Research Trends in Formal Verification Process for Analog and Mixed Signal Design

Vidhya D.S
Research Scholar.
Assam Don Basco Universiy
Guwahati India

R. Manjunath, Ph.D.
Wipro Technologies
Bangalore, India

ABSTRACT

Formal verification is one of the crucial stages of design phase that ensure the preciseness of the circuit design and its corresponding behavior with respect to specific system design. From last decade, there has been an abundant formal verification techniques introduced by the research community for analog and mixed signal circuits that used as an interface between analog and digital components. Owing to the maximized sophistication, and shrinking sizes of chip, analog, and mixed signal verification is encountering challenges in increasing verification requirement that calls for analyzing the prior techniques. The prime purpose of this paper is to discuss the most standard models, and techniques introduced till date and to excavate the various facts about their effectiveness in the area circuit design principles. The paper also discusses some of the critical research gaps explored from the study.

Keywords

Analog and Mixed Signal Circuits, Formal Verification, Mixed Signals

1. INTRODUCTION

In recent past huge technical advancement in Integration, Technologies has enabled to produce small sized and lesser weighted integrated circuit or chip, which contains millions of transistors. Integration technologies are broadly classified into three categories namely i) small-scale integration (SSI), ii) medium-scale integration (MSI) and iii) large-scale integration (VLSI) [1]. In VLSI, there is a lot of manufacturing defects, due to stuffing of transistors in one IC, and this is tested by giving input signals from the generator and evaluates responses using a logic analyzer. For manufacturing, an IC requires following three steps i) Design, ii) Verification and iii) Testing. In VLSI, ICs are classified into analog, digital, and mixed signal components depend on their function. Especially VLSI ICs paves for implementing Systems-of-Systems (SoS), which makes design problem enormous and complex [2]. The complexities lie at design stage, and most of the further processes in VLSI ICs making are automated and handled by advanced CAD tools. It is a non-trivial task to verify the accuracy and correctness of the design for multiple scenarios

Because of extremely large dimension in the design space. Therefore, a technique has been developed called formal verification, where design meets all the specification of inputs without meticulous input-output combinations. With the advancement of the technology for design principles pertaining to integrated circuits as well as semiconductors, System-on-Chip has evolved. However, embedded system design calls for using both analogs as well as mixed signals to be used in System-on-Chip design. Fig.1 exhibits the integrated circuits that are required in the interfaces for the real-time design process.

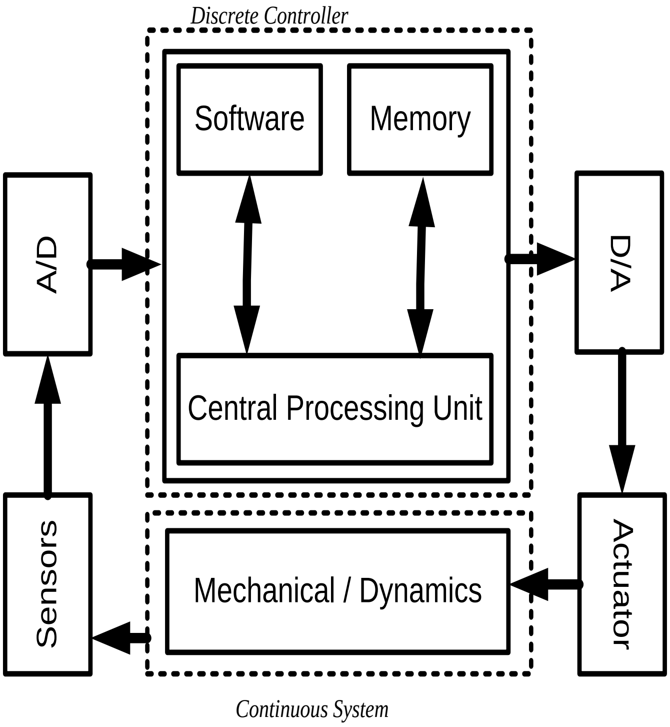

Fig 1: Design Principle of Embedded system

An efficient design principle of analog and mixed signal design are using analog signals as the front end interface while digital signals on the back end interfaces. Apart from this, some significant operation is also included in the design principle as synthesizing frequency, conversion of analog as well as digital signal, and generation of time references. It is also known that analog circuits are deployed for the biasing purpose which is highly required for appropriate, precise, and stable functionalities of the embedded system design. In this direction, computer-aided design tools like CAD are normally used to mitigate the design challenges of such circuits.

