ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

CGRA - A New Paradigm for Reconfigurable Computing

Sivakumar R[#1],Sarathkumar K.J[#2],Thilak M.B[#3],Janakiraman N[#4],Selva kumar[#5],

K.L.N. College of Engineering, Sivagangai, Tamilnadu, India

ABSTRACT — In recent years, there are an increasing number of surveys and many conferences being focused on Reconfigurable Computing. Many energy-efficient reconfigurable architectures and mapping algorithms have been proposed which provides high speed and energy saving by low power consumption. Also reduced size and component count results in low cost along with improved time-to-market, and improved flexibility and upgradability. Coarse Grained Reconfigurable hardware are mostly stressed due to their operator level CFBs, word level data paths, powerful and very area-efficient data path routing switches, promising for achieving energy-efficient flexible designs for an application domain. In this paper we focus on important aspects in the research and development of CGRAs.

I. INTRODUCTION

There are three common solutions to design architecture for an application. First method among them is Application Specific Integrated Circuit (ASIC). ASIC is specifically designed for a particular application and hence they are very fast and efficient. Once the fabrication is completed, then they cannot be altered. Hence if we need any changes in architecture then we are forced to redesign the entire architecture and also refabrication is required. This is an expensive process.

Second method is software programmed microprocessors, where the processor executes a set of instruction that is needed for computation. It is more flexible but lacks performance. It has high overhead because it needs to read the instruction from memory, decode it and then execute it. Also the instruction are only determined during fabrication. The Third method, Reconfigurable computing devices, bridges the gap between the ASIC and Software programmed micro-processors by providing flexibility greater than ASIC and performance greater than software programmed microprocessors. It uses reconfigurable hardware's (like FPGA) for computational purposes where the computation is done through functional elements called logical blocks that are configured through configuration bits.

Both FPGA and reconfigurable are used to speed-up the performance of various applications. This paper is organised as follows. The section II discusses about the Coarse-Grain Reconfiguration Architecture. Then the section III discusses about the related works where we will discuss about the available CGRAs. Then we will discuss about the characteristics of CGRA in section IV. Then in section V we will provide the conclusion and future works.

II. CGRA

Usually a reconfigurable block is combined with a general purpose microprocessor. In this way the processor maps the instructions that can be done efficiently to reconfigurable block and the microprocessor performs instructions that cannot be done efficiently in reconfigurable block.

The FPGA uses Fine grained reconfigurable arrays of functional elements. But this creates burden such as less efficient due to huge routing area and poor routability. Coarse-grained architectures provide better programmability and performance. In particular, coarse-grained reconfigurable architectures (CGRAs) have been proven to be effective solutions for computation-intensive applications such as multimedia applications, by providing higher efficiency in terms of power, area, and configuration cost when compared to fine-grained architectures such as FPGAs.

However, as is the case with many other forms of reconfigurable computing, the practical drawbacks of a CGRA have mostly been on the software side – primarily due to the complexities involving mapping applications to utilize such architecture. The application mapping is usually done manually, which is an extremely difficult task since a comprehensive understanding of the target architecture is required of the application developer.

In case of energy efficiency, the active power consumed by a reconfigurable fabric of functional blocks can be given as

$$P_a \approx N_a.F_a.V^2$$

Here N_a represents the number of functional blocks performing the computations, F_a is the operating frequency, and V is the supply voltage of the chip.

Coarse Grained Reconfigurable Architecture is a circuitry which is optimized for specific application. It consists of interconnection of array of processing elements (PEs) which consists of similar or different logical units. The CGRA consists of Host processor, Reconfigurable logic and Interconnection network.

1. Host Processor: The host processor may be VLIW processor (e.g. ADRES), DSP processor (e.g. Montium) or General purpose microprocessor (e.g. MOLEN). They execute non-loop and outer-loop code are latency constrained and do not offer significant amounts of instruction-level parallelism. In these situations, CGRAs are ineffective as the majority of the resources remain idle.

