Departmental Papers (ESE)   Department of Electrical & Systems Engineering

6-1-2011

High-Reliability Computing For The Smarter Planet

Heather Quinn
ISR3, Space Data Systems

Andrea Manuzzato
ISR3, Space Data Systems

Paul Graham
ISR3, Space Data Systems

André DeHon
University of Pennsylvania, andre@seas.upenn.edu

Nicholas Carter
Intel Corporation

Follow this and additional works at: http://repository.upenn.edu/ese_papers

![Part of the] Part of the Engineering Commons

High-Reliability Computing For The Smarter Planet

Abstract

As computer automation continues to increase in our society, the need for greater radiation reliability is necessary. Already critical infrastructure is failing too frequently. In this paper, we will introduce the Cross-Layer Reliability concept for designing more reliable computer systems.

High-Reliability Computing For The Smarter Planet

Heather Quinn[1], Andrea Manuzzato[1], Paul Graham[1], Andre' DeHon[2],
Nicholas Carter[3]

[1]ISR3, Space Data Systems, LANL, PO Box 1663, MSD440, Los Alamos NM 87545
[2]Electrical and Systems Engineering, University of Pennsylvania, 200 S. 33rd St, Philadelphia, PA 19104
[3]Intel Corporation, 2200 Mission College Blvd, RNB6-61, Santa Clara, CA 95054

Abstract. As computer automation continues to increase in our society, the need for greater radiation reliability is necessary. Already critical infrastructure is failing too frequently. In this paper, we will introduce the Cross-Layer Reliability concept for designing more reliable computer systems.

INTRODUCTION

The geometric rate of improvement of transistor size and integrated circuit performance, known as Moore's Law, has been an engine of growth for our economy that has enabled new products and services, created new value and wealth, increased safety, and removed menial tasks from our daily lives. Affordable, highly integrated components have enabled both life-saving technologies and rich entertainment applications. Anti-lock brakes, insulin monitors, and GPS-enabled emergency response systems save lives. Cell phones, internet appliances, video games, and mp3 players enrich our lives and connect us together. Over the past 40 years of silicon scaling, the increasing capabilities of inexpensive computation have transformed our society through automation and ubiquitous communications. In this paper, we will present the concept of the smarter planet, how reliability failures affect current systems, and methods that can be used to increase the reliable adoption of new automation in the future.

THE SMARTER PLANET

Recently IBM has been presenting the idea of a "smarter planet." In smarter planet documents IBM states, "With computational power now being put into things we wouldn't recognize as computers, any person, any object, any process or service and any organization — large or small — can become digitally aware, connected and smart. With so much technology and networking available at such low cost, what wouldn't you enhance?[1]" In these documents, IBM discusses increased computer automation of roadways, banking, healthcare, and infrastructure, as automation could create more efficient systems that could reduce traffic jams, increase transparency in the financial institutions, increase information connectivity between doctors and hospitals, and reduce infrastructure strain. Increasing computation, though, necessitates that we increase reliability, as unmitigated failures in automobiles, infrastructure, medical-implantable devices and avionic systems could be dangerous.

