A 32-Point Fft using Vedic Mathematics

N. Suresh
M.Tech, Associate Professor
Department of ECE
ASIST,Paritala, A.P

V. Phani Kumar
B.Tech, Student
Department of ECE
ASIST,Paritala, A.P

R. Jyothi
B.Tech, Student
Department of ECE
ASIST,Paritala, A.P

M. Navya
B.Tech, Student
Department of ECE
ASIST,Paritala, A.P

U. Manikanta
B.Tech, Student
Department of ECE
ASIST,Paritala, A.P

S. Kiran
B.Tech, Student
Department of ECE
ASIST,Paritala, A.P

Abstract:- **This paper "FFT using Vedic mathematics" proposes a novel technique in order to improve speed and reduce the time delay in FFT Multiplication process. Fast Fourier Transform is widely used in various fields like image and signal processing, Voice recognition systems etc. Vedic mathematic comprises of 16 sutras in that one of sutras namely Urdhwa Tiryakbhyam sutra used here to improve the performance of FFT multiplier. Here, the proposed method "FFT using Vedic mathematics" compared with the one of the popular multiplier namely modified Radix-2 Booth multiplier in terms of delay which shows better result.**

1. INTRODUCTION

FFT is a highly efficient procedure for computing the DFT of finite series and requires less number of computation than that of direct evaluation of the DFT.FFT reduces the computations that of direct the calculation of the coefficients of the DFT can be carried out iteratively. FFT reduces the computation time required to compute a Discrete Fourier Transform andimprove the performance by a factor 100 or more over direct evaluation of the DFT. The complex multiplication in FFT algorithm is performed based on the Array and modified Booth'salgorithm. These algorithms are quite popular in modern VLSI design. A new approach to multiplier design based on ancient Vedic Mathematics is used to improve the speed of multiplication. Urdhwa Tiryakbhyam sutra in Vedic mathematics explores a novel method for implementation of 32-point FFT. using array multiplier was replaced by Vedic mathematics in Radix-2 algorithm. By using this Vedic mathematics in FFT algorithm, it improves the Speed of FFT algorithm which is used to compute the discrete Fourier transform. using array multiplier was replaced by Vedic mathematics in Radix-2algorithm.By using this Vedicmathematics in FFT algorithm, it improves the Speed of FFT algorithm which is used to compute the discrete Fourier transform.

2. FFT ALGORITHM

The Fast Fourier Transform is a highly efficient procedure for computing the Discrete Fourier Transform (DFT) of a finite series and requires less number of computations than that of direct evolution of DFT. The FFT is based on computation on decomposition and breaking the transform into smaller transforms and combing them to get total transform.The DFT of a sequence can be evaluated using the formula

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi nk/N} \quad 0 \le k \le N-1$$

Here FFT multiplication was implemented based on Radix-2 algorithm bits are taken as input and we get two output bits dependence upon the arrow operations. The basic block diagram of DIF-FFT was shown in Figure.2.1.

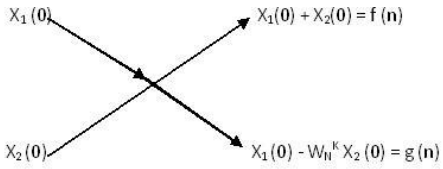

Figure.2.1: Basic butterfly structure of DIF-FFT using Radix-2 algorithm

3 . 32-POINT FFT

In 32-point FFT, there are five stages, which contain unique butterfly structure. Input is taken in normal order at first stage and the inputs for other stages dependence upon output generated by preceding stage butterfly structure. The output is taken in bit-reversal order at last stage. The block diagram for 32-point FFT is shown in Figure.2.2.

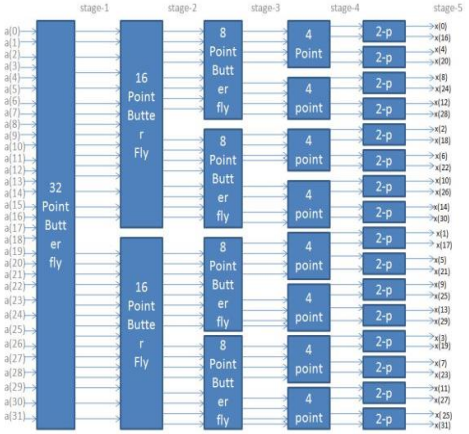

Figure 2.2: 32-point FFT block diagram

4. 32-POINT FFT BUTTERFLY STRUCTURE:

The butterfly structure of 32-point FFT is shown in Figure 2.3. For 32-point input, we require five butterfly stages to get the required output. In each stage of butterfly structure, we need to operations as shown in the Figure.2.1.

