Estimation of Software Reliability on the Basis of Bits for Embedded System

Sanjay Kumar Chauhan[1] Rajesh Mishra[2] Rajendra Bahadur Singh[3]

Gautam Buddha University, Noida, Uttar Pradesh, India
sanjay_itm07@hotmail.com, contact No.- +918005246558

Abstract - As software in an embedded system has taken responsibility for controlling both software, and hardware components, the importance of estimating more accurate reliability for such software has been increased. To estimate the reliability of software, we use software which is a of collection of bits so while analyzing of system the software reliability in terms of bit it gives better result comparison to software reliability in terms of time analysis. The software is the collection of bits and during the execution bits are processed and it defines the level of execution. While we analyze the bit we justify the performance of system and estimate the reliability in a better way. However, many researchers have developed software reliability models assuming that software failures are caused by only software faults, which might lead to inaccurate reliability estimation.

In this paper a Software Reliability model is proposed in which the performance of the existing reliability model is improved.

Keywords – Software Reliability, Embedded Software Reliability, Embedded System, Bit-rate, Clock speed, TPT, BSC.

INTRODUCTION

Software is safety-critical if a failure can directly cause loss of human life or have other catastrophic consequences [1], examples include systems that control aircraft, nuclear reactors, and medical devices. Clearly the reliability and correctness of such software needs to be demonstrated with high assurance, and regulatory agencies in safety-critical industries typically require system providers to meet stringent certification requirements [2]-[4].

Software reliability is defined as the probability of the failure-free operation of a software system for a specified period of time in a specified environment [4]. For mission critical systems, redundancy techniques are commonly applied to achieve high reliability. For embedded systems consisting of software and hardware components, redundancy can be achieved by applying extra copies of these components (in parallel) to handle the system workloads [5]-[7]. Various software reliability growth models (SRGMs) exist to estimate the expected number of total defects (or failures) or the expected number of remaining defects (or failures). Some well known SRGMs are Goel model (1985), Goel and Okumoto model (1979), Kececioglu model (1991), Musa and Ackerman model (1989), Musa et al. Model (1987), Yamada et al. Models (1983, 1985, 1986) etc [8]-[18]. These models have some limitations. Each model can provide good result for a particular data set, but no model is good for all data sets [19]-[28].

In this paper we present a software reliability model which is successfully used to solve the reliability modeling problem according to embedded software size and its execution rate, that are having limited number of bits.

1. EMBEDDED SOFTWARE RELIABILITY

Software is a set of instruction that have several bits to elaborate the bits (0/1). Embedded software are basically design for a specific task for example press, air-condition, aircraft etc. that have a specific length, and can be countable in bits, or during execution we can justify each and every clock cycle for its bits involved. According to clock speed we can also define its reliability of instructions per cycle execution, if we justify the probability of failure free operation then we can easily estimate the reliability problem.

Reliability is the function of time and probability in hardware reliability calculation but software is not having the wear out time so we can say that a software may not fully depend upon time so estimation of software reliability is quite crucial in terms of time. While we consider bit rate and probability of its failure then we can easily derive the reliability in form of total probability theorem and Bernoulli Trials. So the reliability model with respect to bits can be processed data for failure free operation per clock, and estimate the failure per cycle until the whole execution completes.

1.1 SOFTWARE RELIABILITY IN PER CYCLE EXECUTION

Let a system that have ‘n’ bits that execute in one clock cycle and each bit having its own failure probability, where P_f is failure probability per bit and R is reliability. According to the reliability theory reliability is compliment of probability of failure occurred, so reliability occurrences per bit in each cycle is:

$$R(n) = 1-P_f(n) \tag{1}$$

The above expression shows the reliability per bit, where R(n) is the reliability of n^{th} bit, $P_f(n)$ is the probability of failure of n^{th} bit, n is the number of bit per cycle. In case of over-all reliability of n bits in a clock cycle, the execution fails if any bit is missing in the clock. The over-all reliability of ‘n’ bit’s is:

$$F(R_n)=R(1)UR(2)\ldots.UR(n) \tag{2}$$

The above expression shows that the over-all reliability is the union of the all individual bit reliability.

1.2 SOFTWARE RELIABILITY IN TERM OF TOTAL PROBABILITY THEOREM (TPT)

Let a system clock cycle execute 8 bits data (it can have ‘n’ bits), and all bits having its own failure probability ($B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8$), and the shaded portion shows the probability of failure for clock.
See figure 1.

