Fault Tolerant Memory In Processor - SuperComputer On a Chip

Niranjan Soundararajan
Computer Science Department
Duke University
Durham, NC 27708-0129
niranjan83@rediffmail.com

Arrvindh Shriraman
Computer Science Department
University of Rochester
Rochester,NY-14623
ashriram@cs.rochester.edu

April 25, 2005

1 Abstract

Soft errors are adding another dimension to the present day architecture design space. Different techniques like redundant multithreading are evolved for handling them. The Memory In Processor (MIP) architecture provides fine grain processor memory integration. This integration provides efficient support for redundant multithreading within a functional unit. Detecting errors in intermediate stage of functional units reduces flushing of incorrect results and provides competent support for error recovery. As interconnect dominance increases, parasitics-induced phenomenons (i.e crosstalk) cause increased errors in bus lines. We integrate the Error Correction Codes (ECC) in the execution pipeline to protect results from faults occurring during transmission.

2 Introduction

Transient faults due to neutron and alpha particle strikes pose a significant challenge to processor design in the billion transistor era [1] [2]. With the advent of gigascale level transistor integration, soft errors continue to rise exponentially in combinatorial circuits [3]. Most of current designs [4] [5] [6] have been trying to exploit increased on-chip real estate and simultaneous multithreading techniques to detect and recover from soft errors. These designs are efficient because they do not change the micro-architecture apart from the support for buffering of final results from original thread.

We have developed the Memory In Processor (MIP) Supercomputer On a Chip (S.C.O.C) design to overcome the Von-Neumann bottleneck. The fine-grain memory-logic integration, shown in Figure 1a, gives rise to a low latency high bandwidth interface and enables us to promote functional unit density to SRAM integration levels [7][8]. In [9] we specified the technological requirements and the execution model for MIP S.C.O.C .

In this paper we study support for fault tolerance in the MIP S.C.O.C. The processor-memory integration allows MIP to buffer intermediate states and identify faults at the earliest. This reduces the amount of data flushing to a minimum. The result obtained from a functional unit is guaranteed to be correct due to error recovery scheme provided. This tremendously boosts performance compared to conventional techniques [6]. Interconnect dominance in the gigascale era leads to data corruption during transmission. ECC is provided for data stored in memory in conventional architectures. The integration of processor and memory enables MIP to efficiently provide ECC as part of the pipeline within each functional unit.

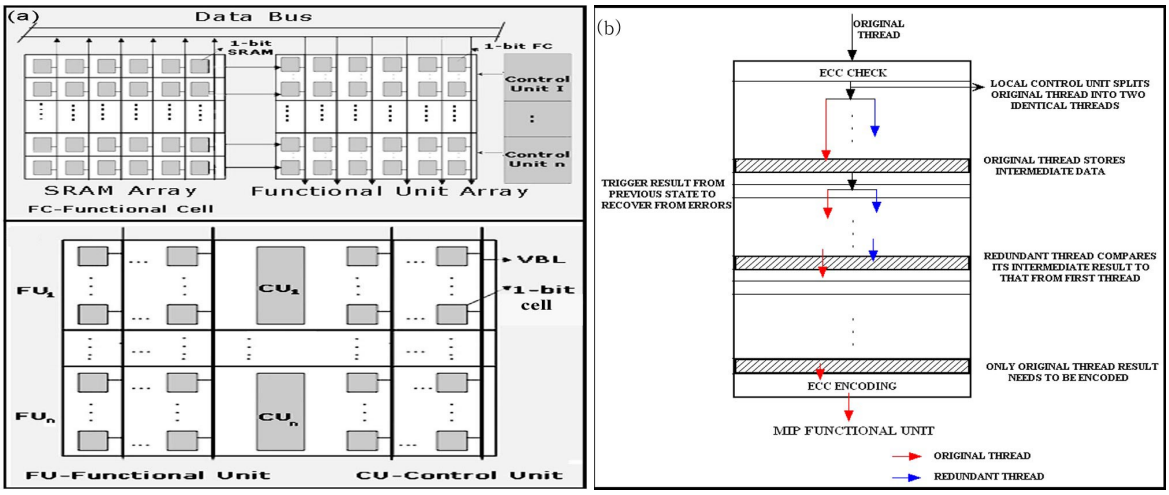

Figure 1: a) Fine-grain merging of SRAM and computing logic in MIP S.C.O.C. b)Overview of the fault detection and recovery scheme in MIP S.C.O.C.

3 Soft Error Detection in MIP S.C.O.C

In the MIP S.C.O.C, the control unit is decentralized and handles control signals of each functional unit independently. The control unit replicates the original incoming thread. The redundant thread follows the original thread in the pipeline. Since a row in MIP functional unit can also store data, specific rows buffer intermediate data from the original thread. The redundant thread compares the buffered intermediate data from original thread to its corresponding intermediate output. If they are identical, the execution proceeds, otherwise error is detected and recovery triggered.

