

A Survey of Reconfigurable Architectures

Mahendra Pratap Singh
Research Scholar
Department of Computer
Science
Mohanlal Sukhadia University
Udaipur, India

Manoj Kumar Jain
Associate Professor
Department of Computer
Science
Mohanlal Sukhadia University
Udaipur, India

ABSTRACT

A new architecture type that is recently evolving is the reconfigurable architecture which combines the benefits of ASIPs (Application Specific Instruction Set Processors) and FPGAs (Field Programmable Gate Arrays). Reconfigurable computing combines software flexibility with high performance hardware. FPGAs are generally employed to construct a reconfigurable block as it provides an instant time-to-market advantage. Reconfigurable devices like FPGA offers improved computational efficiency as compared to traditional processor architectures. Reconfigurable block in these architectures provides the required flexibility for a large variety of embedded applications. Design space exploration of reconfigurable block involves a wide range of alternatives like logic block granularity in FPGA, interconnect topology, etc. The goal of this paper is to explore the reconfigurable architectures.

Keywords

Reconfigurable Architectures, FPGA, FGRA, CGRA.

1. INTRODUCTION

Reconfigurable architectures are those with a base processor and a reconfigurable block providing the required adaptability in processor micro-architecture for embedded applications [1]. Reconfigurable architectures improve upon flexibility and performance by bridging the gap between processors and Application-Specific Integrated Circuits (ASICs) [2].

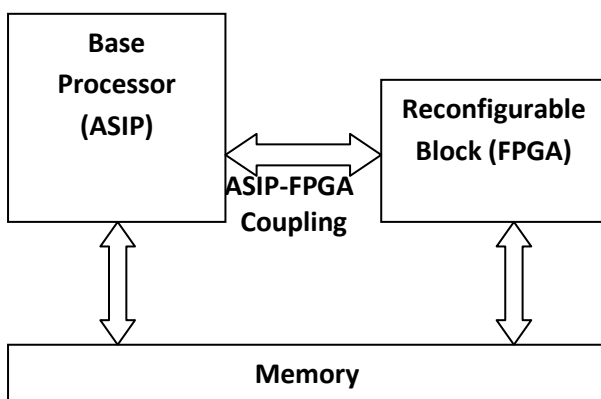


Fig 1: rASIP Architecture

To understand the basics of reconfigurable architectures, let us consider the architecture of rASIP (Partially Reconfigurable Application Specific Instruction Set Processors). As can be seen from Figure 1, a rASIP architecture consists of three components namely, the base

processor, the ASIP-FPGA coupling and the FPGA architecture [3]. These three components can be designed using numerous design alternatives. Base processor may belong to RISC or VLIW architectures. Issues involved in base processor design are number of pipeline stages, pipeline control, width of the instruction, etc. the interface of base processor and reconfigurable block can be tightly coupled or loosely coupled. In a tightly coupled interface, reconfigurable block can access processor internal registers, pipeline registers and is guided by several control signals coming from the base processor. In a loosely coupled interface, reconfigurable block interacts with external processor ports only.

Programmable Logic Devices (PLDs) were initially used as reconfigurable devices, but Field-Programmable Gate Arrays (FPGAs) are preferred as reconfigurable devices because it provides support for large SRAMs, and application-specific combinatorial blocks compared to PLDs.

2. RELATED WORK

Reconfigurable block can be either fine-grained or coarse-grained. Fine-grained reconfigurability can be achieved by Look-Up Tables (LUTs) and its implementation is done with RTL (Register Transfer Level). Coarse-grained reconfigurability can be achieved by multiplexing various functional units. Fine-grained architectures are less efficient because of a huge routing area overhead and poor routability. Coarse-grained architectures are preferred because of path widths greater than 1 bit.

Due to increased complexity, building blocks inside an FPGA typically contain more than one LUT, more than one flip-flop, and a mix of arithmetic, combinatorial, and multiplexing logic. [1].

Reiner Hartenstein et al [2] surveyed different architectures for coarse grain reconfigurable hardware namely primarily mesh-based architectures, architectures based on linear arrays, crossbar-based architectures, etc.

A generic rASIP for private key cryptographic applications has been proposed [3] in which fast SRAM based scratch pad was added to reconfigurable block to speed up the application execution.

A number of applications utilizing reconfigurable hardware and some example systems used in these applications have been discussed [4]. Reconfigurable hardware design issues which are critical to embedded system designers are also covered in the literature.

A. Chattopadhyay et al [5] proposed two major phases of rASIP design flow. According to them, the design space exploration of rASIPs can be divided into two phases i.e. pre-fabrication and post-fabrication. During pre-fabrication phase,

applications are analyzed using either static or dynamic profiling. Profiler helps in taking decisions regarding memory hierarchy, number of registers, processor architecture, etc. During post-fabrication phase, architectural design decisions concerning re-configurable block are taken [5].

B. Mei et al [15] proposed a novel architecture called ADRES (Architecture for Dynamically Reconfigurable Embedded System), with tight coupling of a VLIW (Very Long Instruction Word) processor and a coarse-grained reconfigurable matrix.

3. FPGA AS A RECONFIGURABLE ARCHITECTURE

FPGAs are the best known reconfigurable architectures in which bit-level configuration blocks are used. A generic FPGA architecture (as shown in Figure 2) consists of many programmable logic blocks implementing digital logic functions. Programmable routing switches connects the input and output pins of each logic blocks. FPGA logic blocks are based on transistors, gates, multiplexers, look up tables, etc. [14].

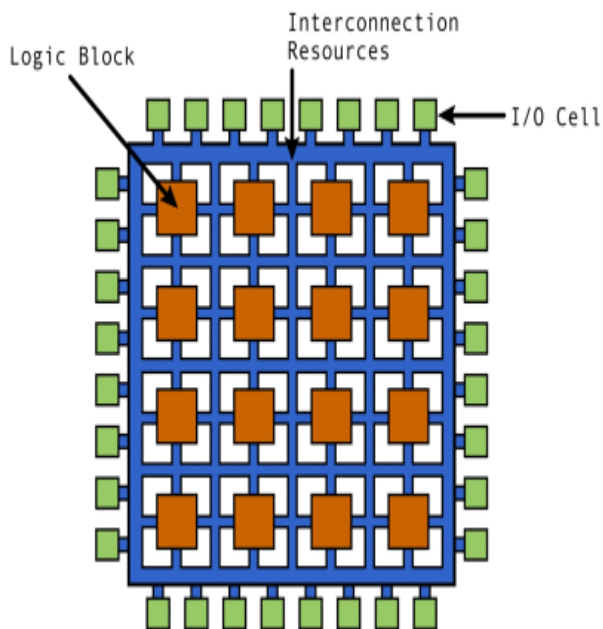


Fig 2: FPGA Logic Block Architecture

The use of large LUT (Look up Tables) in a reconfigurable block improves performance but it becomes more expensive due to increased logic block area. So, it is very crucial to choose an appropriate LUT size balancing both the performance and logic density [6].

FPGA supports implementation level reconfiguration and it has been implemented offering different product families by different vendors like Xilinx [13].

Programming technologies used for designing reconfigurable block using FPGAs are static memory, flash, anti fuse, etc [11].

3.1 SRAM-Based FPGAs

SRAM programming technology utilizes static RAM cells to control multiplexers. SRAM-based FPGAs are used to program routing interconnect of FPGAs and Configurable Logic Blocks (CLB). Major drawback here is that it requires external devices to store configuration data due to the volatile nature of SRAMs.

3.2 Flash-Based FPGAs

Flash or EEPROM based FPGAs are an alternative to SRAM based FPGAs offering several advantages including the non-volatile nature of EEPROMs, improved area efficiency as compared to SRAMs, etc. Also, as in case of SRAMs, there is no requirement for external permanent memory for programming the chip on power-up. Disadvantage of Flash-based FPGAs is that this programming technology utilizes non-standard CMOS process [11].

3.3 Anti-Fuse FPGAs

Anti-fuse programming technology in FPGA offers advantages of requiring low area, low resistance, non-volatile nature, etc. A major drawback is that anti-fuse programming technology based devices cannot be reprogrammed.

4. FINE-GRAINED RECONFIGURABLE ARCHITECTURES

FGRAs consist of a large number of small logic blocks based on LUTs (Look up Tables). FGRAs offer flexibility in embedded systems with power efficiency. Due to bit level configuration, these architectures offer high performance but at the cost of high complexity [11].

FGRAs have the disadvantages of large routing area, large volume of configuration data, low area efficiency for arithmetic operations, reduced clock speed and bandwidth, etc.

Atmel AT40K Architecture: Atmel AT40K is a fine-grained architecture consisting of a symmetrical array of identical cells. This architecture (figure 3), has distributed SRAMs. There is a direct horizontal, vertical, and diagonal cell-to-cell connection provided by 8-sided core cells. Here, due to small cells, we have a large number of cells providing greater functionality. Each cell in this architecture implements Boolean operations to provide the required functionality.

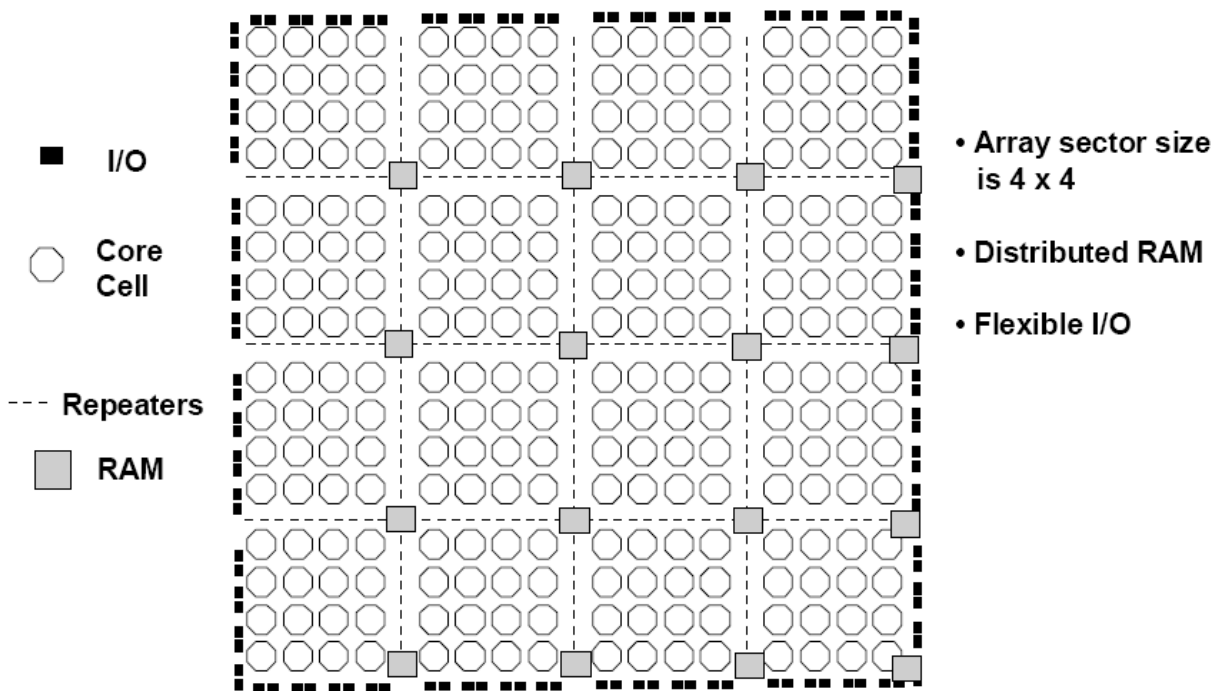


Fig 3: AT40K Device Overview

5. COARSE-GRAINED RECONFIGURABLE ARCHITECTURES

CGRAs are composed of a number of function units (FUs) organized as a mesh network. In CGA, register files hold temporary values and these values are only accessible by a subset of FUs. The FUs can execute common operations like addition, subtraction, and multiplication. When compared to FPGAs, CGRAs have the advantages of short reconfiguration times, low delay characteristics, and low power consumption.

Coarse grained reconfigurable architectures try to overcome the disadvantages of FPGA-based computing solutions by providing multiple-bit wide datapaths and complex operators instead of bit-level configurability. In contrast to FPGAs, the wide datapath allows the efficient implementation of complex operators in silicon. Thus, the routing overhead generated by having to compose complex operators from bit-level processing units is avoided.

CGRAs are less flexible than FGRAs, but they are easier to program and reconfiguration is faster.

Based on the arrangement of processing elements, CGRAs can be classified as Mesh-based architectures, architectures based on linear arrays, Crossbar-based architectures, etc.

5.1 Mesh-Based Architectures

In mesh-based architectures, processing elements are arranged in a rectangular array having vertical and horizontal connections. This architecture provides parallelism and efficient use of communication resources. This architecture is less complex as compared to FPGA due to less number of processing elements.

DP-FPGA (Data path FPGA) [16] implements regularly structured data paths. DP-FPGA has bit-sliced ALUs and a routing architecture similar to FGRAs.

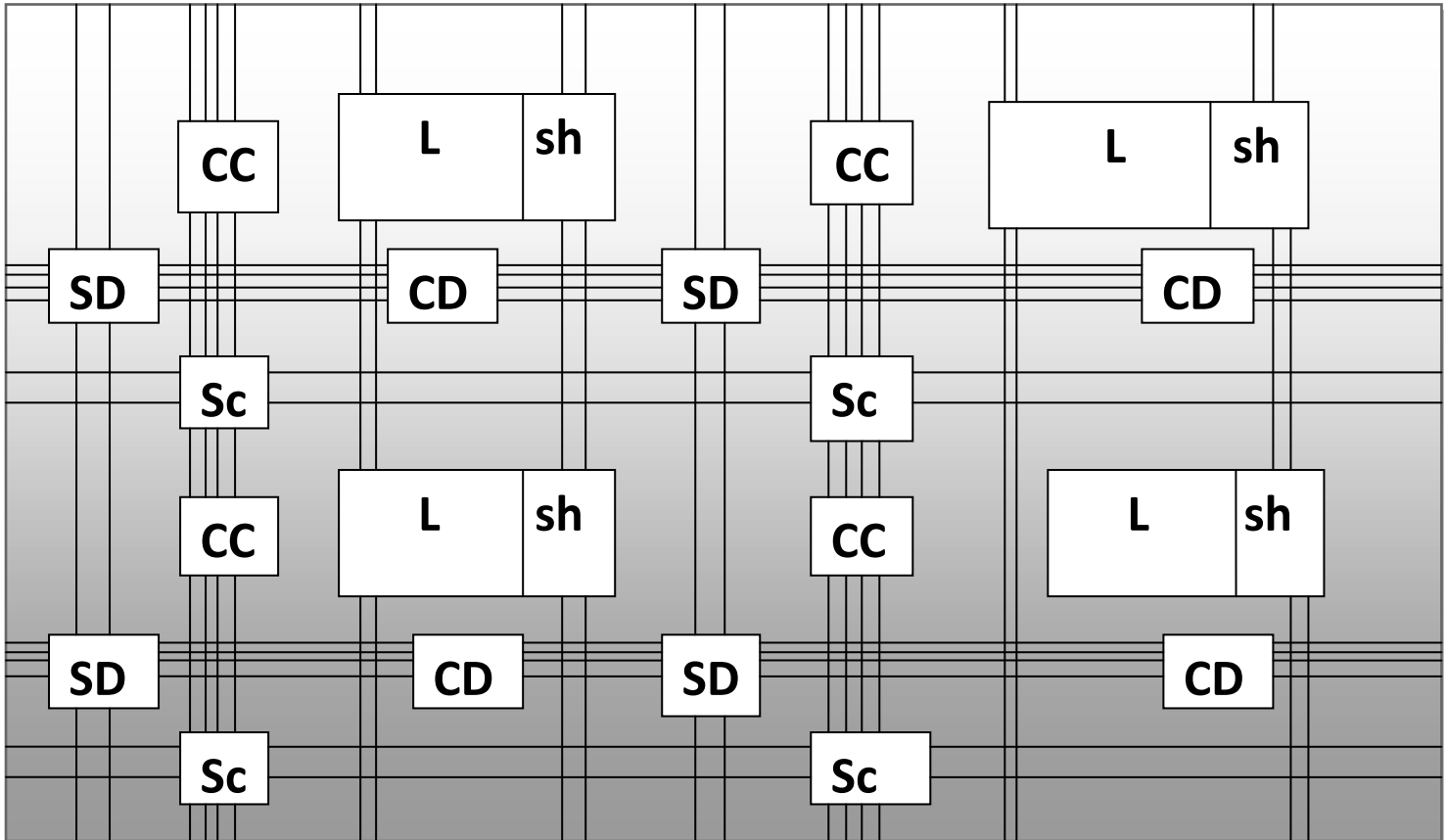
DP-FPGA architecture (figure 4) includes 3 components namely control logic, data path, and memory. Data path block consists of 4-bit slices. It consists of routing resources for data (horizontal) and control signals (vertical) using four bit buses. Shift block is another resource which supports single-bit or multi-bit shifts.

