Indian Journal of Science and Technology, *Vol 10(6), DOI: 10.17485/ijst/2017/v10i6/110837, February 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Survey on Multigrained Reconfigurable Architecture using Parallel Mapping Method

T. Siva Sankara Phani[1], B. Ananda Krishna[2] and Ranjan K. Senapati[1]

[1]Department of ECE, KL University, Green Fields, Vaddeswaram, Guntur – 522502, Andhra Pradesh, India, phani.tammireddy@gmail.com, ranjan.senapati@kluniversity.in
[2]Department of ECE, Guntur Engineering College, NH 5, Opposite Katuri Medical College, Yanamadala, Guntur - 522019, Andhra Pradesh, India; anand_bk@rediffmail.com

Abstract

Objectives: In previous methodology, only ALU operations are seen with respect to the network on chip with this power and delay is high whereas in present research, Multi-grained reconfigurable architecture is used which reduces power and delay. In this method we can also perform multiple FFT, DCT, FIR and channel encoder. **Methods:** For execution, the number of instructions is flapped by associating pipelining technique which is splitted in stages. Every stage completes an area of parallely connected instructions at a time. A pope is created, with the interconnection of stages where instructions are fed at amend, progresses through these stages and exits at other end. The group of Function Units (FU'S) was connected in a mesh style network in CGRA'S. Here, the subsets of FU's are accessed only with help of segregation of register files through CGRA's that which carries temporary values. In general word operations like addition, subtraction and multiplication were carried out by these function units. **Findings:** For better execution characteristics of parallel mapping on MGRA, more PE utilisation rate and less memory access overhead are considered as resulting conditions. Lastly, Multi Grained Reconfigurable Architecture (MGRA) is the proposed research work where processing element consists of a multiple operations like FFT, DCT, FIR, Channel Encoder etc. The proposed architectures require multiple processing elements to execute parallel process as to reduce PE's complexity. A new folding tree algorithm is proposed (MGRA) with CRGA is proposed to eliminate PE's. This method reutilizes PE's to redistribute data from multiple nodes and the controller is integrated with current CGRA to scan the processing nodes with common expression executions. By using nested loop pipelining the Multi grained reconfigurable architecture comprises the advantages of low power and delay when compared to the existing architectures. **Application:** MGRA used in communication, digital signal processing.

Keywords: CGRA, Multi-Grained Reconfigurable Architecture (MGRA), Network-on-Chip, Parallel Mapping, Reconfigurable Computing

1. Introduction

Spatial computing could be a method that usually uses significant amount of simple parallel processing factors, that which operate at a time, to execute a one application or application kernel. Examples of spatial computing systems are committed hardware, typically within the form of application designed built-in circuits, or in latest years, configurable hardware corresponding to area programmable gate arrays.

The configuration particulars are generated by the compiler through selective mapping on the reconfigurable fabric with the specification suited. This approach of generating this configuration knowledge leads to

**Author for correspondence*

hardware synthesis which links to placement and routing step. A few methods from the area of hardware synthesis are adopted for mapping the applications onto the CGRA Reconfigurable fabric. In this paper, the primary aspects of compiling onto a CGRA employing dataflow and execution paradigm on the orchestrator where overlooked. Further, the compiler is designed to focus on multiple Architectural variants of the reconfigurable fabric. The mapping process of utility partitions is examined by the unique illustration of the fabric.

Reconfigurable processors were extensively related to discipline Programmable Gate Array (FPGA) designs. A programmable logic cell matrix with a grid connected strolled strains is assembled in FPGA. Moreover, there were input- output pins on the area that furnish an associated interface between the FPGA and an interconnection between connecting strains and the pins of chips outside. However, to attain the attention of flexibility, the FPGAs are generally great-grained. This adaptability has its position for the computational requirements, which are either no longer prior noticed or vary notably chosen among various needed applications. However, in lots of instants this extreme stage of flexibility is senseless and would influence in massive overheads of area, prolong and power consumption.

