

# A 16-CORE PROCESSOR WITH SHARED MEMORY AND MESSAGE-PASSING COMMUNICATION

Chandrashekhar Jatla<sup>1</sup>, Basavaling.Hiremath<sup>2</sup>, Waheeda<sup>3</sup>, Preeti.Patil<sup>4</sup>

<sup>1</sup>M.Tech (PG student), Electronics and Communication, L.A.E.C, Karnataka, India

<sup>2</sup>Asst.prof, Electronics and Communication, L.A.E.C, Karnataka, India

<sup>3</sup>M.Tech (PG student), Electronics and Communication, L.A.E.C, Karnataka, India

<sup>4</sup>M.Tech (PG student), Electronics and Communication, L.A.E.C, Karnataka, India

## Abstract

*In this project shared memory architecture is designed with 16-core processors, which are connected in star topology and share common memory for program and data. Processor is expected to run at 160 MHz, with effective instruction cycle speed at 1MHz (160 x 1). Shared memory is called 16-Core memory as the memory has hex ports. Any processor can write/read from the memory at the same time. If collisions occur, they are handled by priority method. The main aim of the project is to design “A 16-Core Processor with Shared-Memory and Message-Passing Communications”. The processor has 16 processor cores and 2 memory cores. Message-passing communications are supported by the 3x6 2D Mesh NOC, and shared-memory communications are supported by shared memory units in the memory cores.*

**Keywords** – Chip Microprocessor, Message Passing, Multi Core, Network-On-Chip, NoC, Shared Memory, Inter-core communication.

\*\*\*

## 1. INTRODUCTION

POWER budgets of embedded processors are bearing higher pressure than before, driven by the massive employment of mobile computing devices, the advancement of applications in communication and multimedia systems even exacerbates the situation. Fortunately Chip Microprocessor emerge as a promising solution and many efforts are taken to increase the parallelism and optimize memory hierarchy still enhancing Performance. However even Managing to rebalance performance and power, multicore architecture still introduces new challenges on inter-core communications, which soon becomes the key for further performance improvement.

Especially, the efficiency of inter-core communications has direct impact on the performance and power metrics of embedded processors. When a certain applications is mapped on a multicore system, if is usually pipelined and the throughput depends on both computing capability and communication efficiency between cores. Although there are various performance – enhanced technologies, such as super-scalar and Very Long Instruction Word (VLIW) etc. mature solution for inter-core communications are still in absence temporarily. Hence, the next stage research should focus on improving the efficiency of inter core communications.

In the history of microprocessor, shared-memory communication is the most often used mechanism due to its simple programming model. However, it fails to provide

sufficient scalability to cater to the increasing core numbers. Therefore, multicore processor designers turns to the message passing communication mechanism, which shows more scalability and potential to be employed in the next-generation embedded multicore processor.

In this paper, we attempt to summarize the key features of the shared-memory and message passing communication. We show that different inter-core communication methods match with different scenarios, which implies that we could obtain a higher performance and power efficiency by integrating both inter-core communication mechanisms.

We propose a 16-core processor adopting hybrid inter-core communication schemes with both shared-memory and message passing inter core communication. A 2D Mesh Network-on-Chip (NoC) is adopted to support message-communication Meanwhile, a cluster-based memory hierarchy including shared memory enables shared-memory communications.

We also propose a hardware-aided mailbox inter-core synchronization method to support inter-core communication, and new memory hierarchy to achieve higher energy efficiency. A prototype 16-core processor chip has been fabricated in TSMC 65nm Low Power(LP) CMOS process and shows full functions.

This paper is organized as below. Section II describes the key features of the 16-Core processor, III overview on related work. Section III details its design and

implementation. Section IV presents the measured result with the fabricated chip. Section V concludes the paper.

## 2. MOTIVATION AND KEY FEATURES

The primary motivation of our work is to improve the performance and power efficiency of embedded multicore processor while still maintaining flexibility, in other words, to reduce the efficiency gap between multicore processor and ASICs as shown in fig1. Several key features are implemented which are detailed in the following subsections.

