TERPS: The Embedded Reliable Processing System

Hongxia Wang, Samuel Rodriguez, Cagdas Dirik, Amol Gole, Vincent Chan, and Bruce Jacob

Dept. of Electrical & Computer Engineering
University of Maryland, College Park
College Park, MD 20742
{hwang,samvr,cdirik,amol,vgchan,blj}@eng.umd.edu

http://www.ece.umd.edu/~blj/integrity/

Abstract — TERPS is a fault-tolerant computer design that significantly reduces the threat of electromagnetic interference (EMI), using hardware checkpoint/rollback-recovery. TERPS tolerates EMI by periodically checkpointing processor state into a special safe-storage device. The detection of EMI invokes rollback, which recovers processor state from a previously check-pointed state and resumes normal execution. Rollback results in loss of performance dictated by the EMI duration; TERPS ensures forward progress of the system provided EMI events are separated by some minimum time interval (e.g., at least 5.12µs for our prototype processor running at 100MHz). The performance overhead of our mechanism is reasonable: 5–6% overhead when checkpointing every 128 processor cycles.

I. Introduction

As feature sizes of integrated circuits decrease, the circuits' susceptibility to EMI increases. Therefore, integrated circuits will face potential failures from electromagnetic disturbances. A wide range of techniques can be used to protect the storage and transmission of data from EMI, but few techniques are available to protect the processor from virtually chip-wide failures. We propose a fault tolerant processor design called TERPS. We concede that with the assumption of availability of EMI detectors and a small subset of control and memory logics necessary to control checkpoint/rollback-recovery, the EMI caused malfunctioning of the processor could be avoided. TERPS works by backing up the processor state critical to the correct operation of the processor periodically. It checkpoints and stores the state into an external safe storage device that is more tolerant of EMI. Upon the detection of the EMI, the system automatically flushes the processor pipeline and reloads its state from a previous check-pointed precise state retrieved from the safe storage. This mechanism provides fault-tolerance even under the condition that EMI induces virtually unlimited number of faults within the processor core. As long as the check-pointed states are kept in the safe storage and they are precise, the processor can safely resume its execution from the reloaded check-pointed states as if the EMI did not occur at all. This saves the processor from losing a significant portion of its data compared to a shutdown or reboot.

II. Architecture

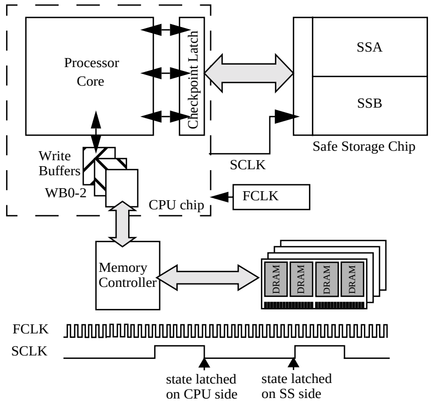
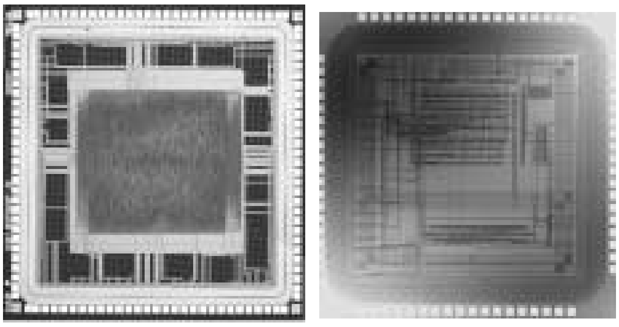


Figure 1: TERPS Architecture.

Fig. 1 shows the TERPS block diagram which includes the processor core, safe storage, and checkpoint latches. In Fig. 1, FCLK is the fast internal clock of the processor, and it is used to generate a slower clock, SCLK, which triggers the checkpoint and rollback events. SCLK is 128 times slower than FCLK. A snapshot of the processor state is recorded at every checkpoint, which is designed to occur on the falling edge of SCLK. An EMI check is performed on the rising edge of the SCLK to determine if it is necessary to initiate a rollback. If EMI is detected, a rollback is initiated wherein the processor pipeline is flushed and its state is reloaded from the safe storage. After rollback, the processor resumes operation starting from the reloaded state. By executing a checkpoint/rollback, the disruption caused by the occurrence of EMI is minimized and only a handful of recent results are discarded by the processor. The discarded instructions will be re-executed, and there will be data loss in this case. This mechanism ensures forward progress of the processor as long as the EMI disturbance is not continuous (i.e. for our prototype processor, forward progress is assured if EMI events are separated by at least 5.12µs). However, more realistic mechanisms need to be considered. These mechanisms should ensure the data accuracy of the safe storage, account for the speed discrepancy between the CPU and safe storage, guarantee the valid reloaded state, and handle precise interrupts. The designs of the multi-level safe-storage, the check-pointed latch, and the write buffers offer possible implementations of these mechanisms.

The safe storage is designed using various techniques that tradeoff speed for EMI tolerance. We solve the tighter timing requirement of the safe storage by inserting a CPU checkpoint latch between the processor and the safe storage to ensure that the data is valid for a significant fraction of an SCLK cycle. Two banks of safe storage that save the two most recent checkpointed states ensure that at least one valid state always exists. With the more realistic assumption that the EMI detection delay is at most one SCLK cycle, a corrupt write caused by EMI overwriting one bank of the safe storage is not disastrous because rollback will always recover the older state. An older checkpointed state will always be valid and rollback can be performed correctly. A multiphase commit protocol that takes precise interrupts into account is implemented by maintaining a window of memory store instructions in a series of memory



(a) Processor Core

(b) Safe Storage

Figure 2: Chip die photos fabricated through MOSIS.

