A Fault Tolerant Technique- Triple Modular Redundancy for Distributed control System

Deepti R. Katare
Dept.of electrical engineering
KKWIEER,Nashik
Nashik,India

Prof.N. N. Jangle
Dept.of electrical engineering
KKWIEER,Nashik
Nashik,India

Abstract—Distributed control system has been widely used in the recent years. In this seminar brief description is presented about distributed control system. Hence a distributed control sys-tem (DCS) is a computerized control system, in which controller element are not centrally located but distributed throughout the system. That means it is a type of automated control system that is distributed throughout a machine to provide instructions to different parts of the machine. Instead of having a centrally located device controlling all machines, each section of a machine has its own computer that controls the operation.A Fault tolerant technique is used for the DCS and one of them is Triple Modular Redundancy. Triple Modular Redundancy (TMR) uses three functionally equivalent units to provide redundant backup. Fault Tolerance is a high performance system and they have ability of system to continue error-free operation in presence of unexpected fault. TMR defend the FPGA circuit by creating three copies of a circuit and choosing the output based on a majority vote between the three. For making a fault tolerant system Triple Modular Redundancy (TMR) is used. Triple Modular Redundancy (TMR) is commonly used in dependable systems design to ensure high reliability against soft errors.

Index Terms—*Distributed control system, Fault tolerant, Triple Modular Redundancy, Matlab Simulation.*

I. INTRODUCTION

DCSs are increasingly being applied in many fields in recent years, for example, avionic control, nuclear plant control, process control systems, automatic manufacturing control sys-tems and other autonomic systems, because of their attractive advantages, such as the high control performance, reliability and extensibility. With the increasing complexity of a DCS, the possibility of hardware faults and software failures increases. However, a DCS is a kind of hard real-time system, in which the consequences of not executing a task before its deadline may be catastrophic (for instance, threat to human lives or significant economic loss). Thus, a fundamental requirement of DCSs is to complete all real-time tasks within their specified deadlines even in the presence of faults. Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown. Fault tolerance is particularly sought after in high availability or life-critical systems. The ability of maintaining functionality when portions of a system break down is referred to as graceful degradation. A fault-tolerant design enables a system to continue its intended operation, possibly at a reduced level, rather than failing completely, when some part of the system fails. The term is most commonly used to describe computer systems designed to continue more or less fully operational with, perhaps, a reduction in throughput or an increase in response time in the event of some partial failure. That is, the system as a whole is not stopped due to problems either in the hardware or the software. An example in another field is a motor vehicle designed so it will continue to be drivable if one of the tires is punctured, or a structure that is able to retain its integrity in the presence of damage due to causes such as fatigue, corrosion, manufacturing flaws, or impact In order to obtain the high reliability of an distributed real-time system, several different models (techniques) have been developed to realize fault-tolerance in last several decades, namely, (1) Triple Modular Redundancy (TMR) model, (2) Primary Backup (PB) model, and (3) Recovery Block model.

The second method is Triple Modular Redundancy. In TMR, three processors run redundant copies of the same workload and mask errors by voting on their outputs. Since it requires triplicate hardware, TMR is expensive and used only in the most critical core of fault-tolerant systems. Redundancy is a common approach to improve the reliability and ease of use of a system. The system will be expensive while Adding redundancy and complexity of a system increases with the high reliability of modern electrical and mechanical components, many applications do not need redundancy in order to be successful. However, redundancy may be an attractive option but in case of failure it is very expensive. On the way to clarify Triple Modular Redundancy, it is necessary to elaborate the idea of triple redundancy.

