International Journal of Engineering & Technology, 7 (2.16) (2018) 57-60

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

FPGA Prototyping of Micro-Blaze soft-processor based Multi-core System on Chip

G. Prasad Acharya[1]*, M. Asha Rani[2]

[1]Sreenidhi Institute of Science and Technology, Yamnampet, Hyderabad, Telangana, India
[2]JNTUH College of Engineering, JNTUH University, Kukatpally, Hyderabad, Telangana, India
**Email: gpacharya@sreenidhi.edu.in*

Abstract

The increased demand for processor-level parallelism has many-folded the challenges for SoC designers to design, simulate and verify/validate today's Multi-core System-On-Chip (SoC) due to the increased system complexity. There is also a need to reduce the design cycle time to produce a complex multi-core SOC system thereby the product can be brought into the market within an affordable time. The Computer-Aided Design (CAD) tools and Field Programmable Gate Arrays (FPGAs) provide a solution for rapidly prototyping and validating the system. This paper presents an implementation of multi-core SoC consisting of 6 Xilinx Micro-Blaze soft-core processors integrated to the Zynq Processing System (PS) using IP Integrator and these cores will be communicated through AXI bus. The functionality of the system is verified using Micro-Blaze system debugger. The hardware framework for the implemented system is implemented and verified on FPGA.

Keywords: *Multi-core System-on-chip, FPGA, Micro-Blaze, IP integrator*

1. Introduction

Today's demand for high speed, high system integration and low power consumption in electronic systems are well complemented by the development of programmable devices like CPLDs and FPGAs. The computational speed of a processor can be enhanced in two ways (i) Instruction-Level Parallelism (ILP) and (ii) Thread-Level Parallelism (TLP). The single processor based system utilizes ILP uses instruction pipeline that fetches instruction while previous instructions are decoded and executed. The pipelining and superscalar architectures are based on ILP. The multiprocessing and/or multitasking processors utilize multiple processing elements to perform the given task in parallel. These processors use the TLP mechanism to enhance the speed of execution speedup. The Latest FPGA devices provide a prototyping platform [2-3] for highly parallel and distributed Multi-core System on Chip (SoC) that exploits processor-level (hardware) parallelism. Due to its ability of utilizing multiple processors for parallel execution of given task, the multi-core SoC may work with lower frequencies thereby reducing the power consumption without any degradation on system performance.

The FPGA based multi-core SoCs [1,4,5] are developed using a new approach called Hardware/Software Co-Design approach that allows greater design flexibility and scalability, reduced design cycle time and cost. Due to these advantageous features, the number of researchers working on this design approach has been increased significantly. These MPSoC architectures are broadly categorized as heterogeneous and homogenous architectures. The heterogeneous architecture integrates a number of processors with different capabilities on a single chip. Each processor is dedicated for performing a specialized functionality thereby improving the overall system performance. The homogeneous architecture integrates a number of identical processors onto a single integrated circuits die that allows execution of various data independent task in parallel.

This paper is organized as follows: The concept and the architecture of homogenous multiprocessor system are presented in section 2. Section 3 describes the implementation details of Micro-Blaze based multiprocessor system while Section 4 presents the Hardware/Software Co-design of the proposed multi-core SoC. Section 5 presents an application that was developed to verify the functionality in both the platforms: FPGA board and SDK platform. Finally the conclusion and proposed further extension of the work is presented in Section 6.

2. Multi-core System on Chip Architecture

The performance of a processor has traditionally been enhanced by increasing the processor's operating (clock) frequency. This approach has been stalled over the years as it increases the processor's power consumption as well as the core complexity and clock issues with multiple clock domains because the various components of the processor need to operate at different clock frequencies. The other approach for maximizing the performance of a microprocessor involves Instruction-Level Parallelism (ILP).

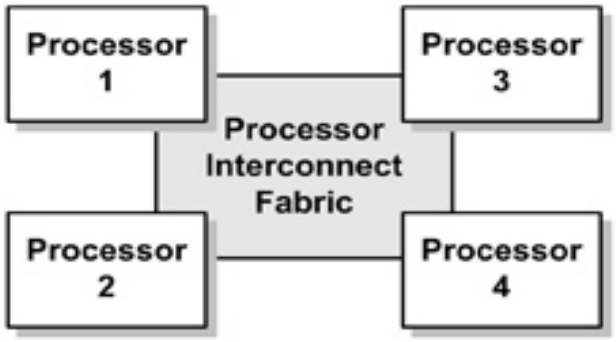
2.1 Instruction-Level Parallelism and Thread-Level Parallelism

The ILP facilitates parallel execution of instructions in at software-level and hardware-level. The software-level parallelism utilizes instruction pipeline that fetches instructions from memory while previous instructions are being decoded and executed by other units of the processor. This kind of architecture is also referred as pipelined architecture. The hardware level works upon dynamic parallelism wherein the CPU decides at run time the instructions that can be executed in parallel. This kind of architecture is referred as superscalar architecture.

