Optimization and AMS Verification of On-Chip Temperature Sensor on SOC Level

M.Tech Thesis

By

JITESH PRASAD

DISCIPLINE OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2020

(This page is intentionally left blank)

Optimization and AMS Verification of On-Chip Temperature Sensor on SOC Level

A THESIS

Submitted in partial fulfillment of the
Requirements for the award of the degree
Of
Master of Technology

By
JITESH PRASAD

DISCIPLINE OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

(This page is intentionally left blank)

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Optimization and AMS Verification of On-Chip Temperature Sensor On SOC Level** in the partial fulfillment of the requirements for the award of the degree of **MASTER OF TECHNOLOGY** and submitted in the **DISCIPLINE OF ELECTRICAL ENGINEERING, Indian Institute of Technology, Indore**, is an authentic record of my own work carried out during the time period from JULY 2018 to JUNE 2020 under the supervision of **Dr. Santosh Kumar Vishvakarma**, Associate Professor, Indian Institute of Technology.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Jitesh Prasad
16/06/2020

JITESH PRASAD

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

22-06-2020
Dr. Santosh Kumar Vishvakarma

JITESH PRASAD has successfully given his M.Tech. Oral Examination held on 22nd of June ,2020.

Dr. Santosh Kumar Vishvakarma
Signature(s) of Supervisor(s) of M.Tech. Thesis
Date: 22-06-2020

Convener, DPGC
Date: 22-06-2020

S. Mukherjee
Signature of PSPC Member #1
Dr. Shaibal Mukherjee
Date: 22/6/2020

22/6/2020
Signature of PSPC Member #2
Dr. I. A. Palani
Date:

(This page is intentionally left blank)

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to my M.Tech thesis supervisor **Dr. Santosh Kumar Vishvakarma,** Associate Professor, IIT Indore. In numerous instances in this project I was stuck in my progress, he advised me in most efficient and useful ways and motivating me to make the right decision when I was out of options or in moments of distress. His positive approach and confidence in my work helped me go through the project and internship work with ease. I am thankful to him for providing me with the opportunities that shaped my M.Tech to be as it is today.

I would also like to thank **Shubhra Singh**, Senior Design Engineer, and Manager, NXP Semiconductors, India, as she helped us in the domain of course work as well as introduced us to a whole new level of perspective of the VLSI Engineering.

I would also like to thank **Mr. Atul Kumar, Mr.Prateek Singh, Mr. Harsh,** Senior Design engineer, and whole NXP AMS India Design Team for helping me by showing me the right path and clearing my doubts. I am sincerely grateful to my M.Tech batch mate **Ruchika** and **Prashant**, for their valuable discussions wherein I went to them with a problem in each one of them and came out with many ideas.

I sincerely acknowledge the support of **IIT Indore** and **MHRD** for supporting my M.Tech, by providing lab equipment and facilities, and TA scholarship, respectively.

Last but not least, my work would not have been possible without my parents' encouragement, whose tremendous support helped me stay positive and overcome the worst of hurdles. To them, I will forever be grateful.

(This page is intentionally left blank)

DEDICATION

To my Teachers and my Family

(This page is intentionally left blank)

ABSTRACT

Analog and Mixed Signal (AMS) designs are essential components of today's modern Integrated Circuits (ICs) used in the interface between real-world and digital signals. Analog and mixed-signal circuit designs are gaining the popularity of the semiconductor industry as the compact design demand and more functions equipment increases in the market. In the era of 5G, AI, Self-Driving Cars, and the Internet of Things (IoT), the need for Mixed-Signal System-on-Chips (MSSoCs) is increasing gradually. The Simulation Program with Integrated Circuit Emphasis (SPICE) tools is still the primary technique to verify analog circuits. However, the SPICE simulation's speed is still an issue as the complexity of the degine is increasing node by node. Other verification strategies are required to verify the complete analog and digital functionality within the time before the market. Current industrial verification methodologies, each addressing specific verification challenges, are useful for detecting and eliminating design failures.

Nevertheless, decreases in first-pass silicon success rates illustrate the lack of cohesive, efficient techniques to allow a predictable verification process that leads to the highest possible confidence in AMS designs. For desire digital simulation even for sophisticated analog design with more efficiency in the speed factor, then SPICE Universal Verification Methodology (UVM) is used. This thesis will present the study of AMS and AMS verification with an On-chip CMOS Temperature sensor followed by the results. This approach is generally used to verify SoCs in the automotive industry.

Alongside as the demand for AMSSoCs is increasing, On-chip thermal monitoring is becoming crucial as VLSI circuits are becoming more sophisticated with time and recent technologies. On-chip thermal monitors provide information about the power and thermal management structures, which is essential to prevent chip temperatures from destroying the device or maintain the lifeline. Several sensors are being placed on the chip to monitor the exact data of the on-chip temperature. These sensors are compatible with the technology node, not even taking large area and proves accurate data with standard process variation and required power supply variation. The studied temerature sensor is based upon a self-stabilized feedback architecture.

LIST OF PUBLICATIONS

- Neha Gupta, **Jitesh Prasad**, Rana Sagar Kumar, Abhinav Vishwakarma and Santosh Kumar Vishwakarma, "A Robust Low-Power Write-assist Data-dependent-Power-Supplied 12T SRAM Cell", 23rd International Symposium on VLSI Design and Test 2019 (VDAT-2019)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

VLSI	Very Large Scale Integration
SOC	System-on-Chip
MSSoCs	Mixed signal system on Chips
IP	Intellectual Property
P&R	Placement and Routing
IOT	Internet of Things
IC	Integrated Circuit
UVM	Universal Verification methodology
UVC	Universal Verification Component
UHD	Ultra High Definition
HLS	High Level Synthesis
RTL	Register Transfer Level
PI	Physical Implementation
HW	Hardware
SW	Software
TCL	Tool Command Language
DFT	Design for Testability
VHDL	VHSIC Hardware Description Language
SPICE	Simulation Program with IC Emphasis
DRV	Design Rule Violation
DRC	Design Rule Check
LVS	Layout Vs Schematic
FET	Filed Effect Transistor MOS Metal Oxide Semiconductor

CHAPTER: 1

INTRODUCTION

The trend in increased applications for consumer electronics and commercial electronics does not seem to reach an end anytime soon. Wearable electronics are becoming popular and have a range of applications such as health monitoring, navigation, media, communication, and more applications that are probably right around the corner. So it seems when hearing about the next big thing of tomorrow, the Internet of Things (IoT), AI, Autonomous Vehicles, and 5G. In today's world, where any physical device could be accessed from anywhere through the interne

t, and Car is running on the road without any driver. Search missing keys on the internet and their location will be given in seconds, Patients with heart disease could have an embedded chip inside the chest that monitors the heart. Information about the patient's heart can be sent to computers at a hospital, which runs powerful algorithms, potentially predicting a stroke and saving lives. This is some of the forecast potentials of the IoT and AI, and it is predicted that, in the future, more than 50 billion devices will be connected on the internet [1, p. 3]. For the semiconductor business, all these wearable and physical devices connected to the internet means high demand for rapid communication. Hence, the need for Mixed-Signal Systems-on-Ship (MSSoC) increases as well as the complexity of these chips, due to the extensive range of possible applications. SoC refers to the integration of different electronic intellectual property (IP) and custom design blocks into a single integrated chip, as depicted in Figure 1.1

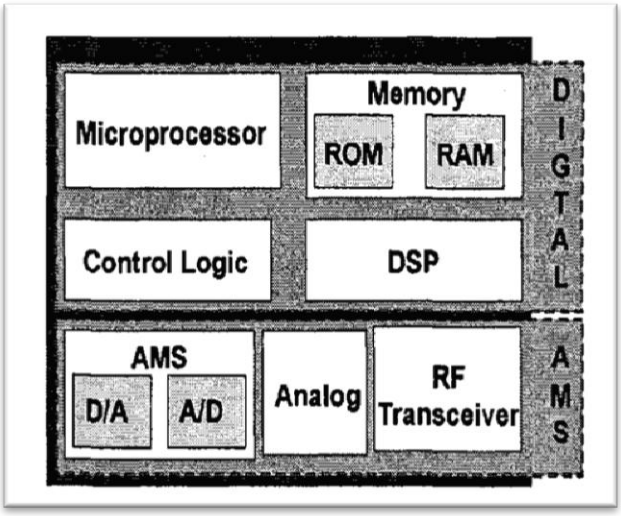


Figure 1.1 System On Chip

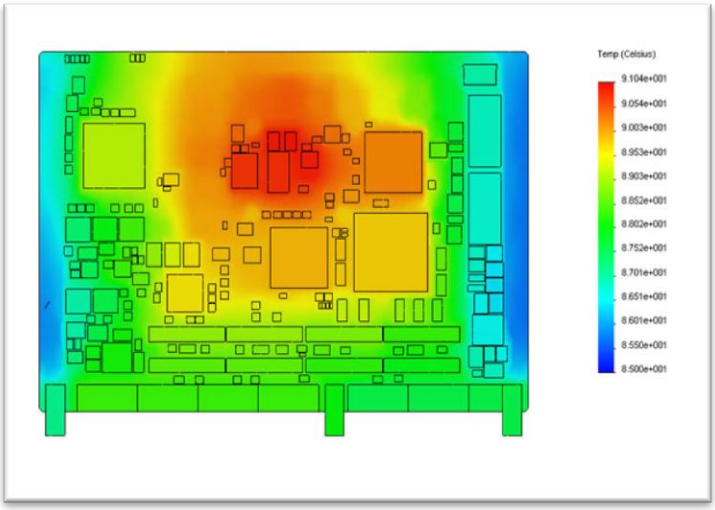


Figure 1.2 SoC Hotspot

The current situation of the soc is more complicated than earlier, which required valid and exact verification. The main reason for this question is how to verify the whole soc, which consists of millions of transistors. The only thing that can be done is to ensure that the chip has been tested for as many test-cases as possible before it reaches the end-product. With the ever-increasing the less time to market, the increasing complexity of the system on chip is sometimes causing less care verification of the entire system. In modem VLSI design, the thermal monitoring unit is an essential part of the IP as With the trend continuing for more complexity, increased speed, and higher levels of integration; the on-chip thermal monitoring unit is directly connected to power uses and functions, so the TMU is one of the most vital parts of the modern IC. As the complexity is increasing sharply with the integration of millions if transfer on the chip, the bug is also is increasing in the same manner. It says hidden, and to find it, we need to throw verification. If a bug finds after the post-silicon level, it will cost a colossal amount as the whole chip process is needed to start again. So it necessary to find a bug before it is taped-out.

1.1 **MOTIVATION**

In the past few years, the world has noticed the fantastic demand for SOC and Application Specific Integrated Chips(ASIC). This integration's primary intention is to improve the turn-around time of ASICs by adopting SoC design approaches [2]. The verification from convention and Morden methodology is sometimes time taking or depends on the requirement process. Equivalence checking and model checking are Formal methods proving to be very successful for the validation of digital designs. Advanced tools and languages, which support techniques like constrained random stimulus generation and assertion-based functional coverage analysis, have enabled the verification of incredibly intricate digital designs [3, 4]. on the other part, analog circuits are still verified using the differential equation models and SPICE simulators. AMS circuits have been evolving rapidly over the past few years. With the integration of millions and billions of transistors on a chip, the verification with a conventional for formal techniques is sometimes time taking as error full. Improving the verification quality of these complex SoCs comprising both digital and AMS circuits requires either an impractical increase in the number of resources or the Usage of advanced verification methodologies. The main aim of the verification is to verify the functional verification during the initial phase of the circuit. Functional bugs can occur due to simple mistakes like the incorrect connection of the wires of a bus and connecting active high signals to active low inputs at an interface. SPICE is beneficial for performance verification, especially for an analog circuit with full detail, but it is not that sufficient for function verification aimed at accuracy. As the complexity or size of a circuit being simulated increases, the time required for SPICE simulation becomes a showstopper for verifying the circuit adequately. This has been a tradition followed for digital circuits since the invention of hardware description languages (HDLs), and it has proved to be very successful in catching functional bugs in the initial stages of the design.

For accurate parallel verification of both analog and digital parts, the industry adopted a new kind of methodology suitable for analog and digital both. Verification of analog designs is complicated not only because analog signals are continuous in time and value, but also due to the increasing process variability, number of parameters, and physical effects which have to be considered. The modern HDL language like Verilog AMS and VHDL is kind of suitable for both, but it has some foundation too for accuracy for a different type of circuit. With this consideration, we will examine Universal Verification Methodology with a suitable environment and see how it reuse the setup as required for the whole IP.

1.2 MAIN GOAL

the main aim of this thesis is to study the verification flow and study the functionality of the thermal monitoring unit. In the design of today's SoCs, functional verification complexity rises exponentially, with hardware complexity doubling exponentially with time [5]. However, more than 75 % of the full-chip-making time is dedicated to verification itself[5]. As SoCs became multifunctional, more analog blocks are integrated on a single chip. Therefore, mixed-signal verification difficulties became more visible to verification teams. When bock level verification is completed, it is required to develop a technique through which analog and digital sub-blocks of design are considered holistically. The studied verification methodology is UVM, which is class libraries based verification environment that can be sued again for different IP once programmed. Taking advantage of constraint random stimuli generation offered by UVM, the engineering effort has been turned into building automatic checkers instead of writing directed tests. The typical approach in UVM based verification flow is to define a test case and simulate the design under test (DUT) with random constraint stimuli to put the design into corner cases. Automatic checks continuously check the functionality according to the generated stimuli. Furthermore, coverage mechanisms are used to measure the inspected specification of the DUT and point out to the verification closure. The challenge of such systems is that verification becomes a real issue, and with the time-to-market requirement, it is nearly impossible to perform all the necessary verification in time. According to [12], 70% of the logic design phase of a chip is devoted to verification, and 60% of SoC re-spins are due to mixed-signal errors. The increasing functionality of SoCs has been an issue for quite some time. Hence, methodologies for improving the verification process of integrated circuits (IC), such as the universal verification methodology (UVM), have appeared. However, UVM is used to verify digital parts of the chip, and analog parts are verified by the SPICE tools as well as Verilog A. Spice tools yield accurate simulation of the analog circuits, but they are time-consuming. To overcome this, we are using Verilog A and Verilog AMS. With the increasing complexity of the SoCs' analog parts, performance becomes an issue, and other approaches are needed.

Among the essential functions of AMS, designs are the processing of the analog signal on the system's front and back ends.

1.3 ORGANIZATION OF THE THESIS

This thesis is structured as follows in the first chapter, an introduction about the recent demand for VLSI tech and motivation, followed by the goal of the project. In the second chapter and Problem, Formulation is discussed with the review of different verification approaches. Next, verification and its utilized methods in the digital world are studied., a brief study of analog circuit simulation is given. In the third chapter, after a comparison study of different modeling approaches, what is the mixed-signal circuit and mix signal flow is discussed. In the fourth chapter, AMS Verification is discussed with the UVM verification methodology. It is in chapter five, where the main feature and one of the huge problems of the VLSI industry are discussed, which is the Temperature Monitoring Unit. Finally, in chapter six, the results of AMS Verification on the TMU is described.

CHAPTER 2

REVIEW OF PAST WORK AND PROBLEM

2.1 PAST WORK

The focus of this thesis work to present the UVM verification method to verify the TMU. Since different verification methodologies are used to verify digital and analog blocks, it is crucial to have a holistic view of both digital and analog sub-blocks of the design at chip-level verification. As it was presented in [13], to be able to use advanced verification techniques like randomization and, therefore, to avoid writing analog real-valued input stimulus in an orderly way, transaction-level modeling is extended into the analog domain. In [13], the analog transaction was introduced. The proposed technique in [13] generates an analog stimulus through passing data structures to the driver.

Via sequencer and driver communication transactions are passed to the driver. Driver wiggles DUT's pins according to the specified protocol and by decoding parameters within a received transaction. Hence "transactions offer an abstraction of the protocol." [13] UVM_seq item class holds parameters that are required to define an analog wave (like slope and value of a certain point in time for a linear signal). During the run task of the driver, when it sends a request to the sequencer for a new sequence item, those parameters within the sequence item will get randomized. As an example, the generation of a harmonic analog signal was presented in [13]. The randomized spectrum of the signal was delivered by the UVM seq item class, and the harmonic signal was obtained. It is *System Verilog's* Direct Programming Interface (DPI), which provides the possibility to use various algorithms written in *C language*. System Verilog like many other modern languages, provides the feasibility of interconnection with libraries written in C. System Verilog layer and foreign language layer are completely isolated. Communication between these two layers is through function calls between languages. The most important advantage of the proposed method in [13] is that by randomly generating complex analog stimuli, it is possible to detect design bugs much faster than writing direct tests.

2.2 DESIGN ISSUE

In the design of nano scale ciuit design engineers has to deal with some design essues to maintain accuracy and profit. Figure 2.1 shows issues in AMSSOC design . the power optimization is huge issue As the popularity of mobile devise and and long battery life demands is increasing nowadays . clock gating, adaptive voltage scaling (AVS), dynamic clock frequency and voltage scaling (DCVS), and static voltage scaling (SVS) and power gating has already been addressed in the case of digital circuits, low power design various techniques . Power consumption problem is divide in to two categotyes

Figure 2.1 Typical AMS-SoC design and fabrication issues[16]

Dynamic power: during the operational phase of transistors, when switching occurs. Static power: leakage power due to capacitive current Leaking or drain gate voltage lowering and other leakage phenomena during the non-operational phase of transistors through the gate of the transistor. During run time operation, power densities associated with recent technology evolutions have led to the apparition of thermal gradients and hot spots. Temperature is the most important design concern in modern integrated circuits. Speed, cooling budgets, reliability power consumption, etc. are circuit parameters that will be affected by Temperature. Thermal Monitoring Unit (TMU) we use the study of the behavior of the chip relying on the information to help us monitor the reliability and lifecycle of the chip.

CHAPTER 3

MIXED SIGNAL SYSTEMS

3.1 INTRODUCTION

A huge evolution has been experienced During the last decade in the computer and telecommunication industry. New standards and protocols to improve the channel capacity, robustness and reliability being introduced during This ongoing evolution. In addition to economic decisions, a number of technical barriers had to be crossed to enable these achievements. Discrete components and expensive technologies are used during the assembly of Former electronic products. New circuit topologies have emerged that overcome earlier existing problems with traditional topologies, all Thanks to the huge investments and persisting research, cheaper technologies have become feasible. For fully integrated transceivers, The use of CMOS technology has recently been proven. Production volumes, reduce the overall costs, and diversify its products is need to be increased by telecommunication companies To address this market expansion. Consolidate a large market share from the short time-to-market of its product is even more important. Consumer electronic systems are typically mixed-signal systems with analog and digital components. Different types of operations performed by The different mixed-signal system components. Various skill sets and design methods required for the SoC design needs. In this chapter, the typical structure of a mixed-signal system is discussed. The issue faced during Nanoscale CMOS design and the target fabrication technology is discussed. Mixed Signal Systems Modern consumer electronic systems (e.g., mobile phones, BLURAY players, health monitoring systems) have profound impacts on society. Analog and digital, and mixed-signal components and their interfaces heterogeneous systems consisting of diverse on a single board or die of Modern consumer electronic systems designed as. Typically, they are analog/mixed-signal systems-on-chip (AMS-SoCs) or mixed-signal systems where analog and digital portions are integrated on the same die for cost-performance tradeoffs [8, 9]. The current situation of CMOS design is the complexity present the following design challenges: variability, leakage, power, thermal issues, reliability, and yield [10, 11]. The demand growth of mobile appliances in which battery life dictates system performance. The situation is worsened for such highly complex chips because the time-to-market has been reduced significantly. In this situation, methodologies for fast design space exploration are much more important than ever to timely produce error-free chips as Soc systems are becoming more complex. For digital and analog circuits Numerous simulation tools are used to aid in the design process, but the additional physical/electrical effects such as (parasites) of circuit elements after the physical layout stage, as the technology progresses and scales down, are making the calculations more complex [12]. A very dramatic effect on the output is experienced

every time due to The parasites presented in the circuit. The accuracy of the physical design process cannot reach using some design tools for analog and mixed-signal circuits on the schematic design level since they do not account for parasites of the circuit. If we see the design cycle of mixed-signal components is significantly long as accurate, circuit-level simulation is very CPU intensive. For example, the simulation time for a TMU on a full parasitic netlist is of the order of many days to weeks. Thus, the speedup of the design process can have a dramatic impact on time-to-market. When such circuits are designed using Nanoscale technology, This situation is further aggravated where the transistors are modeled using hundreds of parameters. Due to the many parasitic effects, It is also very difficult to accurately predict the performance of AMSSoC component circuits in high-frequency applications. The original design is iteratively adjusted by attempting different values of the design variables To meet the desired design specifications.

