

FFT Implementation using Modified Booth Multiplier and CLA



Senoj Joseph, I. Shyam, K. Salai Mathiazhagan, R. Vishnu

Abstract: In the field of digital signal and image processing the Fast Fourier Transform (FFT) is one of the rudimentary operations. Telecommunication, Automotive, Hearing devices, Voice recognition systems are some of the applications of Fast Fourier Transform. DFT is implemented using FFT which is a type of algorithm that computes DFT in a fast and efficient manner. This project concentrates on the development of the Fast Fourier Transform (FFT), based on Decimation In Time (DIT) domain, Radix2 algorithm, using VHDL as a design entity. The objective of this project is to establish an efficient design that computes FFT in a faster way. In this project FFT is implemented using modified booth multiplier and CLA and simulated on Xilinx ISE.

Keywords: Modified Booth Multiplier, CLA, FFT, Wallace Tree dder

I. INTRODUCTION

A huge number of digital data has to be processed at extreme speed in many applications of DSP. In operations like filtering, convolution in DSP the multiplier plays an important role. The multiplier is also the most needed tool in microprocessor. For the processing of high speed data fast processors are essential and most important. Among the other blocks in the digital system the multiplier is the one which takes more time to process and thus the overall delay is affected by the multiplier. Applications like the Discrete Fourier Transform (DFT) is one of the most utilized tools in DSP. It is used for analyzing and designing systems in the frequency domain.

$$X(k) = \sum_{n=0}^{N-1} (x(n) e^{-j(\frac{2\pi}{N})kn})$$

The above equation is the general equation of DFT. This equation requires a greater number of additions and multiplications which makes the DFT a little complex algorithm. To avoid these complexities, we go for FFT[7]. FFT (Fast Fourier Transform) is the type of DFT computing methods to decrease the delay caused by standard DFT. The FFT uses a completely different algorithm to convert the given signals from its original domain to frequency domain. The FFT uses the factorization method to transform the given signals.

Revised Manuscript Received on February 05, 2020.

* Correspondence Author

Dr. Senoj Joseph*, Associate professor, Dept. of Electronics and Communication Engineering, Sri Krishna College of Technology, Coimbatore, India.

I. Shyam, Dept. of Electronics and Communication Engineering, Sri Krishna College of Technology, Coimbatore, India.

K. Salai Mathiazhagan, Dept. of Electronics and Communication Engineering, Sri Krishna College of Technology, Coimbatore, India.

R. Vishnu, Dept. of Electronics and Communication Engineering, Sri Krishna College of Technology, Coimbatore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

It literally halves the number of computations and thus increases the speed of the system especially when the input is of long data set. And in the FFT implementation we will be using some multipliers and adders.

It is those multipliers and adders which also influences the performance of the system. The nature of the multipliers plays a major role in the functioning of the system. So, we decided to use the modified booth multiplier which accepts both signed and unsigned inputs, and gives less delay and accurate output. Several efficient multipliers like Vedic multipliers have been considered but it could compute only unsigned inputs and thus a modified booth multiplier is used to compute FFT.

II. OVERVIEW

The Fast Fourier Transform [7] works on the radix-2 algorithm which is often denoted as butterfly module. The multiplication is implemented by the modified booth multiplier and addition by CLA.

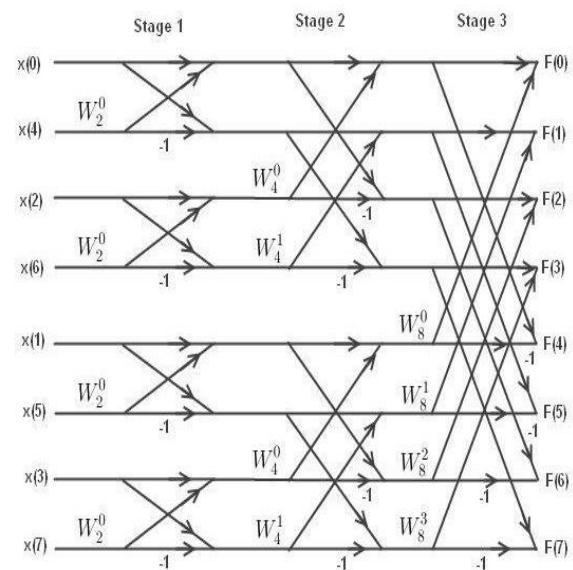


Fig.1 Butterfly diagram

In the normal booth multiplier, the multiplication between two signed binary numbers occurs in two's complement notation. Here the multiplication occurs between the adjacent bits of the N-bit multiplier. After each multiplication the output must be added with its corresponding adjacent output which generates partial products. The number of partial products is higher in normal booth multiplier and thus it requires large number of transistors and area.



