ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Hardware-Software Co-Verification and FPGA Prototyping of OBC-2 ASIC

Prachi K Amin[1], Tanna Jaimin[2], Rahul Dhingani[3], Himanshu Patel[4]

M.E. Student, Dept. of VLSI & ESD, GTU PG SCHOOL, Gandhinagar, Gujarat, India[1]

Scientist/Engineer, MSCED/MRSA, Space Applications Centre (ISRO), Ahmedabad, Gujarat, India[2,3,4]

ABSTRACT: The Hardware-Software co-verification is carried out to confirm the feasibility of proposed system and to speed up the verification of ASIC as complexity of design increases. The unique advantage that FPGA Prototyping brings as a verification method is that FPGAs are able to run closer to system speed than any other verification method and can interface with wide variety of test/actual stimulus. This Paper defines Hardware-Software co-verification and FPGA Prototyping of On-Board Controller-2 (OBC-2) ASIC. OBC-2 ASIC is based on DW8051 embedded micro-controller soft core surrounded by various digital modules and memory blocks. Hardware Software co-verification is used for checking the functionality of each modules of OBC-2 ASIC and FPGA Prototyping is used to verify the total functionality of OBC-2 ASIC before going to production stage.

KEYWORDS: HW-SW co-verification, FPGA prototyping, OBC-2 ASIC.

I. INTRODUCTION

Today's embedded systems typically consist of both hardware and software components[1]. Embedded system design poses challenges in performance evaluation, selection of appropriate parts for system implementation, and verification of such systems for functional properties.

Various technologies have been used in the development of embedded systems like; microcontroller, DSP processor, ASIC, and FPGA[3]. The increased complexity of SoC has led to significant increase in the verification efforts that are imperative to meet the time-to-market demands[5]. Design verification which including functional and timing verification of ASIC takes the major part of design cycle So, developers are moving towards hardware-software co-verification using different methods. First is Hardware-Software co-verification based on virtual prototype machine and the other is Hardware-Software co-verification on the hardware-developed board[2]. The speed of simulation in virtual prototype machine is slow and it is also difficult to debug the software during the co-verification for SoC. The advantages of Hardware-Software co-verification on the hardware-developed board is that the verification speed is very high and its real-time response is excellent as compared to virtual prototype machine. So, Hardware-Software co-verification of SoC on hardware developed board is very efficient.

As the core is used by ASIC/SoC at present is the common CPU or DSP in industry, Hardware-Software co-verification on the hardware-developed board can be used to verify SoC. So, FPGA prototyping method comes into picture. FPGA prototyping has emerged as a major ASIC/SoC verification technology to test and verify hardware modules of ASIC/SoC very quickly and to make hardware and software operate in real time environment.

This paper is intended for efficient Hardware-Software co-verification of On-Board Controller-2 (OBC-2) ASIC is surrounded with different digital modules and memory blocks. Xilinx Virtex-5 FPGA framework is used for verification and prototyping of OBC-2 ASIC design.

This paper organized as follows: Section II is details of OBC-2 ASIC. Section III defines flow of HW-SW co-verification. Section IV shows implementation results. Section V defines conclusion.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

II. RELATED WORK

In [1], authors used a novel C simulation based hardware-software co-verification environment and co-design methodology for computer intensive applications. They modeled whole object platform based on ISS (instruction set simulator) so, it gives much flexibility in case of modification. This method is suitable for software oriented implementation that need application specific accelerated hardware due to cost requirement and timing constraints. From this paper we studied that hardware model can co-simulate with software in a uniform process to increase simulation speed. In [2], authors have compared two hardware-software co-verification methods; virtual prototype machine and hardware development board and the function of the SoC has been verified using hardware platform on ARM prototype system. We studied and compare the characteristics of both the methods. From that it is clear that because of the core is used by SoC in industry, hardware development board is the best method to verify the SoC. It provides good flexibility, real time behavior and verification speed is very high as compared to virtual prototype machine. We also studied several aspects that must be satisfied while working on ARM prototype system for verification of SoC. In [3], authors have discussed multiple scenario approaches for system prototyping on FPGA platform with large numbers of FPGAs. From this paper**,** we studied Multiple scenario for HW/SW co-verification of SoC for design preparation and important points related to design partitioning. In [4], Prototyping verification methodology of SoC and implementation of verification methodology on FPGA board for LEON3 SoC is proposed. It builds hardware platform and translate the ASIC design into FPGA platform and then does prototyping verification by using two verification methodology for single IP and whole system and implants in Linux operating system. From this**,** we studied that FPGA prototype platform solves error as soon as possible so as to detect the defect of design which cannot found before fabrication process and help to reduce the cost and speed up the design to market. In [5], authors present verification framework for IEEE 802.11ac WLAN SoC using synopsys HAPS platform with system architecture and HW/SW design flow.

