ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter82872020.pdf
https://doi.org/10.30534/ijeter/2020/82872020

Reset Logic Verification of an IOD at System on Chip Level using GateSim

Kakarlamudi Lakshmi Maidhili[1], Fazal Noorbasha[2], Allamsetty Vamsi[3], Kakarla Hari Kishore[2]
[1]M.Tech Student, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, India.
shinestarmaidhili@gmail.com
[2]Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, India.
fazalnoorbasha@kluniversity.in
[3]Senior Silicon Design Engineer, Advanced Micro devices, Hyderabad, Telangana, India.
vamsi.int@gmail.com

ABSTRACT

These days SOC became very essential part and it is a great revolution in electronics world. Basically, for any SOC or (IP) verification results are very important as we cannot predict that our chip we design will meet our expected performance or not. As a consequence, the failure of the SOC / IP is more common in development. SOC / ASIC verification begin as soon as architecture or micro architecture is established in any industry. This continuous procedure is structured to ensure the accuracy of the basic design specification prior to the tape-out. Product life cycle along with performance of the chip is prime factor in any electronic device. In order to increase our confidentiality of chip Gatesim that is Gate Level Simulations (GLS) are introduced in Verification era. Particularly the 14 nm technology nodes and below are responsible for much longer simulation run times and are also responsible for larger memory. As a result, in order to complete the testing specifications in time, it is extremely necessary for GLS to be started as early as possible in the design process and for the simulator to run in high-performance mode. In this paper we mainly focus on verifying reset test logic for Input Output Die (IOD) part of entire SOC is working as expected or not using Gate level simulation simply referred as Gatesim.

Key words: SOC, GLS, Boot logic, Verification

1. INTRODUCTION

In today's electronic design automation Verification is very crucial. Functional verification is a necessary step in the development of today's complex designs.
Most of the Design and development cycle of an SOC or any IC is verification. Hardly verification is of 70% of product development cycle. Whereas 20 % is spent for implementation and rest 10% is for design part as shown in the bar graph in Figure 1.

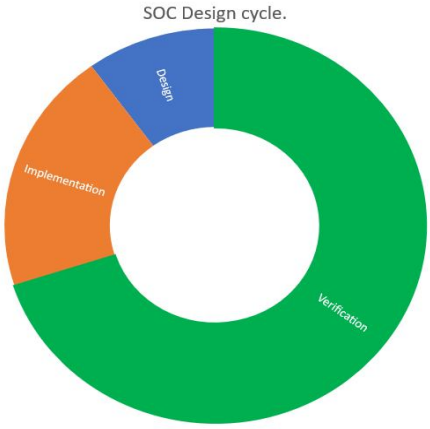

Figure 1: SOC design cycle

SoCs are built by a large number of in-house and third-party IPs. The integration of multiple processor cores and IPs is a difficult task. It is even more difficult to verify the different scenarios that come with such complex designs. It has become essential to carry out a hardware-software verification to cover the functionalities presented by both hardware and software structures. Because SoCs usually require a minimum of one CPU, it is important to check and see how the CPU responds with different input stimulations. Confirmation of SOC/ASIC starts off evolved as quickly as structure or micro architecture is described in any type of region. This cyclic technique to ensure the useful correctness of the information layout specification before tape-out. Our checks are placed in RAM, and the processor reads as well as implements those recommendations. Generally we use C or CPP to write confirmation checks, compiled as well as converted into hex code especially for the processor being used, on the way to be filled proper into memory. We keep in mind that after energy-on and reset sequences are executed, the CPU assessments out boot code and additionally in the end start appearing instructions from RAM. System Verilog/UVM is used in validating IPs t block level verification [1][6]. A lot of precise checks are required at Chip level we require to compose precise checks at chip diploma to ship offers from the IP to the rest of the additives in the machine. Vector unit's requirement to be created for every and every IP further to device level checks like safety, pad multiplexing, and so forth.

This is extraordinarily critical since it may make or damage the machine with also a bit connection mistake.

1.1 Verification models

Verification at SOC level is mainly categorized into two. They are Functional Verification and Non- Functional verification. Non- Functional Verification is again classified into Timing and Performance verification. In timing verification all the timing (static timing analysis - setup, hold, capture etc.) and frequency related like input frequency as per specifications is reflecting at the output or not is verified. In Performance verification microprocessor related events and memory performance like read from and writing into respective memories are as expected are verified Whereas in Under Functional verification all the respective features, protocols are verified. Figure 2 shows the types of verification.