Triple modular redundancy (TMR) is a technique commonly used to provide design hardening. It is used to protect sequential circuits, or storage elements. Conventional TMR technique has been proved effective in protecting sequential circuits Fault

Tolerance is a high performance system and they have ability of system to continue error-free operation in presence of unexpected fault. The system must not suddenly fail but continue executing part of its workload. A fault occurs within some hardware or software component. A fault is due to radiation effect, external disturbance, wear out failures. A fault might not always results in an error, but the same fault may outcome in numerous errors. Similarly a single error may arise a numerous failures. Triple Modular Redundancy (TMR) is the most widely adopted one for hardening circuits implemented on SRAM based FPGAs. Triple Modular Redundancy (TMR) is used for making a fault tolerant system and it can be applied based on different granularities, such as device redundancy, system redundancy, module redundancy or logic element re-dundancy. The Triple Modular Redundancy (TMR) technology allows protection of the functionality of FPGAs against single event upsets (SEUs). Field-programmable gate arrays (FPGAs) gives high-performance for digital signal processing and real-time communication systems. Triple Modular Redundancy (TMR) is the most popular SEU mitigation technique for FPGAs. TMR safeguards the FPGA circuit by creating three copies of a circuit and choosing the output based on a majority vote between the three. TMR also masks the effects of SEUs as well as the less critical transient and soft data errors. While TMR is very effective at protecting FPGA circuits from soft errors, it is expensive in terms of the circuit area, power, and circuit timing. FPGAs are progressively more used in space for reconfigurable radios and other high-performance computing tasks.

II. FAULT AND FAULT TOLERANCE

A. Types of Faults

Transient Fault : appears once, then disappears

Intermittent Fault :occurs, vanishes, reappears; but: follows no real pattern (worst kind).

Permanent Fault :once it occurs, only the replace-ment/repair of a faulty component will allow the DS to function normally

B. Fault Tolerance

Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively designed system in which even a small failure can cause total breakdown. Fault tolerance is particularly sought after in high-availability or life-critical systems. The ability of maintaining functionality when portions of a system break down is referred to as graceful degradation A fault-tolerant design enables a system to continue its intended operation, possibly at a reduced level, rather than failing completely, when some part of the system fails. The term is most commonly used to describe computer systems designed to continue more or less fully operational with, perhaps, a reduction in throughput or an increase in response time in the event of some partial failure. That is, the system as a whole is not stopped due to problems either in the hardware or the software. An example in another field is a motor vehicle designed so it will continue to be drivable if one of the tires is punctured, or a structure that is able to retain its integrity in the presence of damage due to causes such as fatigue, corrosion, manufacturing flaws, or impact. A highly fault-tolerant system might continue at the same level of performance even though one or more components have failed. For example, a building with a backup electrical generator will provide the same voltage to wall outlets even if the grid power fails. A system that is designed to fail safe, or fail-secure, or fail gracefully, whether it functions at a reduced level or fails completely, does so in a way that protects people, property, or data from injury, damage, intru-sion, or disclosure. In computers, a program might fail-safe by executing a graceful exit (as opposed to an uncontrolled crash) in order to prevent data corruption after experiencing an error. A similar distinction is made between ”failing well” and ”failing badly”. Fail deadly is the opposite strategy, which can be used in weapon systems that are designed to kill or injure targets even if part of the system is damaged or destroyed. A system that is designed to experience graceful degradation, or to fail soft (used in computing, similar to ”fail safe) operates at a reduced level of performance after some component failures.

III. TRIPLE MODULAR REDUNDANCY

Triple Modular Redundancy (TMR) is the most widely adopted one for hardening circuits implemented on SRAM based FPGAs. Triple Modular Redundancy (TMR) is used for making a fault tolerant system and it can be applied based on different granularities, such as device redundancy, system redundancy, module redundancy or logic element redundancy. The Triple Modular Redundancy (TMR) technology allows protection of the functionality of FPGAs against single event upsets (SEUs). Field-programmable gate arrays (FPGAs) gives high-performance for digital signal processing and real-time communication systems. Triple Modular Redundancy (TMR) is the most popular SEU mitigation technique for FPGAs. TMR safeguards the FPGA circuit by creating three copies of a circuit and choosing the output based on a majority vote between the three. TMR also masks the effects of SEUs as well as the less critical transient and soft data errors. While TMR is very effective at protecting FPGA circuits from soft errors, it is expensive in terms of the circuit area, power, and circuit timing. FPGAs are progressively more used in space for reconfigurable radios and other high-performance computing tasks. Triple Modular Redundancy is widely used in dependable systems design to ensure high reliability against soft errors. Conventional TMR is effective in protecting sequential circuits but cant recover soft errors in combinational circuits. A new redundancy technique called the Space-Time Triple Modular Redundancy is discussed in this paper, which improves the soft error tolerance of the combinational circuit. Triple modular redundancy (TMR) is a technique commonly used to provide error free circuit. It is used to protect sequential circuits, or storage elements. Conventional TMR technique has been proved effective in

