VLSI Implementation of High Speed and High-Resolution FFT Algorithm Based on Radix 2 for DSP Application

Nooshin Mahdavi, Rozita Teymourzadeh, Masuri Bin Othman

VLSI Implementation of High Speed and High Resolution FFT Algorithm Based on Radix 2 for DSP Application

N. Mahdavi, R. Teymourzadeh, *IEEE Student Member*, Masuri Bin Othman

***Abstract*-- Using Fast Fourier Transform (FFT) is indispensable in most signal processing applications. Designing an appropriate algorithm for the implementation of FFT can be efficacious in digital signal processing. Sophisticated techniques such as pipelining and parallel calculations have potential impacts on VLSI implementation of FFT algorithm. Furthermore, a mathematic approach such as floating point calculation achieves higher precision. In this paper, an efficient algorithm with using parallel and pipelining methods is proposed to implement high speed and high resolution FFT algorithm. Latency reduction is an important issue to implement the high speed FFT on FPGA. The Proposed FFT algorithm shows the latency of 5131 clock pulse when N refers to 1024 points. The design has the mean squared error (MSE) of 0.0001 which is preferable to Radix 2 FFT.**

***Index Terms*—Radix, FFT, Butterfly, VLSI, Floating point.**

I. Introduction

Fast Fourier transform is a faster version of DFT which was invented by Cooley & Tuky in 1965 [1]. There are several types of FFT algorithms. Radix-2 is a practical algorithm to calculate FFT by using decimation in time (DIT) method.

The main idea of decimation in time is to sunder the input string of h_n with length of $N = 2^v$ into two substrings with odd and even indexes. The N-point DFT of h_n can be achieved by suitable combining to $N/2$-point DFT of each substring. Each one of them can be reduced to $N/4$-point DFTs. This procedure will continue till only 2-point DFTs remains [2]. The final result is the formation of FFT algorithm. In 2006 petrovsky [3] implemented Radix 2-4 based on parallel-pipeline FFT-processors at structural level. He achieved 0.00201 MSE on his FFT algorithm. This paper shows the implementation of high speed and high resolution FFT based on parallel pipeline and floating point Radix-2 with the minimum MSE of 0.0001.

N. Mahdavi is with the Department of Electrical Engineering, Arak University, Iran (e-mail: mahdavi_n2@yahoo.com).

R. Teymourzadeh is with Department of Electrical Engineering, Bangi, Malaysia (e-mail: rozita60@vlsi.eng.ukm.my).

Masuri Bin Othman is with Department of Electrical Engineering, Bangi, Malaysia (e-mail: masuri@vlsi.eng.ukm.my).

The next section describes the principle of the FFT structure. The mathematical formulation and block diagram of pipeline FFT algorithm is described in section III. Section IV shows implementation and design result in brief. Finally, conclusion is expressed in section V.

II. Principle of the FFT Structure

The computation of N-point FFT via decimation-in-time algorithm requires $O(N \log_2 N)$ complex multiplication time [7] and it will be increased if basic butterfly computation is used for adder and multiplication in decimation-in-time FFT algorithm. Additionally a complicated controller is also required. Furthermore, implementing fixed point FFT algorithm, results the output differs significantly in comparison to the expected output.

To obtain the high speed and high resolution FFT algorithm, implementation of the floating point pipeline FFT is applied.

The proposed method shows high performance of the FFT algorithm. The principle architecture is based on using a memory to keep input and output data. Thus, it reduces hard ware complexity. All the blocks are designed to operate with the same clock frequency. These stages implement as pipeline and parallel to reduce FFT calculations time.

III. FFT Block Diagram Description

Figure 1 shows the main block diagram of the FFT algorithm. Each block will be investigated separately.

A. Radix-2 Butterfly Algorithm

First, the RAM is initialized by external microprocessor and the data are loaded to the RAM by bit reserve address pin. The Radix 2 Butterfly block is the primary prefatory step taken in DIT FFT [8,9]. This block calculates the complex number in equation 1 and 2 as follow.