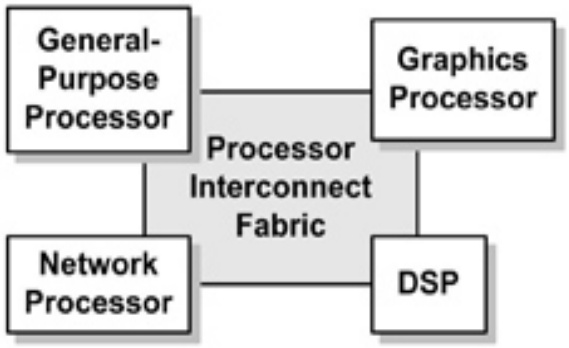
The increased system complexity and data dependencies issues of ILP has led to the development of multi-core SoC which integrates multiple cores/processors onto a single processor die that exploits Thread (Processor)- Level Parallelism. The TLP architectures utilizes multi-core/processor CPU architecture which consist of multiple processing elements, each capable of executing instructions in parallel, thereby increasing the system performance.

2.2 Homogenous vs Heterogeneous MPSoC

The multi-core processors are classified into Homogenous (symmetric) and Heterogeneous processors. The homogeneous multiprocessor architecture provides a much simpler platform on which to build applications in which all processors are identical. Figure 1 (a) demonstrates the architecture of homogenous multi-processor system which consists of four identical processor cores that are interconnected through the Processor Interconnect Fabric. The major advantages of homogenous multicore processor is that each core is capable of replicating any other core thus by enabling load sharing among the cores thereby providing symmetric operation.
The heterogeneous multi-processor architecture as shown in Figure 1 (b) has multiple processing cores each of which are having specialized capabilities and hence can provide opportunities to improve performance and efficiency. Since each processor in HMA is specialized for a different type of workload, this architecture may hence restrict the sharing the workloads among the available processors.



(a) Homogenous multi-core architecure



(b) Heterogenous multi-core architecure

Figure 1: Multi-core processor architectures

2.3 Architectural concepts of multi-core SoC

The development of semiconductor technology has allowed the designers to integrate multiple processor cores onto a Multi-core SoC [11]. The multi-core SoC technology provides the following architectural support by which the performance of the processor can be greatly improved within acceptable power consumption and chip-area.

Multiple cores

Most of the current general-purpose multi-core SoCsare symmetric in terms of instruction set and performance. A homogeneous architecture with shared global memory has been undoubtedly becoming popular and more effective for parallelism. In contrast to homogenous architecture, a heterogeneous architecture integrates multiple cores into a single Silicon die that may differ in at least two of the following architectural features: Instruction Set Architecture (ISA), functionality and performance. The multi-core SoCs are based on Instruction Level Pipelining (ILP) architectures wherein instructions are decoded and executed in stages so that the overall throughput can be enhanced. The effectiveness of ILP based multi-core SoC is limited by data dependencies among the instructions and the number of instructions that can be executed in parallel. These drawbacks of ILP based multi-core SoC shall be countered by simultaneous multithreading using Thread-Level Pipelining (TLP) architectures. The TLP architecture that integrates multiple processor cores is capable of executing instructions in parallel.

Interconnection Networks

Sharing a common bus by all the processors has been a common and historical way of interconnecting the individual processors in a shared memory multiprocessor. The shared bus provides a broadcast medium of communication among the processors and is responsible for cache coherency. Each processor is provided with one or two levels of local cache memory between the shared bus and the processor to keep the record of I/O traffic.

Memory Controllers

The memory controller is a separate IC or integrated with another IC that controls the transfer of data among the individual processors, shared memory and I/O devices. Dynamic Random Access Memories (DRAMs) are used for the storage of user data and the main purpose of DRAM controller is to increase the throughput of the shared memory in the TLP environment rather than low latency.

