Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Simulation Based Fault Tolerance Approach for Embedded System Design

[1]Pallavi*, [2]Rajesh Garg, [3]NitikaBansal
[1]Student (CSE Deptt), Ganpati Institute of Technology and Management, Bilaspur, India
[2]Lect. (ECE Deptt) Seth Jai Prakash Polytechnic for Engineering, Damla, India
[3]Assistant Professor (CSE Deptt.) Ganpati Institute of Technology and Management, Bilaspur, India

Abstract- Fault tolerance approach is basically used for tolerate the faults. A fault tolerance approach will work even if there is a single or multiple faults in the system. For analyzing the fault Monte-Carlo Simulation is mainly used which calculate the fault index of each and every module by running the project many numbers of times e.g. 1000 times, each module are choosing randomly using probability distribution. The objective of the simulation is to find out the effect of uncertainties, faults and damages in the embedded system design. The Monte Carlo simulation technique is used in research work to provide greater flexibility in estimating fault index. The main objective of work is to analyze the fault in embedded system design by calculating the fault index of each module in the network of embedded system.

Keywords- Fault tolerance, Embedded System, Monte-Carlo Simulation, Fault Index

I. INTRODUCTION

Today in every aspect of life, Embedded systems are used but the designing of Embedded systems is not an easy task .Many unknown problems arises in embedded system design because of faults or error. So we use fault tolerance approach. Fault tolerance is the property that allows a system to continue operating properly in the event of the failure of some of its components. If its operating quality decreases at all, the decrease is proportional to the cruelty of the failure, as compared to a simply designed system in which even a small failure can cause total breakdown. Fault tolerance is particularly required after in high availability or life critical systems.

A. Fault tolerance

It can be defined as a way to tolerate the faults". In other words, it concerns all activities that are performed to reduce the uncertainties or faults associated with certain tasks. Types of fault tolerance are discussed below:

Types of Fault tolerance

- **Hardware fault tolerance**: -Hardware fault tolerance is the most common application of these systems, designed to prevent failures due to hardware components.Typically, components have multiple backups and are separated into smaller "segments" that act to contain a fault, and extra redundancy is built into all physical connectors, power supplies, fans etc.
- **Software fault tolerance**: -Software fault tolerance based more around reversing programming errors, driver failures, operator errors, bad command sequences using real time redundancy, or static "emergency" subprograms to fill in for programs that crash. There are many ways to conduct such fault regulation, depending on the application and available hardware.

B. Embedded System

An Embedded system is one that has computer-hardware with software embedded in it as one of its most important component [8]. Embedded systems are spreading out to various areas of our everyday life. Such systems are all around us in our cars, homes, workplaces, and even in our pockets [1]. Embedded system is a computer system with a dedicated function within a larger mechanical or electrical system with real-time computing constraints.Fault tolerance can tolerate functional-unit faults with minimum cost overhead. In fault tolerant computer system, programs that are considered robust are designed to continue operation despite an error, exception or invalid input instead of crashing completely.

C. Fault tolerance architecture for embedded system

Fault tolerance architectures are classified into following categories [3]:

Duplication of frequently failing units (Power supply units):

- The simplest part of fault tolerant architecture is designing a system with duplicated power supplies. This approach works in the systems where power densities are high and PSUs fail frequently because of heavy load.

- In this architecture, when one PSU fails, the other takes over. This means that, despite two PSUs being present, only one will take the full load, stressing the active PSU. This mode is also known as hot-stand-by mode.
- So we improve the architecture, modifications are made so that PSUs share the load equally and when one of them fails, the other takes 100 per cent of the load. This mode is known as load-sharing mode.

D. Designing of Embedded Systems

There are two views of embedded system design process. One is Top-down view and other is Bottom-upview. In the Top-down view, designer starts with the system requirements. In next step, specifications, more detail description is created of requirements. But the specifications states only how the system behaves, not how it is built.

In Bottom-up view, designer starts with components to build a system. . Bottom-up view is needed because designer does not have perfect understanding into how later stages of design process will turn out. Decisions at one stage of design are based upon estimates of what will happen later: how fast can designer make a particular function run? How much memory will be needed? How much system bus capacity will be needed.

