dac:hk_i3v3_d${socket}");
push(@hdrs, 'currentio' => $v);
```

```
# add commumn names:
push(@hdrs, 'info_1'      => 'Col names and units only apply on the full file.',
            'info_2'      => 'For SEE files the file name represent collumn 3.',
            'col1_name'   => 'run_time',
            'col1_unit'   => 'sec',
            'col2_name'   => 'measure_time',
            'col2_unit'   => 'sec' );

my $next = 3;

for (my $i=0; $i<scalar(@list_name); $i++){
  push(@hdrs, 'col'."$next".'_name' => 'SEE_count_'.$list_name[$i],
              'col'."$next".'_unit' => 'events');
  $next++;
}

for (my $i=0; $i<scalar(@list_name); $i++){
  push(@hdrs, 'col'."$next".'_name' => 'readout_'.$list_name[$i],
              'col'."$next".'_unit' => 'LSB');
  $next++;
}

push(@hdrs, 'col'."$next".'_name'   => 'ADC_filt_mean',
            'col'."$next".'_unit'   => 'LSB');
$next++;

push(@hdrs, 'col'."$next".'_name'   => 'ADC_filt_hLim',
            'col'."$next".'_unit'   => 'LSB');
$next++;

push(@hdrs, 'col'."$next".'_name'   => 'ADC_filt_lLim',
            'col'."$next".'_unit'   => 'LSB');
$next++;

DAC::write_fileheaders(*F, \@hdrs);

for (my $i=0; $i<scalar(@list_name); $i++) {
  $SEE_count[$i] = 0;
  $filename_E_list[$i] = $filedir.$timestamp."_CASE_SEE_".$list_name[$i].".txt";
  #localize the file glob, so FILE is unique to the inner loop.
  local *FILE;
  open(FILE, '>',"$filename_E_list[$i]")
    or die("Can not open output file $filename_E_list[$i] ($!)");
  FILE->autoflush(1);
  DAC::write_fileheaders(*FILE, \@hdrs);
  #push the typeglobs to the end of the array
  push(@FH_array, *FILE);
}

# == make a readout list ===========================================================
  # set list of parameters which need to be read out from the DAC.
  @cmdlist_R_dac = ("dac:$list_name[0]", undef,
                    "dac:$list_name[1]", undef,
                    "dac:$list_name[2]", undef,
                    "dac:$list_name[3]", undef,
                    "dac:$list_name[4]", undef,
                    "dac:$list_name[5]", undef,
                    "dac:$list_name[6]", undef,
                    "dac:$list_name[7]", undef);


  # set list of parameters which need to be read out from the rest.
  @cmdlist_R_others = ("dac:$list_name[8]", undef,
                       "dac:$list_name[9]", undef,
                       "dac:$list_name[10]", undef,
                       "dac:$list_name[11]", undef,
                       "dac:$list_name[12]", undef );
```

```
  # set list of parameters which need to be set.
  # set all parameters of the chip to be shure that each SEE effect is
  #   only valid for one 100Hz frame.
  @cmdlist_W_dac  = ("dac:$list_name[0]", $value_W[0],
                     "dac:$list_name[1]", $value_W[1],
                     "dac:$list_name[2]", $value_W[2],
                     "dac:$list_name[3]", $value_W[3],
                     "dac:$list_name[4]", $value_W[4],
                     "dac:$list_name[5]", $value_W[5],
                     "dac:$list_name[6]", $value_W[6],
                     "dac:$list_name[7]", $value_W[7]);

   @cmdlist_rw = ('dac:wait_dev_drdy_n', 1,
                   @cmdlist_R_dac,
                   @cmdlist_R_others,
                   @cmdlist_W_dac   );

  @cmdlist = @cmdlist_rw;
  for (my $i = 1; $i < $nr_blocks; $i++){
     @cmdlist = (@cmdlist,@cmdlist_rw);
  }

# == set setting on the FPGA =======================================================
  # write settings to DAC
  print "Set DAC according to write list...\n";
  EGSE::mixed_multiple(undef, @cmdlist_W_dac);

  # Setup science ADC.
  print "Setup ADC ...\n";
  DAC::setup_madc($opt_adcgain, $opt_adcfreq, $opt_filter,
                  $opt_adcgain, $opt_adcfreq, $opt_filter);
  print "sleep for 10 seconds to let the ADC settle...\n";
  sleep(10);

  print "Setup SEL-setection...\n";
  # Start SEL interface (with correct timer settings (400ms = 40 periods))
  DAC::write_reg('dac:pwrdowntime_set', $SEL_pwrdowntime); # 200 ms
  DAC::write_reg('dac:resettime_set', $SEL_resettime);   # 200 ms - 5us
  DAC::write_reg('dac:sel_inhibit_time_set', $SEL_inhibittime);        # 5 us
  # clear the counter by writing (anything) towards them
  DAC::write_reg('dac:sel_cnt_i0', 0); # reference current latch-up
  DAC::write_reg('dac:sel_cnt_i1', 0); # digital current latch-up
  DAC::write_reg('dac:sel_cnt_i2', 0); # IO current latch-up
  # set limits SEL check
  DAC::write_reg('dac:sel_limit_i0', $SEL_reflimit); #negative gain in path!!
  DAC::write_reg('dac:sel_limit_i1', $SEL_1v8limit);
  DAC::write_reg('dac:sel_limit_i2', $SEL_3v3limit);
  # frequency of checking the currents, time in us.
  DAC::write_reg('dac:sel_cnt_set', $SEL_cntset); # 1000us is def. value.
  # turn sel-interface ON
  DAC::write_reg('dac:sel_sequencer_mode', 1);

  print "Select DRDY_n pulse to synchronize... (socket: $socket)\n";
  # select sensitivity for DRDY_n of devive 0
  #  (0 = extender position, 1 = board position)
  DAC::write_reg('dac:drdy_select', ($socket-1));