Shared Memory Support

The shared memory provides a communication platform for the processor cores among themselves by accessing the necessary data. These processor cores will be communicating over the interconnection network as discussed earlier. The shared memory space facilitates the migration from a sequential programming model to a parallel one. Most of the general-purpose SoCs today support a shared memory between cores and maintain a cache-coherent memory system. The cache coherency is the technique that enables the processors to read data from shared memory and copy it in its local cache while still maintaining consistency when shared variables are updated by some other processors.

3. Micro-Blaze Based Multi-Core SOC

This paper is aimed to design and implement a prototype of a basic Micro-Blaze based Multi-core SoC [8, 12] whose block diagram layout is as shown in Figure 2. It consists of a Zynq Processing System (PS) and six Micro-Blaze soft-core IPs. The Zynq PS is a feature-rich hard dual-core ARM Cortex-A9 Application Processor Unit (APU) which can run a variety of Real-Time Operating Systems and provides a strong processor candidature in today's latest SoCs for high end embedded applications such as industrial automation, video surveillance and so on. The Micro-Blaze processor [7] is an optimized and completely programmable soft core Intellectual Property (IP) for implementing in Xilinx Field Programmable Gate Arrays (FPGAs) and is based on Reduced Instruction Set Computer (RISC) architecture. This prototype model of multi-core SoC has other basic components like Processor System Reset, Micro-Blaze Debug Module (MDM) and AXI peripheral Interconnect through which the PS, Micro-Blaze

cores and other modules will communicate with each other. The processor cores are optionally provided with associated Local Memory blocks and these memories are also optionally equipped with Error Correction Capabilities (ECC).

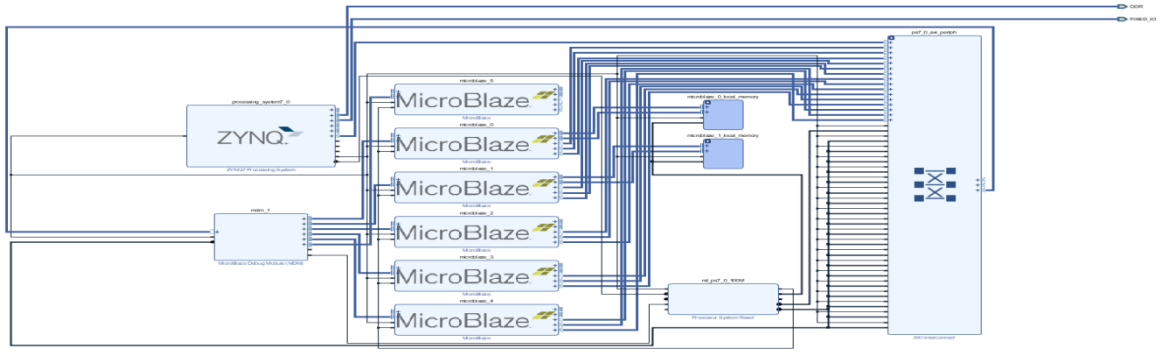


Figure 2: Proposed Xilinx Micro-Blaze soft core based MPSoC

4. Micro-Blaze Based Multi-Core SoC

The multi-core SoC architectures are developed using hardware/software co-design approach [6, 9, 13] that improves the processing speed, cost and time to market. This approach is based on the reuse of predesigned and pre-verified commercially available third party IPs that helps in improving the overall performance and system reliability. The hardware part of this approach in this work include the design of MPSoC in Zynq FPGA that utilizes the ARM Cortex-A9 MP as a hard core processor, Micro-Blaze soft core processors [13-14] and other glue logic. A software application can be developed for the hardware using any of high level programming languages C, C++. After Hardware/Software partitioning, Hardware and software parts of the co-design are developed concurrently and then integrated to meet the design functionality and specifications. The design flow for the hardware/software co-design is illustrated in Figure 3.

4.1 FPGA implementation of Multi-core SoC Hardware

The 7000-series Xilinx Zynq FPGA [10, 16] has been selected as the target device for prototyping the system. The Zynq-7000 as shown in Figure 4, integrates a powerful dual core ARM Cortex-A9 Microprocessor as a hardcore Processing System and FPGA fabric as the Programmable Logic (PL) area [17]. The PS provides a platform for software developers to develop software applications as a standalone ARM processor that can run a wide variety of operating systems. The FPGA fabric can be used to develop and implement a wide variety of hardware logic ranging from smaller designs (like adders, counters etc.) to a more complex embedded processor that may integrate one or more soft core IPs, memory IPs and I/O peripherals.

