

Implementation And Verification of Soc Self Reliant To Pin Multiplexing Changes Using UVM Method

Gururaj,
E&CE DEPT.
RIT, Hassan
INDIA

(guruvnh4@gmail.com)

Rohith P. Asst. Prof.
E&CE DEPT.
RIT ,Hassan
INDIA

Abstract: *As the size and complexity of System-On-Chip (SoC) design is increasing rapidly, a portable and structured verification environment is desired to cope with time to market. The pin multiplexing complexity i.e number of functions/peripherals (IPs) multiplexed on a single I/O pad is growing rapidly as many functions/IPs are getting integrated into. There are SoCs where four or more functions/peripherals are multiplexed on a single chip pad as the application usage of each IP/function is mutually exclusive in nature. This pin multiplexing option changes across SoCs and also in between different releases for the same SoC. Over time, it is being observed that this I/O pad multiplexing becomes major bottleneck in reusing/porting/creating testbench and verification environment across SoCs and releases. In this paper, we are proposing a new testbench layer (i.e “Function mapping”) in between the device under verification (DUV) and the various testbench components. This new layer reduces the overall verification cycle time.*

Keywords : *SoC, Verification, test bench architecture, pinsheet (pinmux), bus functional model(BFM), Monitor, Driver, IP (intellectual property).*

I. INTRODUCTION

Now a days with submicron technology the area of semiconductor soc has become extremely small ,so which leads to design of multimillion gates logic into a few mm square area .the SoC's also have been embedding lots of IP's into a single chip, so the functionality of these SoC's includes the connectivity(with multimedia,Gips,GPS,Blutooch,etc) multimedia processing, connectivity and multimedia devices(like display, camera and process with 2D,3D graphics ,video compression algorithm, touch screen control).So considering the enormous functionality the soc can have quickly became very demanding small area with handheld devices. Although the

challenge of high logic functionality embedding in a single monolithic platform has been solved with miniaturized Gate channel width but the challenge of connectivity very high number of peripherals to chip still remains In summary soc have become more pin constraint than area constrained, so there is a strong need of pin multiplexing in soc. This also raise the big challenge in terms of verification of multiple functionality embedded in a single submicron monolithic IC.The another major challenge in the design of complex soc is frequent change of pin multiplexing specification. Functional verification consumes almost 75% of design resources [1] in a typical SoC design cycle and as the design is becoming increasingly complex, SoC verification is the most challenging task. There is always a limitation on number of I/O pads a SoC can avail and this is to save die area and cost per device. In the present generation SoC design, there is strong focus on re-use of existing cores/IPs.The pinmux option can vary from four or more functions on a single I/O pad along with the possibility of a [2] secondary level of muxing on each of these muxed functions. This pinmux is one of the important specifications of the SoC design. The changes in pinmux directly impact the SoC verification environment across SoCs and across releases of same SoC. This paper proposes a method, which generates verification environment from this specification automatically, which helps to meet the challenge of incremental verification [1], dynamic verification [3] etc.

II. a) LIMITATIONS OF TRADITIONAL APPROACH

Several approaches [5][6][7][8] discuss the challenge and scope of reusing SoC verification. Re-use is must to make low time to market. Figure 1 depicts standard testbench architecture [3] where SoC I/O pads are directly connected with drivers/monitors/BFMs [4]. As shown in Fig.1 the monitors/BFMs and drivers are directly interfaced with the DUT through top level SoC I/O pads. Now in the next SoC or in the same SoC with a newer version of pinmux sheet, when this pinmuxing option is

changed and SPI is muxed with some other peripherals (IPs), then the above pin connection have to be modified. Such changes have to be done for all the muxed peripherals manually; hence maximum time will be consumed so it has high time to market. This becomes major bottleneck for SoC verification where the multiple revision of pinmux is quite obvious in today's SoC, which is designed in competition driven market place.

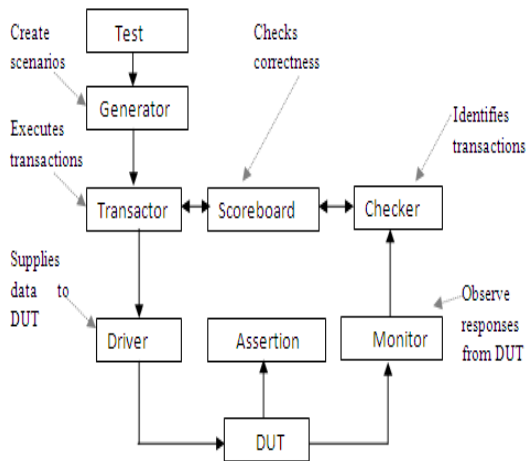


Fig 1: Standard traditional testbench architecture

b) PROPOSED APPROACH

Fig. 2 shows that one side of the new layer is connected with different BFM's and other side is connected with the DUT I/O pads. When the "pinmux" option changes, the different crossbar connections of Function Mapping layer changes accordingly and this is done automatically.

