International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

On-Chip Hardware Accelerator for DSP Applications

Swati Sanjay Patil , Nagaraja B. G

Abstract: ***High speed computing systems developed for multimedia streaming application demand high throughput and which can be achieved by designing hardware accelerators for data processing. This article presents new hardware accelerating platform comprised of heterogeneous multi core processing elements integrated on single chip FPGA. This kind of multi core platform can boost multimedia applications through parallel processing. The proposed multi core platform has been realized on FPGA and few DSP applications are executed on the processing elements of the platform to validate its performance. The performance of the proposed hardware accelerator has been compared with existing standard computing platforms frequently used for multimedia applications. The comparison shows that the proposed on-chip multi core accelerator has enhanced the execution speed of DSP applications while providing optimum throughput.***

Keywords : Hardware Accelerator, Multimedia Computing Systems, FPGA, Custom Computing Cores, Reconfigurable Architectures, General Purpose Processors.

I. INTRODUCTION

Multimedia computing systems must have resources Arithmetic and Logic Unit (ALU), memory, input/output devices, etc., to compute complex multimedia applications. The computing systems used in multimedia applications have been categorized into General Purpose Processors (GPP), Application Specific Integrated Circuits (ASIC), Reconfigurable Architectures (RA) and Application Specific Instruction Processors (ASIP) such as DSP, Math Co-processor. 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. Today's rising trend of computing systems depend innovation in device structure as well as in architecture toward the customized programmable technologies.

Manuscript published on 30 September 2019
* Correspondence Author
Swati Sanjay Patil*, Jayawantrao Sawant Polytechnic, Hadapsar, Pune, Maharashtra, India Email: sspatil1555@gmail.com
Nagaraja B. G, Jain Institute of Technology, Davanagere, Karnataka, India, Email: nagarajbg@gmail.com

Since the digital multimedia computing architectures demand high degree of flexibility and performance, researchers has started designing high speed computing system architectures and hardware accelerators to make vital contributions for the rising computational demands.

So, high speed computing systems are developed as heterogeneous systems having both hardcore and softcore processing elements to execute multi facial reactive real-time applications. The general purpose software processor based systems brings flexibility whereas the system developed on reconfigurable devices enhances the performance of multimedia applications. This kind of emerging technology and heterogeneous system architectures could provide platform for optimization execution of multimedia applications. So, design of multi core heterogeneous multitasking systems have got significant demand in both industry and academia.

II. LITERATURE SURVEY

The invention of microprocessor made the industry to contribute toward innovation in developing efficient computing systems and increase the performance of multimedia systems. The computing system architectures consists of registers and data structures such as ALU, memories, bus arbiters which are not visible to the programmer and these are advanced from one generation to the next. The single core embedded systems [1] has several chips integrated on a Printed Circuit Board (PCB).These kind of single core embedded systems provide greater flexibility to real time applications at cost of communication delay between processor and interfacing chips made available on PCB. The single core computing systems could perform sequential execution and serve only one application at a time which leads to poor utilization of available resources. Thus, researchers were motivated to develop multi core computing architectures. The multi core computing architecture would have two or more processing cores working together to achieve higher computing performance. These multi core computing architectures enable parallelism that brings the ability to run multiple tasks simultaneously and speed up the application execution. Parallel processing utilize techniques like Instruction Level Parallelism (ILP) and Thread Level Parallelism (TLP) which could divide complex problems into smaller tasks and execute them simultaneously on multi core architecture [1]. Glue logic and software device drivers were generated to connect processing elements and hardware co-processors to peripheral devices through communication interfaces.

There are architectures, integrated 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. Increased flexibility and performance of computing systems 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 devices are capable of configuring custom logics, memory and softcore processors to support rapid prototyping or custom hardware for high speed computing applications [6] [7]. The computation intensive kernels [9] like MIMO, OFDM and multimedia processing, can be accelerated by developing computing systems on FPGA. The computing systems developed on FPGA devices bring higher performance to the application execution by phenomenon of configuring custom digital circuits dynamically and modify via software while minimizing instruction fetch, decode and execute bottleneck [1] [7]. The dynamic reconfiguration ability of FPGA devices can configure computing architectures without altering the hardware physically and thus FPGA brings flexibility and low cost computing solutions for complex multimedia applications. So, multimedia computing systems must have resources of custom computing elements to enhance the execution speed of application.