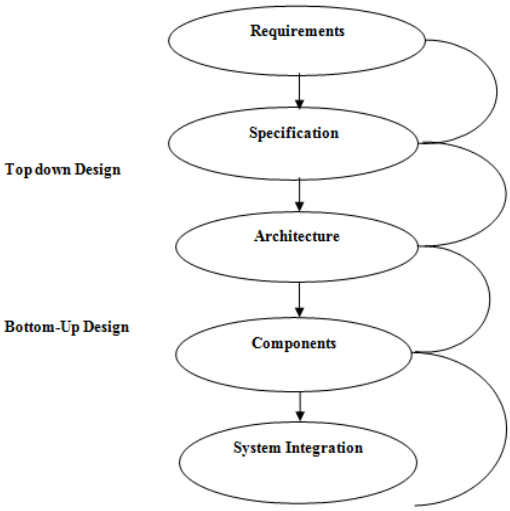


Fig. 1. Embedded system design process

II. ROLE OF SIMULATION IN EMBEDDED SYSTEM DESIGN USING FAULT TOLERANCE APPROACH

Simulation is an important tool used for analyzing the faults in embedded system design. Simulation is performing experiments with the model for the purpose of understanding the behavior of the system. It is necessary to consider only those parts of the system that affect the problem under analysis. In simulation technique, we run the module for many no. of times, with the specified input values like three weights estimates.

Monte Carlo simulation methods are mostly used in fault tolerance approach in which random numbers are generated to simulate embedded system many number of times. In Monte Carlo simulation, a random value is selected for each of the tasks, based on the weights of estimates. The model is calculated based on this random number value. The result of the model is recorded, and the process is repeated. A typical Monte Carlo simulation calculates the modules hundreds or thousands of times, each time using different randomly selected values. This technique is used to solve deterministic as well as stochastic problems.

III. PROPOSED WORK

Designing of embedded system is not an easy task because during designing it suffers many conditions moreover bad or good. When analyzing the faults which generate the weights with each module of the embedded system design generated randomly. Once the samples are generated, then they are used to calculate weights using mean and standard deviation formula.

This network contains designing of embedded system with 13 nodes and 13 modules. Node is represent N and Module is represent M. Circle is represent the node and without circle is define the module. The Dark circles represents the start and finish node.The project network of the process designing of embedded system on which the simulator is applied for analyzing the fault is given in figure 1. Analyzing the network for embedded system design is to find out the faults in form of modules and then choose the faultier module.

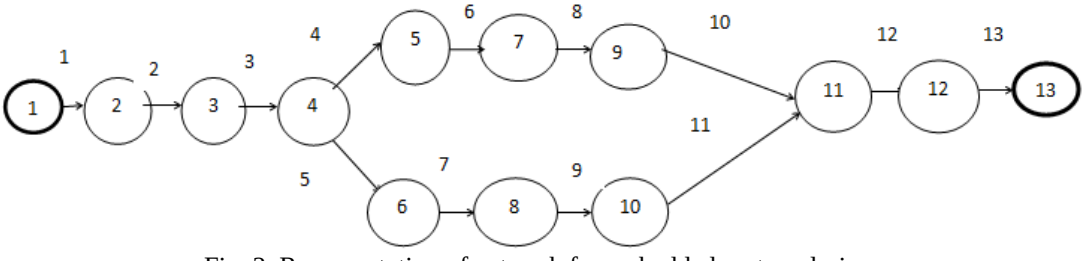


Fig. 2. Representation of network for embedded system design

In this figure circle represent the **node** of the network diagram for embedded system design which is called as **event**. Modules are represented by arrow\edges between two nodes. Module is also known as activity. **Start** and **Finish** node is represent by Dark circle.

Analyzing the network for embedded system design is to find out the faults in form of modules and then choose the faultier module and apply redundancy using fault tolerant method. The Table 1 represents the start of a module S1 [n], Finish of a module F1 [n].

