Validation of SOC-Based Hardware IPs with Diagnostic Software

P.Siva Kalyani[1], Dr.P.Maniganda[2]

[1]Assistant Professor,[2]Associate Professor,

[1]ECE Department

[1]Annamcharya Institute of Technology and sciences, Rajampet, India.

***Abstract*—**Diagnostic software is used in post silicon validation to validate SoC based hardware IPs automatically. Post-silicon validation has become one of the main goal in industries is to detect the bugs and remove them and also it is hard-to detect the rarely- occurring bugs that have slipped through pre-silicon verification. During the bug materialization while execution of unnatural random tests may be masked or not able to observe from the test outputs The ability to evaluate the bug-masking rate of a test provides great value in generating or effective tests are performed for high coverage deterioration by this also sometimes it is not able to find the bugs. To end this type of problems, an efficient method is to be proposed, automated diagnostic software development that can be used in post-silicon validation to validate various SOC based hardware IPs like, UART, DMA, IIC, GPIO, Timers and many more. Test cases are being developed by using C/C++ programming. The test cases can be executed randomly, rigorously, regressively can also be used to perform stressing especially on respective functionalities or IP features. This might help in finding the bugs at SOC IP level, hitting every corner cases etc. And the same software can also be used across the platforms, so that development time comes down, hence reduction in production cost.

Index Terms—Post silicon verification, pre silicon validation, UART, IIC, GPIO, SPI, ARM based development board.

I. INTRODUCTION

In semiconductor industries, the major cost involves in scheming and manufacture the IC die, sometimes while making of the die, it may cost in millions of dollars. If any bug propagates after making the die, it will impact huge loss to the industries. Hence, before making the first piece of IC, it is required to verify the silicon design or a particular IP design thoroughly before release to the market. The silicon industry follows this verification into various stages; broadly they are classified as verification and validation.

Verification is a process of testing the design against a given requirement before sending the silicon for production. This can be done using several methods like software simulations, synthesis, static formal analysis and FPGA/hardware emulation. During this verification the IPs are validated on FPGA platforms not on actual CHIPS/Silicon.

Validation is a process of which verifies a real silicon or a particular IP's intellectual property, for all its specific functionalities and electrical correctness in a lab set up. Here the validation is performed on original chips.

The main advantages of using these SOCs are miniaturization size, reduces the cost, Shipped with high margin and reduces time to market. SOC is used in various well-known products such as cell phone, digital multimedia players, game console and other consumer electronic devices.

EXISTING SYSTEM

In the current problem-solving environment IPs are validated physically, validation is not automatic. There is possibility of gone astray bugs in SoC IPs. Time to market is taking more time, due to labor- intensive efforts as these tools, are dependent on individual efforts, finding the number of bugs depends on the skill set and forbearance of engineer while running manual tests cases so, there will be missing the some of the bugs. To accomplishment over this kind of labor-intensive efforts a Automated Diagnostic software development for SOC based IPs post silicon validation is anticipated

PROPOSED SYSTEM

Advance technology in semiconductor enables us to integrate billions of transistors in a single chip. This ever-growing complexity pose a demanding problem for SoC based microprocessor or microcontroller design verification and attainment coverage closure within levelheaded time is becoming immensely very difficult to detect the bugs. High-end SoC based microprocessors often include multifaceted features (e.g. in transactional memories, security enhancement, signals and multiprocessor memory consistency) that are hard to verify in the early stages of verification. Therefore, post-silicon validation, that is, the validation effort carried out on the first silicon prototype, aims at catching all the remaining bugs that were not detected in the pre-silicon stage.

Therefore, post-silicon validation, that is, the validation effort carried out on the first silicon prototypes, aims at catching all the lingering bugs that were not detected in the pre-silicon stage.

