Defence Science Journal, Vol. 52, No. 1, January 2002, pp. 33-39

Optimising Model for Memory Fault Tolerance in Onboard Computer

Suresh V. Mathew, Sasikumar Punnekkat and Abdul Salam

Vikram Sarabhai Space Centre, Thiruvananthapuram - 695 022

ABSTRACT

This paper presents an optimising model for integrating the traditional reliability prediction methodology with simple analytical techniques to facilitate the designer to decide upon the memory fault-tolerant choices of an onboard computer. In this exercise, the hardware reliability estimates of a circuit without any error correction as well as that of a circuit with error detection and correction were calculated. The failure rates of each component and soldering have been accounted for in these prediction procedures. A suitable probability distribution is chosen for data errors and is analytically combined with the hardware reliability predictions to study the trade-offs. An optimum strategy for introducing the hardware error correction logic in the circuit is presented.

Keywords: Reliability, fault tolerance, optimising model, memory errors, error correction logic, onboard computer, mission critical systems, real-time systems

1. INTRODUCTION

Study of the possible fault tolerance options and associated reliabilities is of utmost importance in safety/mission critical systems, such as onboard computers of launch vehicles. Modern mission critical real-time systems, such as onboard computers in launch vehicles, have a very high requirement for reliability. This is achieved using both fault avoidance and fault tolerance design. Through fault avoidance techniques, the elemental stability of hardware and software is increased. For hardware fault avoidance usage of high quality components like MIL 883B class, thorough screening, derating, packaging (to protect from heat, shock, vibration), test and evaluation are the general methods. In the case of software, the methods employed include requirement analysis, configuration control, code walk through, and verification and validation.

To realise fault tolerance, redundancy in space, time or combined domains is necessary. The aim is to maintain correctness and timelines even in case of a fault. Due to the real-time nature of these systems, it is essential that the exploitation of time redundancy does not jeopardise the timeliness attribute of the system[1] and it is customary to adhere to static redundancy techniques. Of late, the relevance of memory hardware soft errors has been realised and subjected to extensive studies[2]. In case of memory circuits, mirror images are employed to assist in error detection. Hence in most of the real-time systems, data error detection and correction logic is hardwired. This calls for addition of hardware, which in turn will increase the complexity and thereby reduce the reliability. Hence, an optimising model is required to assess the associated trade-offs.

Traditional reliability prediction methods, such as parts count and part stress take into account only the hardware aspects, such as quality of components, derating, environment of operation, etc. It generally

Received 03 November 2001

uses exponential model of reliability distribution specific to the hardware characteristics. However, probability distribution of data errors cannot be modelled using exponential distribution. Recent research in reliability modelling has come up with Markov models for system error states. Usually the system modelling tends to become more complex and several advanced software tools have been developed[3] to manipulate the state vector matrix to arrive at state probability. However, these techniques have not so far gained wide acceptance by the designers.

In this context, an attempt has been made to bring out a simple approach combining traditional reliability prediction with a probability distribution of data errors. This approach gives a theoretical framework to establish an optimum strategy to incorporate an error detection and correction circuit. A case study of memory circuits of an onboard computer is presented. Traditional reliability prediction in conjunction with suitable analytical techniques has been applied to compare reliability figures of two hardware circuits and to assist the designer in choosing the one with optimum reliability.

2. SYSTEM DESCRIPTION

An onboard computer in a launch vehicle carries out vital functions, such as navigation, guidance and control. Major tasks executed by the onboard computer are digital autopilot, guidance computation and sequencing functions. The control-related functions are executed as periodic tasks, where the periods of execution are few milliseconds. The architecture of the onboard computer employs dual redundancy with cross-strapping[4]. All application programs reside in PROMS, while parameters amenable to last minute changes are kept in RAM. The CPU is a 32-bit processor. A real-time software executive manages the resources of the computer and schedules various tasks. Memory circuits contain a bank of 128 x 8 K RAM and controlled by field programmable gate array (FPGA) chip.

