FAULT TOLERANCE FOR MULTI-CORE AND MANY-CORE PROCESSORS

Vanessa VARGAS

PhD candidate in Nano Electronics and Nano Technologies
Université de Grenoble Alpes - France
Professor at Universidad de las Fuerzas Armadas ESPE
Department of Electrical and Electronics- Ecuador

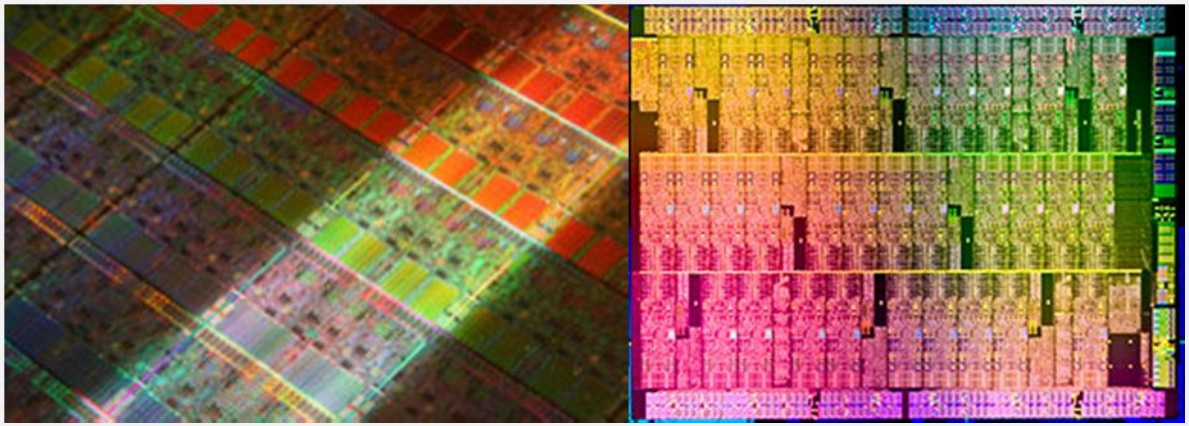
OUTLINE

- Introduction
- Motivation
- Background
- Work Done
- Conclusions

INTRODUCTION

The use of multi-core and many-core platforms is becoming more frequent in computing systems to improve the overall performance.

- *Speed:* Performing parallel processing.
- *Efficiency*: Choosing the appropriate form of multiprocessing to achieve maximum concurrency.
- *Reliability* :Implementing fault tolerant applications by **redundant execution**.



INTRODUCTION

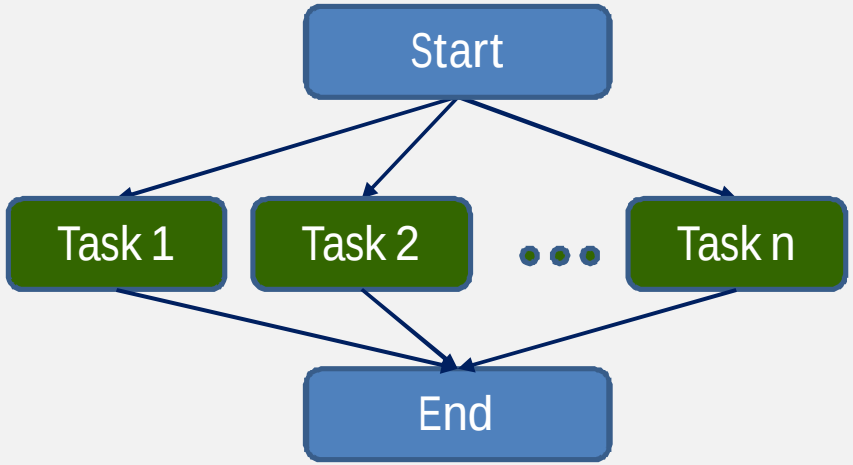
When dealing with multi-core and many-core processors. It is important to consider the following characteristics:

- Chip complexity: Potential source of errors.
- Multi-threading: Allows redundant techniques for transient fault detection and recovery.
- Error reporting architecture: Manufacturers have introduced the machine check error registers.
- Multiprocessing Mode
  - SMP (Symmetric Multi-Processing).
  - AMP (Asymmetric Multi-Processing).

INTRODUCTION

The use of multi-core and many-core processors is a current solution to achieve parallelism.

- Multicore and many-core are very sensitive to SEUs due to the higher degree of miniaturization and the huge number of memory cells.
- However, multiplicity of cores gives the opportunity to run in parallel several copies of the same application to achieve fault tolerance.



OUTLINE

- Introduction
- **MOTIVATION**
- Background
- Work Done
- Conclusions

MOTIVATION

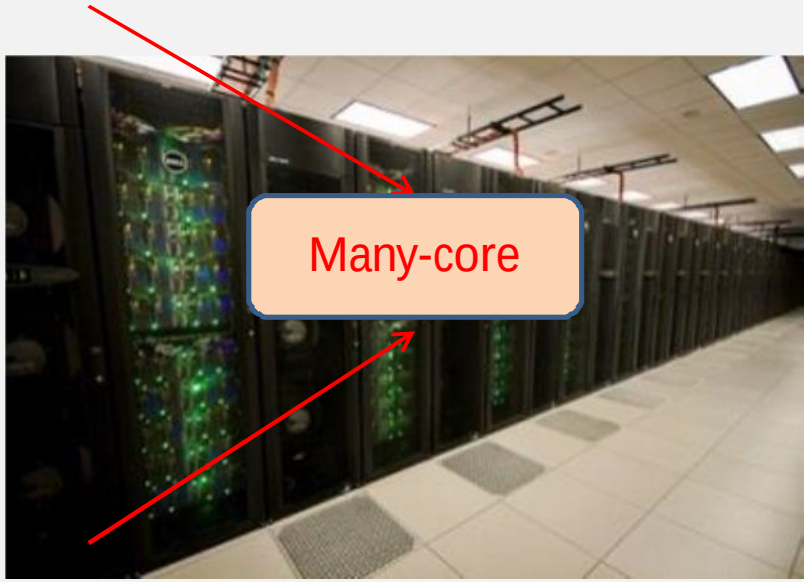
SUPERCOMPUTERS Top500 (June 2016)

1er de Top500 : Sunway TaihuLight - Sunway MPP, NRCPC, 93.01 Petaflops

Sunway SW26010 260C 1.45GHz, Sunway NRCPC
10,649,600 cores
15.31 MW
National Supercomputing Center in Wuxi China