$$output1 = input1 + W^k \times input2 \qquad (1)$$

$$output2 = input1 - W^k \times input2 \qquad (2)$$

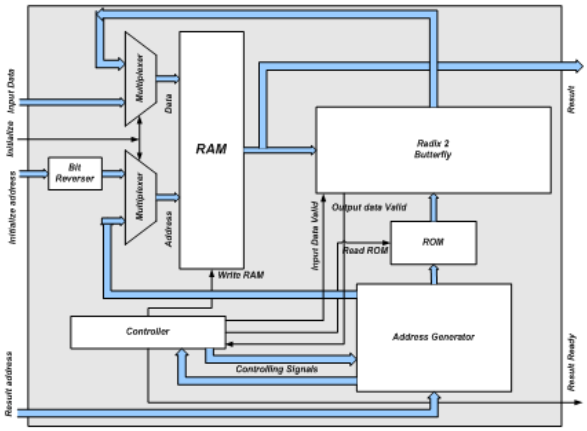

Fig. 1. The FFT block diagram

Hence, one complex multiplication and two complex additions are required in this calculation. If $output1 = X_{01} + iY_{o1}$, $output2 = X_{o2} + iY_{o2}$ and $W^k = X_{w^k} + iY_{W^k}$ we have:

$$X_{o1} + iY_{o1} = \left(X_{i1} + \left(X_{W^k} \times X_{i2} - Y_{W^k} \times Y_{i2}\right)\right) + i\left(Y_{i1} + \left(X_{W^k} \times Y_{i2} + Y_{W^k} \times X_{i2}\right)\right) \quad (3)$$

$$X_{o2} + iY_{o2} = \left(X_{i1} - \left(X_{W^k} \times X_{i2} - Y_{W^k} \times Y_{i2}\right)\right) + i\left(Y_{i1} - \left(X_{W^k} \times Y_{i2} + Y_{W^k} \times X_{i2}\right)\right) \quad (4)$$

Thus, each butterfly requires four real multiplications and six real additions. If the above equations are implemented using fixed point calculations, the error possibility of the result will increase. These errors [4] consist of round-off, overflow and coefficient quantization errors. To reduce the errors and provide high resolution, floating point adder/subtractor is replaced. This adder/subtractor operates in 3 stages including mantis alignment, mantis addition/ subtraction and normalization. Thus, each of the above phases can be used as stages for pipeline implementation. We convert the first phase into two stages in order to achieve higher clock frequency [5], [6]. Figures 2 and 3 show the block diagram of the pipeline adder/subtractor and multiplier. It is not necessary to adjust the exponents in multiplier. Since the normalized inputs are loaded to the multiplier, the mantises are multiplied and the exponents are added and amended. The result may be shifted one bit leftward. Thus the exponent will be decremented one unit.

