OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

High Performance MAC Unit for FFT Implementation

Tinju Tresa[1], M. A.Shameem[2], Sandeep Sreedharan[3]
[1](M. Tech Vlsi Design, VIT University, Vellore
[2](M. Tech Vlsi Design, VIT University, Vellore
[3](M. Tech Vlsi Design, VIT University, Vellore

ABSTRACT: *In this paper we have proposed an efficient way of implementing a Fast Fourier Transform (FFT) processor using high performance pipelined Multiply and Accumulate (MAC) unit. The multiplication unit is implemented using Modified Radix 4 Booth Multiplier algorithm. The proposed multiplier circuits are based on the modified Booth algorithm and the pipeline technique which are the most widely used to accelerate the multiplication speed. The adder unit is implemented using an area efficient Carry Select Adder (AECSA). As a result we can achieve lower area as compared with that of a normal Carry Select Look ahead Adder (CLSA). The implementation is done using Verilog HDL code. The simulation of the over all design is carried out using NC launch. The synthesis of our design is done using RTL compiler in Cadence. Analysis of the synthesis report shows the design to be of high performance and to be area optimised.*

Keywords: *Area Efficient CSA, DFT, DIF algorithm, 8point FF, MAC unit, Modified Radix 4 Booth's multiplier, RCA, BEC, ECA.*

I. Introduction

The digital signal processing (DSP) is one of the core technologies in multimedia and communication systems. Many application systems based on DSP, especially the recent next generation optical communication systems, require extremely fast processing of a huge amount of digital data. Most of DSP applications such as fast Fourier transform (FFT) require additions and multiplications. Since the multipliers have a significant impact on the performance of the entire system, many high-performance algorithms and architectures have been proposed to accelerate multiplication [4]. The MAC unit determines the speed of the overall system; it always lies in the critical path. Developing high speed MAC is crucial for real time DSP application. Moreover, with the ever-increasing demand for portable electronic products, an electronic component with low power consumption would surely lead the market trend. Therefore, it is needed to design a low-power MAC unit. Many researchers have attempted in designing MAC architecture with high computational performance and low power consumption. In order to improve the speed of the MAC unit, there are two major bottlenecks that need to be considered. The first one is the partial products reduction network that is used in the multiplication block and the second one is the accumulator. Both of these stages require addition of large operands that involve long paths for carry propagation [3]. Various multiplication algorithms such as Booth [5], modified Booth, Braun, Baugh-Wooley have been proposed. The modified Booth algorithm reduces the number of partial products to be generated and is known as the fastest multiplication algorithm. Many researches on the multiplier architectures including array, parallel and pipelined multipliers have been pursued and the pipelining is the most widely used technique to reduce the propagation delays of digital circuits [4]. Much different architecture were proposed for MAC implementation. Li Hsun proposed a low-power Multiplication-Accumulation Computation (MAC) unit using the radix-4 Booth algorithm, by reducing its architectural complexity and minimizing the switching activities [6]. Elgibaly proposed a fast pipelined implementation to lower the MAC architecture's critical delay [7]. Fayed et al. proposed new data merging architecture for high speed multiply accumulate units [8,9] The architecture can be applied on binary trees constructed using 4:2 compressor circuits. Increasing the speed of operation is achieved by taking advantage of the available free input lines of the compressor circuits, which result from the natural parallelogram shape of the generated partial products and using the bits of the accumulated value to fill in these gaps. This results in merging the accumulation operation within the multiplication process. In this paper, we introduce a high speed and area-efficient merged Multiply Accumulate (MAC) Units. The N point sequence FFT is represented using following equation [1]

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{nk}; \quad 0 \le k \le N-1 \tag{1}$$

The two different Radix 2 algorithms are Decimation in Time DIT and Decimation in Frequency DIF algorithms. In both these algorithms N inputs are divided into two N/2 sequences. In this paper we make use of DIF algorithm because of its improved accuracy and better immunity to noise. For DIF algorithm the output point's frequency is subdivided. The output obtained by this method will be in bit reversed order [1].