The design principle of embedded system also calls for enhancing the quality of the sophisticated integrated system with consolidated constraints of the system maximizing the productivity. Hence, CAD tools highly assist in proving the design concepts, analyze the system behavior, and study the loopholes of integrated circuits for assisting the designer to opt for the optimal strategies of design [3]. However, CAD tool is now a primitive concept of verifying the system design of mixed signals, where in more standard manner, formal verification surfaces up. In the last decade, formal verification has been studied so far, and various implemented techniques, as well as standard models, have been evolved. However, this area of research is witnessed by very irregular trend of evolution of techniques to improvise the design verification process. Hence, this paper cumulatively discusses the standard models that has been adopted for the purpose of formal verification process and attempts to check the effectiveness of the standard design verification process. The present paper also reviews some of the significant research contribution to illustrate the current scenario, as well as trend of formal verification techniques used till date.

Section II discusses the issues of the verification process followed by Section III that discusses the standard models for signal verification. Section IV discusses certain prominent studies followed by Research Gap in this field discussed briefly in Section V. Finally; Section VI makes some concluding remark of the review of this paper.

2. ISSUES OF FORMAL VERIFICATION

The issues of the verification process surface throughout the various stages of the design phases [4][5]. Unlike the process involved in the design of the digital system, there are wide varieties of differences in the characteristics as well as the requirements that exist among the various classes of the analog-mixed signal design process. For an example, the various challenges among the verification classes in the execution stages of the analog circuits are the functionalities represented directly in terms of continuous electrical quantities. However, such factors are highly sensitive to the various extrinsic factors like temperature and noise properties of the signal along with higher order physical effects like different parasitic and leakage of current. Therefore, as a result, the physical implementation is highly subjected to the frequent abstraction, analyzing, and involves an iterative process of modification of system design.

Moreover, the issues associated with the design verification process ranging from precise operation of the integrated design to ensure the system specification parameters like energy consumption on multiple hardware components, area of chip design are highly required to be considered and has to be mitigated precisely in order to deliver the precise and error-free designs. For an example, a clock jitter is the functional verification issues of majority of the analog and mixed signal design where PLL (Phase-Locked Loop) designs are usually adopted. Jitter can be represented as the deviation in a system clock output transition from its normal position. The phenomenon of clock jitter effect is highly apparent where the conversions between analog, as well as digital signals, are carried out at maximized resolution and enhanced data rates.

Another issue in the formal verification method at the functional level is the stability requirement of Δ-∑ modulators which is considered to be the best solution if the integrate output remains bounded under a constraint input signals [6][7]. It was also seen that gain parameters also influences the stability factor, and hence careful and precise selection of such parameters are highly mandatory in the formal verification process. The entire design flow of the system calls for mitigating the issues of verification system. For a uniform and persistent flow of design, a compliance certificate approving the correspondence between design levels is highly required to ensure the preciseness of the final product and its adherence to the given system specification. Fig.2 exhibits the flow of the design methodology, where it can be seen that process starts with the design of the individual blocks that are evaluated individually and then integrated to formulate a final system. However, the verification process can also be expensive in nature of the cumulative system is exhibited at the transistor level.