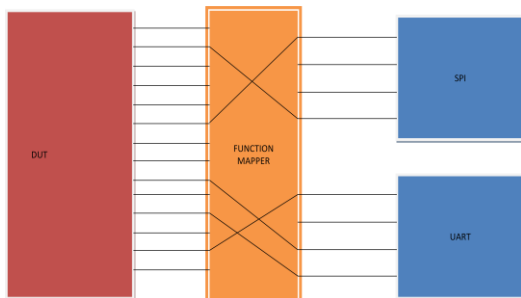


Fig 2: Function mapping layer connected with different BFM's and DUT

c) FUNCTION MAPPING LAYER

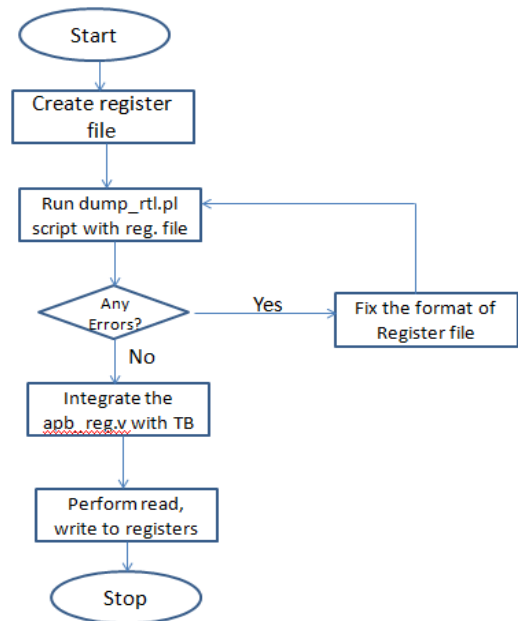


Fig 3: Flowchart to automatically generate SoC verification environment from pinmux sheet

The flowchart mentioned in Fig.3 shows the process for the generation of 'Function Mapping' layer inside the testbench. An automated script/tool is run with the pinmux sheet as an input. Based on the intelligence imbibed inside the script/tool, an analysis of the format of the pinmux sheet is done. The script generates a set of output files. These generated files are simple verilog files, which are directly portable into any testbench environment.

III. IMPLEMENTATION

The proposed methodology gives the novel solution for efficient and faster verification of pin multiplexing logic at the soc level. In this project an automated flow can be build which takes customer input as in a simple text,xls format and generates a multiplexing verification suite within short time.

The project work involves

1. The automation using perl to generate pin multiplexing verification suite.
2. Generation of control logic pin mux.
3. Design of four peripherals (SPI,RS2 32,I2C,GPIO) pin multiplexing and verifying the same.
4. Taking new customer change and demoing that it can be verify in few weeks.
5. Also designing SPI,RS 232,I2C,GPIO Bus function models.
6. UVM verification methodology.

Designing so can check transaction, monitor and also automatically showing the entire functionality is covered by code coverage and function coverage methodology.

a) SYSTEM ARCHITECTURE

The below fig.4 shows the complete architecture of the pin multiplexing system of system on chip which contains the sub blocks and details of each module is explained below.

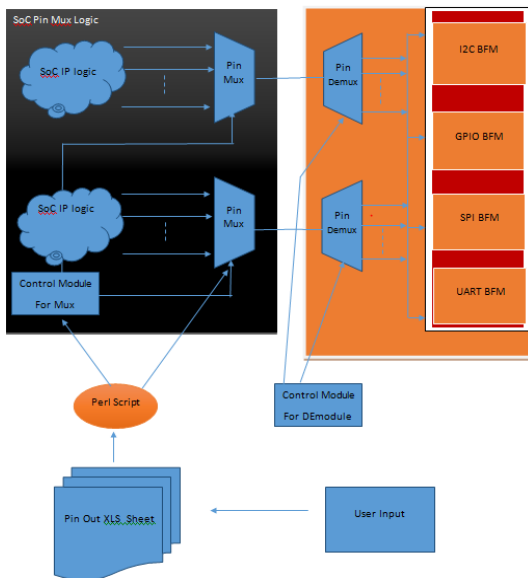


Fig 4: system Architecture showing pin muxing

- User input: user can give the required pin for selecting the any required ip for verification.
- Pin out xls sheet: which gives the information about the pin connectivity that are entering into the xls sheet from the xls sheet and read the pin selection.
- Perl script: This will take the input file from the pin out xls sheet and read the complete file and it will generate the required condition to the control module.
- Control module for multiplexer: this is the selection part of the complete module, here it will helps to select the required ip which is coming from the perl scripts read file.
- Soc Ip logic: which contains the all soc module like SPI, I2C, UART, GPIO are at one module so for that the verification of the each module is very difficult to maintain so the pin multiplexing is used to maintain the configuration of each module.
- Demultiplexer: This demultiplexes the signal from the multiplex module and gives the required selection to the bus functional module.