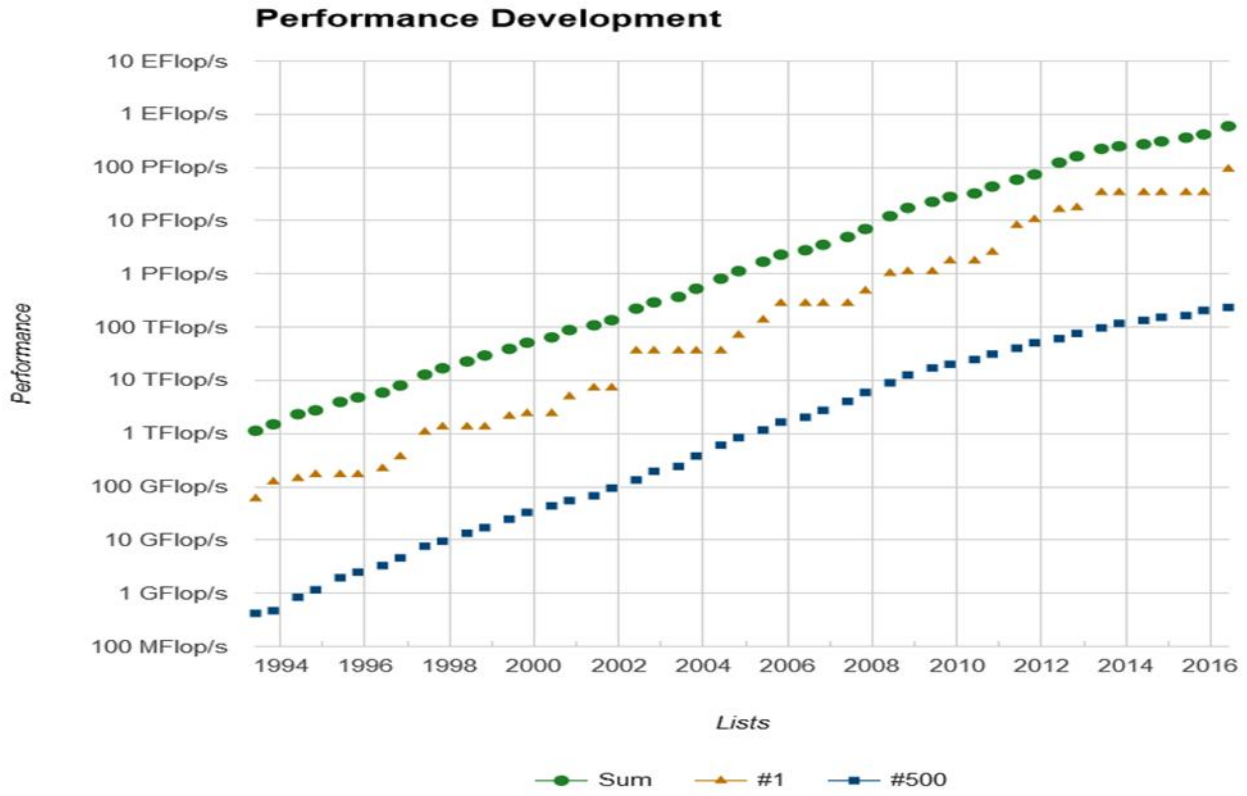
2nd de Top500 : Thiane-2, NUDT, 33.86 Petaflops

ivybridge 12c/proc, 2.2GHz + Intel XeonPhi,
3 120 000 cores
17.81 MW
TH Express-2,
National University of defense technology, China



MOTIVATION

In HPC systems, the use of many-core processors is crucial to satisfy the growing demand of performance and reliability without a critical increase of power consumption.



MOTIVATION

This exponential growth face many challenges:

Power	• Limited power budget
Space	• Fit in available floor space
Cost	• Fixed financial budget
Memory technology	• Feed compute power & cost efficiently
Network technology	• Connect nodes power & cost efficiently
Software	• Scale to utilize the growing compute capacity
RELIABILITY	• Failure rates should not grow with machine size
And others …	

MOTIVATION

CONCERNING THE RELIABILITY

Evaluate fault tolerance technique under radiation and fault injection campaigns.

Evaluate the impact of the use of fault tolerance techniques on performance and energy consumption.

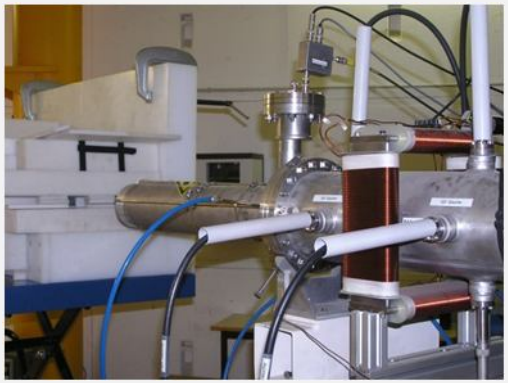


FIGURE 1. RADIATION EXPERIMENT

OUTLINE

Introduction

Motivation

BACKGROUND

- Multiprocessing modes
- Fault Tolerance

Work Done

Conclusions

MULTI-PROCESSING MODES

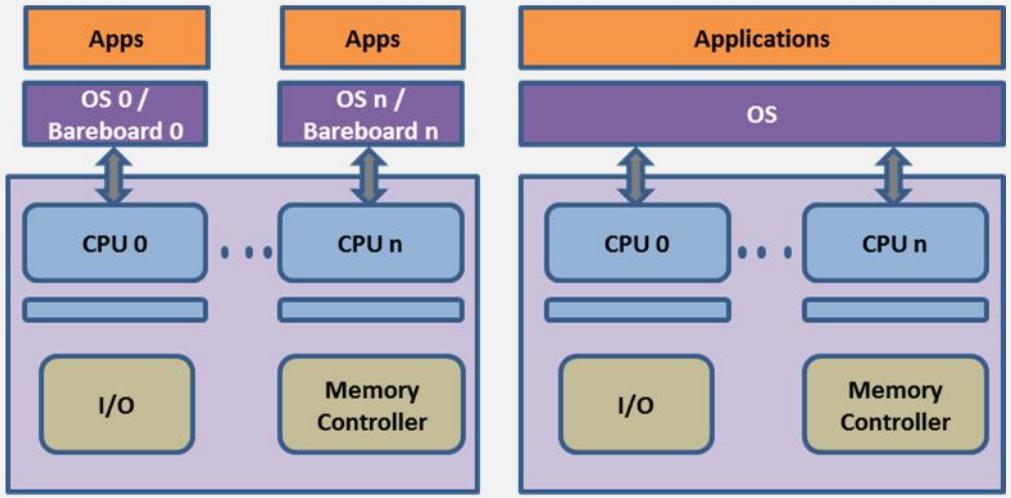


FIGURE 2. SCHEMES OF AMP AND SMP PROCESSING MODES

SMP

- Single OS is responsible for achieving parallelism in the application.
- It dynamically distributes the tasks among the cores, manages the organization of task completion, and controls the shared resources.

AMP

- The cores run independently of each other, with or without OS.
- They have their own private memory space, although there is a common infrastructure for inter-core communications.

FAULT TOLERANCE

A system is considered as fault tolerant when facing a fault, it continues working correctly.

Fault tolerance can be obtained by redundancy.

1	• Spatial Redundancy
2	• Temporal Redundany
3	• Both of them

Spatial vs temporal redundancy

SPATIAL

It uses different physical components

It can separate identical data signals in space

ADVANTAGE

- It lacks an inherent maximum operating frequency.

DISADVANTAGES

- It requires more area and components.
- Penalty in performance

TEMPORAL

It uses the same physical components

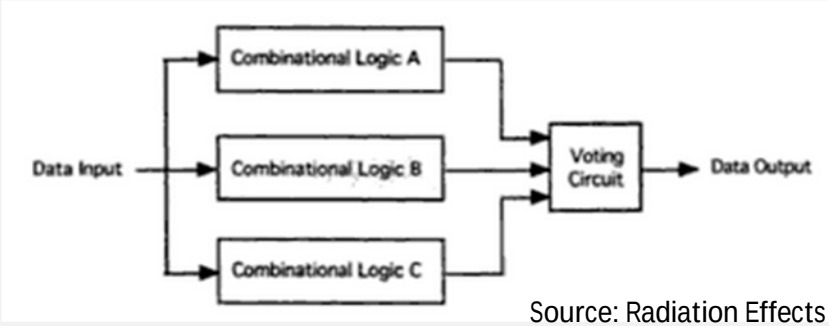
It can separate identical data signals in time

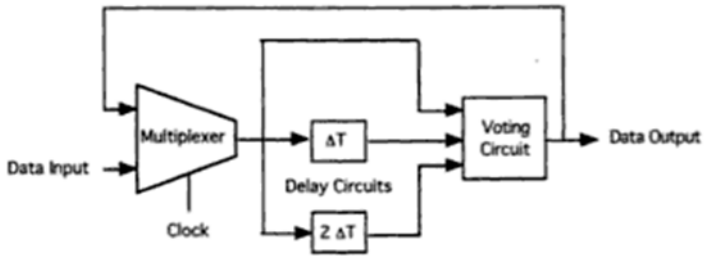
ADVANTAGE

- Fewer components.