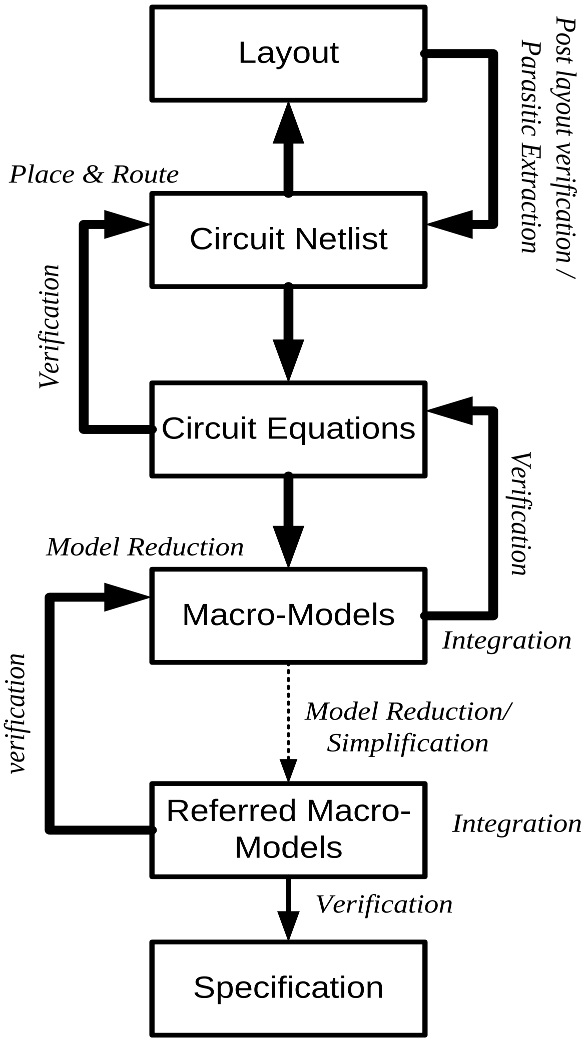

Fig 2: Flow of design methodology

Solution to above-mentioned problems can be performing the integration at a higher level than in the implementation level so that the analysis of the cumulative design can be applied. Hence, a sophisticated, cost-effective, and error-free verification process is required that can mitigate the frequent problems and potential loopholes in system design of mixed signals. The next section, therefore, discusses certain standard models of the verification process, which were introduced in the past decade and still considered to be adopted by various researchers.

3. STANDARD MODELS FOR VERIFICATION OF SIGNALS

This section discusses some of the standard formal models that were frequently consulted by various researchers in the past for the verification process. The reviews of the literature have excavated these standard models as being prominently used for the study of enhanced verification process of mixed signal designs.

3.1 Model Checking

Model checking is considered as one of the potential verification technique that performs involuntary verification process for the dynamic system properties. Model checking was preliminarily developed for discrete finite-state system and was found to be successfully used for verifying and validating the hardware circuits and communication protocols. The mechanism of model check allows the exploration of the complete state space for evaluating if the system meets the genuine specification. The design principle of algorithms in model checking is supported by mathematical approach for mapping the system to ensure if the system caters up the specification that is given as temporal-logic approach.

3.1.1 Related Work of Model Checking

Cimatti et al. [8] have considered a diagnosis system connected to a feedback control loop between a plant and its controller. The inputs of the plant are the commands issued by the controller; its outputs are measurements returned back to the controller. The role of the diagnosis system is to observe the inputs and the outputs to the plant and to report a state estimation, tracking the evolution of the unobservable state of the plant. The authors have proposed a novel approach to the verification of diagnosability, with emphasis on its practical applicability. The work is based on a new conceptualization of the problem, with the twin plant construction and the use of temporal logic formulae to describe the context of a diagnosability problem. This is one unique approach to diagnosability that enables the direct exploitation of symbolic model checking technology. Renesse and Aghvami [9] have worked on the similar direction using SPIN Model checker for performing formal verification of ad-hoc routing protocols. In model checking, Binary Decision Diagrams (BDDs) are used efficiently to encode the transition relation of the finite-state model. Recently model checking algorithms based on Boolean satisfiability (SAT) procedures have been developed to complement the traditional BDD-based model checking. These algorithms can be broadly classified into three categories: (1) bounded model checking which is useful for finding failures (2) hybrid algorithms that combine SAT and BDD based methods for unbounded model checking, and (3) purely SAT-based unbounded model checking algorithms. Studies in this direction was carried out by Amla et al. [10] who have describes eight bounded and unbounded techniques, and analyzes the performance of these algorithms on a large and diverse set of hardware benchmarks. Most recently, Mancini et al. [11] showed how by combining Explicit Model Checking techniques and simulation it is possible to carry effectively out (bounded) System Level Formal Verification of large Hybrid Systems such as those defined using model-based tools like Simulink. The authors use an explicit model checker (namely, CMurphi) to generate all possible (finite horizon) simulation scenarios and then optimize the simulation of such scenarios by exploiting the ability of simulators to save and restore visited states. The author showed feasibility of their approach by presenting experimental results on the verification of the fuel control system example in the Simulink distribution.

