

TESTING NEUTRON-INDUCED SOFT ERRORS

Participants:

N. Vijaykrishnan, Prof. of Computer Science and Eng. Dept.
K. Ünlü, Prof. of Mechanical and Nuclear Eng. Dept.
Y. Xie, Prof. of Computer Science and Eng. Dept.
M. J. Irwin, Prof. of Computer Science and Eng. Dept.
Ramakrishnan Krishnan, Ph.D. student at Electrical Eng. Dept.
Rajaraman Ramanarayanan, Ph.D. student at Electrical Eng. Dept.
S. M. Çetiner, Ph.D. student at Mechanical and Nuclear Eng. Dept.

Services Provided:

Neutron Beam Laboratory, Reactor Core -Penn State Breazeale Nuclear Reactor

Sponsor:

NSF, Department of Energy, INIE Mini Grant

INTRODUCTION

Soft errors are transient circuit errors caused due to excess charge carriers induced primarily by external radiations. Radiation directly or indirectly induces localized ionization that can flip the internal values of the memory cells. Our current work tries to characterize the soft error susceptibility for different chips working at different technology node and operating voltage.

BACKGROUND AND RELATED WORK

Advances in VLSI technology have ensured the availability of high performance electronics for a variety of applications. The applications include consumer electronics like cellular phones and HDTVs; automotive electronics like those used in drive-by-wire vehicles, and million dollar servers used for storing and processing sensitive and critical data. These varied applications require not only higher throughput but also dependability. Even if a microprocessor is shipped without any design errors or manufacturing defects, unstable environmental conditions can generate temporary hardware failures. These failures, called *transient faults*, cause the processor to malfunction during operation time. The major sources of transient faults are electromagnetic interference, power jitter, alpha particles, and cosmic ray neutrons. Studies in [1, 2] have shown that a vast majority of detected errors originate from transient faults. Even a single-bit error may eventually lead to a computation failure. Therefore, managing the soft errors is a critical problem to solve in fully realizing dependable computing.

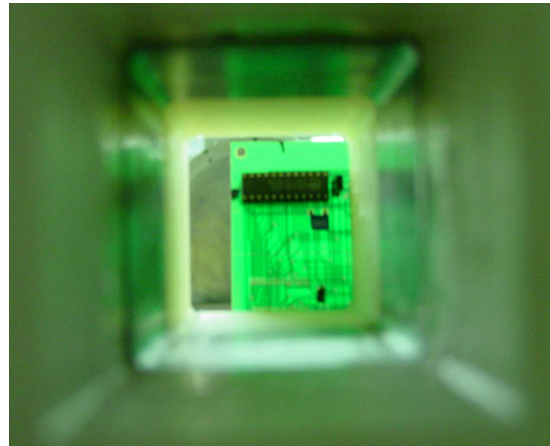


Figure 1. Test chip as seen from the narrow opening in the polyethylene/lead shield.

This study was intended to observe the effect of both thermal and high energy neutrons on Soft Error Rate (SER). In order to investigate the effect of ^{10}B on SER, a thermal neutron beam from a beam port connected to the core of a nuclear reactor is used. This setup is shown in Figure 3. Detailed studies on different memory chips at different voltages and power levels showed the effects of ^{10}B fission caused by thermal neutron absorption on SER in memories [4]. In this report, results from the study of the effect of

thermal neutrons on a Motorola based M68HC11 micro-controller and an Intel PXA270 processor are presented.

Another setup was also built to study the effect of fast neutrons on SER. Therefore, the circuit board for this set-up is placed by the periphery of the reactor core via a vertical standpipe in order to observe the effect of fast neutrons on soft error rate. A schematic drawing of fast neutron irradiation facility at BNR is shown in Figure 2.

EXPERIMENTAL SETUP

Penn State Breazeale Nuclear Reactor was used as the neutron source in the experiments. The maximum rated power of the reactor is 1 MW in the continuous mode, and 2000 MW in the pulse mode. The reactor power is adjusted from 10 W to 1MW observe the soft error rate dependence on neutron flux. No pulse-mode operation has been performed. Figures 1, 2 and 3 show the test chip and the experimental setups. For the beam port that was used in the experiments, the beam tube looks at the D₂O tank to get a well-thermalized beam. The average thermal flux at the exit of the beam port is about 3×10^7 neutrons/cm²sec.