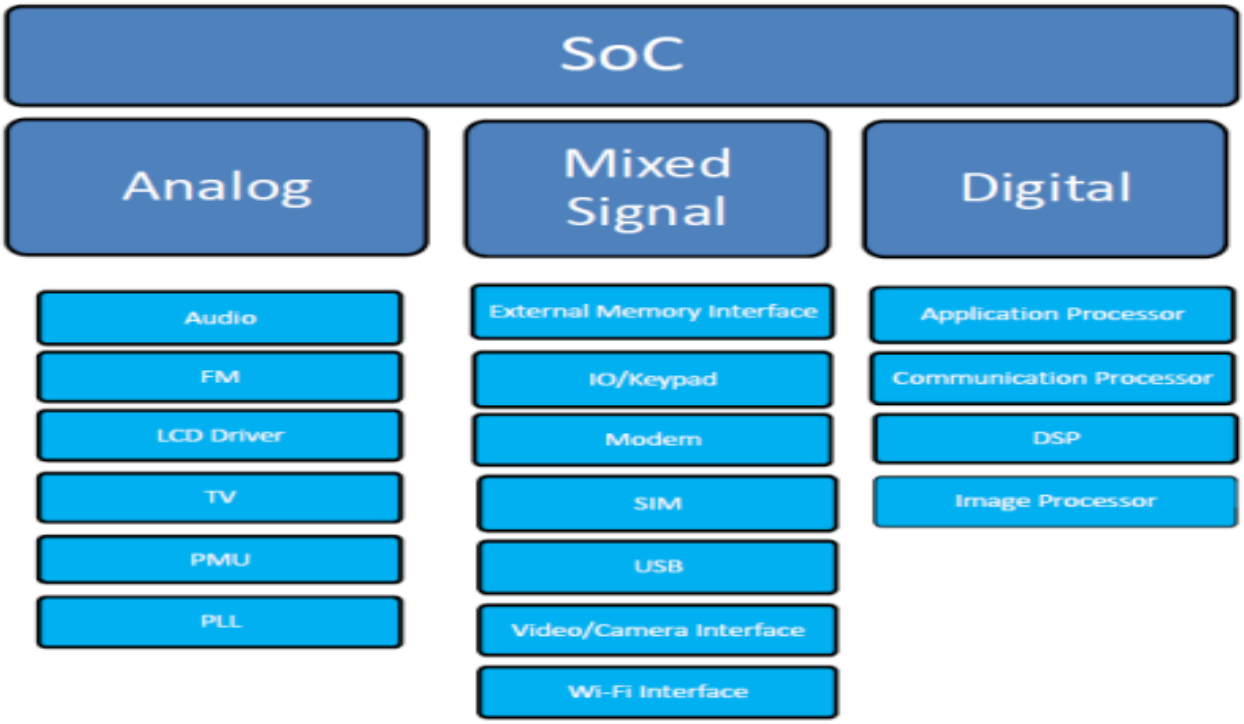


Figure 3.1 Breakdown of a typical SoC

A large number of design variables results in an enormous amount of different possibilities for alternative design tradeoffs. Thus, in the context of efficient design of AMS-SoCs, several questions arise:

i. the fast layout of AMS-SoC components can be generated in how many ways?
ii. Fast design optimization performance in efficient ways?
iii. Efficient techniques for sample the design space?
iv. Is there an optimal metamodeling methodology for RF IC components?
v. What is an efficient RF IC design flow that can use metamodels effectively?
vi. What algorithms can be used for AMS-SoC design optimization?

The design-time constraints on the designers establish the need to decrease the simulation, design, and optimization times. These efforts will be silicon accurate if full-blown parasites are taken into account. The full-blown parasites include resistances (R), capacitances (C), self-inductance (L), and mutual inductances (K), which may be due to active devices or interconnects. Creating the physical design, particularly when nanoscale technology is used, involves very time consuming manual labor. The classical design flow needs many manual iterations involving physical layout and schematic adjustments. The number of manual iterations depends on the skill and qualifications of design engineers. Hence, designers usually encounter problems in the post layout stage of the design process that affect the design specifications and skew the output of the designed system. Because of that, the layout stage, which is a time-consuming process by itself, usually takes more than one iteration, and in the worst case, the designer has to go back to the drawing board. Therefore in many cases, the final design becomes suboptimal with designers trying to fit their design into tight specifications and time to market deadlines. The design process for analog circuits can take a very long time. Once the circuit design is complete, it needs to be adjusted by parameterizing the design variables to produce the desired output. It is very difficult to predict the performance of analog circuits in high-frequency applications due to many parasitic effects [13].

The reuse of a large part of the design can be drastically accelerated By following a systematic design methodology consistently. Typically a top-down design and bottom-up verification method are used [8]. Therefore, different abstraction levels are identified. Using models of the composing blocks At each level, the behavior of the overall circuit is evaluated. Its output variables are given the input variables in Each model's calculations. The format of these variables depends on the abstraction level. Higher-level model meets its specifications at a minimum cost (i.e., Minimum power consumption or chip area), and lower-level model parameters are assigned a value this verified Top-down implementation. This design methodology is supported by a number of behavioral models at different levels.

3.2 ABSTRACTION'S LEVEL

the definition of different abstraction levels is the definition of A top-down & bottom-up verification methodology. In the digital design methodology, these levels are well defined, as seen in Figure 3.2. signal transfers are usually described, At the behavioral level, under the form of an algorithm. Signals are represented using real's and integers. The next level is the register-transfer level. The execution time frame of the algorithm is partitioned into clock cycles. Moreover, all data have a binary representation. All operations are scheduled on a number of functional blocks (multipliers, accumulators, registers, etc.) At the functional level. Control signals are clearly determined. After that, at the gate level, elementary logic gates are formed from functional blocks. In the end, the logic gate is replaced by the transistor level. In the analog

domain, these different levels of abstraction are less distinguished, as seen in Figure 3.2. At the functional level, the basic signal flow is described in terms of mathematical functions. No conservation laws on the interconnecting nodes have to be satisfied. One level lower, at the behavioral level, these mathematical functions are replaced by a number of high-level blocks, e.g., linear transfer functions, op-amps, A/D Converters, etc. The conservation laws are now enforced. Still one level lower, at the macro level, the circuit consists of elementary components, such as resistors, capacitors, controlled sources. By adding more detail to these models, second-order effects (slew rate, finite gain, etc.) can be taken into account as well. Finally, the circuit is decomposed into its elementary components, and all the design parameters can be assigned values at the circuit level, [8].

3.3 MODELING REQUIREMENTS

Setup and setting p the proper libraries are the most important task of the flow; to check the design with the market standards and visualize our work, it's really imped to work with the right library. The complexity of the analog part is sometimes required using different libraries for a different part. Once analog building-block models are developed, they can be instantiated to perform a system design. The higher-level description is used as a specification and reference When implementing a lower level of the design. When developing models, some requirements have to be fulfilled. The functionality should be modeled in a generic, parameterized way. All circuit aspects that influence this specification should be included in the model When one wants to evaluate a certain specification. In general, a tradeoff can be made between the accuracy of the model and the necessary evaluation time. In the first stages of a top-down implementation, less accurate models can be allowed to get a rough estimate of the design space.

Finally, all the models should be implemented in the industry-standard model. This will ease the exchange and reuse of models. For simulation and synthesis, in the digital model, Verilog and VHDL language are used. In the analog domain, however, the Verilog-AMS standard is about to be finalized, and commercial simulators will soon be released.

3.4. Existing Analog and Digital Design Flow

3.4.1 Digital Design Flow

There are various formats in the industry for the digital design flow. Generally, every company has a different flow, and they modify their design, according to their resources and research lab. The main reason behind the different models of different companies is they use different computer-aided design tools. The following shows major steps in digital design:

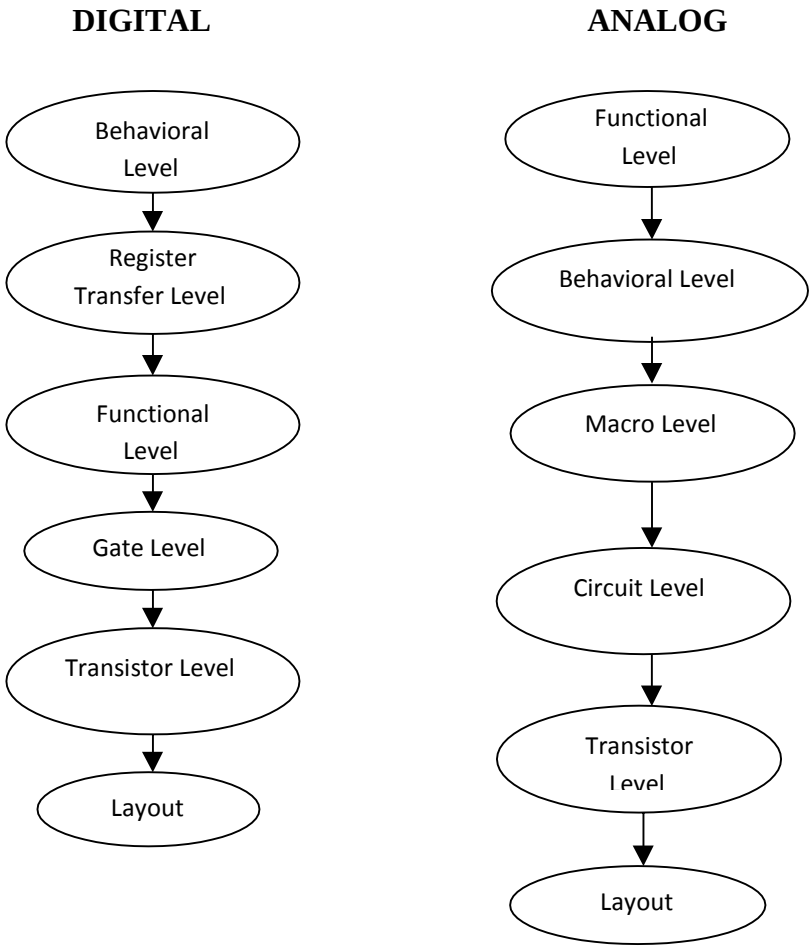


Figure 3.2 Different abstraction levels

- High-level System Design
- Architectural Exploration
- RTL Simulation
- Design For Test
- Timing-Driven Logic Synthesis with Scan Insertion
- Gate-Level Simulation
- Floorplanning and Timing-Driven Placement
- Extraction and Delay Calculation
- Pre-Clock Tree Synthesis Timing Check
- Clock Tree Generation
- Pre-Route Golden Verilog netlist Verification
- Routing
- Post-Layout Static Timing Analysis

- Physical Verification (DRC & LVS)
- Manufacturability
- Tape-Out

3.4.2 Analog Design Flow

Design flows are dependent on the design environment's CAD tools. CMC's analog design flow was selected for the analog part of the proposed mixed-signal design flow. The steps will be further described in the next section. The primary objective of the analog design flow is to produce working parts, and the secondary objective is to make the parts reusable.

The following shows the major steps in a typical analog design flow:

System-Level Design

- Set Design Goals and Priorities
- Preview Spec Gate
- Functional Capture & Architectural Exploration
- Packaging Selection
- Verification: Simulation and Test Plan
- Partitioning & Block Behavioral Modeling
- Top-Level & Block Specification Documentation
- First-Pass Gate

Block-Level Design

- Topology Selection & Block Schematic Capture
- Test Plan Update
- Pre-layout Simulation with Estimated Parasitic
- Optimization
- Layout
- DRC/LVS
- Post-layout Simulation

- **Block-Level Gate & Risk Assessment**
- Block Specification Update
- **Pre-top Level Gate**
- Top-level Layout Design
- **Post-layout Gate**
- Complete Documentation & Test Plan
- Create ROL (read-only library) & Archive
- Tape Out

3.5. Mixed-Signal Design Flow

If we discuss mixed-signal systems design approaches, there are two approaches. In the first approach, the mixed-signal system is decomposed into pure digital and pure analog subsystems, and each of these sub-systems is designed individually using analog and digital design methodologies. Designers should find a way or model for the connection port of analog and digital systems. Finding the circuit equivalent of the analog sub-system could be very difficult, and in most cases, it is not accurate enough to model the actual behavior of the analog circuit. On the other part, the digital part is being simulated with analog enforcement, which is different and sometimes not that accurate for the digital part. In this approach, the simulation of the overall system is only possible at the final stage of the design and after completing the layout. Any changes at this stage are difficult and time-consuming. Designers can only simulate the effect of digital and analog interactions at the system level. Before the post-layout simulations, we can see some similarities between the two flows when we were comparing different abstraction levels in the digital and analog design flows. The second approach for designing the AMS is based on analog-digital co-design. In this approach, after the system-level design, the behavioral / RTL model of the overall system will be developed and verified. VHDL-AMS can be used to have a mixed-signal model of the chip. The digital part would be described using an RTL synthesizable subset of the language, while the analog part would be partitioned into functional blocks at the functional or behavioral level, e.g., Filters, VCOs, Opamps, etc. The whole model can be simulated using test benches written in VHDL-AMS. The digital part of the chip can be synthesized using a logic synthesizer to produce a gate-level netlist. Analog blocks are individually designed, and Standalone Verified at the transistor level. The next step is the block design, which has different steps for digital and analog blocks. After completing each block, it is possible to test the block in the interaction with

other blocks using the test benches developed earlier. Standard cell place and route tools can produce the layout of the digital part from the gate-level netlist. Those elements related to the digital part are used to compute delays that are stored in SDF (Standard Delay Format) files. The final simulation can be done using the extracted layout view of the overall chip Figure 3.2 shows this approach.

3.5.1. PROPOSED MIXED SIGNAL DESIGN FLOW

The proposed mixed-signal design flow is based on analog-digital co-design methodology and has two distinguished levels of abstractions.

- System-Level Design
- Block Level Design

A) System-Level Design

Figure 3.3 shows this part of the design flow. The following major steps shall be followed at this level:

Set Design Goals:
setting the clear objectives of the current design and identification of a possible solution for achieving the objectives and selection of implementation for the solution. The design process starts with a clear statement of the problem. Performance metrics (speed, power, noise, etc.) and other specs like descriptions of functions, estimated and projected operating constraints (bias, thermal, I/O impedance, proximity, etc.) are the points of concern.

Preview Spec Gate:

The right decisions have been made before proceeding further, and a peer review should be done to ensure that the overall scope of the design project is acceptable. With goals and priorities set and resources planned.

System-Level modeling:

The preview system specs are captured with Matlab/Simulink or VHDL-AMS for a more precise definition and verification of the specs. At this stage, the use of reusable (IP) blocks should be considered as their availability can have a strong impact on the selection of architecture and development time.

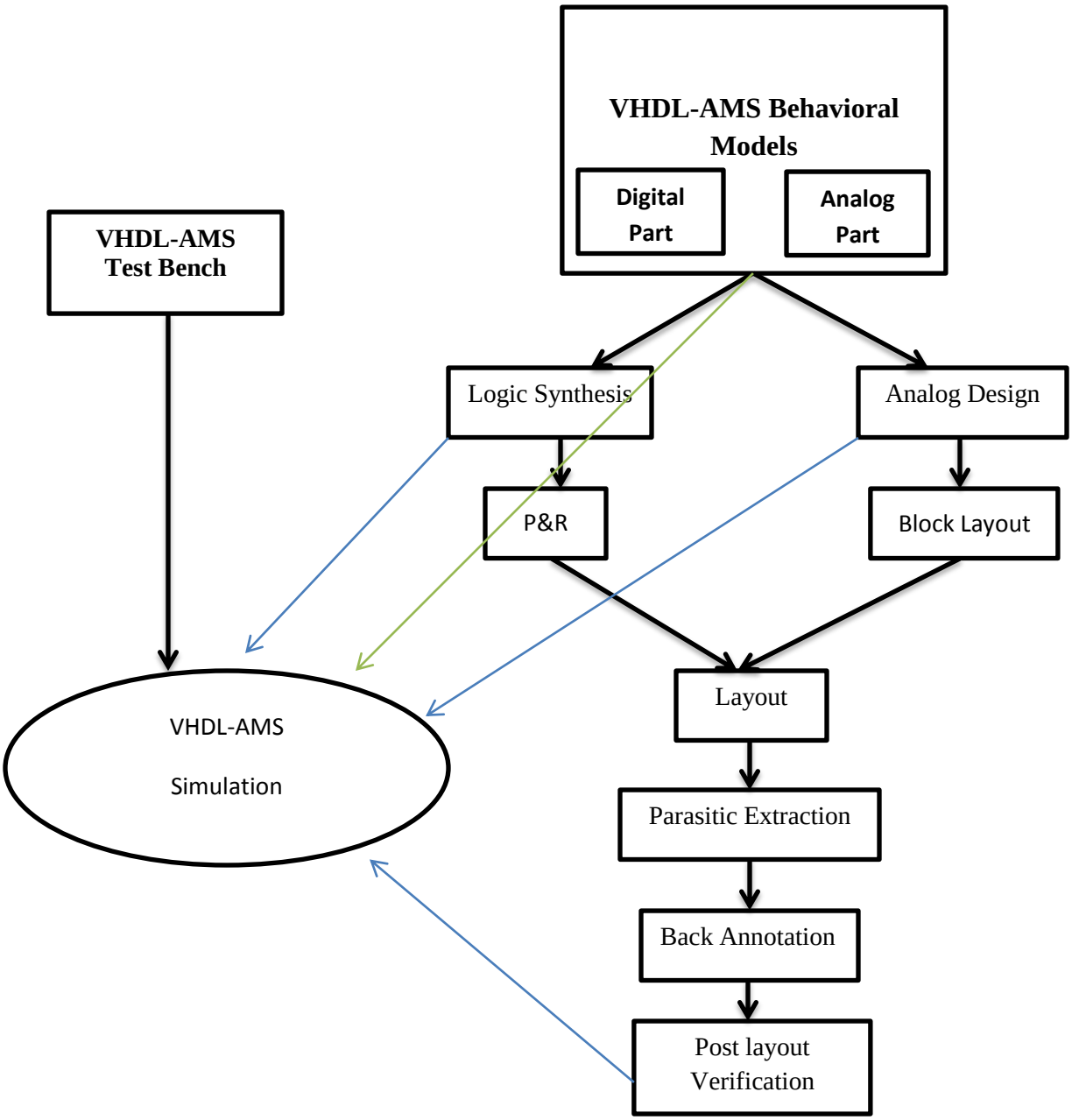


Figure 3.3 Analog / Digital co-design

Packaging Selection

The design engineer should select the package carefully as design, or the final product has to interact with the outer world, and the unsuitable package me affects the overall behavior of the final package.

It can improve the accuracy of behavioral modeling if a packaging model is available and enhance the validity of simulation/test plans.

