

Survey on Coarse Grained Reconfigurable Architectures

Vaishali Tehre

Research scholar

Electronics Department

G.H. Rasoni College of Engineering Nagpur

Ravindra Kshirsagar, PhD.

Vice Principal

Priyadarshini College of Engineering, Nagpur

ABSTRACT

Recent advancement in the semiconductor technology allow the hardware engineers to integrate complex modules like processors, peripheral devices, and memory in a single *System-on-a-Chip (SoC)*; where testability, power minimization and management, area minimization are the important system level considerations. Performances both in terms of processing speed and power consumption are becoming more and more challenging in SOC designing. Novel system on chip architectures should be able to execute multiple performances demanding applications while maintaining low power consumption, small area, non-recurring engineering costs and short time to-market. Hence a lot of research is going on to implement CGRA in SOC because Coarse-grained reconfigurable architecture can provide both performance and flexibility. This paper gives a guided tour over a decade of development in CGRA and their significance in SOC design

1. INTRODUCTION

Starting in the mid 1970, Designers trying to implement more functionality into a small space by creating system on chips. Due to advancement in the technology computationally intensive applications are becoming integral part of embedded system which enables designers to implement increasingly complex SoCs using both Application Specific Integrated Circuits (ASICs) and Field Programmable Gate Arrays (FPGAs).

FPGAs are widely used as a reconfigurable solution to accommodate new system requirement or updates in various applications which is not feasible in ASIC implementation. The flexibility and short time to market are the advantages of the fine-grained FPGA implementations which provide better performance for various applications in SoCs. But when computationally intense applications like signal processing, multimedia, communications are implemented on FPGA; it comes with a significant more cost of area, power consumption and speed, due to the huge routing area overhead and timing penalty in comparison with customized application specific integrated circuits (ASICs). To solve this problem, a number of reconfigurable systems have been developed using coarse-grained structures [1, 2] which provide the flexibility of software combined with the performance of hardware. They also have the potential to bridge the performance/power gap between FPGAs and ASICs.

In recent years the consideration of coarse grained reconfigurable architectures in SOC design is growing [3]. Many researchers are trying to design a CGRA as a part of SOC; so that power and area can be minimize, while maintaining high performance, efficiency and flexibility [4, 5]. Maximum of the work in reconfigurable area has focused on the efficient design with respect to system performance and compiler. Power consumption is another important aspect in the reconfigurable architecture designs. This paper, gives the

survey of Coarse Grained Reconfigurable Architecture development in last decade and their application domain.

The rest of the paper is arranged as follows. Section 2 describes the basic architecture of Coarse Grained Reconfigurable Architecture. Section 3 focus on the development of different coarse grained architecture and describe the factors that were considered in the designing of coarse grain architectures and their target application. Section 4 concludes the paper

2. COARSE GRAINED RECONFIGURABLE ARCHITECTURES

By providing multiple-bit wide data-paths and complex operators a coarse grained reconfigurable architectures tries to overcome the disadvantages of bit-level configurability in FPGA-based computing. The wide data-path in CGRA allows efficient implementation of complex operators in silicon. Thus, the routing overhead is avoided which generated due to compilation of complex operators from bit-level processing units. The coarse grain reconfigurable architectures also have different ways to interconnects processing elements. The connections are multiple bits wide, which implies a higher area usage for a single line. On the other hand, the number of processing elements is typically several orders of magnitude lower than in an FPGA. Thus, much fewer lines are needed, resulting in a globally lower area usage for routing. The lower number and higher granularity of communication lines allows also for communication resources, which would be quite inefficient for fine grained architectures. Examples for such resources are time-multiplexed buses or global buses, which connect every processing element [6].

The CGRA follows three steps in their development, in first step; the designer conceives a generic model: an array of coarse grained processing element (PEs) interconnected by mesh like network which is surrounded by input and output resources and memory blocks. This model is called architecture model. In the second step, the designer writes down the architecture model as a parameterizable description which is called architecture template. The template outlines the granularity, type and disposition of PEs, the possible network interconnections, and the organization of the memory components. Templates are flexible descriptions because they can be modified by adjusting the value of parameters. Parameters regulate certain characteristics of the architecture, such as the number of lines and columns (width and height) of the array, the number of available reconfiguration contexts, the number of internal registers within the PE, and the interconnection network. At the third step, an architecture instance is generated by fixing the value of each template parameter. The architecture instance is a well defined description of architecture, which may be synthesized, evaluated, and/or simulated.