2. Processing Element: PE's are logical units that can perform basic ALU operations or Load/Store operations, which can be dynamically configured through the configuration cache. It may consist of homogenous or heterogeneous logical blocks. PEs are connected to each other PEs through the inter connection network. They enable sharing of results among them. The interconnect structure can vary from mesh-based to fully-connected.

3. Interconnection network: There are 5 types of interfacing techniques that are available.

    a. Type 1 External processing unit: In this architecture the Central Processing Unit (CPU) and reconfigurable units are separated. Here communication costs are relatively high.
    b. Type 2 Attached processing unit: In this architecture the reconfigurable unit is attached using the same I/O interface where other devices are connected. The communication cost is lower than the first architecture.
    c. Type 3 Co-processor: The reconfigurable unit acts as a coprocessor. The communication cost is lower than above architectures.
    d. Type 4 Reconfigurable functional unit: Here the reconfigurable functional units are within the host processors.
    e. Type 5 Embedded processor: In this architecture the CPU is embedded inside the reconfigurable fabric.

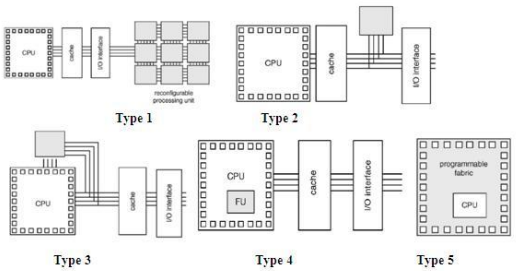

Fig. 1 Interconnection between CPU and Reconfigurable unit

III. RELATED WORKS

Many groups have developed coarse grain reconfigurable arrays over the last decade. We briefly discuss some of the architecture for better understanding of our work.

The Matrix Architecture: Multiple ALU architecture with Reconfigurable Interconnect experiment is a multi-granular array of 8-bit BFUs (Basic Functional Units) with procedurally programmable microprocessor core. Each BFU contains an 8 bit ALU, 256 words of 8 bit memory and control logic. The ALU features the standard set of arithmetic and logic functions and a multiplier.

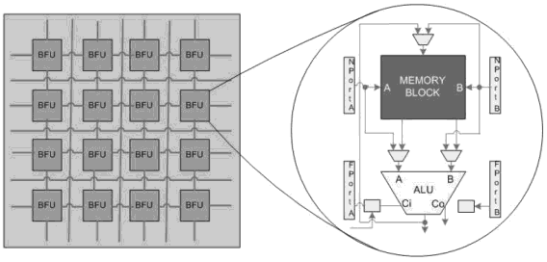

Fig. 2 MATRIX Architecture

A BFU can serve as an instruction memory, a data memory, a register-file-ALU combination, or an independent ALU function. The routing fabric provides 3 levels of 8-bit buses: 8 nearest neighbor (8NN) and 4 second-nearest neighbor connections, bypass connections of length 4, and global lines. Due to the routing resources of the MATRIX, instructions may be routed over the array to several ALUs. Thus, a high compression rate for instructions can be achieved.

RAW: RAW (Reconfigurable Architecture Workststation) system comprises of RAW microprocessor which is a homogenous array of processing elements called tiles. 16 tile is arranged in a 4 by 4 array in prototype chip. Each tile comprises a simple RISC-like processor consisting of ALU, register file and program counter, SRAM-based instruction and data memories, and a programmable switch supplying point-to point connections to nearest neighbors. The CPU in each tile of the prototype chip is a modified 32 bit MIPS R2000 processor with an extended 6-stage pipeline, a floating point unit, and a register file of 32 general purpose and 16 floating point registers.

