

# A Review on performance Optimization Techniques in Coarse-Grained Reconfigurable Architecture for Multimedia Applications

S.Munaf<sup>1\*</sup>, A.Bharathi<sup>2</sup>, A.N.Jayanthi<sup>3</sup>

<sup>1</sup>Assistant professor (Sr.Gr), ECE Department, SRIT.

<sup>2</sup>Professor, IT Department, BIT

<sup>3</sup>Associated Professor, ECE Department, SRIT

\*\*\*

**Abstract** - High grade of flexibility and efficiency, coarse-grained reconfigurable architectures (CGRAs) are acceptable for the realization of computing-intensive applications. The context cache memory handling issues are create a performance degradation Reconfigure the context cache unit either in PEs or hybrid context replacement algorithm to reduce memory overhead issues. In a application side each operations having different level co computation time, memory requirement due to this kind of process the overall performance get degrade. A mapping also a flow for improving application's performance by accelerating critical software parts, called kernels, on the Coarse- Grain Reconfigurable Array is proposed. This study gives a path to enhance the performance of CGRA in Application Level.

**Key Words:** Cache replacement algorithm, reconfigurable architecture (CGRA), context cache, context management, multimedia applications.

## 1. INTRODUCTION

A Reconfigurable technique plays is a very process in the transitional among Application Specific Integrated Circuits (ASICs) and general-purpose processors. In multimedia and DSP applications to enhance their looping performance Coarse-grain reconfigurable is more suitable one. CGRA having an array of Processing Elements (PEs) with 16 bit ALU connected with a Reconfigurable interrelate arrangement. With this Array structure can able to reduce delay, area, power consumption [1]. To accelerate the DSP application's a perfect detected kernel along with good mapping algorithm supports to improve application's performance. In the CGRA Priority-based mapping algorithm is most suitable one. For enhancing increasing the working clock frequency. It is restricted by the delay of critical path, and results in without priority work queue formed and hence higher power consumption, thus affects energy efficiency seriously. If we go to expand the array size of CGRA by take on more computing resources to obtain higher processing parallelism. Then the reconfigurable arrays reconfiguration becomes very critical, since they required more reconfiguration process and also dynamic reconfiguration of configuration data very difficult.

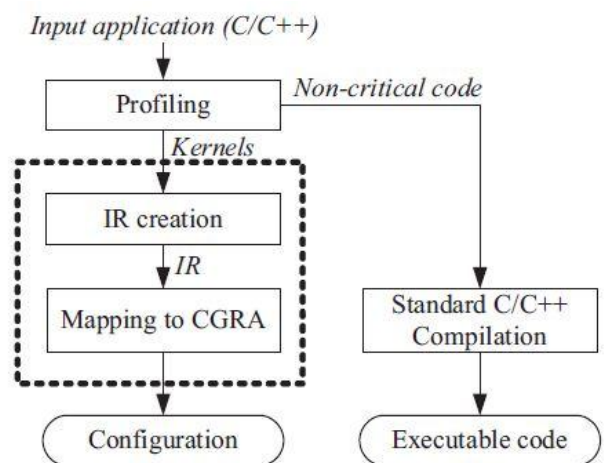


Fig 1 Mapping flow diagram for processor- CGRA Architecture

## 2. Mapping flow

Profiling is a form of dynamic program analysis that estimates, the space (memory) or time density of a program, the usage of particular instructions, or the frequency and duration of function calls. The mapping flow is visualized in Figure 1. Profiling is performed in the input Source code for identifying the critical code sections. This step output is the kernels and the noncritical code segments. The kernels will be mapped on the CGRA, while the non-critical code will be executed on the processor. For mapping the critical parts on the CGRA, the CDFG is a model of computation are used as an Intermediate Representation (IR) The Control Data Flow Graph (CDFG) model of computation selected and extensively used in mapping as the IR.

The communication mechanism requirement between the processor and the CGRA are replaced by kernels software description with calls to CGRA. As a call to CGRA is reached in the software, the processor initiates the CGRA and the suitable design is loaded on the CGRA for executing the kernel. The data required the execution are handled by the shared data memory. When the CGRA executes a specific significant software part, then the processor in an idle state for reducing power consumption. After the execution, the CGRA introduces a direct interrupt to processor the data required for executing the remaining software. Then, the

execution of the software is contained on the processor and the CGRA remains idle.

The total execution cycles after partitioning the application on the processor and the CGRA are:  $Cycles_{hw/SW} = Cycles_{proc} + Cycles_{CGRA}$  where  $Cycles_{proc}$  [2] represents the number of cycles needed for executing the non-critical code, and  $Cycles_{CGRA}$  corresponds to the cycles that are required for executing the software kernels on the CGRA. The  $Cycles_{CGRA}$  have been normalized to the clock frequency of the microprocessor. The  $Cycles_{thw/sw}$  are multiplied by the clock period of the processor for calculating the total execution time  $thw/sw$  after the partitioning.