3. CGRA DEVELOPMENT

In the past years, several approaches for coarse grained reconfigurable architectures have been published. In this paper, several example architectures are presented to give an overview over the developments in the area of coarse grain reconfigurable computing. A time line of the presented architectures is given in Table 1 together with basic properties of these architectures.

Table 1

Public Year	Name of system	PE arrang ement	Reconfi guration model	Data path width	Application domain
1996	RaPid	Linear array	Static	16 bit	DSP Application
1998	Morph osys	Mesh	Dynamic	8 or 16 bit	Multimedia
1999	Chess	Mesh	Static	4 bit	Multimedia, Motion Estimation
2001	DReA M	array	Dynamic	8 or 16 bit	Mobile communicati on and DSP
2003	DRAA	Linear Array	Dynamic	8 bit	Multimedia
2005	ADRE S	array	Dynamic	32 bit	Multimedia
2007	MORA	Linear array	Dynamic	8 bit	Multimedia
2008	DRMP	1D array	Dynamic	32 bit	Wireless Communicat ion, MAC
2009	PACT XPP- III	array	Dynamic	16 bit	High performance Signal processing
2009	FloRA	2D array	Dynamic	24 bit	Kernels, multimedia, DSP
2010	SmartC ell	array	Dynamic	8 bit	Multimedia ,DSP
2011	SYSC ORE	array	Static	32 bit	biomedical monitoring application

3.1. The Reconfigurable Pipelined Datapath (RaPiD) Architecture

RaPiD is a coarse-grained architecture developed in mid 90s. It was targeted for DSP applications [7] [8]. It consists of linear array of functional units (ALUs, multipliers, Registers and RAMs). The functional units in RaPiD are interconnected using a set of ten segmented buses that run the length of the data path. The buses in different tracks resegmented into different aims at speed-up of highly regular, computation-intensive tasks by deep pipelines.

3.2. MorphoSys Architecture

This architecture has been designed to operate on 8 or 16-bit data, to provide high performance for word-level operations. The variable wire propagation delays are not in MorphoSys which are characteristic of FPGAs. This reconfigurable architecture was targeted for the applications in multimedia domain [9] [10]. It combines both coarse grain and fine grain reconfiguration techniques to optimize hardware (See figure 1). The MorphoSys has comparable performance to ASICs with the added benefit of being able to be reconfigured for different applications in one clock cycle. As shown in Figure 1 RC-Array is the reconfigurable part of the system which

contains an 8 by 8 array of reconfigurable cells (RCs). The context memory is used to store configuration data. During the execution, the context word is loaded from the memory to the context registers of the reconfigurable cell. In the MorphoSys there is an embedded data memory which gets the data from the external memory and feed the RC-Array with the appropriate data. The DMA controller controls all data movements between the MorphoSys memory elements and the external memory. The general purpose 32-bit RISC processor controls the sequence of operations in MorphoSys.

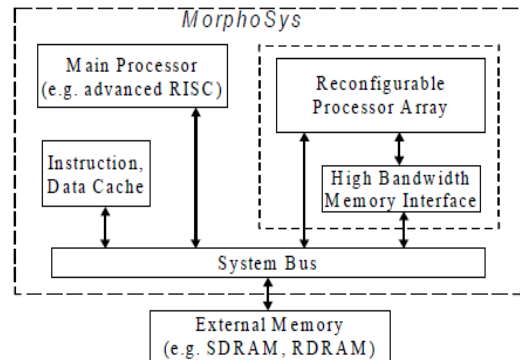


Figure 1

3.3. The CHESS Architecture

A reconfigurable architecture CHESS (figure 2) targeted for applications like multimedia, motion Estimation [11]. The fundamental computation component in this architecture is a 4-bit Arithmetic array (ALU) with 16 instructions. Each ALU has adjacent switchbox which serves as a cross point with 64 connections. Hence it needs about 64 bits to configure the switches and connections. The 4-bit buses are used for routing. As each ALU has a corresponding switchbox associated with it the routing area consumes up to 50% of the total area. Embedded RAM areas support high memory requirements. Switchboxes can be converted to 16 words by 4 bit RAMs if needed. RAMs within switchboxes can also be used as a 4-input, 4-output LUT. An ALU data output may feed the configuration input of another ALU, so that its functionality can be changed on a cycle-per cycle basis at runtime without uploading.