  #calculate the mean value of the ADC readout. This inshures that low drifts and
  #  power on of the reference are not being detected as SEE effects. This assumes
  #  that a SEE is only a short moment present
open(AVR, '>', $filedir.$timestamp."_ADC_avarage_file.txt")
  or die("Can not open ADC_average file. ($!)");
AVR->autoflush(1);

  for (my $i=0; $i<@list_name; $i++) {
    if ($list_name[$i] =~ /madc_conv/i) {
#      print "calculating average of ADC readouts\n";
      for (my $j=0; $j<$length_filt; $j++) {
```

```
        $ADC_array[$j] = DAC::madc_svalue(EGSE::get("dac:madc_conv$socket"));
        print AVR "$ADC_array[$j]\n";
#         print "$ADC_array[$j] $j\n";
      }
#       print join("\n",@ADC_array)."\n";
      $ADC_mean = mean(@ADC_array);
      $hLim[$i] = $ADC_mean+$ADC_spread/2;
      $lLim[$i] = $ADC_mean-$ADC_spread/2;
      $filt_hLim = $ADC_mean+$filt_range/2;
      $filt_lLim = $ADC_mean-$filt_range/2;
    }
  }
close(AVR);

$start_time = time; # start time of the measurement.
print "start of the measurement: $start_time\n";
# == Readout in packets ========================================================
  for (my $i = 0; $i < $nr_pack; $i++) {

     # make a new file if number of lines is more than max_lines in a file.
     #  This is to prevent that the "full file" becomes to large.
     if (( ((($i)*$nr_blocks)-$lines_stored) / ($max_lines) ) >= 1) {
       print "\nsaving  : $filename_F\n";
       $lines_stored = ($i)*$nr_blocks;
       $file_index++;
       close(F);
       $filename_F = sprintf("%s%s_%03D"."_CASE_full_file.txt",
                             $filedir,
                             $timestamp,
                             $file_index);
       open(F, '>', $filename_F)
         or die("Can not open output file $filename_F ($!)");
       F->autoflush(1);
       DAC::write_fileheaders(*F, \@hdrs);
       print "opening : $filename_F\n";
     }

     $send_time = time;
     # $run_time = time from start of the measurement upto sending a packet
     #   value will increase and can be used as an X-axis value.
     $run_time = $send_time - $start_time;

     my @value_pack = EGSE::mixed_multiple(undef, @cmdlist);
     # $meas_time time it took to get 1 packed from the FPGA
     #   Value will be aprox. constant, unless a SEL is detected by the FPGA.
     my $meas_time = time - $send_time;

     print "\r$run_time ($i of $nr_pack) SEE counts: ".join(" ",@SEE_count)."..." ;

     while(@value_pack) {
       # split packs up in blocks
       @value_R = extract_block(scalar(@list_name), \@value_pack);

       # check for each value in a block if there is an uncorrect value (SEE)
       for (my $i=0; $i<scalar(@value_R); $i++) {
         # convert only ADC data to a signed value
         if ($list_name[$i] =~ /madc_conv/i) {
           $value_R[$i] = DAC::madc_svalue($value_R[$i]);
         }
         if (($value_R[$i] < $lLim[$i]) or ($value_R[$i] > $hLim[$i])) {
           $SEE_count[$i]++;
           # SEE detected!, make file handler to look at corresponding file:
           my $FH = $FH_array[$i];
           print $FH join("\t",$run_time,
                               $meas_time,
                               $SEE_count[$i],
                               $value_R[$i],
                               $hLim[$i],
```

```
                              $lLim[$i])."\n";
          # change limits for SEL to new readouts (SEL have been detected)
          if ($list_name[$i] =~ /sel_cnt/) {
            $lLim[$i] = $value_R[$i];
            $hLim[$i] = $value_R[$i];
          }
        }
        # calculate average value of the ADC, and store this as new limits
        if ($list_name[$i] =~ /madc_conv/i) {
          if (($value_R[$i] < $filt_lLim) or ($value_R[$i] > $filt_hLim)) {
            push(@ADC_array,$ADC_mean);
          } else {
            push(@ADC_array,$value_R[$i]);
          }
          shift(@ADC_array);
          $ADC_old[0] = $ADC_mean;
          $ADC_old[1] = $hLim[$i];
          $ADC_old[2] = $lLim[$i];
          $ADC_mean   = mean(@ADC_array);
          $hLim[$i]   = $ADC_mean+$ADC_spread/2;
          $lLim[$i]   = $ADC_mean-$ADC_spread/2;
          $filt_hLim  = $ADC_mean+$filt_range/2;
          $filt_lLim  = $ADC_mean-$filt_range/2;
        }
      }
      # print each block as a row inside the 'full file'
      print F join("\t",$run_time,
                        $meas_time,
                        @SEE_count,
                        @value_R,
                        @ADC_old)."\n";
    }
  }

  #close files
  close(F) or die("Couldn't close file: $!");
  for (my $i = 0; $i<scalar(@FH_array); $i++) {
    close($FH_array[$i]) or die("Couldn't close file: $!\n");
  }
  print "\n";

#  # turn sel-interface OFF
#  DAC::write_reg('dac:sel_sequencer_mode', 0);

  print "Data is stored in directory : \n\t- $filedir\n";
  print "All data (starting from 000): \n\t- $filename_F\n";
  print "SEE detections in           : \n\t- ".join("\n\t- ",@filename_E_list)."\n";


EDB::disconnect();
EGSE::disconnect();
```