MORA- A Coarse Grained Reconfigurable Architecture

Sayali Dhamale
UG Student
Department of Electronics & Telecommunications Engineering
Sinhgad Institute of Technology & Science

Vaishnavi Kokate
UG Student
Department of Electronics & Telecommunications Engineering

Pooja Naniwadekar
UG Student
Department of Electronics & Telecommunications Engineering
Sinhgad Institute of Technology & Science

R. R. Kubde
Assistant Professor
Department of Electronics & Telecommunications Engineering
Sinhgad Institute of Technology & Science

Abstract

This paper gives an overview of MORA architecture, a coarse grained reconfigurable processor for leading edge multimedia applications. MORA architecture is a 2-D array mesh based of such coarse grained reconfigurable processors. MORA asserts several advantages throughout the design cycle over traditional FPGA systems. The paper contains basic principles of coarse grained reconfigurable architectures (CGRAs) and the vast range of design options available to a CGRA designer, covering a large number of existing CGRA designs.

Keywords: MORA; CGRA; FPGA; ASIC; Reconfigurable Cells; Processing Element

I. INTRODUCTION

Leading- edge multimedia applications, including image processing, digital signal processing, video stream operations and others, demand for high-performance computations with the potential of matching the rapid evolution of the algorithms. Due to the rapid advances in algorithms and applications in DSP, there has been an increasing demand for the computing platforms to be easily compliant, low cost while constantly delivering maximum performance at all times. The simultaneous demand for high computational speed and flexibility makes reconfigurable architectures to be attractive solutions. In reconfigurable computing, a crucial role is covered by fine-grained Field Programmable Gate Arrays (FPGAs). FPGAs consist of a matrix of reconfigurable logic cells and offer a performance achievable through multiple levels of parallelism. Yet, FPGAs exhibit some limitations that have to be overcome at high- performance architectures. Course Grained Reconfigurable Architecture (CGRA) overcomes these limitations with their high level of flexibility and efficient routing structures [1].

CGRAs have numerous advantages over traditional FPGA approaches. First, ease of application development, several of the described platforms of CGRA use high level language (HLLs) such as C or C++, rather than constructing a design in a hardware description language such as VHDL or Verilog. Second, CGRAs have high-performance and lower power consumption per operation when compared to FPGAs and DSPs. This is accomplished through more efficient use of silicon per operation as compared to FPGAs.

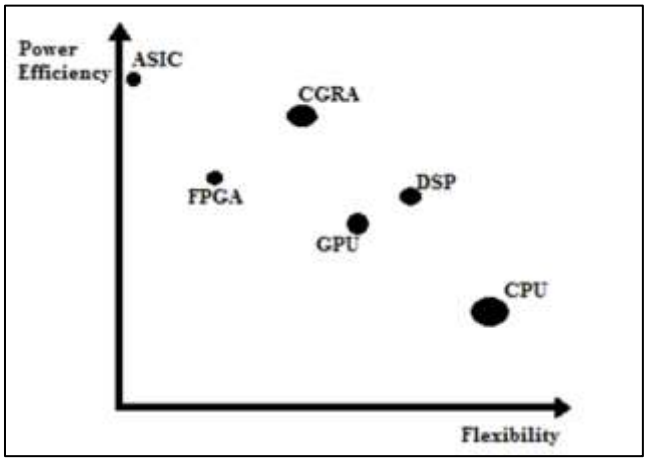

Fig. 1: Power Efficiency vs. Flexibility of ASIC, FPGA, GPU, DSP and CPU

II. MORA

MORA is a 2-D mesh-based CGRA system, consisting of an array of reconfigurable cells arranged in four 4X4 quadrants, shown in Fig.2. A distinctive characteristic of MORA is the massive reduction in the number of actual transistors per core. In fact with each MORA core requires just over 60,000 transistors, the entire array of 64 cells requires only a fraction of the resources of other similar architectures. MORA provides for most of the basic arithmetic and logic functions required by the target domain of media processing, while providing reasonably high throughput with minimum resource utilization [1].

Another trait that sets MORA apart from existing CGRA as well as FPGA is the simple application mapping. Unlike other architectures, MORA acts more as a general platform and does not optimize for particular algorithm for maximum throughput, implementation of an application.