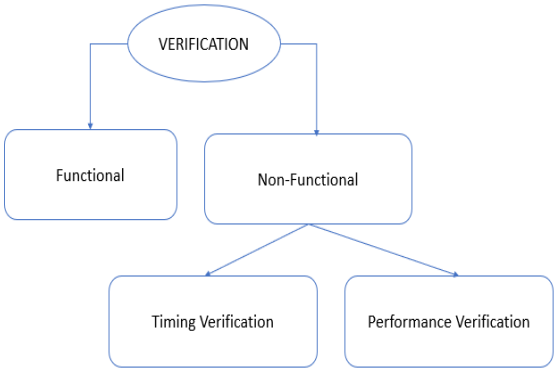

Figure 2: Types of SOC Verification

1.2 Technology Options

The Verification methodologies may be system level, Block level, IP level, SOC level, Physical verification. There are five verification Technology options [7]. They are Simulation based Verification; Emulation based Verification, Formal and Semi–Formal verification, Hardware and Software co-verification.

2. SOC ARCHITEURE AND RESET TEST SCENARIO

The AMD Embedded G-Series SOC platform is a high-performance, low-power System-on - Chip (SOC) architecture featuring enterprise level error-correction code (ECC) memory support, dual and quad core versions, integrated discrete GPU, and I / O controller on the same die[8]. The general architecture of a SOC is described in Figure 3. A SOC is an integrated circuit (IC) that incorporates all electronic components of a computer or other electronic device into a single chip. SOC includes digital, analog, mixed-signal, and sometimes radio-frequency functions on a single chip substrate. AMD's The Fusion chip consists of both the graphics and CPU integrated into a single chip. Multinational Semiconductor Company AMD's any SOC can be categorized into CCD (Core Complex Die) and IOD (Input Output Die). The entire SOC consists of 4 to 12 CCDs and one IOD. The number of CCDs present depends on the product price/performance class. Fusion is a multi-core microprocessor architecture that incorporates computing and graphics cores into a single processor package. A graphics processing unit or GPU is a complex electronic circuit designed to rapidly manipulate and modify memory in such a way as to facilitate the production of images in a frame buffer intended for display output. Here the processor core is an X86 core and the architecture has been implemented in processors from AMD, Intel, VIA, Cyrix, and many others.

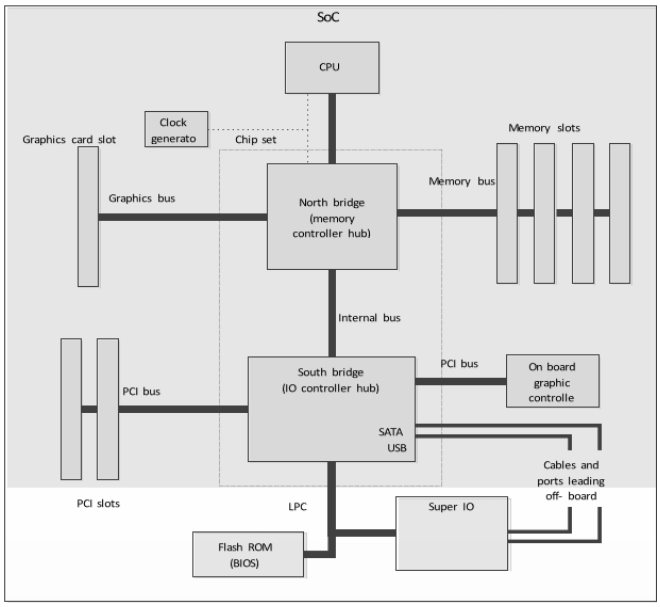

Figure 3: High level Fusion SOC Architecture with connectivity

Northbridge is used for handling communications among core CPU and other Ips and memory. In certain instances, it deals with RAM, PCI Express (or Accelerated Graphics Port (AGP)) video cards, and Southbridge. The south bridge is the motherboard IC responsible for the hard drive controller, the I/O controller, and the integrated hardware. Integrated hardware can include sound card and video card on motherboard, USB, PCI, ISA, IDE, BIOS, and Ethernet. South Bridge, on the other hand, is the main controller of all input output functions USB, audio, serial, BIOS network, ISA bus, interrupts controller, and IDE channels. Memory is of basically two types; one is volatile and other is non-volatile. Interfaces are basically connecting all the components like CPU, North Bridge, South Bridge, DDR-Dram, graphics processor. Block basically provides the clock for all other blocks. DDR- DRAM is the volatile memory which is basically increases the performance as it works on faster clock then the hard disk. To manage communications between a core processor and the motherboard we use input output controller hub [9][14]. Normally low pin count bus is used for communication in modern super input output chips rather than Industry Standard Architecture (ISA) Hey would the central processing unit. This normally occurs through an Low pin count bus interface on the South bridge chip of the

motherboard. Peripheral component interconnect shortly called as PCI is a high speed bus used for adding internal components to any personal computer or desktop. The advantages of a bus is that it makes parts or interchangeable. The clocking block which basically provides the clock to all the IPs of the SOC such as North Bridge, South Bridge, and Cores etc? As every block of IP require its specific clock speed so there will be many PLL and DLL which will make the clock to the requires speed.