For the test set-up at the reactor core (Figure 3), a long irradiation standpipe is used to hold a bucket that contains the test board and the protection layers. The stand pipe is deep enough to place the board right at the periphery of the reactor. The

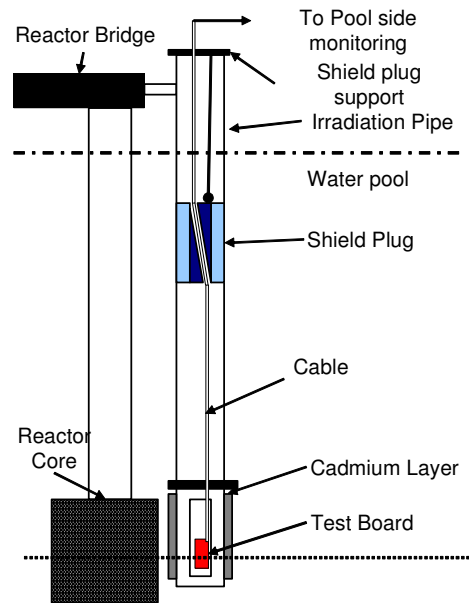


Figure 2: Test set-up at the reactor core

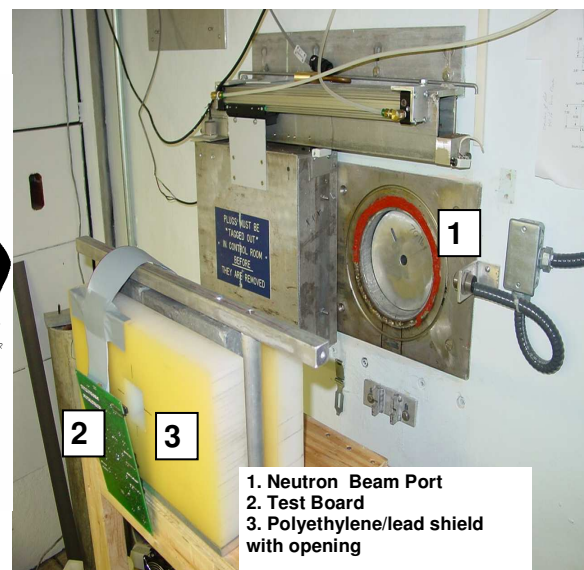
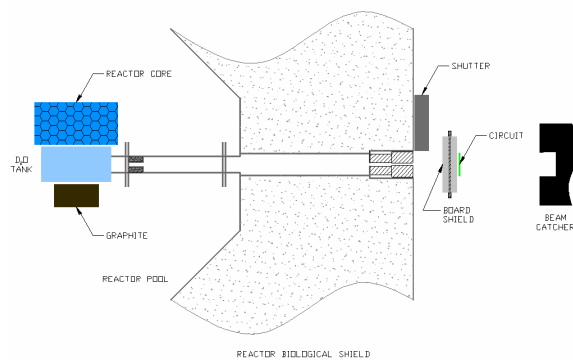


Figure 3. (Left) Simplified layout of the test board, beam tube and the reactor. (Right) Test chip placed in front of the beam tube with the polyethylene/lead shield.

fast neutron flux at the core boundary is 5×10^{12} neutrons/cm²sec, and thermal flux 1.3×10^{13} neutrons/cm²sec at 1-MW steady state reactor operation. The high neutron flux allows for accelerated testing of the phenomenon.

The experimental set-up for the beam port experiments has the test boards' set-up as shown in Figure 3 and interfaced with computer by different means. The M68hc11 microcontroller was used in the form of a prototyping circuit board from HVW technologies [5]. As a prototyping system it is a quick and efficient means of testing code and external circuitry. The board is interfaced through serial port connection and a terminal client is used to program the board for testing.

The PXA270 processor which is part of a Single Board Computer with Debian linux operating system installed on a USB drive is interfaced through a crossover cable and a secure shell client is used to connect to it from a remote host. Also, the debug port messages are also obtained using a separate serial cable and terminal client on the remote host.

The experimental setup for the tests at the reactor core consists of a custom board interfaced with a computer through a GPIB card (from National Instruments). The board itself has off-the-shelf SRAM memory chips. The board is controlled through a LabVIEW interface. The controlling application consists of simple routines to read and write a user specified value across the whole memory. During the readout, it compares the written value to the value in each address.

The circuit board is secured in the bucket that is supported by the stand pipe, and connected to a PC outside using a 35-ft long cable. This configuration allows continuous read-write, and for changing the operating conditions without interrupting the experiment. The selected section of the board is tested on-line multiple times in the actual setup before the reactor is started. The board is exposed to neutron flux after the reactor reaches the stable power level.