## 3. RELATED WORK

**“Hsiao-Ping Juan, Nancy D. Holmes, Smita Bakshi, Daniel D. Gajski, October 5, 1992”** This paper describes about the Top-Down Modeling of RISC Processors in VHDL modeling technique which consists of two main modeling levels: specification level and functional level. This Paper was used as reference in the designing RISC Processor in HDL language (Verilog). Some of the Functional module like RISC Processor Architecture was utilized in designing the Quad core was referred from this paper.

**“Wael M ElMedany, Khalid A AlKooheji, 2004”** This Paper describes about the design and Implementation of a 32bit RISC Processor on Xilinx FPGA. This was taken as reference in Hardware implementation of a single RISC Processor which was further implemented to Multi core and Quad core was designed and verified on an hardware.

**“Peter Manoilov, Plamena Krivoshieva”, 2008** this paper describes about the effectiveness of shared memory blocks with various organizations and architecture in multicore systems on FPGA is investigated. A method of quad-port memory implementation on FPGA, using the existing dual-port RAM FPGA cells and adding a doubled clock is shown.

**“Antonino Tumeo, Matteo Monchiero, Gianluca Palermo, Fabrizio Ferrandi, Donatella Sciuto, 2010”** This paper describes about a framework of designing a shared memory multiprocessor on a programmable platform (FPGA platform)..

**“Niklaus Wirth, 2011”** This Paper describes the design of RISC Architecture and its Implementation with an FPGA, This paper was referred in designing the architecture in pipeline format, which reduces the number of cycle for execution of multi-instruction.

**“Zhiyi Yu 2009”** This Paper describes about A 167-processor computational platform consists of an array of simple programmable processors capable of per-processor dynamic supply voltage and clock frequency scaling, three algorithm-specific processors, and three 16 KB shared memories; and is implemented in 65 nm CMOS. All processors and shared memories are clocked by local fully

independent, dynamically halttable, digitally-programmable oscillators and are interconnected by a configurable circuit-switched network which supports long-distance ...

## 4. DESIGN AND IMPLEMENTATION

The disadvantages of conventional approach are that it consumes more power, the design utilizes more number of gates, uses four clock cycle and requires one clock per request for each processor, it requires memory arbiter to communicate with all the processor, separate logic circuit as to be designed and for each processor having dedicated memory.

To overcome these disadvantages shared memory architecture for quad core processor is designed with less number of gates. The system consumes less power and latency is also less. Architecture of shared memory quad core processor

### 4.1 Shared Memory

In this approach the shared memory, as the memory has 16 ports. Any processor can write/read from the memory at the same time (assuming that there are no collisions). However, write address collisions are handled by priority method.

#### The shared memory consists:

- DATA input
- DATA output
- ADDRESS
- READ/write operations
- Single Clock

#### Properties of Memory:

- Random access:
  - Accessing any memory location takes the same amount of time.
- Volatility:
  - Volatile memory.
  - Needs power to retain the contents.
- Non-volatile memory:
  - Retains contents even in the absence of power.
- Basic types of memory:
  - Read-only memory, **ROM**.
  - Read/write memory, **RAM**.

#### ROM, Read-only Memory:

- ROM characteristics are:
  - Cannot be written into this type of memory.
  - Non-volatile memory
  - Most are factory programmed (i.e., written.)
- Programmable ROMs (PROMs)
  - Can be written once by user
  - A fuse is associated with each bit cell

- Special equipment is needed to write (to blow the fuse)
- PROMS are useful:
  - During prototype development.
  - If the required quantity is small.

