HNSJ Humanitarian & Natural Sciences Journal
ISSN: (e) 2709-0833
www.hnjournal.net
Peer-Reviewed Journal
Indexed at: NSP - SJIF
Arabic Impact Factor 2020 = 0.44

RESEARCH TITLE

CONTINUITY OF MEMORY OPERATION IN PRESENCE OF ERRORS

Samira Abu Shernta[1], Ali A. Tamtum[2*], Khalid S . Aleja [2*], Mustafa S. Agha[2*], Khairyah F. Alhadar [5*]

[1,2,3,4*,5] Department of Electrical and Computer Engineering, Elmergib University, Libya

[1] *sm.aboushrintah@elmergib.edu.ly* , [2]*aatamtum@ elmergib.edu.ly* , [3]*ksaleja@elmergib.edu.ly*,

[4*]*msaghamusa@elmergib.edu.ly*, [5]*kheahaddr132@gmiail.com*

*Corresponding author email:

HNSJ, 2022, 3(5); https://doi.org/10.53796/hnsj3535

Published at 01/05/2022 **Accepted at 20/04/2022**

Abstract

Modern critical systems require continuity of operation which requires computing systems that utilize fault tolerance criteria. The operation of systems in the presence of faults is essential for safety and reliability such as in electric power distribution systems, telecommunications, medical life-support, nuclear reactor control, transportation, automotive, aircraft, and space vehicles. Continuity and reliability of service in such systems is essential. To meeting the severe reliability requirements inherent in certain future computer application, the technique of Triple Modular Redundant (TMR) is used. Essentially, this technique depends on voting two out of three system output levels. In this paper a fault-tolerant system is proposed using TMR configuration for memory module with spare model for - line self - reconfiguration. A voter is designed to pass reliable data and signals to the output. The voter has the ability to analyze the error and stop the system on the proper time. The proposed system is tested using MATLAB simulation. A set of different faults are injected in different modules of the system in different data pater. The simulation results demonstrate the capability and accuracy of the proposed system with in the presence of faults as well as the proposed system ability of errors handing.

Key Words: Triple Modular Redundant (TMR); Critical systems; Memory; Error detection.

Introduction

Recently, many techniques have been introduced to overcome limitation of system failure when faults occur. Triple Modular Redundant (TMR) configuration is one of the most important techniques that used for meeting the severe reliability requirements in smart computing systems. Continuity and reliability of service while operating in the presence of limited faults are the goals of this study. The author in [1] stated that the TMR technique required tight synchronization between different units which achieved by using a single and very reliable clock to insure continuity of operation in fault tolerant systems. A fault tolerant system is a system that its behaviour is compatible with its specification in presence of faults in some of its components [2]. To fulfil the two main conditions, continuity and reliability, fault tolerant techniques are necessary to make sure that the system is a fault-tolerant system which continues to operate satisfactory in the presence of faults [3]. Many publications represent different types of faults in different operating systems. In [4] and [5] the authors represented detailed information regarding fault time latency and transient faults. Error detection checks that are employed in computer systems can be of different types, depending on the system and the fault of interest. Most error detection mechanisms are presented in [6] where the authors clearly presented and compared them. Choosing the best error detection technique and the class of fault the technique is best fitted is presented in [7] where several fault handling techniques and their implementation as well as the classes of faults are presented in. Multi-Version techniques based on the use of two or more versions or "variants" of a piece of software, executed either in sequence or in parallel are presented in [8]. Dynamic recovery is generally more hardware-efficient than voted systems, and it is, therefore, the approach of choice in resource-constrained systems especially in high performance scalable systems.. Its disadvantage is that computational delays occur during fault recovery where fault coverage is often low and special operating systems may be required [9]. The authors in [5, 10, 11] introduced mechanisms of error prediction and error coverage. In [12, 13], the authors used other error detection methods such as Watchdog timers which have been used since the early days of digital systems especially in embedded systems .

The concept of redundancy implies the addition of information, resources, or time beyond what is needed for normal system operation. The redundancy can take one of four forms, including hardware redundancy, time redundancy, software redundancy, and information redundancy. The concept of hardware redundancy became more common and more practical, the cost of replicating hardware within a system is decreasing simply because the cost of hardware are decreasing. The Hardware redundancy means the addition of extra hardware, usually for the purpose either detecting errors or tolerating faults[14].

