



# Performance Enhancement Technique in Multiprocessors

Kavitha V.  
Ph.D Scholar, Jain University

K.V. Ramakrishanan, PhD  
Professor Emeritus

## ABSTRACT

Design of a programmable Multi core processor to implement compute several complex multimedia applications is presented. The Processor is expected to complete the given task with minimum latency. The hardware must adhere to minimal area and power requirements. This paper gives design details for enhancement of performance parameters of multi core processors. It is necessary to optimize the processor performance at both architectural and execution levels. At architectural level use of locally synchronous clocking mechanism will eliminate the use of global clock tree with the help of asynchronous handshake protocol. At execution level completion time is reduced by 30% with the concept of reconfigurable instruction set processor and parallelism at data, memory, instruction and task level.

## General Terms

Multiprocessor, Instruction set, Globally asynchronous and locally synchronous (GALS), Multi Reconfigurable Instruction set processor system on chip (MRPSoC), Multigrain-MRPSoC, Reconfigurable Instruction set Processors (RISP), Multigrain parallelism.

## Keywords

GALS-Globally asynchronous and locally synchronous, GPP-General Purpose processors, ASIP- Application Specific integrated Protocol, ASIC- Application Specific integrated circuits, MPSoC- Multi Processor System on chip, RISP-Reduced instruction set processor, MRPSoC- Multi-Reconfigurable instruction set processor system on chip, RFU -Reconfigurable functional unit.

## 1. INTRODUCTION

Computer technology has made vast and incredible progress over the years since the first general purpose processor (GPP) was introduced. In the late 90's, the emergence of multiprocessors led to a growth rate of 60% due to high performance improvements. In the recent past ten years growth rate increased to 80-100% due increased performance achieved by Multiprocessors. The performance of such processors have shown fastest growth rate when compared to supercomputers, mainframes and minicomputers. To meet these growth rates parallel computing is inevitable. Performance is always the main objective in all recent applications like video streaming, voice recognition, internet of things, image and video processing and computer aided designs require higher and higher computing capability. In scientific computing higher computational performances is always welcome. As multiprocessors are used in a wide range of applications, there is a need to meet the design metrics of multiprocessors like power, flexibility, robustness, cost, area, performance, completion time of application and many more. Such multiprocessors are expected to show much higher

performance when more hardware resources are added to the machine i.e. they should be scalable. The design of multiprocessors architecture must address the loss in processor efficiency due to the following fundamental issues like (i) longer completion time of tasks,(ii) Long memory latencies, (iii) synchronization, (iv) Problems in communication between processors and lastly (v) Underutilization of individual processor efficiency.

Designer must address these problems and include sufficient flexibility and scope for parallelism in the given task. Long latency in execution can be overcome with instruction pipelining and queuing techniques but resource sharing and scheduling becomes a major setback in such applications. This will limit the speed of computation process. Issues of clocking and synchronization among multiple cores can be overcome by innovative and effective design of architectural components. Idle time of processors can be reduced by comprehensive scheduling and resource management techniques.

## 2. MOTIVATION

Performance enhancement has become a major and important need to satisfy the computational requirements of the current multimedia applications. Multiprocessor architecture innovations are needed to sustain the high performance growth rate of computers. There are various obstacles that make design of multiprocessors and implementation a challenging task, like limited parallelism available in the program, long memory latencies, and high cost of communications and issues of synchronization. A comprehensive study of various techniques for effective design of overall multiprocessor on a single chip incorporating all the architectural and software perspectives have not yet been carried out. In this work a comprehensive and detailed study and analysis of various methods to experimentally verify the efficiency of a high performance multiprocessor on chip is carried out. Hence the goals and objective of this work are,

- i. To study the various performance enhancement issues of multiprocessors
- ii. Analyse the various techniques to optimize the performance with architectural and software perspective.
- iii. Implement a design which improves the overall performance of the multiprocessor with respect to completion time, memory latency, parallelism, synchronization and clocking.
- iv. Present a comparative analysis of complete design with the existing techniques.

### 3. BRIEF SURVEY OF LITERATURE

A survey of three papers giving details of techniques used to meet some of the objectives of this work is given below.

[1] This paper describes platforms which consist of Multi Reconfigurable Instruction Set Processor System on Chip (MRPSOC). Reconfigurable Instruction set processor (RISP) consists of a microprocessor core that can be extended with reconfigurable logic. Concepts of RISP and use of Reconfigurable functional units for parallel and advance computation is proposed in this paper. MRPSOC can run applications in parallel and accelerate the performance due to its reconfigurable function unit (RFU) retaining s programmability.