Table1.Estimated weight of each module

Module No	S1[n]	F1[n]	W1(i)	W2(i)	W3(i)	MUE(μ)	SIGMA(σ)
1	1	2	1.5	5	8	4.9	1.1
2	2	3	0.2	4.5	6	4.0	1.0
3	3	4	1	8.5	4	6.5	0.5
4	4	5	0.7	9	10	7.8	1.6
5	4	6	2	8	11	7.5	1.5
6	5	7	0.5	7.5	8.5	6.5	1.3
7	6	8	3	6.5	7	6.0	0.7
8	7	9	4	7	9	6.8	0.8
9	8	10	3.5	5.5	7.5	5.5	0.7
10	9	11	4.5	9.5	10.5	8.8	1.0
11	10	11	0.8	11	12	9.5	1.9
12	11	12	0.1	9.3	10.2	7.9	1.7
13	12	13	1	3.5	9.5	4.1	1.4

Table 1 also shows the three weight estimates for each module and their corresponding mean time and standard deviations. The data given in this table is input to the simulator to compute fault index.

The whole procedure of designing a simulator for define the fault indexis described in the algorithm_1. In this algorithm N represent the total no. of node, M represent the total number of module represent these module.

Algorithm 1:

Step-1:Read the input data for embedded system design which involves list of modules M and nodes N and starting node START_NODE S1[n] and finish node FINISH_NODE F1[n] for modules i, where i = 1, 2… n. Appropriate parameters for each module MEU (μ_i), SIGMA (σ_i) are distributed according to beta distribution using the 3 weight W1(i),W2(i),W3(i) estimates where,

Mean (MEU), $\mu_i = (W1[i]+4*W2[i]+W3[i])/6$

Standard Deviation (SIGMA), $\sigma_i = ((W2[i]-W1[i])/6)^2$

Step-2: Generate weight samples, W[i] for Module m, where m = 1, 2...n using Gaussian distribution (Box Muller transformation).

$S= (-2 \log_e r1)^{1/2} * \cos(2\Pi * r2)$

$W[i] = \sigma_i * S + \mu_i$

[Where r1 and r2 are two uniform random numbers in range (0, 1)]

Step-3: Make forward pass to determine completion time of the embedded system design. Equations used during forward pass are:

a) $E_F[i] = E_S[i]+W[i]$ where i = 1,2...n

b) $E_N[j] = MAX\{E_F[$all activities terminating in j$]\}$where j = 1,2...m

c) $E_S[i] = E_N[START\_NODE[i]]$ where i = 1,2...n

Step-4: Make backward pass to determine critical activities. Equations used during backward pass are:

a) $L_S[i] = L_F[i] - W[i]$ where, i = 1,2...n

b) $L_N[j] = MIN\{L_S[$all activities originating in j$]\}$ where, j = 1,2...m

c) $L_F[$every activity terminating in node j$]=L_N[j]$ where, j = 1,2...m

Step-5: Mark the critical or faulty modules.

If $L_S[i] - E_S[i] = L_F[i] - E_F[i] <= ERROR$

Step-6: Repeat the above steps (2 to 4) for 1000 runs and compute critical index for each module i, where i = 1, 2....n.

Step-7: End.

IV. OUTPUT OF SIMULATOR

The outputs show the fault index table and graph. We read the input data for embedded system design which involves module m and node n and starting node and finish node for modules. The Mean (μ) and Standard Deviation (σ) for every module m will be distributed according to beta distribution using three weights optimistic weights [W1(i)], most likely weights [W2(i)], Pessimistic weights [W3(i)]. We apply forward pass and backward pass to generate the weights using module m for design of embedded system.

The program is executed for 1000 number of simulations so as to get the correctness in the results. Table No 5.1 shows Module No. and Fault Index.

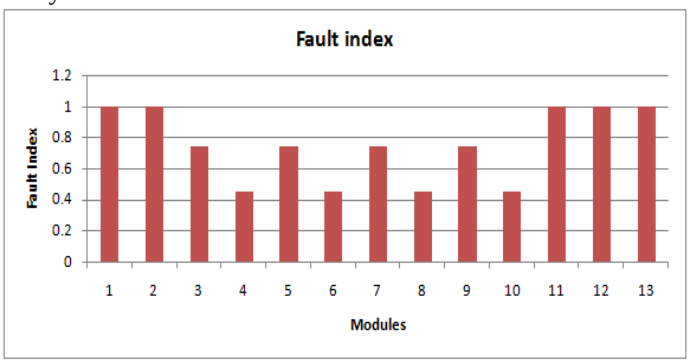
V. RESULTS OF SIMULATION

- Module no. 1, 2, 11, 12 and 13 has the fault index of 1 that means these modules are 100% faulty as given in table 5.1. These are the modules which appear on critical path during all simulation runs. These modules need to be focused very carefully.
- We can use fault tolerance approach in these faulty modules and apply duplicacy method.