This aspect makes bug detection and diagnosis challenge automatically. Even more significantly, bugs may manifest during a test's execution, but become masked and go unobserved by the time the test completes. During these situations the precious prototype's execution cycles are shattered. The types of tests deploy in post-silicon validation vary widely, from application scraps, to compatibility tests, to embarrassed random tests. These latter ones are particularly valuable in trying to exercise corner-case scenarios, since a vast number of variants can be generated with little designer's effort. One additional assistance that they bring to the validation effort is that they can often be engender directly in the

microprocessor under verification (i.e., on-platform), enabling an efficient use of the usually in shortsupply number of platforms and prototypes available.

As part my curriculum project it's very difficult carry out entire SoC validation, but my goal is to develop automated scripts using C/C++ on Linux platform for few IPs, as listed below.

1. UART_DMA
2. IIC
3. GPIOs
4. SPI

In the above mentioned list, minimum number of automated test cases is being developed. ARM Cortex and ARM7 based processor has been selected to demonstrate the validation diagnostic software. The typical view of IPs inside the SOCs are shown in the figure given below describe how hard to validate.

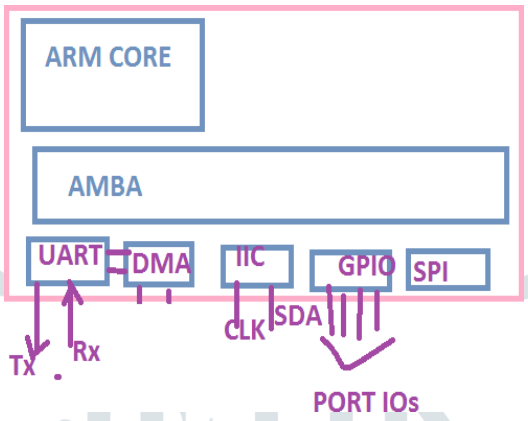


Figure: Typical view IPs inside SOC

A. UART DMA Validation

For each IP the minimum number of test cases has been validated. The features of UART DMA IP are listed below.

- DMA read transaction from system memory to UART FIFO
- DMA write transaction from UART FIFO to system memory.
- UART RX test
- UART TX test

The lists of registers in UART are as follows.

The algorithm for validation the feature is listed below.

- Test Case1: DMA write transaction from UART FIFO to system memory.

Purpose: To verify the RBR access test on UART0. Test Procedure:

- Reset UART and Wait till UART comes out of reset
- Configure UART baud using DLL and DLH registers
- Enable UART [1,0]x70[FAR]==1.
- FAR: FIFO Access Register. Read-write. Reset:

0. 1=Enable a FIFO access mode for testing when FIFOs are implement and enable UART[1,0]x78 Receive FIFO Write (RFW).Write known data (let's say: 0xaa) into: RFWD (0:7): Receive FIFO Write Data in RFW register. Write-only. Reset: 0. These bits are only valid when UART [1,0]x70[FAR]==1. When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO.When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into UART [1, 0]x00[RBR].

- Read RBR register, it should be 0xaa, otherwise return fail

Similarly, for the other features of UART DMA IP, the test plan is developed and code has been developed using C language.

B. IIC Validation

The features of IIC are listed below:

- Write default data to EEPROM
- Read the default data from EEPROM

Example for the one of the test case is listed below.

- Send a start sequence
- Send the IIC address of the slave with the R/W bit low (even address)

- Send the location of internal register number you want to write to
- Send the data byte to be written
- Send the stop sequence to the slave

C. SPI Validation

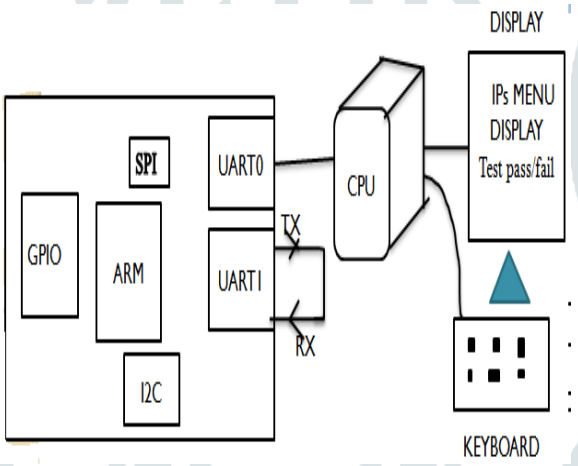
The features of IIC are listed below:

