The International Journal Of Engineering And Science (IJES)
||Volume||2 ||Issue|| 4 ||Pages|| 48-50||2013||
ISSN(e): 2319 – 1813 ISSN(p): 2319 – 1805

Implementation of PS2 Keyboard Controller IP Core for On Chip Embedded System Applications

[1,]Medempudi Poornima, [2,]Kavuri Vijaya Chandra
[1,]*M.Tech Student* [2,]*Associate Proffesor.*
[1,2,]*Prakasam Engineering College,Kandukuru(post), Kandukuru(m.d), Prakasam(d.t).*

----------Abstract----------
In many case on chip systems are used to reduce the development cycles. Mostly IP (Intellectual property) cores are used for system development. In this paper, the IP core is designed with ALTERA NIOSII soft-core processors as the core and Cyclone III FPGA series as the digital platform, the SOPC technology is used to make the I/O interface controller soft-core such as microprocessors and PS2 keyboard on a chip of FPGA. NIOSII IDE is used to accomplish the software testing of system and the hardware test is completed by ALTERA Cyclone III EP3C16F484C6 FPGA chip experimental platform. The result shows that the functions of this IP core are correct, furthermore it can be reused conveniently in the SOPC system.

Date of Submission: 8 April 2013 Date Of Publication: 25, April.2013

I. INTRODUCTION

In electronic design a semiconductor intellectual property core, IP core, or IP block is a reusable unit of logic, cell, or chip layout design that is the intellectual property of one party. IP cores may be licensed to another party or can be owned and used by a single party alone. The term is derived from the licensing of the patent and source code copyright intellectual property rights that subsist in the design.IP cores can be used as building blocks within ASIC chip designs or FPGA logic designs It IP cores in the electronic design industry have had a profound impact on the design of systems on a chip. By licensing a design multiple times, IP core licensor spread the cost of development among multiple chip makers. IP cores for standard processors, interfaces, and internal functions have enabled chip makers to put more of their resources into developing the differentiating features of their chips. As a result, chip makers have developed innovations more quickly.

II. TYPRES OF IP CORES:

The IP core can be described as being for chip design what a library is for computer programming or a discrete integrated circuit component is for printed circuit board design.

2.1 SOFT CORES:

IP cores are typically offered as synthesizable RTL. Synthesizable cores are delivered in a hardware description language such as Verilog or VERILOG. These are analogous to high level languages such as C in the field of computer programming. IP cores delivered to chip makers as RTL permit chip designers to modify designs (at the functional level), though many IP vendors offer no warranty or support for modified designs.IP cores are also sometimes offered as generic gate-level net lists. The net list is a boolean-algebra representation of the IP's logical function implemented as generic gates or process specific standard cells. An IP core implemented as generic gates is portable to any process technology. A gate-level net list is analogous to an assembly-code listing in the field of computer programming. A net list gives the IP core vendor reasonable protection against reverse engineering.Both net list and synthesizable cores are called "soft cores", as both allow a synthesis, placement and route (SPR) design flow.

2.2 HARD CORES:

Hard cores, by the nature of their low-level representation, offer better predictability of chip performance in terms of timing performance and area.Analog and mixed-signal logic are generally defined as a lower-level, physical description. Hence, analog IP (SerDes, PLLs, DAC, ADC, etc.) are provided to chip makers in transistor-layout format (such as GDSII.) Digital IP cores are sometimes offered in layout format, as well.Such cores, whether analog or digital, are called "hard cores" (or hard macros), because the core's application function cannot be meaningfully modified by chip designers. Transistor layouts must obey the target foundry's process design rules, and hence, hard cores delivered for one foundry's process cannot be easily ported to a different process or foundry. Merchant foundry operators (such as IBM,Fujitsu, Samsung, TI, etc.) offer a variety of hard-macro IP functions built for their own foundry process, helping to ensure customer lock-in.