### 3. CGRA PE Architecture

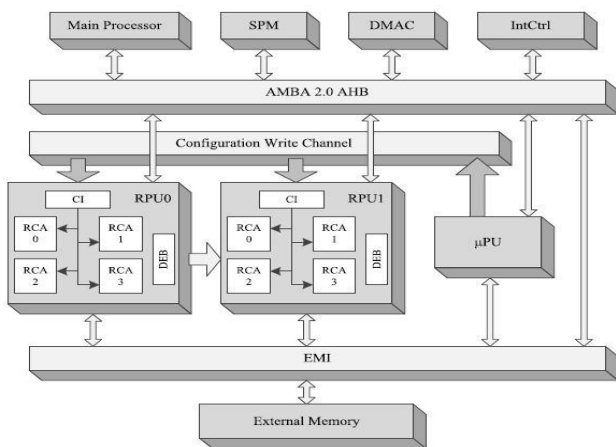


Fig 2 CGRA architecture

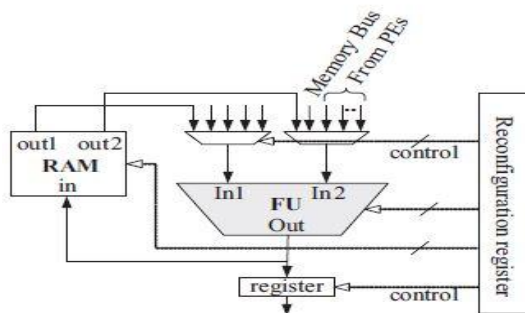


Fig 3 CGRA PE architecture

The configuration memory structure of the CGRA shown in (Figure 2&3), it holds the whole configuration for setting for the execution of application's kernels. Configuration caches distributed in the CGRA and to enhance operation speed reconfiguration registers available in the PEs. A configuration cache stores a few contexts locally, which can be uploaded on one by one basis. The configuration contexts can also be loaded from the configuration memory.

### 4. Multilayer Context Structure

In order to improve their efficiency and understand arrangement of pipeline the context structure organized with three layers, which are called configuration word (CW), context group (CG), and core context (CC), respectively, as shown in Fig. 4. The CW shows the index of the mapped RCA and the index of its corresponding CG, as well as the data transmission command for the RCA[9]. The layer2 describes the number of CCs for the RCA and the indexes of these CCs in sequence. For different number. The third layer, incorporates the detail configuration for the operations and routers of RCA.

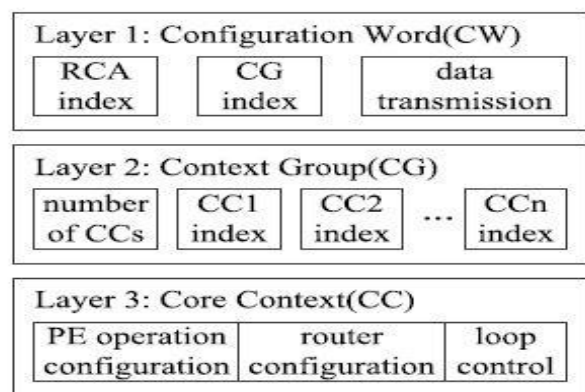


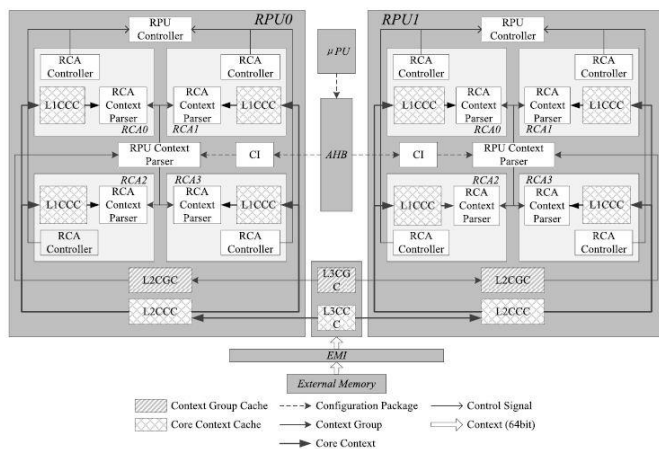
Fig 4 Multilayer Context Structure

In order to improve their efficiency and understand arrangement of pipeline the context structure organized with three layers, which are called configuration word (CW), context group (CG), and core context (CC), respectively, as shown in Fig. 4. The CW shows the index of the mapped RCA and the index of its corresponding CG, as well as the data transmission command for the RCA[9]. The layer2 describes the number of CCs for the RCA and the indexes of these CCs in sequence. For different number. The third layer, incorporates the detail configuration for the operations and routers of RCA.