KressArray [17] is an architecture with very wide data paths, which compels to reduce the communication resources to achieve a feasible chip design. KressArrays are dynamically partially reconfigurable with an additional control unit.

5.2 Architectures based on Linear Arrays

In this architecture, processing elements are arranged as linear arrays providing connections between neighbors which provides direct mapping of pipelines.

RaPiD (Reconfigurable Pipelined Data path) [18] was proposed to speed-up highly regular, computation-intensive tasks by deep pipelines. Several parallel segmented 16 bit buses constitute routing and configuration architecture. This architecture is based on the idea of providing a number of different computing resources like ALUs, RAMs, Multipliers, and registers. PipeRench [19] is a dynamically reconfigurable architecture which allows configuration of processing elements to change in each execution cycle. This architecture also provides a global bus for data transfer, apart from mostly unidirectional nearest neighbor connects.



- L: Logic Block
- Sh: Shift Block
- CC: Control Connection Block
- CD: Data Connection Block
- SC: Control Switch Block
- SD: Data Switch Block

5.3 Crossbar-Based Architectures

This architecture provides arbitrary connections between processing elements, which makes routing much easier. A major drawback is associated with the high implementation cost of a full crossbar.

PADDI-1 (Programmable Arithmetic Device for DSP) architecture [20] was proposed for fast prototyping of DSP data paths, comprising of eight processing elements connected by a multilayer crossbar.

PADDI-2 is a successor to PADDI architecture consisting of 48 processing elements. To connect these processing elements, a hierarchical interconnect structure is selected which includes linear arrays of processing elements forming clusters and restricted crossbar to interconnect these clusters.

5.4 Xilinx Virtex Architecture

CLB (Configurable Logic Block) is the basic unit of Virtex FPGA and they are interconnected with programmable input output blocks (IOB). In this architecture, LUTs (Look up Tables) can be configured as SRAM cells (As shown in figure 5).

Configuration Logic Block (CLB) is the basic cell of Virtex FPGA. CLB contains circuitry (figure 5), that allows it to efficiently perform arithmetic operations. In this architecture, programmable input output blocks (IOBs) are interconnected to CLBs.

6. CURRENT STATUS

A summary of technical details of some of the recent coarse grained reconfigurable architectures has been presented [21] in Table 1.

DReAM (Dynamically Reconfigurable Architecture for Mobile Systems) is a 16-bit dynamic, array based architecture which was designed to be used for mobile devices. ADRES (Architecture for Dynamically Reconfigurable Embedded System) is a flexible VLIW (Very Long Instruction Word) based processor architecture with 32-bit data width. MORA (Multimedia Oriented Reconfigurable Array) is a linear array based 8-bit dynamic architecture, consisting of processing elements (PEs) and a control unit including the configuration memory. DRMP (Dynamically Reconfigurable MAC Processor) is a 32-bit dynamically reconfigurable coarse grained processor architecture designed to be used for wireless communication domain. SYSCORE is statically reconfigurable 32-bit array based processor architecture designed for biomedical monitoring applications.

Extensive research has been done on FPGAs and fine-grained reconfigurable architectures and a number of commercial tools have been proposed over the years.

However, as far as coarse grained reconfigurable architectures are concerned, there have been consistent efforts on developing tools but not much has been explored about developing compilation techniques for these types of architectures.

A generic architecture template, the dynamically reconfigurable ALU array (DRAA) is proposed which utilizes efficient compilation techniques to map applications on coarse grained reconfigurable architectures [23]. There is a need to develop mapping techniques for more general architecture domains.