In particular, we have found that with extreme-scale systems or systems in extreme environments, the effect of radiation-induced failures can dominate other reliability problems. In extreme-scale systems, such as supercomputing or infrastructure computing, the vast number of components in the system contribute to increase the likelihood of radiation-induced failures. In extreme-environment systems, such as avionics, the heightened radiation environment increases the likelihood of radiation-induced upsets. To illustrate both the exteme-scale systems and the extreme environment systems, we will present examples from automotive and avionic systems.
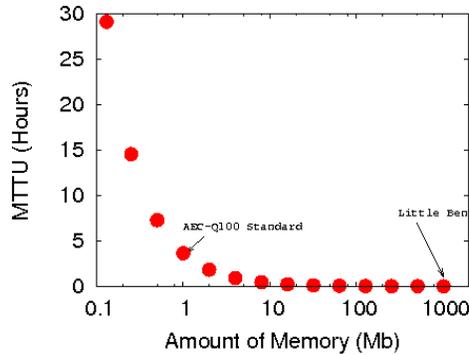
Figure 1: Worldwide mean-time-to-upset in hours for different car memory sizes.
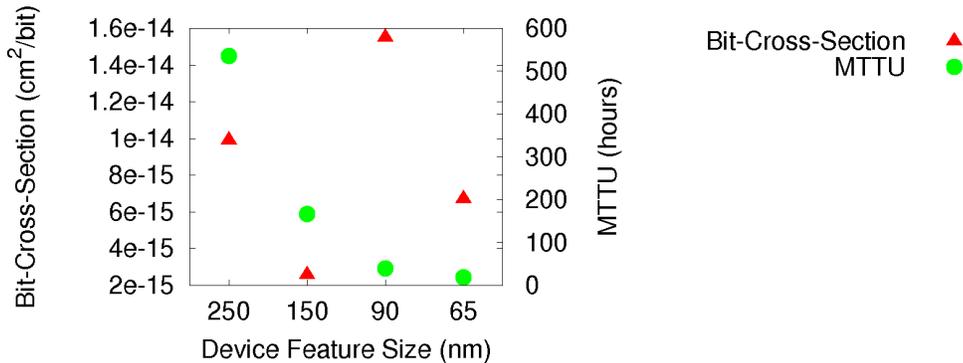

Figure 2: Mean-Time-To-Upset (MTTU) decreases as feature sizes shrink [shown at a flux of23,082 neutrons-cm2/hour corresponding to the flux seen by an airplane flying at 60,000 feet over the North Pole where the magnetic field lines converge]; sensitivity per bit (Bit-Cross-Section) does not show direct correlation with feature size.

AUTOMOTIVE SYSTEMS

One area of recent concern has been the reliability of cars. The most recent car reliability standard, "Failure Mechanism Based Stress Test Qualification for Integrated Circuits (AEC-Q100-Rev-G)," states that "Soft Error Rate (SER)] testing is needed for devices with large numbers of SRAM or DRAM cells ($\geq$ 1 Mbit). For example: Since the SER rates for a 130 nm technology are typically near 1000 FIT/Mbit, a device with only 1,000 SRAM cells will result in an SER contribution of 1 FIT." In Figure 1 the mean-time-to-upset (MTTU) in hours is shown for the worldwide population of cars as a function of the memory capacity in each car. These calculations take into account that approximately 250 million [2] cars are on the road every day with an average time on the road of three hours. Of the 250 million cars, we assume that the new cars are more likely to have more memory, so the MTTU is further rated to indicate that 60% of all cars on the road are manufactured in the last ten years. From this graph we see that if all of these cars had only 1Mb of memory, a single-event upset will occur approximately every 3.6 hours. For the computer-driven Grand and Urban Challenge [3] cars, such as Little Ben [4], which can have up to 128MB of memory, the MTTU is 12.6 seconds. These autonomous vehicles are indicative of the level of electronics we may see in cars over the next 10–20 years as automobile electronics continues to assist with more of the driving functions, such as automated parking and distance sensing cruise control.

While error-correcting codes will likely correct some of these errors in both of these scenarios, the increasingly likelihood of multiple-bit upsets, where a single ionizing particle creates multiple memory bit flips, can make correcting the errors expensive. Depending on the layout of the memory devices, multiple-bit upsets can be as high as 90% [5] of all events. Furthermore, single-event functional interrupts that destroy entire pages of memory are now as common as single-event upsets in dynamic random access memories (DRAMs). While there are many error-correcting codes that can correct multiple

faults simultaneously, such as Reed-Solomon, often times they require a particular memory usage pattern that will make random accesses to memory very expensive.