Coarse-grained Reconfigurable Architectures (CRAs) draws additional concentration as it tends the soaring crisis and costs with custom hardware potencies. Nevertheless, where CRAs possess the advantage of both hardware's efficiency and also the software's flexibility with the shortage of adequate compilation technology of effective mapping applications (customarily loops). For the Reconfigurable ALU Array (RAA) with wide varieties of reconfigurable architectures[2], the challenge of mapping comes from the idiosyncrasies of the structure as well because the traits of the applying. To beat the boundaries and maximize the performance, some mapping sub problems have been addressed within the literature (e.g., temporal partitioning for constant array dimension and pipeline vectorization for throughput improvement[3], reminiscence operation sharing for restricted memory bandwidth[4] and knowledge context switching for loop-carried dependency). One of the drawback seen as core mapping of inserting and routing the operations of a loop body onto the ALU array within the context of CRAs is one that still requires a continuous study.

The combination of 2D array of Processing Elements (PEs) with interlined programmable interconnects constructs the reconfigurable ALU Array (RAA) architectures from the most renowned class of CRA'S. Even though these PE's perform limited operations like addition, subtraction and multiplication by using dynamic reconfiguration design while runtime of 2D array data path numerous algorithms can be designed with high end applications very efficiently. Generally SIMD-style computations are incorporated in traditional CRA's which are reliable for configuring, cache storage for sharing multiple data through instructions. But the drawback seen is individual PE's dependency in the execution models is highly limited. This initiated many researchers to work towards Multiple Instruction Multiple Data (MIMD)-style CRAs, where dependency od PE's is configured to project its own instructions separately that allowed more adaptable and reliable configurations of loop pipelining with a chance of loop's multiple iterations in a pipeline.

In between DSPs and FPGAs, the Coarse-Grained reconfigurable computing fabrics are seen with numerous features and application capabilities. The program languages like C or C++ were used to configure the tools to meet the market requirements as same as GPP and DSP. To attain the multi-level parallelism[7], various arithmetic operations with more flexible computational throughputs were also designed by using clusters of reconfigurable processors.

In contrast to FPGAs, the coarse grained Reconfigurable Architectures CRA's, the width of the data-path is more than one bit. Sincelast15 years, many projects have been investigated and effectively designed techniques where the reconfiguration is coarse-grained and is carried out within a processor or in group of processors. In such type of methods the reconfigurable unit is a specialized hardware architecture which helps as common reconfiguration as much faster than that of FPGAs. Due to this fact, the known appliance domain led to design full customized data paths, which are significantly lengthy and high in energy.

In this paper, a multi grained reconfigurable architecture design is proposed by using nested loop pipelining method thereby proposing this technique's advantages in MGRA that reduces the delay and power when compared to the previous works.

The rest of the paper is organized as follows. Section II describes the existing works on Reconfigurable Architectures. Section III we introduce the research work .the results were presented with discussions in Section IV and finally the conclusion is stated in Section V.

2. Existing Works

The hardware programmability, software modifications and operation & communication of architecture

is enabled in these reconfigurable architectures that are highly competitive to the existing programmable architectures. With the allocation of selected process, routing resources and memory initiated its conditions for operations and processes. Reconfigurable architectures possess more advantages than traditional application-specific hardware accelerators. Hence, the inactivated resources are reconfigured at the time of run time. The one more advantage of this architecture is enabling mapped functionality without any auxiliary hardware with no costs and also leads to extension of platform's life span. The *granularity* of the device is considered as the size of elements used in reconfigurable architectures.

To enable bit level manipulations, the Fine-grained devices generally utilizes small Look-Up Tables (LUT) and this feature made these devices highly different in nature and improved its application to suit effectively for any algorithm. Generally fine grained architectures are used to enable bit level manipulations that make these devices extremely versatile and suits it to any algorithm used. In hardware utilization, fine grained architecture's may be inefficient but here for processing input elements, it is settled separately for more number of clock cycles. When compared to fine grained architectures, the coarse grained architectures uses size of ALU's to full scale processors size for building block elements. In these coarse grained architectures, as shown in figure-1, the large computational elements are constructed in the form of arrays or like small programmable kernels and state machines. The advantages like low configured data leads to less reconfiguration time and also lower hardware overhead for routing resources is seen in this CGA construction.