3.2 Run-Time Verification

Run-Time verification process was designed as the superior version of the model spacing for enhancing the computational efficiency level. The run-time verification process usually consist of various software modules, called as an observer, that surveil the proper execution of the program and evaluates its conformity with the system requirements and specifications, which are usually written in temporal logic or as a state machine. Just like Model checking, run-time verification process also can involuntarily check the test runs. Offline verification process is used to analyze the stored execution traces while online verification process is used to check the system operation potentially in the online stage thereby controlling the complete operation to meet the safety requirements of the system. Run-time verification process is also known for its scalable nature as it can meet the demands of a large number of verification of systems.

3.2.1 Related Work of Run-Time Verification

Barringer et al. [12] have presented a rule-based framework for defining and implementing finite trace monitoring logics, including future and past time temporal logic, extended regular expressions, real-time logics, interval logics, forms of quantified temporal logics, and so on. The authors introduce the temporal, finite trace monitoring logic EAGLE. The logic offers a succinct but powerful set of primitives, essentially supporting recursive parameterized equations, with minimal/maximal fix-point semantics together with three temporal operators: next-time, previous-time, and concatenation. The next-time and previous-time operators can be used for defining future time respectively past time temporal logics on top of EAGLE. The concatenation operator can be used to define interval logics and an extended regular expression language. Rules can be parameterized with formulas, and with data to allow for the expression of data constraints, including real-time constraints. Atomic propositions are boolean expressions over a program state, Java states in the current implementation. The logic is first introduced informally through two examples where after its syntax and semantics is given. Finally, its relationship to some other important logics is outlined. Meredith et al. [13] gives an overview of the Monitoring Oriented Programming framework (MOP). In MOP, runtime monitoring is supported and encouraged as a fundamental principle for building reliable systems. Two instances of MOP are presented: JavaMOP (for Java programs) and BusMOP (for monitoring PCI bus traffic). The architecture of MOP is discussed, and an explanation of parametric trace monitoring and its implementation is given. Chen and Rosu [14] have presented a parametric extension together with a mature, optimized and thoroughly evaluated implementation of monitoring-oriented programming (MOP). MOP was first proposed in 2003 as a software development and analysis framework based on runtime verification intuitions and techniques. It was further described and extended, but, up to now, it was not able to handle parameters in specifications, and was not shown, through large-scale performance tests measuring run-time overhead, to be feasible in practice. An implementation of JavaMOP was carried out to support these, together with decentralized monitor indexing algorithms for reducing the runtime overhead. Stoller et al. [15] have introduced the concept of Runtime Verification with State Estimation and show how this concept can be applied to estimate the probability that a temporal property is satisfied by a run of a program when monitoring overhead is reduced by sampling. To validate the approach, the authors have presented a case study based on the mission software for a Mars rover. The results of the case study demonstrate high prediction accuracy for the probabilities computed by our algorithm. They also show that the technique is much more accurate than simply evaluating the temporal property on the given observation sequences, ignoring the gaps. Bartocci et al. [16] have presented Adaptive Runtime Verification (ARV), a new approach to runtime verification in which overhead control, runtime verification with state estimation, and predictive analysis are synergistically combined. In ARV, predictive analysis based on a probabilistic model of the monitored system is used to estimate how likely each monitor instance is to violate a given temporal property in the near future, and these criticality levels are fed to the overhead controllers, which allocate a larger fraction of the target overhead to monitor instances with higher criticality, thereby increasing the probability of violation detection. Since overhead control causes the monitor to miss events, Runtime Verification with State Estimation (RVSE) was used to estimate the probability that a property is satisfied by an incompletely monitored run A key aspect of the ARV framework is a new algorithm for RVSE that performs the calculations in advance, dramatically reducing the runtime overhead of RVSE, at the cost of introducing some approximation error. The utility of ARV is demonstrated on a significant case study involving runtime monitoring of concurrency errors in the Linux kernel.