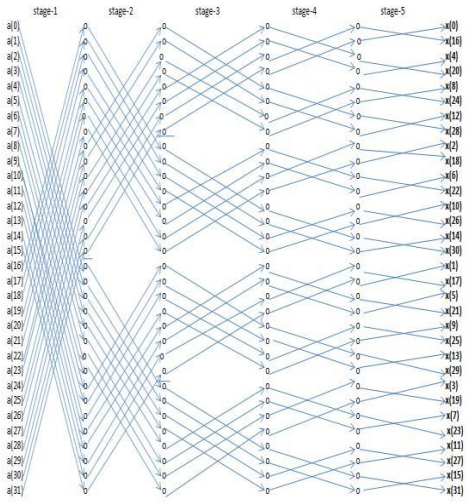

Figure 2.3:32-point FFT butterfly structure

5. MODIFIED RADIX-2 BOOTH ALGORITHM

Booth's multiplication algorithm is an algorithm which multiplies 2 signed or unsigned integers in 2's complement form. This approach uses fewer additions and subtractions than more straight forward algorithms.

Procedure to multiplication can be stated as follows: Load multiplicand and multiplier in the registers B and Q. We also need third register A, which is initialize to 0(zero).Set sequence counter value equal to number of bits in the register. If the multiplier Qn,Qn+1 digits are either 00 or 11, then simply decrement sequence counter(sc). Then all bits of the A, Q are shifted to the right one bit. If the multiplier Qn,Qn+1 digits are 10 ,then subtract multiplicand to A and decrement sequence counter (sc).Then all bits of the A,Q are shifted to the right one bit. If the multiplier Qn,Qn+1 digits are 01 ,then add multiplicand to A and decrement sequence counter (sc).For every operation compare sequence counter (sc) value with '0'.if it is not equal to zero then continue the process. If it is equal to zero then end it result is in register AQ.

The flow chat for booth multiplier is shown in Figure 3.1.

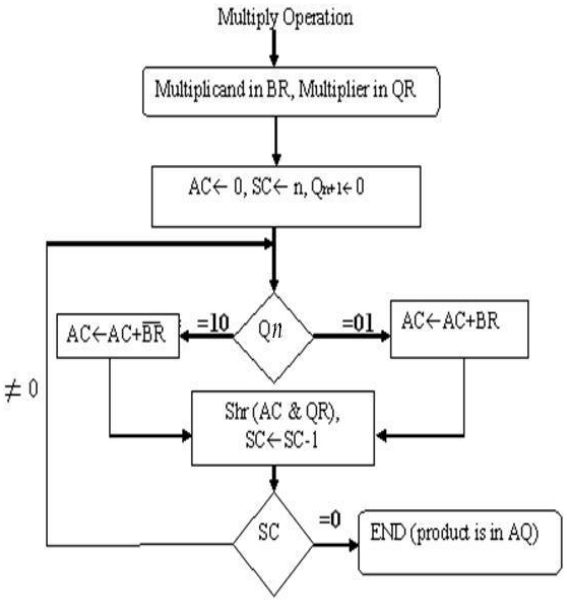

Figure 3.1: flow chat for radix-2 booth multiplier

6. PROPOSED VEDIC MATHEMATICS

Vedic mathematics is mainly based on 16 Sutras (or aphorisms)dealing with various branches of mathematics like arithmetic, algebra, geometry etc. The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. The multiplier is based on an algorithm Urdhva Tiryakbhyam(Vertical & Crosswise) of ancient Indian Vedic Mathematics. Urdhva Tiryakbhyam Sutra is a general multiplication formula applicableto all cases of multiplication. It literally means "Vertically and crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The parallelism in generation of partial products and their summation is obtained using Urdhava Triyakbhyam explained in figure below. The algorithm can be generalized for n x n bit number. The proposed 8*8 Vedic multiplier produces was explained in Figure 4.1..

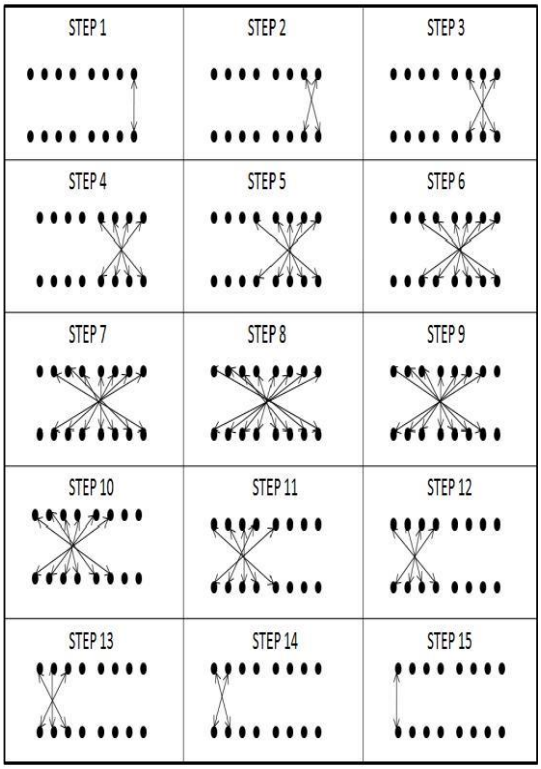

Figure 4.1:Multiplication Process in Urdhwa Tiryakbhyam
Equations obtained in Vedic multiplier based on UrdhavaTriyakbhyam sutra are