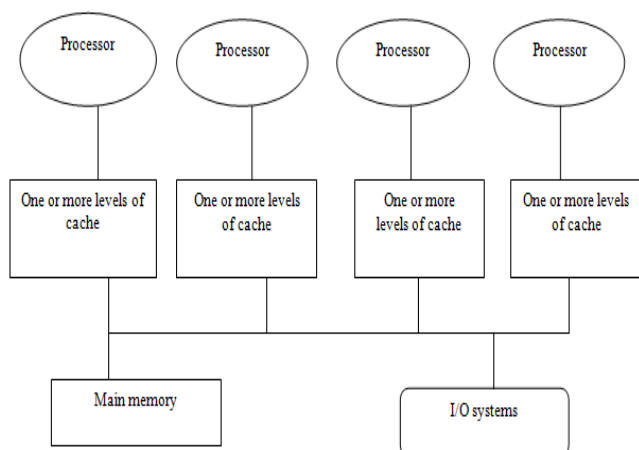
#### EPROM, Erasable PROMs:

- EROM characteristics are:
  - Can be written several times.
  - Offers further flexibility during system prototyping.
  - Can be erased by exposing to ultraviolet light.
  - Cannot erase contents of selected locations.
  - All content is lost on re-write.
- Electrically erasable PROMs, or EEPROMs.
  - Contents are electrically erased.
  - No need to erase all contents
  - Typically a subset of the locations is erased as a group.
  - Most EEPROMs do not provide the capability to individually erase contents of a single location.

#### RAM, Read/Write Memory:

- Commonly referred to as random access memory, **RAM**:
  - Volatile memories.
- Two basic RAM types:
  1. Static RAM, **SRAM**:
    - Retains data with no further maintenance.
    - Typically used for CPU registers and cache memory.
  2. Dynamic RAM, **DRAM**:
    - A tiny capacitor is used to store one bit.
    - Due to leakage of charge, DRAMs must be refreshed to retain contents.
    - Read operation is destructive in DRAMs.

### 3.2 Design Of Shared Memory



**Fig 3.1:** Basic structure of a centralized shared-memory multiprocessor

Shared-memory machines usually support the caching of both shared and private data. Private data are used by a single processor, while shared data are used by multiple processors which provide communication among the processors through reads and writes of the shared data. When a private item is cached, its location is migrated to the cache, reducing the average access time as well as the memory bandwidth required. Since no other processor uses the data, the program behavior is identical to that in a uni-processor.

When shared data are cached; the shared value may be replicated in multiple caches. In addition to the reduction in access latency and required memory bandwidth, this replication also provides a reduction in contention that may exist for shared data items that are being read by multiple processors simultaneously.

Basic structure of a centralized shared-memory multiprocessor is shown in figure 3.1. Multiple processor-cache subsystems share the same physical memory, typically connected by one or more buses or a switch. The key architectural property is the uniform access time to all of memory from all the processors.

### 3.3 Design of Processor

Unpipelined implementation is not the most economical or the highest-performance implementation. Instead, it is designed to lead naturally to a pipelined implementation. The number of dependent steps varies with the machine architecture. A non-pipelined processor executes only a single instruction at a time.

#### Specification of Processor

- Architecture contains 23 (alu) instructions (6 Arithmetic+ 8 Logical+4 Datapath+5 Branching Instruction).
- Pipelined architecture
- Four stage instruction execution (IF, ID, IE, ST).
- Harvard memory architecture(one for code memory and main memory).
- 8 bit data and 8bit address bus.
- 8 bit memory mapped I/O register.
- 1 Special purpose status registers.
- 13 General purpose CPU registers. Uniform instruction width for all the instruction.

