International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 20 (2017) pp. 9225-9230

Survey on High Speed Computing System Architectures

Swati Sanjay Patil
[1]*Jayawantrao Sawant Polytechnic, Hadapsar, Pune, Maharashtra, India.*
Orcid Id: 0000-0002-5435-2322

Arun Kumar G
GM Institute of Technology, Davangere, Karnataka, India.

M H Nerkar
Government College of Engineering, Jalgaon, Maharashtra, India.

Abstract

Nowadays modern embedded systems are being demanding high speed computing processors for complex real life applications. There are many researchers had developed architectures for computing systems to develop high speed embedded systems that execute multifarious applications. This article presents the survey of various high speed computing system architectures presented in literature to develop embedded systems for real life applications. Finally, this article summarizes the computing systems architectures and proposes optimal computing system architecture for multimedia applications.

Keywords: General Purpose Processors, Field Programmable Gate Array, Application Specific Instruction Processors, Reconfigurable Architectures.

INTRODUCTION

Embedded computer systems have many hardware resources like arithmetic and logic units, storage space, input/output devices, etc., which are required to solve complex problems. The computer system architectures used in data processing applications are divided into four categories: General Purpose Processors (GPP), Application Specific Integrated Circuits (ASIC), Application Specific Instruction Processors (ASIP) such as DSP, Math Co-processor and Reconfigurable Architectures (RA). General purpose processors are flexible, but inefficient and offer relatively poor performance, whereas ASICs and ASIP are efficient and give better performance, but they are inflexible [1] [2]. Reconfigurable architectures make a tradeoff between these two extremes thereby computationally intensive applications are enhanced through hardware reuse. Figure 1 shows a "Semiconductor Pendulum" which is intended to provide interpretations of semiconductor technologies for embedded systems design. Imagine a long pendulum swinging back and forth between standardization and customization. There are various forces acting on and reacting to the pendulum, as shown in the figure 1. When the pendulum swings too far toward standardization due to new device, new architecture and software innovation, there would be reacting forces as need for differentiation, need for value addition and market confusion due to supply-demand imbalance. On the other hand, when the pendulum swings too far toward customization due to Data Automation (DA), Computer Aided Manufacturing (CAM) and Computer Aided Transcription (CAT), there will be reacting forces demand for better operational efficiency, improved cost effectiveness and faster time to market. Today's rising trend of embedded systems depends on innovation in device structure as well as in architecture toward the customized programmable technologies.

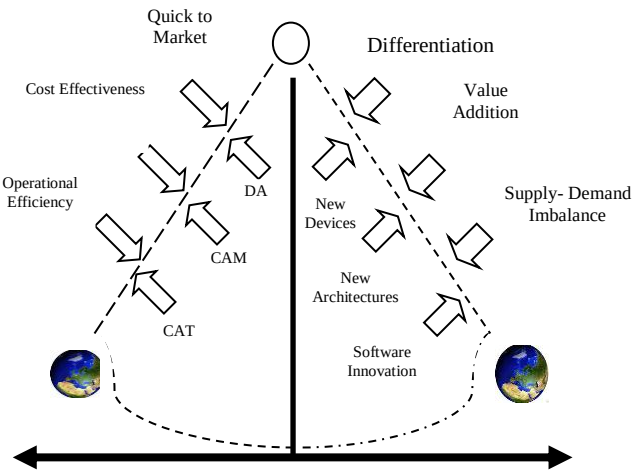

Figure 1: Semiconductor pendulum [2]

A new digital wave propelled [1] [2] by the digital consumer products and network in Figure 2 shows three big waves which drove electronics industry since 1970. Since the digital consumer products require high degree of flexibility, performance and personalization, they require high speed embedded system architectures and technologies to make vital contributions for the rising second digital wave. So, the high speed embedded systems for reactive real-time applications are implemented as mixed software-hardware systems. The

software system (General Purpose Processor) brings flexibility, while hardware system (Reconfigurable Architecture) enhances the performance. The emerging process technology and multi core architectural enhancements would play major role in optimization of real time applications and they require efficient resource management methodologies. So, design and optimization of real-time multitasking systems have received significant attention from both academia and industry in recent years.