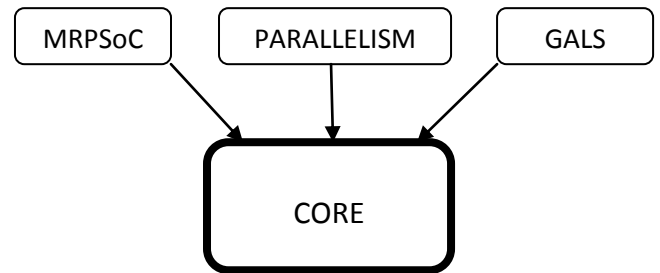
[2] This paper describes the potential parallelism in multimedia application and Multigrain parallelism in CMP. To accelerate the multimedia computing, parallelism is achieved at all the levels of executions. A method based on multigrain parallelism in CMP is proposed which utilizes the parallelism from Data-Level, Instruction-Level, Thread-Level and Memory-Level. This method lets the programmer to successfully layout the multimedia related program.

To address the issues of parallelism, a parallel programming methodology is proposed in this paper, which utilizes the multigrain parallelism in CMP, for multimedia application. However, hardware must fully exploit the parallelism, at the same time programmer should explicitly parallel programme the tasks. In this method, programmer should take the core Architecture, the number of cores on chip, the interconnection and the communication protocol into consideration to achieve the expected results.

[3] This paper describes a Globally Asynchronous and Locally Synchronous (GALS) is an architecture that retains the benefits of synchronous systems, yet avoids the problems due to global clock tree. The GALS architecture is composed of large synchronous blocks (SBs) which are synchronized by a local clock and communication between the multiple processors is done asynchronously with each other. As the global clock tree is eliminated in this, the power to drive this clock tree is saved. Clock skew requirements in the SB clock nets relax thus easing the design issues. However, the GALS architecture comes with two overheads of its own: one is due to the asynchronous protocol signals, the other related to local clock generation. For the GALS architecture to be profitable, it is vital to keep these overheads low compared to the gain achieved by eliminating the global clock tree.

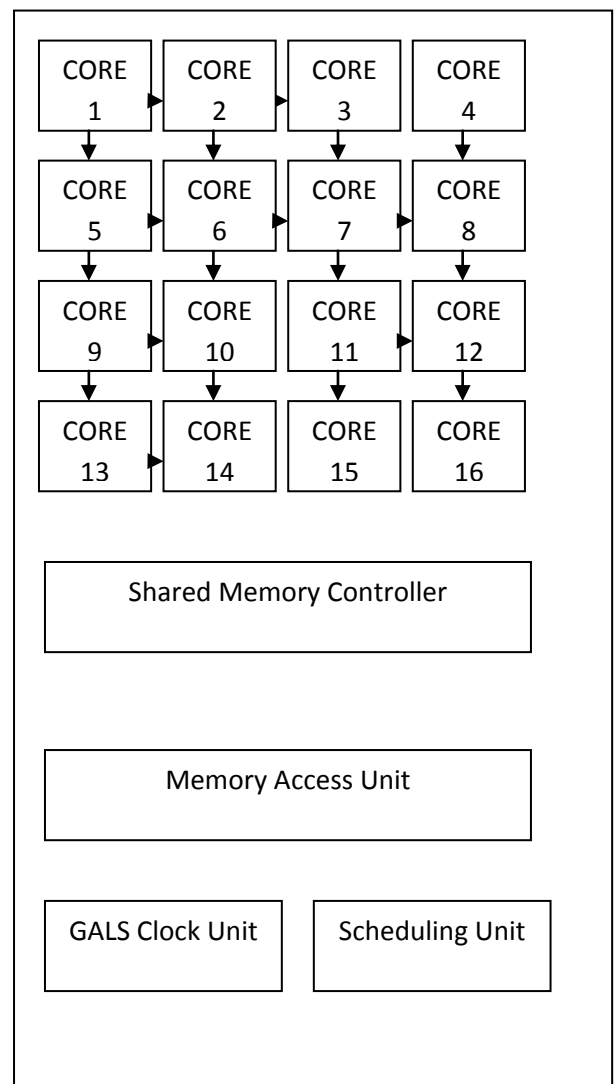
### 4. METHODOLOGY

Experimental wok of this project is carried out in 2 phases. In first phase a uniprocessor is designed with features of parallelism, reconfigurable logic and GALS clocking techniques. Later a matrix of 4X4 such processors are built to form a multi core processor on chip as shown in Fig.2.