FFT Implementation using Modified Booth Multiplier and CLA

Eventually the delay increases with these factors. The Modified Booth Multiplier helps to reduce the delay, number of transistors required by reducing the number of partial products generated [1]. The partial products are generated by shifting, adding every column of the multiplier term and multiplying it with appropriate bits. In the modified booth multiplier, the partial products are generated not for each column instead we consider only the second column thus reducing the number of computations, this method is called as Radix-4 booth recoding. In the radix-4 method we compare three bits at a time unlike booth multiplier [6].

III. PROPOSED WORK

The FFT [3] is implemented using Modified Booth Multiplier and CLA. In FFT there are maximum number of multiplications and additions which requires an efficient multiplier and thus modified booth multiplier fills that space. We have developed HDL code for booth multiplier and CLA and since it was consuming a large area as well as delay, the modified booth multiplier is implemented with CLA [6].

IV. SYSTEM DESIGN

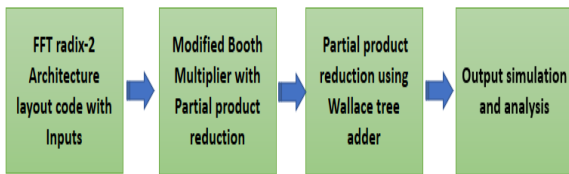


Fig.2 Block Diagram

The block diagram of the HDL code is shown in Fig.2. The design mainly comprises HDL modules written in VHDL language. The top layout consists of the VHDL code developed for the FFT module. The operation of the FFT is explained in the butterfly diagram. The 3-point FFT is implemented by incorporating 3 Stages with inputs propagating from stage 1. The addition process is carried out by CLA (Carry Look Ahead Adder) and the multiplication is processed using a modified booth multiplier. The booth multiplier module consists of partial product generation, encoding and full adder modules. The modified booth multiplier consists of same as that of the booth multiplier but the partial product reduction module is added at the end which incorporates, Wallace tree adder [4] which is an efficient process for grouping the terms in the encoding process as 3 instead of 2, is used as a separate module. The CLA architecture comprises of full adders and components relevant to the operation. The final FFT output is simulated and various analysis including area, power and delay analysis is done using Xilinx ISE.

V. MODIFIED BOOTH MULTIPLIER DESIGN

The modified booth multiplier diagram is shown in Fig.3. Partial products generation stage, partial products addition stage, and the final addition stage are the different stages which every multiplier consists of. The modified booth algorithm [1] also called the fastest algorithm can be used to reduce the number of partial products generated.

Wallace Tree Adder structures have been used to sum the partial products in reduced time. The aim of ours is to minimize the computation time by using booth algorithm [2] for multiplication. It is totally comprised of concealing the partial products and multiplier bits. Different modified booth algorithms have been proposed according to how many numbers of bits are used to encode the multiplier [5]. The number of partial products is reduced. In the proposed modified Radix -4 Booth Algorithm, multiplier has been divided in groups of 3 bits and each group of 3 bits have been considered according to modified Booth Algorithm for generation of partial product [7]. For the production of partial products of only two rows which can be included in the final stages Wallace tree technique is utilized by high speed designs. Half adder, full adder, unit adder are employed in Wallace tree structure [4] to reduce partial products. The accumulation of the partial products is improved by the Wallace tree. The speed, area and power consumption [2] of the multipliers will be in direct proportion to the efficiency of the Accumulation rate.

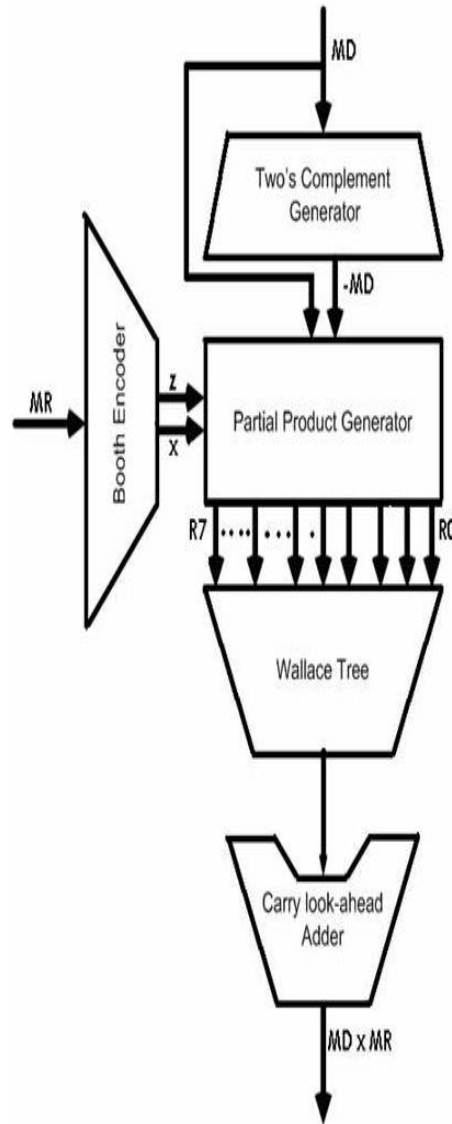


Fig. 3 Modified booth Multiplier

VI. RESULT & DISCUSSION

The FFT implementation is done using modified booth multiplier and CLA. The analysis is done using cadence and certain important parameters like area, power, delay has been measured successfully and it is inferred that modified booth multiplier is the efficient one to be used on FFT. The overall performance of the system is efficient when compared to other architectures Fig.4 is the simulation output window consisting of FFT output corresponding to its inputs. Table.1 is the analysis of the parameters respective to their units.

