IJCSN International Journal of Computer Science and Network, Volume 5, Issue 6, December 2016
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 1.02

Implementation of Fast Fourier Transform Accelerator on Coarse Grain Reconfigurable Architecture

[1] Vaishali Tehre, [2] Dr. Pankaj Agrawal , [3] Dr. R.V. Kshrisagar

[1] Electronics Department, G .H. Raisoni college of Engineering Nagpur University
Nagpur, Maharashtra, India

[2] Department of Electronics, G.H. Raisoni Academy of Engineering and Technology Nagpur, India,
Nagpur, Maharashtra, India

[3] Priyadarshini Indira Gandhi College of Engineering
Nagpur, Maharashtra, India

Abstract - Recent technology growth permit engineer to design various complex applications on single-on-chip (SoC) related to communication, Image Processing, video processing, digital signal processing. In all of these complex algorithms, FFT blocks are one of the most computation concentrated. Here we first introduce a novel Coarse-Grain Reconfigurable Array (CGRA) which is used as a hardware accelerator to optimize the performance of system. The architecture consist of processing elements (PEs), configuration controller and interconnection network on a single chip. Subsequently we present a mapping of different length of Fast Fourier Transform (FFT) algorithms on them. In this paper, we have considered radix-(2, 4) FFT accelerators which are mapped on 4X4 PE CGRA templates. We estimated their power and energy consumption. A Field Programmable Gate Array (FPGA) is used to implement prototype of CGRA. Based on the measurements, we have compared results with other implementations.

Keywords - *Coarse Grain Reconfigurable Architecture (CGRA), Low Power, FPGA, ASIC, Processing Element.*

1. Introduction

Running multiple applications on single platform has become crucial requirement of today's embedded system. This constrains on portable embedded system makes difficult to manage energy resources. For example, the users want their device to work on different applications such as wireless communication, image processing, multimedia, real-time performance. At the same time they expect the long batteries life. To overcome all above problem coarse grain reconfigurable architecture (CGRA) is introduce. Coarse-Grain Reconfigurable Array (CGRA) is a powerful solution to industrial and academic research needs. Due to their array-based structure, they provide high throughput and parallelism. These features make them ideal for processing computationally intensive signal processing algorithms.

Examples of such general purpose CGRAs are BUTTER [1], Morphosys [2], ADRES [3] and PACT-XPP [4]. CGRA is flexible as compare to ASIC and consume low power as compare to FPGAs and have potential to bridge the performance and power gap between FPGA and ASIC. CGRAs consist of an array of a large number of function units (FUs) interconnected by a mesh style network. The FUs can execute common operations including addition, subtraction, and multiplication. Compare to FPGA, CGRA have short reconfiguration times, low delay characteristics, low power consumption, High performance and software design specification. CGRA is domain specific but functions can be change within domain. CGRA is area efficient hence in small space system placed with numbers of gates.

In recent years 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. 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

MATRIX (Multiple ALU architecture with reconfigurable interconnect experiment) is mesh style coarse grain architecture developed at MIT by Andre Dehon in 1996 and designed for general purpose application [3]. MATRIX provides parallel configurable dataflow, dynamic control and deployable resources. Network of MATRIX is interconnect in three level i.e. nearest neighbor interconnect, length four bypass interconnect and global lines. RaPid is static and limited dynamic system and develop for DSP application. Function units are linearly arranged in system, targeted for high performance by using deep pipelining [4]. The paper [5] describes a coarse grained architecture DRAA which is designed for multimedia application.