Table 2. Fault Index

Module No.	Fault index
1	1
2	1
3	0.768
4	0.451
5	0.768
6	0.451
7	0.768
8	0.451
9	0.768
10	0.451
11	1
12	1
13	1

The graph 1 shows the faulty or critical modules. If module is containing high weight (fault index) that means module is faulty. We can find out the faulty or critical modules using graph. Then we determine the fault index using module in the graph. Graph show the faulty module. If fault index is contain high weight of module that means high chances of fault. For example graph contain the 5 modules are faulty (1, 2, 11, 12, 13) and remaining modules are faulty free. It means we can apply fault tolerance approach in these five modules because more chances of fault in five modules. Fault tolerance approach is containing the duplicacy method in these modules and can be tolerates the faults. From the graph one can easily identify the critical or faulty modules.



Graph 1. Fault Index bar chart

Hence the modules 1, 2, 11, 12 and 13 needs to be taken care very carefully to reduce the chances of fault occurrences in project. In these modules we can apply fault tolerance approach. Fault tolerance approach is containing the redundancy or duplicacy method.

VI. CONCLUSION

Many difficulties arise in embedded system design because of error or faults. So we use fault tolerance approach for such case. Fault tolerance is particularly required after in life critical systems. In fault tolerance approach redundancy or duplicacy method can be used to tolerate the faults

Research strongly focuses on fault tolerance approach where we find out the modules which are faulty or critical in network diagram for embedded system design. For this purpose PERT and Monte Carlo Simulation techniques are used. PERT can estimate the critical/faulty modules only once that is why we use the concept of simulation which is mainly used to run the experiment no. of times.

It is also concluded that higher weights that contain faulty module (criticality) index. Computing fault index of modules that help where fault arise in each module. The main objective of work is to analyze the fault in embedded system design by calculating the fault index of each module in the network of embedded system.

ACKNOWLEDGEMENTS

I express my gratitude to all those who have encouraged me for my thesis. I would like to thank my thesis supervisor Er. NitikaBansal, Assistant Professor, Department of Computer Science and Engineering, Ganpati Institute of Technology and Management. He has been highly available throughout the whole process of my thesis. I am very grateful to Dr. Rajesh Garg, Assistant Professor, Seth Jai Prakash polytechnic, Damla, without his help I cannot proceed.

Special thanks to Er. Narender Singh, Head of Department, Computer Science and Engineering, Ganpati Institute of Technology and Management, who always helped in my work and fulfills all essentials.

REFERENCES

[1] Mehdi Modarressi, Hani Javanhemmat(2014),"*A Fault-Tolerant Approach to Embedded- System design using software stand by sparing*", Computer Engineering Department, Sharif University of Technology, Tehran, Iran.

[2] Jerry Banks, John S. Carson,"*Discrete-Event System Simulation*",Barry L. Nelson and David M. Nicol.

[3] S.A SrinivasaMoorthy ,"*Introduction to Fault-Tolerant Embedded System*: Article",D4X Technologies Pvt Ltd,Chennai.

[4] Deo, Narsingh (2003), "*System Simulation with Digital Computer*", Prentice Hall of India, New Delhi.

[5] Sharma, S.D., "*Operations Research*", KedarNath and Ram Nath, Meerut.

[6] H.Ammar, B. Cukic, C.Fuhrman and A.Mili(1999)," *A Comparative Analysis of Hardware and Software Fault Tolerance : Impact on Software Reliability Engineering*", Institute for software research Falrmont, WV 26554 USA.

[7] D.S. Hira," *System Simulation*," S.Chand.

[8] Raj Kamal," *Embedded System (Architecture, Programming and Design).*

[9] Paul Krzyzanowski(April 2009)"*Distributed system hardware fault*".

[10] Pengcheng Huang, HoeseokYang(2014),"*On the Scheduling of Fault-Tolerant Mixed-Criticality Systems*", Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland .

