An Approach to Establish Tool Based Formal Method to Verify the Critical System Design: A Case Study

Preeti B Tadakal, Manju Nanda, J. Jayanthi

Abstract - **Embedded systems in recent critical applications are becoming more complicated with increasing capacities and running more complex functions. Powerful methods are required to verify a complex system design used in any safety critical applications (like Avionics, nuclear and railways systems).This paper proposes to use model checker, a formal method based technique to verify the design of an indigenously developed safety critical system, enhanced Fatigue Meter. The outcome of this analysis is verified with the conventional analysis (Manual analysis) results to prove the effectiveness of formal method based approach to analyze the system behavior.**

Keywords - Behavioral model, Formal methods, Safety Critical Systems, Simulink design verifier.

I. INTRODUCTION

Critical systems are systems that, if they fail or malfunction, may threaten human lives and generate financial losses. The critical systems can either be life-critical systems (also called safety-critical systems) or else mission-critical systems (also called business-critical systems). Risks are increased for systems having requirements for long lifetime, deployed in large numbers, intensively used by several people, and/or complicated or even impractical to repair while operating. Typical examples are unmanned space ships, satellites, banking applications, security systems, etc [9]. Designing a robust hardware system for such applications is a tough task. The hardware design can be thought of as one big state machine. Whereas a state machine has a large number of states, the length of the paths through the state machine is unbounded [3]. Various real-world systems and processes need complex timing, reliability and high performance condition which are complex and time consuming to verify with conventional testing approaches.

Formal methods are mathematical techniques [10], for developing software and hardware systems, frequently supported by tools. Formal methods offer a set of notations that are used to construct mathematical models of systems and techniques for automatic verification of such models. Over the years, formal methods have been extensively and successfully applied in a wide range of problems and in practical applications in both academia and industry for the specification and analysis of many different systems [4].

Manuscript received June, 2015.

Preeti B Tadakal, *M.Tech, Digital Electronics and Communication Systems, VTU CPGS, Bangalore, Bangalore, India.*

Manju Nanda, *Principal Scientist, Aerospace Electronics and Systems Division, CSIR – National Aerospace Lab, Bangalore, India.*

J. Jayanthi, *Sr. Principal Scientist, Aerospace Electronics and Systems Division, CSIR – National Aerospace Lab, Bangalore, India.*

The main goal of this work is to evaluate the applicability of formal methods for critical system design analysis. Here we consider eFM (Enhanced fatigue meter) as a case study to demonstrate the efficient use of formal methods based verification approaches for the behavioral analysis of the system design.

An eFM is designed with a scope to utilize it as part of structural health monitoring and as well Integrated Vehicle Health Monitoring system of aircraft using the advanced technology. It is to be developed with a design meeting the airworthiness necessities for fighter and/or transport aircraft. The design and analysis of embedded systems and monitoring equipment need detailed knowledge of their real-time aspects, in addition to the functional requirements [14].

The main objective of this paper is to describe the use of formal techniques for the effective verification of critical system design. The paper is divided into various sections. Section II describes about the formal method and conventional simulation techniques used in the system design analysis. The uniqueness of this paper is how the Simulink model of any system is used for formal verification is described in the section III. The section IV & V gives the detail information about the modeling methodology, verification techniques and the analysis of the results.

II. RELATED WORK

A. Formal methods

Formal methods are mathematics based languages, techniques and tools that are applied at any part of the system design life-cycle [6] [10]. These mathematics based methods as well make use of enhancement techniques at any stage to guarantee the precision, completeness and regularity of specification. By providing specific and definite description mechanisms, formal methods make possible the development of the critical systems. The representation used in formal methods is called a formal specification language. The formal specification languages are based on set theory and first order predicate calculus. The languages have a formal semantics which are used to convey specifications in a clear and unambiguous manner. Formal specification languages are used to develop the model of the most complex systems by means of comparatively simple mathematical elements, such as sets, relations and functions.

Formal methods decrease the amount of errors in the final product and are a cost-effective mode of developing high integrity system. These are a particular sort of mathematically based techniques used for the specification, implementation, development and testing of software and hardware systems [10]. They profit in accuracy and testability of the software

ISSN: 2278 – 7798

[6]. In current days automated verification tools have been successful in finding bugs in "real-world" hardware protocols and device drivers [5], tools such as Stateflow/ Simulink are commonly used in automotive and avionics industry for modeling.