DISADVANTAGES

- Latency penalty.
- It has a maximum operating frequency and therefore not used in commercial processes faster





Source: Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices

FAULT TOLERANCE IN MULTICORE

Taking advantage of the multiplicity of cores, various redundancy techniques can be considered.

1. Temporal redundancy
2. Data value redundancy
3. Information redundancy for error detection in multicore designs
4. Redundancy in execution

For evaluating any technique it is possible to do it by fault injection or by radiation test campaigns.

Redundancy in execution

The replication of state machine is used

Replication copies of a process is performed.

Copies follow the same sequence of execution and produce the same result if inputs are the same.

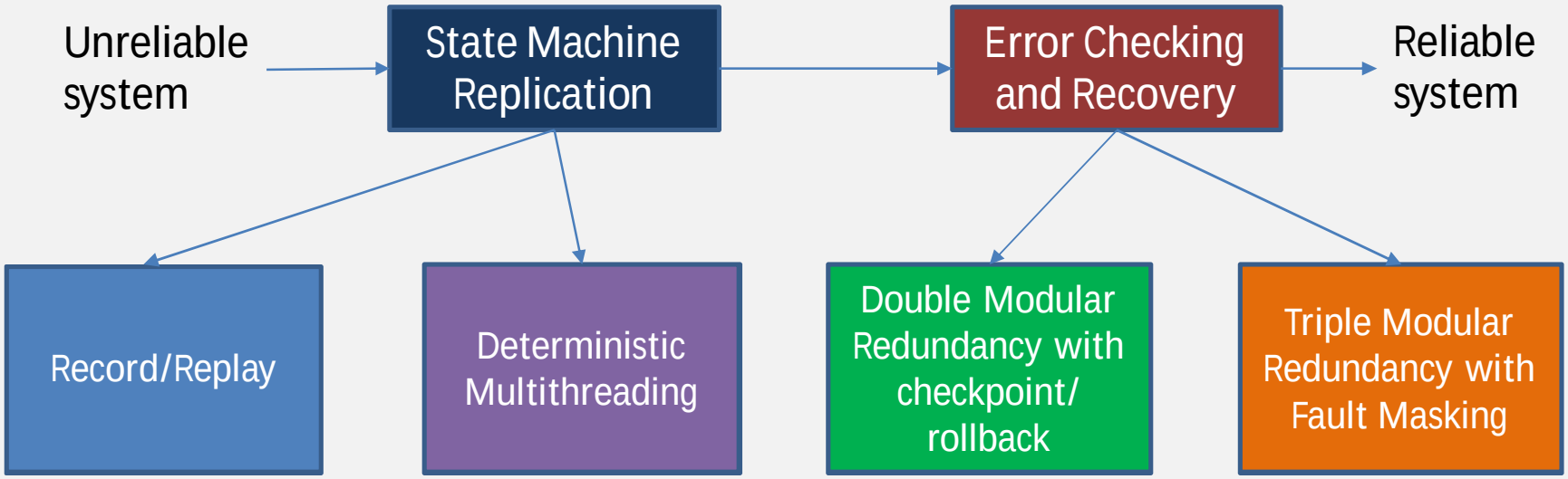
It should ensure that redundant processes not diverge in the absence of failures.

Divergent causes are:

- Asynchronous signals
- Nondeterministic functions (gettimeofday)
- In multi-core
  - Access to shared memory

The record / replay method ensures that access to shared memory is done in the same order.

Redundancy in execution



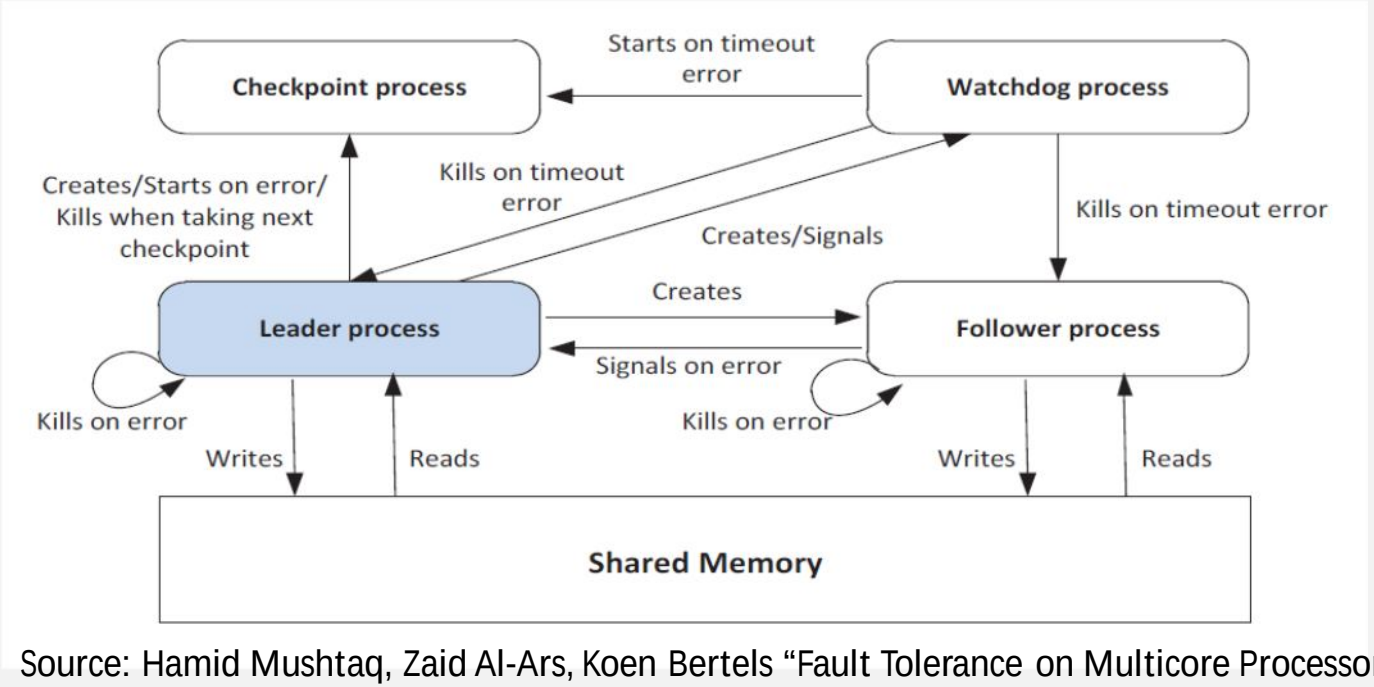
Redundancy in execution

Deterministic multithreading	• by using locks, barriers and creating threads. • Problem: Slow down application.
Double Modular Redundancy DMR	• It allows error detection.
Triple Modular Redundancy TMR	• It allows error detection and correction by a voter.