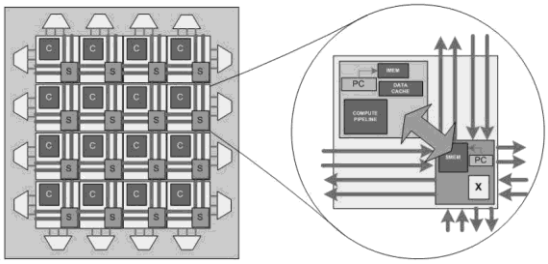

Fig. 3 RAW architecture

Both the data memory and the instruction memory consist of 32 Kilobyte SRAM. While the instruction memory is uncached, the data memory can operate in cached and uncached mode. If the data memory should be too small for an application, the virtualization of memories has to be done by software, which is generated by the compiler.

MorphoSys: The complete MorphoSys chip comprises a core processor, a frame buffer, a DMA controller, a context memory, and an array of 8 by 8 reconfigurable cells. The core processor is a MIPS-like TinyRISC CPU with an extended instruction set for manipulation of the DMA controller and the reconfigurable array. The frame buffer is an internal data memory for blocks of intermediate results, similar to a data cache. The DMA controller is used for both loading configuration data from the main memory and for data transfers between the main memory and the frame buffer. The context memory is used to store the configuration data for the reconfigurable array. This memory supports multiple contexts, which can be changed dynamically during execution.

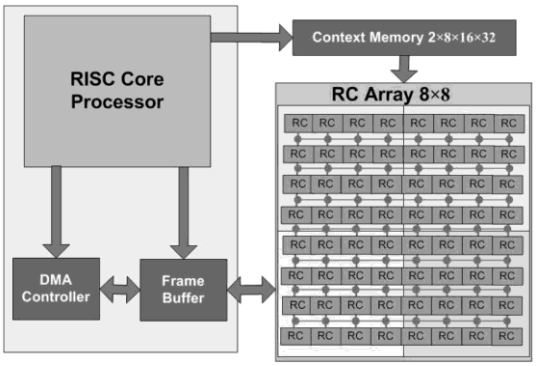

Fig. 4 MorphoSys Architecture

The reconfigurable part of MorphoSys comprises an 8 by 8 array of mesh connected processing elements, also called reconfigurable cells. The array is divided into four quadrants of 4 by 4 cells each. A single reconfigurable cell features a 16-bit datapath, comprising an ALU-Multiplier, a shift unit, two input multiplexers, a register file with four 16 bit registers, and a 32 bit context register for storing the configuration word. The multiplexers are used to select operands from the register file, the data bus from the frame buffer, or from other cells over the interconnect structure.

REMARC: The REMARC (Reconfigurable Multimedia Array Coprocessor) consists of an 8 by 8 array of processing elements, which is attached to a global control unit. The control unit manages data transfers between the main processor and the reconfigurable array and controls the execution of the nano-processors. It comprises an instruction RAM with 1024 entries, 64 bit data registers and four control registers. Communication lines contain nearest neighbor connections between each adjacent nano-processors and additional horizontal and vertical data buses for each row and column.

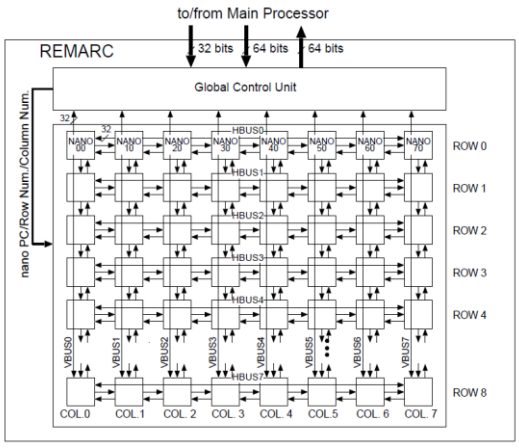

Fig. 5 REMARC Architecture

The Horizontal and vertical buses have double width (32 bits) and allow data from a data output

register to be broadcast to processors in the same row or column respectively.

IV. DISCUSSION

Many CGRA-based systems have been proposed in various papers and some of the models have been implemented. Each design has different scalability and performance. Here we will focus on these reconfigurable processors and critically analyse them on various aspects.