3. MODEL INTEGRATING HARDWARE RELIABILITY WITH DATA ERRORS

Due to the mission critical nature of onboard computers, the designer is inclined to add as many fault-tolerant features as possible for the computer. This approach results in added weight, cost and eventually a reduction in overall reliability. The level of redundancy to be incorporated in such a system involves a critical judgement about various contrasting views and obtaining a reasonable trade-off. To help the designer in this scenario, a general optimising model integrating hardware reliability and data errors is being brought out. It is assumed that the data errors are statistically independent.

3.1 Reliability Model

The circuit without any error correction scheme will be referred to as circuit A. Similarly, the circuit with built-in k-bit error detection and correction be represented as circuit B. Obviously the hardware reliability of circuit B will be smaller as compared to that of circuit A, since circuit B will contain more number of components. However, circuit B will have better availability than circuit A since it will be able to mask the specified number of bit errors. In a complex system with active redundancy, where such a circuit itself becomes a component of the total system, the overall system reliability needs to take into account both the hardware reliability as well as the availability factors (though indirectly). Only then a judicious comparison of the circuits can be made. The model presents solution to one such scenario.

$$\sum_{i=1}^{n} p_i = 1$$

Let h_0 represent the hardware reliability of circuit A and h_1 represent the hardware reliability of circuit B. It is also assumed that both h_0 and h_1 are given by standard exponential distributions. Let X be the random variable representing the number of bits in error, i.e., $X = 0$ represent no-bit errors, $X = 1$ represent single-bit errors and $X = k$ represent k-bit errors. Assume that p_i = probability ($X = i$). The term error mode distribution means the fraction of errors of one particular error mode. Suppose k-bit error corrections have been incorporated in circuit B, where $1 \leq k \leq n$, then its effective reliability (comprising both hardware and data) is given by

$$h_1(p_0 + p_1 + \cdots + p_k)$$

So, to compare the reliability of circuit A with circuit B, one needs to compare these.

$$R_0 = h_0 p_0 \text{ with } R_1 = h_1 \sum_{i=0}^{k} p_i$$

$$h_0 p_0 < h_1 \sum_{i=0}^{k} p_i$$

i.e.,if $h_0 p_0 < h_1 p_0 + h_1 \sum_{i=1}^{k} p_i$.

i.e.,if $\dfrac{(h_0 - h_1)}{h_1} < \dfrac{\sum_{i=1}^{k} p_i}{p_0}$

So circuit B will be more reliable as compared to circuit A, if $R_0 < R_1$.

i.e., if $(h_0 - h_1)p_0 < h_1 \sum_{i=1}^{k} p_i$ (1)

4. ARCHITECTURE OF MEMORY CIRCUITS-CASE STUDY

The objective of this case study is to make a comparative study of two hardware memory configurations to select one, with better overall reliability. Circuit A has no error correction logic, but can detect the memory errors by comparison. Details are given in Section 4.1. On the other hand, circuit B (details given in Section 4.2) has built-in, single-bit error correction logic using additional hardware. This change in configuration and addition of hardware has resulted in the reduction of hardware reliability. Now the question is to what level of data errors can circuit B hold better overall reliability?

4.1 Function of Circuit A

Circuit A contains (Fig. 1) a bank of 8 RAM chips, each 128 K x 8. This is arranged as a main RAM and image RAM (four each). The control logic for data storage and retrieval is implemented in an FPGA chip. The same data is stored in the main RAM and image RAM. While it is read back, the data in the main and the image locations are compared. If any mismatch is detected, the processing logic will pass on the information to the CPU and onboard computer (OBC) will be switched to the redundant chain.