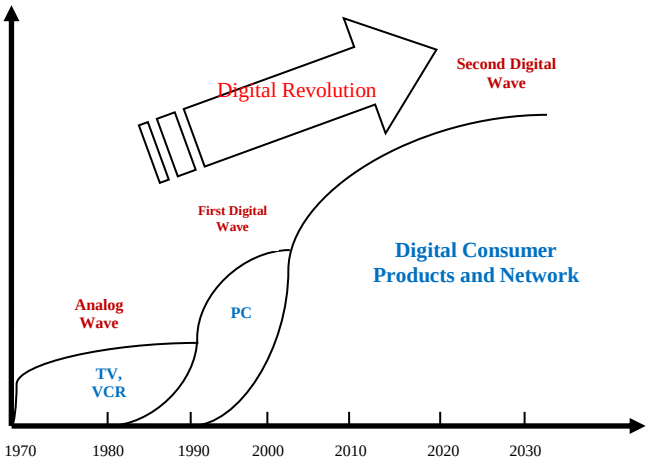

Figure 2: Second Digital Wave [2]

LITERATURE SURVEY

The literature of various computing platform architectures for computation of multimedia applications is demonstrated below.

Single Core Computing Architecture

Since the invention of microprocessors, the industry has successfully continued to innovate and increase the performance of embedded systems. The architecture of a processor refers to instruction set, registers, and data structures, (i.e. ALU, memories, bus arbiters which are not visible to the programmer) to the programmer and these are advanced from one generation to the next. In the single core embedded system design [1], a commercial microprocessor is integrated with several selected supported chips on a Printed Circuit Board (PCB) as shown in figure 3.The single core architecture brings flexibility to embedded systems at the expense of performance due to communication delay between the processor core and the supported chips made available on a PCB. The single core architecture support sequential execution and could serve only one application at a time but it may fail in utilization of the supported chips effectively. Thus, researchers have been motivated towards multi core architectures.

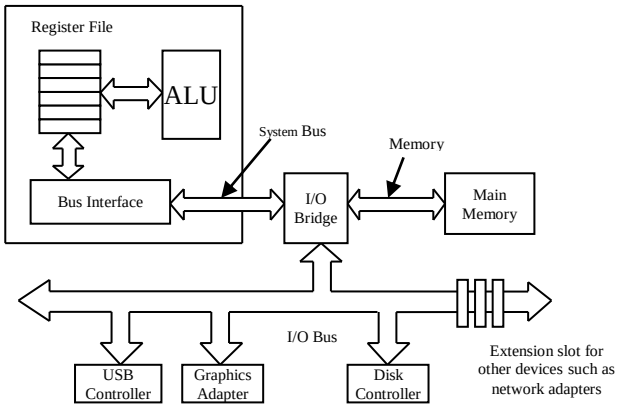

Figure 3 Single core computing architecture

Multi Core Computing Architecture

Multi core computing architecture is shown in figure 4, consists of two or more processor cores working together on a single chip integrated circuit to obtain high performance for embedded system. These multi core architectures support faster execution of applications by the technique of parallelism that brings the ability to work with multiple problems simultaneously. Parallelism is important where complex problems are divided into smaller tasks and they could execute simultaneously on multi core architecture [1]. The multi core architecture can make use of Instruction Level Parallelism (ILP) and Thread Level Parallelism (TLP). ILP is a measure of the number of instructions that can be performed during a single clock cycle whereas TLP measure the set of parallel instructions that can be performed in a single clock cycle. Parallel instructions are the instructions which do not depend on each other for execution. Software drivers and glue logic has been generated to connect processors to peripheral devices and hardware co-processors through communication interfaces. The multi core computing architectures are described in two architectural forms as homogeneous and heterogeneous computing system.

