Review of Parallel Polynomial Multiplier based on FFT using Indian Vedic Mathematics

Shilpa Jumde
M.Tech Scholar
Department of E & T
BDCE, Sevagram, India

R. N. Mandavgane
Associate Professor
HOD, Department of E & T
BDCE, Sevagram, India

D. M. Khatri
Assistant Professor
Department of E & T
BDCE, Sevagram, India

ABSTRACT
In general, most of the operations performed by any complex system need a multiplier. Hence, multiplier based on FFT is the desired aim. In this paper, we have presented a review of parallel polynomial multiplier based on FFT using Indian Vedic mathematics. Parallel polynomial multipliers were optimized for throughput and area resources, respectively. These multipliers are used for multiplication of different polynomial numbers based on exponential type, power type, etc. FFT system is used for multiplication so complex multiplier is the main part of this design. The coding of the design can be done in VHDL. For synthesis and simulation of the design Xilinx ISE EDA tool can be used.

Keywords
FFT, Polynomial multiplier; Vedic Mathematics; VHDL; XILINX.

1. INTRODUCTION
We Polynomial multiplication is used in various scientific and engineering applications, such as digital signal processing, cryptography and coding theory. Polynomial arithmetic is an important part of computer algebra systems and has an important impact on the performance of these types of packages. The design of hardware accelerators for polynomial multiplication is an important research topic in recent works. Polynomial multiplication is carried out by using three algorithms viz. Schoolbook, Karatsuba and FFT. The FFT algorithm is used for multiplying and powering dense polynomials. FFT algorithms for polynomial multiplication are categories into two parts viz. using FFT based on complex floating-point computations and using number theoretical properties, for example FFT which perform polynomial multiplication over a finite field. The last part is the mostly used in polynomial multiplication and is implemented using multi-cores on FPGA, and the multiplication is implemented on GPU [1].

The Fast Fourier Transform is the collection of efficient algorithms that perform the Discrete Fourier Transform. The DFT is the computation of the point-value representation of a sequence of samples 'N'. It has so many applications in digital signal processing. Digital Signal Processing (DSP) kernels such as the Discrete Fourier Transform (DFT) are commonly used in real-time applications such as wireless

LAN applications. Since DFT computations require a large amount of arithmetic operations, Fast Fourier Transform (FFT) processors are advancement in DFT to meet the real-time requirement. Different FFT algorithms, like the Radix-4 and the Split-Radix FFT algorithm, used to reduce the number of computations [3].

In recent years, there has been some development in the design of FFT processors based on multi-path pipelined that provide a high throughput. FFT processor architectures are come in picture to utilize the OFDM transmission, such as a single path delay commutator (SDC), multi-path delay commutator (MDC), single path delay feedback (SDF), and multi-path delay feedback (MDF). Comparing with various FFT architectures, the MDF architecture is frequently used as a solution to provide a throughput rate of more than 1 GS/s. However, for applications that provide a throughput rate of over 2 GS/s, the number of data-paths can be increased to 8 or 16, which increases the cost of hardware. Therefore, the area becomes larger because the memory modules are duplicated for the 16 data path approach. To reduce the area and power consumption, several FFT algorithms and dynamic scaling schemes are used. Use of radix algorithm makes the architecture of the FFT processor and the complexity of the implementation small. A small radix is desirable because it results in a simple butterfly. Similarly, a high radix reduces the number of twiddle factor multiplications. The radix rk algorithms easily achieve a simple butterfly and a reduced number of twiddle factor multiplications. The radix-2 algorithm is a widely used algorithm for FFT processors, but it requires many complex multipliers. The radix-4 algorithm is also used for high data throughput FFT architectures, but requires a 4-point butterfly unit with high complexity. The radix 25 FFT algorithm and architecture is generally used in order to reduce the number of complex multipliers [4].

In most of the DSP applications, multiplier is the main of the system. The speed of that system is mainly depends upon the multiplier. If the multiplier is efficient for performing fast operations then the overall speed of the design automatically increases. So, there is need of high speed multiplier in every system which consists of multiplier. In some DSP applications, polynomial multiplications are carried out such as in FFT processor. Hence, if we replace the multipliers used by that system by most efficient multiplier based on Vedic mathematics then the speed of operation of that system will increase and the system will become more efficient. Hence, polynomial multiplier based on FFT using Vedic mathematics is a necessary choice.