The proposed MPSoC shown in Figure 2 integrates six Xilinx Micro-Blaze soft cores in the FPGA fabric where each Micro-Blaze core will function as an individual processor. These processors will communicate with each other along with the PS and other peripherals via AXI Peripheral Interconnect. A debug module with six ports is integrated by which each of the Micro-Blaze processors can be debugged via the corresponding debug port. The block level design for the proposed MPSoC system is created using IP Integrator and validated in Xilinx Vivado environment 15]. HDL wrapper for the block design is then generated and synthesized using Vivado synthesis tool. The synthesized Vivado checkpoint is implemented in the target device and the bitstream is generated. The device (7-series Zynq FPGA) utilization summery for the implemented MPSoC is given in the Table 1.

4.2 SDK environment and Software Application

Xilinx Software Development Kit (XSDK) is available either as an integral part of Vivado Design Suite or as a separate embedded software. The XSDK provides an Integrated Design Environment for software developers to create embedded applications on MPSoC developed using Xilinx's All Programmable SoCs and Micro-Blaze soft-core microprocessors. In the present work, we have developed an embedded software application to implement an arithmetic and logic unit (ALU). We have run this application in all the integrated Micro-Blaze cores one after another. The application program has been debugged using an integrated Micro-Blaze Debug Module (MDM) and verified its functionality. All the ARM CPUs and Micro-Blaze soft-processors can be debugged in a single debug session using a single JTAG cable.

Figure 5 shows the snapshot of debugging process for the ALU functionality written as application software that can be run either on any or all of the Micro-Blaze processors. The figure shows the debugging of the program on MicroBlaze0 wherein num and num2 are the inputs for the ALU and the variable are labeled as alu_output_add, alu_output_sub, alu_output_mult and alu_output_div are the outputs of the ALU for addition, subtraction, multiplication and division respectively.

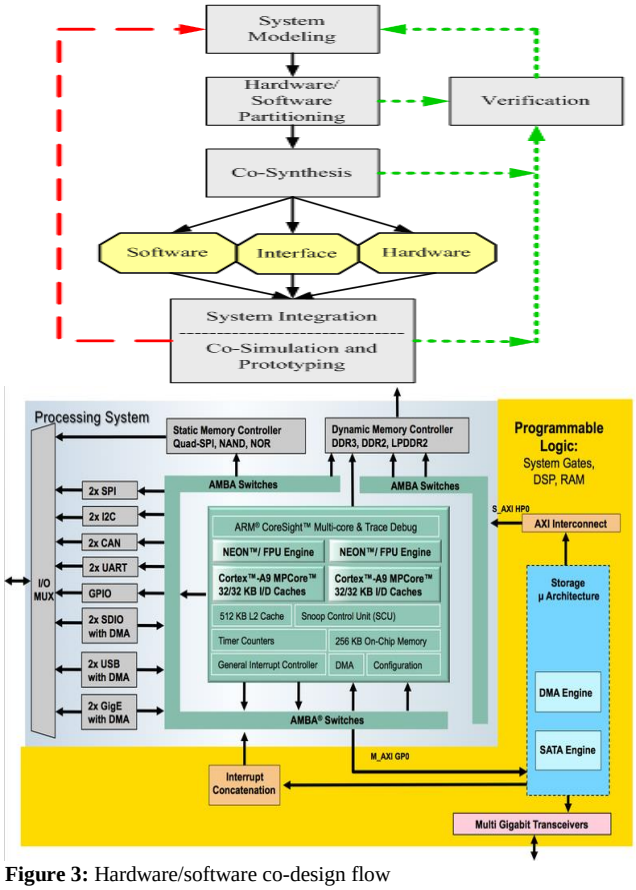


Figure 3: Hardware/software co-design flow

Figure 4: Zynq® 7000 All Programmable SoC Architecture (Courtesy: www.xilinx.com)

Table 1: Device Utilization Summery

S. No	Resource type	Available	Used	% of utilization
1	LUT as Logic	53200	15678	25.14
2	LUT as memory	17400	2301	13.22
3	Slice registers	106400	17391	13.22
4	Multiplexers	26600	683	2.57
5	Block RAMs	140	54	38.57
6	DSPs	220	6	2.73
7	Bonded IOBs	200	21	10.5

4. Conclusion and further work

The work presented in this paper has demonstrated the use of Xilinx's Vivado Tool as Computer Aided Design (CAD) tool for Hardware/Software Co-design of MPSoC. The hardware part of the proposed multi-core SoC is implemented on Zynq FPGA by integrating Zynq (Dual Core Cortex-A9) hard-core processor and six Micro-Blaze soft-core processors into a single platform. The software application for the target device is also developed concurrently. Both the hardware and the software parts are then integrated and debugged using Xilinx SDK tool. The work also demonstrates that the Hardware/Software Co-design approach not only improves the performance of the system but also the design cycle time and the cost can be reduced to a greater extent.