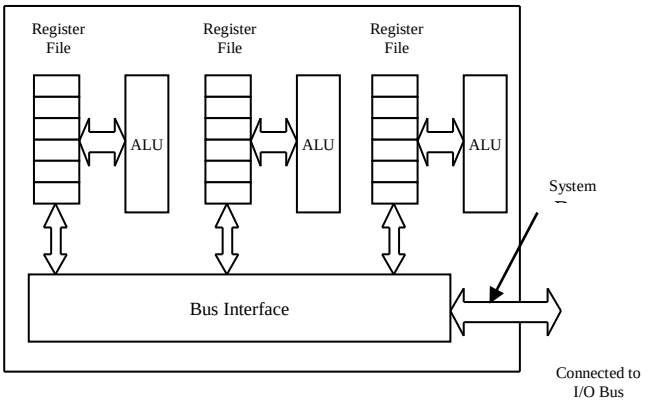

Figure 4: Multi core computing architecture

Homogeneous Computing System: A computing platform in which group of similar resources i.e. Processing Elements (PE) are integrated through high speed network to support execution of parallel applications [3] [4] called Homogeneous Computing System.

Heterogeneous Computing System: Diverse set of resources which acts as PEs are interconnected with a high speed dissimilar network called Heterogeneous Computing System and which supports execution of computationally intensive parallel and distributed applications [4,5].There are architectures, which integrate Central Processing Unit - Graphical Processing Unit (CPU-GPU) with fully addressable shared memory [6], designed for execution of distributed applications and also they eliminate CPU-GPU data transfer overhead.

Reconfigurable Computing Architecture

The increasing requirements of more flexibility and higher performance inembedded systems design demand a new computing paradigm [7] [8] reconfigurable architecture such as Field Programmable Gate Array (FPGA). The FPGA promises greater flexibility without compromising performance and flexibility. The FPGA device contains both memory and logic elements along with an intellectual property (IP) processor core to support rapid development of computing system or custom hardware for high speed embedded systems [6] [7]. The complex applications like computation intensive kernels [9], MIMO, OFDM and image processing can be accelerated by using reconfigurable architecture as computing systems. The FPGA devices presents higher performance while reducing instruction fetch, decode and execute bottleneck [1] [7]. The FPGA devices also bring the phenomenon of configuring custom digital circuits dynamically and modify via software [1] [7]. This ability of FPGA devices create and modify digital logic circuits without physically altering the hardware and thus these devices provides more flexible and low cost solution for real time applications. So, High Speed Computing Systems (HSCS) should have one or more resources of such kind as processing elements to enhance the speed of real time application. This phenomenon of dynamic reconfiguration of an application is enabled by the availability of high density programmable logic chips FPGA. The FPGA architecture [8] [9] is available in two forms.

Fine-grained FPGA Architecture: FPGA device consists of a matrix of programmable logic cells with a grid of interconnecting lines running between them. In addition, there are I/O pins on the perimeter to provide an interface between the FPGA, interconnecting lines and chip's external pins. However FPGA devices tend to be somewhat fine-grained in order to achieve high degree of flexibility. This flexibility requires for the situations where the computational requirements are unknown in advance or vary considerably among the needed applications. Extreme level of flexibility is unnecessary in many situations and it would result in significant area, delay and power overheads [9]. Today, further improvements are being made by embedding coarse-grained elements such as memories, multipliers, and processors within the fine-grained programmable fabric of the FPGA architecture.

Coarse-grained FPGA Architecture: The data-path width of coarse grained elements of reconfigurable architectures is more than one bit. Over the last 15 years, many projects have been investigated [9] [10] [11] and successfully built systems based on Coarse-grained reconfiguration performed within a processor or amongst processors. The Coarse-gained reconfiguration procedure is much faster than that found in Fine-grained FPGAs. The Coarse- grained elements can implement a specific function more efficiently than Fine-grained programmable logic [9] [11]. However, since Coarse-gained FPGA architecture is not as flexible, they only benefit applications which utilize them. In this research, Coarse-grained elements required for targeted computing architecture is configured on Fine-grained FPGA architecture [11] to support fast execution of multimedia applications.