The computing system developed on FPGA [2] [6] [7] [9] [11] compute the applications at the off-chip softcore processor interface overhead. The growing demand for the high quality multimedia applications, especially audio, video and graphic processing [4] [8], enable the design of multi core computing platform by integrating the array of programmable logic resources i.e. hardcore and softcore processing elements together as demonstrated in literature [10] [11] by several researchers. The Molen polymorphic processor [12] equipped with a GPP and reconfigurable processing elements is an example for multi core 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 and high quality video processing, high bandwidth communications, high speed computation etc. Typically, on-chip multi core heterogeneous computing system can be designed by integrating reconfigurable array of hardcore PEs and soft core processor on a single chip FPGA device to promise the trade off between flexibility and performance. These on-chip multi core computing architectures [15] integrate microprocessors, block memory, reconfigurable logic blocks, and multiple IP cores and are now practical and commercially available. The researchers in [16] [17] designed multi core high speed computing system on FPGA device to enhance the execution speed of the applications. In this article, researchers attempted to prove that the on-chip multi core computing systems can bring optimum performance to the multimedia applications by configuring hardware accelerator architecture on FPGA device and run few selected real life applications.

III. ARCHITECTURE OF HARDWARE ACCELERATOR

The on-chip heterogeneous architectures provide energy efficient computing by combining traditional processors with Reconfigurable cores (R-Cores). The R-Cores are customized hardware computing systems having Reconfigurable Processor cores (RP-Cores) such as Power PC, MicroBlaze customized softcore processors integrated with reconfigurable memory and communication core. Although R-cores are effective in increasing performance, they can also benefit the computing system in terms of energy efficiency. So, this article is aimed to design an on-chip hardware multi-core hardware accelerator on Field Programmable Gate Array (FPGA) for consumer multimedia applications. The reconfigurable area of FPGA devices can be utilized for configuring R-cores and RP-cores to support computing and processing of multimedia techniques. The proposed on-chip computing system architecture is designed with single RP-core and multiple R-cores on a single chip FPGA to support both software and hardware tasks of multimedia application.

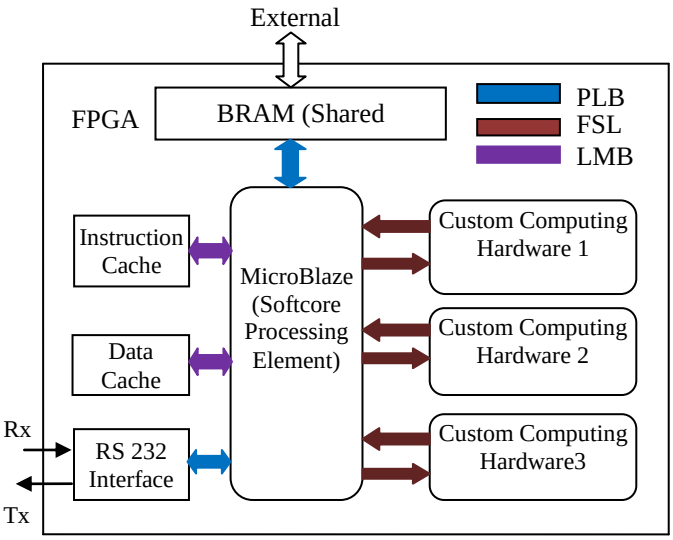

Fig. 1. On-chip multi core hardware accelerator

In this research, the proposed architecture is shown in figure 1 has been realized on a single chip FPGA device using Xilinx Embedded Development Kit (EDK). The On-chip Hardware Accelerator (OHA) has equipped with MicroBlaze softcore processor, multiple custom computing hardware and memory are integrated on single chip FPGA through communication protocol. The MicroBlaze in OHA is a 32-bit RISC processor equipped with instruction and data cache. The custom computing hardware configures its custom hardware for hardware tasks and also supports communication interface with external peripherals.