Redundancy in execution

Mixed Modelling

- Deteministic Multithreading
- DMR



Source: Hamid Mushtaq, Zaid Al-Ars, Koen Bertels "Fault Tolerance on Multicore Processors using Deterministic Multithreading"

FIGURE 3. EXAMPLE OF REDUNDANCY IN EXECUTION

OUTLINE

- Introduction
- Motivation
- Background
- **WORK DONE**
  - **Freescale P2041RDB**
    - TMR in AMP mode
    - Fault Injection in SMP
    - Radiation Tests in AMP y SMP mode
  - **KALRAY MPPA-256 (Multi Purpose Processing Array)**
    - Fault Injection in AMP mode
    - Radiation Tests in AMP mode
    - Fault Injection in mixed mode
    - Evaluating Fault Tolerance Technique
- Conclusions

FREESCALE P2041

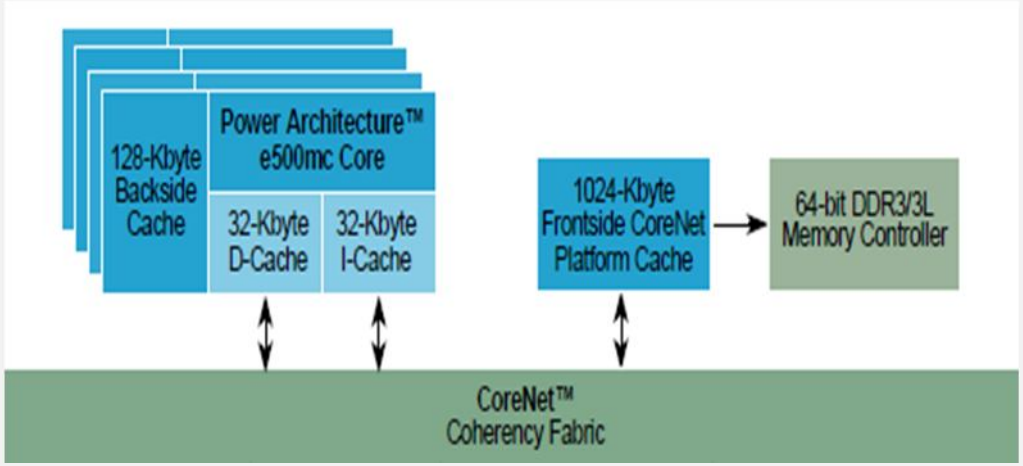


FIGURE 4. QORIQ P2041 MEMORY ARCHITECTURE

Built on	• Power Architectures technology
Manufactured	• 45nm SOI technology
Based on	• four e500mc cores(32-bit superscalar processor)
Operation Frequency	• up to 1.5 GHz

TMR in AMP mode

Environnement

This work studied a fault injection method to emulate the effects of SEUs in multicore processors. Test campaigns were performed on a TMR application to identify the potential weaknesses that may be a challenge for critical applications.

Multicore processor target:	Freescale P2041 Quad-core
Programming environment and compiler tool:	CodeWarrior Development Studio
Benchmark Application	40x40 Matrix Multiplication

TMR in AMP mode

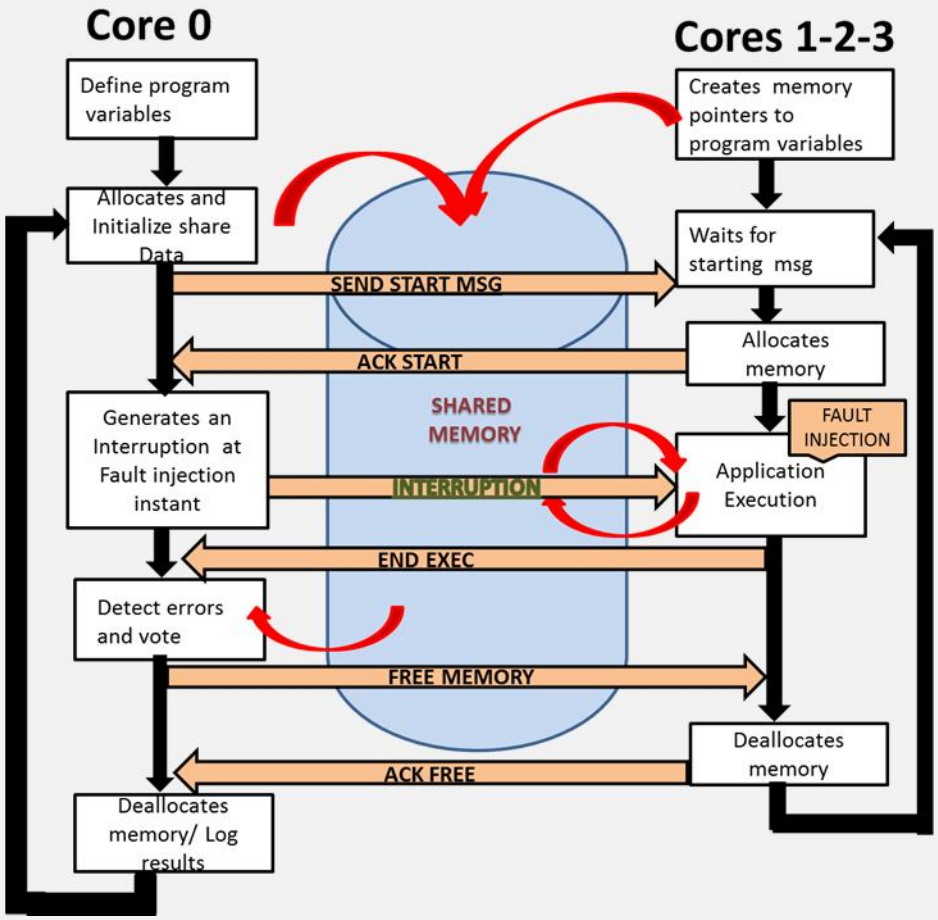


FIGURE 5. FAULT INJECTION STRATEGY IN PROCESSOR REGISTER

TMR in AMP mode

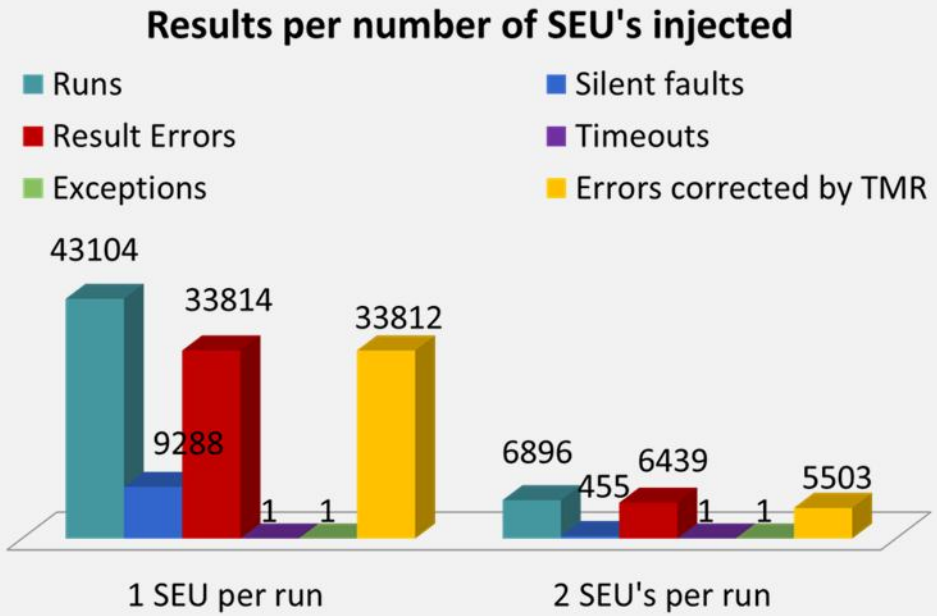


FIGURE 6. FAULT-INJECTION CONSEQUENCES

EXPERIMENT

- It was run 50000 times.
- Injection of one or two SEUs per execution.