Model-based design offers a promising approach for detecting and correcting errors in early stages of system design [7] [8]. In this methodology, a designer first develops a model, with mathematically accurate semantics, of the system under design, and performs extensive analysis with respect to correctness requirements prior to generating the implementation from the model. Formal Methods is a very dynamic research area with a wide range of methods and mathematical models. In present circumstances, there is no existing method that fulfills all the security related needs of building a safe formal specification. Researchers and practitioners are constantly working in this area and thus gaining the benefits of using formal methods [2].

B. Simulation

Simulink is a graphical extension to MATLAB for modeling and simulation of systems. Simulink is a block diagram environment for multi-domain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. In Simulink, systems are drawn on screen as block diagrams. Many elements of block diagrams are available, such as transfer functions, summing junctions, etc., as well as virtual input and output devices [9]. Simulation plays a significant role in system validation by enabling to verify the system against its requirements at each stage of the design flow. Simulation also permits to validate environment assumptions by running experiments to compare models and reality. Simulation techniques ease debugging. For instance, it is often simpler to debug a circuit using a simulator rather than testing the actual silicon, whose internal state and implementation details may remain hidden. Simulation is also a privileged way of design space exploration, as it allows to quickly investigate the consequences of potential changes brought to the system under design. Simulation is scientifically well-understood, implemented in numerous industrial tools, and relatively easy to use, and most system engineers being previously familiar with this technology. However, simulation is not free from drawbacks and limitations, some of which are general to all forms of conventional dynamic analyses [15].

III. UNIQUENESS OF THE WORK

In this work we verify the system design using formal methods for behavioural analysis of the eFM using Simulink, Simulink contains the Simulink blocks, state flow graphs, and truth tables to represent all the process in the system. The model represents the detail information about the system design that contains different states, the transition paths between the states and their behaviour. eFM system model is verified using the Simulink design verifier which makes use of the formal verification technique [11].

The Simulink design verifier tool makes analysis easy by automatically considering the mathematical relations between the states and the transition paths between the states. The system design is validated against the requirements early in the phase without having to generate code [11].

IV. APPROACH

A. enhanced Fatigue Meter (eFM)

The eFM is used for measuring the acceleration due to gravity 'g' values experienced by the aircraft. It receives the sensor inputs from a MEMS based accelerometer and computes the 'g' crossings based on the various algorithms. The 'g' value detection in the range of -10.0 g to +10.0g with a resolution of 0.05g. It provides the interface with built-in (internal) or external accelerometer as desired by the aircraft application. The system provides interfaces for keyboard, VFD (Vacuum Fluorescent Display) and serial communications. The Graphical User Interface software that runs on the host machine communicates with the enhanced Fatigue Meter through USB uploads/downloads the configuration parameters and the recorded data. The eFM system is designed so that it can be mounted either at CG or within a radius of 1.6m from CG. Configuration of the aircraft type, aircraft name, no of GLOCK, GUNLOCK and sortie for a particular aircraft is entered using the keyboard [12][13] [14]. The top level block diagram of the eFM system is shown in the "Fig.1".

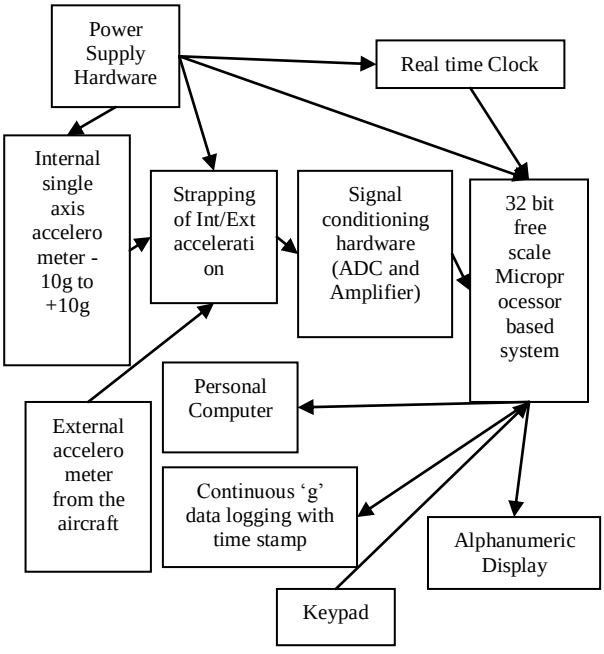

Figure1. eFM System top level block diagram.