III. ALTERAS HW/SW DESIGN FLOW:

Generally the design flow starts in the Quartus II software by making a new project, subsequently, from the Quartus II the SOPC Builder tool can be accessed.In SOPC Builder the system is put together with ready intellectual property (IP)cores and/or user made IP cores. After the system is generated by the SOPC Builder tool, once again, the Quartus II software is used to add additional block s to the system, if needed. Before the system is synthesized pin assignment is made (this is also done in the Quartus II software). After synthesis and fitter operation, the Quarus II Programmer tool is used to configure the FPGA device. For software development the Nios II IDE is employed. When software debugging the GNUPro tools are used, these tools are integrated in the Nios II IDE. If real-time systemlevel debugging is required, a tool named SignalTap II logic analyzer can be utilized, thistool can be accessed from the Quartus II.

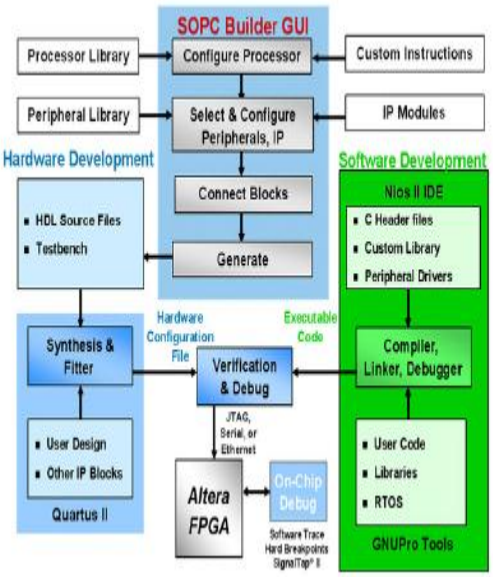

Fig 1 Software Architecture

IV. SYSTEM GENERATION IN SOPC BUILDER

The project starts by instancing the CPU, Altera´s Nios II CPU is used. The Nios II processor family consists of three cores–fast (Nios II/f), economy (Nios II/e), and standard (Nios II/s) cores – each optimized for a specific price and performance range. All three cores feature a general-purpose RISC CPU architecture and share a common 32-bit instruction set architecture. The Nios II/e is used in this project due to the fact that it is optimized for minimum logic usage, it uses 600-700 logical elements (LEs) and two M4ks. Next step is to instance the on-chip RAM and the JATG UART. In the on-chip RAM settings, 32 bits memory width and total size of 20kByte is set, the on-chip RAM is built up of M4k memory blocks. The JATG UART Read/Write FIFO is set to be constructed of registers instead of M4ks, this option is set to save M4ks, a other method to save M4k memory blocks is to reduce the Read/Write FIFO.

V. ONCHIP EMBEDDED SYSTEM

This chapter discusses how HW/SW embedded systems are built of reusable building blocks or intellectual property (IP) within Alteras design environment. In addition software development, performance measurements and debugging is described and illustrated. The chapter starts by describing how a simple basic embedded system, consisting of a CPU, Onchip RAM and a UART, is developed. In order to utilize the systems correct function a simple application is written and ran. In Section 3.2. debugging support is added to the platform and its functions are explained. Section 3.3. describes how a high resolution timer is integrated into the system and how it is configured, with the purpose of measuring and analyzing software performance. There is also a description on how the timer is used in applications, to measure the duration of several executions. In section 3.4. the system is extended by adding PIOs (Paralell Input Output), which in this case are used to control four LEDs with a push-button. Section 3.5. shows how the performance in the system can be enhanced, by migrating software functions to hardware. In section 3.6. the platform is further extended by integrating a RTOS kernel. The section illustrates and describes how the RTOS kernel is integrated into the system, both in HW and SW, and how it can be used in application development. In this section an external SRAM is integrated in the system. The section describes how the interface to the of-chip SRAM is connected to the Avalon data bus. Subsequently a application that verifies the external SRAM is developed.