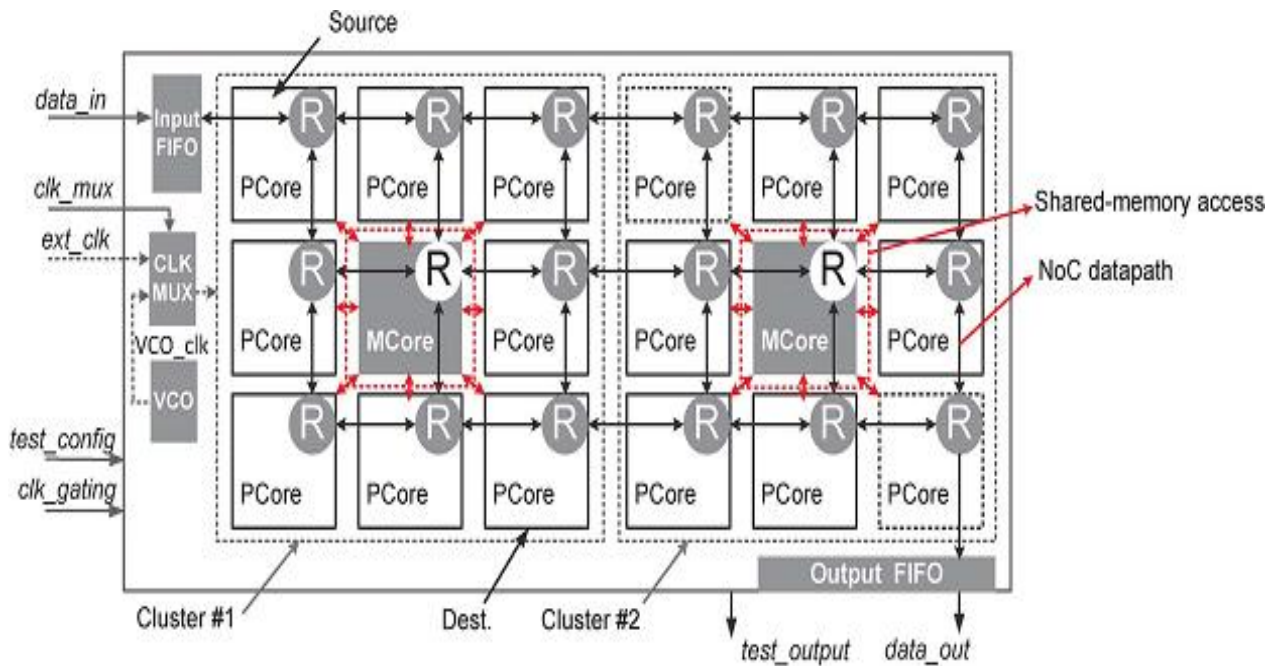


Fig 3.2 Architecture overview of the proposed 16-core processor

#### 4. PROPOSED ARCHITECTURE:

The processor has 16 processor cores and 2 memory cores. Message-passing communications are supported by the 3x6 2D Mesh NoC, and shared-memory communications are supported by shared memory units in the memory cores.

The proposed cluster-based memory hierarchy makes the processor well-suited for most embedded applications. The processor chip has a total 256 KB on-chip memory, while each processor core has an 8 KB instruction memory and a 4 KB private data memory, and each memory core has a 32 KB shared memory.