2. LITERATURE SURVEY
C. P. Rentería-Mejía et.al [1] proposed a Hardware Design of FFT Polynomial Multipliers. In this paper, they present the design of two FFT polynomial multipliers using parallel and sequential architectures. Parallel and sequential polynomial multipliers were optimized for throughput and area resources, respectively. The designs are described in generic structural VHDL, synthesized on the Stratix EP4SGX230KF40C2 using Quartus II V. 13, and verified using SignalTap. The hardware synthesis and performance results show that the designed

multipliers present a good area throughput trade-off and they are suitable for high-performance scientific computing applications. Their work presents the design of two polynomial multipliers based on FFT. In this case, they used FFT based on complex fixed-point computations and R22SDF architecture. Parallel and sequential polynomial multipliers were optimized for throughput and area resources, respectively. Also, the designed multipliers were parameterized for polynomials of 8, 16, 32, 64, 128, 256 and 512 coefficients. The synthesis results show that the designed polynomial multipliers use few area resources and have a good throughput. The parallel polynomial multiplier uses 53 % more resources and its throughput is in average 1.81 times bigger than the sequential polynomial multiplier. Also, the designed multipliers carry out the polynomial multiplication in less time than the corresponding software simulation in Maple 15, which was performed on an Intel Core i7-3770 CPU @ 3.40 GHz □□8 running on Windows 7. Therefore, taking into account the synthesis and hardware verification results, they conclude that the designed multipliers are suitable for high-performance scientific computing applications [1].

Table 1. Performance results for pp and sp multipliers [1]

N	No of cycles		Mult. Time (μs)		Throughput (Gb/s)	
	PP	SP	PP	SP	PP	SP
8	40	68	0.64	1.12	1.05	0.60
16	74	127	1.21	2.11	1.19	0.68
32	140	242	2.30	4.24	1.29	0.70
64	270	469	4.43	7.76	1.37	0.78
128	528	920	8.74	15.86	1.39	0.77
256	1042	1819	17.29	31.82	1.42	0.77
512	2068	3614	34.31	63.44	1.43	0.77
1024	4118	7201	68.19	126.22	1.44	0.78

Lo Sing Cheng et.al [2] presents an Efficient FPGA Implementation of FFT Based Multipliers. Finite field multiplication is one of the most useful arithmetic operations and has applications in many areas such as signal processing, coding theory and cryptography. However, it is also one of the most time consuming operation in both software and hardware, which makes it pertinent to develop a fast and efficient implementation. In this paper, they have proposed a novel FFT based finite field multiplier to address this problem. For their purposes, they will use its efficient computation for polynomial multiplication. The FFT performs polynomial multiplication in O(nlog(n)) time compared to the classical method time of O(n2). The idea of using the FFT for finite field multiplication has been researched extensively. In this paper, they have presented a new finite field FFT multiplier implementation, which uses the NTT approach. The NTT approach had many advantages including reduced complexity, reduced registers and elimination of error, making this, to our knowledge, the fastest and first of its kind [2].

E. Theochari et.al [3] presents a reusable Intellectual Property cores for the efficient implementation of Digital Signal Processing (DSP) applications such as wireless LAN. More specifically, a Split-Radix FFT algorithm implementation architecture, whose applicability for these communication systems has been proven, was designed using reusable VHDL. Four different implementations of the Split-Radix butterfly element are presented. These different butterfly elements allow tradeoffs between performance, power consumption and hardware complexity. Finally, for demonstration purposes, comparison results of Split-Radix and Radix-4 implementations on Virtex and Virtex-II devices are also presented. Reusable IP cores for the efficient implementation of the FFT were presented. Four alternative Radix-4 and Split-Radix butterflies suitable for computing FFT were used. The various alternative architectures were implemented in various FPGA devices and in combination with efficient use of the hardware resources available in the target devices lead to various performance, area and power consumption trade-offs [3].

A.Ronisha Prakash et.al [4] proposed a FFT Processor Using Conventional and Vedic Algorithm for Performance Evaluation. Recently digital signal processing has received high attention due to the advancement in multimedia and wireless communication. Accordingly Orthogonal Frequency Division Multiple Access (OFDM) technique based on Time Division Duplex (TDD) is an attractive technology for high data rate wireless access in multichannel communication. The modulation and demodulation of OFDM are done by Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) respectively. In this paper we propose a Vedic algorithm for the implementation of multiplier that is to be used in radix 25 512-point FFT processor. The multipliers based on Vedic mathematics are one of the fastest and low power multiplier. It enables parallel generation of partial product and eliminates unwanted multiplication steps. Thus Vedic multipliers ensure substantial reduction of propagation delay in FFT processor. The FFT processor employing Vedic multiplier reduces hardware complexity in area and power in FPGA implementation. The proposed processor have been designed in Xilinx and implemented using Spartan 3E FPGA kit with a supply voltage of 1.2 V. The delay and power obtained using the Vedic multiplier are 173.60ns and 11×10-2 W respectively. In this paper, the Vedic algorithm and the eight parallel data-path 512-point modified radix 25 FFT processor have been proposed for OFDM-based WPAN applications. The number of complex Booth multipliers and twiddle factor LUTs are reduced using Vedic algorithm. The proposed radix 25 FFT processor is the most area-efficient architecture for the eight parallel 512-point MDF FFT processors. The delay obtained by this proposed FFT processor is about173.60ns. The proposed architecture has potential applications in high-rate OFDM-based WPAN systems [4].