ISSN: 2278 – 7798

B. Modeling

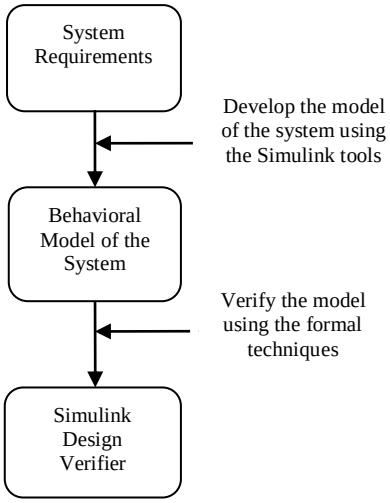

Figure2. The Workflow for Modelling and formal verification of a system.

The requirements of system under consideration, eFM are used for modeling. The blocks from the Simulink tool box are used to develop the design where the final model represents the behavioral model of the design. The design process is often viewed as a sequence of steps that transforms a set of specifications described informally into detailed specifications that are used for manufacturing. All the intermediate steps are characterized by a transformation from a more abstract description to a more detailed one. [1].

The functional specifications of eFM are given as a set of explicit or implicit relations which comprises of inputs, outputs and possibly internal (state) information like 'g' value measurement by the accelerometer, algorithm for the 'g' crossing, number of sorties completed etc[2]. A set of properties that the design must satisfy are safety properties (e.g., deadlock, emission of undesired outputs, etc.) and liveliness properties (e.g., expected response to an input, etc.), which states that no matter what ever maybe the inputs the system does not go into a specific undesirable configuration [1]. The work flow of the system modeling and formal verification of that model against the system requirements is given in "Fig.2". According to the steps specified in the work flow we will model the eFM system and verify it.

The eFM system is validated to analyze its mode detection, voltage to 'g' conversion, 'g' reading function, analog to digital conversion, and 'g' crossing behaviors. Behavioral analysis plays important role in any system design because if any one function of the system fails it leads to the total failure of the system which cannot be entertained in any of the safety critical applications. The system requirements are first used to develop the behavioral model that model is used to check the correctness of the system design using the formal technique based Simulink design verifier tool.

V. Modeling Analysis

The system functional logic is modeled in the Simulink as shown in the below "Fig.3". The model consists of mode detection block, ground mode and air mode operational blocks. The sub blocks in air mode are accelerometer input blocks, voltage to 'g' value conversion block, critical 'g' lock and unlock values, 'g' crossing algorithm and output display blocks. The eFM operates in two modes of operation that is air mode and ground mode. The Simulink model of the eFM system includes all the mode of operation and the conditional logic for the transition between the modes. The system is in the ground mode if the weight on wheels of the flight experience the weight on them i.e. WOW=1 else they will be in the air mode i.e. WOW=0. Whenever the flight is on the ground the system displays the information about the previous flight. Similarly when the flight is in the air mode it measures the acceleration due to gravity 'g' with the help of the MEMS accelerometer.

The output of the accelerometer is in the form of the voltage which is converted in the respective 'g' value with the help of the equation given below [12] [13] [14].The corresponding voltage for any 'g' value is given as V,

$$V = (((x+10)*0.13) + 0.33) \quad (1)$$

Here 'x' is the acceleration due to gravity 'g' value [13].

Unlike traditional testing methods in which test scenarios and expected results are expressed with concrete data values, formal analysis techniques let you work with models of system behavior instead of concrete data values [11].