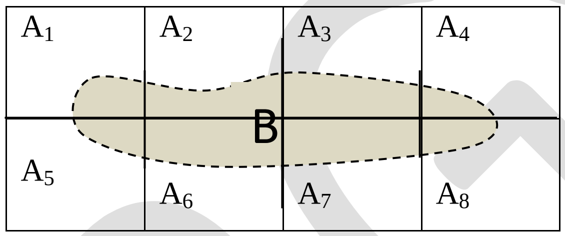


Figure 1:- A system clock having 8 bit.

The shaded portion represents ‘B’, and $B = B_1 + B_2 + B_3 + B_4 + B_5 + B_6 + B_7 + B_8$. Where $B_1, B_2, B_3, B_4, B_5, B_6, B_7, B_8$ is the failure probability of each bits in clock and A_1 to A_8 are mutually exclusive or disjoint event (that bits are process) of clock ‘C’. So $C = A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7 + A_8$.

The total probability theorem let us to compute the probability of an event occurring by enumerating all the different condition that can occur which gives the probability of failure that occurs in the cycle execution ‘P_{fc}’.

So ,

$$P_{fc} = P(B \cap C) \tag{3}$$

$$P_{fc} = P(B\cap(A_1UA_2\ldots UA_8))$$

$$P_{fc} = P((B\cap A_1)U(B\cap A_2)..U(B\cap A_8))$$

$$P_{fc} = \textstyle\sum_i^8 P(B \cap Ai)$$

$$P_{fc} = \textstyle\sum_i^8 P(B \mid Ai)\, P(Ai) \tag{4}$$

This give the failure probability occurs in clock execution.

1.3 RELIABILITY IN TERM OF BERNOULLI TRIALS

For ‘n’ bits, data are processed in per clock cycle, and having failure probability ‘P_e’ for each bits, so the probability of successes is:

$$Q = 1- P_e \tag{5}$$

Failure probability in term of Bernoulli Trials is the probability of achieving exactly k failure in *n* trials.

Let the probability for ‘k’ failure in ‘n’ bits data process, and ‘p_c’ is the failure probability of clock cycle.

$$P_c = \binom{n}{k} P_e^{\,k}\ (Q)^{n-k} \tag{6}$$

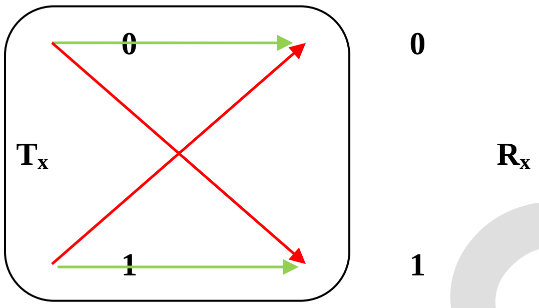
$$P_c = \binom{n}{k} P_e^{\,k}\ (1- P_e)^{n-k}$$

$$P_c = \frac{n!}{k!(n-k)!} P_e^{\,k}\ (1- P_e)^{n-k} \tag{7}$$

Bernoulli trials in term of binary symmetric channel (BSC), Lets probability for error per bit is P_e

$P(0|1) = P(1|0) = P_e$ (red) $\longrightarrow$

$P(0|0) = P(1|1) = 1 - P_e$ (green) $\longrightarrow$



Data process

Figure 2: Binary Symmetric channel

According to Bernoulli trials P_c is the define for 'n' bit clock size having 'k' errors. But if any one bit may miss, that cause failure. That minces for failure free operation 'k' have the minimum value '1'. So for 'k = 1' P_c will be.

$$P_c = \binom{n}{1} P_e^{\,1}(1- P_e)^{n-1} \tag{8}$$

For reliability Rc of a clock if there failure occurrence is P_c then according to reliability theorem.

$$R_c = 1 - P_c \tag{9}$$

So,

$$R_c = 1 - \binom{n}{1} P_e^{\,1}(1- P_e)^{n-1} \tag{10}$$

2. RELATIONSHIP BETWEEN RELIABILITY AND CLOCK SPEED

The clock cycle is the time between two adjacent pulses of the oscillator that sets the tempo of the computer processor. The number of these pulses per second is known as the clock speed, which is generally measured in Mhz (megahertz, or millions of pulses per second) and even in Ghz (gigahertz, or billions of pulses per second).

A system may provide variable response while we are varying the clock speed, this may affect the reliability of the system in different terms (ex.- In term of failure, in term of utility, in term of cost or optimization).

A clock speed may cause failure in execution of the bits. If the clock speed is higher than the band limit of the system then due to internal capacitance a system can't operate beyond the clock speed (ex. ARM TDMI 7 operation is crucial above 30MHz). If the frequency is beyond its range then due to extra attenuation (the cutoff frequency at which the current gain drops by 3 decibels (70% amplitude)) the bits can be missed and failure occur in the system, while increasing the frequency we observe that the fault rate will increases.