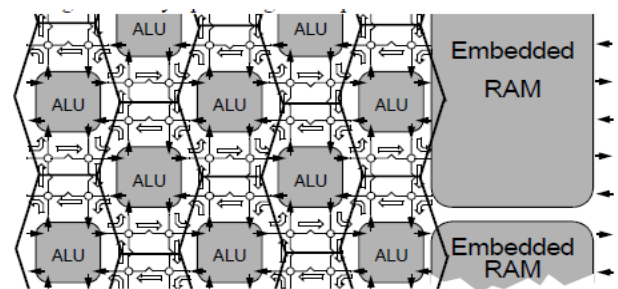


Figure 2

3.4. Dynamically Reconfigurable Architecture for Mobile Systems (DReAM)

This is a 16 bit Reconfigurable Architecture mainly designed for mobile devices [12]. Basically it consist of reconfigurable processing units(RPU) which is responsible for executing all required arithmetic data manipulations for the data-flow oriented mobile application parts, as well as to support necessary control-flow oriented operations. The complete

DReAM array architecture connects all RPU's with reconfigurable local and global communication structures (see figure 3). In addition, the architecture will provide efficient and fast dynamic reconfiguration possibilities for the RPU's as well as for the interconnection structures, e.g. only partly and during run-time while other parts of the reconfigurable architecture are active. Paper [13] describes the potential of DReAM for SoC-solutions in adaptive air interface candidate systems for future generations of wireless communication systems. DReAM can be easily updated for future mobile signal processing to provide an acceptable trade-off between flexibility and application performance requirements. Those application parts which are too complex for the DSPs and require different types of flexibility, so that ASIC implementations would be also not possible can be mapped onto this reconfigurable architecture. A special bridge is responsible for the data transport between the DReAM array and the system bus. By separating this conversion from the DReAM array the design is more flexible and can easily be adapted to other SoC buses like the Core connect architecture

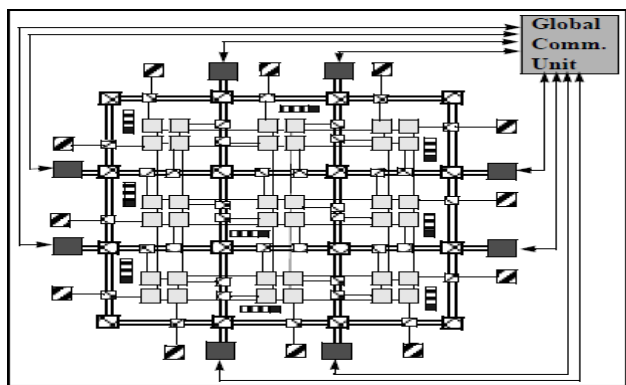


Figure 3

3.5. Dynamically Reconfigurable ALU Array

The DRAA is a coarse grained generic architecture template [14]; which defines RAA architectures that are characterized by a regular ALU array data-path with a fast memory interface.

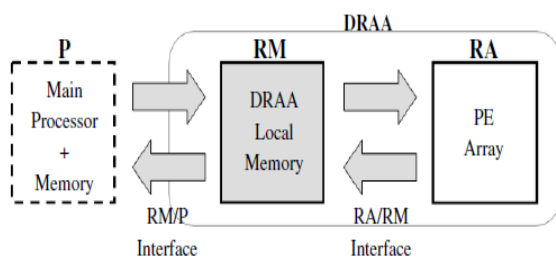


Figure 4

Figure 4 provides a simplified Performance oriented view of DRAA. This model has two components: the PE array (RA) and the memory subsystem. The memory subsystem further consists of three components: the DRAA local memory (RM), the RA/RM interface, and the RM/P interface where P stands for the processor-memory subsystem. The RM local data memory is essential block present in the DRAA to reduce the memory access cycles for media applications. This memory has a finite capacity, which will determine the amount of data caching. The RA/RM interface represents the memory

interface of the PE array, typically with large bandwidth. The RA/RM and the RM/P interfaces have generic array-level operations associated with them. The RM/P interface, which represents the main memory interface of the RM, often employs the DMA (Direct Memory Access) capability. Besides the parallel memory access and data caching (to exploit data locality), the memory subsystem may offer hardware addressing support to the PE array. For deterministic memory accesses (especially when dealing with stream data), the memory subsystem may be instructed to provide data according to the scan pattern of the application, thus eliminating the need for the PE array to generate addresses and requests to the memory subsystem. Memory architectures for performance estimation, provides a simplified view of the memory subsystem of DRAA architectures

The experiments of DRAA using multimedia as benchmarks shows that the memory architecture can have quite different effects on the application performance depending on the characteristics of the application, which also highlights the need for memory architecture evaluation early in the design process

3.6. ADRES (Architecture for Dynamically Reconfigurable Embedded System)

This is flexible processor architecture template designed for embedded application in SoC with low cost targets in terms of area and power consumption [15, 16, 17, and 18]. A VLIW processor is the main processing unit of ADRES which include array of tightly coupled configurable processing cells for purpose of acceleration (figure 5). The VLIW is programmable processor and it has a virtually unlimited capacity of operations. In contrast, the accelerator has a limited stack of operations. The local data register file is a part of the configurable processing units, which supports modulo technique. In acceleration mode the functional units of the VLIW form the first row of the array. Data transport within the array is carried out by orthogonal busses. Data input from various possible sources can be done with the help of two configurable port sides in the functional units. Data output is possible also for both horizontal and vertical distribution. The data-width through the whole architecture is 32 bits. There are up to eight functional units organized in a row in the VLIW. The horizontal data bus is used for the communication of units with each other. With the common register file a part of the units can communicate vertically for data load and store. After an initial boot phase, reconfiguration can be done dynamically without stalling the array.

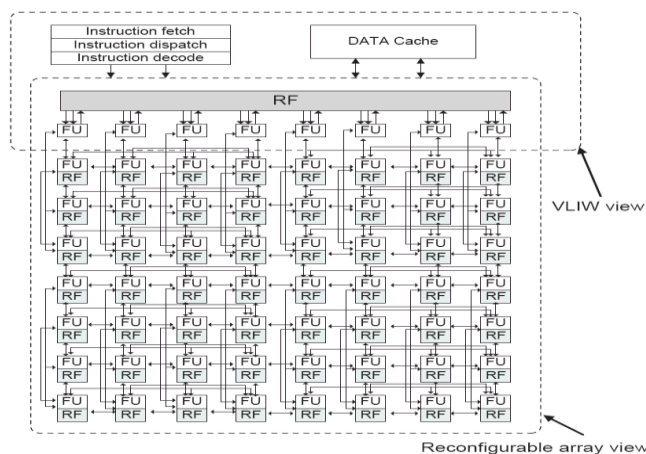


Figure 5

3.7. MORA Multimedia oriented Reconfigurable array)

MORA consists of a scalable 2D array of identical Reconfigurable Cells (RCs) organized in 4X4 quadrants and connected through a hierarchical reconfigurable network [19]. A 256*8-bit SRAM acting as an internal data buffer, an 8-bit Processing Element (PE) and a Control Unit incorporating the Configuration Memory are the main elements of this architecture. The latter holds the control program of the RC, which is loaded during the “configuration phase” of the system. The MORA architecture (Figure 6) can loosely be compared to an island-style FPGA architecture where the Configurable Logic Blocks (CLBs) are replaced by a small DSP-style processor, called Reconfigurable Cell (RC). In MORA storage for data is partitioned among RCs by providing each RC with internal data memory. Each individual RC is a tiny Processor-in-Memory (PIM). This approach allows computations to be performed close to memory, thereby reducing memory access time and power hungry interactions between logic elements and data memory. It also provides for high memory access bandwidth needed to efficiently support massively parallel computations required in multimedia applications while maintaining generality and scalability of the system

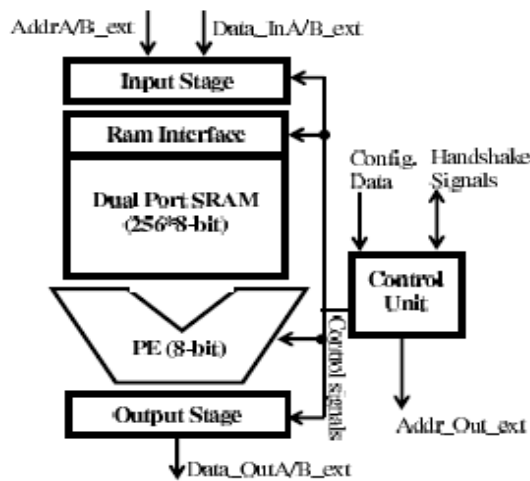


Figure 6

3.8. Dynamically Reconfigurable MAC Processor (DRMP)

The Dynamically Reconfigurable MAC Processor is a coarse-grained, dynamically reconfigurable SOC architecture, designed specifically for implementing the wireless MAC layer in consumer hand-held devices. It uses a Reconfigurable Hardware Co-Processor to delegate critical tasks. The coprocessor can reconfigure packet-by-packet, handling up to 3 data streams of different protocols concurrently. This architecture is capable of replacing up to three MAC processors in a wireless device. Its heterogeneous and coarse-grained functional units, requirements of limited connectivity between these units, and the idle time of hardware resources promise very modest power consumption, suitable for mobile devices [20].

In the DRMP, the functionality of wireless MACs has been partitioned to a microprocessing unit (MPU) and a Reconfigurable Hardware Co-Processor (RHCP) (Figure 7). The MPU implements management and high level control functions of the MAC. The remaining functionality primarily includes time-critical operations associated with packet

transmission and reception. To optimize power-efficiency, the RHCP has coarse-grained, heterogeneous, function specific Reconfigurable Functional Units (RFUs). The Reconfigurable Hardware Co-Processor: The RHCP interacts with the MPU through an Interface and Reconfiguration Controller (IRC) which delegates tasks to flexible RFUs. The RFUs carry out the tasks requested by MPU, and have a uniform interface. They are dynamically and individually reconfigurable.

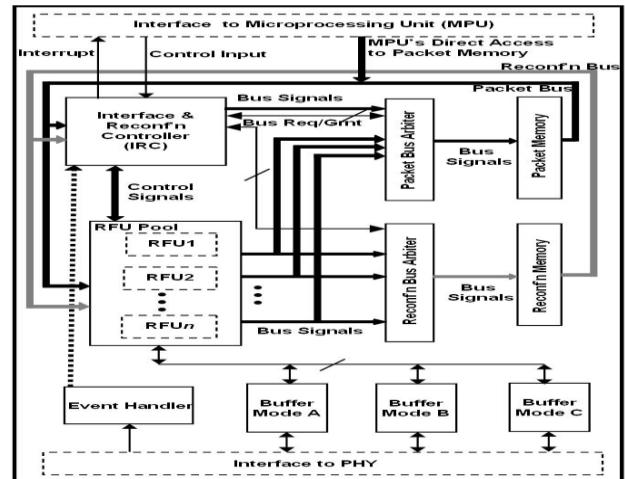


Figure 7

3.9. PACT XPP-III

XPP-III is heterogeneous reconfigurable processor architecture [21]. Two types of processing cores are embedded into a framework of stream-based components such as DMA controllers, stream communication crossbars, memory arbiters, buffer memories and I/O. The coarse-grained reconfigurable XPP Array provides high parallel processing performance for typical stream based applications such as video processing and software defined radio. The Array is composed by ALU-Processing Array Elements (PAE) that perform 16-bit arithmetic, RAM-PAEs that represent storage elements, and bottom-line (BL-) PAEs used to close the routing paths at the Array's bottom. The other processor type is the Function PAE (FNC-PAE) which is a general purpose VLIW-like processor with eight parallel, non-pipelined 16-bit ALUs and embedded instruction caches tightly coupled memories. XPP-III is a strictly modular and hierarchical design. The four different PAEs which form the array are arranged in a rectangular grid. All other components and the FNC-PAEs are linked with identical and simple 16-bit point to point data streams and 64-bit wide pipelined memory channels. All links provide handshake mechanisms for self synchronization of communication. The modular and hierarchical approach simplifies customization and verification of a processor for specific SoC designs but also provides a structural guideline.

3.10. SYSCORE Architecture

It is a novel CGRA architecture, target for biosignal processing in wearable and implanted devices. Low power consumptions is an important criteria for biosignal processing device. Hence more importance is given to low power consumption rather than performance while designing this architecture. The architecture provides significant energy savings by: eliminating the fetch-decode steps of traditional processors (via reconfiguration); significantly reducing the number of intermediate data RAM accesses (via systolic data

reuse), reduced logic switching (via compact functional units); and by voltage scaling (via parallelism)[22]. The architecture is shown in Figure 8 there are two main elements: Configurable Function Units (CFUs) and Round About Interconnect (RAI) units. Two Direct Memory Access (DMA) units inject data into the architecture from the West and North and one DMA units collects data from the data reuse in the SYSCORE architecture. Array configuration and DMA operations are controlled by the host processor.