The most known hardware fault tolerance technique is triple modularity redundancy (TMR), which has been used in many fault tolerant systems. The use of TMR technique and its advantages as well as the use of multistage TMR with replicate voters are presented in [15] and [16]. In [17], a commodity chip multiprocessors (CMP) design with features for providing system-level soft error protection, is described with dual modular redundant (DMR) and triple modular redundant (TMR) systems. In [18], A hypothetical triple-modular redundant computer is subjected to a Monte Carlo program on the IBM 704, which simulates component failures. Two types of namely duplex and triple modular redundancy (TMR) systems are presented in [19]. More application and representations of TMR are presented in [20-22]

In this paper a fault-tolerant system is proposed using TMR configuration for memory module with spare model for –line self- reconfiguration. A voter is designed to pass reliable data and signals from memory modules to the output.

The Proposed TNR System

TMR Technique Review

The most known hardware fault tolerance technique is triple modularity redundancy (TMR), which has been used in many fault tolerant systems. The hardware unit (M) represented in Figure1 is triplicated and all three units work in parallel. The outputs of these three units are given to the voting element (V). The voting element accepts the outputs from the three modular and delivers the majority vote as output.

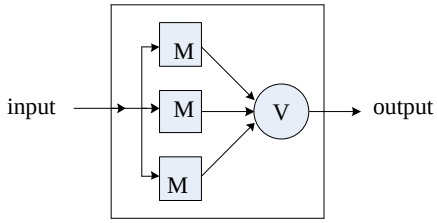

Figure 1: *Triple Modularity Redundancy (TMR) organization*

Clearly, the TMR organization can completely mask the failure of the one hardware unit. One of the features of TMR is that no explicit actions need to be performed for error detection, recovery, etc, TMR is particularly suitable for transient faults, since in the basic TMR the voter does not "remove" the faulty unit after an error occurs. This scheme cannot handle the failure of two units. In fact, once one unit fails, it is essential that both units should be work correctly (so that the voter can get a majority voted output). Due to this, the reliability of the TMR system becomes lower than a simplex system once a failure occurs.

The TMR scheme depends on the voting element. However, the voting element is typically a simple and highly reliable circuits. Another implementation aspect of TMR is that it requires tight synchronization between the different units. This has been frequently achieved by using a single clock. This requires the clock to be very reliable.

The Proposed System

Triple Modular Redundancy (TMR) configuration is the most efficient method to tolerate many types of faults and masking many types of errors at the system level. It is suitable for real time applications and online system reconfiguration where instant maintenance is not possible such as in Autopilot and unmanned space vehicles. This configuration tolerates the following set of faults:

- Faults effecting the operation of memory modules and system buses.
- Faults produced from programs, compilers used to produce those programs
- Design and manufacturing faults in memory modules.

Whereas the set of occurred errors that can be masked by this configuration includes the following classes:

- Data bus errors.
- Address bus Errors.
- Control and timing bus errors.
- Memory transient errors.
- Memory intermittent errors.
- Memory buses errors.

In a TMR configuration permanent errors caused by any faulty module are detected but not tolerated. Therefore the faulty module has to be replaced by a good one in order to resume system functions.

Real-Time applications cause long down-time and increases Mean Time To Repair MTTR. In such a system the MTTR should be zero in order to recover from those errors and to continue system operations to achieve a high reliability.

To overcome a wide range of those errors and to tolerate that set of faults, a good configuration is proposed for a high reliable and available system with Self-Reconfiguration. In this proposed configuration, memories modules are treated separately as the memory modules form also TMR subsystem.

According to this proposed configuration, the three memories (1,2,3) work in parallel and execute the same code and perform the same task. All signals outgoing from these memories are passed through a voter that compares these signals and passes the majority matched ones. If one memory does not match with the other two then the selected majority output from the voter is passed to the memory modules (or to the external I/O devices). Then that memory or its system bus is considered faulty and is given a time to recover from transient faults. If the same memories shows faulty outputs for more than a pre-specified attempts, it is considered as permanent faulty module and the whole system enters a reconfiguration procedure by bringing the spare memory to replace the faulty one.

The voter should also be designed in such a way to work as a comparator and by pass buffer. The voter should also have the mechanism to reconfigure the system by isolating (disconnecting) a faulty module and invoking (connecting) the spare module. The other task of the voter is to load the invoked memory with the current state of the other two memories by a roll-forward recovery procedure and resuming the system operation.