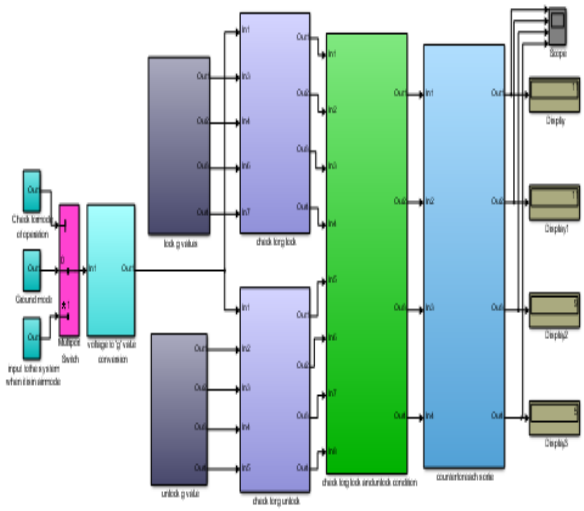

Figure3. Behavioural model of the eFM developed in the Simulink.

The model of eFM is verified to detect the errors in its design. The blocks in the model with errors like dead logic,

ISSN: 2278 – 7798

integer overflow and division by zero are detected and being highlighted and the blocks without the errors are verified and proven to be correct. For each block that contains an error, the signal range boundaries are calculated and test vector are generated that produce the error in simulation. The signal range boundaries are taken from the system requirements. The eFM behavioral model includes models of test scenarios and verification objectives that describe desired and undesired system behaviors as per the project requirements. Formal analysis performed with such models complements design simulation and provides a deeper understanding of the design [11] for the designer and proof of correct system design for the verification & certification group.

The design error-detection analysis is automated, and it does not require any additional user input and produce the analysis results in the form of detailed report which is in the HTML form as shown in the "Fig.4". The report details the dynamic execution scenarios under various conditions. This information provides critical feedback for design improvement against the project requirements [11].

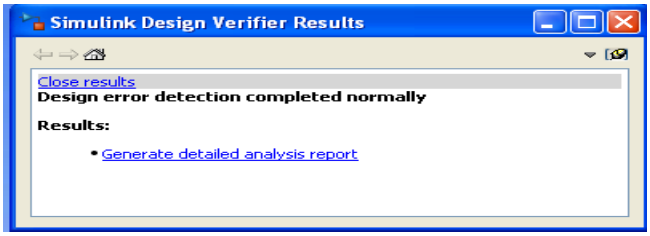

Figure4. The analysis report for the design error detection in system by SDV.

The report provides information like the model name, mode of operation, analysis status, and analysis time. Permissible ranges for all signals on all blocks are provided to in order uncover the root cause problems in the design. In the model, blocks are marked as green, yellow, or red. Green blocks have been proven unable to cause integer overflow or division by zero. Yellow blocks occur when the analysis fails to produce conclusive results or when the time limit for the analysis was exceeded.

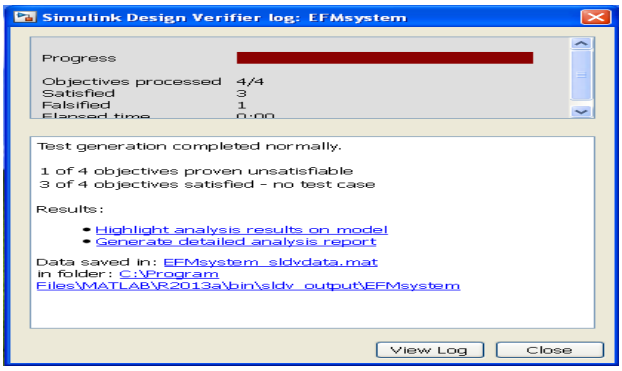

Figure5. The analysis report for the test case generation for the system by SDV.

When an error is found in the model execution path, all subsequent blocks in that path exhibit integer overflow and division by zero are marked yellow. Red blocks have design errors. For red blocks, SDV generates test cases that reproduce the problem during simulation or testing. The report for the test cases generated for the error blocks in the system is as shown in the "Fig.5". This report gives the description about the test objectives that are satisfied and unsatisfied and also the information about the model items present in the system. It also develops test cases for the unsatisfied objectives. [11]. The system is verified against the requirement properties to make sure that all the properties are proven to be correct. The result for eFM design is shown in the "Fig.6".