The architecture consists of regular ALU array data-path with a fast memory interface. MorphoSys [6] is also designed to handle multimedia application. The System is dynamic and programmable elements are arranged in mesh style. Since the routing architecture is design using crossbar to solve the routing overhead problem and give high performance with low power. ADREs[7] is flexible processor architecture template designed for embedded application in SoC with low cost targets in terms of area and power consumption. A VLIW processor is the main processing unit of ADRES which include array of tightly coupled configurable processing. cells for purpose of acceleration SmartCell [8] designed which is targeted for high data throughput and computationally intensive applications. It can be configured to operate in various modes, such as SIMD, MIMD, and systolic array. SmartCell system is an energy efficient architecture for stream processing. Some architectures are also developed for biomedical application such as SYSCORE [9]. This architecture consumes low power, and design for real time processing of biomedical signals. The architecture provides significant energy savings by eliminating the fetch-decode steps of traditional processors, significantly reducing the number of intermediate data RAM accesses, reduced logic switching, and by voltage scaling. 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.

The rest of the paper is organized as follow: section 2 briefly explains the Architecture, interconnection and configuration of newly designed coarse grain architecture.section3 describe the design and mapping of parallel FFT on the mention architecture. Section 4 discusses the synthesis results and performance evaluation of FFT mapped on prototype architecture followed by conclusion in section 5

2. Baseline CGRA

The overview of basic architecture is described in this section. This architecture is used for performing various computational intensive tasks. Proposed architecture is composed of identical 16 bit processing element in two dimensional 4x4 mesh style networks as shown in Figure1. The arrays of 16 Processing elements are reprogrammable and called reconfiguration unit. This reconfigurable unit controlled or configured by a unit called controller. The configuration context or instructions are stored in a memory called context memory. There is two layer interconnection network i.e. crossbar connection and inter-cell connection. Proposed architecture is flexible to mapped various computing style. Operation can be performed through any PE of architecture it shows that architecture is dynamic and various task can be performed by various PEs.

2.1 Processing Elements

Architecture consists of array of 16 block called as Processing element (PE). All PE is organized in tiled structure. The single PE composed of input output multiplexer, arithmetic and logic unit and local data register as shown in figure2. The ALU supports the standard set of arithmetic and logic functions including NAND, NOR, XOR**,** shift, and add. The ALU also includes a 16 bit multiply operation. Each PE contributes its 16-bit output(s) to a wide Data Bus of an on chip connection networks and is able to select one data bus entry for each of its inputs. The PE read their configurations from the Configuration Bus. Each PE is configured with an operation and input selections. For example, an ALU can be configured to perform addition, subtraction, logical AND, maximum and minimum, multiplication etc. There are direct connections from any PE to any other PE. This complete interconnect structure avoids expensive place and route algorithms hence delay timing is low and no need of extra synchronization require. This connection is static and dataflow is bi-directions. More compact interconnect may be developed in the future simultaneously with compiler

2.2 Configuration Controller

The global configuration memory contains the different configurations, i.e. VLIW programs and interconnect

schemes, A VLIW program and an interconnect scheme are grouped to build one configuration type for each PE. Each configuration field contains several control words which is used to control all operation performed by PE . All configuration fields are memory mapped from the Controller point of view. Thus, the Controller is able to change a single configuration field of a processing element by writing to the individual address. This can be called as partial reconfiguration. Configuration of processing elements results in a convention data path for a particular calculation.