protecting sequential circuits . TMR in FPGAs a Space application must consider the effect energetic particles (radiation) can have on electronic components. SEUs may modify the logic-state of any static memory element (latch, flip flop, or RAM cell) or cause transient pulses in combinatorial logic paths. Since the user programmed functionality of an FPGA depends on the data stored in millions of configuration latches within the device, an SEU in the configuration memory array might have adverse effects on the expected functionality of the user implemented design. Above Fig.1 gives information about Structure of TMR

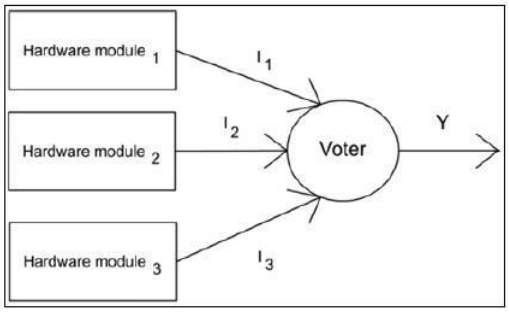

Fig. 1. Structure of TMR

Similarly, Single Event Transients (SETs) have a high prob-ability for recognition at flip flop inputs where, if registered, causes a soft-error in the user data. Static upsets in the configuration memory are not necessarily synonymous with a functional error; however, soft-errors are by definition a functional error. Upsets might or might not have an effect on functionality. However, an gathering of upsets in the con-figuration memory is eventually certain to lead to a functional failure. Design mitigation techniques, such as triple module redundancy, can harden functionality against SEUs and SETs, while the SEUs are corrected so that static-errors do not accumulate and soft-errors do not propagate. Implementing triple redundant circuits in other technologies, such as ASICs, is traditionally limited to protecting only the flip flops of the users design from SEU, because logic paths in linking the flop-flops are typically hard-wired, non-reconfigurable gates. For such fixed logic technologies, this is sufficient protection from SEUs, but can still leave the circuitry vulnerable to SETs. For a technology that is vulnerable to SETs, further protection can be achieved through full module redundancy. Full module redundancy is the required implementation of TMR in FPGAs, because all the logic paths, not just the flip flops, are susceptible to SEUs. This means that three full copies of the base design will be implemented to protect circuit functionality from SEUs, as well as SETs. However, the method for constructing TMR circuitry for Virtex FPGAs provides the additional advantages of complete data retention and independent recovery. The correct implementation of TMR circuitry within the Virtex architecture depends on the type of data structure to be mitigated. These data structures can have categorize into four different types like throughput logic, state-machine logic, I/O logic, and special features

A. FPGA

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing hence ”field-programmable”. The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). (Circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare.) FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be ”wired together”, like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory Field-programmable gate arrays (FPGAs) are an attractive target for high-performance digital signal processing and real-time communication systems. FPGAs have been used to implement communication-specific processors for well over a decade. Their ability to combine flexibility with good per-formance makes FPGAs popular for software-defined radios. Reconfigurable radios are also becoming more attractive for space-based applications. The ability to reconfigure the FPGA resources with an updated radio configuration reduces the amount of hardware needed on the spacecraft. FPGAs are increasingly used in space for reconfigurable radios and other high-performance computing tasks. The problem with using the popular SRAM- (static-random access- memory-) based FPGAs in space is the presence of high-energy particles that may alter the operation of the digital circuitry or the state of static memory cells. These errors, called soft errors, do not cause any physical damage to the device but interact with state of memories or other digital circuits. For example, charged particles can occasionally invert the contents of a memory cell. Such an event is called a single event upset (SEU). Because most of the FPGA area is devoted to static memory cells to store the FPGA configuration memory, FPGAs are very sen-sitive to radiation. Any FPGA design operating in space must consider the effects of high-energy radiation and implement some form of SEU mitigation. Triple modular redundancy (TMR) is the most popular SEU mitigation technique for FPGAs. TMR protects the FPGA circuit by creating three copies of a circuit and choosing the output based on a majority vote between the three. TMR masks the effects of SEUs as well as the less critical transient and soft data errors. Although TMR is very effective at protecting FPGA circuits from soft errors, it is costly in terms of the circuit area, power, and circuit timing.