Based on the previous examples, we can now summarize the state of the art in CGRA and try to predict the main trends for the future. The Core idea in Matrix Architecture is to make the amount of the basic resources of a computing system. The basic resources of a computing systems are computational units, instruction storage and data memory. These basic resources should be adaptable to the application requirements.

The main aim of RAW is to obtain a simple, highly parallel computing architecture comprises of several dreadful tiles connected among each other by nearest neighbour connections. The tile consists of computation facilities as well as memory, leading to implement a distributed memory model. All architectural details are disclosed to the compilation framework. Especially, the processors lack hardware for register renaming, dynamic instruction issuing or caching, which is found in current superscalar processors. Due to the lack of these features, the execution model uses statically scheduled instruction streams generated by the compiler, thus moving the responsibility for all dynamic issues to the development software. However, RAW provides the possibility of flow control as a backup dynamic support, if the compiler should fail to find a static schedule.

MorphoSys is famous example of 8x8 grid with a more sophisticated interconnect network. In MorphoSys, each node has an ALU and a small local register file. RAW architecture is more general system which node is small MIPS processor with memory, registers, and a processor pipeline.

The REMARC architecture is a reconfigurable coprocessor aimed at multimedia applications like video compression, video decompression, and image processing. The processor has been extended by special instructions to access the REMARC architecture in the same way as a floating point coprocessor is accessed.

V. CONCLUSIONS

We have learnt many things while working on this survey, which of course will guide us to complete our final project. We have discussed only the architectural features of the CGRA's. Another important aspect that is needed for the high performance of CGRA models are smart algorithms that will enable us to implement our applications onto these fabrics. These applications are mapped to these CGRA's by the Compilers. A good compiler should be able to map an application with least possible number of PEs. Hence they form the important tool that increases the performance of the CGRA.

REFERENCES

[1]. J. Glossner, D. Iancu, L. Jin, E. Hokenek, M. Moudgill, "**A software-defined Communications baseband design**," Communications Magazine, IEEE, Vol. 41, pp. 120-128, 2003.

[2]. Bruno Bougard, Bjorn De Sutter, Sebastien Rabou, David Novo, Osman Allam, Steven Dupont, "**A Coarse-Grained Array based Baseband Processor for 100Mbps+ Software Defined Radio**".

[3]. Akira Hatanaka and Nader Bagherzadeh, "**A Modulo Scheduling Algorithm for a Coarse-Grain Reconfigurable Array Template**" 1-4244-0910-1/07/$20.00 _c 2007 IEEE.

[4]. Jong Kyung Paek, Kiyoung Choi, Jongeun Lee, "**Binary Acceleration Using Coarse-Grained Reconfigurable Architecture"** IEEE Vol. 38, No. 4, September 2010.

[5]. Reiner Hartenstein, "**Coarse Grain Reconfigurable Architectures**"

[6]. Yongjun Park, Hyunchul Park, Scott Mahlke, "**CGRA Express: Accelerating Execution using DynamicOperation Fusion"** IEEE CASES'09, October 11–16, 2009,

[7]. Allan Carroll, Stephen Friedman, Brian Van Essen, Aaron Wood, Benjamin Ylvisaker, Carl Ebeling,and Scott Hauck, "**Designing a Coarse-grained Reconfigurable Architecture for Power Efficiency**"

[8]. Yongjun Park, Jason Jong Kyu Park, and Scott Mahlke, "**Efficient Performance Scaling of Future CGRAs for Mobile Applications",** IEEE 978-1-4673-2845-6/12/$31.00 c 2012 IEEE.

[9]. Gregory Dimitroulakos, Michalis D. Galanis and Costas E. Goutis, "**Exploring the Design Space of an Optimized Compiler Approach for Mesh-Like Coarse-Grained Reconfigurable Architectures**",1-4244-0054-6/06/$20.00 ©2006 IEEE.