III. ARCHITECTURE DETAILS

A. ASIC Details

The design of OBC-2 ASIC is targeted for space-borne microwave remote sensing payloads. The OBC-2 ASIC Architecture is based on 8-bit DW8051 embedded micro-controller soft core. The DW8051 is interfaced with other peripheral modules through Special Function Register (SFR) or Memory Bus as shown in fig 1. According to the requirement, our more focused on digital modules of ASIC.

B. Target Board Details

Xilinx Virtex-5 FPGA is chosen to prototype the OBC-2 ASIC. So, Target hardware is designed around Xilinx Virtex-5 FPGA, HCT drivers, LVCH drivers, Watch Dog Timer, RS-422 drivers & receivers, Serializer, Oscillator, MIL-STD-1553 transceiver and pulse transformer as shown in fig 2. Design of OBC-2 ASIC with surrounding modules is implemented on the target hardware through Xilinx Platform USB cable.

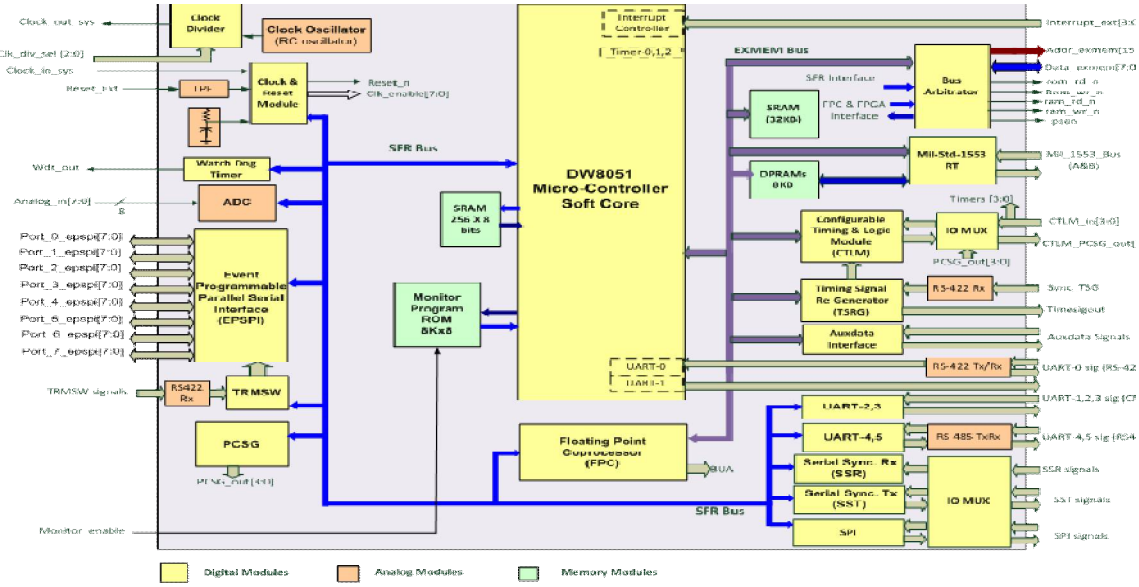


Fig 1. Block Diagram of OBC-2 ASIC

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

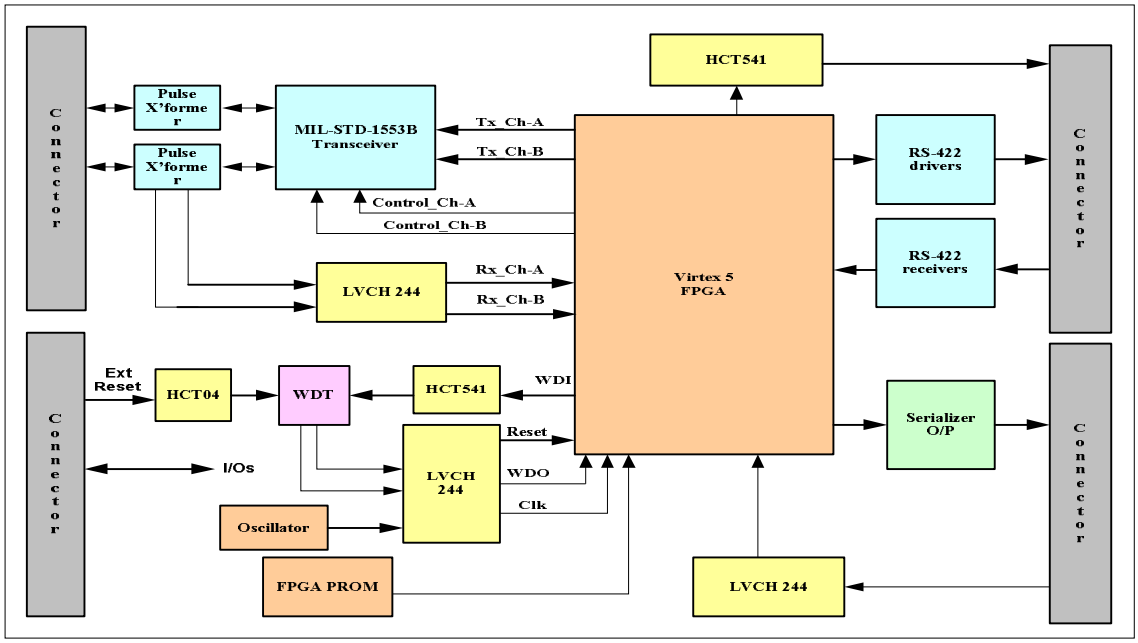


Fig 2. Block Diagram of Target Board

IV. HARDWARE-SOFTWARE CO-VERIFICATION

Proposed hardware-software co-verification flow is shown in fig 3.

A. Hardware Design Flow

OBC-2 ASIC Hardware design flow is started from HDL design entry of all digital modules in Xilinx ISE Project Navigator software tool. The generated top HDL file then synthesized and then simulation carried out using Modelsim software tool. Next is implementation of design which includes further steps which are translate, map and place & route and generation of 'mcs' file for Target Virtex-5 FPGA. That 'mcs' file is then downloaded to Virtex-5 FPGA board using Xilinx Impact software tool. The implemented circuit then can be examined using software application that will be described in next section.

B. Software Design Flow

Software design flow consists of programming in 'C' language that contain test vectors generation for different modules of OBC-2 ASIC. After writing the program in Keil compiler for different modules of OBC-2 ASIC, next is to debug, compile and build of program then generate hex file. Generated hex file is then converted to 'mif'(memory initialization file) which is Program memory compatible file and combine with Top level HDL code. Test bench performs the clock & reset generation, data generation and capturing of data.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

