Error in Software fault tolerance

ManishaSingh,RekhaTripathi

Department of Computer Center,AwdeshPratap Singh University,Rewa 486003,Madhya pradesh

Abstract

In this research we compare two approaches in an actual space experiment. Radiation hardening is an effective presently it a costly solution to this problem. Here a moor relation in digital electronics used in space is radiation-induced transient error. Here we assess the effectiveness of software implemented hardware fault tolerance.

1. Introduction

Radiation, can cause transient faults in electronic systems, that is alpha particles and cosmic rays, such type of faults cause errors known as single-event upsets (SEUs). Single event upsets are a major cause of concern in a space environment, and also have been observed at ground level. [2]An example effect in combinational circuit, e.g. an arithmetic logic unit can also lead to incorrect results.[1]

Radiation hardening is a fault avoidance technique which is used for electronic components used in space. However, these components are lag behind todays commercial components, in term of performance. The need for low cost, state-of-the-art high performance computing systems in space has been created a strong motivation for investing new fault tolerance techniques (FTT). Using commercial off the self-component (CTOs) has been suggested for building cheaper and faster systems, and as opposed also to radiation hardened components. Fault tolerance system have limited component of COTs. Shift techniques provides low cost solutions for enhancing the reliability of systems without any changing the hardware.

2. Research setup

The Stanford ARDOS protect [2] is an research that carried out on the computing test-bed of the NRL-801: Unconventional Stellar Aspect (USA) experiment on the advanced research and global observations satellite (ARGOS) that is launched in 1999 February.The ARGOS satellite [3] has a sun-synchronous, with a mission life of three years having 834-kilometer altitude orbit. The aim of the computing test –bed in the USA experiment on ARGOS is the comparative evaluation of approaches to reliable computing in a space environment, including radiation hardening of processors. In this the experiment utilizes 32-bit MIPS R3000compatible processors. The hardboard uses the harries RH3000 radiations hardened set of chip having features a self-checking processor pair configuration and has error detection and error correction(EDEC) Hardware for its 2MB silicon on insulator SRAM memory. The COTS board uses only COTS components and also use 3081 microprocessor from IDT. It has 2MB of SRAM and has no hardware error detection technique except for internal cache memory parity. Vx Works is on both board of operating system. It is possible to update the software on the boards which is based on the results received during the mission, and test different shift technique. This research paper presents preliminary results of our experiment, we continue to collect and analyze error data.

3 Error in Hard board

The hardboard has hardware EDEC, for memory and self-checking processor pair. Moreover the data and address buses having parity bits. Upon a mismatch between the master and the shadow processors, an exception is generated for lead to a system halt and reset uncorrectable memory errors or sayparity errors also lead to system halt.

In this paper several errors have been observed in the hardboard. These errors have been occurred during the execution of two tests i.e. a memory test that checks for a fixed pattern in a memory block, and a program that generates a sine table compares it against a store table. Here four errors occurred in the first program and three errors in the second program. There has also one more exception that led to a system halt. For all other errors, firstly the programs have detected the error, then reported it and continued their execution. That means both the processors master and shadow were in agreement on the errors. So, the errors were not upset in one of the processors. They were not cases of double errors that were not correctable by the EDEC hardware either. We may not be able to pinpoint the source of these errors, but the evidence suggests that they occur in that place which is common for both the processors. In such a place the data buffer between memory and processors.

In comparisons of hardboard error rate is lower than COTS board. This discrepancy may be due to the different SRAM components on two boards, and not due to the different processors.

4. Error in software

4.1 Software implemented EDEC

There is no hardware for EDEC to protect the main memory of the COTS board. In the previous stage of the experiment, we observed that SEUs corrupted the memory, forcing for frequent system resets. We implemented EDEC in software and use periodic scrubbing to protect the code segments of operating system and application programs. This improves the availability of the COTS board. We are able to run the board continuously for more than a month with software EDEC, which as opposed to a few days without software EDEC.

Hundreds of memory bits flips have been observed by running memory which tests on the COTS board the average memory error rate calculated based on these tests and the correction of error done by software EDEC is about 5.5 Upsets/MB-day. It has been analyses that a single practical can affect multiple-bit Upsets(MBUs)[1][4]. In our research, MBUs constitute about 3 percent of the memory errors. Our software EDEC is designed to handle MBUs and it has successfully corrected all the cases of MBUs.

4.2 SoftwareError detection and recovery

Transient errors that occur is any processor can be detected by executing a program multiple times, and compare the outputs produced by each execution. By the programmer or by the operating system duplication can be done at task level, and it can also be done in instruction level, during program compilation. Here we use one more techniques called error detection byduplicated instructions (EDDI) that uses the latter approach computation results are come from master and shadow instructions are compared before writing to memory. In this mismatching the program umps to an error handler that will cause the program to restart.

EDDI technique can only detect some of the control-flow errors. To enhance the detection coverage for this type of error one more technique was developed called control-flow checking by software signatures (CFCSS). CFCSS technique is an assigned signature method where the unique signatures are associated with each block during compilation. For constant operands these signatures are embedded into the program for using the immediate field of instructions. On the time of execution of instructions .a run time signature is generated and compared with the embedded signatures.

To facilitate error recovery, we split a program in module and run each module as a separate task. A main module controls the all other modules executions. When one of the error detection techniques detect an error, the erroneous module is aborted and restarted without corrupting the context of the other modules.

EDDI and CFCSS are software techniques which used for detecting hardware errors. These techniques are not requiring any changes in hardware or any type of support from the operating system. We have applied the combination of EDDI and CFCSS for sorting two algorithms i.e. insertion and quick sort. After each execution we do assertion to check undetected error for every sort algorithm, we check that the data is sorted correctly. For the FFT algorithm, firstly we calculate a checksum of the results and then compare it against the expected checksum that is stored in the program should implementing EDDI plus CFCSS have detected a total of 116 errors, no undetected error should be found there, and more than 95% of the recoveries have been successful.

5 Conclusions

All the hardware fault tolerance technique used in this board for the case of undetected error is despite by the result from the hardware shows here. Even though if single points of failure are eliminated by better design and additional fault tolerance techniques, but in software it may still be required for high reliability.

The software implemented for error detection and recovery techniques that are used in ARGOS have been effective for the error rate observed in the CTOS board. Then also hardware EDEC would be preferable for main memory, software EDEC has provided acceptable reliability for our experiment. In this research paper, results show, that COTS with SHIFT are viable techniques for low radiation environment.

References

[1] Shirvani, P.P. and E.J. McCluskey, “Fault-Tolerant Systems in a Space Environment: The CRC ARGOS Project,” *CRCTR98-2*, Stanford University, Stanford, CA, Dec. 1998.

[2] Ziegler, J.F., et al., *IBM J. Res. Develop.*, Vol. 40, No. 1, (all articles), Jan. 1996.

[3] Wood, K.S., et al., “The USA Experiment on the ARGOS Satellite: A Low Cost Instrument for Timing X-Ray Binaries,” Published in *EUV, X-Ray, and Gamma-RayInstrumentation for Astronomy V*, ed. O.H. Siegmund& J.V. Vellerga, SPIE Proc., Vol. 2280, pp. 19-30, 1994.

[4] R. Reed, et al., “Heavy Ion and Proton-Induced Single Event Multiple Upset,” *IEEE Trans. Nucl. Sci.*, Vol. 44, No. 6, pp. 2224-9, July 1997.