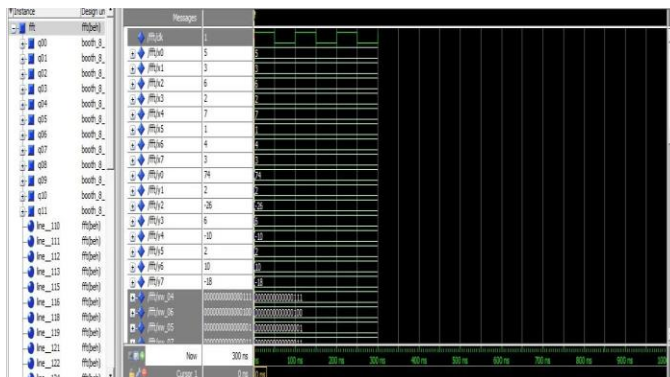


Fig.4 Simulation

Table.1

FACTORS TO BE NOTED	MOD. BOOTH MULTIPLIER IN FFT
Delay (ns)	8.481
No of gates	9496
Power (mW)	119

VII. CONCLUSION

High speed, less area and efficient FFT architecture is proposed using modified booth multiplier. This architecture is 30% more efficient in terms of gate count (area) and 2.2% more efficient in terms of delay compared to that of normal booth multiplier implemented on FFT thus it proves to be a systematic design that can be used in many applications involving DSP operations and the demand of high speed DSP based processors can be compensated using this proposed architecture. Table.2 shows the comparison of the basic design parameters between Modified and Normal Booth multiplier implemented on FFT.

Table.2

Factors	Modified Booth Multiplier	Normal Booth Multiplier
Delay (Ns)	8.481	8.67
No. Of Gates	9496	12,324
Power(Mw)	119	142

REFERENCE

- Li-Rong Wang, Shyh-Jye Jou, Chung-Len Lee. "A well-structured modified booth multiplier design". 2018 IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT).
- Suganthi Venkatachalam, Hyuk Jae Lee, Seok-Bum Ko. "Power Efficient Approximate Booth Multiplier". 2018. IEEE International Symposium on Circuits and Systems (ISCAS).
- Lo Sing Cheng, Ali Miri, Tet Hin Yeap. "Efficient FPGA Implementation of FFT based Multipliers" 2015. Canadian Conference on Electrical and Computer Engineering, 2015.
- Neeta Sharma, Ravi Sindal, "Modified Booth Multiplier using Wallace Structure and Efficient Carry Select Adder" 2017. International Journal of Computer Applications (0975 -8887) Volume 68-No.13, April 2017
- Divya Govekar, Dr Ameeta Amonkar, "Design and Implementation of High Speed Modified Booth Multiplier using Hybrid Adder" 2017. International Conference on Computing Methodologies and Communication (ICCMC).
- Chengdong Liang, Lijuan Su, Jinzhao Wu, Juxia Xiong, "An Innovative Booth Algorithm". Conference: 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC).
- https://en.wikipedia.org/wiki/Fast_Fourier_transform

AUTHORS PROFILE



Dr. Senoj Joseph, has completed his M.Sc (Electronics) from Cochin University of Science and Technology in the year 1999, M.Tech (Electronics) from Cochin University of Science and Technology in the year 2001 and Ph.D in the field of I & C from Anna University in the year 2016. He has about 17 years of teaching experience and presently working as an Associate Professor in Electronics and Communication Engineering Department, Sri Krishna College of Technology, Coimbatore. He has published 10 papers at International journals and 11 Conferences. His research interest includes VLSI and Image Processing.



I. Shyam, UG student pursuing final year BE ECE (Electronics and Communication Engineering) at Sri Krishna College of Technology, Coimbatore. His research interest includes VLSI, Low Power VLSI Design.



K. Salai Mathiazhagan, UG student pursuing final year BE ECE (Electronics and Communication Engineering) at Sri Krishna College of Technology, Coimbatore. His research interest includes VLSI, Low Power VLSI Design.



R. Vishnu, UG student pursuing final year BE ECE (Electronics and Communication Engineering) at Sri Krishna College of Technology, Coimbatore. His research interest includes VLSI, Low Power VLSI Design, ASIC Design.