P0 =A0 * B0

C1P1 = (A1 * B0) + (A0 * B1)

C5C4P3 = (A3 * B0) + (A2 * B1) + (A1 * B2) + (A0 * B3) + C2

C7C6P4 = (A4 * B0) + (A3 * B1) + (A2 * B2) + (A1 * B3) + (A0 * B4) +C3 + C4

C10C9C8P5 = (A5 * B0) + (A4 * B1) + (A3 * B2) + (A2 * B3) + (A1 * B4) + (A0 * B5) + C5

C13C12C11P6 = (A6 * B0) + (A5 * B1) + (A4 * B2) + (A3 * B3) + (A2 * B4) + (A1 * B5) + (A0 * B6) +C7 + C8
C16C15C14P7 = (A7 * B0) + (A6 * B1) + (A5 * B2) + (A4 * B3) + (A2 * B5) + (A1 * B6) + (A0 * B7) +C9 + C11
C19C18C17P8 = (A7 * B1) + (A6 * B2) + (A5 * B3) + (A4 * B4) + (A3 * B5) + (A2 * B6) +(A1 * B7) + C10+C12 + C14

C22C21C20P9 = (A7 * B2) + (A6 * B3) + (A5 * B4) + (A4 * B5) + (A3 * B6) + (A2 * B7) +C13 +C15 + C17
C25C24C23P10 = (A7 * B3) + (A6 * B4) + (A5 * B5) + (A4 * B6) + (A3 * B7) + C16 + C18+ C20

C27C26P11 = (A7 * B4) + (A6 * B5) + (A5 * B6) + (A4 * B7) + C19 + C21 + C23

C29C28P12 = (A7 * B5) + (A5 * B6) + (A5 * B7) + C22 + C24 + C26

C30P13= (A7 * B6) + (A6 * B7) + C25 + C27 + C28
P14 = (A7 * B7) + C29 + C30.

7. RESULTS & CONCLUSION

Here the proposed method "FFT using Vedic mathematics" is compared with existing multipliers in terms of delay with different families of Xilinx10.1 shows that the proposed method has a better result shown in below table.

Table.1:Time delay(ns) results of the FFT algorithm implementation on xilinx 10.1

Families	Radix-2 modified booth algorithm	Proposed Urdha tiryabhyam
On Spartan-3	26.877	24.865
On Spartan-3A & Spartan-3AN	24.089	20.360
On Spartan-3E	23.703	19.333
On Virtex-2	17.962	16.213
On Virtex-4	15.255	13.938
On Virtex-5	12.295	8.879

REFERENCES

[1] "Design and Implementation of 32-point FFT using Radix-2 Algorithm for FPGA Implementation" by Asmita Haveliya, IEEE, Proc. Vol. 978-0-7695-4640-7/12.
[2] "High speed asic design of complex Multiplier using Vedic mathematics" by Prabir Saha, Banerjee and Partha Bhattacharyya, IEEE, Proc.vol. 978-1-4244-8943-5/11.
[3] "Novel high speed Vedic mathematics multiplier using compressors mathematics" by Sushma R. Huddar and Sudhir Rao.
[4] **"**Design and performance analysis of 32 and 64-point FFT using Radix-2 algorithm" by K.Sowjanya & B.Leela Kumari, International Conference (IRAJ) ISSN: 978- 81-927147-9-0, 14th July-2013.
[5] "High speed Vedic multiplier" international journal of engineering research volume no.3 issue no: special 2, pp: 73-76, 22 March 2014.
[6] "Design and Implementation of 32-point FFT using Radix-2 Algorithm for FPGA" by Afreen.Fathima, IOSR Journal of Electrical and Electronics Engineering (ISOR-JEEE) e-ISSN: 2778-1676, p-ISSN: 2320-3331, Vol. 9, Jan 2014.
[7] "Efficient implementation of 16-bit Multiplier-accumulator using radix-2 modified booth algorithm and spst adder using verilog" by addanki purna ramesh, Dr.a.v.n.tilak and Dr.a.m.prasad, international journal of vlsi design & communication systems (vlsics) vol.3, no.3, June 2012.
[8] "Efficient Design and Implementation of FFT", by Sneha N.khrede, Meghana Hasamnis, International Journal of Engineering Science and Technology (IJEST), ISSN: 0975-5462 NCICT Special Issue Feb 2011.

[9] "Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics" International Journal of Computer Applications (0975 – 8887) Volume 59– No.6, December 2012.
[10] "Design and performance analysis of 32 and 64-point FFT using multiple radix algorithms" by K.Sowjanya and Leela Kumari.Ballvada, International Journal of Computer applications, ISSN: 0975-8887, Vol.78-1, September 2013.