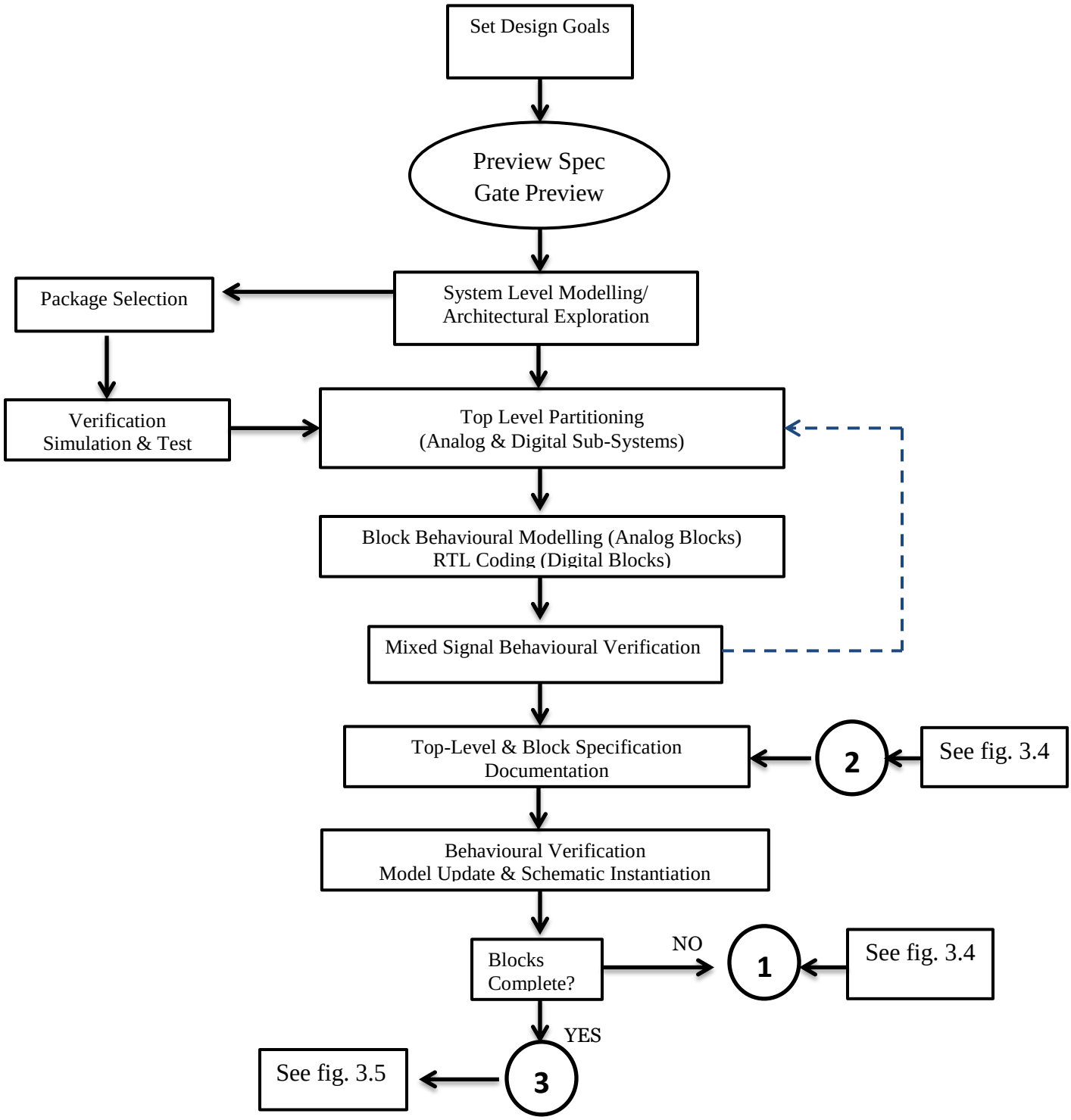


Figure 3.3 System Level

Verification: Simulation and Test Plan

verification plan, including a simulation plan and test plan, should be considered. The plan about top-level netlist and each analog and the digital block is modeled, and simulated.type of simulation like transient, ac, noise, frequency domain at each level will be used how the stimuli will be created, how the interference between the blocks will be modeled, what tests to be performed, what test equipment to be used, how to bias the chip, what supply decoupling is satisfactory, what DFT techniques should be used to facilitate testing and diagnosis. Some of verification plan details can also be derived after design partitioning and behavioral modeling. It is an iterative process between verification planning and partitioning and behavioral modeling.

Top Level Partitioning

The architecture is partitioned into digital and analog blocks, and each of the digital and analog blocks is further partitioned into basic sub-blocks. The architecture must be partitioned in a way that maintains as much hierarchy as possible, makes use of common implementable functional blocks, minimize critical connections between blocks, and must be consistent with the chosen packaging technology in terms of electrical, mechanical, and thermal characteristics.

Block Behavioral and RTL coding

Behavioral modeling can be done for both analog and digital blocks using VHDL-AMS .The overall behavioral model of the system can be simulated and verified. There could be different levels of abstraction for each behavioral model starting with simple models and then designing more complex models. Digital blocks can be modeled both at the behavioral and RTL levels. Power domain and clock strategy must be considered to obtain the optimal power distribution and consumption, to enhance rout ability, reduce interference among blocks, and facilitate clock signal generation and distribution.

Mixed Signal Behavioral Verification

Top level behavioral simulations must be performed to achieve satisfactory functional and performance results against the preview specs before proceeding further down the flow. Otherwise the designer needs to go back to structural mapping, partitioning and behavioral modeling until the targets are met.

Top-Level and Block Specification Documentation

When the results are satisfactory, the designer needs to document the functionality, performance, interface conditions, physical size, and power consumption for the top-level design as well as each autonomous and reusable block. The documentation is crucial in tracking the design optimization process, helpful in guiding design convergence and essential for passing the gating process. It is also required for revision tracking.

B) Block Level Design

Block-level design for digital and analog blocks is different. The design of digital blocks is based on digital synthesis and standard cell libraries. It is a process that can be done using available synthesis tools. Designing analog blocks is done by the designer and his knowledge of analog circuit design. Figure 3.4 shows this part of the design flow. The steps for each of these design processes are explained briefly.

ANALOG BLOCK DESIGN

Topology Selection and Block Schematic Capture

A schematic corresponding to the block behavioral description, and it must be properly linked to the behavioral model for later instantiation. Each schematic should have proper pins by which we can now proceed with symbols for the verification.

Test Plan Update

This is a good time to take into account all the necessities to test the final circuit and revise the test plan since more design details are available at this point and to

Pre-layout Simulation with Estimated Parasitic

When building the block schematic, the design should verify or run a pet simulation to check the pre-order parasitic related problems before layout to mostly prevent and identifying problems early and facilitate design convergence. A rough layout also helps estimate the block size that can be important in the overall design. HSpice or Specter can be used to perform several types of simulations, including DC, transient, AC (noise), and nonlinear frequency domain analysis. A DC analysis is used to establish proper biasing.

Optimization

At this point, on the analog design of each block, performance and yield optimization is performed, if necessary. Fine-tuning of circuit components and biasing or maybe the addition of circuit components is required for the. Even though a nominal design can meet specifications, a significant number of chips may fail when component values are allowed to vary within their tolerance.

Layout

Cadence Virtuoso-XL can be used to perform schematic driven placement After the schematic has been optimized to meet specifications, for the cells or devices and to route the layout. For handling the complexity of a large block design, a Hierarchical layout style is recommended.

DRC/LVS

Frequent design rule checks (DRC) on layout will leads the design to errorless design. The design must be clean of violations against all rules. For deep sub-micron technologies, space fill rules and antenna rules are as critical as other rules and must be observed after layout and DRC, Layout Versus Schematic (LVS) verification is performed to ensure that the netlist created by the schematic and that of the extracted layout match. If they do not, the errors should be corrected

Post-Layout Simulation

Extraction of the layout followed by post-layout simulation ensures that most of the parasitics are as expected, and any unaccounted for parasitic have not significantly affected the design's performance. The post-layout simulation results may indicate that some adjustments or optimization is required, perhaps even going back to block schematic capture.

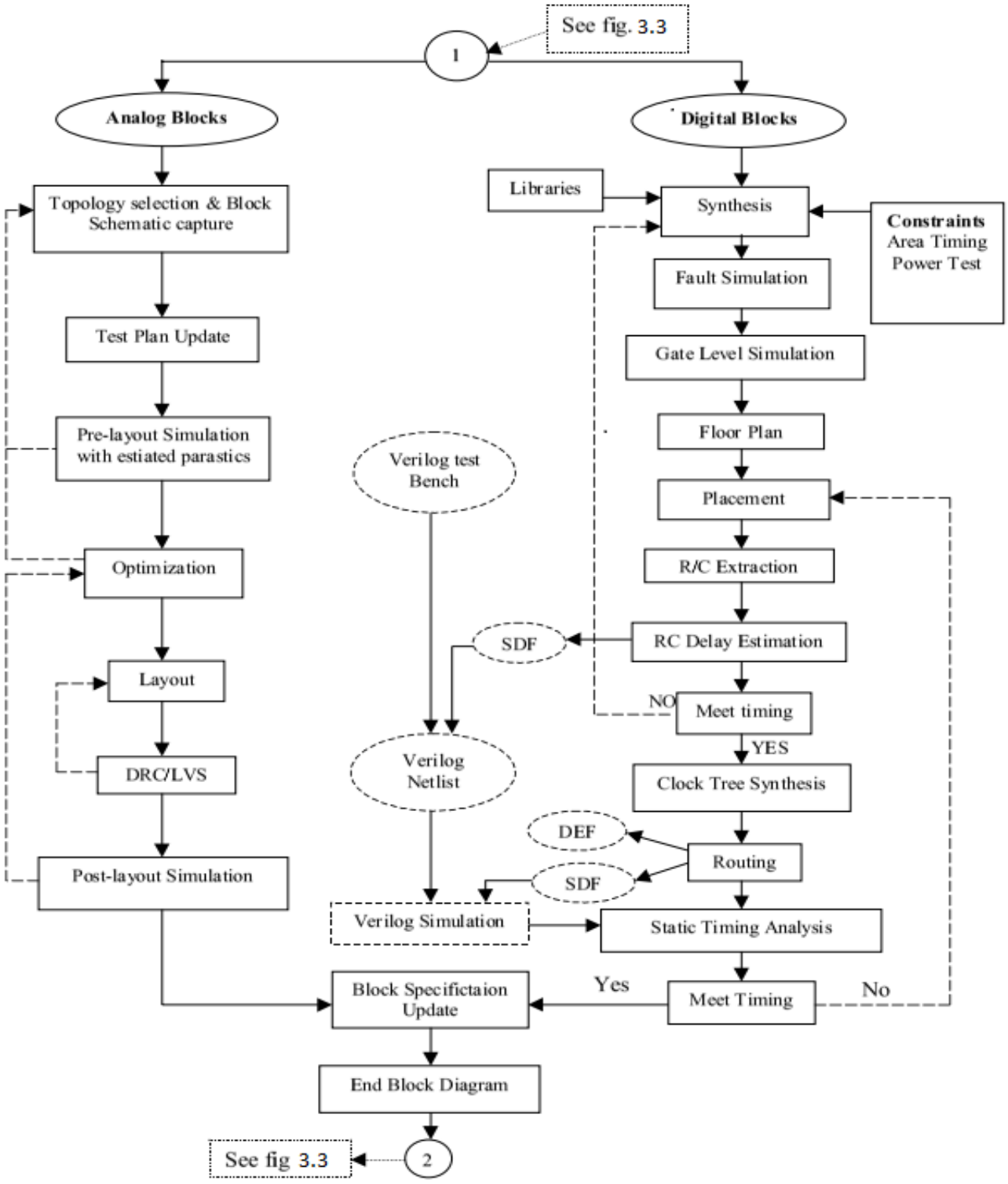


Figure 3.4 Block Design

Digital Block Design

The digital design blocks are mostly the same design flow as in the CMC digital design flow for synthesis. **Logic Synthesis** This step includes the creation of timing budget for digital blocks; scan insertion, technology dependent mapping and optimization. If the design's HDL code is not synthesizable, the RTL code should be modified. The designer should define the design environment, constraints, design rules, technology libraries, and compile strategy. In this thesis, the Design Compiler was used to synthesize HDL description into technology-specific gate-level implementation. After synthesizing the design into a gate-level netlist, timing analysis was done. This process is interactive and might require modifying the original HDL code or the synthesis constraints. A scan chain can be inserted. This process will replace all the flip-flops with their scan-able equivalent. The multiplexed flip-flop scan style is the most commonly used DFT technique.

Gate Level Simulation

The gate-level simulation enables the designer to check the functionality of the structural netlist against the RTL simulation. The test bench used previously for the RTL simulation is used here. Using VHDL-AMS, the designer can verify the functionality of the synthesized block in interaction with analog blocks.

Floor planning

This step involves the creation of rows around the perimeter of the design area for placing the I/O pad cells, core area with spacing the I/O pads, rows or columns, or both in the core area. The designer may also create a power grid prior to placement. This step may also include the placement of cell groups or macroblocks to optimize the connectivity between groups and blocks. The automatic placement tests potential placements for the design and tries to optimize the placement for overlap removal, routing congestion balancing, power balancing, wire length, and timing assurance.

Extraction and Delay Calculation

This step is needed to extract parasitic capacitance and resistance from the layout to calculate and apply delays in static timing analysis and/or full timing simulation using Verilog-XL. The parasitic information is extracted from the layout, and interconnects delays included in the SPF (Standard Parasitic Format) file.

Pre-Clock Tree Synthesis Timing Check

The static timing analysis should be performed using projected parasitic to verify that all timing goals /constraints set after synthesis are still met.

Clock Tree Generation

The designer has to build a clock tree when a large number of cells are clocked by a single driver cell. In this case, we are trying to control the signal skew at the clocked cell's input. It is assumed that the physical library includes timing data in a Timing Library Format (TLF). All modifications to the netlist are saved in a DEF (Design Exchange Format) File for back annotation to the original netlist.

Routing

This step includes global and final routing. Global routing usually consists of a coarse, regular wiring layout based on obstructions resulting from special wiring, clock wiring, and placement. Analyzing the routing congestion map before attempting the final routing is recommended. Final routing creates a detailed regular wiring layout. Post layout timing analysis may be done after routing. It can be done by back annotation of the SDF file.

Post Layout Static Timing Analysis

Using SDF, CAP, and RES files with accurate timing information post-layout simulation and timing, very-fiction can be performed by back annotating the SDF file.

DRC & LVS verification

It is very important to run DRC and LVS on the layout to be sure that the connectivity. The geometry and the spacing are correct, and the layout matches the schematic. This step includes a flat extraction of the layout.

Block Specification Update

After each block is done, it is possible that an update on the block specs is required, and therefore, the respective block documentation will have to be modified. To verify the updated behavioral model and physical layout of each block, the designer needs to perform two simulations from the top level, one with and one without the circuit instantiation of the target block. Other blocks should remain at the behavioral or RTL Level for these simulations. The same test bench created at the partitioning and behavioral modeling stage should be used for this regression simulation.

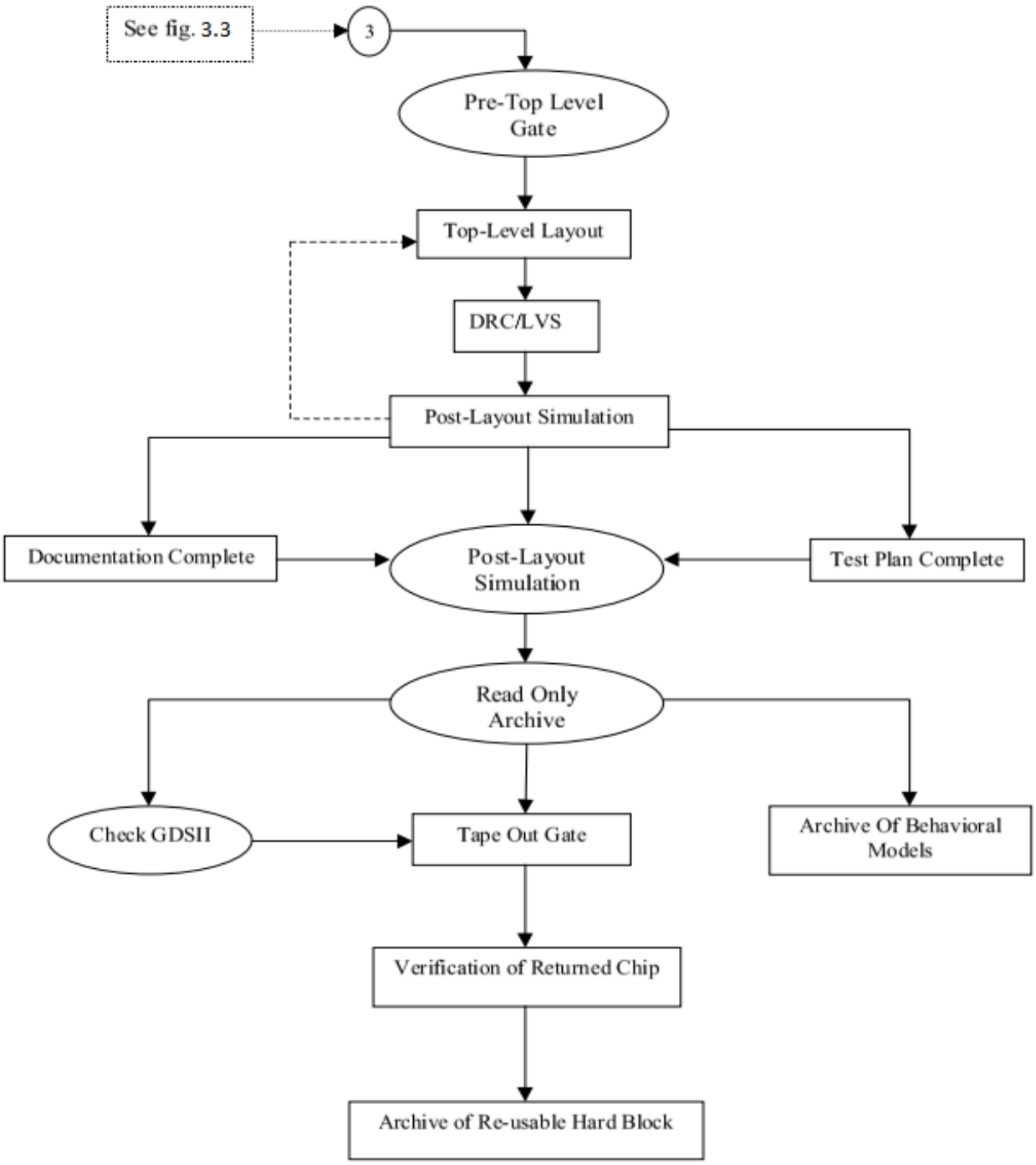


Figure 3.5 Chip Integration

Top Level Layout Design

At this stage of the flow, the layout of all analog and digital blocks is ready, and we have to integrate them. Cadence Virtuoso-XL can be used to perform schematic driven placement for the blocks at the top level based on the top-level schematic created earlier at the partitioning stage. DRC and LVS are performed to

ensure the correctness of the layout. Figure 3.5 shows this part of the design flow. Post layout extraction and simulation are done to verify the top-level parasitic modeling. Any errors revealed by DRC/LVS or any undesirable parasitic revealed by post layout

Post Layout Gate

When the top-level design passes post-layout simulation, a post layout gating should be done. Gating is a peer review of the process to ensure that a post-layout simulation has been performed properly and to check on manufacturing issues such as power/thermal considerations, metal migration issues, power IO to signal IO ratio, ground bounce problem, proper design ID.

Complete Documentation & Test Plan

In parallel, top-level design documentation needs to be completed or updated after successful post-layout simulation, and so does the test plan. At this point, the exact test setup or procedures down to what pin is connected to what instrumentation through what features can be described. All that will converge back to post-layout gating. Complete design documentation and test plan are essential components of passing the post layout gate.

Read only Archive

After passing the post layout gate, a read only library (ROL) should be created to archive the design. Preferably the design data is archived in a standard data formats such as GDSII/DEF/LEF. The design must be frozen at this point, so the right version of the design can be used for debugging later on. Archiving designs using a consistent format, style, directory structures makes re-use easier. For re-use purposes, it is even more important to archive the technology-independent behavioral models than the physical data files. The behavioral models must be properly documented and stored. Before sending out the GDSII file for fabrication, it is desirable to read the file back into Cadence to perform an LVS against the original layout to ensure there are no translation problems occurred.