RESULTS

- 20% of injected faults have no detectable consequences (silent faults).
- If one SEU is injected per execution, the error rate reaches 78% and the TMR corrects 99.99% of them.
- On the other hand, if two SEUs are injected, the error rate reaches 93% while the error correction factor decreases to 85%.

TMR in AMP mode

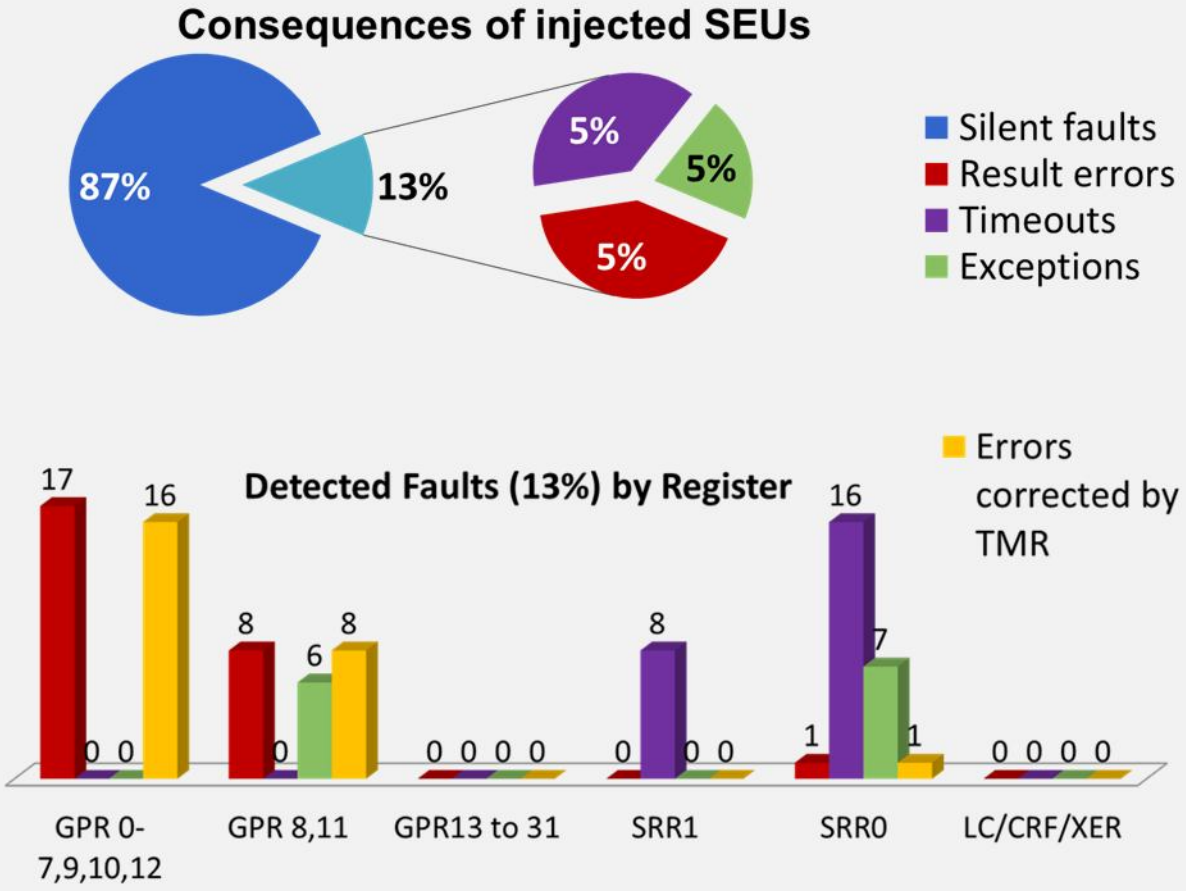


FIGURE 7. FAULT-INJECTION CONSEQUENCES IN PROCESSOR REGISTERS

FAULT INJECTION IN SMP

Environnement

Multicore processor target: Freescale P2041 Quad-core

Programming environment and compiler tool: CodeWarrior Development Studio

Benchmark Applications:
a) Travelling Salesman Problem *(TSP)*
b) 80x80 Matrix Multiplication *(MM)*

Table I summarizes both applications regarding the amount of data and code sections.

Target Application	DATA			CODE	
	Input Variables	Output Variables	Internal variables	Worker Threads	Tasks
TSP	260	17	88	4	~2200
MM	12800	6400	12	4	4

TABLE I. APPLICATIONS SUMMARY

FAULT INJECTION IN SMP

Two test campaigns were performed on each selected application:

a) Fault injection in processor registers.
b) Fault injection in memory region

Target Application	Standard execution time [ms]	Nominal execution time with fault-injection load		Runs per campaign
		In Registers [ms]	*In Memory [ms]*	
TSP	2977	3287	3472	7500
MM	21	23	25	50000