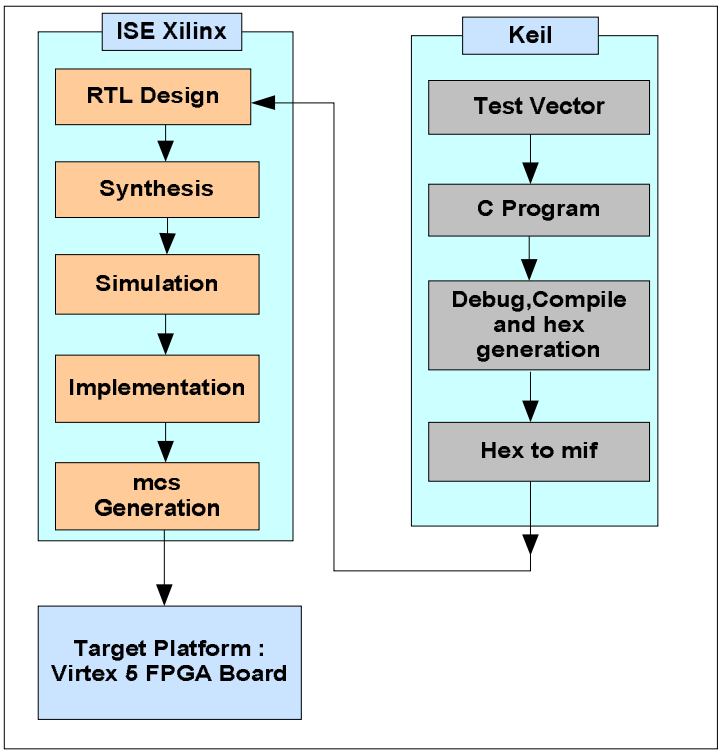


Fig 3. Hardware-Software co-verification Design Flow

V. IMPLEMENTATION RESULTS

To complete the hardware-software co-verification of OBC-2 ASIC; module level simulation and hardware implementation of Auxiliary Data Interface, Programmable Chip Select Generator, TRM Switch Control, Watch Dog Timer, Serial Peripheral Interface, Time Signal Re-Generator, Event Programmable Serial Parallel Interface, Serial Synchronous Transmitter and Universal Asynchronous Receiver Transmitter is carried out on target board successfully.

At the end, top level verification of OBC-2 ASIC by combining all modules is completed in simulation. Fig 4. Shows the simulation results of various modules of OBC-2 ASIC. The prototyping of full chip OBC-2 ASIC is completed successfully.

The design has been implemented on Xilinx Virtex-5 FPGA board. Target Hardware is running on 12 MHz Clock frequency. Fig 5 shows the Virtex-5 FPGA utilization for OBC-2 ASIC implementation. As shown in figure, design utilizes 18,835 LUTs (out of 81,920) 12,843 registers (out of 81,920) and 25 block RAMs (out of 295). And, in terms of FPGA capacity, OBC-2 ASIC design occupies 43% slices.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