The three features of the reconfiguration process are demonstrated and analyzed for multimedia applications, including temporal locality, nonuniform access frequency, and no uniform computation parallelism. The temporal locality describes the objective regularity of reconfiguration process that some certain tasks or subtasks of the target application.

The nonuniform access frequency study the variation of access frequencies of different configuration contexts within the whole application, which is display in the contexts for tasks and subtasks. During the processing of multimedia application, some kernel tasks among the most 80% regularly used, such as prediction, IDCT, and MC, occupy only 10% of the configuration contexts of CG and CC. Each subtask corresponds to an individual computing mode, whose average access times of contexts vary a lot.

### 5. Hierarchical Context Cache Structure



**Fig 5 Hierarchical Context Cache Structure**

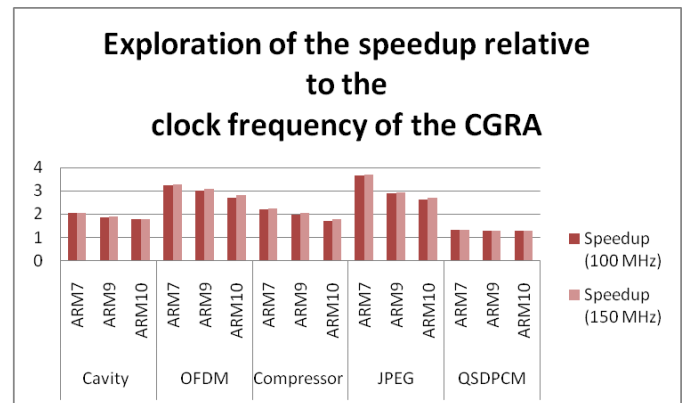
In this architecture, the context caches of CG and CC are reconstructed hierarchically for the level of RPU, RCA, and PE array, where the CGCs are composed of L2CGC (level 2 CGC) and L3CGC (level 3 CGC), while the CCCs include [9]L1CCC (level 1 CCC), L2CCC (level 2 CCC), and L3CCC (level 3 CCC). The proposed context management having advantage from two aspects. First, the context cache size is decreased considerably in average PE array scale. Second, the hybrid context replacement strategy improves the performance with the utilized context cache storage. In spite of the difference in the working frequency, the context cache size.

### 4. Description of Algorithm

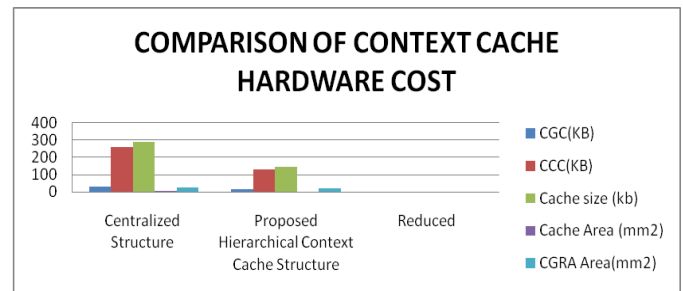
The algorithm is applied to all the application’s kernels, one at a time, for computing the execution cycles on the CGRA [3]. The description of the CGRA architecture is the second input to the mapping process. A mapping application is a scheduling operation, ie mapping the specific PEs and routing the data through specific interconnection. DFG is the first input value to mapping algorithm, seconds description of CGRA, For This CGRA modeled as a unidirectional graph (GA (VP, EI)).

The PE selection is decided by place decision (PD) [4] each PD’s has different kind of operations and execution scheduling, this difference degrades the speed of operation. To enhance the speed they are selecting a priority list based mapping. The priority decision based on the difference between as late as possible completion of execution to As early as possible. The algorithm implemented and executed by C++ Compilers. They are split as two phases of operation one is queue operation phase and another one is ready operation phase with the help of do while looping they are scheduling their operations. So that more critical path issues are reduced, it enhances their speed of operation.

### 5. Results and Discussion



They application speedups for these two different clock frequencies of the CGRAs are shown here. From these results it is deduced that the speedup slightly variation based on the clock frequency of the CGRA becomes smaller. The average speedup for the five applications and for the three ARM-based systems are 2.23 for the clock of 100 MHz, while for the 150 MHz clock. The average speedup is slightly larger since it is equal to 2.27. In this case, the system’s energy consumption is expected to be reduced.