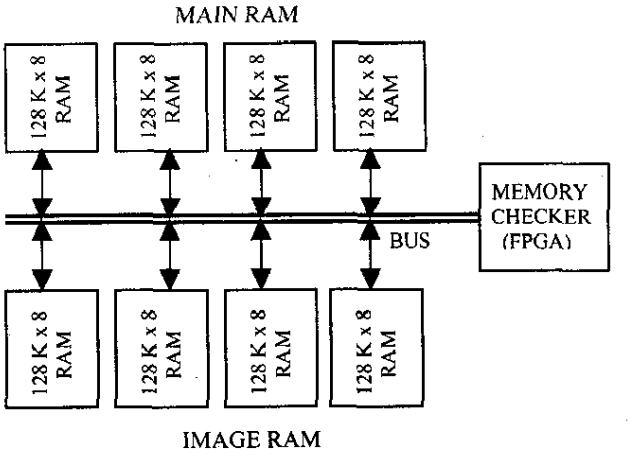


Figure 1. Architecture of circuit A

4.2 Function of Circuit B

In circuit B (Fig. 2), the 8 RAM chips are arranged as data memory and check-bit memory. The control logic for data storage and retrieval is implemented in the two FPGA chips. In this circuit the data and the check bits are read back and single error detection/correction is applied on the

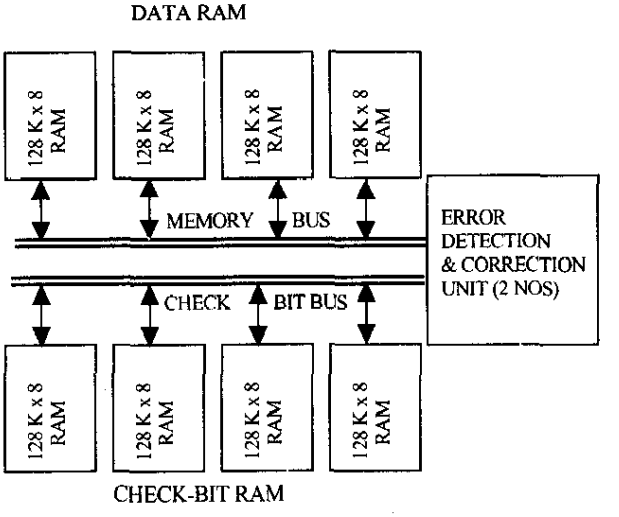


Figure 2. Architecture of circuit B

data. Thus, the circuit takes care of any single bit error. However if it is more than single-bit error, OBC will be switched to redundant chain.

5. RELIABILITY PREDICTION & OPTIMALITY ANALYSIS

As in this case study, it is often the system designer's job to justify the design decisions made from the available options in accordance with the requirements and constraints. Two methods have been explained for arriving at an optimal design choice of whether to use the circuit A having no error correction or the circuit B having memory error detection and correction. The first method is based on traditional analytical approach, whereas the second method shows how one can incorporate the knowledge about error mode probability distributions into one's analysis.

5.1 Assumptions & Conditions

- Reliability analysis is based on the circuit and the input data given by the designer.
- Parts stress method is used for calculation of failure rate.
- Derating details are taken as per the space utility guidelines.
- MIL-HDBK-217F-notice 2 as available in the reliability prediction software-RELEX Reliability Prediction, USA, is used for generating the failure rates.
- Mission time for the calculation purpose is assumed as 1 hr, including the crucial last phase of checkout.
- Exponential model is used for reliability prediction.

5.2 Hardware Reliability

The reliability figures have been computed for both circuits A and B by the parts stress method using RELEX Reliability Prediction package[5] and the values obtained are:

Reliability of circuit A for 1 hr, $h_0 = 0.99999130$

Reliability of circuit B for 1 hr, $h_1 = 0.99998740$

Details of the above calculations are shown in Tables 1 and 2.

5.3 Trade-off using Traditional Method

A memory unit of one byte can theoretically have number of bits in error up to X=8. However, in practice the probability of $X > 2$ is extremely low and those probabilities can safely be ignored. One such reference being, 'if the probability of an error is one in a hundred quadrillion, and if the memory system is running at 100 MHz (10 ns), and if you have 128 MB of RAM (about 1 billion bits), then you would expect on an average to see one single-bit error every one second and one double-bit error every 500 quadrillion seconds (somewhat more than the age of the universe)'[6].