Tape Out

The GDSII file can then be sent out for fabrication.

CHAPTER: 4

ANALOG AND MIXED SIGNAL VERIFICATION

Today there are various efficient, reusable and reliable functional verification methodologies available for Digital Design/SOC's. Verification done using these methodologies ensures 99.99% functional correctness of Digital Design, but same does not hold true when it comes to Analog/Mixed Signal Design/ SOC's. Now due to increase in Analog Mixed Signal SOC's/chips, there is a potential need for methodology or flow to provide similar confidence on functional verification as seen for Digital Design/ SOC's. System Verilog was started to merge a number of disjoint verification languages such as Vera and *e* that were built as a layer on top of Verilog and VHDL. Each of these languages had their own proprietary methodologies (RVM and erm) that provided a re-useable framework to construct, configure, and execute tests. Once System Verilog became established, it needed its own methodology and Mentor created the AVM(Advanced Verification Methodology) in 2006 that was derived from concepts in system. Synopsys converted their Vera-based RVM(Reuse Verification Methodology) library to System Verilog and called it VMM(Verification Methodology Manual), but did not make it publicly available. Mentor and Cadence joined together and created the OVM (Open Verification Methodology) in 2008, which was the merging of the existing AVM with concepts from *a*rm. Finally by 2011, Mentor, Cadence, and Synopsys joined together through Accellera and created the UVM (Universal Verification Methodology). The Register Abstraction Layer was derived from VMM.

Some flows or methodologies being used to verify Analog / Mixed Signal Designs are mentioned below.

Method 1: Using low level non-functional behavioural model.

- Digital verification: - Using current Standard verification methodologies.
- Analog verification: - Verification using circuit simulators.
- Mixed/Signal Verification: - Connectivity testing between Digital and Analog design using very low level Analog behavioural model.

Method 2: Using analog functional behavioural model developed in Verilog, VHDL or Verilog AMS language.

- Digital Verification: - Using current standard verification methodologies.
- Analog Verification: - Using spice/fast spice analog circuit simulator. Analog functional verification done using behavioural model developed in Verilog/Verilog AMS using Analog Mixed Signal simulator.
- Mixed/Signal verification: - Verification using behavioural analog model.

This work will discuss UVM, currently in use. Also which will help to achieve complete functional verification for Analog Mixed Signal Design/SoC's? It also provides good confidence about functional verification and correctness of Analog Mixed Signal design.

4 .1 METHODOLOGY:

Today there are various methodologies to verify Analog Mixed Signal Verification. This frequently results in re-spins of Analog Mixed Signal Design/SoCs for functional errors. Hence many companies lose time to market for Mixed Signal chips resulting into non profitable venture instead of profitable venture. Analog Mixed Signal Verification Methodology (AMSVM) is basically divided into 4 different phases. Each Verification phase targets specific areas of Analog Mixed Signal Design Flow as shown in Figure 4.1[17]**.**

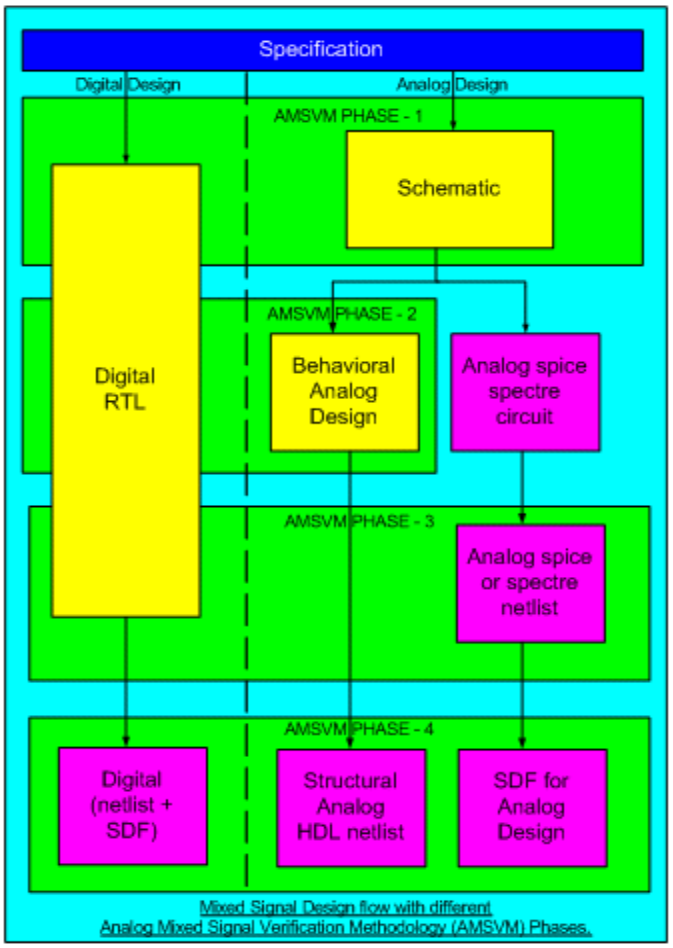
AMSVM PHASE1: Formal Verification

This AMSVM phase targets all connectivity, combinational circuit region of design. This phase also targets connectivity testing for all voltage and current bias from various bias generators to various cells in design. It is used to verify all connectivity bugs introduced during generation of schematic or behavioural model. Verification in this phase can be done using following two different methods:-

Method1: Using Formal Verifier Tool: Create PSL or SVA assertions based on Specification. This formal check targets all connectivity and combinational circuit in design. This method does not require any test case or verification environment development.

Method2: In this method testing is done as part of functional simulation. Verification is divided into two parts.

- Design checks: These design checks verify all bias voltage and bias current values for each library cell model.
- Verification assertion to check connectivity and combinational logic.



Mixed Signal Design flow with different Analog Mixed Signal Verification Methodology (AMSVM) Phases.

This method requires a functional environment to generate various stimuli for design. For design checks describe in Method2 special stimulus generator is required to transmit a signature value on input voltage and bias pin, which will be checked by design checks if behavioral model is in Verilog or VHDL.

Also this phase validates that schematic and behavioral model are structurally equivalent as per specification.

This phase helps identifying below errors in design.

- Any assumptions made during design of schematics which are violating specifications.
- Any combinational or connectivity issues in schematic design in early stage of verification. Combinational logic includes all states or values of encoder, decoder and MUX used in design.

AMSVM PHASE -2: Functional Verification.

This phase targets functional feature verification of Analog Mixed Signal design using Digital RTL and Behavioural Analog Design. Standard verification methodologies like UVM, OVM or VMM can be used to verify functional features for Analog Mixed Signal Design. Tool used in this phase of verification depends on Analog behavioural model. Analog behavioural model used here can be designed in Verilog, VHDL or Verilog AMS.

This phase verifies various aspect of Analog Mixed signal design.

- Functionality of blocks in digital and analog design.
- Various interfaces/protocols between digital and analog design.

Based on methodology used during this phase, functional correctness of design can be measured using similar parameters used for coverage driven verification(CDV).This functional correctness measurement metrics will be solely dependent on approach used for verification. It can be purely directed verification or coverage driven random verification.

At end of this phase of verification, we can ensure that all connectivity and functional features present in Analog Mixed Signal design (model) are verified as per specification.

Analog circuit simulation using “Spectre Solver” or “Fast Spice Solver” would be done in parallel to this phase of verification. There are various ways for analog circuit simulation (not covered here as our target is to achieve functional correctness using this methodology). Analog circuit simulation mainly targets optimization of circuit parameters and various performance parameters like power, current, frequency, speed, yield, etc.

AMSVM PHASE -3: Logic Equivalence Check

Above 2 phases verify that Analog Mixed Signal Design is functionally correct, but they do not ensure that analog spice/spectre netlist used for analog circuit simulation is functionally equivalent to Analog behavioral model. Also they do not check that any assumptions made during circuit simulation of mixed signal design or generation of Analog behavioral model is not creating any lock up conditions for Analog Mixed signal Design.

During this phase Analog Behavioral model is replaced by Analog spice/spectre netlist in verification environment used during AMSVM Phase 2. Few small tests exercising basic functional paths of design are run in this environment. Here pass fail criteria for test can be determined based on following two methods:-

- Self-checking directed tests.
- Post processing of simulation output results.

Both HDL and Fast Spice simulators are needed to run mixed signal simulation in this phase. For example Cadence: IUS and Virtuoso (ultraism).

Test case used during this simulation should be short as mixed signal simulation is very slow. Spice/spectre netlist should be generated with minimum required parameters or depth of extraction to reduce the simulation time. Using same environment and test case here gives a better confidence to design and verification team about functional correctness and stability of the design. It also ensures logic equivalence check between analog spice netlist used for analog circuit simulation and analog behavioural model. This stage ensures that functionally verified design is really used for Physical design layout and floor plan. This phase does same kind of verification as logic equivalence check (LEC) done between Digital RTL and Digital Gate Net list.

Disadvantage: It depends on test list determined by design and verification team, whereas in case of Digital, it is automated testing using EDA tool. So quality of this verification solely depends on completeness of test list.

AMSVM PHASE – 4:

During this phase gate level simulation with SDF is run on post routed Digital and Analog netlist. A common question that arises in everyone’s mind is why we need gate level simulation with SDF on post routed netlist, as we already do LEC for digital design and STA for both analog and digital design so this

phase normally looks redundant.If by mistake designer has placed a timing exception like false path and multi-cycle path then it won't be caught. Normally we have different designers for Analog and Digital team, so there may be chances that some assumption made for analog to digital or digital to analog handoff for timing may be wrong. This gate level simulation with SDF on post routed netlist is to counter check for STA and to catch mistakes the designer has made on placing any timing exceptions.Below Table 1 shows tools required for Analog Mixed Signal Verification Methodology (AMSVM)

	Formal Verifier Tool	Functional Verification Tool	AMS Simulator	Analog Circuit Simulators (spice, spectre , fast spice)
AMSVM Phase 1- Method 1	YES	YES	NO	NO
AMSVM Phase 1- Method 2	NO	YES	NO	NO
AMSVM Phase 2 – Analog Behavioral model in Verilog or VHDL	NO	YES	NO	NO
AMSVM Phase 2 – Analog Behavioral model in Verilog/VHDL AMS	NO	YES	YES	NO
AMSVM PHASE 3	NO	YES	NO	YES
AMSVM Phase 4 – Analog Behavioral model in Verilog or VHDL	NO	YES	NO	NO
AMSVM Phase 4 – Analog Behavioral model in Verilog/VHDL AMS	NO	YES	YES	NO

Table 4.1: List of tools required for different AMSVM phases.

During any Analog Mixed Signal project we required HDL Simulator to simulate Digital portion of design and Analog Circuit simulator for Analog design. If Method 2 is AMSVM Phase- 1 and Analog Behavioral model is designed using HDL languages like Verilog /VHDL. There is no additional cost in project for tools. There may be a question arising when same verification results can be achieved using both methods shown in AMSVM Phase -1, then why 2 methods are described. It is true that results achieved are same, but with Method1 it takes less verification effort/time to achieve those results. So depending on project schedule and cost, it can be decided whether to use Method1 or Method2.

4.2 UVM Verification Environment

A typical UVM based verification environment contains three main building blocks :UVM component (UVC), UVM env (environment) and UVM test. Fig. 4.2 shows this UVM verification architecture.

Universal verification methodology was released based on OVM and VMM methodologies by Verification Intellectual Property Technical Subcommittee (VIP TSC).

UVM test UVM test block is derived from UVM component base class and it is used as the base class for user-defined test classes. User can define multiple test classes and

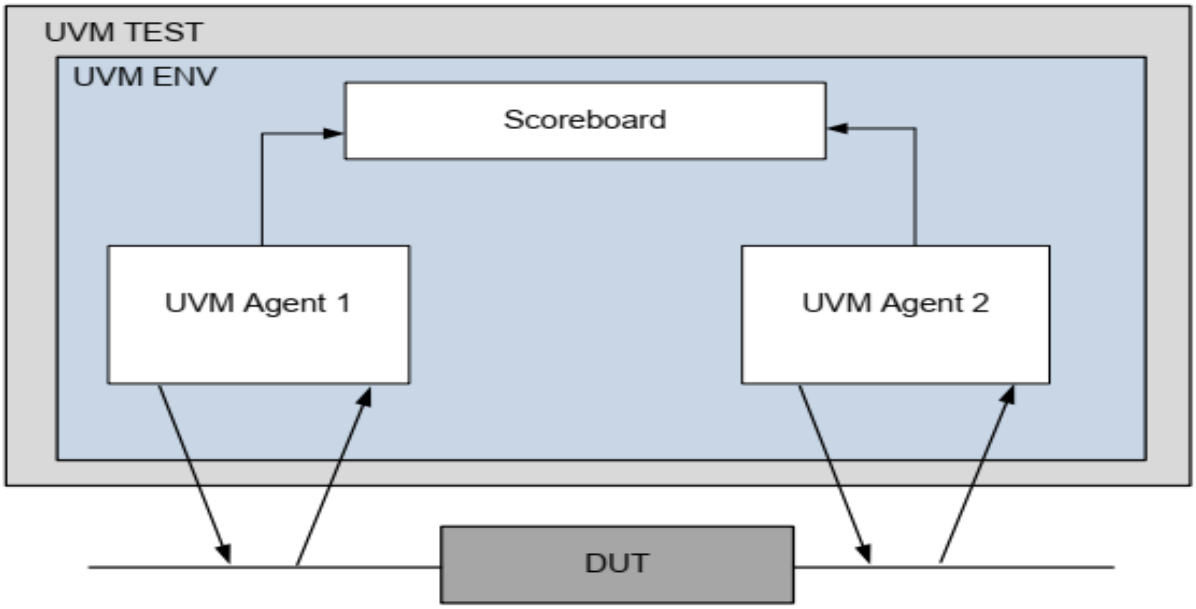


Figure 4.2: Typical UVM environment

Randomly select them via command line. In top-level module, the global run test() task is invoked together with DUT instantiation and interface definition (DUT and test bench interface) and the test which is defined in UVM TESTNAME via simulator command line starts the simulation.

UVM env : The UVM env class is instantiated in UVM test and is derived from UVM component. As it is shown in Fig. 4.2 this block is used in order to encapsulate and configure agents (also known as UVM verification components (uvcs)). It includes one or more agents and it has capability to define them as an active agent or passive agent depending on configuration. It is environment class main task to generate meaningful random stimulus, sampling and monitoring DUT's result, validating the result and collecting coverage .

UVM agent A UVM agent typically includes three main blocks: Driver, Sequencer and Monitor. All these sub blocks communicate using transaction level modeling (TLM) connections. It is through System Verilog virtual interface that agent communicates with DUT. Another agent's property is configuration which

indicates either it is an active or passive agent. User can define any other control parameters using agent configuration. Active and passive mode architectures of UVM agent are shown in Fig. 4.4.

Scoreboard is built on top of the monitor and performs analysis on the data stream received from monitor. Scoreboard has a significant role in self-checking verification environments. Functional behavior of the design under test is analyzed by this component. Scoreboard at least contains two analysis imports including an import to receive input data stream to the DUT, and another import to convey monitored

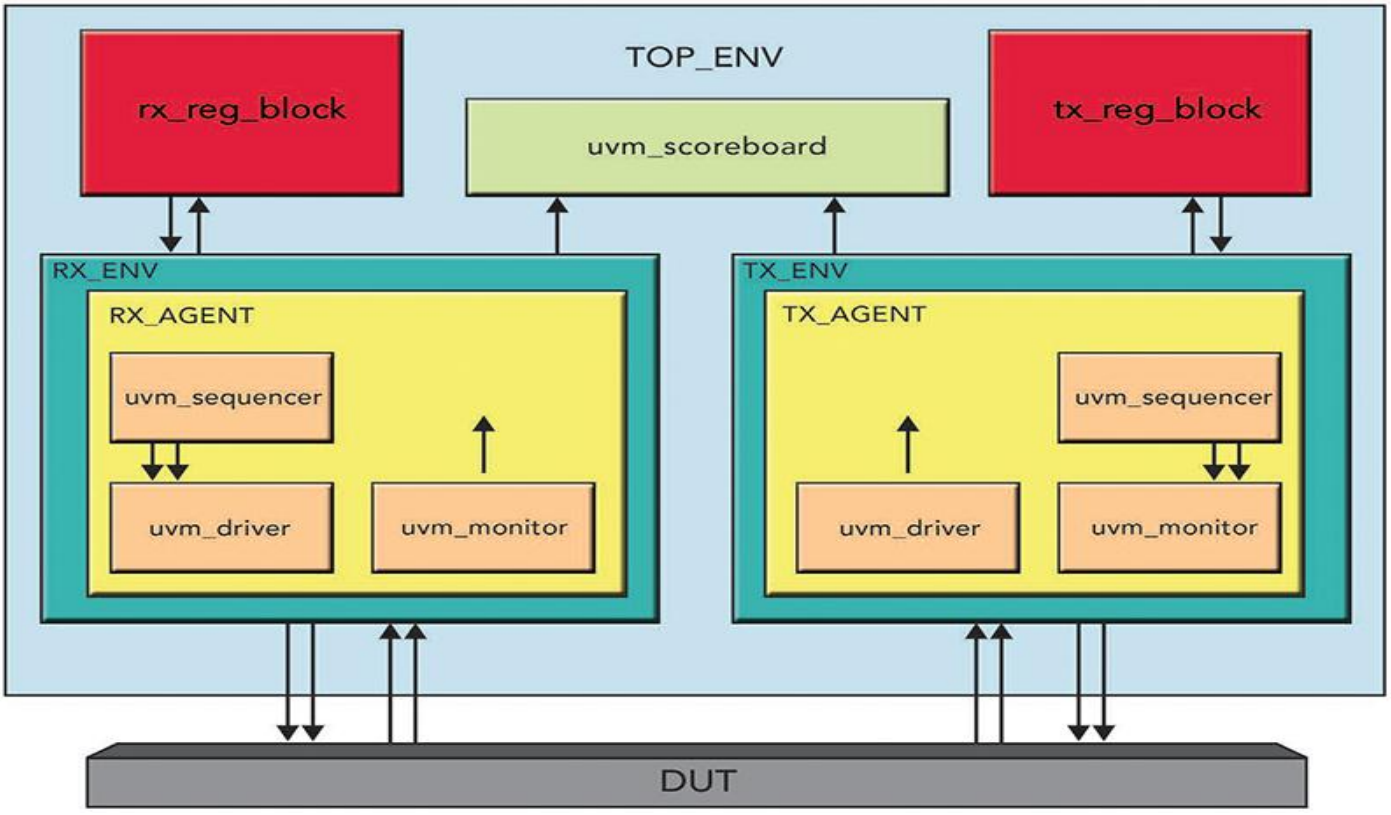


Figure 4.3: UVM environment setup

Activities on the DUT's output pins (see Fig. 4.4). Scoreboard component contains a golden model through which performs a comparison between actual results of DUT (sampled by monitor) and expected results of DUT (Golden model's output in response to the same input vector to the DUT)

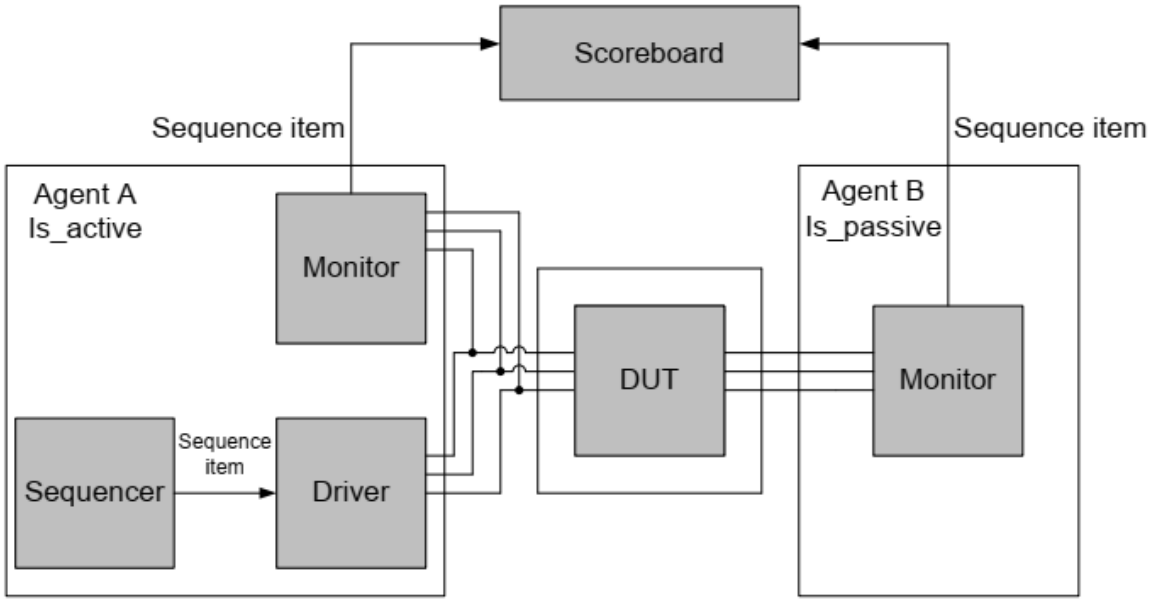


Figure 4.4: UVM agent: in active and passive mode

A typical active agent contains all three subcomponents and in one hand generates random stimulus and drives the DUT pins through driver, and on the other hand samples DUT's input pins (drived by the agent's driver) via monitor. This monitor converts data stream back to transaction level and send it out to other analysis components. A typical passive agent, without driving the pins of a DUT only performs monitoring of the generated result of a sample DUT. This means that only the monitor component is instantiated in a passive mode agent. Agent in this mode is used for coverage collection and checking operations. In Fig. 4.4, agent A is depicted with the is active as a configuration property, while the same flag in agent B is set to is passive. As it is depicted the same passed sequence item by the sequencer to the driver, is generated again by the monitor later. In this way, the DUT input pins are driven by the driver component and are sampled by the monitor component of an active agent. This is a reliable practice to ensure that the incoming sequence item to the driver is properly converted to signal level activities and DUT is driven with desired stimuli.