Compared with the centralized context cache structure in the base architecture, the proposed hierarchical context cache structure is with only half size and 43% circuit area, leading to 9% deduction in the area of the total CGRA.

### Conclusion

A secluding flow of mapping technique improving system performance by executing critical kernel code on the coarse-grain reconfigurable hardware of a shock was studied. Above the Results from mapping and context reconfiguration technique CGRA platform show that the CGRAs are efficient in accelerating in important overall performance improvements. In future combine this we can enhance the memory band with issues.

### References

[1]. R. Hartenstein, “A Decade of Reconfigurable Computing: A Visionary Retrospective”, in Proc. of ACM/IEEE DATE ’01, pp. 642-649, 2001.



[2]. N. Bansal, S. Gupta, N. Dutt, A. Nikolau, R. Gupta, "Network Topology Exploration of Mesh-Based Coarsegrain Reconfigurable Architectures", in Proc. of ACM/IEEE DATE '04, pp. 474-479, 2004.

[3]. G. Stitt, F. Vahid, S. Nematbakhsh, "Energy Savings and Speedups from Partitioning Critical Software Loops to Hardware in Embedded Systems", in ACM Trans. On Embedded Computing Systems (TECS), vol.3, no.1, pp.218-232, Feb. 2004.

[4]. G. De Micheli, Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994.

[5]. J. Becker, M. Vorbach, "Architecture, Memory and Interface Technology Integration of an Industrial/Academic Configurable System-on-Chip (CSoC)", in Proc. Of Workshop VLSI (WVLSI '03), IEEE Press, pp. 107-112, 2003.

[6]. J. Becker, A. Thomas, "Scalable Processor Instruction Set Extension", in IEEE Design & Test of Computers, vol. 22, no. 2, pp. 136-148, 2005.

[7]. B. Mei, S. Vernalde, D. Verkest, R. Lauwereins, "Design Methodology for a Tightly Coupled VLIW/Reconfigurable Matrix Architecture, A Case Study", in Proc. of ACM/IEEE DATE '04, pp. 1224-1229, 2004.

[8]. Michalis D. Galanis, Gregory Dimitroulakos, and Costas E. Goutis "Mapping DSP Applications on Processor Systems with Coarse-Grain Reconfigurable Hardware" 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), Proceedings, 25-29 April 2006.

[9]. Peng Cao, Bo Liu, Jinjiang Yang, Jun Yang, Meng Zhang, and Longxing Shi "Context Management Scheme Optimization of Coarse-Grained Reconfigurable Architecture for Multimedia Applications" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Volume: 25, Issue: 8, Aug. 2017.

[10] F. Thoma et al., "MORPHEUS: Heterogeneous reconfigurable computing," in Proc. Int. Conf. Field Program. Logic Appl., 2007, pp. 409-414.

[11] B. Liu et al., "Reconfiguration process optimization of dynamically coarse grain reconfigurable architecture for multimedia applications," IEICE Trans. Inf. Syst., vol. E95-D, no. 7, pp. 1858-1871, 2012.

[12] Y. Wang et al., "On-chip memory hierarchy in one coarse-grained reconfigurable architecture to compress memory space and to reduce reconfiguration time and data-reference time," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 5, pp. 983-994, May 2014.

[13]. L. Liu et al., "An energy-efficient coarse-grained reconfigurable processing unit for multiple-standard video decoding," IEEE Trans. Multimedia, vol. 17, no. 5, pp. 1706-1720, Oct. 2015.

[14]. M. Kim, J. H. Song, D.-H. Kim, and S. Lee, "Hybrid partitioned H.264 full high definition decoder on embedded quad-core," IEEE Trans. Consum. Electron., vol. 58, no. 3, pp. 1038-1044, Sep. 2012.

## Authors



**Mr. S. Munaf**, received his master's degree in VLSI Design from Anna University of Technology, Coimbatore and his B.E degree in Electronics and Communication Engineering from Anna University Chennai. He received his Diploma in Electronics and Communication Engineering from State Board of Technical Education, Chennai. Now Pursuing Ph.D under Anna University in the area of High Performance VLSI design. He is having 14 years of teaching experience.



**Dr. A. Bharathi**, received her Doctoral Degree in Information and Communication Engineering under Anna University. Field of Specializing in Data Mining. She received her Post Graduate Degree under Anna University and did her Bachelor's Degree at Bharathiar University. She has over 20 years of Teaching Experience.



**Dr. A. N. Jayanthi**, received her Ph.D. degree in the Faculty of Information and Communication Engineering from Anna University. She received her M. E degree in VLSI Design from Anna University and her B. E degree in Electronics and Communication Engineering from Bharathiar University. She is having 18 years of teaching experience. Her area of specialization is VLSI.