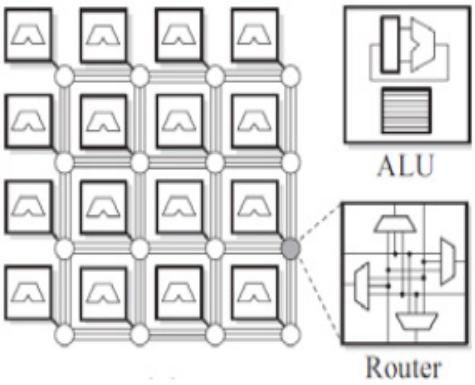

Figure 1. A Coarse-grained Reconfigurable Architecture - Array of Processing Elements (ALU) and a Routing Network.

The drawbacks seen in FPGA are overlooked in CGA with the help of an aid named as couple of bit broad knowledge paths. With the help of silicon intricate operator's efficiency of CGRA is improved for large information courses. The architecture shown in figure 1 is a coarse - grained reconfigurable architecture that consists an array of processing elements and a routing network. Hence, the routing overhead is evaded which generated because of compilation of difficult operators from bit-level processing models. The coarse grain reconfigurable architectures also have special features like, connecting multiple bits broad that generates usage for an individual line and another features that highlights CGRA when compared to FPGA is defining processing element orders. This additional feature leads a global cutback area usage for routing. It highlights the higher granularity and less cut down area for strain conversations as additional conversation assets which may be inefficient for quality grained architectures. Examples shows for such assets are time-multiplexed buses or international buses, which join each processing element[8].

The three step methodology is seen in CGRA:

Step1: The dressmaker conceives a universal model – "architecture mannequin" is defined as a mesh networked model that comprises of an array of coarse grained processing detail (PEs) which are surrounded by input resources, output resources and memory blocks.

Step2: "structure template"- where the dressmaker notifies the architecture model with specific parameters description. The template frames the granularity, the viable network interconnections, variety & disposition of PEs, and the institution of the memory accessories. The count of traces and columns needed for an array that represents width and peak, the reconfiguration contexts quantities to be owned, the PE's inside registers count and the interconnection community etc. was considered as parameters to administrate the specified features of the model chosen.

Step3: A structure instance is generated through fixing the worth of every template parameter.

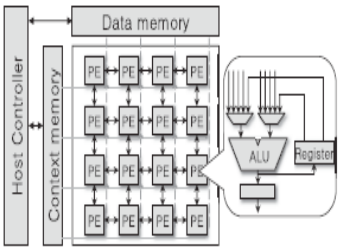

Figure 2. Target Architecture of CGRA.

The 2-D PEA, host controller, data memory and context memory were emphasized in the figure 2 is a typical CGRA architecture. Here PE comprises of number of local registers an Arithmetic Logic Unit (ALU) and a configuration register. Depending on the configuration words, the working of ALUs can be configured to variable word-level operations of fixed-point numbers. There are different combinations of networking for PEs is possible as mesh, mesh plus[9], and morphosys topology[10]. The configuration structure of CGRA can be mannered in two different ways such as: 1) Full-reconfigurable CGRA and 2) Partial-reconfigurable CGRA.

In Coarse grained architecture inputs are fed from one processing element to another processing element when the output is in the same clock cycle where as in CGRA architectures only ALU operations are performed.

2.1 PE-level Mapping

The micro operations were represented with the help of expression trees for the considered loop body in the PE-level mapping as shown in Figure 3.The production of PE –level operation trees is done with a single configured PE where the micro-operation trees are covered with patterns. This operation is an abstraction for a pattern of micro-operations which is developed within one configuration of a PE[11]. For example, if a series of ADD and STORE are used with more than two memory operations can be implemented with one configuration.

Example: In Figure 3, the first two operations may become one node and for the next consequent step, information of the number of memory operations contained in the node is necessary.

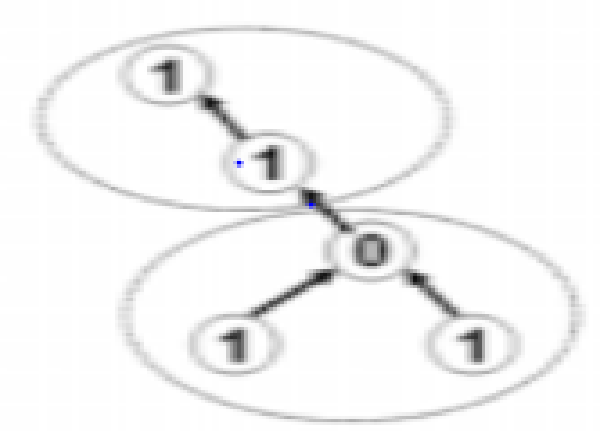

Figure 3. Grouping PE Level Operations.