System Operation

The following assumptions are considered for the proposed system:

1 All three memories are ready to execute the same program.
2 The spare memory is physically connected but logically and electorally disconnected.
3 System is at start state.
4 All named model are loaded with the same copy of the program.
5 Give all general block diagram to used voter and three memories.

The block diagram of the voter are shown in Figure 2 which represents the composition of the Voter.

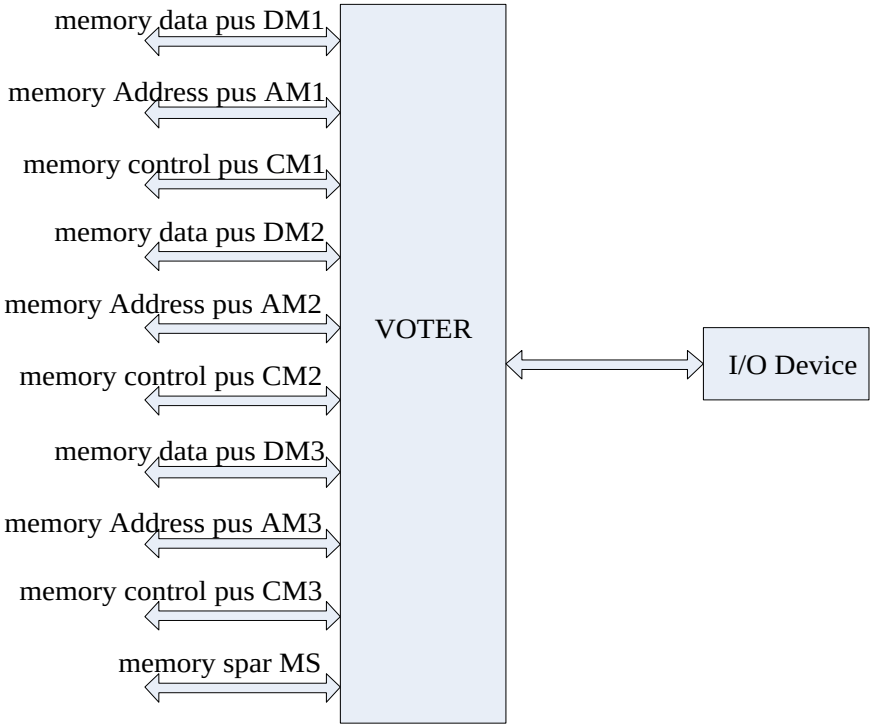

Figure 2: *Block Diagram of the voter*

In the input side of the Voter there are three memories (Data, Address, and Control) as well as a spare memory. Data will be transferred to the I/O devices in case of data saving fail.

Figure 3 represents operating flow chart in which the system starts working by applying Read command. The system is then tested whether it is working or not. If the system working, a check is made on Address, Control and Data. If not, memories are added to the system and the system is tested again. Then, the three memories are tested. In case of error detection, the damaged memory is specified and repaired and the system continues working. Then data writing and saving in the memory is done. On the other side the process of reading data from the memory is running. Then a test is made. In case of an error is detected, error is located and repaired. This process continues until finishing the desired job.

Then data will be transmitted to the processor and then to the I/O devices.

There are two cases "Yes" or "No".

"yes" means there is an error and another test will start to know whether the error is permanent or not. The voter will know in which memory the error occurs.

" no" means that the error is transient and it may be regain by doing the operation again. Once the faulty memory is known, it will be changed by a spare memory to continue the operation .

Case II: If "NO" the data will be sent to the main processor and then into the input ,output devices.