**Fig.1. Uniprocessor with features of MRPSoC, Parallelism and GALS clocking**

Design issues of multi core processors are addressed at both architectural and execution level. At architectural level a reconfigurable functional unit is placed in every core which accelerates the computational speeds allowing parallel computations of critical portions of the program.



**Fig.2 Multiprocessors on chip with controlling circuitry**

Further due to the elimination of clock tree GALS has enhanced the power and area savings. This technique makes use of handshake signals to synchronize between the cores. Architectural level implementations are done using Modelsim



simulation and coding is done in Veilog Hardware Description Language (VHDL).

At the system level or execution level multi grain parallelism technique is used. Here parallelism is achieved at data, memory, instruction and task levels to optimize the performance of the multi processors. The code is written in System-C and compilation is done using Perl compiler.

### 5. EXPERIMENTATION AND RESULTS

Experimentation is done in two phases. In phase I, a single processor is built with enhanced performance techniques such as, reconfigurable instruction set, multigrain parallelism and GALS clocking. In phase II, a matrix of 4X4 processors are built on a single platform and an application is run to test the multi tasking, parallelism and synchronization between the processors on chip. Codes are written in Verilog HDL and System-C, and performance is verified using Modelsim and Pearl compilers. Simulation results of the proposed system in compression with normal multiprocessor on chip are shown in Fig. 3 and Fig.4. Time taken to complete the given task on the proposed MPSoC is 124 ns versus 173ns on a normal MPSoC

```

REG :
=====
R 0(00000000) R 1(00000003) R 2(00000001) R 3
(00000002)
R 4(00000001) R 5(00000002) R 6(fffffffc) R 7
(ecf0fdef)
R 8(fffffffd) R 9(00000009) R10(00000002) R11
(0000ff31)
R12(0000ff12) R13(00000013) R14(00000014) R15
(00000015)
R16(00000016) R17(00fe0117) R18(00fe0118) R19
(00fe0119)
R20(00fe0220) R21(00fe0321) R22(00fe0322) R23
(00ff0423)
R24(00ff0524) R25(00ff0625) R26(00ff0726) R27
(00ff0727)
R28(00ff0728) R29(00000029) R30(00000030) R31
(00000031)
=====

IFU : mem=0xffffffff
IFU : pc= 19 at CSIM 173 ns

-----
ID: - SHUTDOWN - at CSIM 175 ns
ID: - PLEASE WAIT ..... -
  
```

Fig.3 Completion time of Normal MPSoC

```

REG :
=====
R 0(00000000) R 1(00000006) R 2(00000001) R 3
(00000002)
R 4(00000001) R 5(00000002) R 6(fffffffc) R 7
(ecf0fdef)
R 8(00000008) R 9(00000009) R10(00000010) R11
(0000ff31)
R12(0000ff12) R13(00000013) R14(00000014) R15
(00000015)
R16(00000016) R17(00fe0117) R18(00fe0118) R19
(00fe0119)
R20(00fe0220) R21(00fe0321) R22(00fe0322) R23
(00ff0423)
R24(00ff0524) R25(00ff0625) R26(00ff0726) R27
(00ff0727)
R28(00ff0728) R29(00000029) R30(00000030) R31
(00000031)
=====

IFU : mem=0xffffffff
IFU : pc= 12 at CSIM 124 ns

-----
ID: - SHUTDOWN - at CSIM 126 ns
ID: - PLEASE WAIT ..... -
  
```

Fig.4 Completion time of Proposed Multiprocessor

### 6. MAJOR CONTRIBUTIONS

To eliminate the design issues of multiprocessors following techniques were implemented.

In all multimedia computations the major problem of longer completion time exists. This problem is overcome by introducing multi reconfigurable instruction set processor (MRPSoC) architecture. MRPSoC can run applications in parallel with the help of reconfigurable functional unit (RFU). Here the critical portions of the code are identified and run in parallel on RFU. The non-critical instructions are run on the remaining available processors. With effective scheduling this technique will complete the task with 30% less time.

In addition to this the delay in memory access and latency caused due to data fetching is overcome by adding features of multigrain parallelism. Here the parallel processing is not only meant at instruction level but also at data, memory and task level.