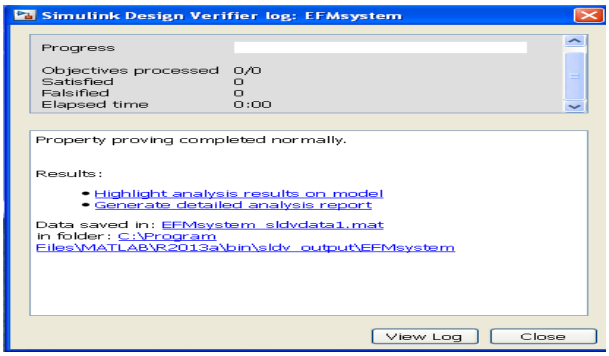

Figure6. The analysis report for the proving the properties of system by SDV

The SDV is the tool that gives complete analysis report for all the tasks performed by it on the eFM system. The behavioral model of the eFM system is free of errors like integer overflow, divide by zero and dead lock.

VI. Results and Discussion

The eFM system design behavior analysis performed by formal and manual technique. The analysis results obtained by the formal verification tool are compared with the manual results to establish the effectiveness of the formal analysis approach. The SDV results of eFM system design were almost similar to the expected results. The brief comparison between results obtained by this two methods is given in the below "TABLE I".

The mode detection function detects & enters into one of the two Modes of aircraft i.e. On_Ground or In_Air. Mode of the Aircraft is detected based on the weight on wheels input (WOW), which is an external input to the eFM unit. If WOW is 1(high), then the system enters into Ground mode and if WOW is 0 (low), then it enters Air mode. The sortie is incremented whenever the aircraft transits from Ground-to-Air and then Air-to-Ground [12] [13]. Out of the nine test cases that we applied only seven test cases passed and two failed in manual verification and formal verified model passed all the test cases. The reason for the failure in the manual approach is that it was difficult to generate the continuous transition

ISSN: 2278 – 7798

scenario. The graph shown in the "Fig.7" gives the comparison between the numbers of passed and failed test cases between the manual verified model and the formal verified model.

TABLE I. Comparison between formal methods (SDV) and manual verification methods.

Tests performed on eFM system	Formal methods (SDV)		Manual verification methods	
	Pass (%)	Fail (%)	Pass (%)	Fail (%)
Mode detection	99.9%	0.1%	77.78%	22.22%
'g' crossing detection	99.9%	0.1%	60%	40%
Peak and trough detection	Almost similar to the expected results		Different from the expected results	

The G read function reads the ADC output and converts the voltage into equivalent g value. The continuous varying voltages will have peaks and troughs which need to be recorded for tracing the flight profile [12] [13] [14].

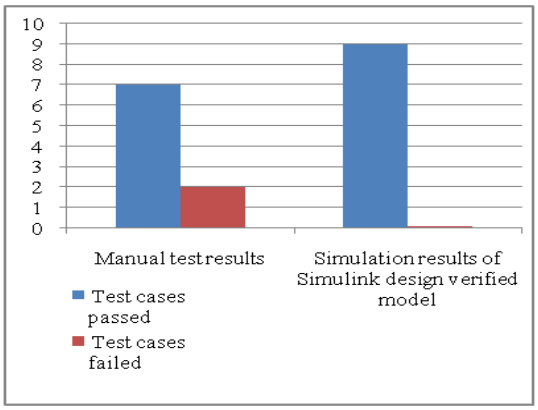

Figure7. Comparison between the numbers of passed and failed test cases between the manual verified model and the formal verified model for the mode detection.

The peaks and troughs are detected based on the g values. The graph in the below "Fig.8" gives the peak and trough results comparison between the expected output, manual results and the results from the formal verified model.

The 'g' crossing function detects g crossings for the configured aircraft. After the above G-read function reads ADC output and converts it to 'g' value, the G crossings function checks the 'g' value with the list of Lock and Unlock values. For any g crossing, if both lock & unlock bits are set , it means the g is crossed once and that particular g crossings count is incremented by one[12][13]. The below graph show in "Fig.9" gives the comparison between the 'g' crossing readings recorded by the eFM system, here we see that the manual verification pass only six test cases out of ten test cases whereas the results of the formally verified model is almost hundred percent pass. The reason for the failure in the manual approach was the unlock bit was set high before its corresponding lock bit is set.

