RESEARCH ARTICLE OPEN ACCESS

Design and Implementation of Quad Core Architecture Using FPGA

V.Prasanth[1], K.Raja Sekhar[2]
[1]M. TECH, ASSOC.PROF., HOD IN ECE, PRAGATI ENGG.COLLEGE, KAKINADA, A.P, INDIA.
[2]M. TECH STUDENT (EMBEDDED SYSTEMS), PRAGATI ENGG.COLLEGE, KAKINADA, A.P, INDIA.

Abstract
The quad processor core is a design philosophy that has become a mainstream in Scientific and engineering applications. Increasing performance and gate capacity of recent FPGA devices permit complex logic systems to be implemented on a single programmable device. The Embedded multiprocessors face a new problem with thread safety [5]. It is caused by the shared memory, when thread safety is violated the processors can access the same value at the same time. Basically the processor performance can be increased by adopting clock scaling technique [4] and micro architectural enhancements. Therefore, designed a new Architecture called quad processor core architecture for SOC applications. This is implemented on the FPGA chip using VHDL.This architecture performs a simultaneous use of both parallel and distributed computing. The full architecture of a Quad processor core designed for realistic to perform arithmetic, logical, shifting and bit manipulate operations. The proposed quad processor core contains Homogeneous RISC processors [3],[7] added with pipelined processing units, multibus organization and I/o ports along with the other functional elements required to implement embedded SoC solutions. The designed Quad core performance issues like area, speed and power dissipation.

Keywords— MPSoC, Quad processor, Architecture, Embedded system design, Fpga.

I. INTRODUCTION

Quad core consists of four connected processors that are capable of communicating. This can be done on a single chip where the processors are connected typically by a multibus. Alternatively, the multiprocessor system can be in more than one chip, typically connected by some type of bus, and each chip can then be a multiprocessor system. A third option is a multiprocessor system working with more than one computer connected by a network, in which each computer can contain more than one chip, and each chip can contain more than one processor. Most modern supercomputers are built this way.

A parallel system is presented with more than one task, known as threads. It is important to spread the workload over the entire processor, keeping the difference in idle time as low as possible. To do this, it is important to coordinate the work and workload between the processors. Here, it is most important to consider whether or not some processors are special-purpose IP cores. To keep a system with N processors effective, it has to work with N or more threads so that each processor constantly has something to do. Furthermore, it is necessary for the processors to be able to communicate with each other, usually via a shared memory, where values that other processors can use are stored. This introduces the new problem of thread safety. When thread safety is violated, two processors (working threads) access the same value at the same time. Consider the following code representation:

A=A+1

When two processors P1 and P2 execute this code, a number of different out come may arise due to the real fact that the code will be split into three parts.

L1: get A;

L2: add 1 to A;

L3: store A;

It could be that P1 will first execute L1, L2 and L3 and afterward P2 will execute L1, L2 and L3. It could also be that P1 will first execute L1 followed by P2 executing L1 and L2, giving another result. Therefore, some methods for restricting access to shared resources are necessary. These methods are known as Thread safety or synchronization. Moreover, it is necessary for each processor to have some internal memory, where the processor does not have to think about thread safety to speed up the processor. As an example, each processor needs to have a private stack. The benefits of having a Multiprocessor are as follows:

1. Faster calculations are made possible.
2. A more responsive system is created.
3. Different processors can be utilized for different tasks.

In the future, we expect thread and process parallelism to become wide spread for two reasons: the nature of the applications and the nature of the operating system. The Researchers proposes two alternative micro architectures that exploit multiple threads of control. i.e. simultaneous multithreading (SMT) and chip multiprocessors (CMP).Chip multiprocessors (CMPs) use relatively simple single-thread processor cores that exploit only moderate amounts of parallelism within any one thread, while executing multiple threads in parallel across multiple processor cores .Wide-issue superscalar processors exploit instruction-level parallelism (ILP) by executing multiple instructions from a single program in a single cycle. Multiprocessors (MP) exploit thread-level parallelism (TLP) by executing different threads in parallel on different multiprocessors.

II. QUAD CORE ARCHITECTURE

The Quad processor architecture is completely designed after implementing towards the embedded concurrent processor. The concurrent processor focuses more on the interaction between tasks, correct sequencing of the interactions or communication between the tasks, and the coordination of access to their resources to input and output devices are the key concerns during the design of concurrent computing system. For this the concurrent components like SIMD array, mapping and duplicate memories are added to each processor as shown in Fig.i.