TABLE II. FAULT - INJECTION CAMPAIGNS

FAULT INJECTION IN SMP

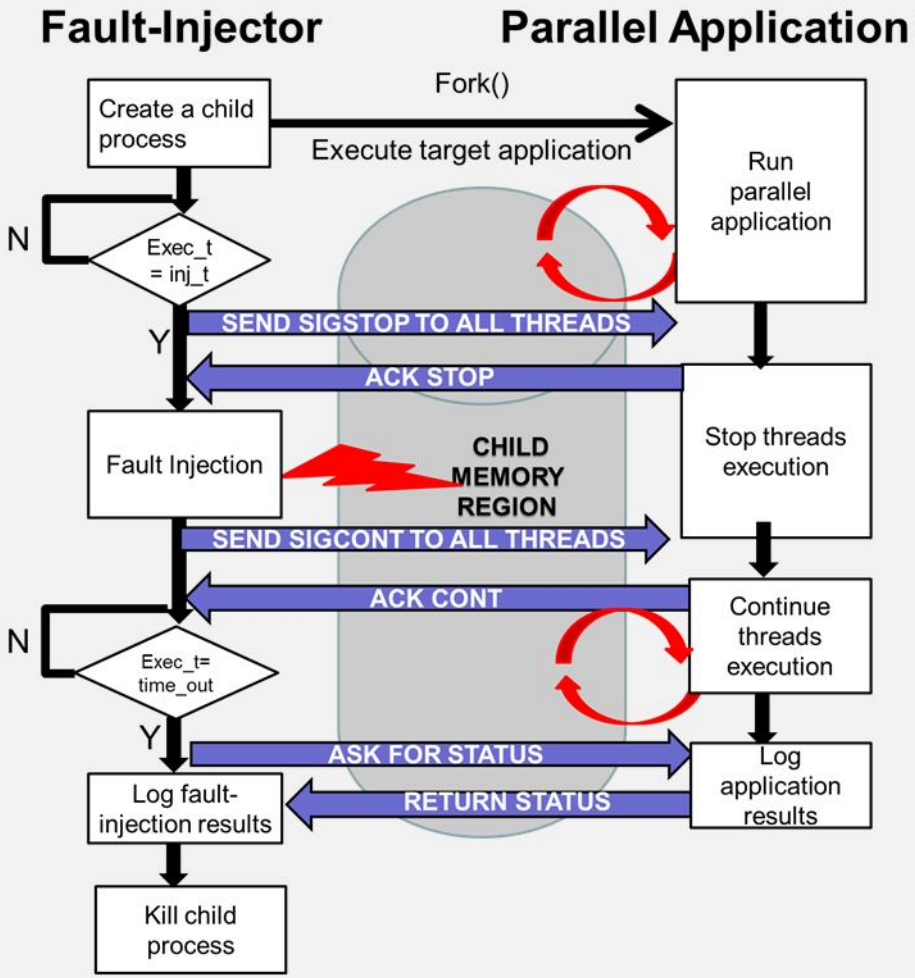


FIGURE 8. PROPOSED SOFTWARE FAULT-INJECTION IN MEMORY REGION

FAULT INJECTION IN APPLICATION RUNNING IN SMP

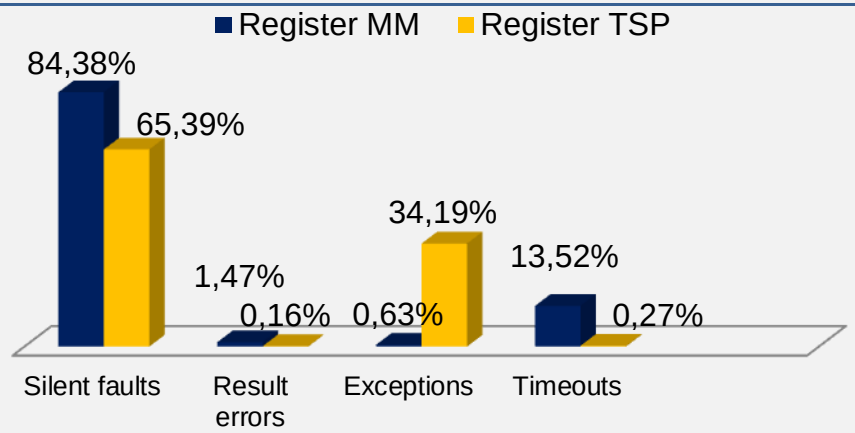
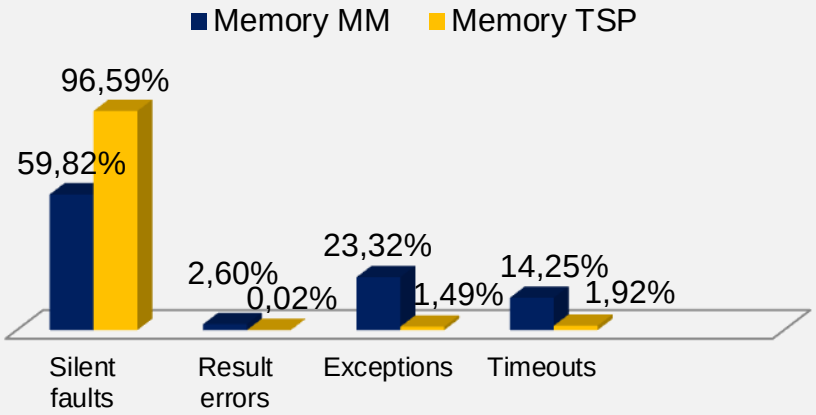


FIGURE 9. FAULT-INJECTION CONSEQUENCES IN PROCESSOR REGISTERS



These campaigns target only the private code memory:

- The initial process stack memory,
- The thread's stacks memory, and
- The process' heap memory.

FIGURE 10. FAULT-INJECTION CONSEQUENCES IN MEMORY REGION

RADIATION TESTS

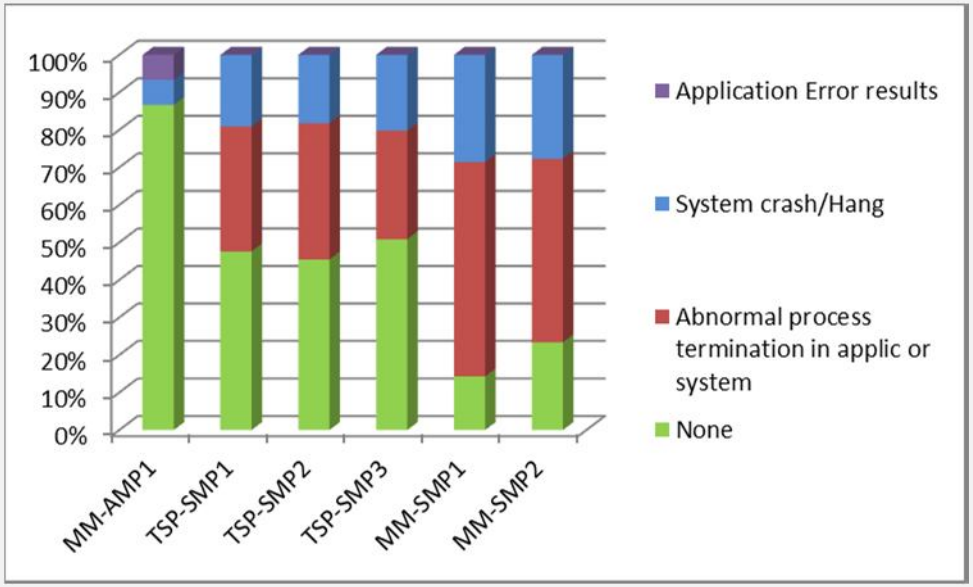


FIGURE 11. CONSEQUENCES OF RADIATION TEST CAMPAIGNS

- From the results, one can see that the reliability of an application depends on the software environment characteristics:
  - Operating system.
  - Multiprocessing mode used.
  - Characteristics of application.

RADIATION TESTS IN SMP MODE

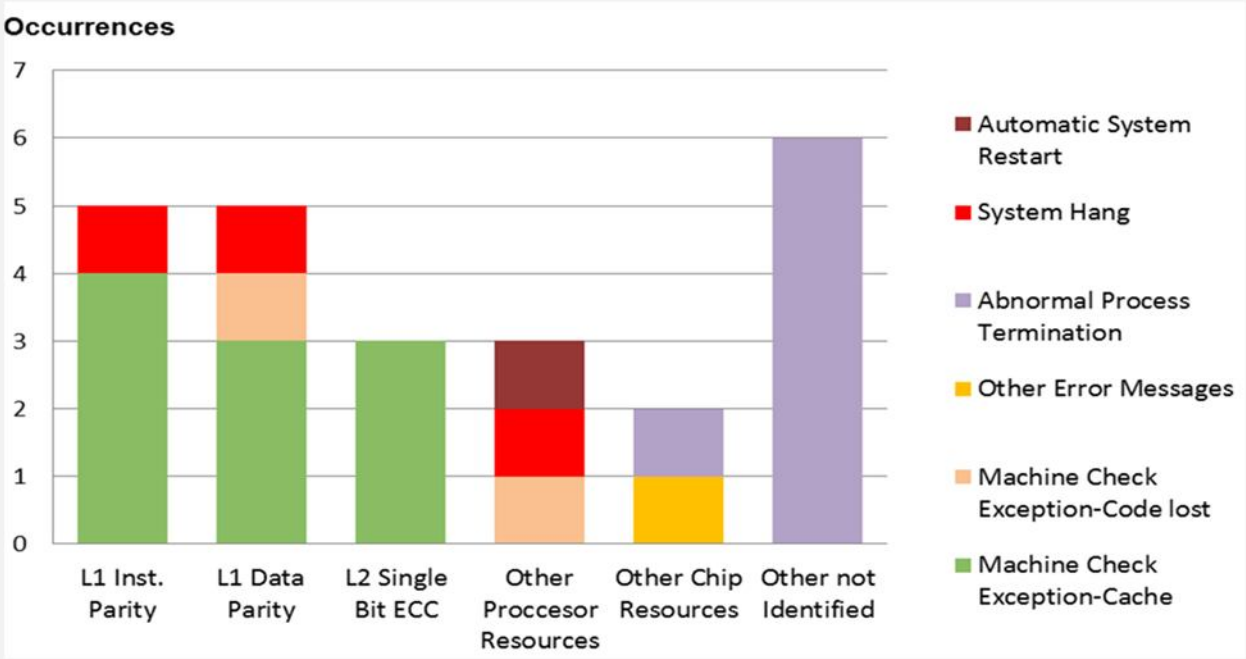


FIGURE 12. ERROR CLASSIFICATION ACCORDING TO OS FAULT

The obtained results revealed that errors may occur in SMP mode, even if the OS is in idle mode.