2.2 Memory Operation Sharing

As shown in figure 3, the memory operation shares the information by aligning the PE-level operation tree on the same line. Here, the conditions for the memory operation's sharing nodes and the procedure to deal the addressing conditions within the mapping flow were identified.

The condition for the memory operation sharing is a technique that which explodes the redundancy of memory operations with in various iterations of a nested loop considered. The inference conditions drawn in this loop is to optimize, initially by the elimination of common sub expressions. Even though it is an optimized loop, there may be some various counter repetitions with same address for some memory operations and also that can be variably identified with those memory operations that call the same address in various iterations with a constant number, where these memory operations are named as "Alignable". From memory access indices, these Alignable operations are easily notified. For suppose, at every epoch, the loop iterate is noted as i and it is incremented by c then two memory read accesses were considered as A[a*i+s] and A[a*i+t] which are Alignable, if the difference(s–t) is divided by a*c. The iteration difference of the selected two memory operations accesses the ditto address in iterations only just by changing (s–t)/(a*c). If and only if the memory operations are pair wise Alignable, then only more than two operations are Alignable.

3. Proposed Multi Grained Reconfigurable Architecture

The reconfigurable architectures are possible to exhibit high speed computing over traditional microprocessor and digital signal processor DSP[12]. This Reconfigurable computing allows designers to control the power of hardware thereby providing the flexibility towards software. This reconfigurable computing systems use Field programmable Gate Arrays (FPGAs) to exploit high speed designs, at the same time it maintained tradeoff between area and power. Proposed Multi Grained Reconfigurable Architecture (CGRA) is a promising approach that performs parallel computation that combines both the high performance application-specific integrated circuits with greater flexibility of general purpose processors. Affine transformation with CGRA provides nested loop pipelining to increase computational speed and polyhedral

model is incorporated to overcome nested loop pipelining in CGRA. The affine transformation is tailored to exploit the parallelism in inner loops and reduce the outer-loop-carried dependence.

Using this approach, higher PE utilization rate and small memory access overhead can be attained, that results in better execution characteristics of Parallel mapping method on MGRA. Finally Multi Grained Reconfigurable Architecture (MGRA) is the proposed research work where processing element as shown in Figure 4 consists of a multiple operations like FFT, DCT, FIR and Channel Encoder etc.

The proposed architectures Figure 4 require multiple processing elements to execute parallel process as to reduce PE's complexity. A new folding tree algorithm is proposed (MGRA) with CRGA is proposed to eliminate PE's. This method reutilize PE's to redistribute data from multiple nodes and the controller is integrated with current CGRA to scan the processing nodes with common expression executions.

By using nested loop pipelining the Multi grained reconfigurable architecture comprises the advantages of low power and delay when compared to the existing architectures.

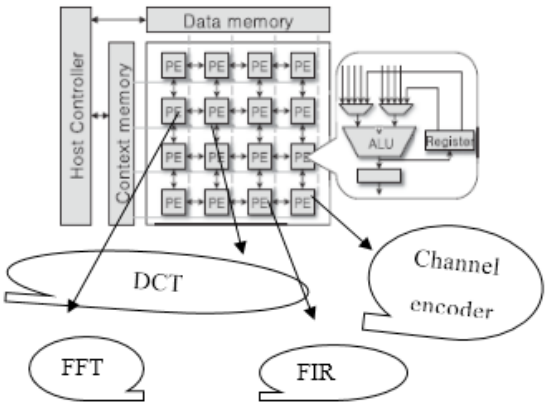

Figure 4. Proposed Multi Grained Reconfigurable Architecture.

4. Approximate Results and Performance Analysis

The Area and power of the proposed design is done by using QuartusII software with family Cyclone II. As shown in Figure 5, the total logical elements in this proposed design is 98 and total number of registers used here is 80.