RESULTS AND DISCUSSION

The M68HC11 is 8-bit data, 16-bit address microcontroller from Motorola with an instruction set that is similar to the older 68xx (6801, 6805, 6809) parts. The E9 series was used for the testing the microcontroller. The M68HC11 is optimized for low power consumption and high-performance operation at bus frequencies up to 4 MHz. The CPU has two 8 bit accumulators (A & B) that can be concatenated to provide a 16 bit double accumulator (D). Two 16 bit index registers are present (X & Y) to provide indexing to anywhere in the memory map. More details on the architecture and programming of the microcontroller can be found at [6].

RS232 serial cables served as a serial interface with Hyper-terminal being used for communicating with the buffalo 3.4 operating system loaded on the microcontroller. The micro-controller has an on-chip SRAM. The pseudo-codes that were used to program the micro-controller were stored in the portion of the SRAM available for the user (address space 0000-0050 Hex). The data used by the pseudo-codes were stored in the 512-byte on-chip EEPROM. After some initial tests, two different testing scenarios were devised on the following criteria. The first one was to repeat a single mathematical operation multiple number of times. The next criterion was giving sufficient delay for the result of the computation to be stored in the registers. Also, the operations were done on data which were read exhaustively from the available on-chip memory so that any errors in the memory could manifest at the output. Finally, the data was transmitted through the serial port and the delay cycles were written to make sure the read-write time was negligible compared to the time for which the data is operated/stored in the micro-controller.

Among the two testing scenarios, the first one was used to read data from the EEPROM and perform additions before outputting the data on to the serial port to be captured by the hyper-terminal. The second was used to perform similar operations on the SRAM memory. Both had a delay loop incorporated in it to make the read-write time insignificant. It is important to note that both the memories were exhaustively read from and the data operated by the pseudo-codes. There were no errors found after the codes were run for 150 minutes each. The reason for this could be due to the high voltage (5V) and capacitances leading to

no memory flips or pulse strong enough to get latched in the accumulator as the micro-controller was fabricated in 2 micron HCMOS technology [7]. Also, since the accumulators and data-path occupies a very small area in the microcontroller, the neutron hit probability will also be very less.

The PXA270 microprocessor architecture is much more complicated than the micro-controller that was tested earlier making it far more difficult to write programs that could utilize the memory. Thus, a simple matrix multiplication program was compiled and run on the processor while SER measurements were made during the exposure to neutron radiation. In these programs, first the source matrices were created after which a suitable delay was introduced after which the actual multiplication of the two source matrices was done and the source and the result matrices were stored in a file. Two identical matrix multiplications were performed one after the other and then the results were compared for errors. The delay values were chosen such that the read/write to the files was negligible. Four preliminary experimental runs were performed and the results are presented in Table 1. Here a USB flash drive was used for Test 1 and Test 2 while a hard drive was used for Tests 3 and 4. Both Tests 2 and 4 were run at scaled down voltage and speed.

Table 1: Preliminary Test Results

	Test 1	Test 2	Test 3	Test 4
Run Time (minutes)	106	154	87	143
Total Failures	8	16	10	25
Segmentation + bus errors	1 + 0	5 + 0	2 + 0	9 + 5
Crashes during boot	1	2	1	2
Irrecoverable crashes	6	5	4	9
Other errors	0	4	3	0
Other errors include connection lost, debug port crashes only, file system corruptions etc				

As seen from the table, apart from the irrecoverable crashes, segmentation faults are the most common types of failures that occur. These segmentation faults and also many of the irrecoverable crashes are caused due to an error called d-cache parity error. Figure 4 (a) shows the debug port message when this error occurs. As the name indicates, this error occurs due to difference in parity while reading back, indicating a bit flip that could have occurred in the data cache. The corruption of the data is thus prevented because of the inherent property of the kernel to issue segmentation faults when this error occurs.

Another type of failure which could have occurred due to a logic error resulted in file system corruption. This kind of failure occurs when the inode fields for different files are updated on to the USB memory. These inode fields store file information such as size, type, date of creation etc. As each file in the USB memory is being modified when a process is running, the processor updates the information in these inode files. Thus the error could have occurred in the inode fields which could be due to a logic error as the errors in the cache leads to segmentation faults and hence are prevented. It was also noted that these file system corruptions occurred when hard drives were used as the USB memory which could mainly be due to a relatively larger number of write backs/updates of inode fields in the case of hard drives.