AVIONIC SYSTEMS

In avionic systems we have seen an increase in the use of Field-Programmable Gate Arrays (FPGAs). FPGAs are programmable devices where the user's circuit is implemented in programmable logic and routing. Due to the flexible architecture, these devices have always been popular for image processing. Recently, we have seen real-time image processing systems being incorporated in unmanned aerial vehicles (UAVs). These systems can collect real-time images from an aerial vantage point and can be helpful guiding wild fire firefighters to and from the hottest parts of the fire or supporting in-theater warfighters.

In the past we have used accelerated radiation experiments to measure the sensitivity of the device to radiation, called the cross-section. As these devices leverage memory cells, the sensitivity to radiation-induced failures is similar to traditional memory, such as static random access memory (SRAM). Figure 2 shows that while the per-bit cross-section has not changed much over several generations of FPGAs, the device's MTTU at airplane altitudes continues to decrease exponentially. As shown in Figure 3, this correlation is due to the increasing amount of memory used in each generation of FPGAs. Therefore, even if the sensitivity to radiation decreases on a per-bit basis, the increasing number of bits in the component negates these effects. Unlike the automotive example, the use of error-correcting codes is not useful for FPGAs. In these cases, triple-modular redundancy is used to mask the effect of radiation-induced failures until the system can be reset.
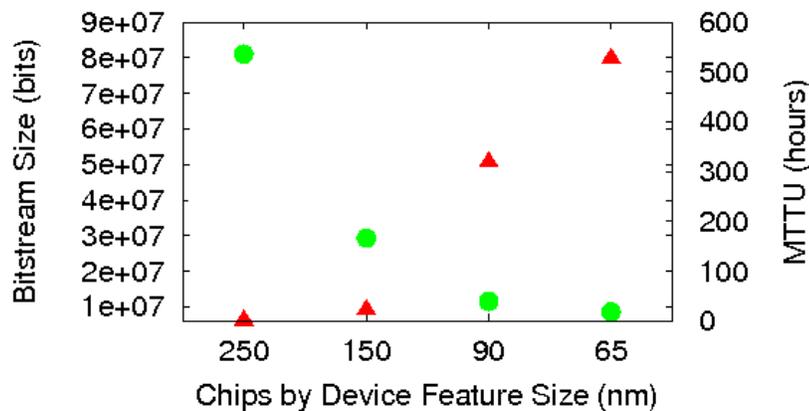

Figure 3: Chip capacity continues to increase as feature sizes shrink; capacity increase correlates well with decreasing MTTU.

CROSS-LAYER RELIABILITY

In both of these previous examples, it is necessary to mitigate radiation-induced failures in electronics to ensure reliable system execution. Unfortunately, radiation is only one source of reliability problem with modern electronics. In many current systems all of these reliability problems are mitigated at the hardware level, so that errors are not observable in the operating system or application. We are finding, though, that the need to mitigate all of these problems in the hardware is expensive in terms of area and power increases tohardware devices. To combat these problems NSF funded a Computing Community Consortium Visioning program in Cross-Layer Reliable Computing [6] to study new methods of reliability engineering. This program

brought reliability researchers from many different fields together to discuss current and future reliability problems. After discussing these problems with the areospace, consumer, infrastructure, large-scale, and life-critical industries we found several key points these researchers converged on:

- Reliability problems in electronic systems must be addressed but the cost of mitigating reliability problems is expensive,
- Information regarding the reliability of electronics is hard to determine and often comes too late in the design cycle to address it,
- No single layer of the computing stack has enough information to mitigate reliability problems efficiently, and
- The onus for system reliability should be shared across the computing stack.