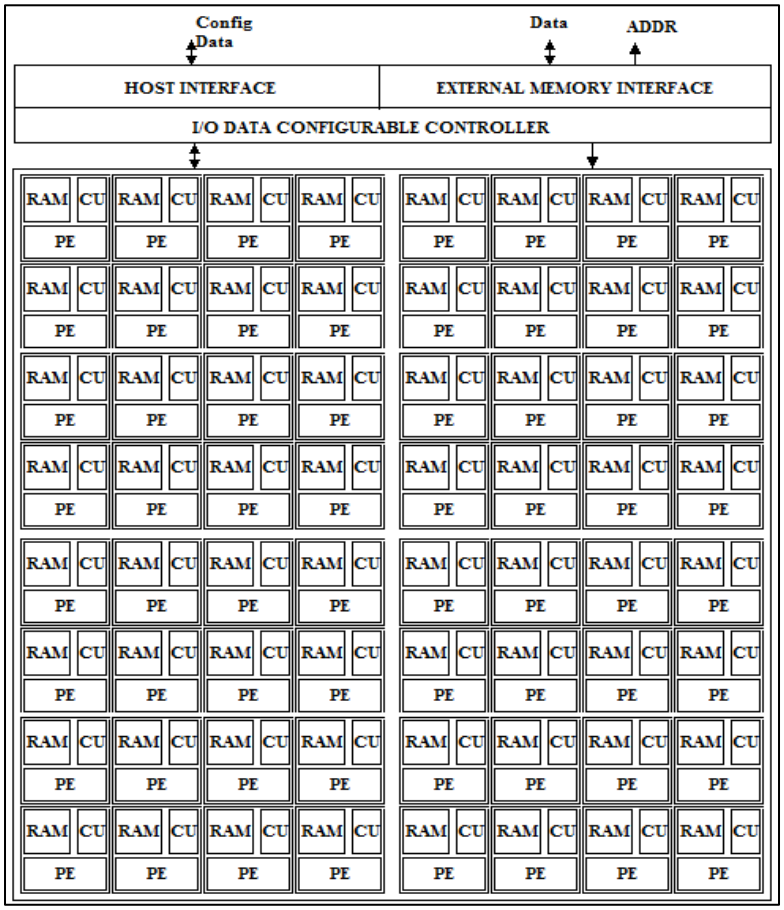

Fig. 2: 2-D array of MORA Processor

III. PROCESSING ELEMENT

The main computational unit of the RC is the processing element PE. It is consist of full range of signed and unsigned integer arithmetic, logical, shifting and comparison operations.

Logic section and an arithmetic section are two sections of PE. Operands are efficiently routed by these two sections. The following sections detail the structure of logic and arithmetic unit of PE.

A. Logic Unit

AND/NAND, OR/NOR, XOR/XNOR, shifting and comparison operations on 8-bit operands are performed by logical unit of PE. Logic unit consists of comparator, logarithmic shifter and an array of AND, OR, XOR gates.
Multiplexer is used to drive the operands to perform the multiple operations at a time such as shifting, comparison etc. It reduces load on the operands by reducing the total number of gates

Shift operation uses an 8-bit logarithmic shifter. One type of shifting operation at a time by the logarithmic shifter of PE i.e. Round Shift or Shift Out is done by the shifter. Comparison is another useful operation performed by PE of MORA architecture. This operation adds the extra applications of MORA like media processing, encryption, pattern matching, network intrusion detection, etc. All these applications require heavy, dedicated comparison operations on incoming operands. A specially designed comparison block is used by the PE of MORA. Area and power consumption depends on the different structure of the comparison operator.

B. Arithmetic unit

A number of operations that require to be supported in the media processing involves repetition. For our category of reconfigurable structures, the ideal size of operand is 8 bit. The problem with signed arithmetic is that data path is limited to 7 bits. There is a 1 bit loss due to a sign bit. Due to this loss of one bit the range of signed arithmetic reduces 0-255 to 0-127. In order to overcome this problem, 8-bit signed and unsigned arithmetic operations are proposed. It consists of two 8*4 hybrid multipliers, a compressor stage and 16- bit carry link address. The main requirement of media processing is arithmetic and logical instruction carried out in repetitive manner. For this purpose synchronisation between registers are required. Output and input of the register of different RC used in the structure should be synchronised with each other. Control unit is used for the synchronisation of the both internal and external structure of the RC. Basically the control unit provide the handshaking within the architecture [1].

IV. CONTROL UNIT

In reconfigurable cell, the main decision making block is control unit. Control unit ensures synchronization within each element of the RC. It consists of an Instruction Memory, Instruction Machine, Instruction Decoder, and Address generator.