Heterogeneous Reconfigurable Computing Architecture

The homogeneous computing systems having similar PEs provide parallel processing to the applications at the expense of number of similar resources, whereas heterogeneous computing systems have dissimilar PEs to support applications but they require dissimilar communication protocols to interconnect heterogeneous resources. The reconfigurable system having FPGA as PE [2] [6] [7] [9] [11] bring phenomenon of dynamic reconfiguration to the applications at the overhead off-chip softcore PE interface. Since growing demand for the high quality multimedia applications especially audio, video and graphic processing [4] [8], a computing platform called Heterogeneous Reconfigurable Computing System (HRCS) has been developed. The HRCS integrates the array of programmable logic resources i.e. hardcore PE and soft core processor i.e. softcore PE together as demonstrated in literature [10] [11] by several researchers. The HRCS could be good candidate architecture to speed up multimedia applications. The Molen polymorphic processor [12] equipped with a “Core Processor,” which is general-purpose processor (GPP), and “Reconfigurable Processor” (RP) shown in figure 5 is an example for HRCS.

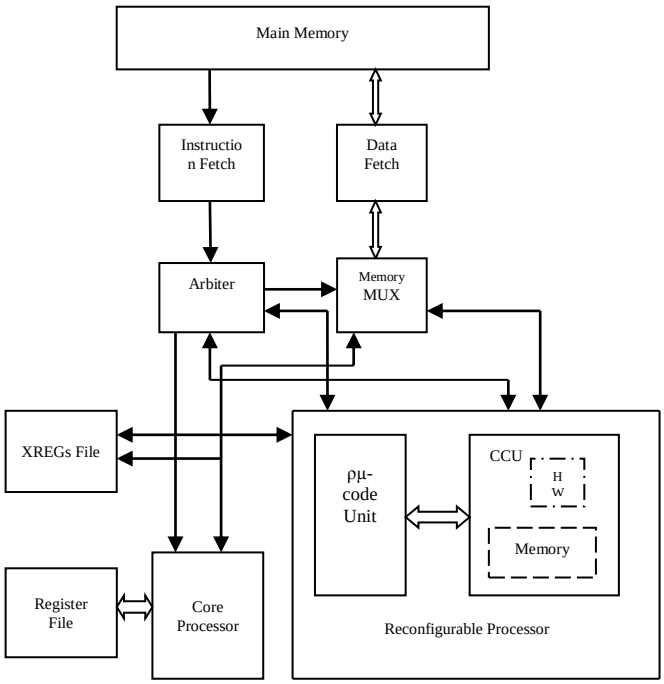

Figure 5: MOLEN Polymorphic processor architecture [12]

On-Chip Reconfigurable High Speed Computing System

As semiconductor technology advances, the integration of softcore PEs and hardcore PEs on a single chip [13] [14] support execution of complex functions like high resolution, high quality graphs/videos, high bandwidth communications, high speed computation etc. Typically the on-chip reconfigurable heterogeneous computing system contain reconfigurable array of hardcore PEs and a soft core processor i.e. softcore PE on a single chip FPGA device. This kind of on-chip heterogeneous architectures promise the flexibility of traditional general-purpose processors while providing the efficiency and high performance of application-specific integrated circuits (ASICs) and ASIPs.

The researchers in [16] [17] designed Reconfigurable High Speed Computing System (RHSCS) on FPGA device and scheduled few bench mark applications onto RHSCS to enhance the execution time of the applications. The researchers proved that the on-chip heterogeneous computing systems can bring optimum performance to the real time applications.