4.2.1 Transaction Level Modeling

In order to increase verification productivity and manage complex system-on-chips verification, higher abstraction level of the design is strongly requested. In an abstraction level higher than RTL, interfaces are defined in terms of transactions. "In transaction-level-model (TLM), the details of communication among computation components are separated from the details of computation components." While the DUT communicates with verification environment at signal-level, it has been proved that it is necessary to handle

Most of verification tasks such as stimulus generation or coverage collection at transaction level [14]. In this abstraction level, complicated data transferring between units is managed at a higher level. Many UVM components communicate to each other using TLM. In this way, components are isolated from other component's modifications. In order to handle the communication between DUT and verification environment which is at signal-level, it is required to provide a layer in the testbench in which UVM components are able to convert transaction-level activity to signal-level activity and vice versa. Moreover, it is through TLM that encapsulated, reusable verification components can be obtained and whereby verification environment construction is facilitated. In UVM base class libraries, a base class called UVM sequence item for all user-defined transactions is declared. This class is extended from UVM transaction and ultimately is inherited from UVM object. TLM communication in basic level is depicted in Fig. 4.5. Transaction is generated by the producer which holds a port (square). Transaction is consumed by the consumer which holds an export (circle). A basic definition of a transaction states that the producer puts a transaction and consumer gets it.

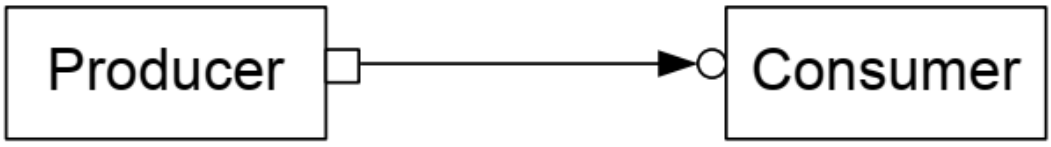


Figure 4.5: Simple producer consumer [14]

Transaction level communication between two components might happen in two different situations as it is shown in Fig. 4.6. If data and control flow are in the same direction, it is producer which puts transaction into consumer (a). While in other situation, when data and control flow directions disagree, it is the consumer which requests transaction from the producer via get port (b). However, it is export (circle), which implements the transaction in both situations.

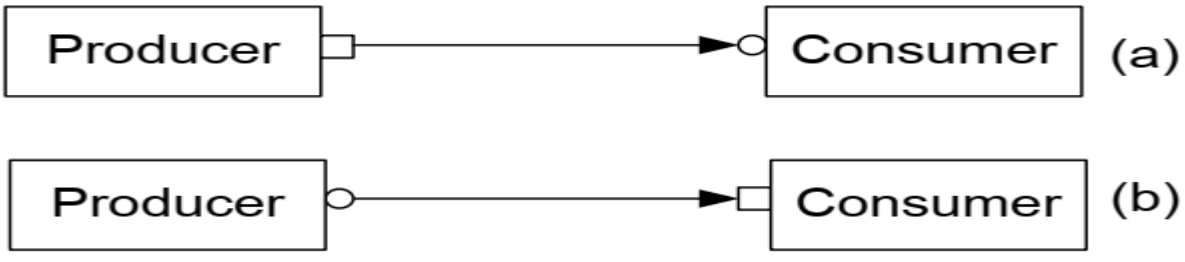


Figure 4.6: Put versus Get

Transaction-level models are utilized in UVM environment efficiently. A UVM based verification environment holds four basic transaction-level components.

Sequencer Sequencer creates transaction-level traffic and pass it to the driver. Extending from UVM sequencer base class, the sequencer is able to communicate with driver in a parameterized way. These parameters are of same type and are called request and response. User can specify a user-defined transaction type to these parameters. Otherwise, they have the default UVM sequence type type. Upon a request from driver, sequencer selects a sequence from listed sequences and passes it to the driver. A sequence is a group of transactions. Multiple sequences together perform a planned scenario. Basically, sequencer's arbiter role controls the flow of these sequences from multiple generators. Through sequencer's seq item export data items (transactions) are sent to driver's seq item port. In Fig. 4.7 basic interactions between a driver and a sequencer to shape a transaction model is shown. In this model the following tasks are executed in depicted order. First, a transaction is created in sequence using UVM create (req) method. By calling wait for grant method, sequencer blocks any activity until the driver asks for the next item through calling get next item method. Only after driver's request, the transaction can be randomized optionally. Next step is to send the transaction via UVM send (req). The retrieved transaction by the driver is consumed and converted to signal level according to the application specification. The driver raises the item done (rsp) flag to send a response back to the sequencer (this is an optional step). Finally, wait for item done () method in sequence, which is a blocking method, gets unblocked.

Driver component is responsible for driving DUT's pins. As it was mentioned in this section, driver is a component which is located in the layer in which transaction level activities are converted to signal-level activities. In other words, driver retrieves transactions from sequencer and translates them to a signal value. This signal drives DUT's pins via System Verilog virtual interface. In UVM base class libraries, the UVM driver is defined as a base class for user defined drivers. When driver's connectivity to sequencer via seq item port and to the DUT via virtual interface are implemented, it can obtain the next available data item from sequencer. Driver can also use another port called rsp port to send a response back to the sequencer. **Monitor** DUT's output pins activity is monitored and sampled by monitor component in order to determine if it behaves correctly. Monitor's performance is in opposition to the driver. In Fig. 4.8 the location of monitor and driver in a mediator layer between DUT and verification environment is depicted. In order to verify DUT's behavior, It is monitor's duty to extract pin-level activities of the DUT and translate them into

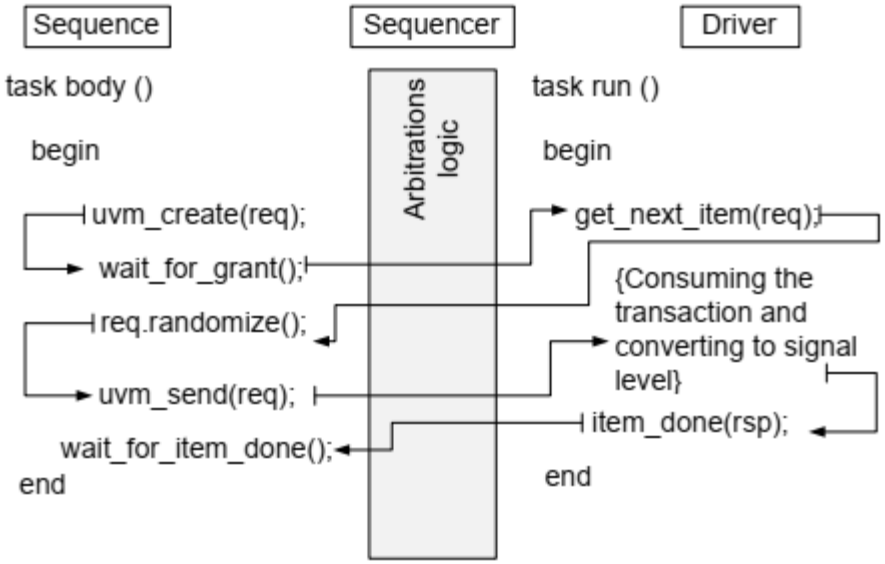


Figure 4.7: Sequence and Driver communication

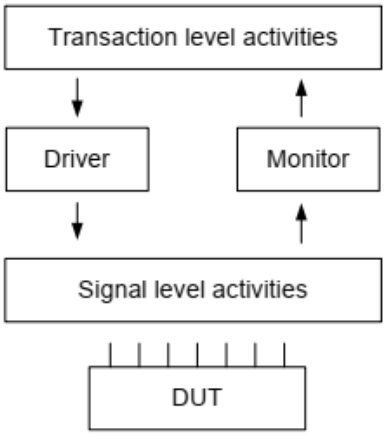


Figure 4.8 : Driver and Monitor in mediate layer

Transaction level and send it to other components for analysis. Monitor's main tasks can be count as basic monitoring and coverage collection. DUT's functional behaviour is verified by other high level components like Scoreboard. Virtual interface, is the way through which monitor collects data from information bus. Collected data might be used in coverage collection or exported to analyser components via UVM analysis port. The major difference is that input/get approach, no matter if it is put or get, presence of a component with export connection is necessary to implement the transaction. While, in TLM analysis communication regardless of target's presence it is important to generate transactions and send it through analysis port. In other words, this port is not dependent on export connection. The transaction simply returns in the absence of any export connection. A base class named UVM monitor, preserves all basic features of user-defined

monitors. Monitor checks the observed data format and generates UVM reports, accordingly. A print function is defined in UVM transactions which can be called by monitors in order to print out the transaction's content.

In order to provide higher levels of productivity, it is of significant importance to utilize various simulation tools during all steps of design development. For an analog circuit designer to gain better productivity it is crucial to model the circuit in a higher level of abstraction and verify its functionality. This is how designers can simply detect some functional defects prior to devoting time to design to the transistor level. Moreover in chip level verification when the design is going to be checked in context of the whole system and verification aim is more to monitor system level behavior or interconnectivity between blocks, detailed results of SPICE models are not requested. As it is discussed in details in , a top-down approach in analog and mixed-

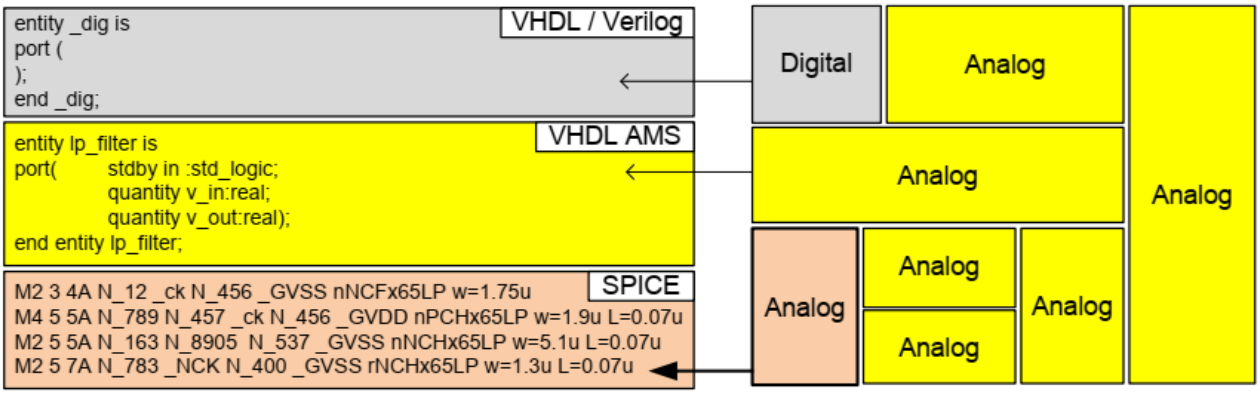


Figure 4.9: Top down approach in analog centric designs

Signal to conclude, since analog and mixed-signal integrated circuit simulation and verification became a challenging task, using only SPICE like simulators is not anymore a rational choice. While digital certification is heavily automated, still analog verification progress cannot satisfy the demanded requirements. Moving forward from chips with thousands of transistors to the chips with millions of transistor. In order to enrich analog and mixed-signal verification quality as well, using abstract models of analog circuits is inevitable. These abstract models are utilized in different levels and design steps. The time-to-market for leading semiconductor companies has been highly decreased taking advantage of abstract models in either bottom-up or top-down manner. It is significantly important to slightly abandon the detailed results of SPICE level simulations in some levels of verification process and use a concise butcomprehensive simulation model instead. In this manner, beneficial digital like strategies are employable in analog circuit verification. One of the prominent advancements of digital verification is transaction level modeling. The

aim of this work is to investigate on a way to utilize the concept of transaction in analog circuit's verification.

4.2.2 Mixed signal Verification Facilitation

SoC level verification of mixed-signal ICs has been getting more attention in last years. Several approaches has been proposed by researchers in order to enhance the verification process. Since digital verification teams were able to build highly automated test benches, it was time to introduce more digital like and consequently more automated techniques for analog and mixed-signal circuits verification. Although analog verification with SPICE is still a golden standard and cannot be ignored, to achieve better simulation speed, different levels of design abstraction are used to model an analog subsystem. Taking into account SPICE and fast SPICE simulation next level of abstraction, analog behavioral modeling (with Verilog-AMS and VHDL-AMS), improved simulation speed significantly. Furthermore, Real Number Modeling (RNM) and pure digital models are the other approaches to describe the analog sub circuits in higher levels of abstraction and therefore to achieve higher simulation performance. Moreover, another crucial factor to choose an

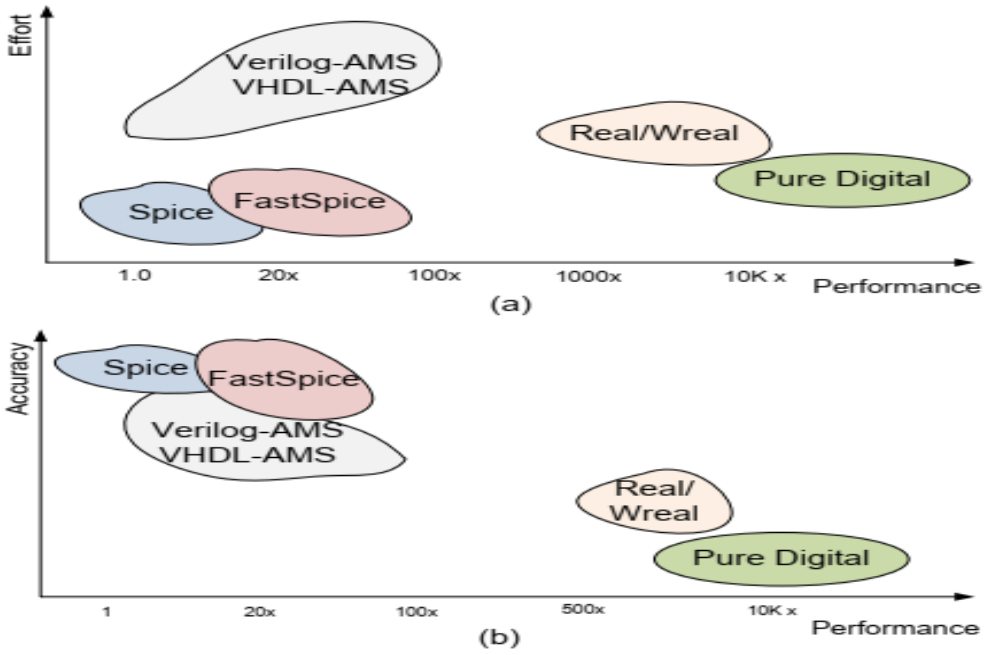


Figure 4.10 performance and accuracy chart

Appropriate abstract model is the required effort to build a simulation environment. To clarify more a comparative chart is shown in Fig. 4.10 The chart shows the distinction between mentioned approaches in terms of required effort for simulation setup in (a) and performance trade off in (b). As depicted in this figure, although RNMs or pure digital models are less accurate models but less effort is required to build a

simulation environment using these models than AMS models. Therefore this is the most apparent advantage of these models in full-chip verification.

CHAPTER 5

THERMAL MONITORING UNIT

5.1 INTRODUTION

As integrated circuits are becoming larger and more complex, thermal constraints are emerging as a dominant performance limiter. It is well known that leakage currents increase exponentially with chip temperature. In a recent report from intel [1], it was reported that every 15ºc increase will cause delay to increase by approximately 10% to 15%. High chip or even high localized die temperatures reduce reliability attributable to electro migration. The relationship between electro migration-induced reliability degradation and temperature is highly nonlinear with even small increases in temperature above a critical threshold causing dramatic reductions in reliability. A power/thermal management block, often termed a power manager, is included on many integrated circuits today to limit the thermally-induced degradation in performance and reliability. For many years, at most one on-chip temperature sensor was used throughout the industry to provided temperature information to the power manager. Unfortunately, performance degradation and even more importantly reliability degradation is localized. If acceptable reliability is to be achieved with the single temperature sensor solutions, a very conservative power management strategy must be adopted to prevent high die temperatures at positions located away from the temperature sensor from reducing reliability In this chapter we will see the temp issue after that the component of the TMU and current comparator.

5.2 Temperature Issues

5.2.1 How heat generated in the circuit?

The temperature of a VLSI chip can be calculated using the following equation

$$T_{CHIP} = T_a + R_0 \frac{P_{TOT}}{A}$$

where *Tchip* is the average chip temperature, *Ta* is the ambient temperature, *PTOT* is the total power consumption, *A* is the area of the chip and $R\theta$ is the equivalent thermal resistance of the substrate, the packaging and the heat sink. As can be seen, circuit temperature is a strong function of IC power density. The heat is generated on the circuit substrate, due to the current flowing through devices and passive

components on the silicon substrate, and at the different levels of metal that provide the interconnections. More precisely:

- The most significant source of heat is the substrate, especially when considering active components. Power dissipated by devices can be obtained from the sum of dynamic, static and short circuit power consumption.
- The secondary source of heat is the set of metal layers, which generates heat by Joule effect. Self-heating is produced due to the electric current circulating through interconnections. Even though this source of heating is smaller, it cannot be neglected, since the different metal layers are separated by silicon oxide, SiO2, which is an insulator that has a worse conductivity than thermal silicon [BM01a].