#### 5. SIMULATION RESULTS

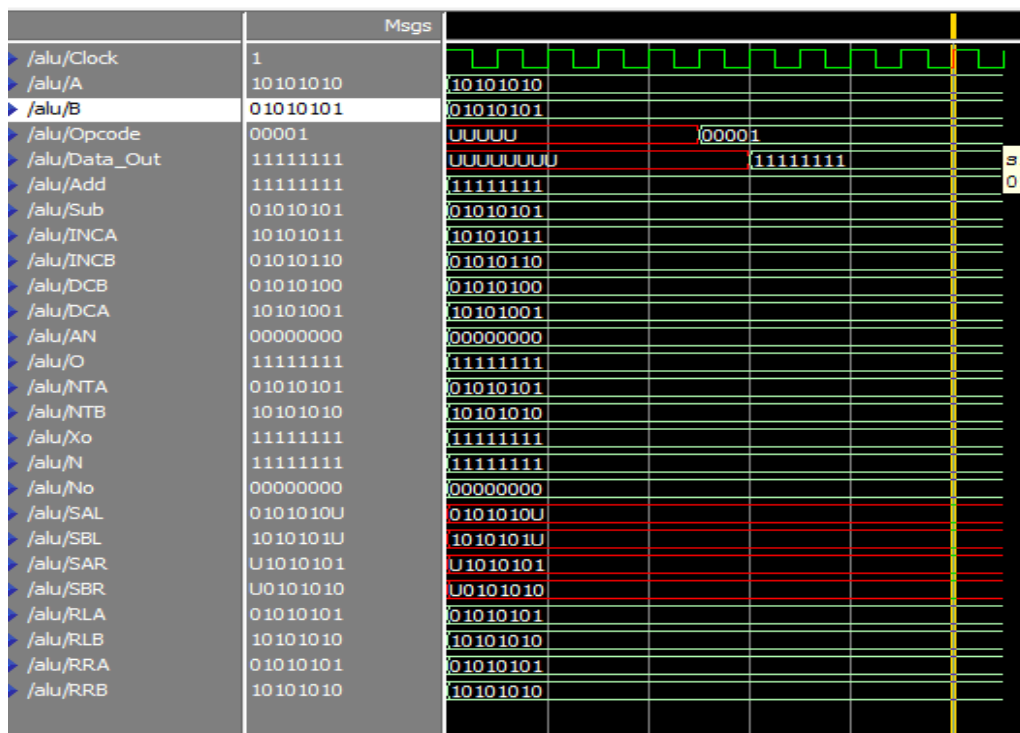


Fig 3: 8-bit ALU Implementation

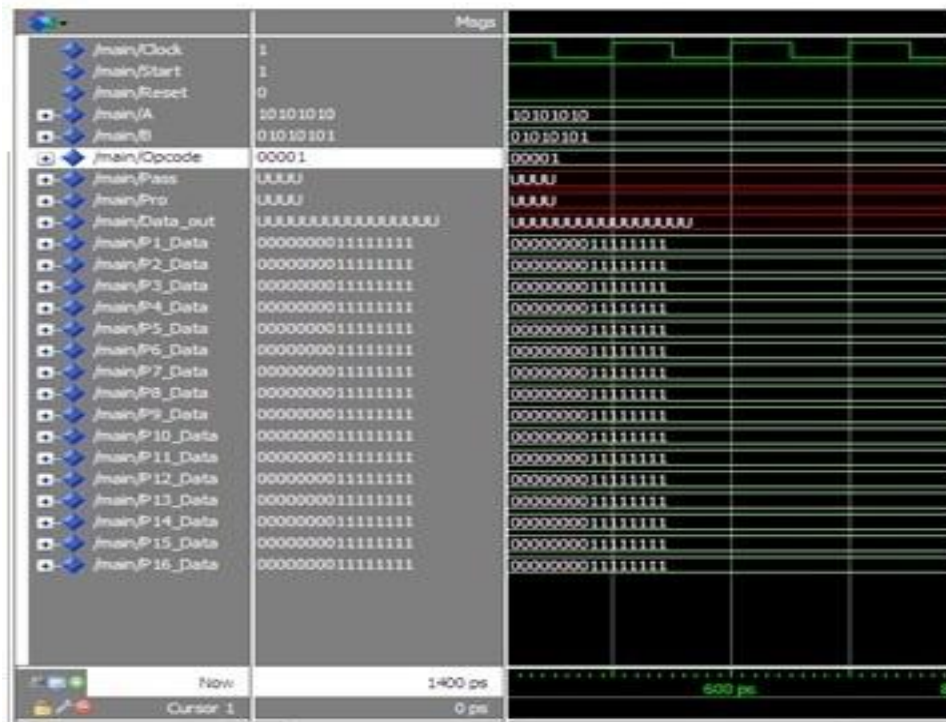


Fig 4: Implementation of 16 - core processor with shared memory and message passing communication.

## 6. CONCLUSION

A 16-core processor for embedded applications with hybrid inter-core communications is proposed in this paper. 1092 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, VOL. 61, NO. 4, APRIL 2014. The processor has 16 processor cores and 2 memory cores. Message-passing communications are supported by the 3 x 6 2D Mesh NoC, and shared-memory communications are supported by shared memory units in the memory cores. The proposed cluster-based memory hierarchy makes the processor well-suited for most embedded applications. The processor chip has a total 256 KB on-chip memory, while each processor core has an 8 KB instruction memory and a 4 KB private data memory, and each memory core has a 32 KB shared memory. The processor is fabricated in TSMC 65 nm LP CMOS with the chip area of  $9.1\text{mm}^2$ , while each core occupies  $0.43\text{mm}^2$ . Typically, the frequency of each processor core is 750 MHz at 1.2 V while dissipating 34 mW, with an energy efficiency of 45 pJ/Op for 32-bit operation and 22 pJ/Op for 16-bit operation.