Also, in some systems, the designer may not have the estimates of parameters of the probability distribution function to be used to characterise the number of bit errors. However, the designer might have the error mode distribution or individual probabilities of failures based on experience and past data. In

Table 1. Hardware failure rates of circuit A (with error detection only)

Component name/ part no.	Function	Failure rate per million hr	Quantity	Total failure rate
FPGA, A1425A	Error detection	1.36042	1	1.36042
SRAM EDI 88128	128K x 8 SRAM store the data/check bits	0.61976	8	4.95808
Solder joints	Interconnections	0.00670	410	2.74700
Total				9.06550

Table 2. Hardware failure rates of circuit B (with error detection and correction)

Component name / part no.	Function	Failure rate per million hr	Quantity	Total failure rate
FPGA, A1425A	Error detection and correction	1.36042	2	2.72084
SRAM EDI 88128	128K x 8 SRAM store the data/check bits	0.61976	8	4.95808
Buffer, 54ACT245	Octal bus transceiver	0.18156	4	0.72624
Solder joints	Interconnections	0.00670	632	4.23440
Total				12.63956

such cases, a simplified analysis based on traditional way can be performed as:

Effective reliability of circuit A

R_0= Hardware reliability * probability(no data error)

$= h_0 * (1- q)$

Effective reliability of circuit B

R_1 = Hardware reliability * [probability(no data error) + probability(single-bit error)]

$= h_1 * (1- q + p*q)$

$= h_1 * [1- q*(1-p)]$

where

h_0 = Hardware reliability of circuit A

h_1 = Hardware reliability of circuit B

q = Probability of data error

p = Error mode distribution for a single-bit error

Assuming various values for q, one can calculate the effective reliabilities of circuits A and B. By generating a plot of these reliability values versus the data error probability, it is easy to find the break-even point where both reliabilities are the same. For this case study, the value of p was assumed to be 0.95 as per suggestions of the design team. The reliability values were computed for both the circuits A and B over a range of values for data error probability. The trade-off