Table 2: Performance Comparison of FFT Processor [4]

Performance	Conventional Algorithm	Vedic Algorithm
Delay	174.600ns	173.600ns
Power	12×10^{-2} W	11×10^{-2} W
Temperature	30.4ºC	28.2ºC

Ali Chamas Al Ghouwayel et.al [5] gives a theory Towards a Triple Mode Common Operator FFT for SoftWare Radio Systems. A scenario to design a Triple Mode FFT is addressed. Based on a Dual Mode FFT structure, they presents a methodology to reach a triple mode FFT operator (TMFFT) able to operate over three different fields: complex number domain C, Galois Fields GF (Ft) and GF (2m). They propose a reconfigurable Triple mode Multiplier that

constitutes the core of the Butterfly based FFT. A scalable and flexible unit for the polynomial reduction needed in the GF (2m) multiplication have also proposed. An FPGA implementation of the proposed multiplier is given and the measures show a gain of 18% in terms of performance-to-cost ratio compared to a "Velcro" approach where two self-contained operators are implemented separately. A parameterizable polynomial reduction unit have been proposed allowing the execution of the GF(2m) multiplication for various sizes of Galois field. The proposed multiplier have been implemented on FPGA devices and the obtained measures show an excess of only 6% in terms of ALUTs compared to the complexity of the dual mode multiplier proposed in. In terms of performance-to-cost ratio, the common operator approach represented by the triple mode multiplier outperforms the Velcro approach by a gain of around 18 %. An approach to design a TMFFT operator is described which will be further developed in near future works. The intended TMFFT operator can be used in two different contexts: modulation / demodulation OFDM and Reed Solomon encoding and decoding over GF (Ft) and GF (2m) [5].

Table 3. Implementation results of the dual and triple mode Multipliers on stratix ii, ep2s15f484c3 device [5]

Multiplier	** n=6	n=8	n=10
Dual Mode	103 ALUTs 3.97 ns	194 ALUTs 4.1 ns	289 ALUTs 4.86 ns
Velcro	131 ALUTs 3.97 ns	245 ALUTs 4.13 ns	364 ALUTs 4.87 ns
Triple Mode	109 ALUTs 3.97 ns	206 ALUTs 4.17 ns	307 ALUTs 4.9 ns
$\eta = \frac{1}{TC} * 10^6$	$\eta_V = 1922$ $\eta_C = 2310$	$\eta_V = 988$ $\eta_C = 1164$	$\eta_V = 564$ $\eta_C = 664$
Performance-to-cost ratio gain	**20.1** %	**17.8** %	**17.7** %

3. PROPOSED WORK

The proposed work is to achieve the FFT core, memory units, complex multiplier based on Vedic mathematics for increase the speed of calculation and IFFT core. Hence, polynomial multiplier based on FFT using Vedic mathematics with high speed is the probable outcome of this work.

Finite field multiplication is one of the most useful arithmetic operations and has applications in many areas such as signal processing and coding theory. However, it is also one of the most time consuming operations in both software and hardware, which makes it pertinent to develop a fast and efficient implementation. FFT based finite field multiplier can be used to solve this problem.

A complex multiplier can be composed with four real multipliers, one adder and one subtracter. Attempts have been made to optimize the realization of the complex multiplier by reducing the number of multipliers and accumulating the partial products; however, the wider the input, the more partial product layers that must be added in order to compute the result. To solve this problem, we can consider the LNS (Logarithmic Number System) to realize the multiplication, which uses a single addition to replace a standard two-operand multiplication.

FFT based parallel polynomial multiplier requires more time for polynomial multiplication. This will make the design more efficient as compare to the conventional design.

This research work targets the design of a FFT based polynomial multiplier using Vedic mathematics.

- To design FFT core.
- To design memory devices viz. registers.
- To design a MUX.
- To design a complex number multiplier using Vedic mathematics.
- To design memory such as RAM.
- To design IFFT core.