Because of these problems, system reliability is often designed as shown in Figure 4. In this scenario, the hardware layers focus on preventing reliability failures and the software layers are blind to reliability failures. The solution to these problems is to build systems that are designed for repair, can filter errors, can detect and correct errors, can adapt to changing scenarios and environments, and degrade gracefully as the system comes to the end of its usable life.
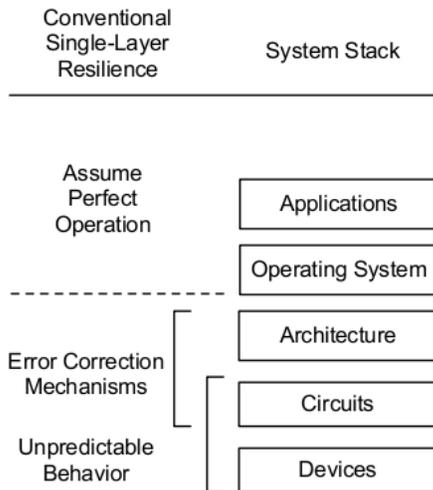

Figure 4: System Stack and Reliability

An illustration of cross-layer system design is shown in Figure 5. In this illustration the application layer determines a bound on the value x. Unlike conventional systems, the cross-layer version shares this information with lower layers. The architecture layer can use this bound to detect misbehaving hardware. However, the architecture layer does not need to capture and correct the problem. When it does notice misbehavior, it can signal the the middleware and OS layers to contain and correct the error before it becomes visible to the application. We are beginning to see scattered solutions with this flavor. Nonetheless, this approach demands a wholesale paradigm shift in the way we design and engineer computer systems. Multi-level solutions are not without their precedent in today's computer systems. They are regularly employed to protect storage and communications. While multi-level solutions have been useful in protecting bulk storage, such as DRAMs and persistent storage, similar solutions for computation currently do not exist. In part, the heterogeneous design of computing systems and their ability to transform data makes posing simple solutions that do not rely on brute-force replication difficult.
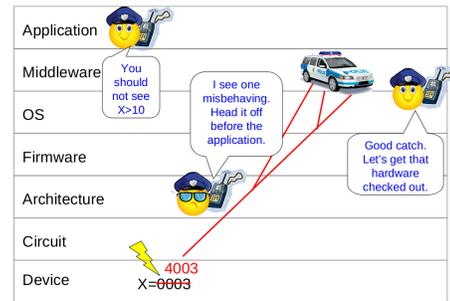

Figure 5: Cartoon Illustration of Cross-Layer Cooperation

Moore's Law feature-size scaling has been an economic and capability engine fueling wealth creation and quality-of-life improvements for the past 40 years. We are now moving into a qualitatively new regime where device energy will be the dominant limitation for the exploitation of additional capacity and where device are inherently unpredictable and unreliable. In this new regime our old solutions to reliability no longer make sense and will lead to an early end to the benefits of scaling. However, by exploiting information rather than energy to tolerate errors at higher levels in our system stack, we can productively exploit these smaller technologies to continue reducing energy while ensuring high system-level safety.

REFERENCES

1. "Conversations for a Smarter Planet," found on the web at http://www.ibm.com/smarterplanet, last accessed on 8/5/2010.
2. U.S. Bureau of Transporation Statistics. Number of u.s. aircraft, vehicles, vessels, and other conveyances. on web at http://www.bts.gov/publications/national_transportation_statistics/html/table_01_11.html, July 2010.
3. U.S. Defense Advanced Research Projects Agency. Darpa grand challenge and darpa urban challenge. on web at http://www.darpa.mil/grandchallenge/index.asp, 2007.
4. Jonathan Bohren, Tully Foote, Jim Keller, Alex Kushleyev, Daniel Lee, Alex Stewart, Paul Vernaza, Jason Derenick, John Spletzer, and Brian Satterfield. Little ben: The ben franklin racing team's entry in the 2007 darpa urban challenge. Journal of Field Robotics, 25(9):598–614, 2008.
5. Gasiot et al. Multiple cell upsets as the key contribution to the total SER of 65 nm CMOS SRAMs and its dependence on well engineering. IEEE Transactions on Nuclear Science, 54(6):2468–2473, Dec 2007.
6. "Cross-Layer Reliable Computing" found on the web at http://relxlayer.org, last accessed on 8/5/2010