International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 11, Issue 05 (May 2015), PP.23-28

Fast Fourier Transform utilizing Modified 4:2 & 7:2 Compressor

Mahesh Daryani[1], Kanak Kumar [2]
[1]*MIT, Bhopal, (M.P.)*
[2]*MIT, Bhopal (M.P.)*

Abstract:- With the advent of new technology in the fields of VLSI and communication, there is also an ever growing demand for high speed processing and low area design. It is also a well-known fact that the multiplier unit forms an integral part of processor design. Due to this regard, high speed multiplier architectures become the need of the day. In this paper, we introduce a novel architecture to perform high speed multiplication using ancient Vedic math's techniques. A new high speed approach utilizing 4:2 compressors and novel 7:2 compressors for addition has also been incorporated in the same and has been explored. Upon comparison, the compressor based multiplier introduce in this paper, is almost two times faster than the popular methods of multiplication. This all design and experiments were carried out on a Xilinx Spartan 3e series of FPGA and the timing and area of the design, on the same have been calculated.

Keywords:- 4:2 Compressor, Modified 4:2 Compressor, Proposed 4:2 Compressor, 7:2 Compressor, Urdhwa Multiplier

I. INTRODUCTION

Digital signal processing (DSP) is the mathematical manipulation of an information signal to modify or improve it in some way. It is characterized by the representation of discrete time, discrete frequency, or other discrete domain signals by a sequence of numbers or symbols and the processing of these signals [1].

The goal of DSP is usually to measure, filter and/or compress continuous real-world analog signals. The first step is usually to convert the signal from an analog to a digital form, by sampling and then digitizing it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of numbers. However, often, the required output signal is another analog output signal, which requires a digital-to-analog converter (DAC). Even if this process is more complex than analog processing and has a discrete value range, the application of computational power to digital signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression. DSP algorithms have long been run on standard computers, as well as on specialized processors called digital signal processor and on purpose-built hardware such as application-specific integrated circuit (ASICs). Today there are additional technologies used for digital signal processing including more powerful general purpose microprocessors, field-programmable gate arrays (FPGAs), digital signal controllers (mostly for industrial apps such as motor control), and stream processors, among others [2-3]. The FFT is one of the most commonly used digital signal processing algorithm. Recently, FFT processor has been widely used in digital signal processing field applied for OFDM, MIMO-OFDM communication systems. FFT/IFFT processors are key components for an orthogonal frequency division multiplexing (OFDM) based wireless IEEE 802.16 broadband communication system; it is one of the most complex and intensive computation module of various wireless standards physical layer (ofdm-802.11a, MIMO-OFDM 802.11, 802.16,802.16e) [4].

II. COMPRESSOR BASED MULTIPLIER

Vedic mathematics is an ancient fast calculation mathematics technique which is taken from historical ancient book of wisdom. Vedic mathematics is an ancient Vedic mathematics which provides the unique technique of mental calculation with the help of simple rules and principles. Swami Bharati Krishna Tirtha (1884-1960), former Jagadguru Sankaracharya of Puri culled set of 16 Sutras (aphorisms) and 13 Sub - Sutras (corollaries) from the Atharva Veda. He developed methods and techniques for amplifying the principles contained in the formulas and their sub-formulas, and called it Vedic Mathematics. According to him, there has been considerable literature on Mathematics in the Veda-sakhas.

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus.

- **4:2 Compressor**

To add binary numbers with minimal carry propagation we use compressor adder instead of other adder. Compressor is a digital modern circuit which is used for high speed with minimum gates requires designing technique. This compressor becomes the essential tool for fast multiplication adding technique by keeping an eye on fast processor and lesser area.