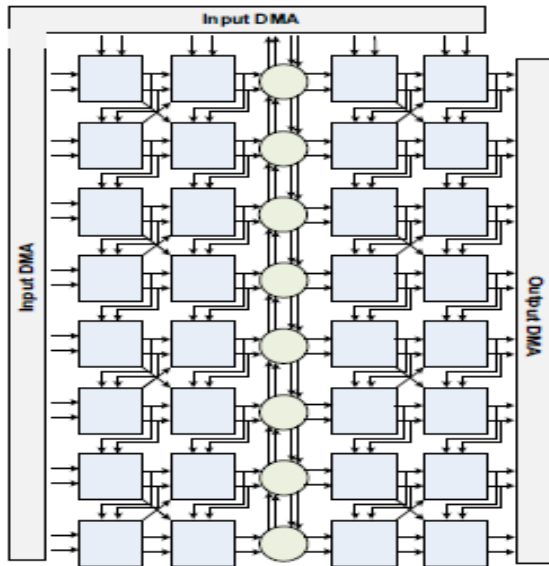


Figure 8

The SYSCORE can operate in three different modes: configuration mode, execution mode and flush mode. The SYSCORE architecture gives 62% average energy savings compared to a conventional DSP and SIMD processor and average speed ups of 30x and 8x compared to conventional DSP and SIMD processors respectively.

3.11. SmartCell

A novel coarse-grained reconfigurable architecture SmartCell [23] [24] designed which is targeted for high data throughput and computationally intensive applications. By integrating a large number of computational units with reconfigurable interconnection fabrics, SmartCell is able to provide stream processing capacity to achieve both performance and power efficiency.

A block diagram of the SmartCell architecture is shown in Figure 9. This microsystem architecture is composed of three major components: the cell unit, the reconfigurable interconnect fabrics and the high-speed data I/O. In a typical SmartCell architecture, a set of cell units is organized in a tiled structure. Each cell block consists of four processor elements along with the control and data memories. The reconfigurable connection fabrics are designed for the inner and inter cell data communications.

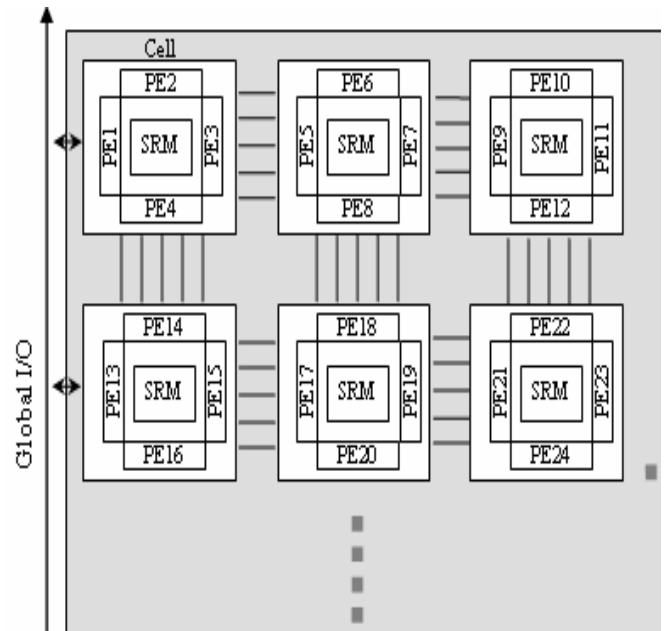


Figure 9

The data flow can be dynamically reconfigured for different applications. The number of PEs involved in the application tasks and the function of each PE can be changed in real time. A 16-tap finite impulse response (FIR) filter has been prototyped on both SmartCell architecture and FPGA chip[26]. The synthesis results in Table 2 indicate that the SmartCell system is about 10.9 times more power efficient than the FPGA

Table 2

	Dynamic power	Core Power
Smart cell	20.3 mW	21.7 mW
FPGA	157.6 mW	232.6mW

3.12. Floating-point Reconfigurable Array

It is a coarse-grained reconfigurable architecture which performs integer operations as well as floating point operations [26]. FloRA can perform a floating-point operation by combining two integer processing elements and controlling them with additional small control logics [27]. It also supports speculative execution to handle control intensive kernels. As a result, FloRA increases hardware utilization and further enhances the flexibility. FloRA contains two dimensional array of processing elements (PE's). These PE's are paired when floating-point operations are performed. It also contains RISC processor, DMA controller and the reconfigurable computing module (RCM) as shown in Figure 10. The RISC processor performs control-intensive code pieces. It also controls other components in this platform. DMA controller transports data from the external memory to local memories of RCM and vice versa. The RCM is used to execute data-intensive kernel code segments.