When a system operate at low frequency, then it can't provide optimum work due to low speed, there cost utility factor is effective which reduces the reliability as a cost effective function (ex. below 1 MHz operation is better to use other low operating frequency microcontroller then ARM TDMI 7, it increase the product manufacturing cost, ultimately it reduces the reliability in term of cost and speed).

For maximum reliability a system may operate at its desire operating range (ex. ARM TDMI 7 provide optimum response in 1 MHz to 30MHz). In the desire range current gain drops is less then 3 decibels (amplitude grater then 70%).

The above discussion gives a relative graph in between reliability and clock cycle.

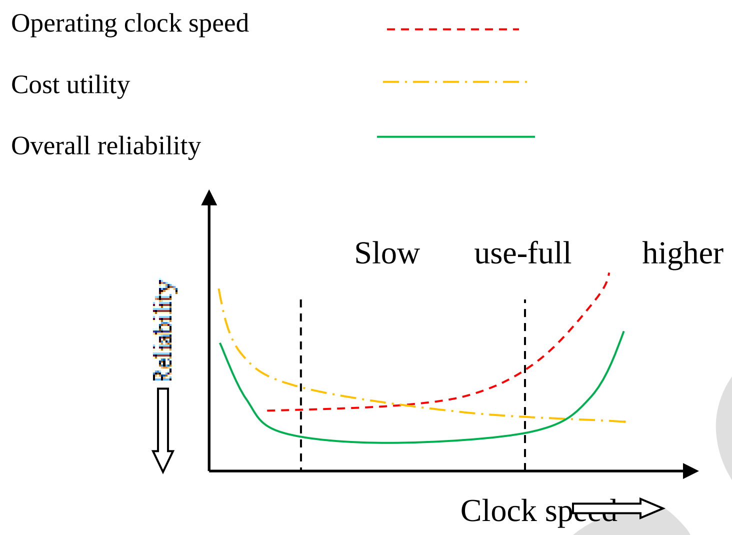


Figure 3: Reliability versus clock speed graph

ACKNOWLEDGMENT

First of all, I would like to thank **Dr. Rajesh Mishra (HOD),** and **Mr. Rajendra Bahadur Singh**, **Faculty Associate**, School of Information & Communication Technology, Gautam Buddha University for giving me support and courage to reach here. I would like to take the opportunity to express my gratitude to people who have played an important role in the completion of this paper. I wish to express my deep sense of gratitude to my supervisor for his valuable guidance and useful suggestions, which helped me in completing the work.

CONCLUSION

In this paper we present a software reliability model in terms of bits which is more efficient to estimate the software reliability. A hardware reliability is fully depend upon time so its reliability versus time graph is much suitable to demonstrate in bath tub curve while a software reliability can't because software does not have wear out time. In case of clock speed versus reliability graph, the bath tub curve achieves better reliability in comparison to the time versus failure/reliability graph. Due to fully dependency of software on bits we can easily demonstrate reliability in terms of bits and it gives a better estimation for software reliability.

REFERENCES:

[1] N. G. Leveson, *Software system safety and computers*, Addison Wesley, ISBN : 0-201-11972-2, Año de publicación, 2001.

[2] M Handbook, *Reliability Test Methods, Plans and Environments*, MIL-HDBK-781 14 July 1987 MILITARY HANDBOOK Reliability Test Methods, Plans, and Environments for Engineering Development, Qualification US Department of Defense, Washington, 1996.

[3] L. Michael, *Handbook of Software Reliability Engineering*, McGraw Hill and IEEE Society Press, 1996.

[4] Zhidong Qin,Hui Chen,Youqun Shi, *Reliability Demonstration Testing Method For Safety –Critical Embedded Application Software*, ICESS2008 IEEE.

[5] J.-C. Laprie, J. Arlat, C. Beounes, and K. Kanoun, "*Definition and analysis of hardware- and software-fault-tolerant architectures*," *IEEE Computer*, pp. 39–51, July 1990.

[6] M. R. Lyu, Ed., *Handbook of Software Reliability Engineering*: Mc-Graw-Hill, IEEE Computer Society Press, 1996, ch. 1–3, and 14.

[7] R. K. Scott, J. W. Gault, and D. F. McAllister, *"Fault-tolerant software reliability modeling,"* IEEE Trans. Software Engineering, vol. SE-13, pp. 582–592, 1987.