5.2.2 Power of Device

First we will review how power is consumed by active components in the substrate. There are three sources of power consumption in the semiconductor substrate:

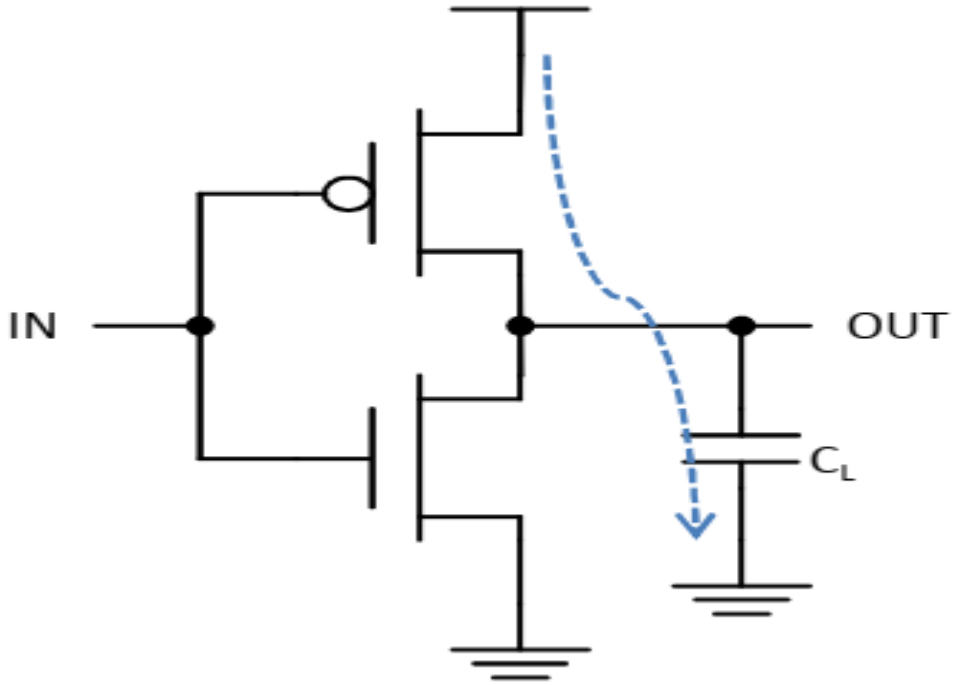


Figure 5.1: Dynamic power in a CMOS inverter.

1. **Dynamic power** due to charging and discharging capacitances in the circuit. Let us take as example a CMOS inverter as shown in Figure 5.1. The power is drawn from the power supply, *VDD*. When the load capacitance *CL* of the inverter is charged at a frequency *f* the energy extracted from the power supply is *CLVDD*^2*f*. However, the energy that is stored in the load capacitance is 1/ 2*CLVDD*^2*f*. Half of the energy is dissipated as heat in the equivalent resistance of the pMOS transistor while charging the load capacitance. The same happens when the load capacitance is discharged; in this case it is the nMOS transistor that dissipates half of that energy. In the case of other conventional CMOS gates, the situation is the same, but considering a pMOS-based pull-up network and the dual nMOS-based pulldown network. More precisely, since most signals in a circuit do not commute at the clock frequency but at a lower rate, an activity factor α is usually included in the dynamic power expression:

$$P(DY N) = \alpha CLVDD^2f$$

The dynamic component is the dominant power in the active mode of a circuit.

2. **Static power (PSTAT IC).** Apart from those logic styles that are characterized by having designed permanent VDD-ground currents, static power is due to the existence of undesired currents in the substrate produced when biasing the transistors. These currents are mainly due to leakage (this is the case of the sub threshold current, Isub), reverse bias of parasitic diodes present in all CMOS structures, reverse, which is smaller and gate tunnelling current, Igate. Thus, static power can be computed as:

$$PSTAT IC = (Isub + Ireverse + Igate)VDD$$

3. **Short-circuit power (PSC).** Since transitions between high and low (and vice versa) are not instantaneous, there is a short time slot in which transistors of the pullup and pull-down networks are simultaneously on, driving the output capacitance at the same time. A current peak flows between power and ground. The power.

Consumption is higher when the edges of the input signals to the CMOS gates are more pronounced. Short-circuit power depends on the duration of the transitions, the size of the pull-up and pull-down transistors and size of the load capacitance.

5.2.3 Power of Interconnection

For the case of interconnections, the power dissipated can be calculated from the general expression of power: $P = I \times V = I^2R$. Power dissipated can be found with Joule's Law where the resistance of the interconnection is given by

$$R = \rho \frac{L}{A}$$

Where ρ is the resistivity (Omhs/m), L is the length of the interconnection material and A is the cross section area of the interconnection material.

5.3 Temperature effect on chip

Once we know from where heat is generated we will briefly overview the effects that high temperatures and gradients have on the key parameters that characterize an IC. There are many circuit parameters that are influenced by temperature. Furthermore, the implications among parameters and temperature are very complex. To deal with these intricate influences we will start by studying the effects of temperature at device level. Next, the impact of temperature on the interconnections will be studied. Finally, the main temperature-related degradation effects will be concisely presented.

Thermal effects on devices

The most basic element of an integrated circuit is the device. We will begin by studying the impact of temperature on two basic device parameters, the carrier mobility in silicon and the threshold voltage. This will provide us with equations relating temperature with the drain current of the CMOS transistor. Finally we will see how these relationships influence the global static power consumption of a system.

Effects on the mobility of carriers

The mobility of the carriers has a nonlinear dependence on temperature rather complicated. The mobility is influenced by two phenomena:

- Phonon scattering, due to random thermal vibrations of the semiconductor atoms, which increases with the temperature.
- Columb scattering, due to the doping atoms repelling (attracting) electrons (holes) that appear in their vicinity.

When the doping concentration is very high columb scattering dominates and mobility increases with temperature. However, at lower doping concentrations phonon scattering dominates, thus mobility decreases with temperature. These implications would lead us to a complex mobility expression dependent both on temperature and carriers concentration [Liu01]. However, most device models only consider the second

phenomenon through an expression that models the mobility dependency on temperature similar to [Liu01]:

$$\mu(T_{dev}) = \mu(T_{nom})(\frac{T_{dev} + 27315}{T_{nom} + 27315})^K$$

where $\mu(Tnom)$ is the mobility at a nominal temperature and K is a temperature coefficient between -2.0 and -1.5. As can be seen, mobility of carriers can be considered as an inverse function of the absolute temperature.

Effects on the threshold voltage

Reducing the threshold voltage produces significant effects on many transistor parameters that directly depend on it. For instance, there is a direct effect in reducing the circuit robustness against noise, because by reducing the threshold voltage the barrier that can activate the transistor is reduced.

Effects on the drain current

The basic operation of a long channel MOS device is described by the following equations.In the saturation region:

$$I_D = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{GS} - V_{THn})^2$$

And the linear reason

$$I_D = \mu_n C_{ox}\frac{W}{L}((V_{GS} - V_{THn})V_{DS} - \frac{V_{DS}}{2})^2$$

where Cox is the gate capacitance per unit area, µn is the mobility of electrons near the silicon surface, W and L are the gate width and length, VGS and VDS denote the transistor gate-source and drain-source voltages respectively, and VT Hn is the threshold voltage. These equations are valid for nMOS devices (for pMOS µp and VT Hp would be used instead). As seen above, there are two temperature-dependent parameters in 5.2.2. Harmful effects due to temperature in VLSI chips 13 these equations, µn(T) and VT Hn(T). As the temperature increases, the mobility of carriers decreases and the threshold voltage also decreases. Thus, there will be an area of operation insensitive to a temperature variation. However, for high currents, reducing the mobility prevails against threshold voltage reduction and a net reduction of ID is observed. Therefore the most important impact is a reduction on the amount of current a transistor can

deliver. This will influence, for example, the charging time of the interconnections that have less power available. For a variation of 40oC the delay in a logic gate can increase a 4% (130 nm)

Effects on the sub threshold drain current

Sub threshold leakage current, Isub, is the drain-source current of a transistor operating in the weak inversion region. This current is due to the diffusion of minority carriers in the channel and is much larger than other leakage currents present in the silicon substrate. It can be expressed by the following equation:

$$I_{sub}(T) = I_{S0}(T) e^{\frac{V_{GS} - V_{TH}(T)}{nkT/q}} (1 - e^{-\frac{V_{DS}}{kT/q}})$$

In this equation kT=q is the thermal voltage (k is the Boltzmann constant and q is the electron charge) and n is the transistor sub threshold swing coefficient. IS0(T) is a technology parameter, dependent on the temperature given by:

$$I_{S0}(T) = \mu(T_{nom}) C_{ox} \frac{W}{L} e^{1.8} (kT/q)^2$$

As equation shows, Isub is a function of temperature, threshold voltage, device size and technological parameters that mostly depend on the threshold voltage. Subthreshold leakage current increases rapidly with temperature.

Effects on static power dissipation

Static power is becoming more and more significant in nanometer technologies. Since static power depends on *Isub* that increases exponentially with temperature, this behaviour can be also observed in static power, with the corresponding increase in heat dissipation. If left uncontrolled, the IC can experiment a positive feedback loop very destructive known as \thermal runaway".

Thermal Effects on Interconnects

Technology scaling causes increased harming effects in interconnects related to temperature. These effects can be summarized as [BM01a]:

- Increasing current density, with the corresponding increase in self-heating.

- Increased thermal coupling.
- Worsen of thermal conductivity because of the use of low-k dielectric materials (lower thermal conductivity than SiO2)

Additionally, Electrostatic Discharge (ESD) protection is potentially reduced in a number of I/O structures because thermally induced open-circuit metal failures appear under short duration high peak currents.

Effects on signal integrity in the power/ground rails

Voltage droops in power and ground rails biasing a logical block may be due to different reasons. On the one hand it can be due to currents generated by the block itself, which produce a voltage difference between the external pin and the block terminal. On the other hand, there may be transitory falls due to transient currents caused by other logic blocks. Regardless of where the variation of current comes from, the voltage drop is more pronounced when the resistivity of the power and ground rails increases. Here temperature comes into play, since an increase in temperature decreases the mobility of the carriers; it causes resistance to increase which concludes with a rise in voltage drop. Electro migration, which is temperature dependent, can also cause problems in the power and ground rails. Apart from increasing resistivity, with the consequent increase in the voltage droop, it can cause more critical errors such as open or short circuits with adjacent tracks.

5.4 Applications

- Temperature Sensitive Storage
- IoT Devices
- Industrial
- Embedded applications
- Computing
- Handheld Gaming
- Portable Electronics
- High Performance SoC

5.5 Component of TMU

The main purpose of the TMU is to autonomously monitors and reports the temperature from one or more remote temperature measurement sites located on chip, when a potential dangerous operating

temperature is about to be reached. This may be to protect a circuit from overheating and causing permanent damage. Help to prevent from operating temperature which may affect reliability. This TMU has a certain level (normal/critical/immediate) threshold and interrupt scheme. This will allow software to take preventative actions at a first temperature threshold and additional measurements at a second critical threshold. For high temperatures this may include downgrading performance of a function to reduce power consumption (preventative) and reducing voltage/frequency to a function.

Working of TMU:

The picture shows the part of analog subsystem on common chassis for monitoring temperature. The DUT subject TMU has 1 Central Unit, several Remote Sites but we are taking under consideration only 3 for the verification, and Thermal Diode for temperature reference. Digital part of the IP or output is connected with Fault correction and control unit (FCCU) and Interrupt controller (INTC). One Time Programmable application software will enable and configure the TMU. This approach requires several temperature sensors judiciously placed at critical points in each core and as the number of cores continues to grow, this approach will require a growing number of on-chip temperature sensors that could be in the hundreds for processors with a large number of cores.

FCCU is a programmable unit that monitors the integrity status of the microcontroller and provides flexible safe state control by collecting errors and leading the device in a controlled way to a safe state when a failure is present. It Generate reset for TMU if FCCU reports threshold violations and use this reset to disable TMU.INTC is a unit that significantly monitors the interrupt generated by the system .

TMU has 4 modes: Calibration, Monitoring, Circuit Test and Disable/Low Power/

Calibration mode : in this mode the system is being calibrated one time with respect to default data , so after that if any error is present between actual and measured temp , will decreases .

Monitoring Mode, threshold violations will be reported to FCCU and interrupt request will be raised for rate, Immediate, average, and critical average threshold interrupts.

Circuit Test Mode, Current/Voltage measurements from the TPA pin under controlled settings will be allowed.

Disable mode: When a dangerous threshold is reached then functionalities that define the reset mode .

Till now we have an idea that what is the basic function and how thee TMU works. Now we will discuss the internal parts of the particular TMU inside the analog block

5.6 Architecture

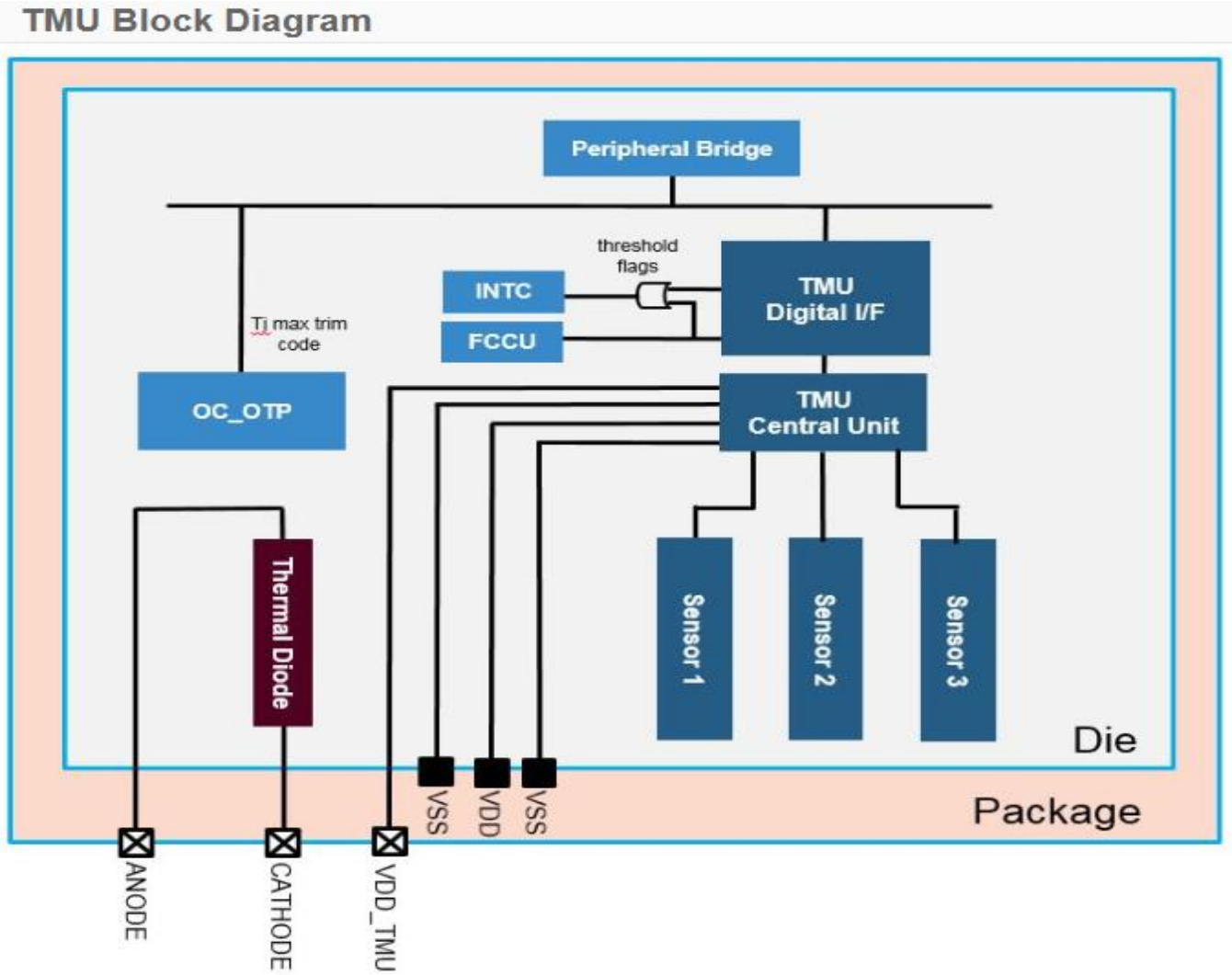


Figure : 5.2 TMU Block diagram

Remote Sensor

Essentially all temperature sensors generate an electrical output signal that carries temperature information. This output signal is typically a voltage, a current, or a period. Usually the temperature sensors are designed to achieve a relationship between the output signal and the chip temperature that is nearly linear. The dominant temperature dependent properties that have been used for many years for on-chip temperature

sensors in both industry and the research community are the temperature dependent characteristics of pn junctions. The pn junction operating is modeled in

$$I(T) = I_s e^{\frac{V_{BE}}{nVT}}$$

where Is is the reverse saturation current, Vd is the junction voltage, VT=kT/q is the thermal voltage (approximately 26mV at 300K) and n is the empirical emission coefficient (1<n<2). The reverse saturation current is itself highly and nonlinearly temperature dependent as well and can be expressed as

$$\mathbf{I_S = I_{SX} A_J \left[T^m e^{\frac{-V_{G0}}{V_T}} \right]}$$

Where AJ is the area of the junction, ISX and m are process-dependent constants and VG0 s the band gap voltage of silicon. The parameters ISX, AJ, m, and VG0 are all independent of temperature If a circuit is designed that forces the ratio of the currents of two diodes to be a constant, it is well-known that the nonlinear temperature dependence of IS vanishes in the difference in the two diode voltages thus resulting in a diode voltage difference that is proportional to T. This concept is widely used to build pn junction based temperature sensors. One simple pn junction-based temperature sensor is shown in Figure 5.3. In this structure, ΔVBE is used as the differential output voltage. It's easy to show that the differential output voltage for this circuit can be written as

$$\Delta V_{BE} = n \frac{kT}{q} \ln(\frac{I_1}{I_2})$$

Where n is emission coefficient of the two diodes. In, the differential output voltage is linear with absolute temperature. This property has been known since the late 1960s and this voltage difference forms what is termed a PTAT (Proportional to Absolute Temperature)voltage. In most works, the diode connected BJT is used instead of a diode as shown in Figure 5.3. Although the notation is a bit more tedious, it can be shown that if ratio between the currents in two diode-connected transistors is a constant, then the differential voltage denoted as Δ*VBE* in Figure 5.3 is also a PTAT voltage provided simple analytical models are used to characterize the BJT devices.

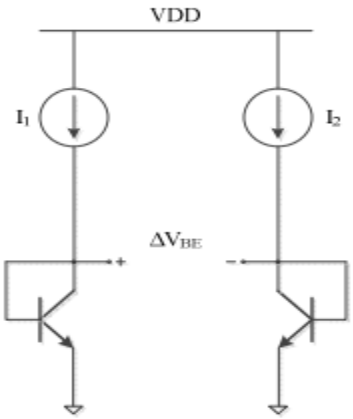


Figure 5.3 Traditional temperature sensor design

Although lots of products and published papers are based on this structure, the area required for on-chip pn-junction based temperature sensors is invariably large and they dissipate considerable power. Even ignoring the size and power concerns, this type of temperature sensor is not practical for power management in standard bulk CMOS processes

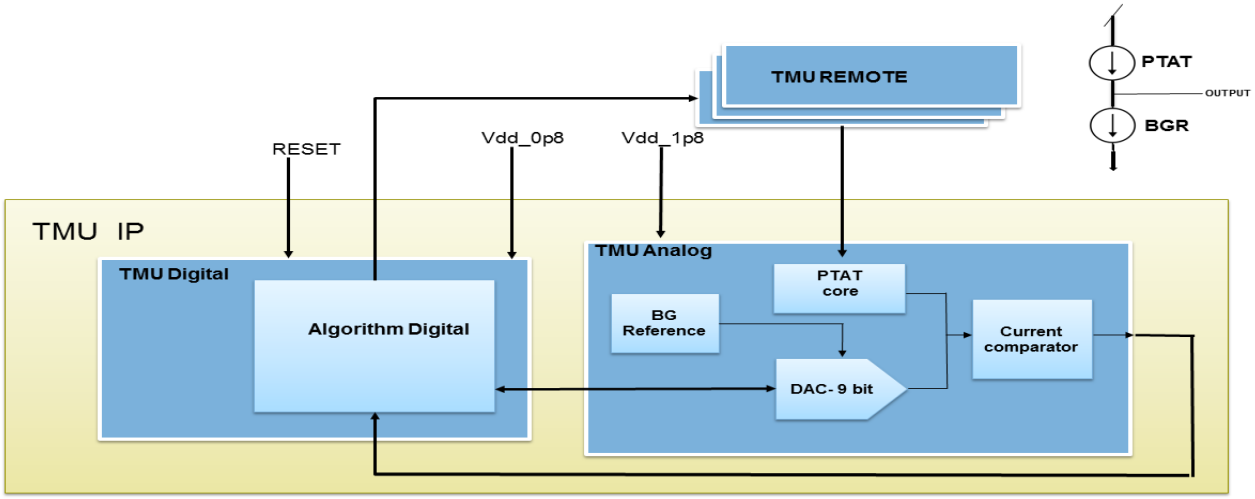


Figure 5.4 Temperature Sensor Design

Digital Block: all the threshold set data is provide under digital block. After the measurement of temp this block send the data or feedback to SAR algorithm. When the SAR algorithm result to again feed to the digital block, the block judge the output value that weather the temp is violating any threshold or not . the below table is the example set value of the threshold. The digital block also store the data of calibration table for the calibration mode.