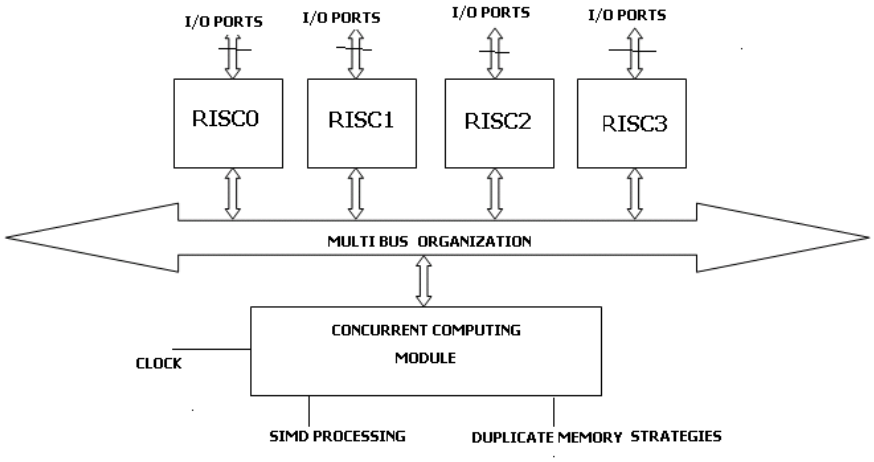

Fig.i: Designed QUAD CORE Architecture

These modules independently work as a quad core that involves the both parallel and distributed concurrent strategy.

III. DESIGN

The proposed quad core executes the multiple tasks. These multi tasks may be implemented by the Height tree evaluation technique and reordering of instruction Execution is necessary for this proposed architecture. The Clock is the heart beat of any processor. The processor executes one instruction within one clock period. The quad core is responsible for many concurrent computing operations and it consists of three main modules. These are Core processor, Quad RISC Processor, I/O Ports. Generic RISC Processor are called scalar RISC Processor because they are designed to issue one instruction per cycle, similar to the base scalar processor. Four representatives RISC based processors from the year 1990, the sun SPARC, Intel i860, Motorola M88100, and AMD 29000.Each processor use 32 bit instruction length. The instruction set consists of 51 to 124 basic instructions. We consider these four processors as generic scalar RISC, issuing essentially only these four processors as scalar RISC, issuing essentially only one instruction per pipeline cycle. Among the four scalars RISC Processor, we choose to examine the sun SPARC and i860 architectures. The sun SPARC is derived from the original Berkeley RISC design.

The concept of parallel and distributed computing Processor Core mainly consists of SIMD array, mapping, and duplicate memories.

Duplicate memories: The most important thing in quad processor mapping is the boundaries and data flow of concurrent processors. Duplicate memories are important and make a design flexible and have a high throughput for both parallel and distributed strategies and are too efficient for concurrent Architecture.

SIMD array: This SIMD array supports the multidimensional array of data. It allows the simultaneous use of multiple processors for solving a task. An SIMD array is a synchronous array of Processing Elements under the supervision of one

control unit and all P.E's receive the same instruction broadcast from the control unit, but operate on different multiple data sets from distinct data streams. It is usually loads data into its duplicate memories before starting the computation. All these are working together as a single processor that involves both the parallel and distributed concurrent strategy.

RISC Processors (RISC): In the present work, the design of an 8-bit data width Quad Reduced Instruction Set Computer (RISC) processor is presented. It was developed with implementation efficiency and simplicity in mind. It has a complete instruction set, program memories and data memories, general purpose registers and a simple Arithmetical Logical Unit (ALU) for basic operations. In this design, most of the instructions are uniform length and similar format. Arithmetic operations are restricted to CPU registers. The Instruction cycle consists of three stages namely fetches, decode and execute. Many numbers of RISC processors give the highest performance per unit area for parallel codes. A larger number of RISC processor cores allow a fine-grained ability to perform dynamic voltage scaling and power down. The RISC processor core with a simple architecture is easier to design and functionally verify. This processor is an economic element that is easy to shut down in the face of catastrophic defects and easier to reconfigure in the face of large parametric variation.

IV. FPGA IMPLEMENTATION

Modern FPGAs are large enough to implement Quad-Processor Systems-on-Chip. Commercial FPGA companies also provide system design tools. To aid our design decisions, we devise a design methodology adapted to the specific challenges we are facing. Our methodology for deriving a digital logic Implementation of the required functionalities encompasses three steps, each of which comprises a set of choices. The criteria we employ to evaluate the final set of choices are complexity and scalability. We define the complexity of a unit by two metrics: the maximum operating frequency and resource usage. Lower complexity is better; that is, the unit has a higher frequency and consumes less resource. We define scalability as the ability of the unit to expand in a chosen dimension with a minimal increase in complexity.