2.1 Reset Process

The device reset sets all registries to their reset values except the Reset flags in clock-controlled registries and the backup domain registries allowed during reset. A system reset can be generated by hardware or software for example if the system reset button is pressed down the total system gets reset. In hardware point of view a pull-down capacitor is added in parallel to reset button. The use of this capacitor is to improve the level of electromagnetic susceptibility EMF. Electromagnetic susceptibility is defined as the ability to operate without fault under electrical disturbances or noise. This bypass capacitor can remove high frequency voltage signals from RF depend thus the transient circuit demand on the power supply unit as shown in Figure 4.

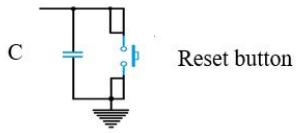

Figure 4: Hardware reset circuit diagram

2.2 Software Reset

The reset or boot technique begins with Power on Reset (POR) where the hardware reset logic causes the middle CPU to start the implementation beginning with the boot ROM on-chip. The boot ROM code uses the given boot pick-up option in addition to the FUSE / strap country and additionally GPIO set-ups to determine the boot circulate behaviour of the SOC. Figure 5 shows the boot code process [21]. In X86 type architecture we force the power switches at 0Xfffffff0. After that it will start loading the Images and thereby it does equipment and IP initialization [15]. Boot Loader does the duty of loading the Flash image in to Memory device. To make a SOC tool besides-up it calls for a chunk of software program known as the Boot ROM. This Boot ROM is to be smaller sized in dimensions to match the on-chip ROM usually it takes around 256Mb and additionally it's miles virtually pricey to exchange.

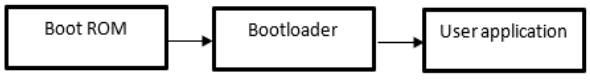

Figure 5: Boot code process

Boot loader typically able to erase the main application, receive a new application, receive a new application, and program the new application to flash memory. Any piece of code that executes at or near boot time that is used to load and or executor main application [17]. Boot loaders are common in almost every end application because they enable field firmware updates by the end customer. These boot loaders typically reside in a single flash sector or a ROM.

2.3 Reset Test Flow for IOD in SOC

The boot processes otherwise the Reset testing for current AMD SoC's relies on the external SPI flash containing a valid boot loader image. If the SPI flash is either empty or corrupted, the SoC is unable to continue booting. Boot code is responsible for initializing all core complexes on SOC. Initialization sequence is divided into early and late stage. To accommodate the programming of the off-chip SPI flash, it is necessary that the SoC's on-chip Boot code be able to accept incoming flash data from an external source as shown in the Figure 6.

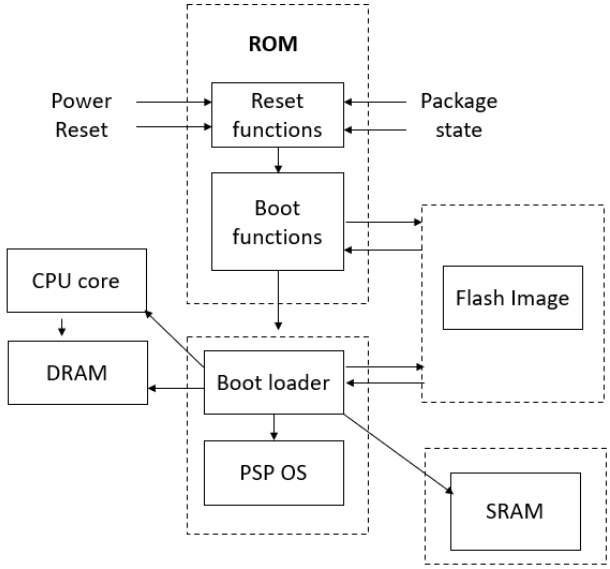

Figure 6: Boot code flow in AMD SOC

The actual Reset test flow starts by loading a Flash image in the form of hex file to the firmware. The main intension of the Hex file is it contains all the register data and required addresses of the design. Authentication of the image received will be done inside RAM memory. The piece of software called as Boot code is responsible for reloading the image from external world. Then on chip module receives data from PC and send to SRAM. Prior to loading the operating system, the BIOS offer basic peripheral support device drivers that are part of the motherboard, including the keyboard, display, and hard disk [16]. Drivers allow the user to change software settings and allow the hardware to boot from a hard disk or other storage device. The generated ROM image will contain set of instructions, data, ROM Keys and Hash.The on-chip USB controller again receives data from the Host PC to the SRAM, from which it is programmed to the external SPI flash device (ROM) [18].