RADIATION TESTS

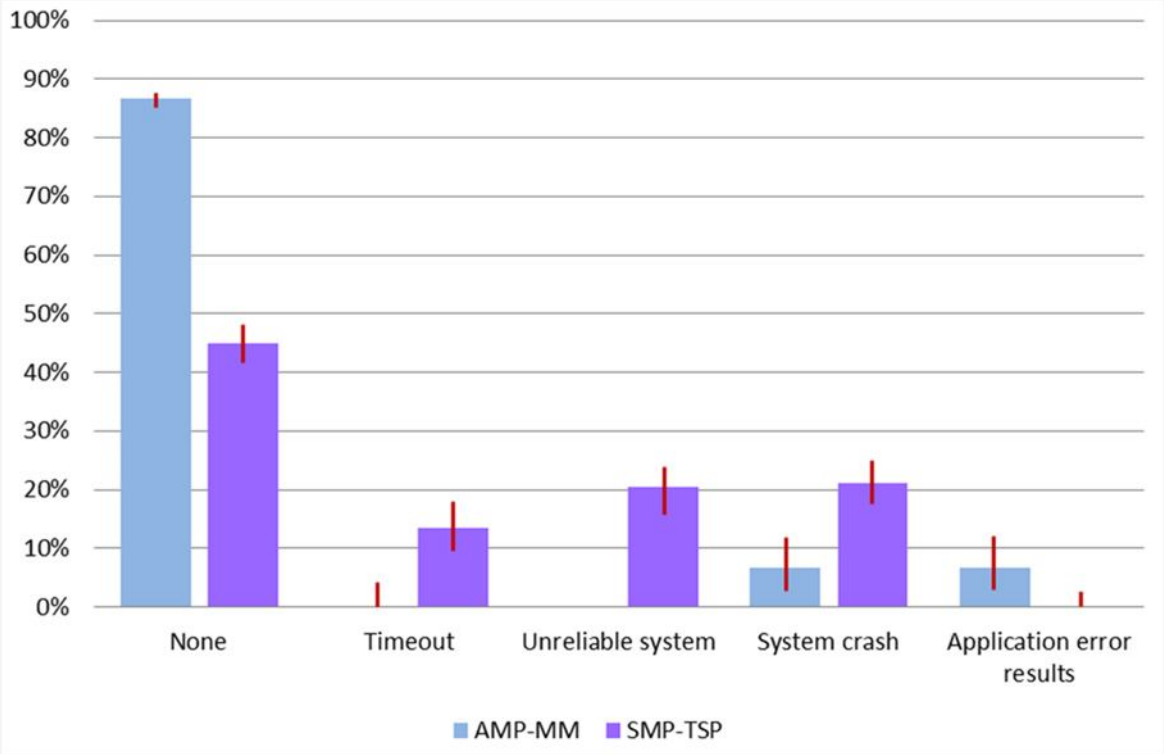


FIGURE 13. SEE CONSEQUENCES ACCORDING TO THE SCENARIO IMPLEMENTED. THE CONFIDENCE INTERVALS ARE SHOWN BY MEANS OF THE RED LINES.

KALRAY MPPA-256

Manufactured	• TSMC CMOS 28HP technology.
Integrates	• 256 Processing Engine (PE) and 32 Resource Management (RM) cores.
Based on	• Core VLIW 32-bit/64-bit architecture.
Operation frequency	• 100 MHz to 600 MHz.
Power Consumption	• 15 W to 25 W.
Peaks performance at 600 MHz	• 634 GFLOPS and 316 GFLOPS for single and double-precision respectively.
Clustered architecture	• 16 compute clusters (CCs) and 2 I/O clusters per device.

Compute Cluster	•multi-banked local static memory (SMEM) of 2MB shared by the 16(PE) + 1(RM).
I/O cluster	•2 groups of quadcore. Each 128 KB shared.