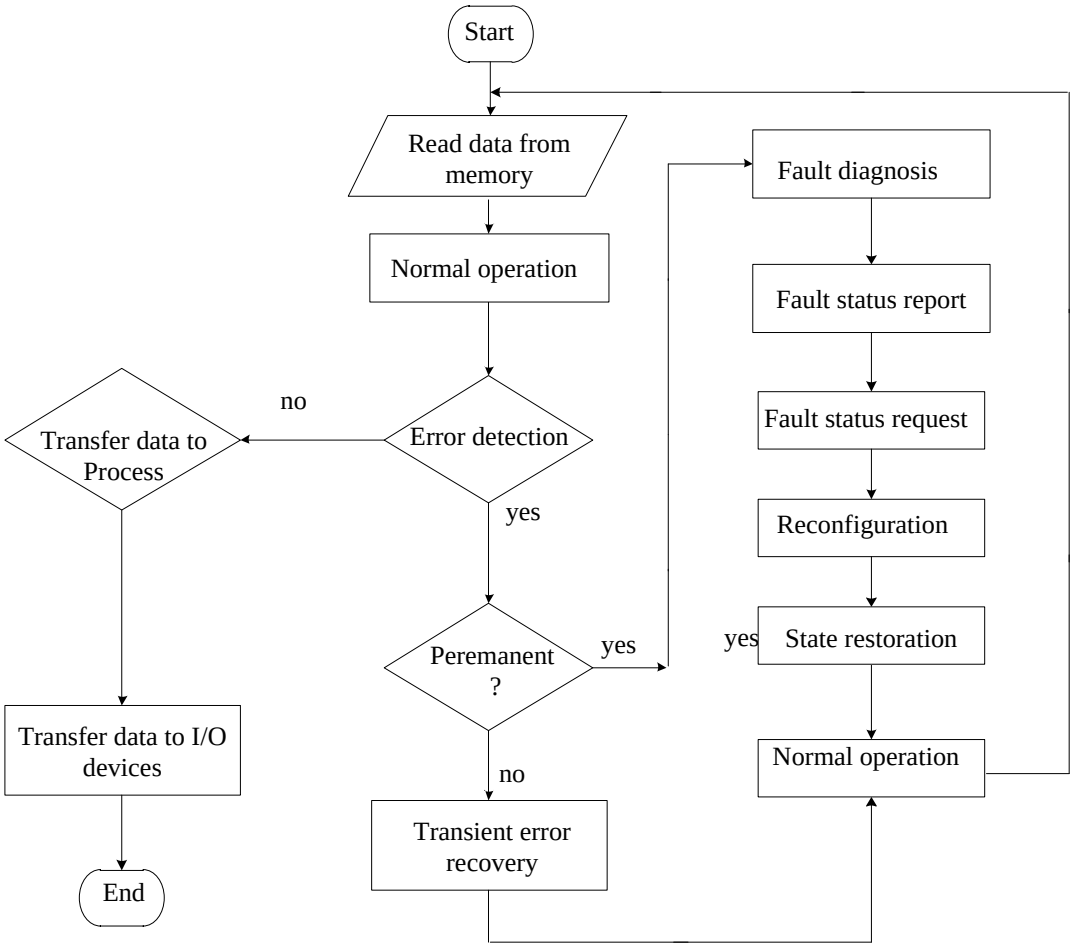

Figure 3: *Operating flow chart*

Numerical results

The results can be summarized in Tables 1, 2, and 3 which represent Memory Data Bus, memory address Bus, Control memory Bus including total time latency.

Table 1: *Memory Data Bus*

Module	Injected faults	Detected faults	Latency (Second)	Coverage	System Recovery (Y,N)
DM1	2	2	2.8352e-005	100%	Y
DM2	0	0	2.8352e-005	0	N
DM3	0	0	2.8352e-005	0	N
Total	2	2	8.5056e-005	100%	Y

A permanent error in DM1 is recorded with latency time = 2.8352e-005 sec; and total latency time = 8.5056e-005 sec; The error in DM1 is temporary and a 100% recovered.

Table 2: *memory address Bus*

Module	Injected faults	Detected faults	Latency (Second)	Coverage	System Recovery (Y,N)
AM1	0	0	2.7243e-005	0	N
AM2	0	0	2.7243e-005	0	N
AM3	6	3	2.7243e-005	3/6*100%=50%	Y
Total	6	3	8.1729e-005	50%	Y

Number of errors =3 and permanent error in AM3. The spare AMs replaces the mean AM3 with 50% coverage.

Table 3: *Control memory Bus*

Module	Injected faults	Detected faults	Latency (Second)	Coverage	System Recovery (Y,N)
CM1	0	0	2.7925e-005	0	N
CM2	3	3	2.7925e-005	3/3*100%=100%	Y
CM3	0	0	2.7925e-005	0	N
Total	3	3	8.3775e-005	100%	Y

The number of errors = 3 and the error in CM2 is permanent. The spare CMs is utilized instead of using the mean CM2.

Conclusions

The concepts and principles of fault tolerance were introduced and investigated. The analysis was devoted to the online error detection and mainly focused on the use of triplication techniques. A

survey of the online error detection techniques and investigation of some previous systems was done, a TMR system configuration was proposed to overcome the limitation of memory operation in presence of errors. In this system the memory module is triplicated with one spare module. A voter was designed to pass reliable data and singles to the output. The voter has the capability to stop the system and analysis the error. It enters the system for roll- back procedure in case of transient error or it replaces the faulty module with the spare one in case of permanent error. Thus system is recovered and resumes its operation on line which achieves the target objective. The system is simulated using MATLAB package to verify the accuracy, capabilities and the behaviour of the proposed system..