Flexible reconfigurable architectures and tools for these architectures optimized for a particular application domain have to be developed.

Adaptability is also an issue which needs to be addressed while developing reconfigurable architectures for a set of application domain.

There have been many efforts to employ reconfigurable architectures in image processing and other applications like multimedia, where fine-grain parallelism is used for low-level operations and coarse-grain parallelism is used for high-level operations.

Table 1: Summary of technical details of example CGRAs

Project	Publication Year	Architecture	Reconfiguration Model	Granularity
DReAM	2001	Array	Dynamic	8 or 16 bit

DRAA	2003	Linear Array	Dynamic	8 bit
ADRES	2005	Array	Dynamic	32 bit
MORA	2007	Linear Array	Dynamic	8 bit
DRMP	2008	1 D Array	Dynamic	32 bit
PACT-XPP-III	2009	Array	Dynamic	16 bit
FloRA	2009	2 D Array	Dynamic	24 bit
SmartCell	2010	Array	Dynamic	8 bit
SYSCOR E	2011	Array	Static	32 bit

7. CONCLUSION

In this paper, a survey of reconfigurable architectures has been presented. The focus of this paper is mainly to explore design alternatives for reconfigurable processors. Reconfigurable architectures have evolved from merely an FPGA architecture to fine-grained and then coarse-grained architectures. Flexibility is a major issue in coarse-grained reconfigurable architectures, which needs to be addressed for future reconfigurable processors. Programming for reconfigurable architectures is another challenging task which needs to be addressed. Role of reconfigurable technologies in future execution environments like nanotechnology has to be studied in detail by researchers.