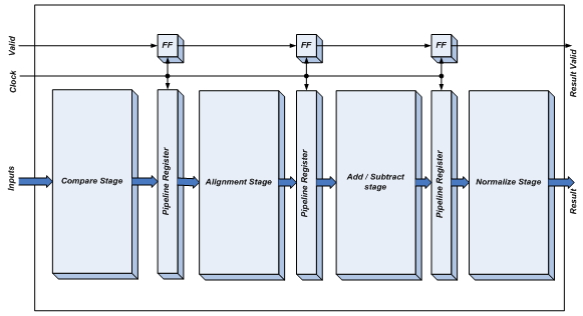

Fig. 2. Pipeline floating point adder/ subtractor

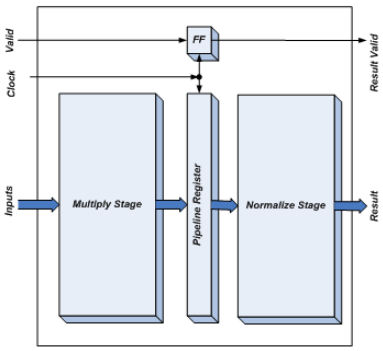

Fig. 3. Pipeline floating point multiplier

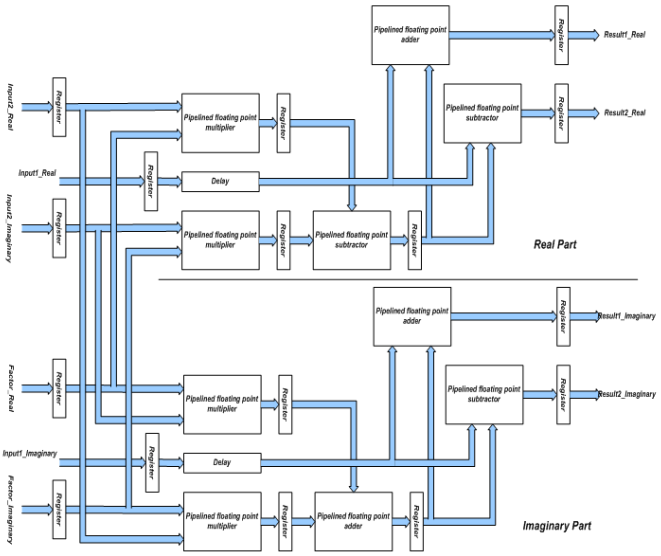

Fig. 4. Internal schematic of the pipeline butterfly algorithm

Figure 4 shows the internal schematic of the pipeline butterfly algorithm. To improve speed calculations in the Radix-2 butterfly algorithm, the pipeline registers are located after each addition, subtraction and multiplication blocks. Hence, the pipeline butterfly algorithm keeps the final result in the register to transfer it to the RAM by the next clock cycle. This FFT algorithm is exactly executed after $\left(N/2\,Log_2 N\right)+11$ clock pulses. The 11 clock pulses delay is created by 11 pipeline registers in adder, subtractor and multiplier in a serial butterfly block. Additionally, parallel design of the FFT algorithm decreases the calculation time significantly.

B. ROM and RAM Blocks

The ROM in Radix-2 FFT algorithm has two look-up tables to save W^k coefficients. These coefficients are inherently complex numbers so the real & imaginary parts are separately kept on that. Since the amplitude of the sine and cosine are the same in the four quarters, (they only differ in the signs) W^k are calculated just for $k = 0$ to $k = N/4$ and related sign are saved for $k = 0$ to N into the ROM.

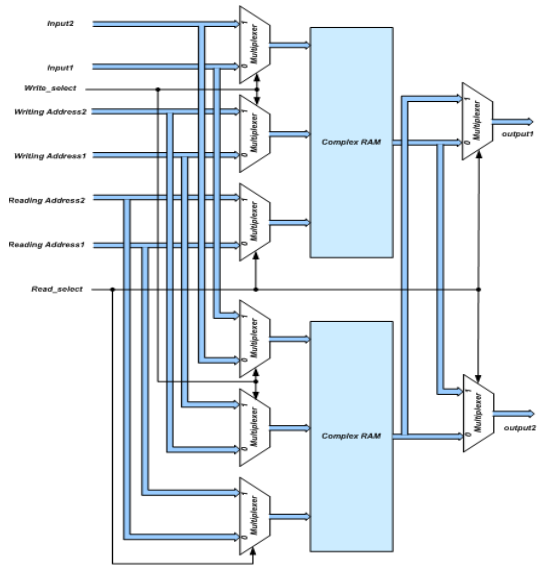

Fig. 5. Internal structure of the RAM

Thus, it avoids using a large number of gates in the FPGA board.

There is a RAM in the FFT block diagram. According to the system workflow, two complex data must be read from the RAM simultaneously and loaded to the butterfly block. Meanwhile, the two outputs of the butterfly block have to be written in the RAM at the same time. Figure 5 shows the internal structure of the RAM with the capability of reading and writing the two complex data simultaneously.

IV. Implementation Result

The new architecture of the Radix-2 FFT algorithm Verilog code was written and simulated by Matlab software. The design code is downloaded to the Virtax-2 FPGA board. From Xilinx ISE synthesize report, it was shown minimum clock period is 9.94 ns (Maximum Frequency is 100.6 MHz).
The chip layout on Virtex 2 FPGA board is shown in Figure 6.

Figure 7 shows the output digital signal of the new Radix-2 FFT algorithm which simulated by Modelsim software.

Figure 8 shows the measured signal spectrum achieved by Radix-2 FFT algorithm for 1024 floating point complex data.