- Read file
- Write file
- Copy file
- Delete file
- Get the file list
- Memory statics
- GPIO validation

The features validating for the GPIO are given below:

- Led blinking on and off
- Dancing leds
- Nibble leds
- Two row leds

Similar algorithms are applied to the different test cases for each and every module. The block for this project is shown below



ARM Development board FIG: Block diagram

The Keil µVision4 is used for the compilation of the code. By choosing the module in the menu as shown in the block diagram the IP is validated automatically and shows the test is pass or fail in terminal. The command line hex file compilation type in Keil is used for the generating the hex file and the flash magic is uses for the dumping the code into the memory of micro controller and the output is observed in the terminal whereas, results are shown in figures.

The ARM based SOC IPs: UART-DMA, IIC and GPIO are validated successfully. IN this ARM 7 and cortex architectures are used to find the bugs, hence no silicon bugs found. These test cases if we run on new silicon, we may find bugs, so that semiconductor companies can confidently release the SOC to the customers. The output images are shown below.

RESULTS

The ARM cortex SOC IPs: UART-DMA, IIC and GPIO are validated effectively. We have taken ARM Cortex and ARM7 architecture based SOC, hence no silicon bugs establish. These test cases if we run on new silicon, we may find bugs, so that semiconductor companies can self-assuredly release the SOC to the customers. The output images are shown below