A single test run was also performed for the memory board at the reactor core using the standpipe setup described earlier. After a 10 minute exposure at 900 MW of reactor power, data was read back from the memory. The data read back from all the memory locations were zeros. The errors were persistent even after repeated read-writes after the exposure. Thus permanent damage has been caused by the thermal neutrons present in the radiation and precautions are being taken to avoid such damages for the future.

FUTURE WORK

The tests on the PXA270 processor have some interesting results that need further investigation. For example, the d-cache error can be prevented by changing the settings on the kernel thus allowing the

process to continue even after data corruption in the memory. Since the result data is stored for further analysis, any error in the result data that could have been caused by the d-cache parity error can be easily detected.

The reactor can be pulsed for a very short duration of time, around 10 msec at FWHM, at which it generates a fast flux of about 1×10^{16} neutrons/cm²sec at the core periphery. This amounts to about a ten order of magnitude increase in the fast flux. The time duration is very limited, yet the amount of fast flux is immense. That might also reduce the experiment times significantly and help perform more tests with various technologies and designs.

```

[segfault
Unhandl'd fault: dcache parity error (0x418) at 0x4001b000
pgd = c3a8c000
[4001b000] *pgd=a352b031, *pte=a2b920ff, *ppte=a2b9203f
Internal error: Oops: 0 [#1]
Modules linked in:
CPU: 0
pc : [] lr : [] Not tainted
sp : c2427d2c ip : 0000075c fp : c2427de4
r10: 00000000 r9 : 0080c000 r8 : 00000000
r7 : c037b640 r6 : 20202036 r5 : 35323620 r4 : 20203336
r3 : 33352020 r2 : 00000000 r1 : 4001b854 r0 : c1eb2844
Flags: nZCv IRQs on FIQs on Mode SVC_32 Segment user
Control: 397F Table: A3A8C000 DAC: 00000015
Process a.out (pid: 1374, stack limit = 0xc24261a4)
Stack: (0xc2427d2c to 0xc2428000)
7d20: c1eb2000 00001000 c037b640 00000000 c03033c8

```

```

segrith.cse.psu.edu 66% du -khs bin
426G bin
segrith.cse.psu.edu 67% ll
total 446404348
cr-Sr-S--- 8240 959265076 876099129 32, 50 Oct 2 1997 bin
.
.
drwxr-xr-x 13 root root 4096 Dec 19 2005 var
segrith.cse.psu.edu 68% cd root/samplecodes/test7/
segrith.cse.psu.edu 69% du -khs *
426G a.out
0 err.out
434G matrix_a
458G matrix_b
394G matrix_c
426G matrix_d
434G matrix_e
394G matrix_f
segrith.cse.psu.edu 70% ll
total 3107434033
?---rw---x 11552 892546336 959789109 943207220 Dec 28 1993 a.out
-rw-r--r-- 1 root root 0 Oct 10 15:09 err.out
?---rws--t 13869 909522483 540549173 926166304 Dec 28 1993 matrix_a
?---srw-r-x 11552 943140128 757084720 808726580 Dec 28 1993 matrix_b
?---rwx--x 8246 842276912 540030005 859124013 Feb 11 1987 matrix_c
?---rwx--x 11552 892546336 959789109 943207220 Dec 28 1993 matrix_d
?---rws--t 13869 909522483 540549173 926166304 Dec 28 1993 matrix_e
?---rwx--x 8246 842276912 540030005 859124013 Feb 11 1987 matrix_f
segrith.cse.psu.edu 71%

```

Figure 4 (a): D-cache parity error

Figure 4 (b): File system corruption

REFERENCES

1. J. SOSNOWSKI, *Transient fault tolerance in digital systems*, IEEE Micro, v. 14 No. 1, pp. 24-35, February 1994.
2. D. P. SIEWIOREK, R. S. SWARTZ, *Reliable Computer Systems: Design and Evaluation*, 2nd edition, Digital Press, Burlington, MA.
3. P. HAZUCHA and C. SVENSSON, *Impact of CMOS Technology Scaling on the Atmospheric Neutron Soft Error Rate*, IEEE Transactions on Nuclear Science, v. 47, No. 6, pp 2586-2594, December 2000.
4. V. Degalahal, *Soft errors: Modeling and interactions with power optimizations*, A PhD Thesis in Computer Science and Engineering, Pennsylvania State University, December 2005.
5. www.HVWTech.com
6. M68HC11E Family Data Sheet at: <http://freescale.com/>
7. Daniels, R.G., A participant's perspective, IEEE Micro, v. 16, No. 6, pp. 21 – 31, December 1996