To determine complexity and scalability, we examine the FPGA mapping of the design. In the first step, we start by choosing a particular architectural technique that provides a certain subset of the required functionalities. The second step considers the structural design choices of each architectural technique in terms of the temporal and spatial parallelism necessary to meet throughput and latency requirements. In the third step, we take into consideration the logic-implementation-specific choices driven by the constraints of the target FPGA chip. It is vital to note that the choices in all three steps are interdependent. For instance, an initially appealing architectural technique may require logic design choices that lead toward a prohibitively expensive implementation. In this case, the choice of the architectural technique has to be reexamined. In case there are feasible implementations of the selected architectural technique, the required performance can be achieved by balancing the trade-off between throughputs and operating frequency. Higher throughput necessitates higher design complexity. On the other hand, to achieve high operating frequency, the design complexity should be kept low. As the resulting maximum frequency is highly influenced by the CAD tools that automatically place and route the circuit.

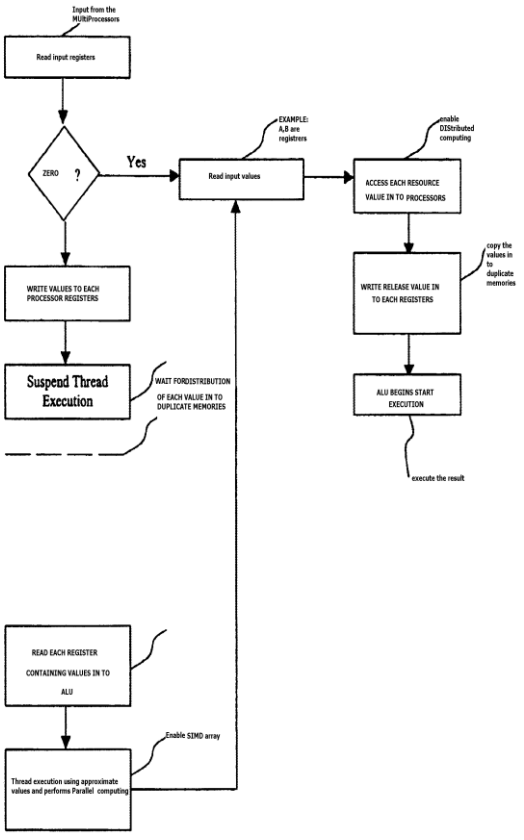

Fig.ii: Flow Chart for Implementation of Quad Core Architecture

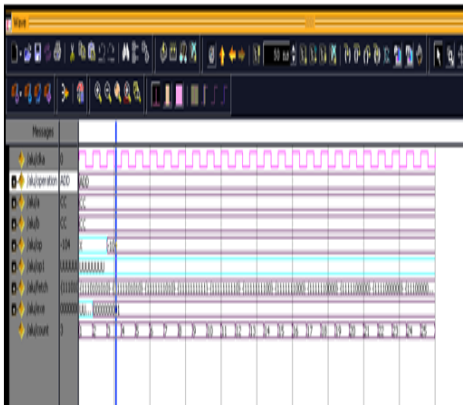

Fig.iii:Simulation results of Single ALU Executing Addition operation.

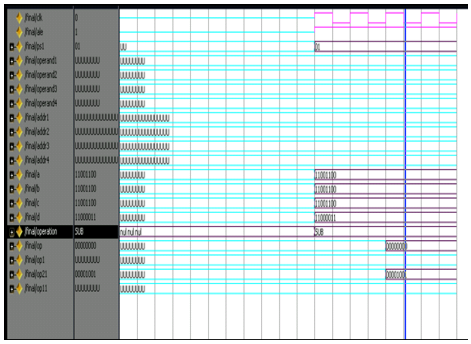

Fig.iv:Simulation results of Single ALU Executing Subtraction operation.

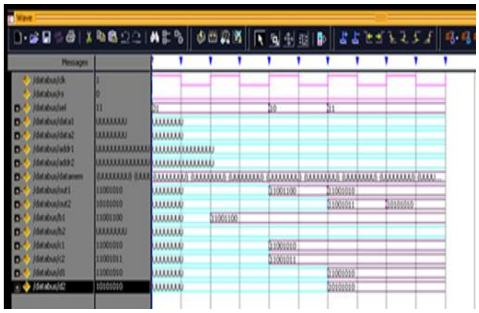

Fig.v:Simulation results of Single Processor Executing Addition operation in both parallel &distributing computing.

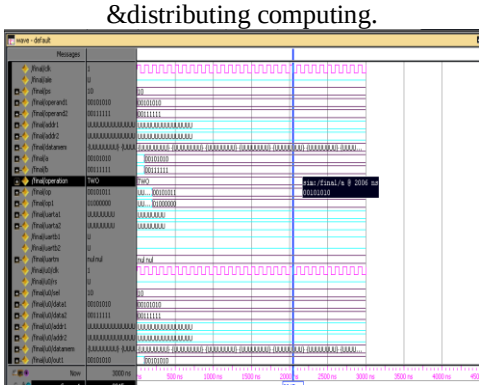

Fig.vi:Simulation results of Executing 2's complement operation.