IV. SIMULATION RESULTS

a) SPI

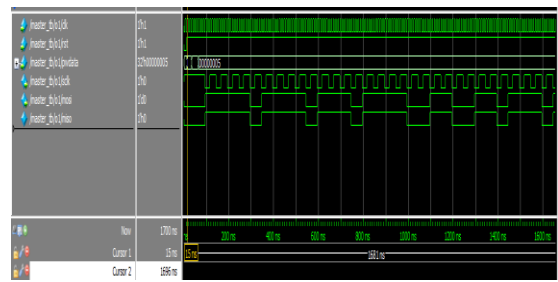


Fig.5 SPI waveforms

b) UART

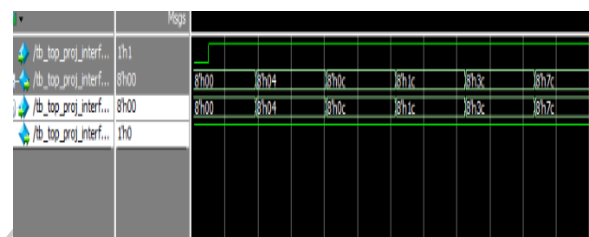


Fig.6 UART waveforms

c) AMBA APB

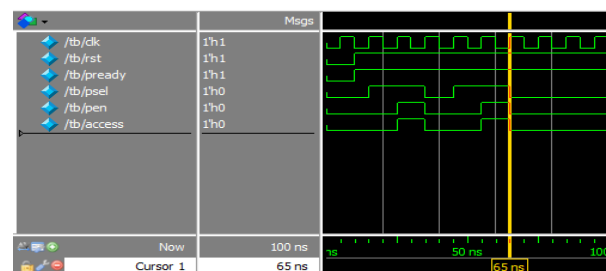


Fig.7 APB waveforms

V. CONCLUSION AND FUTURE WORK

In this stage SPI, UART modules and APB interconnect that uses perl script for control module generation have designed and results have been verified using model sim. And for the given input correct transaction can be observed.

In the next stages I2C, GPIO modules along with these Bus function modules will be developed and verifying the same using UVM (Universal

verification methodology). Using Xilinx software modules will be dumped into FPGA kit to observe the transactions and percentage of coverage (Functional and code coverage).

REFERENCES

- [1] The international technology roadmap for semiconductors: 2011 edition, design, <http://www.itrs.net>.
- [2] P1021 product description and documentation http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=P1021.
- [3] Hunter, A., Piziali, A.; Ziv, A.; Larson, K.; Hemmady, S. "Ensuring Functional Closure of a Multi-core SoC through Verification Planning, Implementation and Execution" Microprocessor Test and Verification, 2008. MTV '08. Ninth International Workshop, Product Type: Conference Publications, Page(s): 7 - 13, Date of Conference: 8-10 Dec. 2008.
- [4] VMT BFM models from Synopsys Inc, <http://www.synopsys.com/dw/dwlibdocs.php>.
- [5] Rui Want, Wenfa Zhan, Guisheng Jiang, Minglun Gao, Su Zhang, "Reuse issues in SoC verification platform", Computer Supported Cooperative Work in Design, 2004, Conference publication, pages: 685-688, Date: 26-28 May 2004.
- [6] Bamford, Noah et al., "Challenges in System on Chip Verification", Microprocessor Test and Verification, 2006. MTV '06. Seventh International Workshop, Conference Publications, Page(s): 52 - 60, Date of Conference: 4-5 Dec. 2006.
- [7] M. Keating and P. Bricaud, "Reuse Methodology Manual for System on-a-Chip Designs," Kluwer Academic Publishers, 1999.
- [8] Saleh, R. et al., "System-on-Chip: Reuse and Integration", Proceedings of the IEEE, **Volume:** 94, Issue: 6, **Page(s):** 1050 - 1069, **Product Type:** Journals & Magazines, Date of Publication: June 2006.
- [9] UVM world <http://www.uvworld.org>.