Radix-4 Modified Booth algorithm [2] is an efficient algorithm that multiplies two signed numbers using 2's compliment form. The number of partial products is reduced by half for this algorithm. The main bottle-neck of speed is in the addition of partial products. The critical path for the multiplier is on the number of partial products. The partial products generated are added using an area efficient carry select adder [3]. The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with Cin =1 in the regular CSLA to achieve lower area and power consumption [10]–[12]. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

II. Methodology

The most basic computational block involved in the FFT module is a butterfly diagram. The entire process involves $\log_2 N$ stages of decimation, where each stage involves N/2 butterflies of the type shown in the Fig1. below

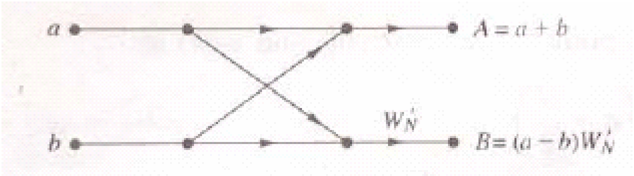


Fig1. Butterfly Diagram

The Fig2.below shows a radix-2 8-point DIF algorithm. The inputs are given by x[n] and the outputs are given as X[n]. The outputs of DIF will be in bit reversed order. It includes three stages.

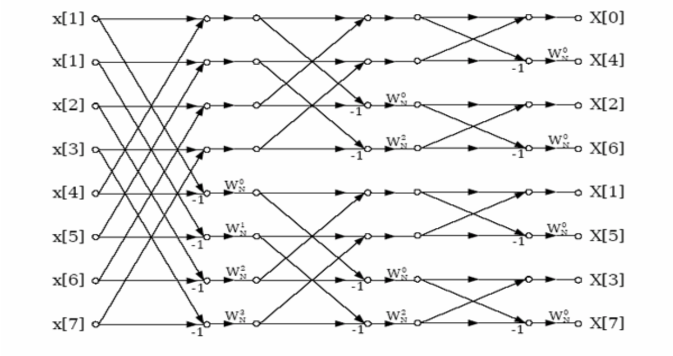


Fig2. Radix-2 8-point DIF algorithm

In the above figure, $W_N = e^{-j 2\Pi / N}$, is the Twiddle factor. Multiplication is done in two steps, generation of partial products and addition of partial products.

Modified Radix 4 Booth Multiplier Algorithm

Multiplication consists of three steps: 1) the first step to generate the partial products; 2) the second step to add the generated partial products until the last two rows are remained; 3) the third step to compute the final multiplication results by adding the last two rows. The modified Booth algorithm reduces the number of partial products by half in the first step. We used the modified Booth encoding (MBE) scheme proposed in [2]. It is known as the most efficient Booth encoding and decoding scheme. To multiply X by Y using the modified Booth algorithm starts from grouping Y by three bits and encoding into one of {-2, -1, 0, 1, 2}Table I shows the rules to generate the encoded signals by MBE scheme. The partial products generated by the modified Booth algorithm are added in parallel using the Wallace tree [1] until the last two rows are remained. The final multiplication results are generated by adding the last two rows.

Xi + 1	Xi	Xi – 1	Operation
0	0	0	0 x Y
0	0	1	1 x Y
0	1	0	1 x Y
0	1	1	2 x Y
1	0	0	-2 x Y
1	0	1	-1 x Y
1	1	0	-1 x Y
1	1	1	0 x Y

Table1. Modified Booth Encoding Table

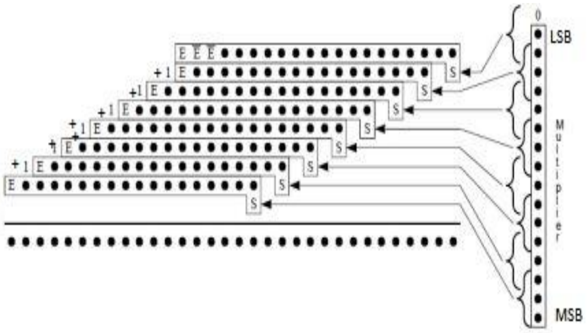


Fig3. Generated Partial Product Scheme

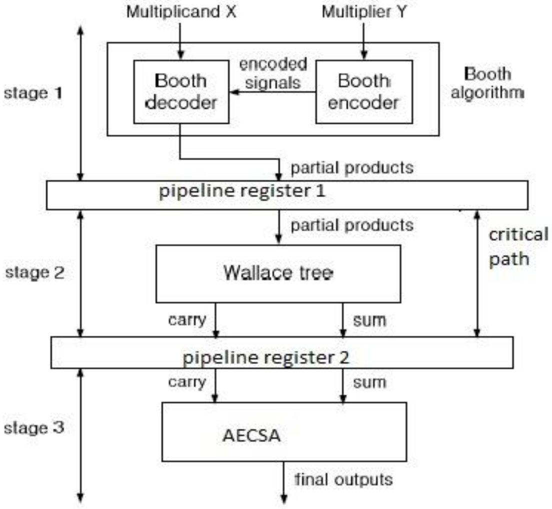


Fig4. Architecture of Modified Booth Multiplier

Fig. 4 shows the architecture of the commonly used modified Booth multiplier. The inputs of the multiplier are multiplicand X and multiplier Y. The Booth encoder encodes input Y and derives the encoded signals and the Booth decoder generates the partial products using the encoded signals and the other input X. The Wallace tree computes the last two rows by adding the generated partial products. The last two rows are added to generate the final multiplication results using an area efficient carry select adder (AECSA).