The work is further being carried out to implement a Thread-Level Parallelism (TLP) to enhance the execution speed of a program by splitting a task into multiple threads and to run them concurrently on different processors, which is the primary objective of any MPSoC. The work will further be extended to develop a self-testable Multi-core SoC using self-checking checkers so that the processor will have the self-checking ability. The self-testable multi-core SoC when equipped with redundant processors and self-testing capability can be deployed in critical applications like avionics, defense and satellites so that it can self-check and – repair in case of any faults.

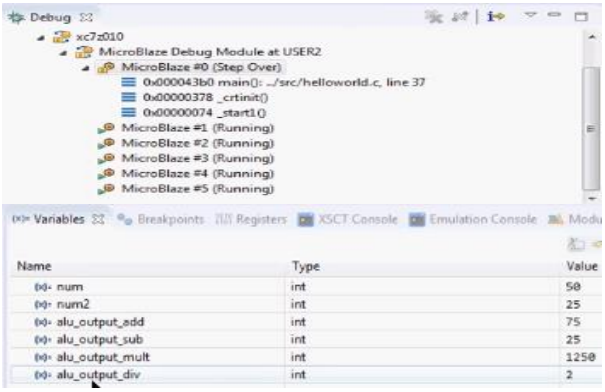


Figure 5: MDM debug window snapshot

References

[1] VasanthAsokan , "Designing Multiprocessor Systems in Platform Studio", Xilinx, WP262(v2.0), 21 November 2007.

[2] J. O. Hamblen, T. S. Hall, and M. D. Furman, Rapid Prototyping of Digital Systems, SOPC edition, Springer, 2008.

[3] ZainalabedinNavabi, Digital Design and Implementation with Field Programmable Devices, Springer, 2005.

[4] TahoDorta, Jaime Jimenez, Jose Luis Martin and Armando Astarloa, "Reconfigurable Multiprocessor Systems: A review", International Journal of Reconfigurable Computing 2010, article ID: 570279.

[5] M. Hübner, J. Becker, Multiprocessor System-on-Chip, Springer, 2011

[6] Iulian NIŢĂ, Gabriel ZDRU, "HW-SW Co-Design of MPSoC Using FGPA IP Cores",U.P.B. Scientiffic Bulletin, Series C, Vol. 75, Iss. 1, 2013, ISSN 1454-234x, pages 135-150,

[7] Xilinx company, MicroBlaze Processor Reference Guide (v5.1) 2005

[8] P. Huerta, J. Castillo, J. I. Mártinez, and V. López, "A microblaze based multiprocessor SoC," WSEAS Transactions on Circuits and Systems, vol. 4, no. 5, Grece, 423-430.

[9] J. Teich, "Hardware/Software codesign: the past, the present, and predicting the Future," in Proceedings of the IEEE, Vol. 100, May, 2012.

[10] http://digilentinc.com/

[11] LazányiJánosGyula, FehérBéla, "Distributed Embedded System Using Single FPGA," in sborníkvedeckychpraci vy3soké skoly b

[12] El Hassan El Mimouni, Mohammed Karim, AMicroBlaze-Based Multiprocessor System on Chip for Real-Time Cardiac Monitoring, International Conference on Multimedia Computing and Systems (ICMCS), © IEEE, 2014.

[13] Pierre G. Paulin, et al, "A Multi-Processor SoC Platform and Tools for Communications Applications", In Embedded Systems Handbook, CRC Press 2004.

[14] Susan Xu and Hugh Pollitt-Smith, A Multi-MicroBlaze Based SOC System: From SystemC Modeling to FPGA Prototyping, 19th IEEE/IFIP International Symposium on Rapid System Prototyping, ©IEEE, 2008.

[15] Vivado Design Suite Tutorial- Embedded Processor Hardware Design, user guide, www.xilinx.com, 2017.

[16] Chang, Chen, Wawrzynek, John and Brodersen, "A High-End Reconfigurable Computing System" IEEE Computer Society, in IEEE Design and Test of Computers, Vol. 22, March 2005,pp. 114-125.

[17] David Pellerin and Milan Saini, "FPGAs Provide Acceleration for Software Algorithms", FPGA Journal, 2004