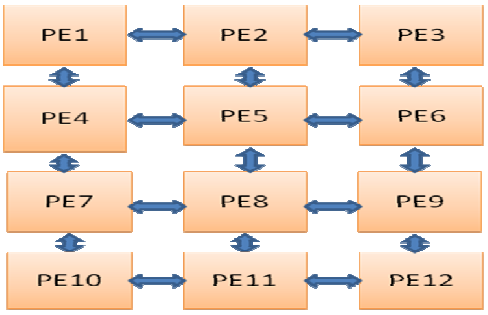

Fig. 1 baseline Reconfigurable PE array

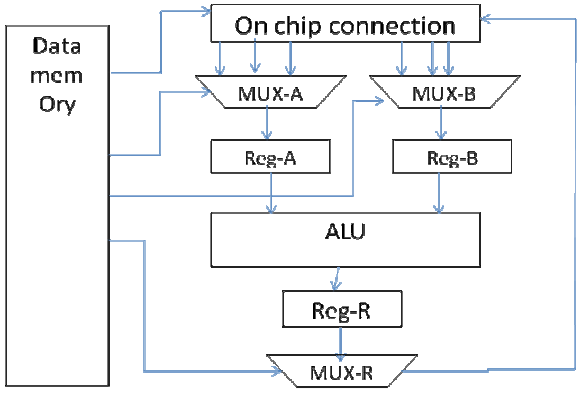

Fig. 2 Single Processing Element Architecture

3. Mapping FFT Algorithm

Fast Fourier Transform is an algorithm to compute Discrete Fourier Transform (DFT). DFT is used to convert a time domain signal into its frequency spectrum domain. FFT is an efficient algorithm to compute DFT. There are many distinct FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory. FFTs are algorithms for quick calculation of discrete Fourier transform of a data vector. The FFT is a DFT algorithm which reduces the number of computations needed for N points from O(N 2) to O(N log N) where log is the base-2 logarithm. The N point FFT is defined in Eq.(1)

$$f_i = \sum_{k=0}^{n-1} x_k e^{-(2\Pi i/n)jk} j = 0,\ldots.n-1. \quad (1)$$

Where f(i) and x(k) are complex input and output and e-(2πi/n)jk is twiddle factor. Radix2 FFT divide the calculation of DFT into log2(N) stages, each of which consists of a group of N/2 2-point DFTs, also called butterfly units (BUs).

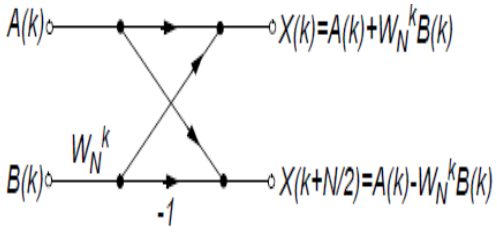

Fig. 3 Radix 2 butterfly unit

$$\begin{aligned} A_r &= x_r + (y_r \times w_r) - (y_i \times w_i) \\ B_r &= x_r - (y_r \times w_r) + (y_i \times w_i) \\ A_i &= x_i + (y_r \times w_i) + (y_i \times w_r) \\ B_i &= x_i - (y_r \times w_i) + (y_i \times w_r) \end{aligned} \quad (2)$$

The complex number butterfly calculation is the basic operation involved in Radix-2 FFT. Comparing with the Decimation-in-Frequency (DIF) FFT, the DIT FFT achieves a more balanced computing load between the two branches in the butterfly operation, which makes it more popular in parallel FFT designs. With some more computational stages the Radix-4 and Radix-8 FFTs are also calculated with this method. The 2 point Radix-2 FFT butterfly calculation is formulated in Eq.2 and shown in figure 3. The complex number operation can be optimized into four real multiplications and six real additions. In our design, eight PEs are used to calculate the single butterfly results in a sequential manner. Figure 4 shows the mapping of PE to implement single butterfly unit. All top PEs are performing multiplication followed by addition and subtraction.

Figure 5 shows one possible way of mapping two butterfly units on 4X4 matrix of PE arrays with data flow scheduling. The intermediate results are shared between all PEs through the inner cell connection unit. To perform

8 points FFT on propose architecture, 12 butterfly units and three stages of computation is required the test bench is written to store necessary configurations and input data into configuration memory, each of which is then process by controller.

The same PEs performing the same butterfly operation six times. The PE has two operand inputs and one outputs and can interact with the inputs and outputs of the neighboring PEs in local and global fashion.. A pattern of interconnections and operations to be performed by each PE in the CGRA defines a 'context'. The context switching is based on the configuration words stored in context memories, each associated to its corresponding PE. Once the configuration words are fetched from the main memory, they are distributed over the reconfigurable array with the help of a pipelined infrastructure. Each configuration word is composed of an address field and data field. The pipelined infrastructure distributes the configuration words based on their address field.