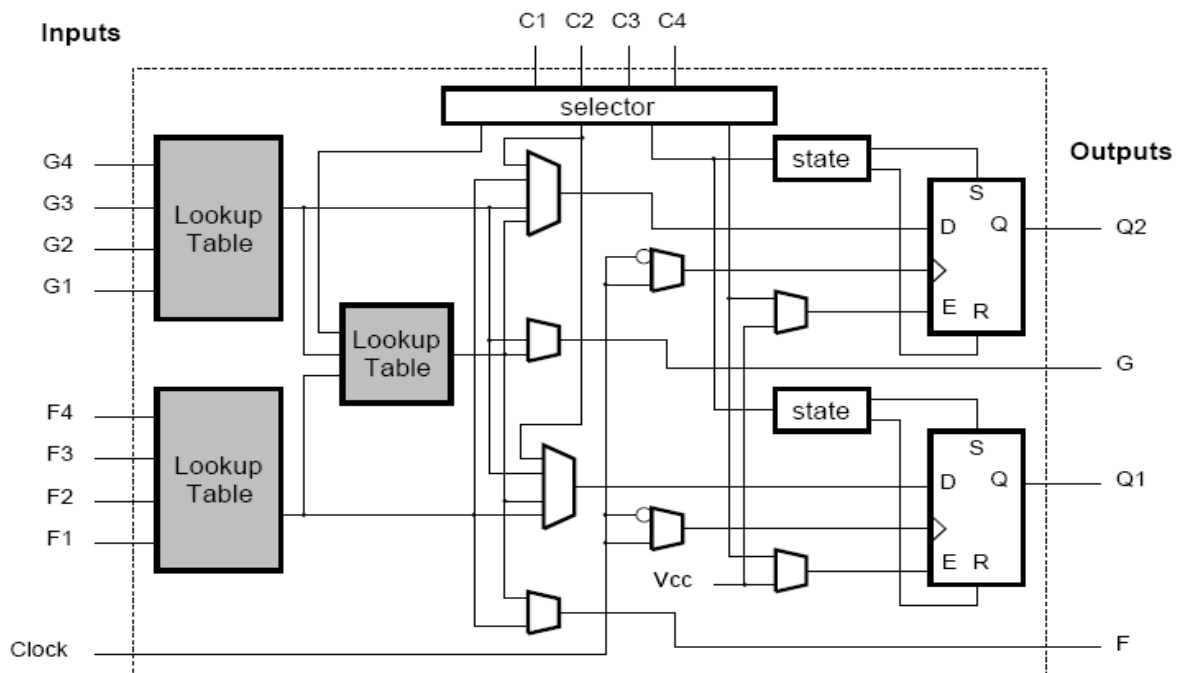


Fig 5: Xilinx Configuration Logic Block

8. REFERENCES

- [1] Anupam Chattopadhyay, “Ingredients of Adaptability: A Survey of Reconfigurable Processors”, Hindawi Publishing Corporation, VLSI Design, Volume 2013, Article ID 683615.
- [2] Reiner Hartenstein, “A Decade of Reconfigurable Computing: a Visionary Retrospective”, DATE '01 Proceedings of the conference on Design, automation and test in Europe, IEEE Press Piscataway, NJ, USA ©2001.
- [3] K. Karuri, A. Chattopadhyay, S. Kraemer, R. Leupers, G. Ascheid, H. Meyr, “A Tool Flow for Design Space Exploration of Partially Re-configurable Processors”, Integrated Signal Processing Systems, RWTH Aachen University 52056 Aachen, Germany.
- [4] Philip Garcia, Katherine Compton, Michael Schulte, Emily Blem, and Wenyin Fu, “An Overview of Reconfigurable Hardware in Embedded Systems”, Hindawi Publishing Corporation, EURASIP Journal on Embedded Systems, Volume 2006, Article ID 56320.
- [5] A. Chattopadhyay, W. Ahmed, K. Karuri, D. Kammler, R. Leupers, G. Ascheid, H. Meyr, “Design Space Exploration of Partially Re-configurable Embedded Processors”, Integrated Signal Processing Systems, RWTH Aachen University 52056 Aachen, Germany.
- [6] Elias Ahmed, “The effect of Logic Block Granularity on deep Submicron FPGA Performance and Density”.
- [7] Oliver Schliebusch, Gerd Ascheid, Andreas Wiefierink, Rainer Leupers, Heinrich Meyr, “Application Specific Processors for Flexible Receivers”.
- [8] Muhammad Rashid, Ludovic Aprville and Renaud Pacalet, “Application Specific Processors for Multimedia Applications”.
- [9] Ian Page, “Reconfigurable Processor Architectures”.
- [10] Damir Kirasic and Danko Basch, “Reconfigurable Processor Architectures: Varieties and Representations”.
- [11] U. Farooq, “Tree-Based Heterogeneous FPGA Architectures”.
- [12] Samuel Garcia and Bertrand Granado, “OLLAF: A Fine Grained Dynamically Reconfigurable Architecture for OS Support”, EURASIP Journal on Embedded Systems 2009.
- [13] Patrick Schaumont, Ingrid Verbauwhede, Kurt Keutzer, Majid Sarrafzadeh, “A Quick Safari through the Reconfiguration Jungle”.
- [14] Jonathan Rose, Abbas El Gamal, Alberto Sangiovanni-Vincentelli, “Architecture of Field-Programmable Gate Arrays.
- [15] B. Mei, “ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-grained Reconfigurable Matrix”, Springer LNCS.
- [16] D. Cherepacha and D. Lewis: “A Datapath Oriented Architecture for FPGAs”; Proc. FPGA'94, Monterey, CA, USA, February 1994.
- [17] R. Kress et al.: “A Datapath Synthesis System for the Reconfigurable Datapath Architecture”; ASP-DAC'95, Chiba, Japan, Aug. 29 - Sept. 1, 1995.
- [18] C. Ebeling et al.: “RaPiD: Reconfigurable Pipelined Datapath Architecture”.
- [19] S. C. Goldstein et al.: “PipeRench: A Coprocessor for Streaming Multimedia Acceleration”; Proc. ISCA'99, Atlanta, May 2-4, 1999.
- [20] D. Chen and J. Rabaey: PADDI: Programmable arithmetic devices for digital signal processing; VLSI Signal Processing IV, IEEE Press 1990.
- [21] Vaishali Tehre, Ravindra Kshirsagar, “Survey on Coarse Grained Reconfigurable Architectures”, International Journal of Computer Applications (0975 – 888), Volume 48– No.16, June 2012.
- [22] Jong-eun Lee and Kiyoun Choi, Nikil D. Dutt, “Compilation Approach for Coarse-Grained Reconfigurable Architectures”, 0740-7475/03/\$17.00 © 2003 IEEE.