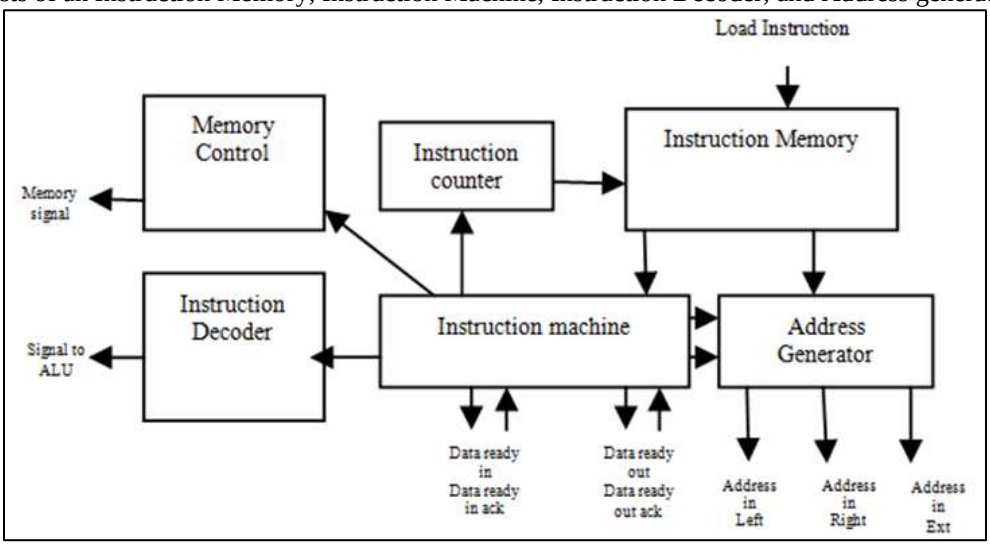

Fig. 3: Control Unit Design

The instruction memory is a 16 word SRAM with each instruction word being 92 bits wide. The instruction word consists of code of the instruction performed on the operand, base addresses of the operands on which the operation is performed, after performing the operation base address for storing the output of the operation, address for traversing the data memory, and to describe the number of times an operation is to be performed. Total of 28 instructions are support by RC, such as arithmetic, logic and memory instructions. Table 1 shows the list of supported instructions and a brief description of the action performed [1]. All these instructions are used to solve the critical arithmetical and logical problems very efficiently.

Table – 1
Description of Supported Instructions

Instruction	*Description*
ADD,SUB,MUL	*Add, subtract, multiply*
(ADD,SUB,MUL)ACC	*Add, sub, multiply with Accumulate*
NOT,AND,OR,NAND,NOR,XNOR	*Combinatorial operations*
CMP	*Compare*
SHIFT	*Shift*
MOVE_B	*Move(Bank to Bank)*
MOVE_O	*Move (RAM to Output)*
PASS_THROUGH	*Simple data pass*
JUMP	*Jump to particular Inst.*
JUMPIF	*Cond. Jump to part. Inst.*

V. MEMORY ORGANISATION

A prominent difference between MORA and various architectures in the same category is that MORA does not use a centralized RAM system. In MORA architecture each RC is provided with a 256x8 bits data memory bank and 8-bit processing element (PE). The Processing Element is a Computational Unit of RC and contains logic section and arithmetic section, with each MORA core requiring just over 60,000 transistors. MORA provides most of the arithmetic and logic functions and high throughput performance with minimum resource utilization. Each port has individual address decoders and read, write control signals. The data memory

allows each Reconfigurable cells to work as a small Processor In-Memory i.e. operations are performed close to memory. Ultimately this allows each RC to work independently of the others and eliminates the possibility of variance for memory resources among RCs. The result is in terms of power, area, memory access time, and reduction in complexity of the interconnect switches. The data memory is made up of 256x8 dual port SRAM array. Each port has its address decoders and read, write control signals. Fig. shows the organization of data memory in each RC. Signal S4 controls the multiplexers; it controls data flow into the memory. These multiplexers write the result of the current PE operation or data from external RC into the specified memory address depending on the instruction word, through the definite port.Signal S6 controls the output multiplexer; it controls the flow of data out of the data memory by aiming it either into the PE of the same RC or into the PE or data memory of a another RC. Media processing frequently requires operations based on matrices and vectors, it is required for the RC to be able to move data within its data memory. This is executed using an additional multiplexer which is controlled through S6. This allows the RC to read data from a memory location through the left port, and transfer it to another memory location, through the right. The read and write operations are implemented during reverse phases of the clock signal. This operation allows the RC to perform the MOVE operation in a single clock cycle. The interchangeably phased read and write allows the RC to perform the read-write sequence in a single clock cycle. As a result, each of the 28 instructions stated earlier take exactly one clock cycle to complete the process.

REFERENCES AND FOOTNOTES

[1] MORA – An Architecture And Programming Model For Resource Efficient Courde Grained Reconfigurable Process By Sai Rahul Chalamalasepti, Sohan Purohit, Martil Margale, Wim Vanderbauwhede- 2009 NASA/ESA Conference On Adptive Hardware Systems.
[2] Coarse Grained Reconfigurable Architectures In The Past 25 Years Overview And Classification By Mark Wijvliet, Luc Waeijen And Henk Coporaal.