The total thermal power dissipation in our proposed system is 45.43mW as shown in figure 6. The total thermal power is the Combination of Core thermal Dynamic thermal power dissipation 3.72mW, Core Static thermal power dissipation 18.04mW and I/O thermal power dissipation 23.67mW.

Flow Status	Successful - Thu Nov 24 11:15:25 2016
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	ALU
Top-level Entity Name	RCA
Family	Cyclone II
Met timing requirements	Yes
Total logic elements	98 / 4,608 (2 %)
Total combinational functions	98 / 4,608 (2 %)
Dedicated logic registers	80 / 4,608 (2 %)
Total registers	80
Total pins	53 / 89 (60 %)
Total virtual pins	0
Total memory bits	0 / 119,808 (0 %)
Embedded Multiplier 9-bit elements	0 / 26 (0 %)
Total PLLs	0 / 2 (0 %)
Device	EP2C5T144C6
Timing Models	Final

Figure 5. Area Report.

PowerPlay Power Analyzer Status	Successful - Thu Nov 24 11:17:56 2016
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	ALU
Top-level Entity Name	RCA
Family	Cyclone II
Device	EP2C5T144C6
Power Models	Final
Total Thermal Power Dissipation	45.43 mW
Core Dynamic Thermal Power Dissipation	3.72 mW
Core Static Thermal Power Dissipation	18.04 mW
I/O Thermal Power Dissipation	23.67 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

Figure 6. Power Report.

5. Conclusion

This paper presented a brief survey on Multi – grained reconfigurable architecture. With the comparison to the existing architectures which consists leading disadvantages of more number of clock cycles, more delay and power consumption where comparatively reduced and said to be in control by using proposed MGRA. The effectiveness of the proposed architecture is highly differentiated with the existing architures by showing the reduced the power and delay.

6. References

1. Hartenstein R, Grunbacher H. A The Roadmap to Reconfigurable computing. Springer-Verilog: Proceedings of Field-Programmable Logic and Applications. 2000 Aug; 27–30.
2. Ahmad Alsolaim, Jurgen Becker. A dynamically reconfigurable system-on-a-chip architecture for future mobile digital signal processing. European Signal Processing Conference. 1999.
3. Bindra A. Reconfigurable architectures chart a new course for DSPs. Electronic Design. 2002 August 5; p. 46–52.
4. Hartenstein R. A decade of reconfigurable computing: A visionary retrospective. Proceedings of the Design, Automation and Test in Europe. 2001; p. 642-49. https://doi.org/10.1109/date.2001.915091
5. Weinhardt M and Luk W. Pipeline vectorization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2001 February; 20:234-48. https://doi.org/10.1109/43.908452
6. Bondalapati K and Prasanna K. Loop pipelining andoptimization for run-time reconfiguration. Reconfigurable Architectures Workshop. 2000.
7. Singh H, et al. MorphoSys: An integrated reconfigurable system for data-parallel and computation-intensive applications. IEEE Transactions of Computers. 2000 May; 49(5):465-81. https://doi.org/10.1109/12.859540
8. Compton K. Northwestern University, Department of Electrical and Computer Engineering: Architecture Generation of Customized Reconfigurable Hardware, Doctor of Philosophy Thesis. 2003.
9. Bouwens F, Berekovic M, De Sutter B, Gaydadjiev G. Architecture enhancements for the ADRES coarse-grained reconfigurable array. Berlin, Germany: Springer-Verlag: High Performance Embedded Architectures and Compilers. 2008; p. 66-81. https://doi.org/10.1007/978-3-540-77560-7_6
10. Lee J, Choi K, Dutt N. Compilation approach for coarse-grained reconfigurable architectures. IEEE Design and Test of Computers. 2003 January/February; 20:26-33. https://doi.org/10.1109/MDT.2003.1173050
11. Cardoso J and Weinhardt M. Fast and guaranteed C compilation onto the PACT-XPPTM reconfigurable computing platform. The Proceedings of Field-Programmable Custom Computing Machines. 2002; 291–92.
12. Bondalapati K. Parallelizing DSP nested loops on reconfigurable architectures using data context switching. The Proceedings of Design Automation Conference. 2001;273-76. https://doi.org/10.1145/378239.378483