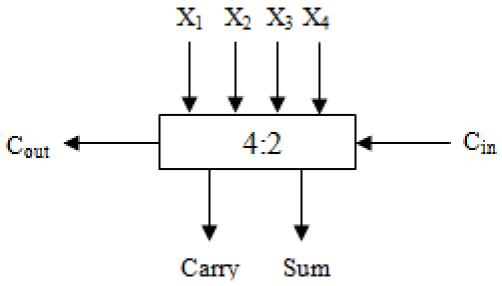
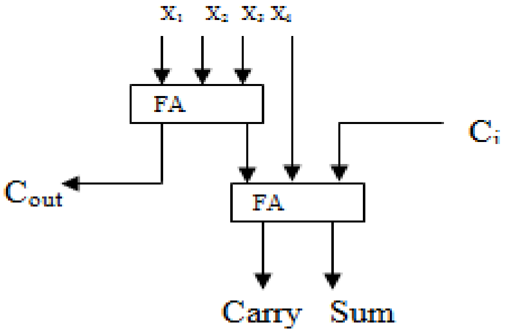


Figure 1: Block Diagram of 4:2 Compressors

4:2 compressors are capable of adding 4 bits and one carry, in turn producing a 3 bit output. The 4-2 compressor has 4 inputs X_1, X_2, X_3 and X_4 and 2 outputs Sum and Carry along with a Carry-in (Cin) and a Carry-out (Cout) as shown in Figure 1. The input Cin is the output from the previous lower significant compressor.

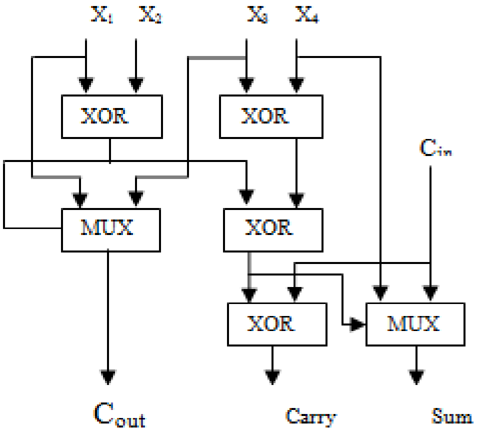
The Cout is the output to the compressor in the next significant stage. The critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adders. Similar to the 3-2 compressor the 4-2 compressor is governed by the basic equation

$$X_1 + X_2 + X_3 + X_4 + C_{in} = Sum + 2 * (Carry + C_{out}) \quad (1)$$

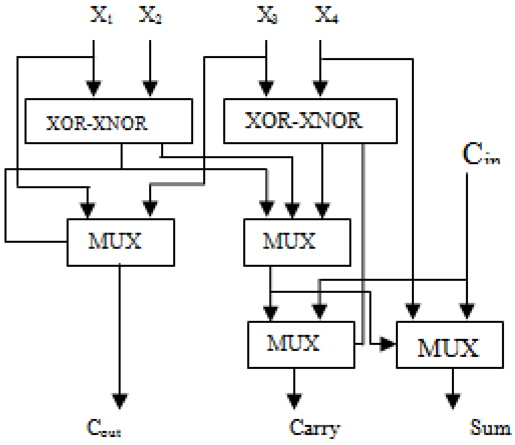


(a)

The standard implementation of the 4-2 compressor is done using 2 Full Adder cells as shown in Figure 2(a). When the individual full Adders are broken into their constituent XOR blocks, it can be observed that the overall delay is equal to 4*XOR.



(b)



(c)

Figure 2: Design of 4:2 compressor using (a) full adder, (b) XOR and Multiplexer, (c) Proposed 4:2 Compressor

The block diagram in Figure 2(b) shows the existing architecture for the implementation of the 4-2 compressor with a delay of 3*XOR. The equations governing the outputs in the existing architecture are shown below

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in} \tag{2}$$

$$C_{out} = (X_1 \oplus X_2).X_3 + \overline{(X_1 + X_2)} \cdot X_3 \tag{3}$$

$$Carry = (X_1 \oplus X_2 \oplus X_3 \oplus X_4) \cdot C_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)} \cdot X_4 \tag{4}$$

However, like in the case of 3-2 compressor, the fact that both the output and its complement are available at every stage is neglected. Thus replacing some XOR blocks with multiplexer's results in a significant improvement in delay. Also the MUX block at the SUM output gets the select bit before the inputs arrive and thus the transistors are already switched by the time they arrive. This minimizes the delay to a considerable extent. This is shown in Figure 2(c).