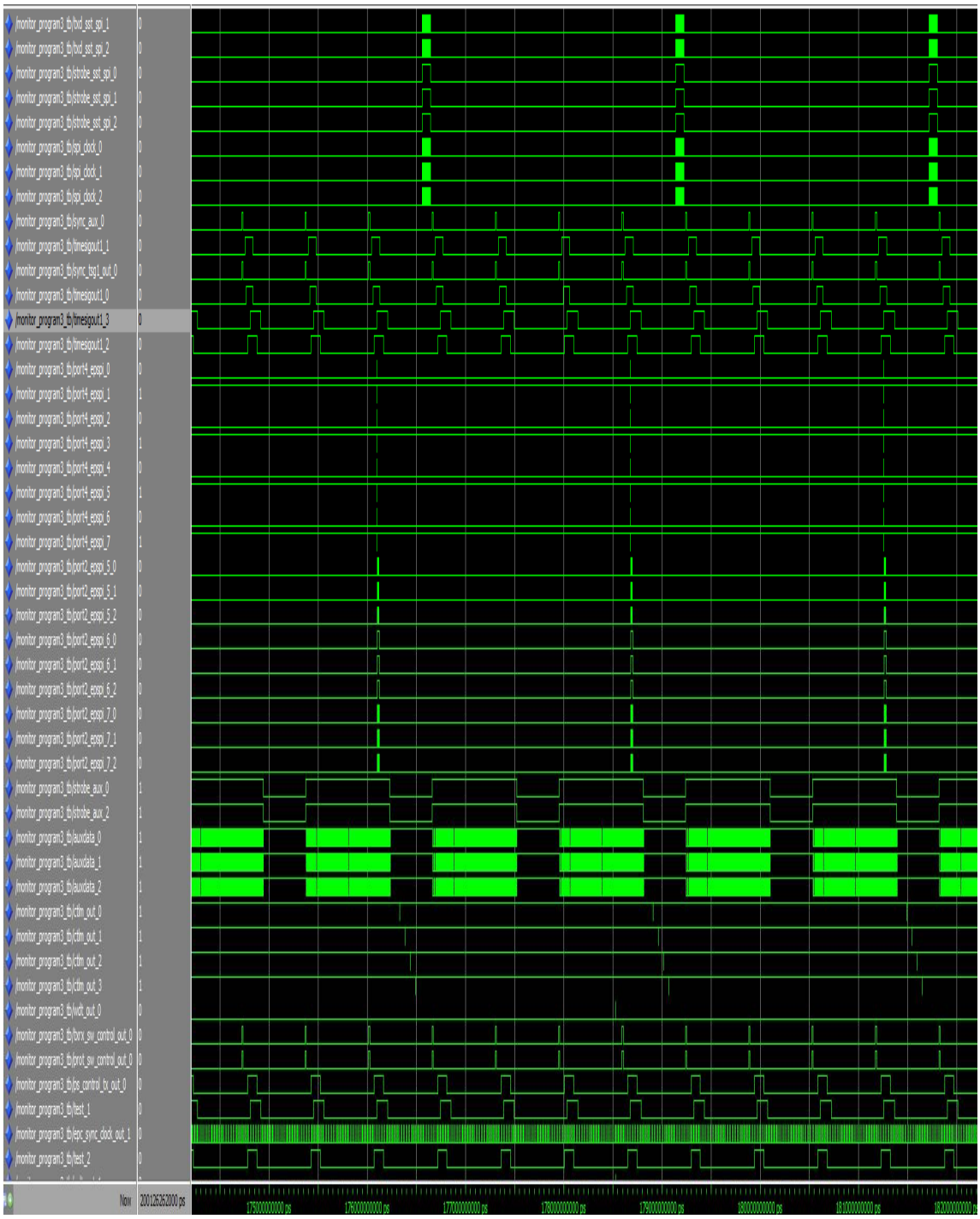


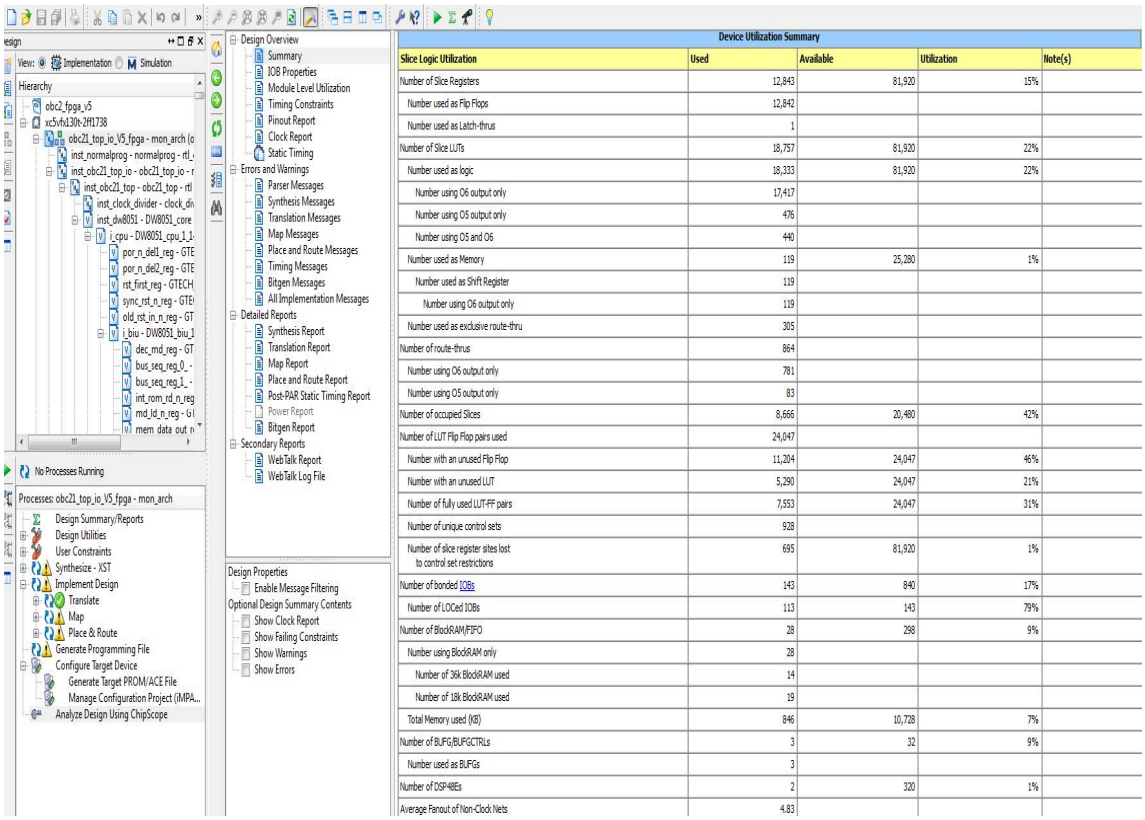
Fig 4. Simulation Result

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016



Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	12,843	81,920	15%	
Number used as Flip Flops	12,842			
Number used as Latch-thrus	1			
Number of Slice LUTs	18,757	81,920	22%	
Number used as logic	18,333	81,920	22%	
Number using O6 output only	17,417			
Number using O5 output only	476			
Number using O5 and O6	440			
Number used as Memory	119	25,280	1%	
Number used as Shift Register	119			
Number using O6 output only	119			
Number used as exclusive route-thru	305			
Number of route-thrus	864			
Number using O6 output only	781			
Number using O5 output only	83			
Number of occupied Slices	8,666	20,480	42%	
Number of LUT Flip Flop pairs used	24,047			
Number with an unused Flip Flop	11,204	24,047	46%	
Number with an unused LUT	5,290	24,047	21%	
Number of fully used LUT-FF pairs	7,553	24,047	31%	
Number of unique control sets	928			
Number of slice register sites lost to control set restrictions	695	81,920	1%	
Number of bonded IOBs	143	840	17%	
Number of LOCed IOBs	113	143	79%	
Number of Block RAM/FIFO	28	298	9%	
Number using Block RAM only	28			
Number of 36k Block RAM used	14			
Number of 18k Block RAM used	19			
Total Memory used (KB)	846	10,728	7%	
Number of BUFG/BUFGCTRLs	3	32	9%	
Number used as BUFGs	3			
Number of DSP48Es	2	320	1%	
Average Fanout of Non-Clock Nets	4.83			

Fig 5. Design Summary Report

VI. CONCLUSION

This paper proposed the hardware-software co-verification and FPGA prototyping for OBC-2 ASIC. As, increasing design complexity burdens the verification process quite a lot. So, adoption of Hardware-software co-verification provides the verification with actual application software which gives the advantage of re-usability of software which reduce the time to develop field application. Adoption