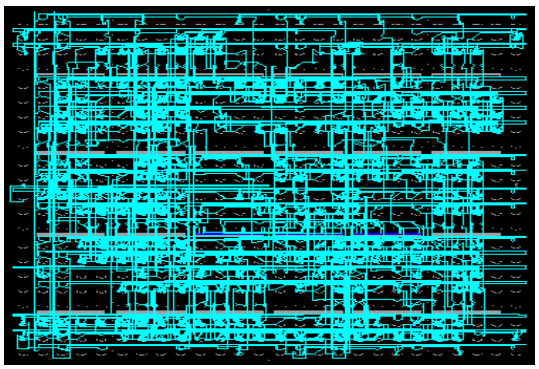

Fig. 6. The core layout on FPGA board

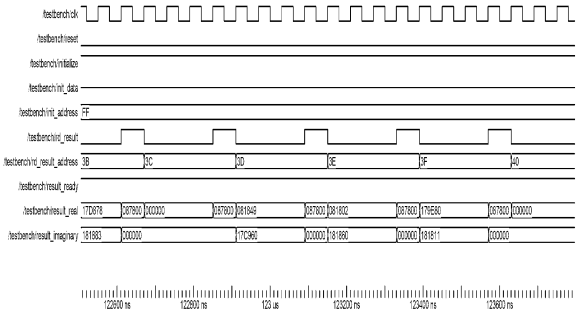

Fig. 7. The output digital signal of the system

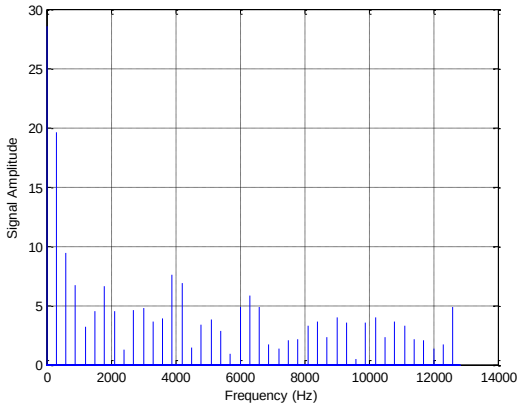

Fig. 8. Signal spectrum of high speed radix-2 FFT algorithm

The output result appeared after 5131 clock pulses which proved the complex calculation of $(N/2\,Log_2 N)+11$ in the system. The System MSE was measured by Matlab software. It shows the mean squared error (MSE) of 0.0001.

V. Conclusion

The new architecture of the high speed and high resolution of the parallel, pipeline and floating point Radix-2 FFT algorithm was designed and investigated. High Speed FFT architecture was obtained by two methods. The pipeline structure and parallel design lead us to have high speed FFT algorithm. Additionally, using a internal RAM makes the design compatible with different type of FPGA board. The implementation result shows the maximum throughput of the 100.6 MHz in Virtex 2 FPGA board. Meanwhile, the resolution was increased by floating point calculation during the FFT process. The proposed FFT algorithm proves the latency of $(N/2\,Log_2 N)+11$, 5131 clock pulses for $N=1024$ with the mean squared error (MSE) of 0.0001.

VI. Acknowledgment

We wish to thank our friend, Dr. Mohamad Amir Amini who has assisted us for completion of this dissertation.

VII. REFERENCE

[1] J. W. Cooley, J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier series," *Math of comp* Vol. 19, pp. 297-301, 1965.

[2] A. Saidi, “Decimation-in-time frequency FFT algorithm. Acoustics, Speech, and Signal Processing” , *ICASS-94, IEEE International Conference*. Vol. 3, pp. 453-456, 1994

[3] A.A. Petrovsky, S. L. Shkredov, “ Automatic Generation of Split-Radix 2-4 Parallel-Pipeline FFT Processors: Hardware Reconfiguration and Core Optimization”, *Proceeding of the international symposium on parallel computing in Electrical engineering (PARELEC'06)*, pp.181-186, 2006.

[4] E. C. Ifeachor, B. V. Jervis, *Digital Signal Processing: A Practical Approch*. Second Editiond. Prantice Hall, 2002.

[5] M. Morris Mano, *Computer System Architectur.* Third Edition. Prantice Hall. 1992.

[6] D. A. Patterson & J. L. Hennssy. *Computer Organization & Design: the Hardware/ Software Interface.* Second Edition. Morgan Kaufman Publishere Inc. 1998.

[7] Lo sing Cheng, A. Miri, Tet Hinb Yeap, “Efficient FPGA Implementation of FFT based Multipliers”, *IEEE Conference Electrical and Computer Engineering*, pp. 1300-1303, 2005.

[8] K. S. Hemmert, K. D. Underwood, “An Analysis of The Double-Precision Floating-point FFT on FPGAs”, 13th *IEEE Symposium on Field-Programmable Custom Computing Machines,* pp. 171-180, April 2005.

[9] Shiqun Zheng, Dunshan Yu, **“**Design and implementation of a parallel real-time FFT processor”, *7th IEEE conference on Solid-State and Integrated Circuits Technology*, Vol. 3, pp. 1665-168, Oct 2004.