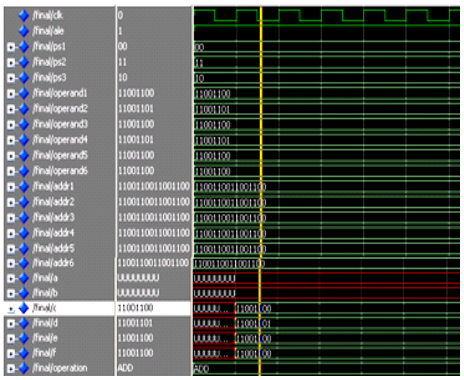

Fig.vii:Simulation results of Quad Core Processor main module

The above figures gives the results of designed quad core architecture which is simulated by using model sim simulator and verifying the parallel and distributed computing which show that the desinge dprocessor exhibits the concurrent results under the quad processorplatform.

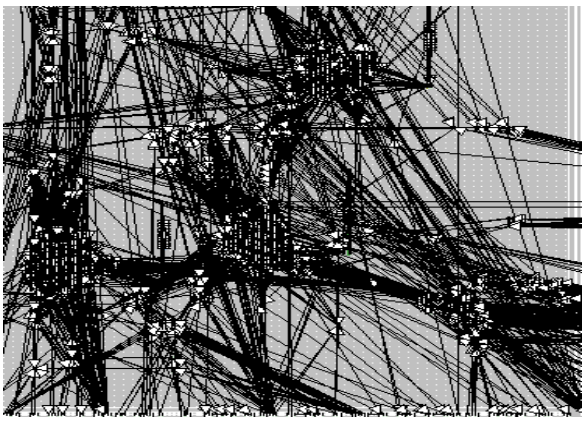

Fig.viii: Floor Planning Diagram.

Floor planning diagram of implemented design represents the total modules in the system and their internal connections of I/O ports as shown in the Fig.viii.

V. CONCLUSION

The designed quad processor core is very efficient and powerful core architecture, which is analyzed by using Xilinx software and simulation by using model sim simulator. Although the quad processor core was based on well-proven superscalar architectural techniques, including parallel execution and register renaming, these techniques were originally devised for fine-grained parallelism among instructions. The designed quad core implementation using FPGA which is very advantageous to the architecture ,because it uses the parallel processing and pipielinig techniques are easily implemented in FPGA.The concurrent operation of quad processor gives the high speed and takes less power.

REFERENCES

[1] Tai-Hua, Lu, Chung-Ho Chen, KuenJong Lee. "Effective Hybrid Test Program Development for Software-Based Self-Testing of Quad Cores" ,IEEE Manuscript received April 03, 2012, revised August 14, 2012, first published December 18, 2012.

[2] Gohringer, D., Hubner, M.Perschke, T., Becker. J. "New Dimensions for Quad core Architectures Demand Heterogeneity", Infrastructure and Performance through reconfigurability The EMPSoC Approach". In Proc of FPL 2010, PP.495-498, Sept 2010.

[3] Lysaght, P. Blodget, B. Mason, J.Young, B.Bridgford. "Invited Paper: Quad core design Methodologies and CAD Tools for Dynamic Reconfiguration of Xilinx FPGAs". In Proceedings of FPL 2009, August 2009.

[4] D. Tullsen, S. Eggers, and H. Levy, "Simultaneous Multithreading: Maximizing On- Chip Parallelism," Proc. 22nd Ann. Int'l Symp. Computer Architecture, ACM Press, New York,1995, pp. 392-403.

[5] J. lo, S. Eggers, J. Emer, H. Levy, R. Sstamm, and D. Tullsen."Converting thread level parallelism into instruction-level parallelism via simultaneous multithreading". ACM Transactions on Computer Systems, 15(2), pp. 323-354, August 1997.

[6] Lance, Hammond, Basem, Ku.umen "ANayfeh, Kunle Olukotun.ASingle-Chip multiprocessor. IEEE Computer", vol. 30, no. 9, pp. 79--85, September1997.

[7] J. Borkenhagen, R. Eickemeyer, and R. Kalla : "A Multithreaded PowerPC Processor for Commercial Servers, IBM Journal of Research and Development", November 2000, Vol. 44, No. 6, pp.1995.

AUTHORS

PRASANTH VARASALA Completed B.Tech in 2004, M.tech in VLSI system Design in 2010 and Pursuing P.hD in JNTUK in Low power VLSI. Areas of interests are low power designing, realtime applications, Embedded Systems, Digital Designing.He is an Assoc. Professor and HOD of ECE in Pragathi Engineering College, Surampalem.

K.RAJA SEKHAR Completed B.Tech in 2010 in Gudlavalleru Engineering College and presently doing his Master degree in Embedded Systems from JNTU Kakinada University in Pragathi Engineering College, Surampalem. Areas of interests are realtime applications, Embedded Systems, Digital Designing.