Interrupts Thresholds kept for AMS simulations	**Degree Celsius**
HIGH Temp Immediate Interrupts	More then 150
High Average Temp Interrupts	Optimum temp
High Temperature Average Critical Interrupts	100
Low Temp Immediate Interrupts	-40
Low Average Temp Interrupts	-10
Low Temperature Average Critical Interrupts	-30

Table 5.1 Temperature Threshold Value

A) Bandgap Voltage Reference

Many circuits, including voltage regulators, analog-digital and digital-analog converters, require a voltage reference that is as precise as possible. Their precision depends on it. That means that the voltage reference would ideally be PVT independent:

- P: manufacturing process variations
- V: supply voltage
- T: temperature

Band gap reference circuits cancel out two opposing variations caused by temperature. That is, if we have two references, one producing voltage V1 with temperature coefficient

$$\partial V1/\partial T=\alpha$$

and the other producing voltage V2 with temperature coefficient

$$\partial V2/\partial T=-\alpha$$

The following operation

$$Vout=V1+V2$$

will produce a temperature-independent voltage:

$$\partial Vout/\partial T=\partial V1/\partial T+\partial V2/\partial T=\alpha-\alpha=0$$

Of course in order to cancel out, the temperature coefficients must have opposing signs, one negative (NTC) and one positive (PTC). A Bipolar Junction Transistor (BJT) can provide both the NTC and the PTC voltages. Thus, let us review a bit the BJT.

Things to know about the BJT

The BJT collector current is defined as:

$$I_{c=}\, I_s e^{(V_{be}/V_t)}$$

where Vbe is the base-emitter voltage. Vt=kT/q is the thermal voltage, which is virtually insensitive to process variations and is defined by the Boltzmann constant k, the charge of an electron q and the temperature T. IS is a parameter that is process and temperature dependent:

$$I_S = I_0 e^{\frac{-V_{G0}}{V_t}}$$

$$I_0 = I_S e^{\frac{V_{G0}}{V_t}}$$

where I0 is a device parameter and VG0 gives the name to this reference circuit. It is the bandgap voltage of silicon, the energy necessary to free an electron from the outer shell of the silicon atom. The bandgap itself is temperature dependent, so VG0 is VG(T) extrapolated from 300ºK to 0ºK. It has the theoretical value of 1.205V. The collector current can also be expressed as:

$$I_c = I_0 e^{-\frac{V_{G0}-V_{be}}{V_t}}$$

which makes it simpler to relate to the bandgap voltage.

Creating the NTC voltage

The negative temperature coefficient is produced by a PN junction. The base-emitter junction of the Bipolar Junction Transistor (BJT) is a common PN junction used in bandgap references. Using the collector current of the BJT, the base-emitter voltage is:

$$Vbe=VG0-(kT/q)logI0Ic$$

So, how does Vbe depends on temperature? Assuming that Ic does not depend on temperature:

$$\partial Vbe/\partial T=-k/qlogI0/Ic$$

Given that I0 is much larger than Ic, the log term is not greatly affected by the BJT current. Normally, the temperature coefficient is around −2mV:

$$\partial Vbe/\partial T\approx-2mV/^{o}C$$

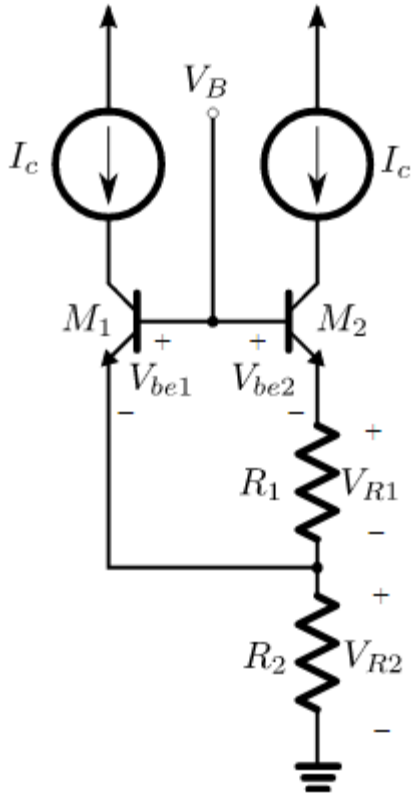
Creating the PTC voltage

The positive temperature coefficient is produced by the thermal voltage Vt. Say we take the difference between two base-emitter junctions of BJTs:

$$Vbe1-Vbe2=VtlogIc1/IS1-VtlogIc2/IS2$$

$$\Delta V_{be} = V_t \log \frac{I_{c1} I_{S2}}{I_{c2} I_{S1}}$$

Simple enough, ΔVbe gives us a voltage proportional to Vt, as well as some scaling of that voltage, if we like. But still... we want the voltage to be referenced to ground, so we have to elaborate a bit more.



Let us find the voltages across the two resistances

$$VR1=VB-Vbe2-(VB-Vbe1)=Vbe1-Vbe2=\Delta Vbe$$

$$VR2=R2\cdot 2Ic=R2\cdot 2VR1R1=2R2R1\Delta Vbe$$

Voilá, VR2 is proportional to ΔVbe and referenced to ground!! We can also replace the Vbe's as:

$$VR2=2R2R1VtlogIcIS2IcIS1=2R2R1VtlogIS2IS1=KVt$$

Since k and q are constants, Vt is only proportional to temperature, with temperature coefficient:

$$\partial Vt\partial T\approx+0.085mV/ºC$$

The thing is, the terms IS are proportional to the area of the transistor and very similar for nearby transistors (in terms of layout). Therefore, the following is pretty accurate:

$$IS2IS1\approx A2A1$$

where A represents the area of the transistor. Then:

$$VR2=2R2R1logA2A1Vt$$

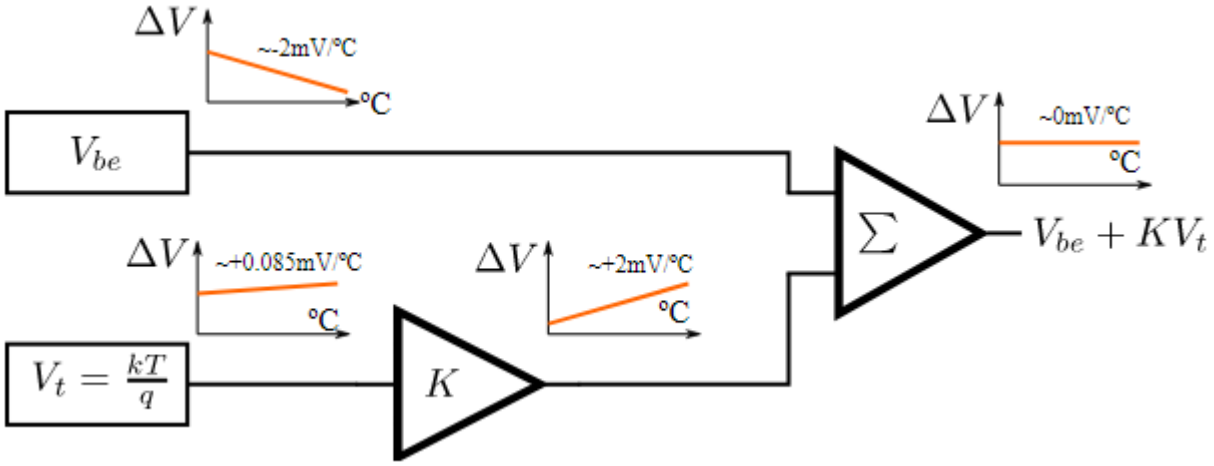
Finally, it is curious that VB already is the sum of a voltage proportional to Vt (PTC) and a Vbe (NTC):

$$VB=VR2+Vbe1=2R2R1logA2A1Vt+Vbe1$$

How VB is biased and how we implement the current sources are questions that will be answered next...

Summing the NTC and PTC voltages

So our main idea is to produce two voltages, one PTC and another NTC, which cancel each other in terms of temperature variations. The diagram would look something like this:



The sum of the two voltages would be:

$$Vbe+KVt=VG0-kTqlogI0Ic+KkTq=VG0+kTq(K-logI0Ic)$$

As we have seen when creating the PTC voltage, K will probably come from a ratio of resistances and/or areas of transistors, and with value $2/0.085 \approx 23.5$. Before moving on, note that after the temperature dependent terms cancel, the reference will stay at the bandgap VG0. If you ever see a voltage reference with a value close to 1.2V, it comes from here! The next obvious question is: how do we implement this with real circuits?

There are several ways, but the main recipe is the following:

1. Generate two currents to bias two different BJTs

2. Create a branch involving two BJTs and one resistance (or equivalent resistance)
3. Find a path that adds a PTC and a NTC voltages
4. Tweak the sizes of the transistors and/or resistances to give the right K
5. Add a startup circuit, or else the circuit may stabilize at zero current every time

B) Digital to Analog Converter :

Output of the digital to analog converter is goes to the current comparator. The current mode DAC is used to compare the current between current generated by OpAmp biased with **BGR** and PTAT current.

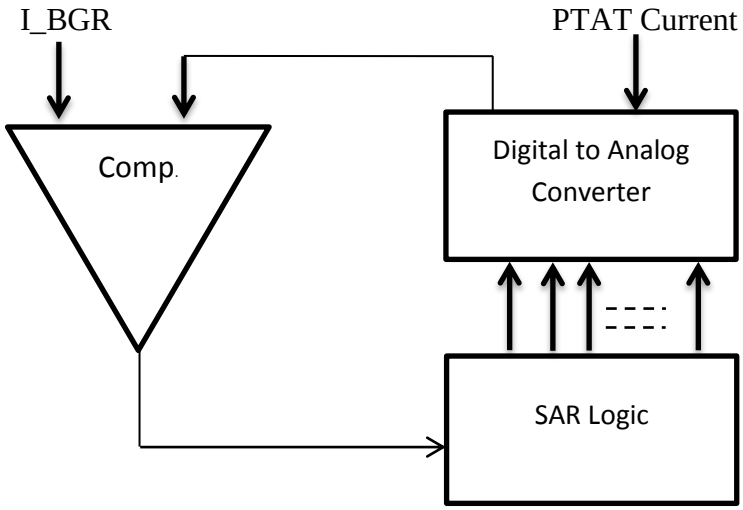


Figure :SAR ADC

Current Comparator

I have completed a design of Traff current comparator that has current signal as a input and output is in the form of a voltage. My design can sense the change of the current level up to 0.2 uA.

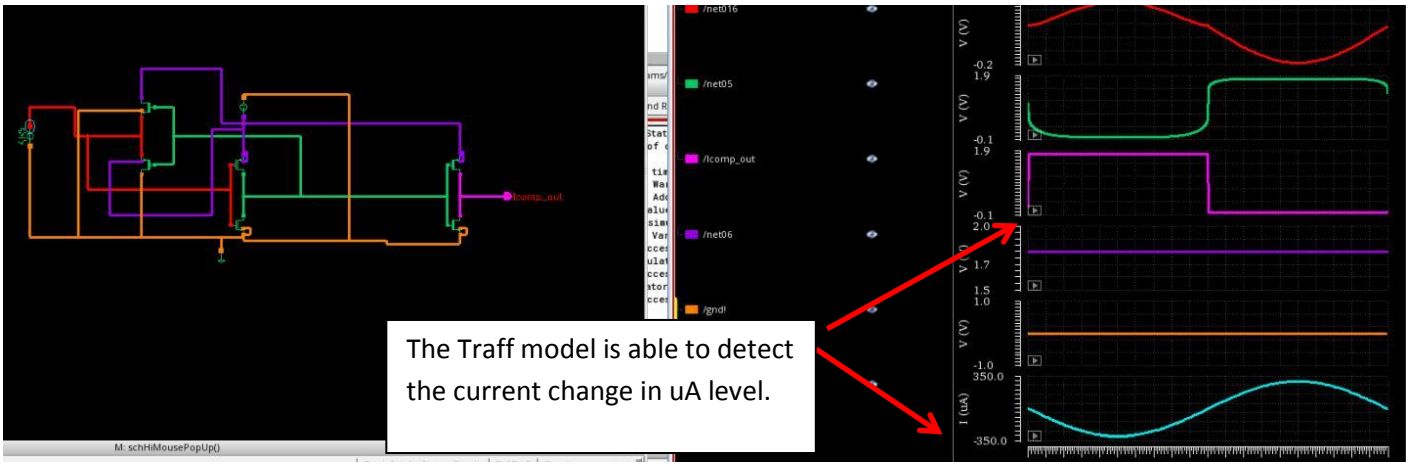


Fig 5.5 . Current Comparator

When the sensor complete the temp measurement, that value continuing pass to analog block for the further measurement and . The mail function of the analog block is try to balance the current coming from ptat secton with the current created with the help of BGR .after calculating the current which is same as temp of the chip , is pass to digital block , where the binary of the current compared with the set threshold , which is come from current comparator. After comparing, FCCU and INTR signal is generated.

5.7 Conclusion:

Till we understand the AMS system, Ams verification with UVM method, and TMU now we will see the verification results on the DTU and examine the output, with effeteness of methodology.

CHAPTER 6

TEST SETUP AND RESULTS

A good way to test the UVV method is to create a realistic scenario of a testbench set up to perform verification of an NFC-A listening device. From Nordic Semiconductor ASAs point of view rigorous testing of their IPs are important and the UVC is designed to contribute to production of error-free designs. Nordic Semiconductor ASA provided their NFC-A listening device for this thesis as well as their original testbench as a starting point for the testing of the UVC. This chapter will describe the original NFC-A listening device testbench and how it was modified with the UVC. A test designed to verify functionality of the NFC-A listening device will be described. A verification plan was set up for measuring functional coverage of the NFC-A listening device based on the capability of the UVC and the NFC-A protocol. Ultimately the goal is to close on 100% functional coverage of the NFC-A listening device based on the verification plan.

6.1 TESTBENCH AND FILES

The test bench is written in System Verilog and simulates the TMU device in a SOC Chip environment. The top module of the test bench sets up the power and clock domains, TMC, test bench and register interfaces, and a polling device verification module for stimulus production. The test bench is bound to assertions specific for the TMU device and starts a System Verilog program. The program resets the environment, sets up loggers and holds an interface monitor and coverage groups. After the initialization step various test are run to verify the design in terms of noise, oscillator calibration, register access, timing requirements, and pattern recognition and field detection.

A couple of modification is necessary when the UVM is included in the test bench , we need to modify some files with respect to the requirement of the test subject here is TMU. To perform the implementation of any circuit, in every stages I shall need as well as get some collaterals which will be needed in several stages. So, in this chapter I shall briefly discuss about the collaterals.

These are the some file of the UVM need to be verified first before simulation:

a) <u>(Dot) V file</u>

These files are written in Verilog or as requirement we can use any hardware descriptive language programmed files. Describe functionality of each standard cell. Define each standard cells I/Os, inputs and outputs. Provide timing information to the standard cell, which will be used to simulate gate level netlist.

b) (Dot) VAMS file

It is the electrical modeling of a standard cell files. This file mostly used to program for analog or mixed signals IPs. This file used to describe analog behavior of a circuit. Describe functionality of whole IPs written in VAMS. This file is used for electrical parameter values like resistor, capacitor, voltage, current etc. which are mostly analog in nature. Define IPs I/Os, inputs and outputs. Provide timing information to the standard cell, which will be used to simulate gate level netlist.

c)(Dot) f file:

A number of programmed files are combined together to simulate a single standard cell. Like what is its behavioral model, what is its test bench model etc. this file is used to describe simulation plan. Behavioral model files and test bench model file combined together.

d)(Dot) cdl file :

This is a SPICE programmed file. Describe the whole wire frame of circuits. As programmed in SPICE, it is mostly used for analog simulation. All the signals are described in analogue. Netlist file of an IP is also a .cdl file.

e) (Dot) scs file:

It is a spectre file for IC simulation in a software environment. Used to describe the simulation environment to a simulation. Here it is used to compile the SPICE level simulation environment or RTL level simulation environment to the SOC according to SPICE simulation or RTL simulation.

f) (Dot) csh file:

It is a SPICE and spectre file. Used to describe how the simulation should runs. Describe the typical temperature for the simulation. Use as a simulation command for simulating.

g) (Dot) tcl file:

It is the tool command language (TCL). Used for controlling multiple tools file in a single script. Describe the depth level of a simulation. Describe to see voltage or current level of a signal. Used to describe the probing parameter of a simulation.

h) (Dot) log file:

Basically it is the simulation result file. Sometimes a single simulation takes couple weeks to complete. So each and every moments of simulation data should be recorded for verifying the results. It describe each and every moment of a compilation or simulation. Larger is the simulation complexity large is the log files like RTL simulated log files is quite small compared to SPICE simulated log files. If the simulation is successful, result or waveform can be visible by opening the log file in simvision or viva.

i) (Dot) arg file:

It is one of a test case file. Used to describe different argument on a test case. Different kinds of arguments are their like make arg, sim arg & comp arg. It describe the simulation factors like temperature and depth of simulation.

g) (Dot) c file:

It is the second one of a test case files. Here user programmed for the mode and kind of oscillator is used for PLL simulation to the test case.

6.2 Verification Plan

The TMU device has a comprehensive set of features that should be verified. However, as the UVM is a protocol UVM, focus is to verify that the TMU device behaves according to protocol. In order to ensure sufficient functional coverage from the testbench, a test plan was need to set up.

Which values are important?

- The power status signal is important. Whether chip is getting proper power o not as well as the simulation graph is sowing the signal or not.
- How the chip is dealing with reset signal. How the generated reset affecting the behavior of the SOC.
- Power on reset and Low Voltage detect signal is how perfect, what is the timing, slew rate and effects of these factor on the output.
- The output signal is able to detect the simulation temperature? If yes does threshold flag is activated?
- If threshold flag is generated than after result FCCU is activated or not?

What are the dependencies between the values?

- power is the key . The LVD and POR signal is dependent of power signal. After reset, the activation, function and deactivation of the TNU module is also depend on the given value for the test, reset signal and clock.

Are there any invalid conditions?

- Due to some mismatch or delay of some other module affecting the function of some module then it's a unwanted situation, need to be solve .

When is the data invalid?

- The conclusion of the verification plan is that each carrier characteristic should be covered by separate coverage points and the dependency between the pulse width and the period should be cross covered.

6.3 Wavesforms

- In the figure 6.1 , the Power signal is up , means the chip is getting the power , and it's now our work to check or set the LVD , reset value . the output is cear and as we epectd in this verification methology.After checking the LVD and POR the module is started and giving the desire output . the Dac value is the digital value of temperature .