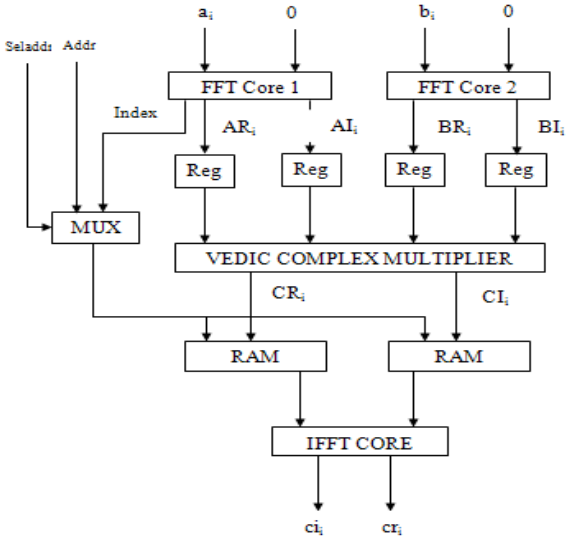

Fig 1: Basic Block Diagram of FFT based Polynomial Multiplier

The operation of polynomial multiplier based on FFT is carried out as follows;

- Calculate the point-value representation of the polynomials A(x) and B(x) using the blocks FFT core 1 and FFT core 2, respectively. The coefficients of these polynomials are processed in serial form and descending order. The point value representations of A(x) and B(x) are obtained in bit reversed order and serial form.
- Calculate the point wise product of the point-value representations (ARi + jAIi and BRi + jBIi) obtained in the step 1, using the complex vedic multiplier. This product is concurrently performed with the step 1 and is stored in the RAMs RP and IP, where the write address is Index.
- Interpolate the coefficients of the polynomial C(x) using the IFFT core, which processes the point wise products Cri + jCIi in ascending order. The coefficients C0-C2n-1 are obtained in bit-reversed order and serial form.

4. CONCLUSION

In this paper, various top-down design methods are shown, using that techniques the parallel polynomial multiplier as well as sequential polynomial multiplier has been designed. VHDL language has been used to describe the design. They use various tools for design, synthesis and simulation of the design and used FPGA kits for the implementation.

5. REFERENCES

[1] C. P. Rentería-Mejía, A. López-Parrado, J. Velasco-Medina, "Hardware Design of FFT Polynomial Multipliers", 978-1-4799-2507-0/14/$31.00 ©2014 IEEE.

[2] Lo Sing Cheng, Ali Miri, Tet Hin Yeap, "EFFICIENT FPGA IMPLEMENTATION OF FFT BASED MULTIPLIERS", 0-7803-8886-0/05/$20.00 ©2005 IEEE.

[3] E. Theochari, K. Tatas, D. J. Soudris, K. Masselos, K. Potamianos, "A REUSABLE IP FFT CORE FOR DSP APPLICATIONS", 0-7803-8251-X/04/$17.00 © 2004 IEEE.

[4] A.Ronisha Prakash, S.Kirubaveni, "Performance Evaluation of FFT Processor Using Conventional and Vedic Algorithm", 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN 2013), 978-1-4673-5036-5/13/$31.00 © 2013 IEEE 89.

[5] Ali Chamas Al Ghouwayel, Amin Haj-Ali and Zouhair El-Bazzal, "Towards a Triple Mode Common Operator FFT for SoftWare Radio Systems", 19th International Conference on Telecommunications (ICT 2012), 978-1-4673-0747-5/12/$31.00 ©2012 IEEE.

[6] P. D. Chidgupkar and M. T. Karad, "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu., volume 8, Issue no. 2, Year 2004.

[7] M. Moreno and Y. Xie, "FFT-based dense polynomial arithmetic on multicores", 23rd Int. conf. on high perf. Comp. systems and applic., June Year 2009, p.p. 378-399.

[8] Sushma R. Huddar and Sudhir Rao, Kalpana M., Surabhi Mohan, "Novel High Speed Vedic Mathematics Multiplier using Compressors", 978-1-4673-5090-7/13/$31.00 ©2013 IEEE.

[9] Laxman P.Thakre, Suresh Balpande, Umesh Akare, Sudhir Lande, "Performance Evaluation and Synthesis of Multiplier used in FFT operation using Conventional and Vedic algorithms", 978-0-7695-4246-1/10 $26.00 © 2010 IEEE.

[10] Parth Mehta, Dhanashri Gawali, "Conventional versus Vedic mathematical method for Hardware implementation of a multiplier", 978-0-7695-3915-7/09 $26.00 © 2009 IEEE.

[11] M. Ramalatha, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", 978-1-4244-3834-1/09/$25.00 © 2009 IEEE.

[12] Sumit Vaidya and Deepak Dandekar, "DELAY-POWER PERFORMANCE COMPARISON OF MULTIPLIERS IN VLSI CIRCUIT DESIGN", International Journal of Computer Networks & Communications (IJCNC), Volume 2, Issue no.4, July Year 2010.