3.3 Proof based and Symbolic Methods

The proof-based verification process has recently emerged out in the research community and is considered to be highly practical in nature as it is believed to be establishing a link between operational description of the model with the desired model characteristics. One of the good examples, to cite this, is the theorem provers that are considered as a formal verification system for evaluating the design properties using formal deductive, which are usually based on the set of inference rules. Although, such deductive methods are not limited by any decidability attributes, but their applications is highly dependent on expertise and potential human intervention that makes the application not only complex but high challenging in terms of verification process of mixed signals. Various research attempts have been seen in the recent years that emphasized on extended the theorem provers with decision techniques for assisting the verification process as well an ensure automation. Certain preliminary research attempts on verifying analog and mixed signal designs have started most recently, where majority of the research approach are found to be based on deductive-based, inductive based, as well as symbolically based methods to perform formal verification of the sophisticated classes of analog and mixed signal designs.

3.3.1 Related Work of Proof based and Symbolic Methods

In [17], the authors used the PVS theorem prover formally to prove the functional equivalence between behavioral specifications of VHDL-AMS designs and approximated linearized models of their synthesized netlist. The verification was applied for DC and small signal analysis. The ideas presented can be considered as a starting point for a methodology to verify analog designs, yet important extensions should be studied more, like avoiding informal linearization, in addition to tackling more complex verification issues especially related to AC analysis. Similar but more elaborate research was done in [18]. The author proposed an approach for specifying and reasoning about implementations of digital systems that are described at the analog level of abstraction. The approach relies upon specifying the behaviors of analog components (such as transistors) by conservative approximation techniques based on piecewise-linear predicates on voltages and currents. Theorem proving was initially used to check for the implication relation between the implementation and the specification [19][20]. In order to automate the verification process, the author proposed afterwards using constraint based techniques instead [21]. In [22] [23] and [24], the authors propose a new symbolic verification methodology for proving the properties of AMS designs using Mathematica. Starting with an AMS description and a set of constraint properties described in the form of generalized recurrence equations, an induction based verification strategy is applied to prove the correctness of the properties. The procedure is iterative in the sense that if the proof is obtained, then the property is verified. Otherwise, generated counterexamples are analyzed, and the constraint refinement is applied and the verification is repeated until the property is verified or a concrete counter- example is identified. Such methodology is suitable for AMS systems that can be described using discrete-time models.

4. PROMINENT STUDIES

During the last two decades, formal verification has been applied to digital hardware and software systems. Recently, however, formal verification techniques have been adapted and applied to the verification of AMS systems as a way to tackle the limitations of conventional simulation techniques [25]. In addition, hybrid semi-formal techniques combining simulation and formally based methods have been developed as a way to benefit from the advantages of these methods, where logical models are used to analyzing the simulation results.

In past 10 years, the research community have witnessed a substantial amount of work done in formal verification method. One of the significant works in formal verification was seen in the year 2004, where Dang et al. [26] have adopted formal verification method for dealing with time-domain properties of analog and mixed-signal circuits whose dynamic behaviour is represented by differential algebraic approach. However, the verification system doesn’t meet the scope of the larger system. A significant study towards formal verification method was done by Sofiene Tahar, who has largely contributed in this field. In his work published in 2006, Tahar et al. [27] have explicitly discussed formal verification method for analog and mixed signal design with respect to model checking techniques. In 2007, Tahar et al. [28] have presented a symbolic verification algorithm for proving the properties of analog and mixed signal design. The symbolic computation algorithm produces a set of recurrence relations; one for each property of the design. After a gap of 3 years, Tahar et al. [29], in 2010, have proposed a verification process for analog circuits considering noise and process variation. The author has used stochastic differential equation using MetiTarski tool automatic theorem prover based on a combination of resolution and a decision procedure for the theory of real closed fields.

Literatures have also witnessed certain unique techniques in verifying mixed signals. Study on this direction was carried out by Ulus and Sen [30] in 2012, who have introduced an assertion-based verification process for analog mixed-signal design. As analog signals usually associated with some tolerance and variation values in the real world, operations over these signals should consider these facts. Ulus and Sen [30] introduced a pair of boundary signals concept to express tolerance and variation values of analog signals in assertions. It was shown that the concept of haloes is useful to express analog signals in a more natural way in assertions.