The equations governing the outputs in the proposed architecture are shown below

$$Sum = (X_1 \oplus X_2) \cdot (X_3 \oplus X_1) \mathrm{T} (X_1 \oplus X_2) \cdot (X_3 \oplus X_1) \cdot C_{in} \tag{5}$$

$$C_{out} = (X_1 \oplus X_2).X_3 + \overline{(X_1 + X_2)} \cdot X_1 \tag{6}$$

$$Carry = (X_1 \oplus X_2 \oplus X_3 \oplus X_4) \cdot C_{in} + \overline{(X_1 \oplus X_2 \oplus X_3 \oplus X_4)} \cdot X_4 \tag{7}$$

- **7:2 Compressor**

Similar to its 4:2 compressor counterpart, the 7:2 compressor as shown in Figure 3, is capable of adding 7 bits of input and 2 carry's from the previous stages, at a time. In our implementation, we have designed a novel 7:2 compressor utilizing two 4:2 compressors, two full adders and one half adder. The architecture for the same has been shown in Figure 4.

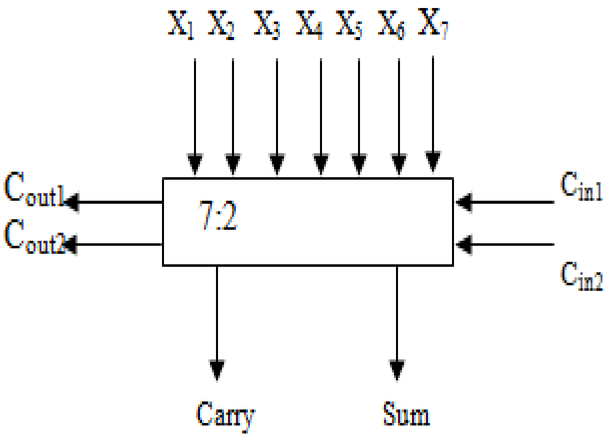


Figure 3: Block Diagram of 7:2 Compressors

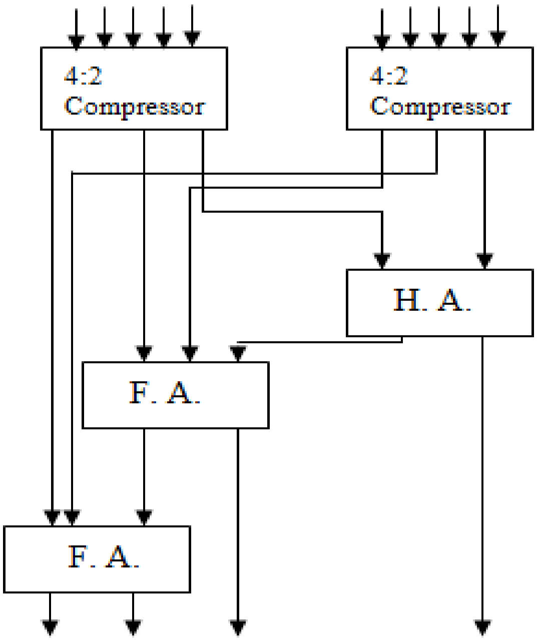
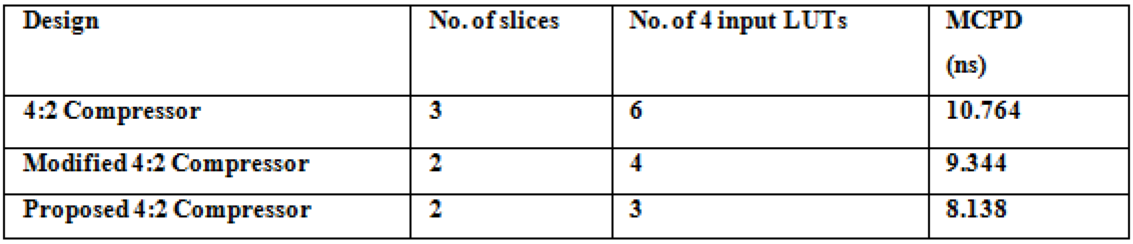