On Chip Heterogeneous Computing System Architecture

In this article, an on-chip heterogeneous computing system is presented with software and hardware cores on a single chip to execute tasks of multimedia applications. The on-chip heterogeneous computing has the potential to achieve greater energy efficiency by combining traditional processors with Reconfigurable cores (R-Cores) such as hardware custom logic, reconfigurable customized memory and communication core or with Reconfigurable Processor cores (RP-Cores) such as Power PC, MicroBlaze customized processors cores. Although R-cores are effective in increasing performance, they can also benefit the computing system in terms of energy efficiency. So, this research is aimed to design an on-chip heterogeneous computing system on Field Programmable Gate Array (FPGA) for consumer multimedia applications.

In this research, the proposed architecture is shown in figure 3 and has been realized on a single chip Virtex-5 FPGA device (XC5VLX110T) using Xilinx Embedded Development Kit (EDK). In the HRCS, a MicroBlaze soft core processor is configured in part of the reconfigurable area of FPGA and the rest reconfigurable area is used for configure multiple RLUs, memory and communication protocols. Wherein HRCS, the MicroBlaze is a 32-bit RISC architecture equipped with instruction and data cache memory of size 4KB each, for storing instructions as well as data while task execution. The RLU reconfigures its custom hardware for hardware tasks and also it support hardware task interface with off chip peripherals.

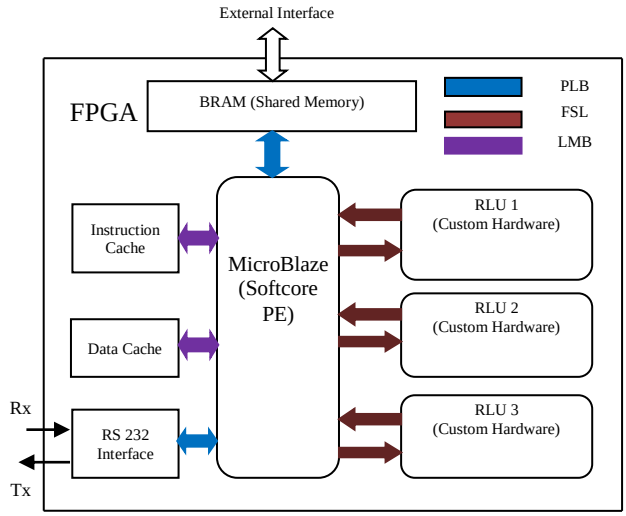

Figure 7: Heterogeneous Reconfigurable Computing System

The on chip 64KB BRAM acts as shared memory for MicroBlaze and RLUs to store executable files, input and output data. BRAM memory controller, configured along with BRAM, helps in loading the task executable files, input and output data from external environment and also control inter process communication between BRAM and MicroBlaze. The data interchange between BRAM and custom hardware of RLU can happen through Fast Simplex Link (FSL) communication protocol. These on-chip functional blocks, MicroBlaze, RLUs, BRAM, instruction and data cache memory, are interconnected through 32-bit communication protocols like Processor Local Bus (PLB), Local Memory Bus (LMB) and FSL. The PLB provide interface between Soft core

processor i.e. MicroBlaze and BRAM through BRAM controller that load instructions, input data and store back output data after computation. The communication protocol LMB supports interfacing of cache memories with MicroBlaze in order to minimize memory access overheads. The FSL used to interface custom hardware configured in RLU with MicroBlaze and it has 32-bit FIFO implemented on BRAM to support data streaming between MicroBlaze and custom hardware.

EXPERIMENTAL RESULTS AND DISCUSSIONS

This section presents the FPGA resources utilization for construction of On-Chip HRCS. In the targeted HRCS, MicroBlaze is equipped with instruction cache and data cache memories for storing program as well as data during executing the tasks of an application. The BRAM memory acts as shared memory for MicroBlaze and RLUs for storing input and output data. These functional blocks, MicroBlaze, RLUs (custom hardware for tasks of application), BRAM, Cache memory and general purpose I/O devices, have been interconnected through on-chip communication protocols like PLB, LMB and FSL as shown in figure 3. The architecture of HRCS has been configured on Virtex-5 FPGA device and the FPGA resources utilization for realization of computing platform HRCS is summarized in table 1.