In 2013, Goswami et al. [31] have discussed formal verification for networked control system. The study had discussed discretized feedback control system, feedback delay and signal drop, and control requirements. Same year, Liang [32], in his paper, has also discussed various mixed signal verification methods that includes behavior modeling, AMS validation, connectivity verification, mixed-signal Verification IP (VIP), multi-power verification, SoC transistor level simulation and mixed signal functional coverage. Most recently (in 2014), Balasubramanian et al. [33] discussed on the various connect modules flavours, their evolution, relative strengths, and weaknesses or gaps that result in their limited applicability. In this paper, the authors have presented the necessity of connecting modules for AMS co-simulation.

Although, various techniques exist in formal verification process, it was seen that artificial neural network (ANN) was found to be one such non-conventional scheme. The first significant research of verification techniques using ANN was witness in 1996 by Materka [34]. The author has used feed-forward neural network to evaluate the circuit parameters. The advantages of this method are short time needed for parameter identification and a relative high accuracy of parameter estimates. In 2010, Hartel [35] from the reputed University of Heidelberg, Germany, has adopted ANN for performing formal verification of mixed signal. The study presents improvements and testing of a neuromorphic mixed-signal VLSI ASIC. In same year, Manjunath et al. [36] have also adopted ANN to develop novel classes with differential feedback.

Hence, reviewing all the above research papers, it can be said that formal verification model was mainly carried out in VLSI; however, soft computational approach is completely missing. Moreover, a recent adoption of ANN is found in soft-computational approach; however, further research has to be carried out using ANN for ensuring cost efficient formal verification techniques for high-speed mixed signal.

5. RESEARCH GAP

After reviewing the potential research papers introduced till date, the proposed study have explored certain research gap as follows:

- Majority of the research techniques e.g. [5][6][7] have used the verification principles on system models, which are found to be equivalent with respect to particular idea of conformance or functionally similar with respect to their input-output behaviour. Hence, such research techniques call for more investigation of intrinsic behaviour of the system model. In this process, the verification method was found to be based on particular characteristics e.g. transient or steady state response characteristics either in frequency or time domain. However, such methods of establishing a correspondence relationship between the designs are usually done through a series of meticulous and expensive testing stage by proving that two expressions are equal that can be a seriously challenging task for any large scale circuit design. Hence, applicability of such verification processes in large scale circuits are prominent research gap.
- The standard concept of model checking, although has some advantages, are also witnessed with potential limitations. Majority of the studies carried out e.g. [8][9][10] etc in model check has used design and verification principles that are quite computationally expensive in nature and hence model-checking process suffers from state-space explosion issues that not only makes the system to undergo a tedious and time consuming verification process, but also tends to system degradation in terms of real-time constraints e.g. memory and time bound resources in system.
- Run-Time verification process was meant to overcome the challenges that are found in model-checking process. However, the primary challenges found in the study of [12][13][14] etc. are the design and development of adequate monitors. As known, run-time verification process works both in online as well as offline monitoring mode and makes the system verification autonomous, but there are certain potential loopholes in such techniques. For example, the offline monitoring system initiates after the entire sequence is provided in the verification stage. Online monitoring system is found to be highly interleaved with the method of reading the sequence and is found to be almost similar in the way the sequence is studied by any automation principle. Although such online and offline monitoring methods have their advantages in run-time verification process, but it is also accompanied by limitation, which were found not be addressed in past studies. Offline monitoring is highly dependent on aggregating the pre-defined simulation or the execution data that can consume a massive amount of time, as well as buffer of the system. Another potential research gap found in the techniques of run-time verification process is that in the majority of the published work the violations of system specification are not discussed by the authors to be identified during the simulation process.
- The new arena of verification is proof-based methods for analog and mixed signal design. One of the primary strength of proof-based verification method is the probability of performing integration of theorem provers with computer algebra that can be a significant contribution in the area of verification of analog and mixed signal design. Although, it is a theoretically sound technique of performing verification, but still proof-based verification system for analog and mixed signal designs are in infancy stage in research community, especially for high speed mixed signal design and verification. Majority of the published work till date e.g. [17][19][20][21] have focused on the verification of the fundamental properties and not emphasized on complexities involved in large scale circuits for real-time applications. Some of the potential research gap found in this verification process is the analysis of AC, analysis in frequency and time domain, error analysis.