Figure 4: 7:2 Compressor using 4:2 Compressor

III. SIMULATION RESULT

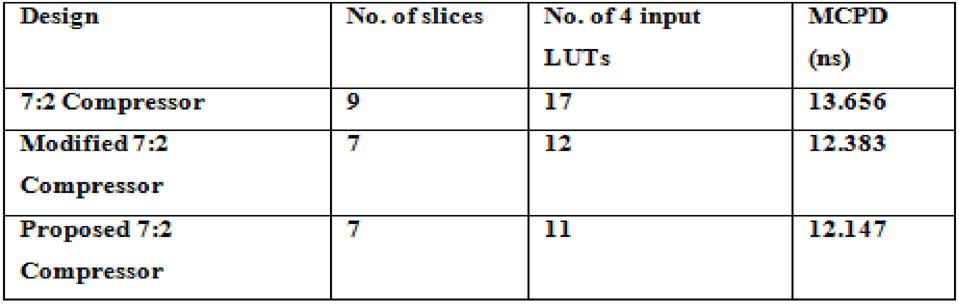
All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx 14.1i updated version. Xilinx 14.1i has couple of the striking features such as low memory requirement, fast debugging, and low cost. The latest release of ISE^{TM} (Integrated Software Environment) design tool provides the low memory requirement approximate 27 percentage low. By the aid of that software we debug the program easily. Also included is the newest release of the chip scope Pro Serial IO Tool kit, providing simplified debugging of high-speed serial IO designs for Spratan-3 FPGAs. With the help of this tool we can develop in the area of communication as well as in the area of signal processing and VLSI low power designing. We functionally verified each unit presented in this paper including all three 4:2 Compressor, 7:2 Compressor, Compressor based Multiplier and 8-point fast fourier transform. We have been found from the results shown in Table 1 to Table 3 respectively, that number of slices used is same in case of 4:2 compressor adder and 4:2 modified compressor adder which is less than slices used in 4:2 proposed compressor adder.

Table 1: Device utilization summary (Spartan 3) of 4:2 Compressors, Modified 4:2 Compressor and Proposed 4:2 Compressor



Design	No. of slices	No. of 4 input LUTs	MCPD (ns)
4:2 Compressor	3	6	10.764
Modified 4:2 Compressor	2	4	9.344
Proposed 4:2 Compressor	2	3	8.138

Table 2: Device utilization summary (Spartan 3) of 7:2 Compressors, Modified 7:2 Compressor and Proposed 7:2 Compressor



Design	No. of slices	No. of 4 input LUTs	MCPD (ns)
7:2 Compressor	9	17	13.656
Modified 7:2 Compressor	7	12	12.383
Proposed 7:2 Compressor	7	11	12.147

Table 3: Device utilization summary (Spartan 3) of Compressors based Multiplier, Modified Compressor based Multiplier and Proposed Compressor based Multiplier

Design	No. of slices	No. of 4 input LUTs	MCPD (ns)
Compressor based Multiplier	108	190	55.050
Modified Compressor based Multiplier	78	143	41.684
Proposed Compressor based Multiplier	79	142	33.164

Table 4: Device utilization summary (Spartan 3) of Compressors based Multiplier, Modified Compressor based Multiplier and Proposed Compressor based Multiplier

Design	No. of slices	No. of 4 input LUTs	MCPD (ns)
FFT using Compressor based Multiplier	191	336	29.905
FFT using Modified Compressor based Multiplier	162	286	22.867
FFT using Proposed Compressor based Multiplier	132	230	22.309