Issues of synchronization between individual cores inside the multiprocessor are eliminated by using globally asynchronous and locally synchronous clocking mechanism. Here a considerable amount of power saving is achieved by introducing asynchronous handshake based protocols among the cores.

### 7. CONCLUSIONS

This paper presents a novel solution to the current multi core processors design issues like longer completion time, long memory latency, issues of synchronization and clocking. It is concluded that the performance enhancement techniques have been implemented for multiprocessors on a chip with 4X4 processors. Proposed system provides a comprehensive and best solution to optimize the performance of multiprocessors on chip with both architectural and software perspective. Comparative analysis reveals that proposed design provides a 30% improvement in completion time of a task when compared to normal multiprocessor on chip designs.

### 8. REFERENCES

- [1] Rahim Soleymanpour, Mazandaran, Behshahr, Iran SiamakMohammadi “A Platform for Multi Reconfigurable Instruction Set Processor System on Chip (MRPSoC)” IEEE Journal, Computer Architecture and Digital Systems (CADS), 2013 17th CSI International Symposium on Digital Object Identifier: 10.1109/CADS.2013.6714244 Publication Year: 2013 , Page(s): 99 – 104.
- [2] Xiaoping Huang, Xiaoya Fan, Shengbing Zhang, Liwen Shi “Investigation on Multi-Grain Parallelism in Chip Multiprocessor for Multimedia Application” proceedings of IEEE Journal, pp.1-4,2009.computer school,Northwestern Polytechnical University,China.
- [3] Thomas Meincke, Ahmed Hemani, Shashi Kumar, PeeterEllervee, Johnny Oberg ,Thomas Olsson, Peter Nilsson, Dan Lindqvist, HannuTenhunen “Globally Asynchronous Locally Synchronous Architecture for Large High-Performance Asics”. Circuits and Systems, 1999.ISCAS '99. Proceedings of the 1999 IEEE International Symposium on Volume: 2 Digital Object Identifier: 10.1109/ISCAS.1999.780794 Publication Year: 1999, Page(s): 512 - 515 vol.2



- [4] Ichiro Kuroda, Takao Nishitani, "Multimedia Processors", Proceedings of the IEEE, Vol.86, No.6, June 1998.
- [5] F. Barat and R.Lauwereins, "Reconfigurable Instruction Set Processors: A Survey", Rapid system prototyping, 11th International workshop, 2000, pp.168- 173.
- [6] L. Pozzi, K. Atasu, and P. Ienne, "Exact and approximate algorithms for the extension of embedded processor instruction sets," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , vol.25, no. 7, pp. 1209-1229, 2006.
- [7] Xiaoping huang, xiaoya Fan, Shengbing Zhang, "The integration of multimedia process unit into an embedded processor", Proceeding of the 2007 IEEE international conference on integration technology, pp:492-495, March 2007, China.
- [8] Jack L.Lo and Susan J.Eggers; "Improving Balanced Scheduling with Compiler Optimization that Increase Instruction Level Parallelism", Department Of Computer Science and Engineering, University Of Washington, 1995.
- [9] A. Lodi, M. Toma, F. Campi, A. Cappelli, R. Canegallo, and R.Guerrieri, "A VLIW processor with reconfigurable instruction set for embedded applications," Solid-State Circuits, IEEE Journal of, vol. 38, no. 11, pp. 1876-1886, 2003.
- [10] Kurt Keutzer, Sharad Malik, A.Richard Newton," From ASIC to ASIP: The next design Discontinuity", 2002
- [11] Yousef Qasim, Pradyumna Janga, Sharath kumar, Hani Alesaimi, "Application Specific Processors", Final\_ECE570\_ASP\_2012, Project Report.
- [12] Nohl .A, Schirrmeister.F and Jaussig.D, "Application Specific processor design: Architecture, design methods and tools", computer\_aided design (ICCAD), 2010 IEEE/ACM International conference on Nov.2010.
- [13] Wayne Wolf, Ahmed Amine jerraya and Grand Martin, "Multiprocessor system on chip (MPSOC) technology", IEEE transactions on computer aided design of integrated circuits and systems, vol.27, no.10, October 2008.
- [14] A novel MRPSOC processor for dispatch time curtailment, Parvathy Asokan, final report, 2014.
- [15] Grotker, Lio, Martin,"System design with system c", Kluwer academic publications, 2002.