## REFERENCES

- [1]. G. Blake, R. G. Dreslinski, and T. Mudge, "A survey of multicore processors: A review of their common attributes," *IEEE Signal Process. Mag.*, pp. 26–37, Nov. 2009.
- [2]. R. Kumar, V. Zyuban, and D. Tullsen, "Interconnections in multi-core architecture: Understanding mechanisms, overheads and scaling," in *Proc. 32nd Int. Symp. Computer Architecture (ISCA'05)*, 2005, pp. 408–419.
- [3]. H.-Y. Kim, Y.-J. Kim, J.-H. Oh, and L.-S. Kim, "A reconfigurable SIMT processor for mobile ray tracing with contention reduction in shared memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, no. 60, pt. 4, pp. 938–950, Apr. 2013.
- [4]. L. Hammond, B.-A. Hubbert, M. Siu, M.-K. Prabhu, M. Chen, and K. Olukolun, "The stanford Hydra CMP," *IEEE Micro*, vol. 20, no. 2, pp. 71–84, 2000.
- [5]. A. S. Leon, B. Langley, and L. S. Jinuk, "The UltraSPARC T1 processor: CMT reliability," in *Proc. Custom Integrated Circuits Conf. (CICC'06) Dig. Tech. Papers*, 2006, pp. 555–562.
- [6]. M.-B. Taylor, J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, P. Johnson, J.-W. Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Stumpfen, M. Frank, S. Amarasinghe, and A. Agarwal, "The Raw microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, Mar/Apr. 2002.
- [7]. Tiler Corp., Tilepro64 Processor Tiler Product Brief, 2008 [Online]. Available: [http://www.tiler.com/pdf/Product-Brief\\_TILEPro64\\_Web\\_v2.pdf](http://www.tiler.com/pdf/Product-Brief_TILEPro64_Web_v2.pdf)
- [8]. S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-tile sub-100-WteraFLOPS processor in 65-nm CMOS," *IEEE J. Solid-State Circuit*, vol. 43, no. 1, pp. 29–41, Jan 2008.



- [9]. Z. Yu, M. J. Meeuwsen, R. W. Apperson, O. Sattari, M. Lai, J. W. Webb, E. W. Work, D. Truong, T. Mohsenin, and B. M. Baas, "AsAP: An asynchronous array of simple processors," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 695–705, Mar. 2008.
- [10]. B. Rogers, A. Krishna, G. Bell, and K. Vu, "Scaling the bandwidth wall: Challenges and avenues for CMP scaling," in *Proc. ACM Int. Symp. Computer Architecture (ISCA'09)*, 2009, pp. 371–382.

## BIOGRAPHIES



**Chandrashekhar Jatla**, B.E: BKIT Bhaki, Bidar. M.Tech:final year(VLSI and ES) L.A.E.C Bidar. Area of interest VLSI and ES design implementation, image processing, filters.



**Basavaling Hiremath**, B.E: AIET Gulbarga-2009 M.Tech;VTU extension centre UTL technologies LTD Bangalore-2012. Research area:analog design,low power VLSI microchip antenna,recent working as Asst.prof in ECE dept,L.A.E.C Bidar



**Waheeda**, B.E: GNDEC, Bidar – 2013. M.tech final year (VLSI and Embedded-System) L.A.E.C.Bidar. Area of interest soc,low power. VLSI design and related to VLSI field.



**Prethi Patil**, B.E:K.B.N.C Gulbarga - 2012 M.Tech: Final year (VLSI and Embedded system) L.A.E.C, Bidar, 585 403, Visveswaraya technological university Belgaum 590018. Area of interest Advances in VLSI architecture, Recent trends in VLSI design, Testing and verification