If Kernels of JPEG decoder is mapped on FLORA Architecture it gives 116.8 times higher performance than ARM9 Processors. This is because, flexible and parallel structure of FloRA is suitable to support various floating-point vector operations while maintaining the performance. With huge increase in demand of various kinds of high performance applications in mobile devices; designing SoC with FLORA architecture can offer better answer to designers.

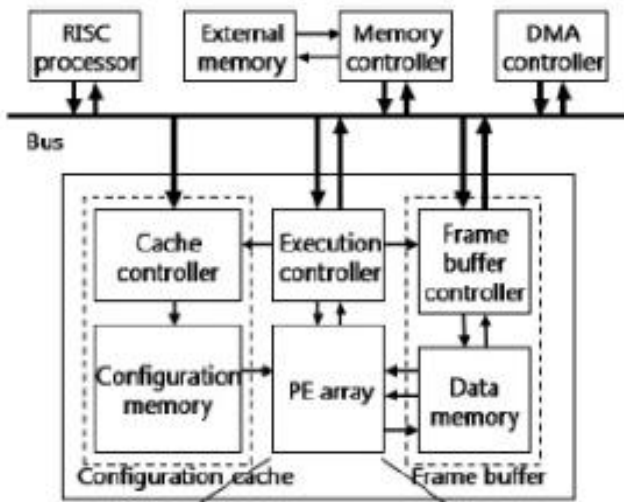


Figure 10

4. CONCLUSION

This paper has given a brief survey on today's coarse-grained reconfigurable architectures. These architectures rising as a new computing platform which have performance like ASIC and flexibility like FPGA. The architectures of different CGRA have been studied in these papers which are used in SOCs and designed for computationally intensive applications. These architectures are found to be very flexible and power efficient for implementing in SOC design.