3. RESET TEST LOGIC VERIFICATION

The Gate Level Simulation begins soon after the Gate Level Netlist is created after our RTL (Register Transfer Logic) synthesis. The netlist can be checked using the formal equivalence test technique with the RTL code as the reference version and the gate-level netlist as the implementation method. The reference is considered as GOLDEN reference.

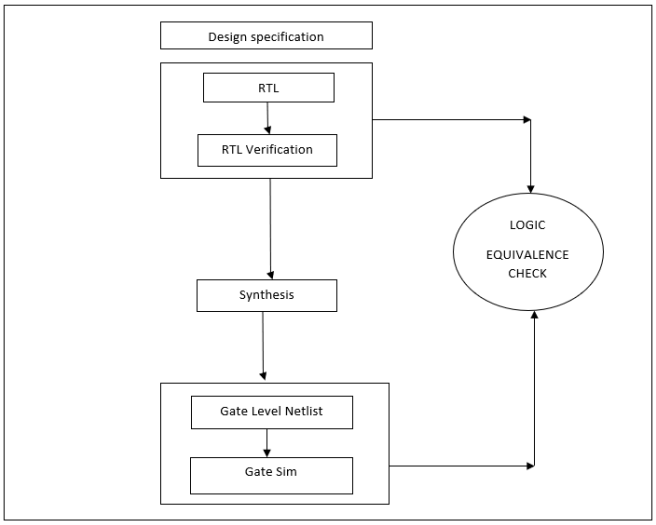

Figure 7: Gatesim Hierarchy

The goal of Gatesim is to verify whether the Netlist meets the Logic as expected with appropriate timing and after a lot of optimization. Each and every sub module of IOD is to be understood and test related configs are brought up with all possible reset test scenario. In this paper COOL RESET sequence of Input Output Die of AMD SOC especially the Data Fabric and Universal Memory controller is generated and verified successfully.

3.1 GateSim

Gate level simulations (GLS) are a net list view of any circuit. A Typical Netlist describes list of all interconnections made between various digital electronic circuits [19]. GLS or Netlist simulations usually need to begin early when functional confirmation is still progressing to flush the GLS flow and confirming that the netlist is hooked up appropriately. Gate level simulations are generally referred to as zero delay simulations. Timing at this moment can be 'zero delay', 'device delay'. These simulations are performed after back annotating initial 'pre format Standard Defined Format (SDF)' as well as finally 'publish layout SDF' Typically, RTL simulations are absolutely no delay simulations and on the all the events happen energetic clock edge [20]. But GLS is not an any hold-up it is thought about as a unit delay version or full timing model simulation. Figure 7 shows the gatesim hierarchy. As per the ASIC Design Flow of once the design is verified and followed by synthesis it goes to the placement and routing which is called as VLSI Backed development. The Netlist obtained soon after the placement and routing of the design is called as Post Physical and Netlist extracted soon after the design specifications and design is done is called as Pre -physical Netlist. The number of layers of metallization is around 11 layers. Figure 8 shows the netlist types.

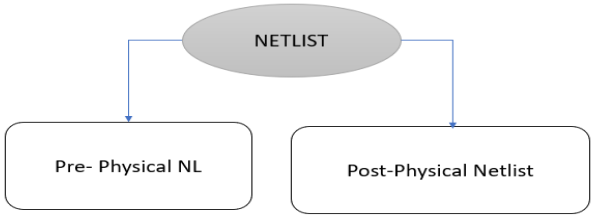

Figure 8: Netlist types

4. TOOLS USED AND VERIFICATION FLOW

UVM stands for Universal Verification Methodology. This provides the best framework to achieve Coverage-Driven Verification (CDV). The verification component is applied to the device under test (DUT) to verify implementation of protocol or design architecture. Figure 9 shows the verification environment.