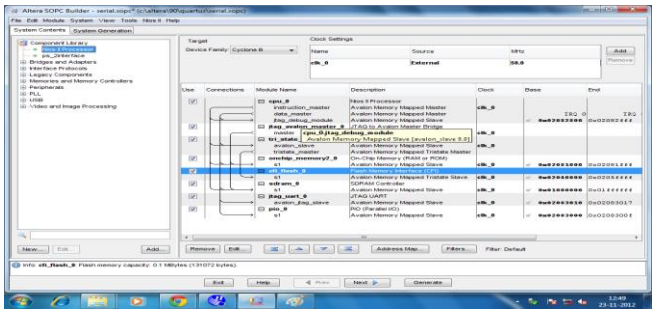

Fig 2 System development:

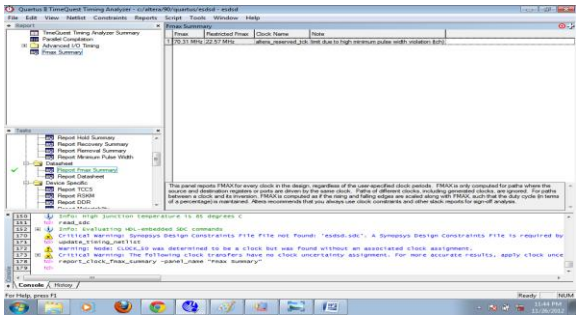

Fig 3 Performance report

V. CONCLUSION

The capabilities of FPGAs have increased to the level where it is possible to implement a complete computer system on a single FPGA chip. The main component in such a system is a soft-core processor. The Nios soft-core processor is intended for implementation in Altera FPGAs. In this thesis project a Verilog implementation of the Nios architecture has been developed, called Nios. Performance of Nios has been investigated and compared to that of the Altera Nios. Performance analysis has shown that, although the Altera and Nios implementations are quite different, their performance is similar. Design analysis has shown that although there is little room for improvement in the current Nios organization, there is a lot of room for experimenting with different processor organizations and architectural parameters.

REFERENCE

[1] X. Wang and S.G. Ziavras, "Parallel LU Factorization of Sparse Matrices onFPGA-Based Configurable Computing Engines," *Concurrency and Computation:Practice and Experience*, vol. 16, no. 4, (April 2004), pp. 319-343.

[2] Altera Corporation, "Nios Embedded Processor System Development," [OnlineDocument, Cited 2004 February 2], Available HTTP: http://www.altera.com/products/ip/processors/nios/nio-index.html

[3] Xilinx, Inc., "MicroBlaze Soft Processor," [Online Document, Cited 2004 February2],vailableHTTP: http://www.xilinx.com/xlnx/xil_prodcat_product.jsp?title=microblaze

[4] K. Compton and S. Hauck, "Reconfigurable Computing: A Survey of Systems andSoftware," *ACM Computing Surveys*, vol. 34, no. 2 (June 2002), pp. 171-210.

[5] R. K. Gupta and Y. Zorian, "Introducing Core-Based System Design," *IEEE Design andTest of Computers*, vol. 14, no. 4 (October-December 1997), pp 15-25.

[6] Opencores.org Web Site, [Online Document, Cited 2004 February 9]

[7] Altera Corporation, "Excalibur Devices," [Online Document, Cited 2004 February 7],Available HTTP: http://www.altera.com/products/devices/arm/arm-index.html

[8] Xilinx, Inc., "PowerPC Embedded Processor Solution," [Online Document, Cited 2004February 7], Available HTTP: http://www.xilinx.com/xlnx/xil_prodcat_product.jsp?title=v2p_powerpc

[9] Xilinx, Inc., "MicroBlaze Processor Reference Guide," [Online Document], 2003September, [Cited 2004 February 2], Available HTTP: http://www.xilinx.com/ise/embedded/mb_ref_guide.pdf

[10] Xilinx, Inc., "PicoBlaze 8-Bit Microcontroller for Virtex-E and Spartan-II/IIE Devices,"[Online Document], 2003 February, [Cited 2004 February 2], Available HTTP:http://www.xilinx.com/bvdocs/appnotes/xapp213.pdf

[11] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*,Kluwer Academic Publishers: Norwell, MA, 1999.

[12] Altera Corporation, "Stratix Device Handbook," [Online Document], 2004 January, [Cited 2004 February 3], Available HTTP:http://www.altera.com/literature/hb/stx/stratix_handbook.pdf