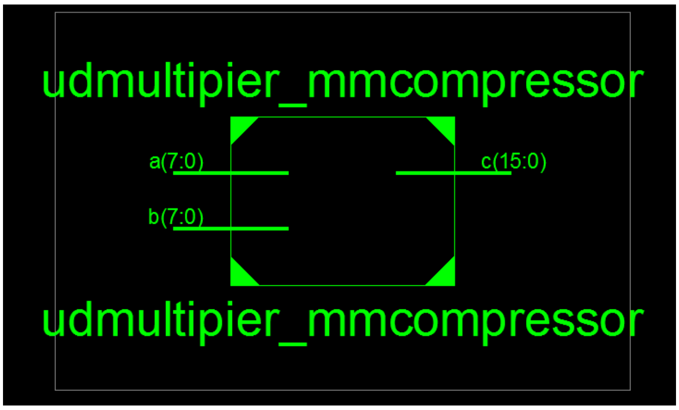


Figure 5: RTL view of proposed Multiplier

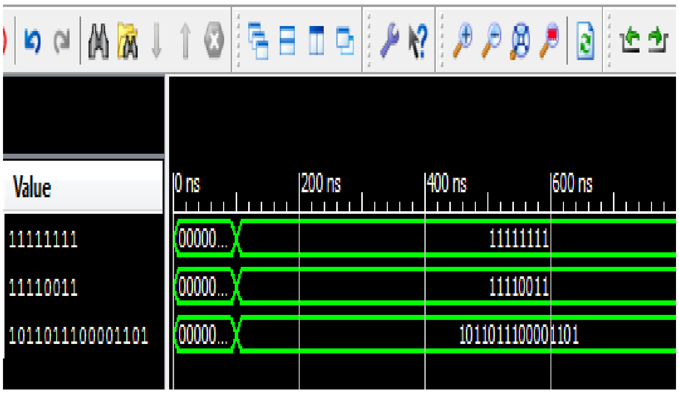


Figure 6: VHDL test waveform of Proposed Multiplier

IV. CONCLUSION

Fast fourier transform (FFT) is used to convert complex and real values into real and complex ones. It requires decomposition of data into stages using butterfly similar to DFT. But the butterfly used in FFT is quite different in terms of coefficients or multipliers. With the increase in number of FFT sequence length the number of coefficients is also increased simultaneously. Delay provided and area required by hardware are the two key factors which are need to be consider. Among all three designs, proposed compressor based multiplier provides the least amount of Maximum combinational path delay (MCDP). Also, it takes least number of slices i.e. occupy least area among all three design.

REFRENCES

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

[2]. Avneesh Kumar, Paresh Rawat, “High Speed Area Efficient 8-point FFT using Vedic Multiplier”, IJERA ISSN: 2248-9622, Vol. 4, Issue 12 (part 4), December 2014, pp. 105-108.

[3]. S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A, “Implementation of Vedic multiplier for Digital Signal Processing”, International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011, Proceedings published by International Joural of Computer Applications® (IJCA), pp.1-6.

[4]. Himanshu Thapaliyal and M.B Srinivas, “VLSI Implementation of RSA Encryption System Using Ancient Indian Vedic Mathematics”, Center for VLSI and Embedded System Technologies, International Institute of Information Technology Hyderabad, India.

[5]. Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, “Vedic Mathematics: Sixteen simple Mathematical Formulae from the Veda”, Delhi(2011).

[6]. Sumit Vaidya and Depak Dandekar. “Delay-power perfor-mance comparison of multipliers in VLSI circuit design”. International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.

[7]. P. D. Chidgupkar and M. T. Karad, “The Implementation of Vedic Algorithms in Digital Signal Procesing”, Global J. of Eng. Edu, Vol.8, No.2, 204, UICEE Published in Australia.

[8]. Asmita Haveliya, “Design and Simulation of 32-Point FFT Using Radix-2 Algorithm for FPGA Implementation”, Second International Conference on Advanced Computing & Communication Technologies IEEE 2012.

[9]. S. Correa, L. C. Freitas, A. Klautau and J. C. W. A. Costa, “VHDL Implementation of a Flexible and Synthesizable FFT Processor”, IEEE LATIN AMERICA TRANSACTIONS, VOL. 10, NO. 1, JAN. 2012.

[10]. Kamaru Adzha Bin Kadiran. “Design and Implementation of OFDM Transmitter and Receiver on FPGA Hardware”, November 2005.