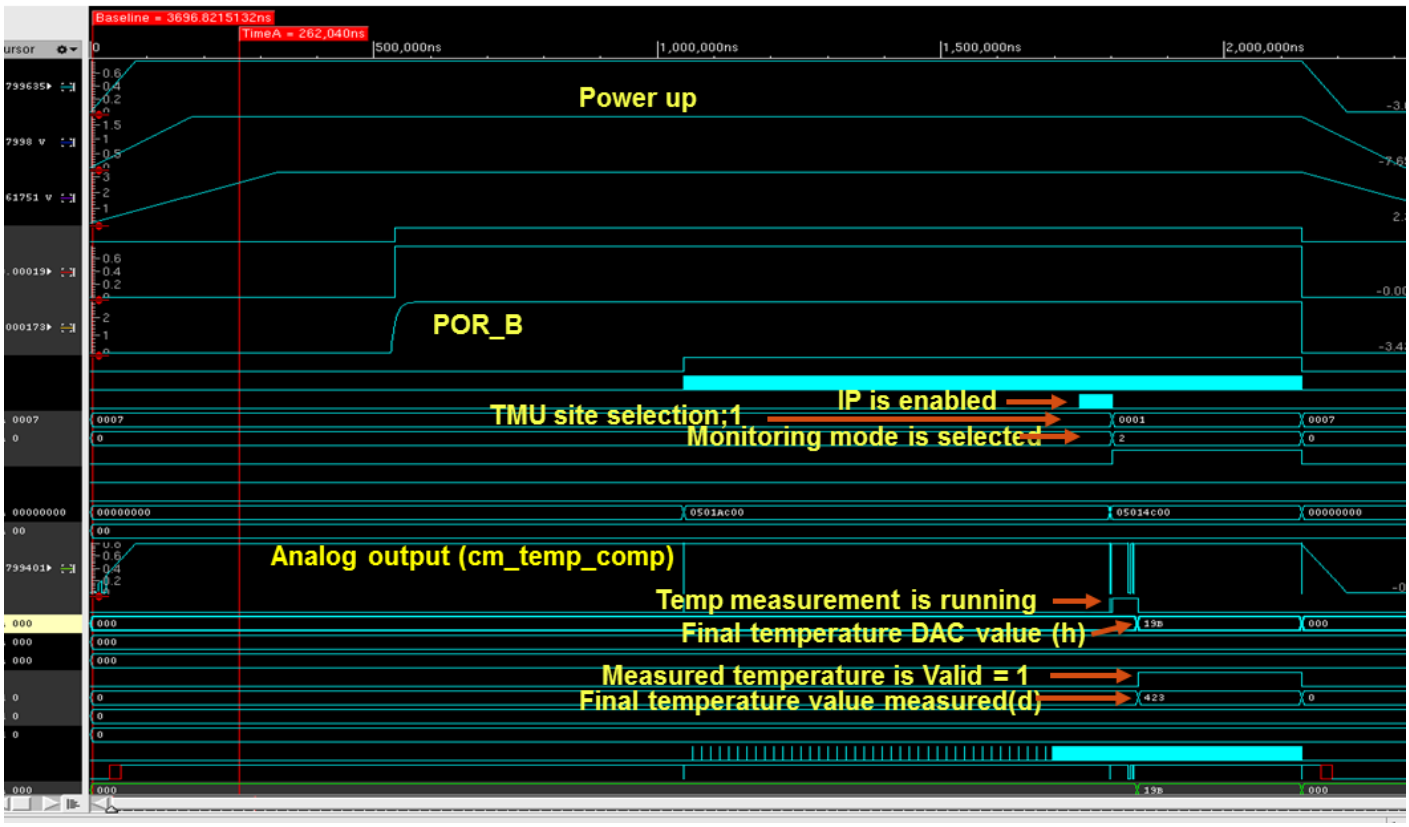


Figure 6.1 Initial look after simulation

- In the figure 6.2, the after result of the figure 6.1. In the previous fig we got that the temp of the chip is 150. so according to the our threshold set its more than average, average critical temp and critical set of 77, 100 and 150 so its really need that all the flag will generated .and in the wave from we can see that with the help of UVM mythology we are getting the right result and FCCU is also active to performs and instruct some task to the other module.

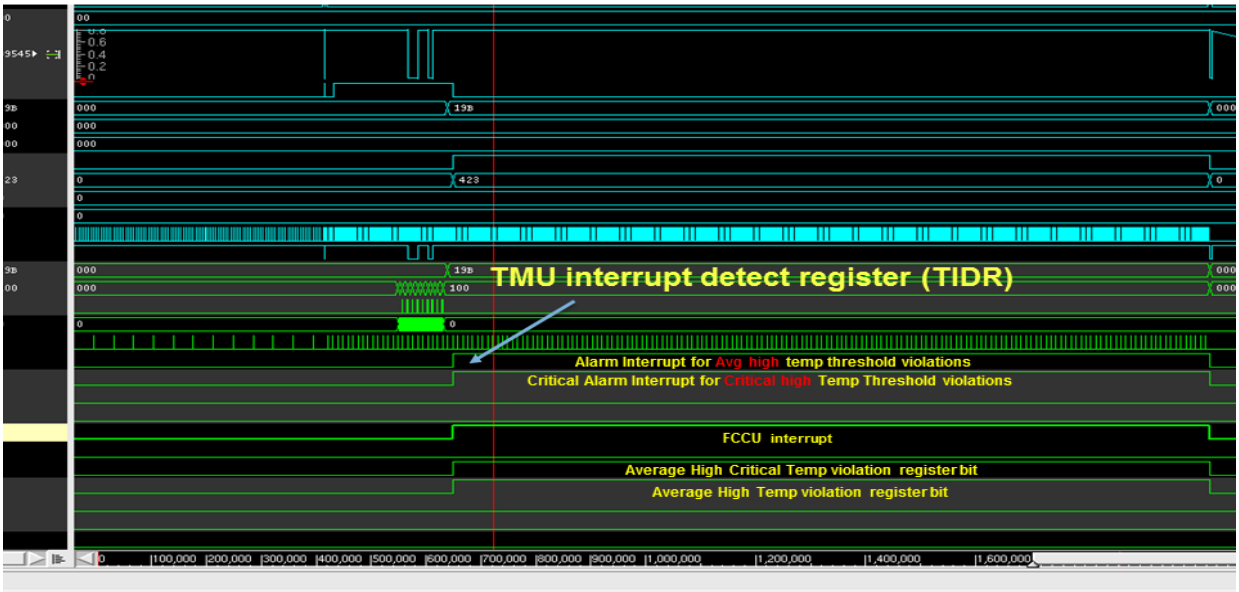


Figure 6.2 Interrupt generated

- The Figure 6.3, show that after the simulation works , the TMU is able to detect the change of the temperature .

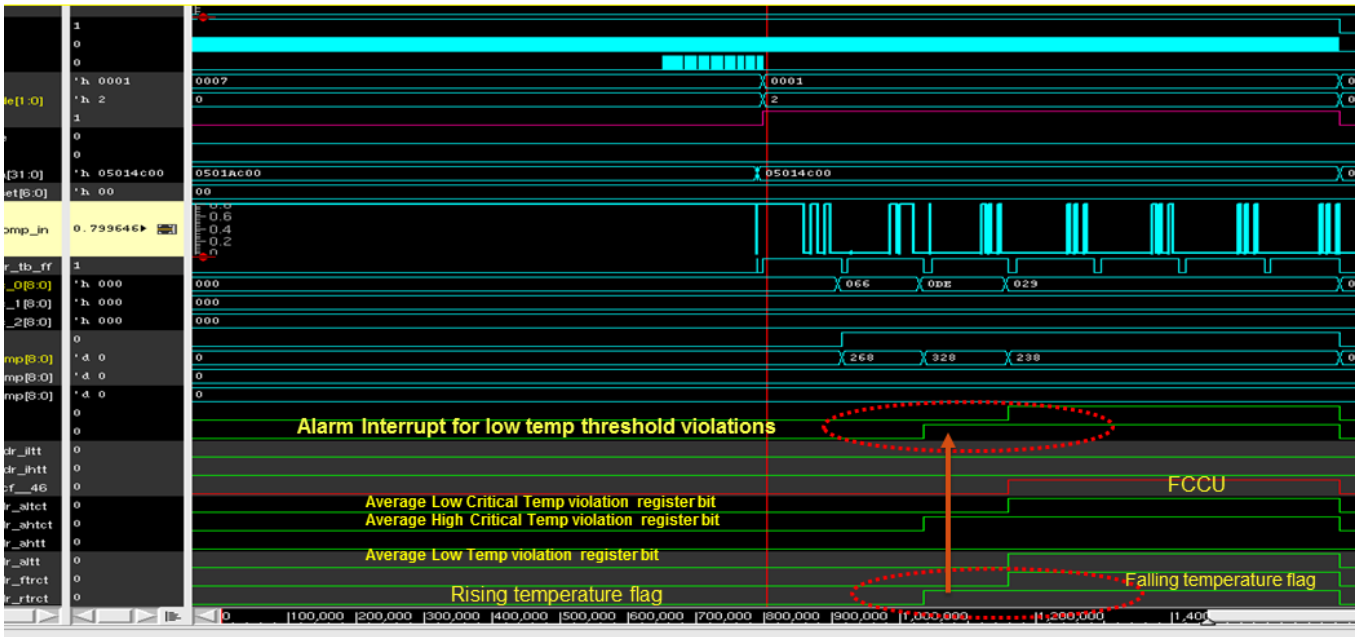


Figure 6.3 Critical Flag Generated and FCCU

- The Figure 6.4 shows that after simulation and verification, the conclusion is that after a particular time when the other sensor is selected, that is also giving the simulation set tamps value. So IP is working fine.

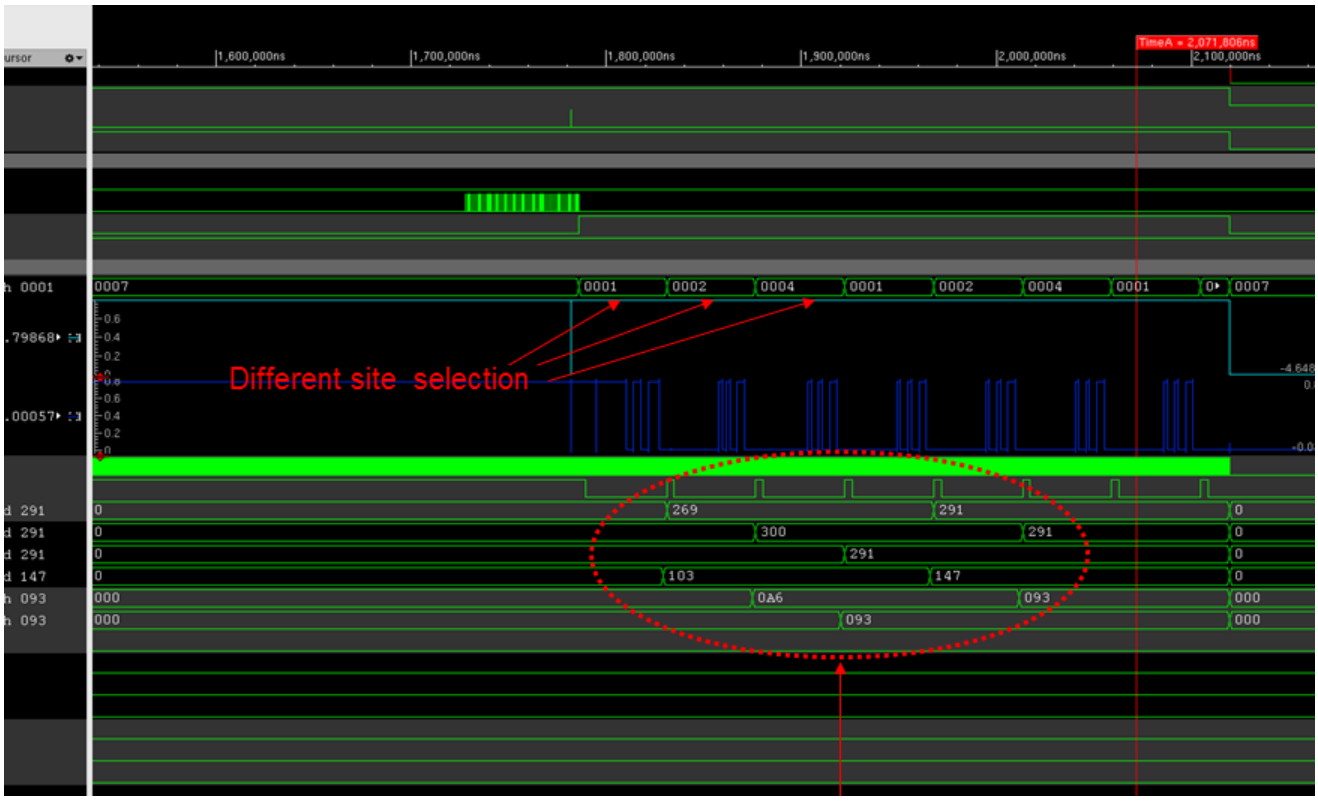


Figure 6.4 Different Sensor Reading

- The final result shows that at every point of time and simulation temperature, the UVM Verification of the Temperature or Thermal Monitoring Unit on system on chip level is completed and all the value matched with the verification plan.

Job id	Sim temp(Deg C)	Cal. Temp	Flags	Comments
				MONITORING MODE
				STATIC TEMP
id0126	-8	-4	No	No violation
id0135	-8	-4	No	No violation
id0133	-8	-4	no	No violation
id0137	-8	-4	No	No violation (ITS IN THE RANGE OF +/- 8)
id0136	-8	-4	No	No violation
id0134	-8	-4	No	No violation
id0125	88	89	ipi_critical_alarm, ipi_alarm,Fccu,ahtt,ahtct	Temp getting 89 which is more than avg; ahtt, ahtch flag.
id0149	150	146/dac-146.5	ipi_critical_alarm, ipi_alarm,Fccu,ahtt,ahtct	AvgHigh and AvgHigh Critical Threshold violations..
id0124	150	150	ipi_critical_alarm, ipi_alarm,Fccu,ahtt,ahtct	AvgHigh and AvgHigh Critical Threshold violations.
id0148	150	148.5dac /148	ipi_critical_alarm, ipi_alarm,Fccu,ahtt,ahtct	AvgHigh and AvgHigh Critical Threshold violations.
id0147	-40	-35	ipi_critical_alarm, ipi_alarm, Fccu,altt,altct	AvgLow and AvgLow Critical Threshold violations.
id0123	-40	-35	ipi_critical_alarm, ipi_alarm, Fccu,altt,altct	AvgLow and AvgLow Critical Threshold violations.
id0146	-40	-35	ipi_critical_alarm, ipi_alarm, Fccu,altt,altct	AvgLow and AvgLow Critical Threshold violations.
				OUT OF RANGE
id0122	165	164.5 dac code	ipi_alarm,,IHTT (OUT OF HIGH RANGE)	Immediate high temperature threshold exceeded/ 164.5 dac code/trns=nil(effected by ips_rdata_inX)
id0121	M43 and -50	N38 and M44.5	ipi_critical_alarm, ipi_alarm, Fccu,altt,altct	AvgLow and AvgLow Critical Threshold violations.
				Temp_PWL
id0128	M8.P25.P15	M4.P25.P17	NA	TEMP IN THE RANGE
id0129	M8.P25.P16	M4.P27.P18	NA	DO
id0127	M8.P25.P17	M5.P27.P18	NA	DO
				TEMP_RATE
id0130	M8.P55.M40	M5.P56.M35	Rtrct>ftrct FLAG , THEN FOLLOW -40 TEMP	AvgLow and AvgLow Critical Threshold violations.
id0131	M8.P55.M40	M5.P55.M35	Rtrct>ftrct FLAG , THEN FOLLOW -40 TEMP	AvgLow and AvgLow Critical Threshold violations.
id0132	M8.P55.M40	M4.P55.M35	Rtrct>ftrct FLAG , THEN FOLLOW -40 TEMP	AvgLow and AvgLow Critical Threshold violations.
				CALIBRATION MODE
id0141	150	146		cgf_site given is 2 and total site will be cal by 0 to cfg_site ()
id0143	150	150		do
id0139	150	143.5		do
id0140	-40			do
id0142	-40	-35		do
id0138	-40	-35		do
id0144	150	148.5		do
id0145	150	146.5		do(mon_mode_ setting)

Output close to OOR_L

Figure 6.5 Results

CHAPTER 7

CONCLUSION

The work of this thesis presents the study of Analog and Mixed signal system and verification ,study of Thermal Monitoring Unit with successful implementation of a UVC for the Universal Verification Methodology. The UVC has been simulated in a testbench with a TMU and was able to close on 100% functional verification of the device according to a verification plan. Reusability has been one of the main focuses in the creation of the UVM, thus components of the UVC can be reused for other UVC projects. Parts of the TMU protocol that is not covered by the UVC could be implemented with minimal effort due to the focus on facilitation of additional functionality. The UVC illustrates how UVM can be used with RVM to achieve digital simulations speeds in functional verification of analog IPs. A sequence library for stimulus generation comes with the UVC to provide users of the UVC with examples and a starting point for creating relevant testbench stimulus.

7.1 Recommendations for Future Work

- Design the driver and the monitor components reusable by replacing the algorithms with calls to interface methods.
- Extend the monitor component with the ability to detect noise on the interface.
- Extend the stimulus library with responses for listening devices.
- Create more reusable scoreboard design.
- Working on Tarff Current Comparator to make it more sensitive towards sensing current

REFERENCES

[1] Dave Evans. “The Internet of Things How the Next Evolution of the Internet Is Changing Everything”. In: (Apr. 2011).
[2] J. David, “Efficient functional verification for mixed signal IP,” in ProceedingsOf the 2004 IEEE International Behavioral Modeling and Simulation Conference,October 2004, pp. 53–58.
[3] IEEE Standard for systemverilog – Unified Hardware Design, Specification, andVerification Language (1800-2009), IEEE Computer SoCiety, Dec. 2009.
[4] S. Palnitkar, Design Verification with e. Prentice Hall Professional Technical
[5] A. Molina and O. Cadenas, \Functional verification: approaches and challenges,"Latin American applied research, vol. 37, no. 1, pp. 65{69, 2007. Reference, 2003.
[6] Sathishkumar Balasubramanian and Pete Hardee. “Solutions for Mixed-Signal SoC Verification Using Real Number Models”. In: (2013).
[7] A. Rath, V. Esen, and W. Ecker, \Analog transaction level modeling," in High Level Design Validation and Test Workshop (HLDVT), 2011 IEEE International.IEEE, 2011, pp. 82{82.

[8] Saraju P. Mohanty, A secure digital camera architecture for integrated real-time digitalrights management, Journal of Systems Architecture - Embedded Systems Design 55 (2009), no. 10-12, 468–480.
[9] Saraju P. Mohanty and Dhiraj K. Pradhan, Uls: A dual-vth/high-kappa nano-cmos universal level shifter for system-level power management, JETC 6 (2010), no. 2.

[10] S. P. Mohanty, N. Ranganathan, E. Kougianos, and P. Patra, Low-Power High-Level Synthesis for Nanoscale CMOS Circuits, 1st ed., Springer, 2008.
[11] , Unified Challenges in Nano-CMOS High-Level Synthesis, Proceedings of the 22nd International Conference on VLSI Design, 2009, pp. 531–531.

[12] , Design of Parasitic and Process-Variation Aware Nano-CMOS RF Circuits: A VCO Case Study, IEEE Trans. VLSI Syst. 17 (2009), no. 9, 1339–1342.
[13] J. Park, K. Choi, and D. J. Allstot, Parasitic-Aware Design and Optimization of a Fully Integrated CMOS Wideband Amplifier, Proceedings of the Asia South Pacific Design Automation Conference, 2003, pp. 904–907.(www.cmc.ca/)
[14] \Universal Verification Methodology (UVM) 1.1 Users Guide," May 2011. [Online]. Available: www.uvmworld.org
[15] S. R. Little, D.Walter, and C. J. Myers, “Analog/mixed-signal circuit verification Using models generated from simulation traces,” in Automated Technology forVerification and Analysis (ATVA), ser. Lecture Notes in Computer Science, K. S.Namjoshi, T. Yoneda, T. Higashino, and Y. Okamura, Eds., vol. 4762. Springer-Verlag, 2007, pp. 114–128.
[16]Metamodeling-Based Fast Optimization Of Nanoscale Ams-Socs Oleg Garitselov University Of North Texas
[17]https://www.design-reuse.com/articles/28333/analog-mixed-signal-verification-methodology.html