6. CONCLUSION

This paper emphasized that formal verification is one of the most integral phase of the design cycle of the digital circuits as it can ensure stabilized system behaviour precisely with the presence of appropriate inputs whose number can be infinite or too large to be covered up. However, along with inherent advantages of formal verification process, it is also shrouded with various issues for which purpose till date majority of the verification techniques are found to be quite sophisticated in nature and hence not computational cost effective. The approach of the verification process is not simple for analog and mixed signal circuits. It was seen that digital circuits are usually framed as discrete event dynamic system where the inputs are sequences of binary values and the circuit behaviour owing to such binary inputs are also binary in nature that corresponds to the paths in the transition graph. Therefore, the processes of verification system for digital circuits are usually realized using graph based techniques. It was also seen that analog circuits are mathematically modelled as continuous dynamic system that are represented by differential algebraic equation where the inputs are real-valued signals depicted over the real-time axis and the behaviour, they show, are usually in the form of trajectories in the continuous space of the system. Therefore, an efficient verification technique should verify the precise behaviour of a circuit with all possible input signals. It was also seen that adoption of neural network was found quite few in the prior techniques, where it is strongly believed that to a large extent the scalability of the verification process can be enhanced using neural network as it can perform parallel processing of multiple real-valued signal inputs. Therefore, our future work will in the direction of adopting artificial neural network in the formal verification process to ensure the verification scalability and provide better system design cost effectively.

7. REFERENCES

[1] Godse, A.P., Bakshi, U.A.2009. Electronic circuits.Technical Publications, pp.744

[2] Veendrick, H. J. M.2010. Nanometer CMOS ICs: From basics to ASICs. Springer Science & Business Media.Application-specific integrated circuits,pp. 762

[3] Brunvand, E..2010. Digital VLSI Chip Design with Cadence and Synopsys CAD Tools. Addison-Wesley, Computers, pp. 571

[4] Pratap, K., Shelja, M., Bedi, K.2013. Formal Specification And Verification Of Reactive Systems. International Journal of Application or Innovation in Engineering & Management, Vol.2, Iss.3

[5] Darvas, D., Adiego, B. F., Voros, A.2014. Formal veri_cation of complex properties on PLC programs.Springer

[6] Ziquan, T., Shaojun, Y., Yueming, J., and Naiying, D.2013. The Design of a Multi-bit Quantization Sigma-delta Modulator.International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.6, No.5, pp.265-274

[7] Clarke, E., Donze, A.., Legay, A..2009. Statistical Model Checking of Mixed-Analog Circuits with an Application to a Third Order $\Delta-\Sigma$ Modulator. Springer

[8] Cimatti, A., Pecheur, C., Cavada, R. 2003. Formal Verification of Diagnosability via Symbolic Model Checking. ACM-Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp.363-369

[9] Renesse, R., Aghvami, A.H.2004. Formal verification of Adhoc Routing protocols using SPIN Model Checker.IEEE MELECON

[10] Amla, N., Du, X.., Kuehlmann, A., Kurshan, R. P., and McMillan, K. L.2005. An Analysis of SAT-based Model Checking Techniques in an Industrial Environment. Springer

[11] Mancini, T., Mari, F., Massini, A.., Melatti, I., Merli, F., and Tronci, E.2013. System Level Formal Verification via Model Checking Driven Simulation. Springer

[12] Barringer, H., Goldberg, A.., Havelund, K.., and Sen, K.,2004. Rule-Based Runtime Verification. Springer

[13] Meredith, P. N., Jin, D., Griffith, D., Chen, F., Rosu, G. 2012. An Overview of the MOP Runtime Verification Framework.Springer

[14] Chen, F., Rosu, G. 2007. MOP: An Efficient and Generic Runtime Verification Framework. ACM press, pp 569-588

[15] Stoller, S. D., Bartocci, E.., Seyster, J., Grosu, R. 2011. Runtime Verification with State Estimation. Proceedings of 2nd International Conference on Runtime verification, ACM-Springer, pp.193-207