Area Efficient Carry Select Adder (AECSA)

Carry Select Adder (CSLA) is one of the fastest adders used in many data-processing processors to perform fast arithmetic functions. AECSA is an efficient gate-level modification to significantly reduce the area and power of the CSLA. The delay obtained by this technique will be slightly higher than that of conventional CSLA due to the use of excess one converter since the excess one value will only be calculated after the first sum is generated.

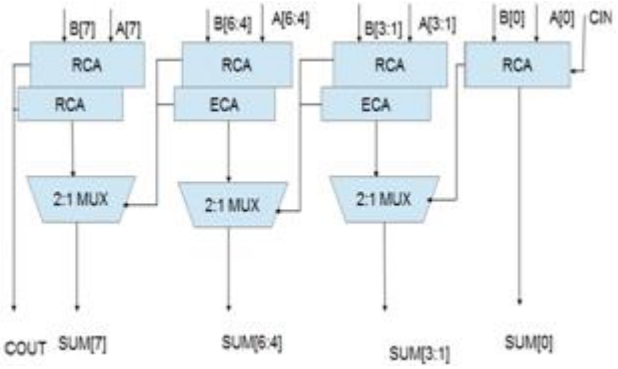
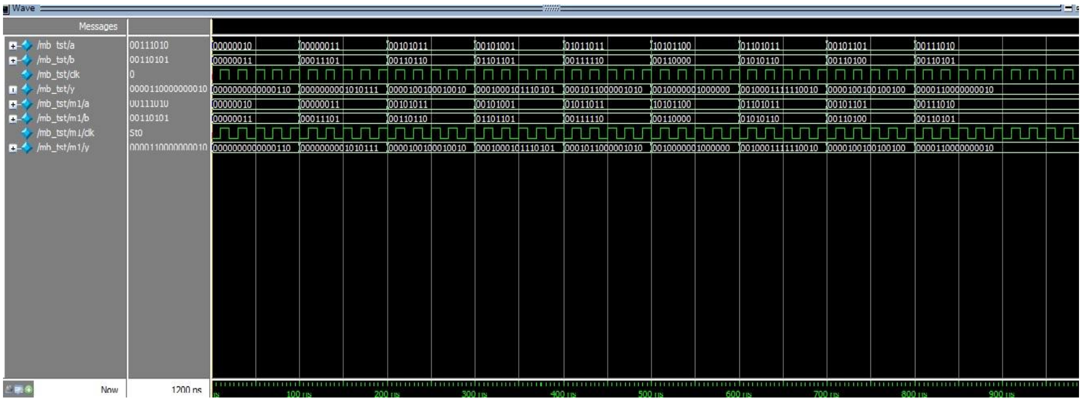


Fig5. AECSA

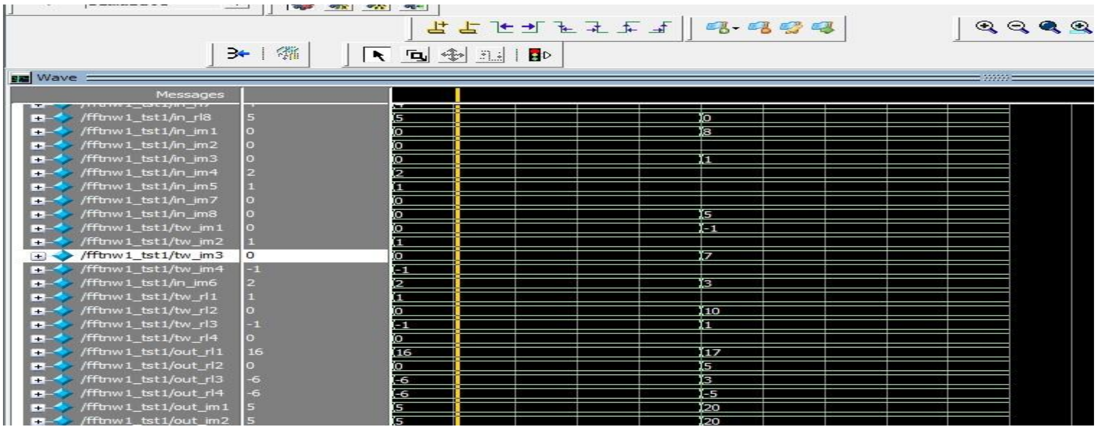
III. Results and Discussions

MAC unit plays a major role in today’s applications. The major aim of MAC unit is to provide high performance and also to reduce the area overhead in the design. The results show that the design provides a high speed along with reduction in area.



```
Messages
/mb_tst/a          00111010
/mb_tst/b          00110101
/mb_tst/clk        0
/mb_tst/y          0000110000000010
/mb_tst/m1/a       00111010
/mb_tst/m1/b       00110101
/mb_tst/m1/clk     St0
/mb_tst/m1/y       0000110000000010
Now                1200 ns
```