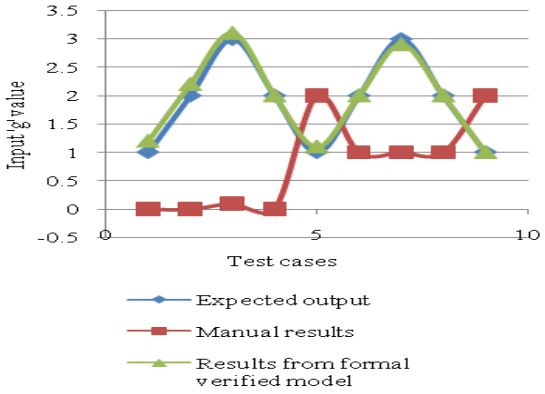

Figure8. The peak and trough results comparison between the expected output, manual results and the results from the formal verified model.

By seeing at the results obtained from the eFM systems verified by both the techniques(i.e manual verification and the formally verified model) we say that formal method based verification techniques for hardware design are more efficient than manual methods.

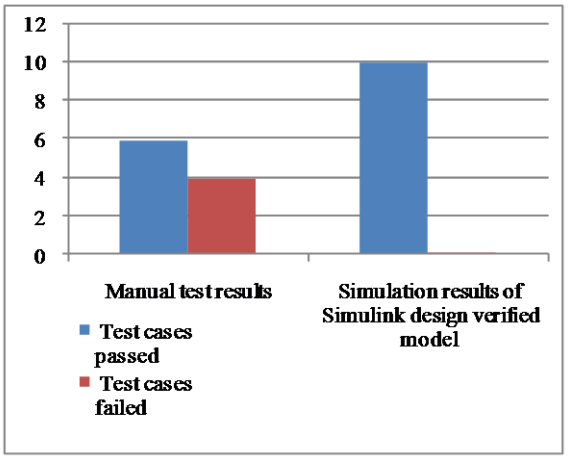

Figure9. Comparison between the numbers of passed and failed test cases between the manual verified model and the formal verified model for the 'g' crossing detection.

The design verification of the system at the earlier stage of the design life cycle reduces the errors in the final model of the system sent for the implementation and increases the confidence in the design. The errors found in design at the later stages leads to increase in the cost and also effects the time to market of that product [11]. The formal method based

ISSN: 2278 – 7798

verification techniques are efficient and give us the detailed report of the design and simulate dynamic scenarios as compared to the manual approach.

VII. CONCLUSION

In this paper propose formal method based verification techniques at the design level. The verification of eFM system and the output show the efficiency of formal based approach as compared to manual verification techniques performed at implementation level. The formal analysis of the safety critical system using the Simulink design verifier does not require any specification language. For future work we are trying to combine two formal method based tool in order to increase the verification features and take more complex safety system as a case study and verify the design.

ACKNOWLEDGEMENT

The authors would like to thank the Director of CSIR-NAL, Bangalore for supporting this work. I, Preeti B Tadakal would like to thank my mentor Dr. Manju Nanda for her guidance and support and all the other members of the ALD and KTMD of CSIR-NAL, Bangalore. I would also like to give my sincere thanks to my parents and Almighty for always being with me and encouraging me in all my work.

REFERENCES

[1] Stephen Edwards, Luciano Lavagno, Edward A. Lee, and Alberto Sangiovanni-Vincentelli, "Design of embedded systems: formal models, validation, and synthesis", Proceedings of the IEEE, vol. 85, no. 3, March 1997, pp. 366–390.

[2] Mona Batra, Amit Malik, Dr. Meenu Dave, "Formal Methods: Benefits, Challenges and Future Direction", Journal of Global Research in Computer Science, 4 (5), May 2013, 21-25.

[3] **"**Using formal methods to verify complex designs", IBM Haifa Research Lab. The IBM center of competence for Formal verification. This is available at website: www.haifa.il.ibm.com/projects/verification/RB_homepage.