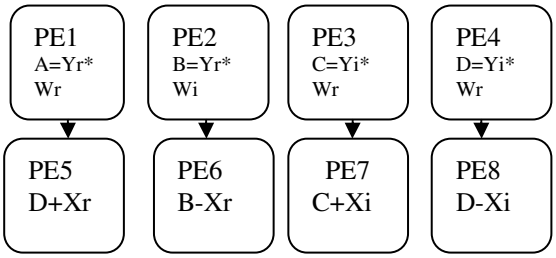

Fig. 4. Mapping of 2 point butterfly unit

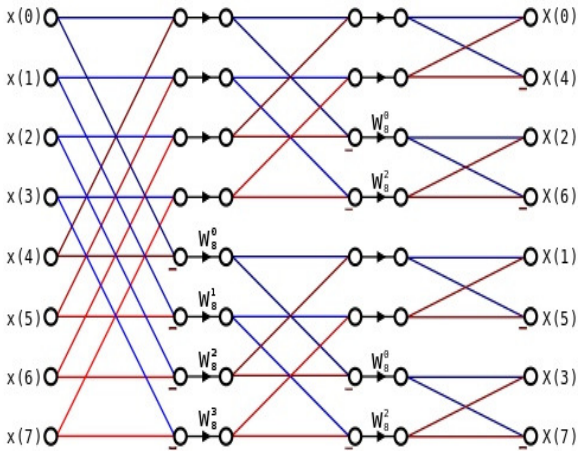

Fig. 5. 8 point butterfly unit

4. Synthesis Result and Power Performance

The prototype of proposes CGRA Architecture is implemented on the Xilinx Virtex-4 FPGA (xc3s100e-5vq100) device. It is synthesized using the Xilinx ISE 14.2software. The implementation summary is given in the Table1.

Table1: Implementation Summary

Logic utilization	**Used**	**Available**	**Utilization**
Number of slice registers	772	10,944	7%
Number used as a flip flops	260	--	--
Number of slice LUTs	3,624	10,944	33%
Number of DSP block	16	32	50%
Number of bonded IOBs	66	240	27%

The performance measures like power, operating frequency, and clock Cycle to evaluate the proposed design is given in Table 2.

Table2: Performance Comparison

Application		Static Power (mw)	Dynamic Power (mw)	Clock cycle	Exe time
CGRA	FFT 2point	168	68	4	6.103 ns
	FFT 8point	168	133	48	0.48us
FPGA	FFT 2point	200	130	--	--
	FFT 8point	499	160	--	--

The Xilinxs XPower Analyzer (XPA) tool is used to calculate the static and dynamic power consumption of the proposed architecture. For calculating the power consumption, the following input files are required:

(i) Physical constraint file (PCF), which specifies the design constraints.
(ii) Native circuit description (NCD) file, which specifies the design resources.
(iii) Switching activity interchange format (SAIF) file, which specify the simulated activity rates of the signals.

The post PAR simulation provides the NCD and PCF files. These files are providing the necessary information for the calculation of total quiescent power of the device. The SAIF file is used to calculate the total dynamic power (or) toggling/switching rates of the device.

5 Conclusion

In this paper, we have presented the design of a large scale template based Coarse Grain Reconfigurable Array on which various algorithm can be mapped. An 8 point Fast Fourier Transform kernels were mapped on it. The simulation and synthesis results show the performance of the Fast Fourier Transform accelerators. Table 2 lists the power consumption (dynamic power *P*Dyn and total core power *P*Core) and energy efficiency (*E*Eff) performance for benchmark applications. All figures are generated at 100 MHz. Frequency. The processing units consume about 51% of total power, with 33% for ALUs and 8% for logic components. The on-chip memories and interconnections consume another 53% of total power. The propose CGRA consumes less power as compared to conventional FPGA when the intensive application mapped on it.