Table 1*:* Virtex-5 FPGA Device utilization for HRCS

Resources	**Module**	**Number of bit slices**
Soft core (MicroBlaze)	MicroBlaze	1599
Hard core (Reconfigurable Logic Unit)	RLU1	500
	RLU2	500
	RLU3	500
Memory	DDR2_SDRAM (256MB)	1687
	dlmb_cntlr (8KB)	7
	Ilmb_cntlr (8KB)	4
Communication Interfaces (Bus controllers)	Dlmb	1
	Ilmb	1
	Mb_plb	96
Debug Module	Mdm (64KB)	97
Timing and reset circuits	proc_sys_reset	30
	xps_timer (64KB)	187
I/O interfaces	RS232_uart (64KB)	97
	DIP_Switches_8bit (64KB)	67
	LEDs_8bit (64KB)	71
Heterogeneous Reconfigurable Computing System		5,444

CONCLUSION AND FUTURE SCOPE

The homogeneous computing systems having similar PEs provide parallel processing to the applications at the expense of number of similar resources whereas heterogeneous computing systems have dissimilar PEs to support applications with dissimilar communication protocols between heterogeneous resources. The reconfigurable system having FPGA as PE, bring phenomenon of dynamic reconfiguration to the applications at the overhead of off-chip softcore PE interface. A hybrid computing platform called Reconfigurable High Speed Computing System, having integrated softcore PE and hardcore PE configured on a single chip FPGA, minimizes communication overheads and also has flexibility to support execution of software tasks as well as hardware tasks. The On-chip HCS provides optimal intermediate computing trade-off for execution of software tasks and hardware tasks in terms of performance and area. The proposed architecture can be optimized and equipped with cores required for multimedia applications.

REFERENCES

[1] Reiner Hartensstein, “Microprocessor is no more General Purpose: Why future reconfigurable platforms will win,” invited paper of the International conference on Innovative Systems in silicon.ISIS’97, Texas, USA, October 8-10, 1997, pp 1- 10.

[2] Makimoto, T., "The hot decade of field programmable technologies," Proceedings of IEEE International Conference on Field-Programmable Technology, 16-18 Dec. 2002, pp.3 – 6.

[3] Jinho Lee, Moo-Kyoung Chung, Yeon-Gon Cho, Soojung Ryu, Jung Ho Ahn, and Kiyoung Choi, "Mapping and Scheduling of Tasks and Communications on Many-Core SoC Under Local Memory Constraint," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on , vol.32, no.11, pp.1748-1761, Nov. 2013.

[4] Taho Dorta, Jaime Jim´enez, Jos´eLuis Mart´ın, Unai Bidarte, and Armando Astarloa, “Reconfigurable Multiprocessor Systems: A Review,”International Journal of Reconfigurable Computing, Volume 2010, Article ID 570279, 10 pages, 2010.

[5] ZeljkoZilic, Prabhat Mishra, and Sandeep K. Shukla, “Guest Editors’ Introduction: Special Section on System-Level Design and Validation of Heterogeneous Chip Multiprocessors”, IEEE Transactions On Computers, Vol. 62, No. 2, Pp. 209 - 210 February 2013.

[6] Nanda kumar V.S., Marek-Sadowska M., "On Optimal Kernel Size for Integrated CPU-GPUs — A Case Study," Computer Architecture Letters, vol.13, no.2, pp.81-84, July-Dec, 2014.

[7] J Lyke, “Reconfigurable Systems: A generalization of Reconfigurable computational strategies for Space

Systems," IEEE Aerospace conference Proceedings, vol. 4, pp 4-1935, 2002.

[8] David B. Stewart, Pradeep K. Khosla, "Real time Scheduling of Dynamically Reconfigurable Systems," Proceedings of the IEEE International Conference on Systems Engineering, Dayton Ohio, August 1991, pp. 139-142.

[9] Ganghee Lee, Kiyoung Choi and Nikil D. Dutt, "Mapping Multi-Domain Applications onto Coarse-Grained Reconfigurable Architecture, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, Vol. 30, No. 5, May 2011.