[10]. Tomoya Suzuki, Hideki Yamada, Toshiyuki Yamagishi, Daisuke Takeda, Koji Horisaki, Toshio Fujisawa, Yasuo Unekawa, Toshiba,Tom Vander Aa, Liesbet Van der Perre, "**High-Throughput, Low-Power Software-Defined Radio Using reconfigurable Processors**" 0272-1732/11/$26.00 @c 2011 IEEE.

[11]. Nozar Tabrizi, Nader Bagherzadeh, Amir H. Kamalizad, Haitao Du, "**MaRS: A Macro-pipelined Reconfigurable System**" Copyright 2004 ACM 1-58113-741-9/04/0004.

[12]. Hartej Singh, Ming-Hau Lee, Guangming Lu, Fadi J. Kurdahi, Nader Bagherzadeh, "**MorphoSys: An Integrated Reconfigurable System for Data-Parallel Computation-Intensive Applications**"

[13]. M.K. Jain, M. Balakrishnan, and A. Kumar. "**Asip design methodologies: survey and issues**". In Fourteenth International Conference on VLSI Design, pages 76–81, 2001

[14]. Hyunchul Park Yongjun Park Scott Mahlke, "**Polymorphic Pipeline Array: A Flexible Multicore Accelerator with Virtualized Execution for Mobile Multimedia Applications**" MICRO'09, December 12–16, 2009,

[15]. T.J. Todman, G.A. Constantinides, S.J.E. Wilton, O. Mencer, W. Luk and P.Y.K. Cheung, "**Reconfigurable Computing: Architectures, Design Methods, and Applications**"

[16]. Yoonjin Kim, "**Reconfigurable Multi-Array Architecture for Low-Power and High-Speed Embedded Systems**" JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.11, NO.3, SEPTEMBER, 2011.

[17]. Michalis D. Galanis, Gregory Dimitroulakos, and Costas E. Goutis, "**Speedups and Energy Reductions from Mapping DSP Applications on an Embedded Reconfigurable System**", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS" VOL. 15, NO. 12, DECEMBER 2007.

[18]. Akeem Edwards, "**Coarse Grained Reconfigurable Architecture**" July 29 2012.

[19]. Justin L. Tripp, Jan Frigo, Paul Graham, "**A Survey of Multi-Core Coarse-Grained Reconfigurable Arrays for Embedded Applications**"

[20]. Scott Hauck, "**The Future of Reconfigurable Systems**" Keynote Address, 5th Canadian Conference on Field Programmable Devices, Montreal, June 1998.

[21]. Andr´e DeHon, "**MASSACHUSETTS INSTITUTE OF TECHNOLOGY ARTIFICIAL INTELLIGENCE LABORATORY**" October, 1996.

[22]. Anne Pratoomtong, "**A survey on Reconfigurable Computing for Signal Processing Applications**"

[23]. Katherine Compton, Scott Hauck, "**An Introduction to Reconfigurable Computing**"

[24]. K. Compton, S. Hauck, "**Configurable Computing: A Survey of Systems and Software**", Northwestern University, Dept. of ECE Technical Report, 1999.

[25]. William H. Mangione-Smith, Brad L. Hutchings, "**Configurable Computing: The Road Ahead**"

[26]. Thijs van As, Siebe Krijgsman, "**Reconfigurable Architectures: A Survey of Design and Implementation Methods**".

[27]. KATHERINE COMPTON, SCOTT HAUCK, "**Reconfigurable Computing: A Survey of Systems and Software**", IEEE" Vol. 34, No. 2, June 2002, pp. 171–210.

[28]. Ken Eguro and Scott Hauck, "**Issues and Approaches to Coarse-Grain Reconfigurable Architecture Development**"

[29]. Adams, C. and J. Gilchrist. "**The CAST-256 Encryption Algorithm.**" First AES Candidate Conference, Aug. 20-22, 1998.

[30]. Ali Azarian, MahmoodAhmadi, "**Reconfigurable Computing Architecture Survey and introduction**" 978-1-4244-4520-2/09/$25.00 ©2009 IEEE.