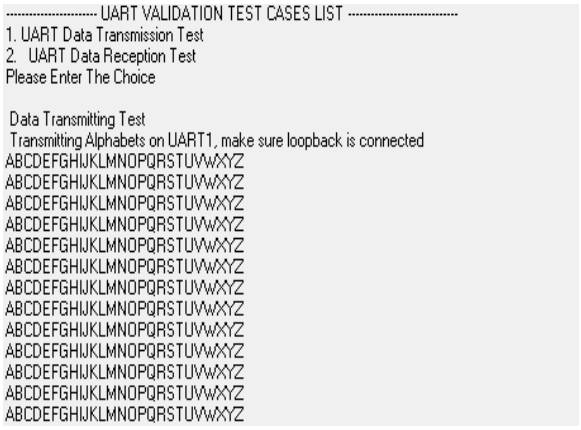


Fig: output of GPIO validation

Fig: output of SPI validation

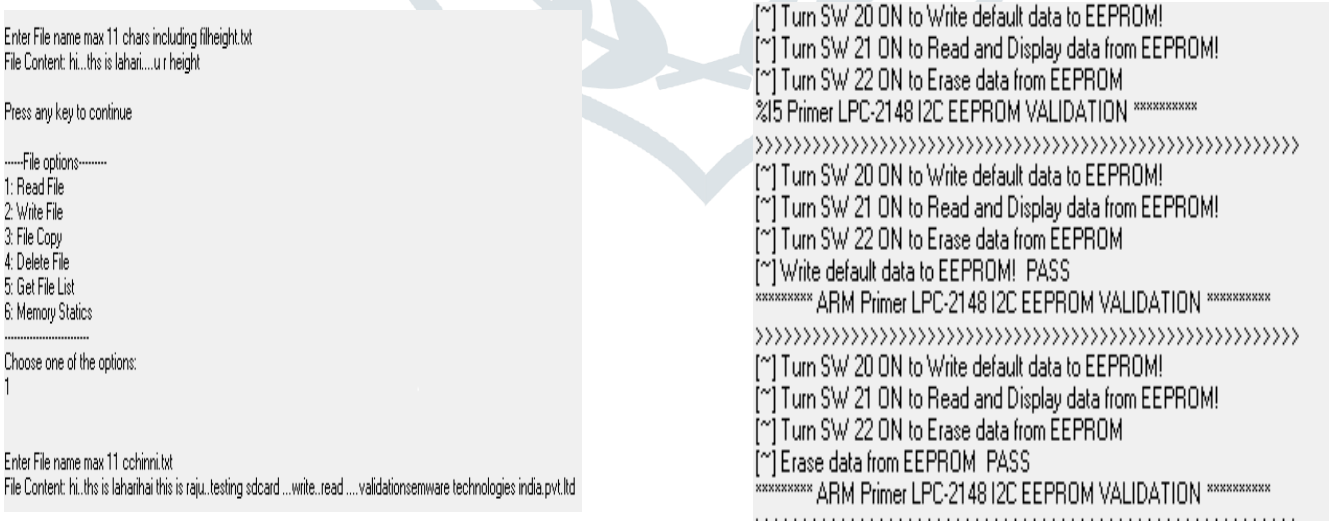


Fig: output of IIC validation

CONCLUSION

Diagnostic software is developed successfully to validate SOC based hardware IPs, for UART-DMA, IIC and GPIO. To validate the selected IPs, the ARM cortex and ARM7 architecture based processors are used, but the developed software is not only limited for ARM, can also be used for other SoCs, to validate listed IPs and also more different features can be added to the list. The developed diagnostic software can also be comprehensive to validate the other IPs with smallest effort. The software developed that can be used to validate in pre-silicon stage or post

silicon stage for new silicon. The best part of this project is to recognize mainly the complete silicon validation process in the industrial level of semiconductors.

ACKNOWLEDGMENT

We thank to our project guide, Mrs. P.Siva Kalyani and, for providing indispensable facilities towards carrying out this work. We are also very thankful to Mr. Samunuri Narayana raju, Embedded System Design Specialist, for all the support in successfully completing this project. We are also very much thankful to our project co-ordinator and entire ECE department faculty in giving the freedom in choosing this project, continuous support and encouragement

REFERENCES

[1] Wolfram Hardt, "An Automated Approach to HW/SW-Codesign," partitioning in hardware-software codesign, IEE Collequim on 13 Feb. 1995 , Page(s):4/1 – 4/11.

[2] Massimo Baleani, Frank Gennari, Yunjian Jiang, Yatish Patel, Robert K. Brayton, Alberto Sangiovanni-Vincentelli, "HW/SW partitioning and code generation of embedded control applications on a reconfigurable architecture platform," International Conf,erence on Hardware Software Codesign, Proceedings of the tenth international symposium on Hardware/software codesign,Estes Park, Colorado , 2002, Page(s):151 –156.

[3] 7. S. Yoo, A.A. Jerraya, " Hardware/Software cosimulation from interface perspective," Computers and Digital Techniques, IEE Proceedings, Vol. 152.

[4] Xue Wang, Gang Yu, Jay Lee, "Wavelet Neural Network for Machining Performance Assessment and Its Implications to Machinery Prognostics," Proceedings of the 5th International Conference on Managing Innovations in Manufacturing (MIM)(2002).

[5] Guofeng Tong, Muammer Koc, Jay Lee, "System Performance Assessment Based on Control System Criteria Under Operating Conditions," Proceedings of the 5th International Conference on Managing Innovations in Manufacturing (MIM)(2002).

[6] Gang Yu, Hai Qiu, Dragan Djurdjanovic, Jay Lee, "Performance Prediction Using Recurrent Neural Network Modeling with Confidence Bounds,"International Conference on Intelligent Maintenance Systems, July 15-17, 2004, Arles, France.

[7] Wang Xue, Liu Chengliang, Jay Lee, "Intelligent Maintenance Based on Multi- sensor Data Fusion to Web-enabled Automation Systems," 2004, July 15-27, 2004- Arles, France

[8] YANG Kun, ZHANG Chun, DU Guoze, XIE Jiangxiang, WANG Zhihua, "A Hardware- Software Co-design for 11.264/AVG Decoder," E.E. Department, Tsinghua University Beijing, China,2006.

[9] W. Kayankit, W. Suntiamorntut, "A Hardware- Software Co-design for Decoder,"Proceedings of the 2009 6th International Conference on Electrical Engineering/Electronics ,Computer, Telecommunications,