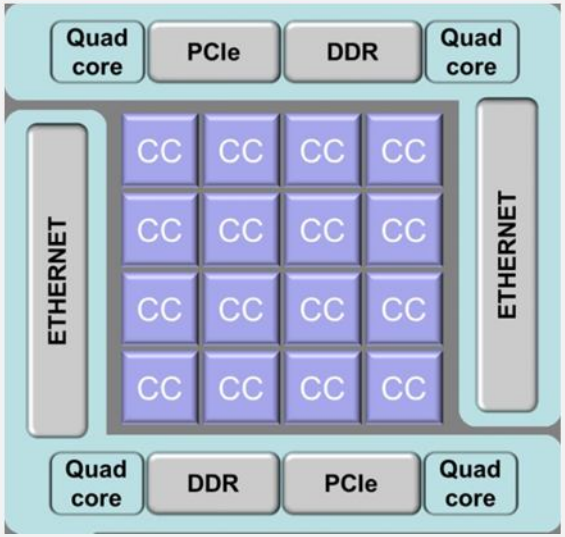


FIGURE 14: MPPA- 256 MEMORY ARCHITECTURE

Fault Tolerance Approach on MPPA

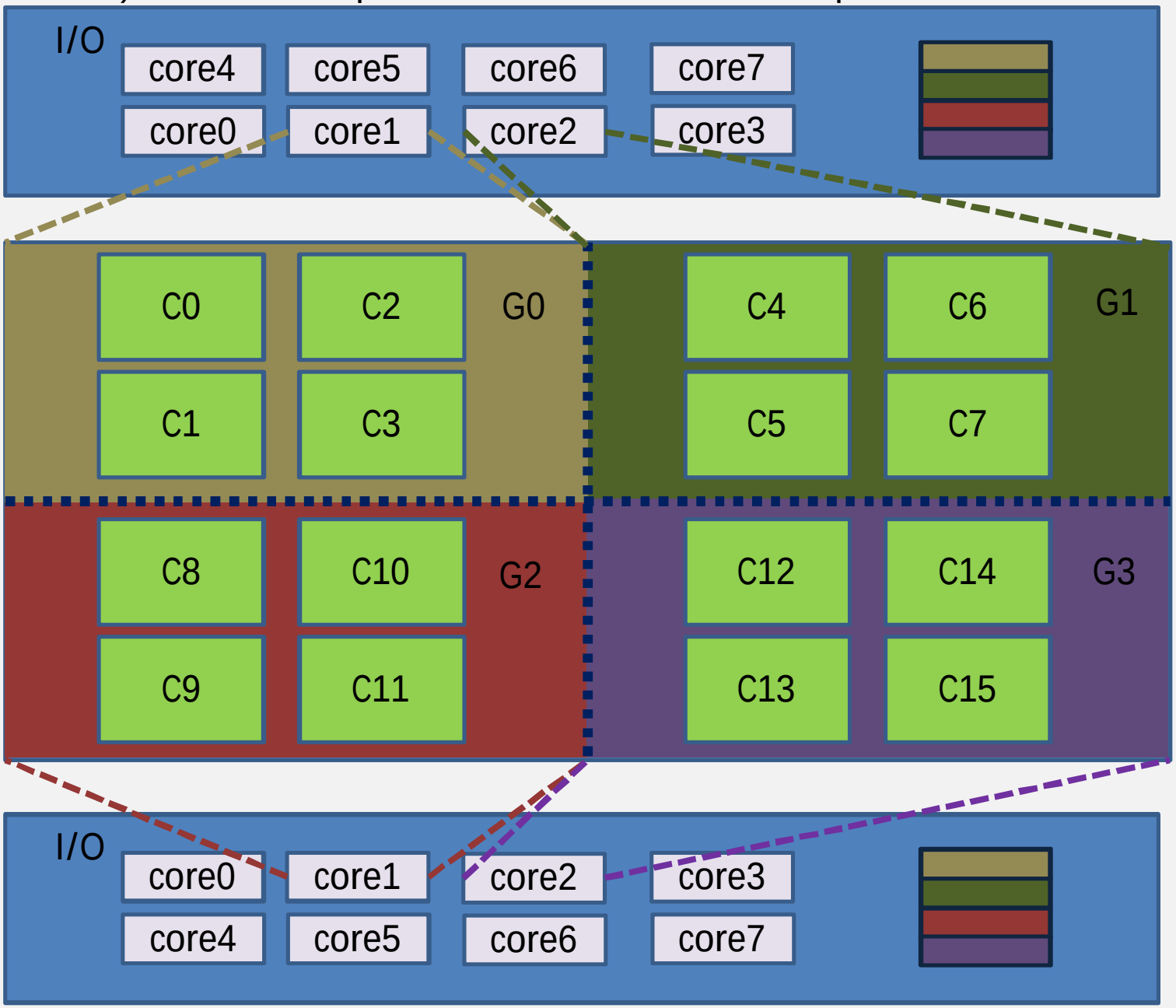
Implemented at application level, it uses the 2 I/O to improve the reliability of the application.

1) • Core 0 Initializes intercluster communications

2) • Core 0 generates a pthread per core:

Core 1, 2	• Master of group of computing cluster
Core 4,5,6	• Voters of the results (TMR –arbiter)
Core 3	• Arbiter of the final results. It logs the results
Core 7 (only of I/O 0)	• Fault Injector.

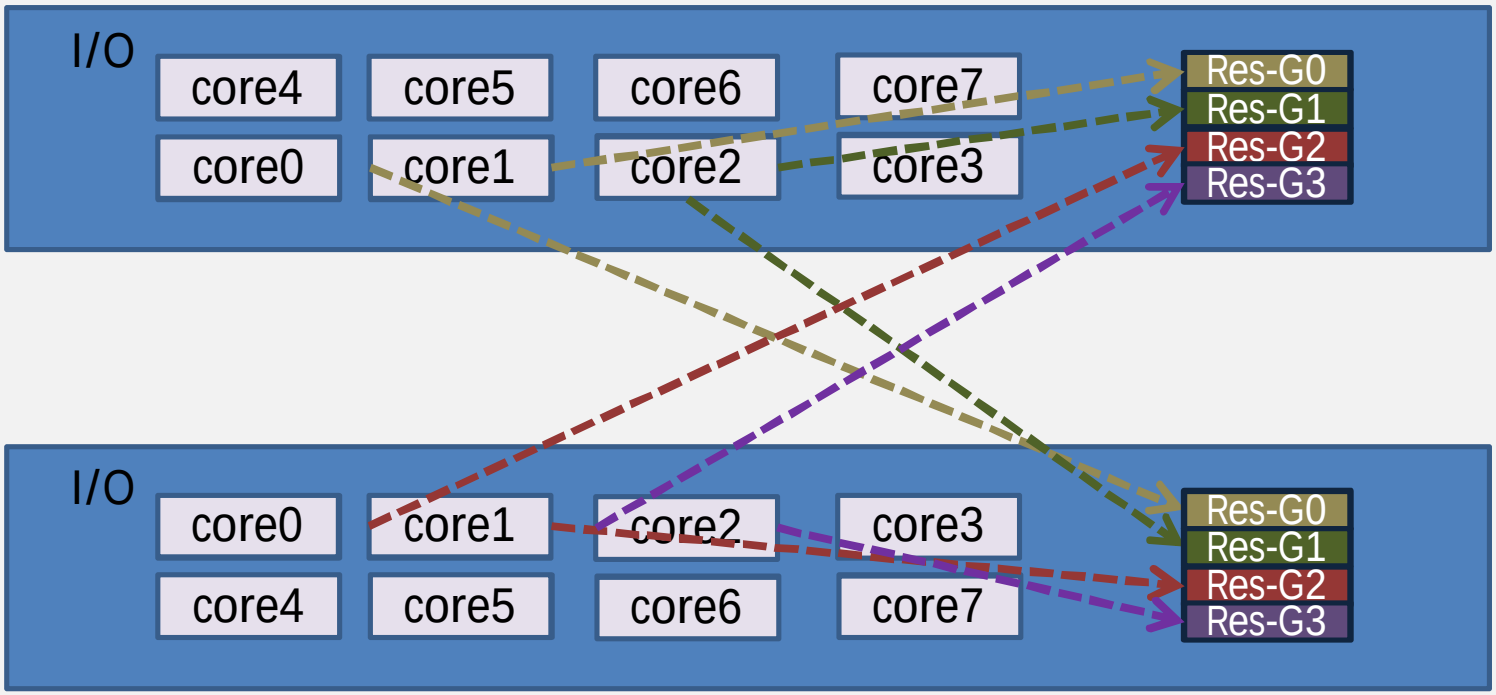
3) Core 1 and 2 spawn and controls clusters computation.



Fault Tolerance Approach on MPPA

3)

- Core 1 and 2 save the results in I/O memory and send the results to the other I/O(core0) via intercluster communication.



Fault Tolerance Approach on MPPA

4) • core 4, core 5 and core 6 of each I/O take the results and each one votes independently of others to obtain the correct result.

5) • core 3 votes based on the responses of core 3, core 4 and core 5 and sends the response to other I/O, including the number of voters that agree.

6) • core3 of I/O 0 logs the correct results .

Core 7 of I/O 0 is the fault injector. → It selects randomly the instant, the core, the register and bit. → It sends an interrupt to the cluster that controls the selected core. → Once in the cluster, the core selected is interrupted via an interprocessor interrupt.

OUTLINE

- Introduction
- Motivation
- Background
- Work Done
- **CONCLUSIONS**

CONCLUSIONS

A comparison of both scenarios SMP y AMP shows that the dynamic response of the device depends not only on the application but also on the adopted multi-processing mode.

A work of De Witte et all. compares the performance of the SMP and AMP modes both with operating systems for a dual-core giving as a conclusion that SMP outperforms the AMP mode. Inferring this affirmation to our work, it is possible to suggest the existence of a trade-off between reliability and performance according to the multi-processing mode selected.

CONCLUSIONS

Designers can improve the dependability of systems through minimizing the consequences of these effects by: Error-correcting codes in memories, error-reporting architectures (machine-check-error registers), etc. Nevertheless, there are some chip areas that remain unprotected.

The inherent redundancy capability of many-cores makes them ideal for implementing fault tolerant techniques such as N-modular redundancy which applies majority-voting.

Fault Tolerance in many-core through redundancy must be evaluated in terms of reliability, power consumption and performance.