The advantages of this approach is that result is buffered at location of computation enabling error detection at an much earlier stage compared to conventional techniques. This reduces the overall traffic in the system. Also, no faulty data proceeds out of the functional unit. This guarantees that no flushing of computed data is required which would otherwise decrease performance of the system. Flushing of incorrect results, at a later stage, would have negative impact on performance of data intensive operations like vector, matrix operations. The buffering scheme in MIP architecture enables us to detect and recover from faults at the intermediate stage itself and reduces the impact on performance.

4 ECC Protected Execution Pipeline

ECC provides efficient protection for data in memory. Interconnect parasitics in gigascale architectures will lead to phenomenons (i.e crosstalk) which cause soft errors in bus lines. It is difficult to incorporate ECC protection within conventional execution pipeline due to the reliance of these architectures on high clock for performance.

ECC protection of results, during execution, is provided in MIP through two additional pipeline stages at the start and end of computation. At the start of a HLF unit, ECC check is provided to recover from any error in data transmitted. At the end of result computation, the data is encoded. This will prevent any chance of data corruption over the bus lines. MIP lends itself seamlessly to incorporate ECC to prevent errors in data transmission. Both the ECC check and encoding are provided as two additional pipeline stages in a functional unit. The control unit replicates the incoming thread only after the ECC check is done. Though adding these

two stages of ECC would increase the complexity of the system, it becomes a necessity for billion device device architectures with a large die area due to increased parasitic effects like crosstalk and soft errors.

Since the two threads follow each other within a few pipeline stages, the result from second thread will be available for fault checking before encoding the final result from original thread. If the comparison indicates that computation is correct, the result from original thread is forwarded.

5 Soft Error Recovery in MIP S.C.O.C

The overview of the fault detection and recovery scheme is shown in Figure 1b. Recovery in MIP architecture can be efficiently incorporated in the fault detection scheme. This is due to the buffering of intermediate results. When the two threads differ in their results at any intermediate stage, the results buffered at previous stage is reused. The previous stage result is split into two threads and execution proceeds from this stage. This step guarantees minimum flushing of computed data and the final result obtained is always correct. This recovery scheme is not very complex, thereby it is efficiently incorporated into the MIP architecture.

6 Conclusion

This work presents an efficient fault detection and recovery mechanism for the MIP architecture. The decentralized control unit localizes the support for redundant threading within the functional unit thereby reducing overhead on the instruction scheduling mechanism. Our design principle of massive parallelism for low clock enables us to incorporate ECC in the execution pipeline with minimal impact on overall performance.

References

[1] J. Zeigler. "Terrestial Cosmic Rays". IBM Journal Of Research and Development, 40(1):pages 19-39, January 1996.

[2] E. Peterson, P.Shapiro, J. Adams, E. Burke. "Problems and Projections in CS for the Next 49 Years". Journal Of ACM , Jan 2003.

[3] P. Sivakumar, M. Kistler, S.W. Keckler, D.C. Burger, and L.Alvisi. "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic". International Conference on Dependable Systems and Networks (DSN) , June 2002.

[4] Shubhendu S. Mukherjee, Michael Kontz, Steven K.Reinhardt. "Detailed Design and Evaluation of Redundant Multithreading Alternatives". Proceedings of 29^{th} International Symposium on Computer Architecture, 2002.

[5] Steven K. Reinhardt and Shubhendu S. Mukherjee. "Transient Fault detection via Simultaneous Multithreading". Proceedings of 27^{th} International Symposium on Computer Architecture, June 2000.

[6] Eric Rotenberg. "AR-SMT: A Microarchitectural approach to fault tolerance in Microprocessor". Proceedings of Fault-tolerant Computing systems, pages 84-91 , June 1999.

[7] N Venkateswaran , Arrvindh Shriraman, Aditya Krishnan, Niranjan Soundararajan, Srinivas Sridharan. "Memory In Processor:Evolution of a novel Supercomputer Architecture". MEDEA Workshop (at PACT Conference), September 2003.

[8] N Venkateswaran, Aditya Krishnan, Niranjan Soundararajan, Arrvindh Shriraman,Srinivas Sridharan. "Memory In Processor:A novel design paradigm for Supercomputers". ACM SIGARCH Computer Architecture News , June 2004.

[9] N Venkateswaran , Arrvindh Shriraman, Niranjan Soundararajan. "Memory In Processor: Processor Design and Execution semantics for massive single-chip performance". WMPP (at IPDPS) , April 2005.