The Block RAM (BRAM) is shared memory supports both MicroBlaze and custom computing hardware to store executable files, input and output data. BRAM memory controller loads the task executable files and data from external devices and also controls data operations between BRAM and MicroBlaze. These on-chip resources MicroBlaze, cache, custom computing hardware,

BRAM are interconnected through communication protocols like Processor Local Bus (PLB), Local Memory Bus (LMB) and Fast Simplex Link (FSL). The PLB provide interface between MicroBlaze and BRAM through BRAM controller to load instructions and data. The LMB provides interface between cache memories and MicroBlaze to minimize memory access overheads. The data interchange between BRAM and custom hardware can happen through FSL protocol. The FSL is a 32-bit FIFO to control data streaming among MicroBlaze and custom hardware.

IV. PERFORMANCE ESTIMATION AND DISCUSSIONS

Computing systems can be made efficient for complex applications by integrating hardware and software processing elements and this kind of hybrid system may enables additional design challenges. The design challenges of the hybrid computing systems has been addressed by using Xilinx design tool chains and enabled the designers to construct custom hardware architectures on single chip FPGA. The Xilinx custom hardware development tool process, shown in fig 2, is followed to develop proposed hardware accelerators by integrating the multiple computing cores on FPGA.

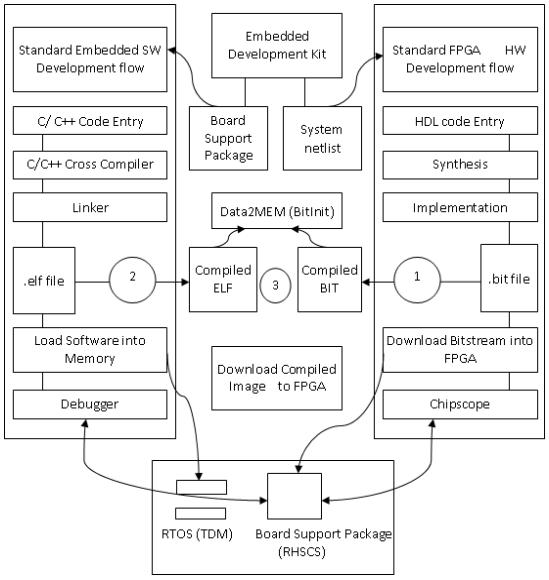

Fig 2. Xilinx custom hardware development tool flow

The standard FPGA hardware development flow enable the designer to develop multi core custom hardware on FPGA whereas standard Embedded software development flow help the designer to develop software which control the custom hardware of hardware accelerator configured on FPGA. The Bit stream Initializer (BitInit) creates the image file of the compiled .elf files and .bit files and then download the image file into memory of the FPGA. The MicroBlaze softcore processor is configured in part of the FPGA reconfigurable area and the rest reconfigurable area of FPGA has been utilized for custom computing hardware, memory and communication bus protocols to interface resources of hardware accelerator. In multi core hardware accelerator, the MicroBlaze executes application in traditional method like fetch, decode and execute whereas the custom computing hardware reconfigures its architecture to the functionality of tasks in an application. In the targeted on-chip multi core hardware accelerator, MicroBlaze is equipped with 4KB instruction cache and data cache to speed up the application execution. The BRAM of size 64KB is shared among MicroBlaze and custom computing hardware for storing data while application execution. These computing blocks of OHA has been interconnected through on-chip protocols like PLB, LMB and FSL.

The proposed OHA architecture has been implemented *ON* FPGA device using Xilinx EDK as demonstrated in figure 2 and synthesized using Xilinx Synthesis Tool (XST). The schematic of the proposed On-chip multi core hardware accelerator architecture is shown in fig 3.