of FPGA prototyping as part of the ASIC verification methodology is an economical way to supplement and extend existing functional verification methodologies. FPGA prototyping for ASIC will be reliable, smoother, run with actual stimulus & speed and save time & money. ASIC prototyping can address well in advances many risks involved with design and time to market. By first prototyping ASIC in FPGA system, an organization can then take the device to production stage in the most viable and reliable means. FPGA prototyping systems offer the advantages to deliver high performance solutions to general computing activity like; emulation of digital logic and ASIC prototyping.

REFERENCES

[1] Gu-gang Gao, Long-xing Shi, Jun Yang, Ming Gu, "Cost-effective Hardware/Software Co-verification Environment and Co-Design Method based on C Prototype", 8th International Conference on Solid-State and Integrated Circuit Technology Proceedings, pp. 2061 - 2063, 2006.

[2] Guozhang Wang, Qiaolin Shi, Zhiguo Yu, Zongguang Yu, "The study of HW/SW co-verification on ARM-prototype system", 9th International Conference onSolid-State and Integrated-Circuit Technology, pp. 1847-1897, 2008.

[3] Katona, M., Djukaric, D., Cvejanovic, D., "Multiple Scenario Approach for Pre-Silicon Hardware/Software Co-Verification", First IEEE Eastern European Conference on the in Engineering of Computer Based Systems, pp.110-119, 7-8 Sept. 2009.

[4] LigangHou, shuqin Gang, Jinhui Wang, HongueZang, XuemeiLi,"FPGA Prototyping Implementation Of LEON3 Soc", International Conference on Industrial Control and Electronics Engineering(ICICEE), pp.1643-1646, Aug.2012.

[5] Sutisna, N.,Lanante, Nagao, Y., Kurosaki, M., Ochi, H., "Live demonstration: Hardware-software co-verification for very large scale SoC using synopsys HAPS platform",IEEE Asia Pacific Conference onin Circuits and Systems (APCCAS), pp.171-172, Nov. 2014.