B. Theoretical basics of triple modular redundancy

The great majority of researchers involved in fault-tolerant digital systems design and development generally agree that the physical replication of hardware is the most common form of hardware redundancy implementation. This means that very similar hardware units or modules executes the same functions and attributes in the fault-tolerant digital system, implementing the same tasks and control strategies. When one of those mod-ules enters in the failure state, a fault-free unit is ready to take

over the faulted ones functionality and attributes. Such kind of physical redundancy can be implemented with three main strategies: passive-, active-, and hybrid techniques. Briefly ex-pressed, the passive hardware redundancy means that the fault-tolerance is achieved by masking the occurred fault, without requiring any intervention on the part of system or operator. In fact, this technique is based on the idea to hide occurrence of faults rather than detect them. In this way the hardware faults are masked, and prevent faults from resulting in errors. The active approach of hardware redundancy (often also called dynamic method) implements fault tolerance by detecting the existence of faults and performing same certain actions to remove the faulty hardware from the considered digital system. Therefore, the active fault detection technique also uses fault location and fault recovery methods in an attempt to achieve fault tolerance. It means that the system can be physically reconfigured to tolerate faults. Obviously, hybrid techniques combine the advantages of both passive and active approaches. However, as the microelectronic components becomes smaller and less expensive, in a same way the expenses of replicated hardware decreases continuously and the hardware redundancy concept become more practical. Certainly, the most common form of passive hardware redundancy is the TMR. There three perfectly identical modules perform the same functions and tasks inside the digital system with a majority decision element determining the output of the system. Usually this last unit it is named voter, by performing a majority vote strategy over the outputs of the three hardware modules. Therefore, if one of the modules enters into a faulty state, the two remaining fault-free modules mask the occurred fault in the digital system. In fact, the passive hardware redundancy technique generally relies on the majority voting mechanism to hide (or mask) the occurred faults and errors inside a hardware system. The block diagram of triple modular redundancy. This concept (also named 2/3 redundancy) was originally envisaged by Neumann . If it is considered that a hardware module operates correctly (without faulty) by emitting the signals Ii (i = 13) to the inputs of the voter, the operation of the TMR system it is described by the logical equation:

$$Y = I_1I_2 + I_2I_3 + I_1I_3 \quad (1)$$

The reliability R(t) of the system is a function of time and it is expressed by the probability that the system will operate correctly throughout the time interval [0, t] (considering that was performing correctly at time t = 0). In a hardware system with n modules connected in parallel the global reliability it is expressed by the mathematical relation

$$R(t) = \prod_{j=1}^{n} Rj(t) \quad (2)$$

$$=1$$

where Rj (t) means the reliability of each component module of the system (j = 13). If these modules are connected in a serial configuration, the above equality transforms into the equation

$$R(t) = 1- \prod_{j=1}^{n} 1 - Rj(t)$$

C. Mathematical analysis of a TMR computer

Triple redundancy with perfect voting circuits, to explain triple-modular redundancy, it is first necessary to explain the concept of triple redundancy as originally envisaged by Von Neumann. The concept is illustrated in Fig. where the three boxes labeled M are identical modules or black boxes which have a single output and contain digital equipment. (A black box may be a complete computer, or it may be a much less complex unit-for example an adder or a gate.) The circle labeled V is called a majority organ by Von Neumann. In this paper it will be called a voting circuit because it accepts the input from the three sources and delivers the majority opinion as an output. Since the outputs of the Ms are binary and the number of inputs is odd, there is bound to be an unambiguous majority opinion. Above Fig.2 gives information about ven neumen TMR