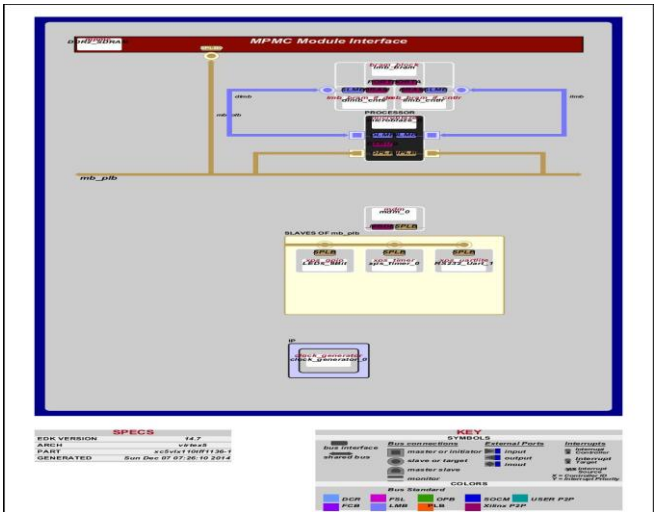

Fig 3. Schematic of on-chip multi core hardware accelerator Architecture on FPGA

The proposed on-chip multi core hardware accelerator architecture configured on FPGA device and the resources utilization for realization of hardware accelerator is summarized in table I.

Table I*:* Resource utilization of the on-chip hardware accelerator on FPGA Device

System Resources	Module	LUTs	FFs	BRAMs
MicroBlaze	Softcore processor	2725	2898	0
Custom hardware developed on FPGA	Custom computing hardware 1	300	300	2
	Custom computing hardware 2	300	300	2
	Custom computing hardware 2	300	300	2
Memory	DDR2_SDRAM	3342	2212	7
	System_lmb_bram	0	0	2
	dlmb_cntlr	2	6	0
	Ilmb_cntlr	2	6	0
Bus controller	Dlmb	1	0	0
	Ilmb	1	0	0
	Mb_plb	152	342	0
Debug Module	Mdm	126	123	0
Timing and reset circuit	proc_sys_reset	69	53	0
	xps_timer	358	287	0

I/O interface	RS232_uart (64KB)	148	131	0
	DIP_Switches_8bit (64KB)	125	64	0
	LEDs_8bit (64KB)	125	64	0
Hardware System Accelerator		8076	7087	15

Test bench has been created with the techniques like Convolution, Fast Fourier Transform, Inverse Fast Fourier Transform, Finite Impulse Response Filter and Orthogonal Frequency Division Multiplexing transmitter, to evaluate the performance of proposed hardware accelerator architecture. As stated in Xilinx custom hardware development tool flow, the test bench applications were described using C++ and Hardware Description Language (HDL) and then executed them on resources of the proposed architecture.The chosen test bench applications were also executed on the popular softcore processor used for running multimedia applications. The acquired execution time of the test bench applications on the chosen computing architectures are demonstrated in table II.

Table II: Hardware software design attributes of DSP techniques

DSP technique	FPGA	MicroBlaze [125MHz]	TMS320C6713 DSK [225MHz]
	Execution Time (ns)	execution Time (µs)	execution Time (µs)
Linear Convolution	6.5	298.5	7.8
Circular Convolution	8.7	145	6.3
16-FFT	4.4	28228	87.3
16-IFFT	5.4	26925	142.2
FIR Filter	19.7	3562	203.1
Binary generator	576	2191.7	1270
S/P conversion	1.2	192.4	8.1
16-QAM	6	128.5	0.9
P/S conversion	4.2	192.4	8.1
Cyclic Prefix	4.7	382.3	8

The table II demonstrates that the performance of the test bench applications has been enhanced on custom hardware configured on FPGA as compared to softcore processors MicroBlaze and TMS320C6713 DSK. Thus the execution of multimedia applications can be accelerated by developing multi core custom computing hardware and that could be efficient hardware platform to accelerate for multimedia applications at low power computations.

V. CONCLUSION

A high speed multi core computing platform having custom hardware and MicroBlaze as processing elements has been realized on a single chip FPGA to enhance the execution of multimedia applications. Test bench was developed by considering few popular DSP techniques to validate the realized multi core hardware platform. The techniques in test bench are executed on the computing resources of the platform, i.e. MicroBlaze softcore processor and custom computing hardware and then the execution time of the chosen DSP applications were tabulated. The execution time of the DSP applications was compared with execution time obtained by running the applications on contemporary computing architectures Microblaze and TMS320C6713 DSK. Finally, the tabulated results show that the On-chip custom computing hardware brought intermediate computing trade-off for execution of DSP application in terms of performance and throughput. Thus the custom computing hardware could be the low cost and low power hardware accelerator platform for complex 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