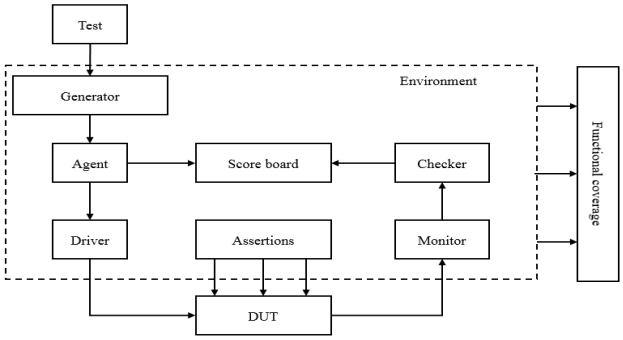

Figure 9: Verification Environment

Simulation tool used is VCS by Synopsys and for Debugging tool used in this work is Verdi by Synopsys and some AMD legacy tools. The boot code verification starts with the checkout in Linux environment. Leo stream which assigns desktops to the users for cloud environments, Leo stream also provides advanced functionality for managing capacity.

4.1 Verdi

Verdi is a synopsis tool used for debugging errors by using graphical environment by loading file. fsdb dump into the Verdi and checking for all the global signals from the particular hierarchy. Figure 8 describes the Dump loaded into the Verdi tool.

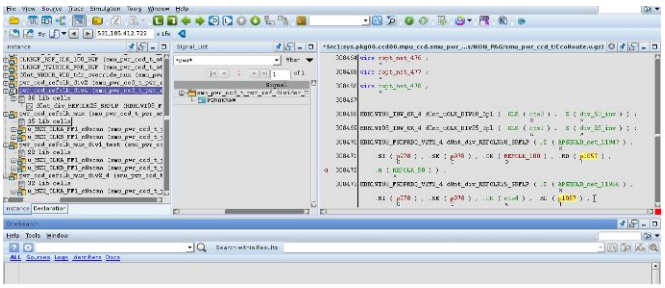

Figure10: Dump loaded inVerdi

This is the graphical environment loaded with all our IOD especially Data fabric and Universal Memory Controller bus signals. A dump with .fsdb extension is loaded into Verdi that contains all the test related modules and top level SOC hierarchy as shown in Figure 10.

4.2 Role of Optimization

The role of Optimization is very crucial in this Reset test verification. Optimization is done as a part of synthesis. This care classified into Logic, Power, Area and Timing optimization. Soon after the design is done with design specifications and implementations it goes under several cycles of Verification. During this process lot optimization will be done. A 32 bit register can be optimized into 23 bit register with rest of 9 lines of the register length is optimized. Similarly during the design many modules can be power off using adiabatic logic technique for optimization. With this many modules and sub modules of the design can be assigned or designed with VDD and ground or power signals. The design will contain hundreds of flip-flops with reset and without reset pins. During simulation of the design non-reset flip-flops will results in 'x' propagation that in turn effects the simulation and blocks many signals.

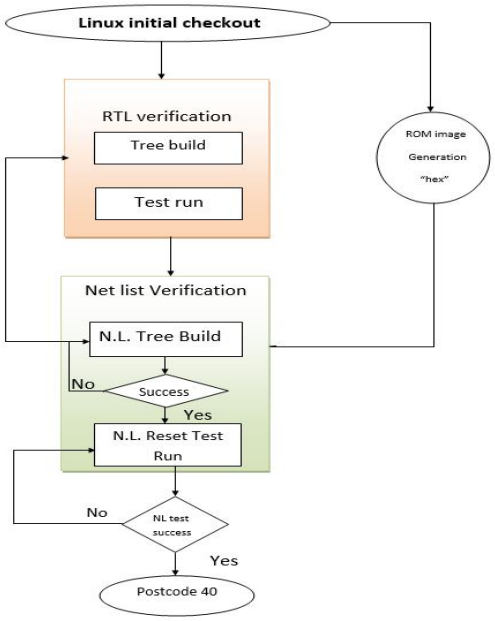

Figure 11:GLS Reset Test flow for IOD Die in AMD

The first step in Reset test flow is Done using Linux checkout shown in Figure 11. This syncs up all the required tools files associated with the SOC enviroment in Semiconductor design and verification.Once the checkout is completed we start with RTL Verfication which enhances to have a reference design for the Gatesim . If and only the RTL Verification reaches Postcode 40 we start with NL part.The flow is mentioned in Figure 11. A ROM image is generated simultaenously after the checkout for loading the boot code image from flash into the Bootloader.The flash image is a set of vector address in the orm of hexa radix (o-f) format. Figure 12 shows the Checkout from Linux environment.

```
amd_axi_uvc.sync OK.

Component Distribution  :1677 (51)  :1671 (49)  :1667 (13)  :1712 (4)  :1711 (4)
Total time to populate the 'genesis' codeline and all its nested components:    45
Total Workarea Size: 22.115 GB

All syncs OK!
```