References

[1] Chris Weaver , Todd Austin, "A Fault Tolerant Approach to Microprocessor Design" *Advanced Computer Architccture Laboratory University of Michigan*, July 2001.

[2] Teijo Lehtonen , Juha Plosila, Jouni Isoaho , "On Fault Tolerance Techniques towards Nanoscale Circuits and Systems " *Turku Center for Computer Science*, TUCS Technical Report, August2005.

[3] N. Kim and S. Gupta "Testing of Digital Systems" , *Cambridge University press 2003*.

[4] Ali H. Maamar , Asma y. Elhawadi, "Self Checking Register file" *Computer Department Higher Institute of Electronics, Beni- Waled*, APRIL 1999.

[5] Lisboa,C.A. Erigson, M.I. and Carro, l.and carro ,L, "System level approaches for mitigation of long duration transient fanlts in future technologies", *12th IEEE European Tcst Symposium (ETS,07),* 2007.

[6] Manoj Franklin , "A Study of Time Redundant Fault Tolerance Techniques for Superscalar processors",

[7] Januu Sosnowski, "Transient fault Tolerance in Digital System", *Warsaw University of technology, IEEE* , 1994.

[8] Subhasish Mitra, "Diversity Techniques for Concurrent Error Detection" *Technical Report , Center for Reliable Computing* , may 2000.

[9] Parg K Lala "Sef- checking and fault –Tolerance Digital Design", *Morgan Kaufmann Publisher*,2001.

Departmcnt of Electrical &computer Engineering , Clemson Universty ,Clemson, USA, 1995 IEEE.

[10] Stanislaw J.Piestrak," Design of fast self -testing checkers a Class of Berger Codes", *IEEE Transaction on Computer*, MAY 1987.

[11] Kim and K. G. Shin , "Evaluation of Fault Tolerance Latency from Real -Time Application 's Perspectives", *IEEE Transactions on Computers*, vol . 49 No 1, Jan. 2000.

[12] Robert Redinbo, " Generalized Algorithm-Based Fault Tolerance: Error Correction via Kalman Estimation", *IEEE Transactions on Computers*, Vol. 47, No. 6, June 1998.

[13] Robert Redinbo, " Generalized Algorithm-Based Fault Tolerance: Error Correction via Kalman Estimation", *IEEE Transactions on Computers*, Vol. 47, No. 6, June 1998.

[14] Constantinescu, "Teraflops Supercomputer : Architecture and Validation of the Fault Tolerance Mechanisms", *IEEE Transactions on Computers*, Sep 2000.

[15] Karri, K. Kim, and M. Potkonjak, " Computer Aided Design of Fault-Tolerant Application Specific Programmable Processors", *IEEE Transactions on Computers*, Nov 2000.

[16] Dutt and N. R. Mahapatra, "Node-Covering, Error Correcting Codes and Multiprocessors With Very High Average Fault Tolerance", *IEEE Transactions on Computers*, Sept 1997.

[17] James E. Smith, " Motivating Commodity Multi-Core Processor Design For System-Level Error Protection", *Kewal K. Saluja* 2006.

[18] George W. Grosline, "The Use of Triple-Modular Redundancy to Improve Computer Reliability", *IBM Journal*, April 1962.

[19] Rami Melhem, "Energy-Efficient Duplex and TMR Real-Time Systems Appeared in the IEEE Real-Time Systems Symposium", *Computer Science Department, University of Pittsburgh*, Dec 2002.

[20] Dmitry Burlyaev, Pascal Fradet, Alain Girault, "Verification-guided voter minimization in triple-modular redundant circuits", *Automatin & Test in Europe Conference & Exhibition (DATE)*, Year: 2014.

[21] Jeffrey Prinzie, Michiel Steyaert, Paul Leroux, Jorgen Chrisiansen, Paulo Moreira, "A single-event upset robust, 2.2 GHz, to 3.2 GHz, 345fs jitter PLL with triple-modular redundant phase detector in 65 nm CMOS", *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Year: 2016..

[22] Pang Zh, Qi Zheng, Zhankui Zeng, Liman Yaung, "The single integrity design and simulation of triple-modular redundant (TMR) computer", *IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and , IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Year: 2017.