References

[1] Brunelli, C., Garzia, F., & Nurmi, J. (2008). A Coarse-Grain Reconfigurable Architecture for Multimedia Applications Featuring Sub word Computation Capabilities. In Journal of real-time image processing (Vol. 3, Nos. 1–2, pp. 21–32). Springer-Verlag. doi:10.1007/s11554-008-0071-3.

[2] Singh, H., Lee, M. H., Lu, G., Kurdahi, F. J., Bagherzadeh, N., & Filho, E. M. C. (2000). Morphosys: An integrated reconfigurable system for data-parallel and computationintensive applications. IEEE Transactions on Computers, 49(5), 465–481.

[3] Mei, B., Vernalde, S., Verkest, D., Man, H. D., & Lauwereins, R. (2003). ADRES: An architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix. Field-Programmable Logic and Applications, 2778, 61–70, ISBN 978-3-540-40822-2.

[4] Baumgarte, V., Ehlers, G.,May, F., Nuckel, A., Vorbach,M. & Weinhardt, M. (2003). PACT XPP-A self-reconfigurabledata processing architecture. The Journal of Supercomputing, 26(2), 167–184.

[5] Ethan Mirsky and AndrC DeHon. "MATRIX: A Reconfigurable Computing Architecture with Configurable Instruction Distribution and Deployable Resources." Pages 157-166,IEEE 1996

[6] Carl Ebeling, Darren C. Cronquist, Paul Franklin, Jason Secosky, and Stefan G. Berg. "Mapping Applications to the RaPiD Configurable Architecture." Pages 106-115, IEEE 1997

[7] Jong-eun Lee, Kiyoung Choi and 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), IEEE 2003

[8] Hartej Singh, Ming-Hau Lee, Guangming Lu, Fadi J. Kurdahi, Nader Bagherzadeh, and Eliseu M. C. Filho. "MorphoSys: An Integrated Reconfigurable System for Data-Parallel Computation-Intensive Applications." sbcci, pp.134, XI BrazilianSymposium on Integrated Circuit Design, 1998

[9] Francisco Javier Veredas, Michael Scheppler,Will Moffat and Bingfeng Mei. "Custom Implementation of the Coarse-Grained Reconfigurable ADREs Architecture Multimedia Purpose"

[10] Cao Liang and Xinming Huang. "SmartCell: An Energy Efficient Coarse-Grained Reconfigurable Architecture for Stream-Based Applications." Hindawi Publishing Corporation EURASIP Journal on Embedded Systems Volume 2009, Article ID 518659

[11] Kunjan Patel, SeamasMcGettrick and Chris J. Bleakley. "SYSCORE: A Coarse Grained Reconfigurable Array Architecture for Low Energy Biosignal Processing." IEEE International Symposium on Field-Programmable Custom Computing Machines. Pages 109-112, IEEE 2011

[12] Seth Copen Goldstein, Herman Schmit, Mihai Budiu, Srihari Cadambi, Matt Moe and R. Reed Taylor. "PipeRench: A Reconfigurable Architecture and Compiler." Pages 70-77, IEEE April-2000

[13] Jong-eun Lee, Kiyoung Choi and 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), IEEE 2003

[14] Cao Liang and Xinming Huang. "Smart Cell: An Energy Efficient Coarse-Grained Reconfigurable Architecture for Stream-Based Applications." Hindawi Publishing Corporation EURASIP Journal on Embedded Systems Volume 2009, Article ID 518659

[15] Chenxin Zhang, Thomas Lenart, Henrik Svensson and Viktor Öwall. "Design of Coarse-Grained Dynamically Reconfigurable Architecture for DSP Applications." In proceedings of IEEE symposium on International Conference on Reconfigurable Computing and FPGAs Pages 338- 343, IEEE 2009

[16] Dongwook Lee, Manhwee Jo, Kyuseung Han and Kiyoung Choi. "FloRA: Coarse-Grained Reconfigurable Architecture with Floating-Point Operation Capability." Pages 376-379, IEEE 2009