Figure 12: Checkout from Linux environment

The Reset test takes the testbench as input and starts initializing all the required library files, configs required for modules and IP s present in the SOC. Here in this paper the Reset test flow for the Input Output Die is done. The testbench is written in Cpp model and system Verilog which completely enables all the registers. The test case uses JTAG interface for the RTL level. Once the test is passed, we proceed to NL Tree build and NL test runs. The actual GLS starts from NL Tree build. The output of the NL Tress build is defined as Compile clean with deposits. Deposits are nothing but forcing a bit 0 or 1 to a particulate register bit to allow the flip-flops drive the particular block. Figure 13 shows the NL Compile clean completed with deposits.

```
recompiling module HDBSVT08_INV_1
recompiling module HDBSVT08_INV_CB_1
recompiling module HDBULT08_BUF_1
recompiling module HDBULT08_BUF_CB_1
recompiling module HDBULT08_BUF_CK_6
recompiling module HDBULT08_DEL_R2V4_1
recompiling module HDBULT08_INV_6
recompiling module HDBULT08_INV_CKY2_10
recompiling module HDBULT08_INV_SKR_4
recompiling module HDBULT08_TIEDIN_4
recompiling module HDBLVT11_BUF_4
recompiling module HDBLVT11_DEL_R2V1_1
recompiling module HDBLVT11_INV_1
recompiling module HDBLVT11_INV_CB_2
recompiling module HDBLVT11_INV_SKRY2_12
recompiling module HDBULT08_MRKBUF_4
recompiling module HDBULT11_BUF_CB_2
recompiling module HDBULT11_DEL_R2V3_2
recompiling module HDBULT11_INV_8
recompiling module HDBULT11_INV_SKF_4
recompiling module HDBULT11_INV_Y2_16
recompiling module HDBULT11_INV_SKR_4
CPU time: 5302.160 seconds to compile + 435.857 seconds to elab
```

Figure 13: NL Compile clean completed with deposits

RTL tree brings up all the required configurations (configs) i.e, all the blocks defined for SOC architecture and the rtl definitions of each and every block included in it and all the sub-blocks. For testing the entire SOC (system on chip) at RTL level in which the booting sequence of soc is defined in terms of cpp file or a .sv file or in a Verilog file which is called as a test bench. The performance of SOC is improved with the GLS activity. The cross-module issues, 'x' propagation issues, bind issues library issues and some syntax errors which cannot be found in the RTL level are found in the GATESIM. Taking RTL model as reference design and its corresponding NL after synthesis which is called as implementation design is referred as Formal Verification in semiconductor industry. We call it as GOLDEN reference. In this NL verification flow various errors including Cross Module Resolution errors and Binding errors are solved .The main reason for Cross module errors is that when ever module defined in RTL soon after completion of the synthesis in Netlist point of view the module may not be defined or defined with other naming convention then the compiler gets the Cross module issues.
After fixing al the cross module and bind and library related

issues we trigger NL Reset Test logic for IOD after 10k cycles of clock the COOL Reset code gets detected and is shown in the Figure 12.

```
250519 '^
250520 info: TEST COMPLETE.info: pass condition detected (Halt)
250521 info: C001C0DE detected
250522 info: COOLCODE detected
250523 info: RESULT:: TEST PASS
250524 EnvCycle:            4180000
250525 *Verdi* : Flush all FSDB Files at 695,000,000,000 fs.
250526 Use of assignment to ${ is deprecated at /tool/pandora64/.package/perl-5.24.0-gcc620/lib/5.
250527 EnvCycle:            4190000
250528 EnvCycle:            4200000
250529 **************INVOKING SIMTOOL PLI FINISH and cosimlib tb_finish ***************
250530 **************DONE SIMTOOL PLI FINISH*****************
```

Figure 14: Reset Test Status Report

The above Figure 14 shows the cool code and boot code test status soon after completion of NL test run. The test used here in the verification of IOD part is cool boot code. There are 2 types of reset they are cool reset and warm reset. Cool reset in theory resets entire SOC chip. If any block or module that needs to be reset then the designer will use cool reset strobe to do it internally called as COOL reset. It is also called as Power Cycling. Figure 15 shows the Output Global signals and STATUS signal results.