5. REFERENCES

- [1] R. Hartenstein, H. Grünbacher (Editors): The Roadmap to Reconfigurable computing - Proc.FPL2000, Aug. 27-30, 2000; LNCS, Springer-Verlag 2000
- [2] A dynamically reconfigurable system-on-a-chip architecture for future mobile digital signal processing (1999) by Ahmad Alsolaim, Jürgen Becker European signal processing conference
- [3] Reiner W. Hartenstein. Reconfigurable Computing: A New Business Model and its Impact on SoC Design. In Proceedings of DSD'2001. pp.103~111
- [4] Jürgen Becker, Thilo Pionteck, Christian Habermann, Manfred Glesner "Design and Implementation of a Coarse-Grained Dynamically Reconfigurable Hardware Architecture" Proceeding WVLSI '01 Proceedings of the IEEE Computer Society Workshop on VLSI 2001
- [5] Allan Carroll, Stephen Friedman, Brian Van Essen, Aaron Wood, Benjamin Ylvisaker, Carl Ebeling, Scott Hauck, "Designing a Coarse-grained Reconfigurable Architecture for Power Efficiency, Department of Energy NA-22 University Information Technical Interchange Review Meeting, 2007
- [6] K. Compton, Architecture Generation of Customized Reconfigurable Hardware, Ph.D. Thesis, Northwestern University, Dept. of ECE, 2003.
- [7] Carl Ebeling, Darren C. Cronquist, Paul Franklin, Chris Fisher, "RaPiD – A Configurable Computing architecture for Compute Intensive Applications," University of Washington Department of Computer Science & Engineering Tech Report TR-96-11-03.
- [8] K. Eguro, RaPiD-AES: Developing an Encryption-Specific FPGA Architecture, M.S. Thesis, University of Washington, Dept. of EE, 2002.
- [9] H.Singh, M.Lee, G.Lu, F. Kurdahi, N. Bagherzadeh, "MorphoSys: A Reconfigurable Architecture for Multimedia Applications," sbcci, pp.134, XI Brazilian Symposium on Integrated Circuit Design, 1998
- [10] H. Singh, et al, "MorphoSys: An Integrated Reconfigurable System for Data-Parallel and Computing-Intensive Applications", IEEE Transaction on Computers, Vol. 49, No. 5, May 2000
- [11] A. Marshall et al., "A Reconfigurable Arithmetic Array for Multimedia Applications," Proc. ACM/SIGDA FPGA'99, Monterey, Feb. 21-23, 1999
- [12] J. Becker, M. Glesner, A. Alsolaim, J. Starzyk: Fast Communication Mechanisms in Coarse-Grained Dynamically Reconfigurable Array Architectures, Proc. of Second Int'l. Workshop on Engineering of Reconfigurable Hardware/software Objects (ENREGLE'00, in conjunction with PDPTA 2000), June 23-24, 2000, Las Vegas, USA
- [13] J. Becker, T. Pionteck, M. Glesner: DReAM: A Dynamically Reconfigurable Architecture for Future Mobile Communication Applications, 10th International Conference on Field Programmable Logic and Applications, Villach, Österreich, 2000.
- [14] Jong-eun Lee, Kiyong Cho, Nikil D. Dutt "Evaluating Memory Architectures for Media Applications on Coarse-Grained Reconfigurable Architectures." Proceedings of the Application-Specific Systems, Architectures, and Processors (ASAP'03) 2003 IEEE
- [15] B. Mei, S. Vernalde, D. Verkest, and R. Lauwereins, "Design methodology for tightly coupled VLIW/reconfigurable matrix architecture: a case study," Proc. of Design Automation and Test Conference in Europe, March 2004.
- [16] B. Mei, "A coarse-grained reconfigurable architecture template and its compilation techniques," Ph.D. thesis. Katholieke Universiteit Leuven, Jan. 2005.
- [17] Bingfeng Mei, Serge Vernalde, Diederik Verkest, Hugo De Man and Rudy Lauwereins: ADRES: An architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix, IMEC, 2003, Kapeldreef 75, B-3001, Leuven, Belgium, DATE 2004
- [18] Marco Lanuzza, Stefania Perri, Pasquale Corsonello, Martin Margala, "A New Reconfigurable Coarse-Grain Architecture For Multimedia Applications," Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007), pp 119-126, 2007
- [19] Custom Implementation Of The Coarse-Grained Reconfigurable Adres Architecture For Multimedia Purposes Francisco-Javier Veredas *, Michael Scheppeler Field Programmable Logic and Applications, 2005. International Conference on 24-26 Aug. 2005
- [20] Syed Waqar Nabi and Cade C. Wells "A Coarse-Grained Dynamically Reconfigurable Mac Processor For Power-Sensitive Multi-Standard Devices" in Field Programmable Logic and Applications 2008 FPL (2008)

- [21] F. Campii, R.K6nig3, M.Dreschmann3, M.Neukirchner4, D.Picard5, M.Juttner6,E.Schiiler7, A.Deledda2, D.Rossi2, A.Pasini, M.Hubner3, J.Becker3, R.Guerried”
RTL-to-Layout Implementation of an Embedded Coarse Grained Architecture for Dynamically Reconfigurable Computing in Systems-on-Chip” System-on-Chip, 2009. SOC 2009. International Symposium on 5-7 Oct. 2009
- [22] SYSCORE: A Coarse Grained Reconfigurable Array Architecture for Low Energy Biosignal Processing Kunjan Patel, S’eamas McGettrick and Chris J. Bleakley The 19th Annual IEEE International Symposium on Field-Programmable Custom Computing Machines 2011
- [23] C. Liang and X. Huang “SmartCell : A power-efficient reconfigurable architecture for data streaming applications,” In Proceedings of IEEE Workshop on Signal Processing Systems (SiPS’08), pp. 257–262, 2008.
- [24] “SmartCell: An Energy Efficient Coarse-Grained Reconfigurable Architecture for Stream-Based Applications” EURASIP Journal on Embedded Systems Volume 2009, Article ID 518659
- [25] Xinming Huang, “SmartCell Architecture, Design and Performance Analysis for Reconfigurable Embedded Computing,” In Proceedings of High Performance Embedded Computing Workshop (HPEC), Lexington MA, September 2008
- [26] Dongwook Lee, Manhwee Jo, Kyuseung Han, and Kiyong Choi, "FloRA: coarse-grained reconfigurable Architecture with floating-point operation capability", 2009 International Conference on Field-Programmable Technology, pp.376-379, Dec. 2009
- [27] Manhwee Jo, V.K.Prasad Arava, Hoonmo Yang, Kiyong Choi, "Implementation of floating-point operations for 3D graphics on a coarse-grained reconfigurable architecture", IEEE International SOC Conference, pp.127-130, Sep. 2007