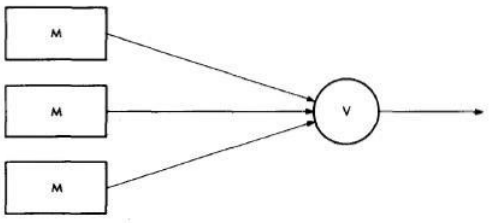

Fig. 2. Ven Nuemen TMR

The reliability of the redundant system illustrated in Fig. 1 is now determined as a function of the reliability of one module, R, assuming the voting circuit does not fail. The redundant system will not fail if none of the three modules fails, or if exactly one of the three fails. It is assumed that the failures of the three modules are independent. Since the two events are mutually exclusive, the reliability R of the redundant system is equal to the sum of the probabilities of these two events. Hence,

$$R = Rm3\text{-}3Rm2(1 - Rm) = 3Rm^2 - 2Rm^3 \quad (4)$$

D. Simulation and Result

The theoretical approaches introduced in the previous paragraph represent an adequate background to modeling and simulate various fault-tolerant structures using the TMR strategy.

For a more convincing presentation, in a first step a redundant system embedding analogue voter has been simulated in Matlab/Simulink software environment. Therefore,

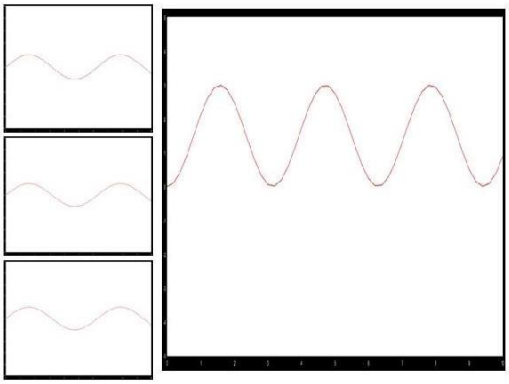

Fig. 3. TMR waveform

it has been considered that the three hardware modules generate analogue signals to the voter inputs. If there are no errors in the system, Fig shows that the three signals (for example sinusoidal waveforms) on the voter inputs are perfectly identical, and the majority decision element outputs the same sinusoidal signal.

This Figure presents the situation when one of the consid-ered hardware modules (for example the unit labeled with 1) enters in a faulty state. In this situation the voter element masks the occurred fault and according to the majority decision logic considers that the sinusoidal waveforms are the right ones and generate this signal to the output.

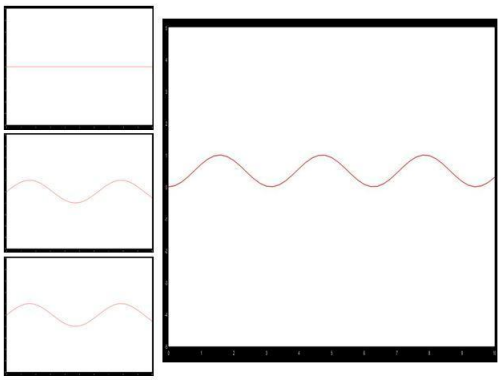

Fig. 4. Fault occurs in 1

The next Fig.present that, it is possible to observe that in case of two hardware modules simultaneously fault the voter outputs the waveform corresponding to the faulty state. Obviously, this is the expected result, because the TMR strategy is able to handle only one fault or error situation in the considered hardware system. Similar waveforms can be also plotted when the system embeds digital voter.

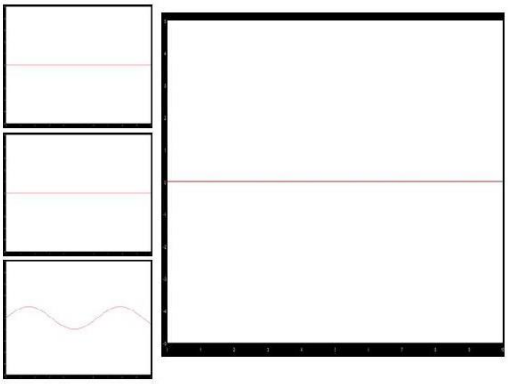

Fig. 5. Fault occurs in 2

IV. TRIPLE-MODULAR REDUNDANCY WITH PERFECT VOTING CIRCUITS

Figure illustrates the triple-modular-redundant configuration that will be used in this analysis. This configuration differs from the one shown in Fig. 1 because it employs three identical voting circuits instead of the one voting circuit previously used. If it is assumed that the voting circuits do not fail, the two configurations have identical reliability.