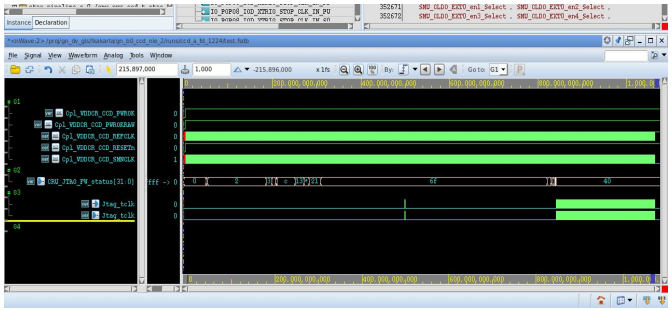

Figure 15: Output Global signals and STATUS signal

The above Figure shows the global signals RESET signal and Clock signals. The CRU_JTAG_STATUS signal which is 32 bit length reaches final Postcode 40. This signal indicates that the NL Reset Test for IOD die of entire SOC with dump reaches the COOL Reset successfully with initializing of all the respective register initializations and transactions.
The signals
Cp1_VDDR_CCD_Resetn is a active low signal which is showing 1'b0 so it is activated successfully for the entire die.
Figure 16 shows the Boot Sequence success for IOD.

```
250459 691090000 INFO    /proj/ba_dv_gls_nobackup/lkakarla/ba_a0_iod_pre_bto_df_
       'PwrMgt: COLD_RESET:DONE:: IOD Boot Sequence successfully completed'^
```

Figure 16: Boot Sequence success for IOD

The above-mentioned figure shows the Boot sequence of IOD. It indicates that all the IOD part completed the Reset Test sequence at SOC level. The top-level view of the Netlist will be in the form of basic gates and buffers. After implementation of design the complete Netlist will be defined at gate-level as a result during simulation we get gate-delay.

5. CONCLUSION

This paper describes how boot code is verified for the entire SOC chip (especially for the IO Die) by Gate Level Simulations. These are used widely in the Semiconductor industry. The common issues which cannot be found in the RTL level verification come out after synthesis are debugged using GLS are described.

ACKNOWLEDGEMENT

It's a genuine pleasure to express our deep sense of gratitude and thankfulness to Nimish Agarwal, Sr. Manager Silicon Design Engineering Server SOC DV at AMD-Bangalore for encouraging us to the highest peak and providing continuous encouragement and guidance and thank Karthikeyun Srinivasan, for his elevating inspiration, immense patience and motivation in the completion the paper. The authors would like to thank AMD-Hyderabad for the facilities provided without which this work might not be completed successfully.

REFERENCES

1. Mahesh Madavath, K Hari Kishore **"**RF Front-End Design of Inductorless CMOS LNA Circuit with Noise Cancellation Method for IoT Applications" International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8, Issue No: 6, Page No: 176-183, April 2019.
2. K.Sarath Chandra, K Hari Kishore **"**Electrical Characteristics of Double Gate FINFET under Different Modes of Operation" International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8, Issue No: 6S, Page No: 172-175, April 2019.
3. P.Ramakrishna, M. Nagarani, K Hari Kishore **"**A Low Power 8-Bit Current-Steering DAC Using CMOS Technology" International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8, Issue No: 6S, Page No: 137-140, April 2019.
4. Avinash Yadlapati, K Hari Kishore **"**Implementation of Asynchronous FIFO using Low Power DFT" International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8, Issue No: 6S, Page No: 152-156, April 2019.
5. P Ramakrishna, K Hari Kishore **"**Implementation of Low Power and Area Efficient 7-Bit Flash Analog to Digital Converter" Journal of Computational and Theoretical Nanoscience, ISSN: 1546-1955, Volume-16, Issue No: (5-6), Page No: 2213-2217, June 2019.
https://doi.org/10.1166/jctn.2019.7875
6. Mahesh Madavath, K Hari Kishore **"**Design and Analysis of Receiver Front-End of CMOS Cascode Common Source Stage with Inductive Degeneration Low Noise Amplifier on 65 nm Technology Process" Journal of Computational and Theoretical

Nanoscience, ISSN: 1546-1955, Volume-16, Issue No: (5-6), Page No: 2628-2634, June 2019. https://doi.org/10.1166/jctn.2019.7942