[4] Milica Barjaktarovic, "The state-of-the-art in formal methods", WetStone Technologies, Inc. January 1998. This is available at website: www.cs.utexas.edu/users/csed/formal_methods/docs/stateFM.pdf.

[5] E. M. Clarke, E. A. Emerson, and J. Sifakis, "Model checking: algorithmic verification and debugging". Commun, ACM, 52(11):74–84, 2009.

[6] S.K.Pandey, Mona Batra. "Formal methods in requirements phase of SDLC". International Journal of Computer Applications (0975 – 8887), Volume 70– No.13, May 2013.

[7] T. Henzinger and J. Sifakis, "The embedded systems design challenge". In FM 2006: 14th International Symposium on Formal Methods, LNCS 4085, pages 1–15, Year. 2006.

[8] A. Sangiovanni-Vincentelli. Quo Vadis, SLD: "Reasoning about trends and challenges of system-level design". Proceedings of the IEEE, 95(3):467–506, 2007.

[9] Dr. Hubert Garavel, Dr. Susanne Graf, "Formal methods for safe and secure computers systems". Federal Office for Information Security 2013. This is available at website: www.bsi.bund.de/…/formal_methods…/formal_methods_study_875.pdf.

[10] Formal methods, specification languages, Formal verification and Formal tools. This is available at Website: www.en.wikipedia.org/w/index.php.

[11] Product Details, Examples, and System Requirements www.mathworks.com/products/sldesignverifier.

[12] Sushma Reddy. "Hardware Design document For Enhanced Fatigue Meter", CSIR – National Aerospace Lab, Bangalore, pages 8-33, July 2012

[13] Srikanth, Lakshmi P, Vinayak Garad, "Software Requirements Data For Enhanced Fatigue Meter", CSIR – National Aerospace Lab, Bangalore, pages 11–25, July 2012.

[14] **"**Enhanced Fatigue Meter (eFM)", CSIR – National Aerospace Lab, Bangalore, December 2010. This is available at Website: www.nal.res.in/pdf/RFQ_eFM_final_21dec.pdf.

[15] MATLAB, Simulation, Simulink Tool Details. This is available at Website: www.mathworks.com

AUTHORS

Preeti B Tadakal completed her graduation in Electronics and Communication Engineering in 2013 from AIET, Gulbarga, Visvesvaraya Technological University, Belgaum, Karnataka, India. She is currently perusing the Masters degree in Digital Electronics and Communication systems from Visvesvaraya Technological University, Centre for Post Graduation Studies, Bangalore (VIAT Muddenahalli), India. Her project based on "Formal Methods" is carried out at CSIR-NAL, Bangalore under the supervision of Manju Nanda, Principal Scientist, ALD, CSIR-NAL, Bangalore.

Manju Nanda she was working as a scientist for 5 years in CSIO Chandigarh and she has designed and developed the Bio-Medical systems and currently she is a Principal Scientist at ALD, CSIR-National Aerospace Laboratories, Bangalore. She has more than 15 years of experience in Avionics, Aerospace, Aircrafts and Safety Critical Embedded Systems and Software. She was involved in Design, Development and Qualification of safety critical embedded systems for medical and civil aerospace applications. She has also worked in

ISSN: 2278 – 7798

realization and implementation of formal methods for critical embedded systems from concept to implementation phase. She is currently working on application of formal methods to safety critical systems with a focus on civil aerospace applications and carrying out advanced research in software engineering.

J. Jayanthi She has been working in the field of real time embedded system for the past 25 years. She has led and executed the project for software design, development and certification of Stall Warning/Aircraft Interface Computer system for SARAS Aircraft as per DO178B Level A for the first time in the country successfully and now the system is flying as part of Avionics in the aircraft (Formal Flight trials are still due).She has designed and developed a microcontroller based Smart Fatigue Meter and enhanced Smart Fatigue Meter for Jaguar aircraft which will undergo flight trials shortly. She has led successfully Independent Verification and Validation for complex safety critical software such as Automatic Flight Control System and Engine Indication and Crew Alert System. She is currently working on application of formal methods to safety critical systems with a focus on civil aerospace applications and carrying out advanced research in software engineering and software health management, reliability and safety assessment.

ISSN: 2278 – 7798