Fig6. Waveform of mac unit



```
Messages
/fftnw1_tst1/in_rl8     5
/fftnw1_tst1/in_im1     0
/fftnw1_tst1/in_im2     0
/fftnw1_tst1/in_im3     0
/fftnw1_tst1/in_im4     2
/fftnw1_tst1/in_im5     1
/fftnw1_tst1/in_im7     0
/fftnw1_tst1/in_im8     0
/fftnw1_tst1/tw_im1     0
/fftnw1_tst1/tw_im2     1
/fftnw1_tst1/tw_im3     0
/fftnw1_tst1/tw_im4     -1
/fftnw1_tst1/in_im6     2
/fftnw1_tst1/tw_rl1     1
/fftnw1_tst1/tw_rl2     0
/fftnw1_tst1/tw_rl3     -1
/fftnw1_tst1/tw_rl4     0
/fftnw1_tst1/out_rl1    16
/fftnw1_tst1/out_rl2    0
/fftnw1_tst1/out_rl3    -6
/fftnw1_tst1/out_rl4    -6
/fftnw1_tst1/out_im1    5
/fftnw1_tst1/out_im2    5
```

Fig7. Waveform of fft

The synthesis of the design is done using RTL Compiler tool from Cadence. The synthesis is carried out for 45nm technology and the reports show that the design is area and speed optimized. The comparison of area and delay for the different adder architectures are given in the table below

Adder	Delay	Area (μm^2)
CSLA[3]	1.719	991
AECSA	1.879	884

Table2. Comparison of AECSA with CSLA

IV. Conclusion

The proposed paper implements a high performance FFT processor that is both area as well as speed optimized. The area can be effectively reduced by the use of an area efficient CSA [AECSA] only with a slight reduction in speed. This adder unit uses BEC in place of RCA as compared to a normal carry select look ahead adder. The BEC unit consists of lesser number of logic gates and as a result reduces the area of the design.

REFERENCES

[1] D. G Manolakis and J .G Proakis , Digital Signal Processing , Principles, Algorithms, and applications, Prentice Hall (India) Publications, 3 rd Edition -1998.

[2] M Ozaki, Y. Adachi, Y. Iwahori, and N. Ishii, Application of fuzzy theory to writer recognition of Chinese characters, International Journal of Modelling and Simulation, 18(2), 1998, 112-116.

[3] B. Ramkumar and Harish M Kittur, Low-Power and Area-Efficient Carry Select Adder, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 20, No. 2, February 2012.

[4] Soojin Kim and Kyeongsoon Cho , Design of High-speed Modified Booth Multipliers Operating at GHz Ranges, World Academy of Science, Engineering and Technology 37, 2010.

[5] A. Abdelgawad, Magdy Bayoumi, High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications, IEEE 2007.

[6] F. Elguibaly, A fast parallel multiplier-accumulator using the modified Booth algorithm, IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, pp. 902-098, Sept.2000.

[7] H. Murakami, et al. A multiplier-accumulator macro for a 45 MIPS embedded RISC processor, IEEE J. Solid-State Circuits, vol. 31, pp.1067-1071, July 1996.

[8] Ayman Fayed, Walid Elgharbawy, and Magdy Bayoumi, A merged multiply accumulate for high-speed signal processing application, ICASSP IEEE 2004.

[9] Ayman Fayed , Walid Elgharbawy, and Magdy Bayoumi, A data merging technique for high speed low- power multiply accumulate units, IEEE 2002.

[10] B. Ramkumar, H.M. Kittur, and P. M. Kannan, ASIC implementation of modified faster carry save adder, Eur. J. Sci. Res., vol. 42, no. 1, pp.53–58, 2010.

[11] T. Y. Ceiang and M. J. Hsiao, Carry-select adder using single ripple carry adder, Electron. Lett., vol. 34, no. 22, pp. 2101–2103, Oct. 1998.

[12] Y. Kim and L.-S. Kim, 64-bit carry-select adder with reduced area,Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.