[10] Juanjo Noguera and Rosa M. Badia, "HW/SW Co-design Techniques for Dynamically Reconfigurable Architectures," IEEE Transaction on Very Large Scale Integration (VLSI) Systems, Vol. 10, No. 4, pp 399 – 415, August 2002.

[11] Boris Kettelhoit and Mario Porrmann, "A layer Model for Systematically Designing Dynamically Reconfigurable Systems," International Conference on Field Programmable Logic and Applications, August 2006, pp. 1-6.

[12] S. Vassiliadis, S. Wong, G. N. Gaydadjiev, K.L.M Bertels, G. K. Kuzmanov and E. M. Panainte, "The Molen Polymorphic Processor," IEEE Transaction on Computers, Vol. 53, Issue 11, pp. 1363-1375, November 2004.

[13] Haluk Topcuoglu, Salim Hariri and Min-You Wu, "Performance Effective and Low-Complexity Task Scheduling for Heterogeneous Computing," IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 3, pp. 260-274, March 2002.

[14] Mohammad I. Daoud and Nawwaf Kharma, "A High Performance Algorithm for Static Task Scheduling in Heterogeneous Distributed Computing Systems," Journal of Parallel and Distributed Computing, Vol. 68, no. 4, pp. 299-309, April 2008.

[15] S. Darba and D.P. Agarwal, "Optimal Scheduling Algorithm for Distributed Memory Machines," IEEE Trans. Parallel and Distributed Systems, Vol. 9, no. 1, pp. 87-95, Jan. 1998.

[16] Mahendra Vucha and Arvind Rajawat, "Dynamic Task Distribution Model for On-Chip Reconfigurable High Speed Computing System," International Journal of Reconfigurable Computing, Volume 2015 (2015), Article ID 783237, 12 pages, December 2015.

[17] Mahendra Vucha and Arvind Rajawat, "A Novel Methodology for Task Distribution in Heterogeneous Reconfigurable Computing System," International Journal of Embedded Systems and Applications, Volume 5, Number 1, pp. 19-39, March 2015.

[18] Makimoto, T., "The hot decade of field programmable technologies," Proceedings of IEEE International Conference on Field-Programmable Technology, 16-18 Dec. 2002, pp.3-6.

[19] Sih G.C, Lee E.A, "A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures," Parallel and Distributed Systems, IEEE Transactions on , vol.4, no.2, pp.175-187, Feb 1999

[20] Hesham El-Rewini, T.G. Lewis, "Scheduling parallel program tasks onto arbitrary target machines", Journal of Parallel and Distributed Computing, Volume 9, Issue 2, Pages 138-153, June 1990.

[21] Gregory Dimitroulakos, Michalis D. Galanis, and Costas E. Goutis, "Exploiting the Distributed Foreground Memory in Coarse Grain Reconfigurable Arrays for Reducing the Memory Bottleneck in DSP Applications", SSIP'05 Proceedings of the 5th WSEAS international conference on Signal, Speech and image processing, Corpu, Greece, Aug. 17 – 19, 2005, pp. 132-137.

[22] M. Sangeetha, Raja Paul Perinbam and M. Kumaran, " A State Transformation based Partitioning Techniques using Dataflow Extraction for Software Binaries", IJCSNS International Journal of Computer Science and Network Security, Vol. 9, No. 2, pp. 264-273, February 2009.

[23] Sotirios Xydis, George Economakos, Dimitrios Soudris, and Kiamal Pekmestzi, "High Performance and Area Efficient Flexible DSP Data path Synthesis," IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 19, No. 3, March 2011.

[24] Chi Wai Yu, Julien Lamoureux, Steven J. E.Wilton, Philip H.W. Leong, and Wayne Luk, "The Coarse-Grained/Fine-Grained Logic Interface in FPGAs with Embedded Floating-Point Arithmetic Units," International Journal of Reconfigurable Computing, Volume 2008, Article ID 736203, 10 pages, 2008.