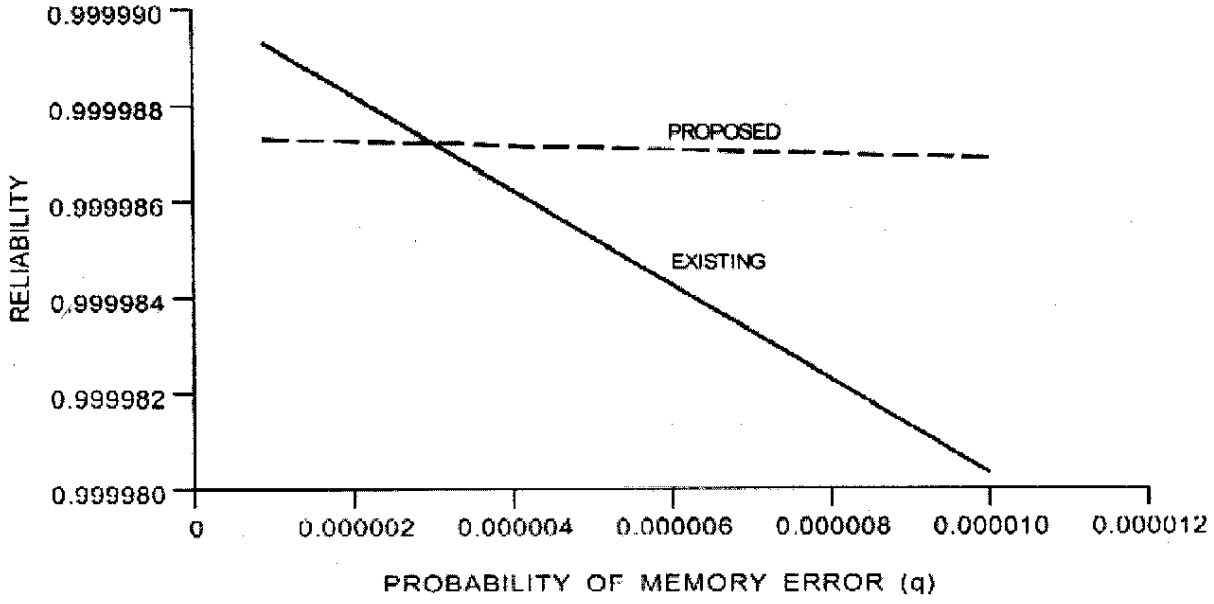


Figure 3. Reliability trade-offs of circuits A and B

graph is plotted in Fig.3. From this graph, it can be seen that both the curves intersect at the point corresponding to a data error probability of 3.2 x 10^{-6}. This implies that the circuit B fares better in terms of reliability and avoids loss of one chain when data error probability exceeds 3.2 errors per 10^6 bytes.

5.4 Optimality Analysis based on General Model

However, one cannot totally rule out the possibility of multiple-bit errors in aerospace environment, as the following example indicates. JAS-2 was a radio satellite developed by the Japan Amateur Radio League, constructed by Nippon Electric Co and was launched on an H-II launch vehicle into a 797 X 1317 km sun-synchronous orbit from Tanegashima Space Centre, Japan, on 17 August 1996. In 1998, it was reported that the satellite suffered a series of 2-bit memory error upsets which made commanding difficult[7]. Since the mission times of satellites are much longer (typically several years), as compared to that of satellite launch vehicles (typically in tens of minutes), there is high risk of exposure to radiation and hence high probabilities for multiple-bit-errors. Hence, the model developed needs to account for such scenarios as well. For events that occur randomly and independently with a random variable (number of successes) X, with a constant rate m per unit time or region, the probability distribution is called Poisson distribution and is given by

$$f(X) = p(X:\mu) = \frac{e^{-\mu}\mu^{X}}{X!} \quad \text{where } X = 0,1,2... \qquad (2)$$

where μ is the average number of successes occurring in the given time frame or region and e = 2.71828.

In the present case, X is the random variable representing X-bit error and success represents happening of such an error. One immediate question may be whether these errors are independent and identically distributed. In the present case, the authors are looking at a region, say a byte of memory and $f(0), f(1),...f(8)$ typically gives them the probability of corresponding number of bits in error, so that they can use $f(i)$ in place of p_i.

Combining inequality (1) and Eqn (2), one can say that circuit B will be better than circuit A,

$$\text{if } (h_0 - h_1)e^{-\mu} < h_1 \sum_{i=1}^{k} \frac{e^{-\mu}\mu^{i}}{i!} \qquad (3)$$

Since only single-bit error correction was employed in Circuit B, the Eqn (3) can be simplified as:

$$\text{Circuit B is better if } (h_0 - h_1)e^{-\mu} < h_1 e^{-\mu}\mu \qquad (4)$$

$$\text{i.e., if } \frac{(h_0 - h_1)}{h_1} < \mu \qquad (5)$$

Using the reliability values as computed for both the circuits A and B (given in Section 5.2), the left hand side of inequality (5) is evaluated to be 3.9 x 10^{-6}. This means that circuit B fares better in terms of reliability and avoids loss of one chain when the expected average data error, m exceeds 3.9 errors per 10^6 bytes. If the expected data error is less than 3.9 errors per 10^6 bytes then circuit A shows better reliability.

The results obtained from both the analyses are reasonably close within the constraints of error data logging and estimation accuracy of the designer. It can be seen that in systems where the error mode probabilities are characterised by some other known probability distribution, a similar analysis can be performed by substituting the corresponding values in inequality-1.