7. K Hari Kishore, Fazal Noorbasha, Katta Sandeep, D. N. V. Bhupesh, SK. Khadar Imran, K. Sowmya "Linear convolution using UT Vedic multiplier" International Journal of Engineering and Technology(UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 409-418, March 2018. https://doi.org/10.14419/ijet.v7i2.8.10471
8. K Hari Kishore, B. K. V. Prasad, Y. Manoj Sai Teja, D. Akhila, K. Nikhil Sai, P. Sravan Kumar "Design and comparative analysis of inexact speculative adder and multiplier" International Journal of Engineering and Technology(UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 413-426, March 2018.
9. G Siri Vennela, K Hari Kishore, E Raghuveera "High Accurate and Power Efficient ECG-Based Processor for Predicting Ventricular Arrhythmia" Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 10, Issue No: 2, Page No: 1180-1121, May 2018.
10. Avinash Yadlapati, K Hari Kishore "Low Power Synthesis for Asynchronous FIFO using Unified Power Format (UPF)" International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 7-9, March 2018. https://doi.org/10.14419/ijet.v7i2.8.10315
11. Chella Santhosh, K. Hari Kishore, G. Pavani Lakshmi, G.Kushwanth, P. Rama Krishna Dharma Teja, R. S. Ernest Ravindran, Sree Vardhan Cheerala, M. Ravi Kumar **"**Detection of Heavy Metal Ions using Star-Shaped Microfluidic Channel" International Journal of Emerging Trends in Engineering Research, ISSN: 2347-3983, Volume-7 Issue-12, Page No: 768-771, December 2019. https://doi.org/10.30534/ijeter/2019/067122019
12. B. Srikanth, M. Siva Kumar, J.V.R. Ravindra, K. Hari Kishore **"**Double Precession Floating Point Multiplier using Schonhage-Strassen Algorithm used for FPGA Accelerator" International Journal of Emerging Trends in Engineering Research, ISSN: 2347-3983, Volume-7 Issue-11, Page No: 677-684, December 2019. https://doi.org/10.30534/ijeter/2019/437112019
13. Nadhindla Bala Dastagiri, Kakarla Hari Kishore, Vinit Kumar Gunjan and Shaik Fahimuddin, "Design of a Low-Power Low-Kickback-Noise Latched Dynamic Comparator for Cardiac Implantable Medical Device Applications", Lecture Notes in Electrical Engineering, ISSN No: 1876-1100, E-ISSN: 1876-1119, pp. 637-645, 2018.
14. Avinash Yadlapati, Hari Kishore Kakarla **"**Low-power design-for-test implementation on phase-locked loop design" Measurement and Control, ISSN: 0020-2940, Volume-52, Issue No: (7-8), Page No: 995-1001, June 2019.
15. Nan Jiang, Abdol Ghaffar Ebadi, Kakarla Hari Kishore, Qahtan.A.Yousif, Mohammad Salmani "Thermomechanical Reliability Assessment of Solder Joints in a Photo-voltaic Module Operated in a Hot Climate" IEEE Transactions on Components, Packaging and Manufacturing Technology, P-ISSN: 2156-3950, E-ISSN: 2156-3985, Vol No: 10, Issue No: 1, Page No: 160-167, January 2020. https://doi.org/10.1109/TCPMT.2019.2933057
16. Mahesh Madavath, Hari Kishore Kakarla, Azham Hussain, C.S. Boopathi **"**Design and Analysis of CMOS RF Receiver Front-End of LNA for Wireless Applications" Microprocessors and Microsystems (SCI), ISSN: 0141-9331, Volume-75, June 2020. Article number 102999.
17. K Divya Madhuri, K Hari Kishore "Implementation of 4-bit Ripple Carry Adder by Adopting Sub threshold Adiabatic Logic for Ultralow-Power Application" Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 12, Issue No: 6, Page No: 11-17, May 2020.
18. B Srikanth, M Siva Kumar, J V R Ravindra, K Hari Kishore **"**The enhancement of security measures in advanced encryption standard using double precision floating point multiplication model" Transactions on Emerging Telecommunications Technologies, ISSN: 2161-3915, Volume: 31, Feb 2020. https://doi.org/10.1002/ett.3948
19. Chella Santhosh, K. Hari Kishore, G. Pavani Lakshmi, G.Kushwanth, P. Rama Krishna Dharma Teja, R. S. Ernest Ravindran, Sree Vardhan Cheerala, M. Ravi Kumar **"**Detection of Heavy Metal Ions using Star-Shaped Microfluidic Channel" International Journal of Emerging Trends in Engineering Research, ISSN: 2347-3983, Volume-7 Issue-12, Page No: 768-771, December 2019. https://doi.org/10.30534/ijeter/2019/067122019
20. B. Srikanth, M. Siva Kumar, J.V.R. Ravindra, K. Hari Kishore **"**Double Precession Floating Point Multiplier using Schonhage-Strassen Algorithm used for FPGA Accelerator" International Journal of Emerging Trends in Engineering Research, ISSN: 2347-3983, Volume-7 Issue-11, Page No: 677-684, December 2019. https://doi.org/10.30534/ijeter/2019/437112019
21. Radhika Rani Chintala, Lakshmi Sri Ram Janjanam, Sai Kousik G, Sai Pawan S ''FPGA Implementation of Katan Block Cipher for Security in Wireless Sensor Networks'' International Journal of Emerging Trends in Engineering Research , ISSN: 2347-3983, Volume-7 Issue-11 Page No: 492-497, December 2019. https://doi.org/10.30534/ijeter/2019/157112019