[8] Kazuhira Okumoto, ― *A Statistical Method for Software Quality Control‖*, IEEE Transactions On Software Engineering, Vol. Se-1L, No. 12, December 1985

[9] Chin-Yu Huang, Michael R. Lyu & Sy-Yen Kuo, ―*A Unified Scheme of Some Non-homogenous Poisson Process Models for Software Reliability Estimation*, IEEE Transactions On Software Engineering, Vol. 29, No. 3, Page 261-270, March 2003

[10] Qiuying Li and Jian Wang―*Determination of Software Reliability Demonstration Testing Effort Based on Importance Sampling and Prior Information*, Springer Berlin Heidelberg, Print ISBN 978-3-642-25907-4, Online ISBN 978-3-642-25908-1, Page 247-255, 2012.

[11] Hoang Pham, ―*An Imperfect-debugging Fault-detection Dependent-parameter Software, International Journal of Automation and Computing*, 04(4), October 2007, 325-328, DOI: 10.1007/s11633-007-0325-8

[12] Xuemei Zhang, Xiaolin Teng, & Hoang Pham, ―*Considering Fault Removal Efficiency in Software Reliability Assessment*, IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 33, No. 1, January 2003

[13] Anthony Iannin et al., ―Criteria *for Software Reliability Model Comparisons*, IEEE Transactions On Software Engineering, Vol Se-10, No. 6, Page 687-692, November 1984

[14] Chin-Yu Huang and Michael R. Lyu, ―*Estimation and Analysis of Some Generalized Multiple Change-Point Software Reliability Models*, IEEE Transactions On Reliability, Vol. 60, No. 2, Page 498-515, June 2011

[15] Katerina Goˇseva-Popstojanova and Kishor S. Trivedi, ―*Failure Correlation in Software Reliability Models*, IEEE Transactions On Reliability, Vol. 49, No. 1, Page 37-49, March 2000

[16] Shinji Inoue and Shigeru Yamada, ―*Generalized Discrete Software Reliability Modeling With Effect of Program Size*, IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 37, No. 2, Page 170-180, March 2007

[17] Kazuya Shibata, Koichiro Rinsaka and Tadashi Dohi, ―Metrics-*Based Software Reliability Models Using Non-homogeneous Poisson Processes*, 17th International Symposium on Software Reliability Engineering (ISSRE'06)

[18] Hoang Pham and Xuemei Zhang, ―*NHPP software reliability and cost models with testing coverage*, European Journal of Operational Research 145 (2003) 443–454

[19] Michael R. Lyu, Sampath Rangarajan and Aad P. A. van Moorsel, ―*Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development*, IEEE Transactions On Reliability, Vol. 51, No. 2, Page 183-193, June 2002

[20] K.B.P.L.M. Kelani Bandara et al., ―*Optimal Selection of Failure Data for Reliability Estimation Based on a Standard Deviation Method*, Second International Conference on Industrial and Information Systems, ICIIS 2007, 8 – 11 August 2007, Sri Lanka

[21] RajPal Garg, Kapil Sharma, Rajive Kumar, and R. K. Garg, ―*Performance Analysis of Software Reliability Models using Matrix Method*, World Academy of Science, Engineering and Technology July, 2010

[22] Sultan Aljahdali and Alaa F. Sheta, ―*Predicting the Reliability of Software Systems Using Fuzzy Logic*, 2011 Eighth International Conference on Information Technology: New Generations

[23] [23] Ralf H. Reussner, Heinz W. Schmidt and Iman H. Poernomo, ―*Reliability prediction for component-based software architectures*, The Journal of Systems and Software 66 (2003) 241–252

[24] Kapil Sharma,Rakesh Garg, C. K. Nagpal, and R. K. Garg, ―*Selection of Optimal Software Reliability Growth Models Using a Distance Based Approach*, IEEE Transactions On Reliability, Vol. 59, No. 2, Page 266-277, June 2010

[25] Norman F. Schneidewind, ―*Software Reliability Model with Optimal Selection of Failure Data*, IEEE Transactions On Software Engineering, Vol. 19, No. 11, Page 1095-1105, November 1993

[26] Dr. Ajay Gupta, Dr. Digvijay Choudhary and Dr. Suneet Saxena, ―*Software Reliability Estimation using Yamada Delayed S Shaped Model under Imperfect Debugging and Time Lag*, International Journal of Computer Applications (0975-8887) Volume 23– No.7, June 2011

[27] Shaik.Mohammad Rafi et al., ―*Software Reliability Growth Model with Logistic-Exponential Test-Effort Function and Analysis of Software Release Policy*, (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 387-399

[28] Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad , *Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value*, MICS I.J. Information Technology and Computer Science, 2013, 02, 1-14