[16] Bartocci, E.., Grosu, R., Karmarkar, A., Smolka, S. A.2013. Adaptive Runtime Verification, Springer

[17] Harrison, M.D., Masci, P., Campos, J. C., Curzon, P.2013. Automated theorem proving for the systematic analysis of interactive Systems, Proceedings of the 5th International Workshop on Formal Methods for Interactive Systems

[18] Kapinski, J., .Deshmukh, J.V., Sankaranarayanan, S., Aréchiga, N.2014. Simulation-guided Lyapunov Analysis for Hybrid Dynamical Systems. ACM-Proceedings of the 17th international conference on Hybrid systems: computation and control, pp.133-142

[19] Illan, D. A., and Eder, K.2014. Formal Veri_cation of Control Systems' Properties with Theorem Proving. arXiv:1405.7615v2 [cs.SY]

[20] Abbasi, N., Hasan, O., Tahar, S.2014. An approach for lifetime reliability analysis using theorem proving. Elsevier- Journal of Computer and System Sciences, Vol.80, pp.323–345

[21] Angelis, E. D., Fioravanti, F.2014. A Pettorossi, and M. Proietti, Verifying Array Programs by Transforming Verification Conditions, Springer

[22] Nakabayashi, T.., Nakabayashi, K., and Kako, F. 2014. An Equation-Based Circuit Design Technique for DC/DC Converters with Symbolic Computation System. International Journal Of Mathematics And Computers In Simulation, Vol.8

[23] Tahar, S., Akhras, I., Nicolescu, G. 2013. On the Verification of a WiMax Design Using Symbolic Simulation.EPTCS, Vol. 122, pp. 23–37

[24] Wei, J., Peng, Y., Yu, G., Greenstreet, M.2013. Verifying Global Convergence for a Digital Phase-Locked Loop. IEEE-Formal Methods in Computer-Aided Design, pp.113-120

[25] Yin, L.2012. Formal Verification And In-Situ Test Of Analog And Mixed-Signal Circuits. Doctorial thesis of Texas A&M University

[26] Dang, T., Donze, A., Maler, O.2004. Verification of Analog and Mixed-Signal Circuits using Hybrid System Techniques. Springer

[27] Tahar, S., Zaki, M. H., and Bois, G.2006. Formal Verification of Analog and Mixed Signal Designs: Survey and Comparison. IEEE North-east Workshop on Circuits and Systems, pp.281-284

[28] Tahar, S., Sammane, G. A., Zaki, M.H.2007. A Symbolic Methodology for the Verification of Analog and Mixed Signal Designs. IEEE-Design, Automation & Test in Europe Conference, pp.1-6

[29] Tahar, S., Narayanan, R., Akbarpour, B.2010.Formal Verification of Analog Circuits in the Presence of Noise and Process Variation. IEEE-Design, Automation & Test in Europe Conference, pp.1309-1312

[30] Ulus, D., Sen, A. 2012. Using Haloes in Mixed-Signal Assertion Based Verification. IEEE-High Level Design validation and Test Workshop, pp. 49-55

[31] Goswami, D., Chakraborty, S., Bhaduri, P., and Mitter, S.K.2013. Characterizing Feedback Signal Drop Patterns in Formal Verification of Networked Control Systems. IEEE Conference on Computer Aided Control System Design, pp.13-18

[32] Liang, C. 2013. Mixed-Signal Verification Methods for Multi-Power Mixed-Signal System-on-Chip (SoC) Design. IEEE10th International Conference on ASIC, pp.1-4

[33] Balasubramanian, L., Poluri, B.K. 2014. Efficient methods for analog mixed signal Verification. DVCON-Conference and Exhibition on Design and verification

[34] Materka, A., and Strzelecki, M.1996. Parametric Testing Of Mixed-Signal Circuits By ANN processing of transient responses. Journal of Electronic Testing, Vol.9, Iss.1-2, pp.187-202

[35] Hartel, A.2010. Improving and Testing a Mixed-Signal VLSI Neural Network Chip. Thesis of University of Heidelberg, Germany

[36] Manjunath, R., Vasudev, S., Udupa, N.2010. Differential learning algorithm for Artificial Neural Networks.International Journal of Computer Applications, Vol.1, No. 18