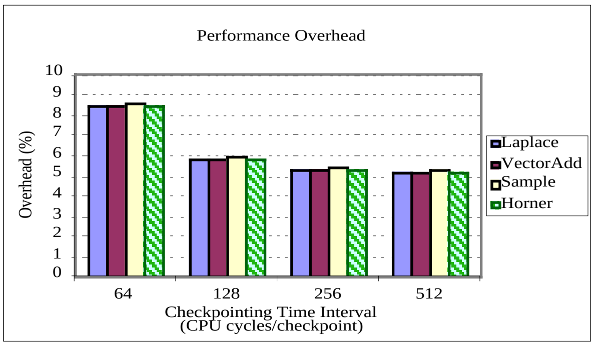


Figure 3: Performance overhead of checkpointing mechanisms.

write buffers to delay the store data before the memory is permanently modified.

III. Physical Implementation

Our initial prototype includes a simple 16-bit RISC designed in house with a 5-stage in-order pipeline that is similar to the DLX/MIPS processor [1], and safe storage memory units. We fabricated the CPU chip using cell-based design and the safe storage chip using full-custom design through MOSIS with the TSMC 0.25µm and 0.5µm process technology respectively. Fig. 2 shows photomicrographs of the chips. The prototype CPU runs at 100 MHz. On the same die as the CPU, we implemented twelve entries for each of the three write buffers to produce a better trade-off between storage requirements and performance overhead. The rationale for this implementation is that less write buffer entries are more likely to be filled thus stalling the processor until the next checkpoint is taken, while more write buffer entries require a larger safe storage space. Our implementation avoids both of these problems.

The checkpoint/rollback mechanism works under the assumption that processor-wide EMI-induced faults can be solved by reloading the processor states from the safe storage. This method migrates most of the responsibility of tolerating EMI to the safe storage. It is advantageous because of: 1) Offloading EMI tolerance requirements onto the safe storage allows us to focus EMI tolerance techniques on a much smaller subset of the design instead of an entire system, and 2) Less design restrictions are placed on the processor itself. These reasons make it possible to design a fast, high-performance processor regardless of its fault tolerance mechanisms. The safe storage is a memory that is specially designed to have significantly better EMI tolerance than the processor by using a variety of architecture, circuit, device, and process-level techniques. Most of these are orthogonal to each other and may be used or left out depending on the level of tolerance required by the system. Our design techniques compromise speed and die area in favor of better EMI tolerance. This is possible because of the very small size of the circuitry involved in checkpoint/rollback. We implement our safe storage cells using a six-transistor cell to maximize the static noise margin (SNM) since SNM is a good measure of the amount of spurious signal needed at the memory cell inputs to corrupt its state. The SNM of different memory cell configurations has been studied [2], showing that the 6T configuration is the best choice to maximize SNM if higher EMI tolerance system is needed [3, 4].

IV. Performance

The configuration of our design results in an average of 5–6% overhead. The overhead takes into account the overhead during normal operation when no EMI is detected in addition to the overhead caused by the stalls due to waiting for write buffer entries when transferring data atomically to the memory controller. The occurrence of EMI results in the loss of four SCLK cycles worth of operation, or 512 processor cycles. In the 512 cycles, 356 cycles are discarded instructions in one rollback window, while 156 cycles are for the actual rollback. For a 100MHz processor, this is equivalent to 5.12µs. Fig. 3 shows the performance overhead for our processor for different checkpoint intervals, different write buffer sizes and several different middle-sized benchmarks. For a checkpoint interval of 128 cycles and a write buffer size of 12 entries, our overhead is only 5–6%. The PCB has been designed and is in test.

V. Conclusions

In this paper, we described a fault-tolerant processor design that significantly reduces the EMI susceptibility of a system. We employed checkpoint and rollback recovery, including practical mechanisms to solve the problems caused by non-ideality of the system. The performance overhead of our mechanism is reasonable: 5–6% when check-pointing every 128 processor cycles. We also showed that our prototype processor assures forward progress even in the presence of EMI, as long as the EMI events are separated by at least 5.12 µs.

Acknowledgments

The work of Hongxia Wang and Samuel Rodriguez was supported by the Department of Defense and Lab for Physical Sciences in College Park MD, Contract MDA90499C2506. The work of Vincent Chan, Cagdas Dirik, and Amol Gole was supported by DoD MURI award AFOSR-F496200110374. Bruce Jacob is supported in part by NSF CAREER Award CCR-9983618, NSF grant EIA-9806645, NSF grant EIA-0000439, DoD MURI award AFOSR-F496200110374, the Laboratory for Physical Sciences in College Park MD, and by Cray.

References

[1] Hennessy, J. and Patterson, D., Computer Architecture A Quantitative Approach, 2nd Ed., Morgan Kaufmann Publishers, 1996.

[2] Seevinck, E. et al, “Static-noise margin analysis of MOS SRAM cells,” *IEEE J.Solid-State Circuits*, vol. SC-22, no.5, pp.748–754, Oct. 1987.

[3] Ishibashi, K. et al, “An alpha-immune 2-V supply voltage SRAM using a polysilicon PMOS load cell,” *IEEE J. Solid-State Circuits* vol SC-25, no.1, pp.55–60, Feb.1990.

[4] Sato, H. et al, “A 500-MHz pipeline burst SRAM with improved SER immunity,” *IEEE J.Solid-State Circuits* vol.SC-34, no.11, pp.1571-1579, Nov.1999.