6. CONCLUSIONS

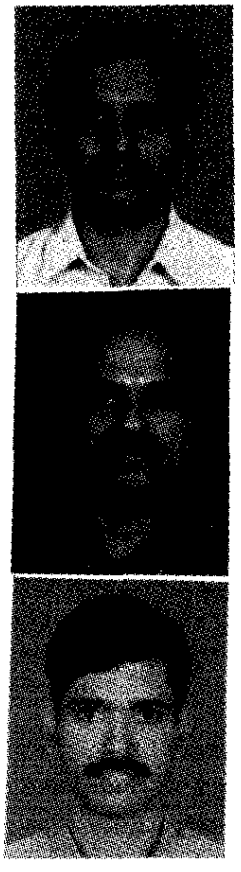
The paper has looked into the need for memory fault tolerance and its implementation in OBC. One of the typical dilemmas related to the reliability of the system, faced by the designers during such an implementation has been described. A traditional method using analytical technique is presented to arrive at an optimum decision on whether to use single-bit error correction or not. Subsequently, a general optimising model is proposed for deciding whether to use k-bit error correction circuit or not.

In this general approach, data errors have been modelled using Poisson distribution. In future work, the authors plan to incorporate other probability distributions as well as to use generic failure models.

REFERENCES

1. Punnekkat, Sasikumar; Burns, Alan & Davis, Rob. Analysis of checkpointing for real-time systems: *Real-Time Syst. J.*,No.1, 2001.
2. Milojicic, Dejan *et al.* Increasing relevance of memory hardware errors-a case of recoverable programming models. Paper presented at the 9th European Workshop of ACM–Beyond PC: New challenges for the operating system. Kokling, Denmark, September 2000.
3. Geist, Robert & Trivedi, Kishor. Reliability estimation of fault-tolerant systems: Tools and techniques. *IEEE Computer*, July 1990.
4. Sudhakara Rao, K. software practices – experience from PSLV mission. Proceeding of the 6th National Convention on Quality and Reliability, Trivandrum, December 1995.
5. RELEX 7 Visual reliability prediction software package. RELEX Corporation, USA. http://www.relexsoftware.com
6. Chattopadhyay, Arijit. Error correction code (ECC) in memory devices– an overview. *In* Lecture notes for ECS154A. University of California, Davis, Fall 2000.
7. Small Satellites Home Page, 1996, University of Surrey, UK. www.ee.surrey.ac.uk/SSC/SSHP/micro/micro96.html

Contributors



Mr Suresh V Mathew obtained his MTech (Industrial Electronics) with distinction from Regional Engineering College (REC), Surathkal (Mysore University) in 1984 after having obtained BSc (Engg) from University of Kerala in 1981. He also received MBA from the University of Kerala in 1998. He joined Indian Space Research Organisation (ISRO) in 1984 and worked in the navigation guidance control area of polar satellite launch vehicle (PSLV) project till 1994. He participated in the development of control electronics for PSLV project. Presently, he is working for the Quality Division Test and Evaluation of System Reliability entity, as the Engineer-in-Charge of Failure Mode it Effects and Criticaly Analysis (FMECA) and Reliability Analysis of Avionics Systems. His areas of research are reliability analysis of mission-critical fault-tolerant systems.

Dr Sasikumar Punnekkat obtained his Master's degree in Statistics (1982) and Master of Technology in Computer Science with honours (1984) from the Indian Statistical Institute. He joined the Indian Space research Organisation in 1984, and was involved in the design, development and testing of software for the PSLV. He was recipient of the Commonwealth Scholarship during 1993-97, and was awarded PhD in Computer Science by the University of York, UK in July 1997 for his research on schedulability analysis of fault-tolerant real-time systems. He was also a Post-Doctoral Research Fellow at the Department of Computer Engineering, Malardalen University, Sweden during Janaury 1999-October 2000. He is presently with the Software Quality Assurance Division of the Vikram Sarabhai Space Centre, India. His research interests span various aspects of real-time and fault-tolerant systems.

Mr Abdul Salam obtained his BTech (Electronics and Telecommunication Engg) from the University of Kerala in 1991. He joined ISRO in 1983 and worked in the Quality Division of Systems Reliability. He has been extensively involved in test and evaluation of avionics packages and is presently involved in failure mode effects and criticality analysis (FMECA) and reliability analysis of avionics systems of satellite launch vehicles.