cong 1.jpg

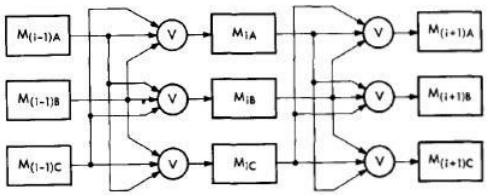

Fig. 6. Ven Configuration

Later, when the unreliability of the voting circuits is taken into account, it will be observed that the voting circuits themselves are redundant in the configuration. Hence single voting circuit failure will not necessarily cause computer failure. Following assumptions are made

1) The non redundant computer is divided into m modules
2) Each module has just one input and one output.
3) The voting circuits do not fail
4) The failures of the modules are statistically independent
5) The modules m are equally reliable

V. CONCLUSION

Considering that DCSs are subject to hardware and software faults, I have presented a fault-tolerant scheduling algorithm named BNPRMFT. Compared with other fault-tolerant scheduling algorithms, BNPRMFT can tolerate not only hardware faults, but also software faults. In our fault-tolerant scheduling algorithm, every task has a primary copy and a backup copy which are independent and assigned to different processors according to a heuristic algorithm which can balance the loads of primary copies and backup copies on each processor. A backup copy is executed only when its corresponding primary copy fails due to a fault. A notification time (NT) is set for a task, before or at which backup copy must start, otherwise it cannot be finished before its deadline. Unlike other fault-tolerant scheduling algorithms for hardware faults, BNPRMFT can execute as many primary copies as possible due to their high control performance. Unlike other algorithms for software faults, BNPRMFT can tolerate hardware faults by executing backup copies assigned to different processors. In order to lower the cost of the algorithm, non-preemptive RM has been employed to schedule primary copies and backward non-preemptive RM has been applied to calculate notification times of tasks in order to leave more time for executing primary copies. Finally, computer simulation has been carried out to testify BNPRMFT. Compared with BPRMFT, BNPRMFT can gain a higher success rate in executing primary copies and lower the runtime overhead for the algorithm implementation I have also presented the review on the fault-tolerant and repairing technique using Triple modular redundancy (TMR). This Literature survey gives many concepts in both TMR design and FPGA selection. Traditional solutions for radiation effects were introduced including hardware redundancy and software improvement for fault tolerance, like time redundancy or software redundancy. Along with the explanation of Triple Modular Redundancy it has been given both the theoretical analysis and also the mathematical analysis of the major-ity voter. The paper presents a fault-tolerant implementation strategy by using the triple modular redundancy concept. This passive hardware redundancy approach allows the faults masking inside digital circuits and represents a very convenient method to implement not expensive and reliable voter elements operating on the majority decision concept.

REFERENCES

[1] Y. Rui, C. Qinqin, L. Zengwu, S. Yanmei, Multiobjective evolutionary design of selective triple modular redundancy systems against SEUs, Chinese Journal of Aeronautics 28,(2015)

[2] C. CARMICHAEL, Triple Module Redundancy Design Techniques for the Virtex TM Series, Xilinx Application Note xapp197, 2001.

[3] Brian Pratt, Megan Fuller, and Michael Wirthlin, Reduced-Precision Re-dundancy on FPGAs International Journal of Reconfigurable Computing, Volume 2011 (2011),

[4] Hung-Manh Pham, Se bastien Pillement, and Stanisaw J. Piestrak, Low-overhead fault-tolerance Technique for a Dynamically Reconfigurable Softcore Processor Transactions on Computers, VOL. 62, NO. 6, JUNE 2013

[5] Johnson B. W. (1989), Design and Analysis of Fault-tolerant Digital Systems. Addison-Wesley series in electrical computer engineering, ISBN 0-201-07570-9.