

# FPGA IMPLEMENTATION OF 16 BIT FFT PROCESSOR USING REVERSIBLE BASED MULTIPLEXER

L. Niteshkumar<sup>1</sup>, Assi. Prof. A. Raganna<sup>2</sup>  
<sup>1</sup>M.Tech (VLSI and ES)  
 Dept of ECE, REVA ITM, Bangalore.

**Abstract:** In present scenario every process should be rapid, efficient and simple. Fast Fourier transform (FFT) is an efficient algorithm to compute the N point DFT. It has great applications in communication, signal and image processing and instrumentation. But the Implementation of FFT requires large number of complex multiplications, so to make this process rapid and simple it's necessary for a multiplier to be fast and power efficient. To tackle this problem an efficient method of multiplication is used that is generation of partial products is done by using multiplexer easily and adding reversible logic circuitry which also lowers power consumption. The purpose of this project is Fast Fourier Transform (FFT) design methodology using proposed algorithm. By combining this approaches proposed design methodology is time-area-power efficient and combination of reversible logic gates for designing multiplexer also improves power and speed of FFT algorithm.

## I. INTRODUCTION

Multiplication is an essential arithmetic operation for common DSP applications such as Filtering, computation of FFTs etc. The dynamic power consumption (i.e., power loss due to switching activity) can be reduced by bypassing technique. The switching activity of the component used in the design depends on the input bit coefficient. This means that if the input bit coefficient is zero corresponding rows or column of the adders need not be activated. If the multiplicand (or the multiplier) contains more zeroes, then higher power reduction can be achieved, which is the frequent in FFT computation. Conventional combinational logic circuits are known to dissipate heat for every bit of information that is lost. This is also evident from the second law of thermodynamics which states that any irreversible process leads to loss of energy. R.Landauer[1] showed that any gate that is irreversible necessarily dissipates energy, and each irreversible bit generates  $k \cdot T \ln 2$  joules of heat where  $k$  is Boltzmann's constant ( $1.38 \times 10^{-23}$  joules/Kelvin) and  $T$  is temperature in Kelvin. C.H.Bennett[2] in 1973 showed that an irreversible computer can always be made reversible. A reversible logic gate is an N input, N output logic device with one to one mapping. This helps to determine the outputs from the inputs and also the inputs can be uniquely recovered from the outputs. Extra inputs or outputs are added so that the number of inputs is made equal to the number of outputs whenever it is necessary.. A Reversible circuit should be designed using minimum number of reversible gates. One key requirement to achieve optimization is that the designed

circuit must produce minimum number of garbage outputs. They must use minimum number of constant inputs. The gate count should also be minimum in a given reversible logic design. Edward Fredkin and Tommaso Toffoli[4] based on the concept of reversibility they introduced new reversible gates known as Fredkin and Toffoli reversible gates. These gates have zero power dissipation and are used as universal gates in the reversible circuits. Reversible logic has received great attention in the recent years due to its ability to reduce the power dissipation. Quantum arithmetic components need reversible logic circuits for their construction. Reversible logic circuits find wide application in low power digital design, quantum computing and nanotechnology. In the year 1994 Shor did a remarkable research work in creating an algorithm using reversibility for factorizing large number with better efficiency when compared to the classical computing theory. After this the work on reversible computing was started by more people in different fields such as nanotechnology, quantum computers and CMOS VLSI. Peres[5] introduced a new gate known as peres gate. Peres gate is also a 3\*3 gate but it is not a universal gate like the Fredkin and Toffoli gate. Even though this gate is not universal gate it is widely used in many application because it has less quantum cost when compared to the universal gate. The quantum cost of the Peres gate is 4. Commonly used reversible gates are NOT gate, CNOT gate, CCNOT gate (Toffoli gate), Peres gate, Fredkin gate, Feynman gate,

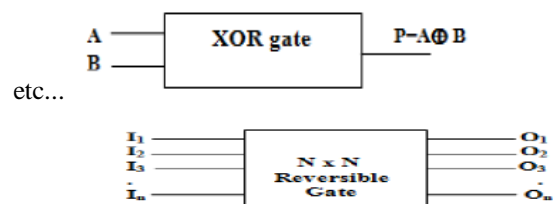


Figure 1 (a) Irreversible gate (b) NxN Reversible Logic Gate. In the reversible XOR gate there is no loss of information bit signals. Since it maps the input vector with output vector which gives the equal number of inputs and output

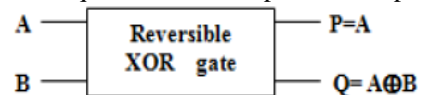


Figure 2. Reversible XOR gate

The input vector of Feynman gate is I (A, B) and the output vector is O (P, Q). The outputs are defined by  $P=A$ ,  $Q=A \oplus B$  and it is shown in Figure. Quantum cost of a Feynman gate is 1.

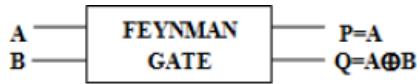


Figure 3: Feynman gate

Figure 1.4 shows a 3x3 Fredkin gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P=A, Q=A'B AC and R=A'C AB. Quantum cost of a Fredkin gate is 5.

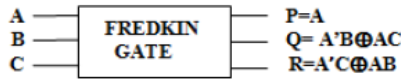


Figure 4. Fredkin gate

The 3x3 Toffoli gate is represent in Figure 1.5. The input vector is I(A, B, C) and the output vector is O(P,Q,R).The outputs are defined by P=A, Q=B, R=AB C. Quantum cost of a Toffoli gate is 5.



Figure 5. Toffoli gate.

A. Discrete Fourier Transform (DFT)

A transform that gives the frequency domain representation of a time domain sequence is termed as DFT [21].It converts time domain data to frequency domain data.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi nk}{N}} = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad k = 0,1,\dots,N-1$$

Where the Twiddle Factor is

$$W_N^{nk} = e^{-j\frac{2\pi nk}{N}} = \cos\left(\frac{2\pi nk}{N}\right) - j \sin\left(\frac{2\pi nk}{N}\right)$$

The Inverse DFT (IDFT) coverts frequency domain data to the time domain

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi nk}{N}} = \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-nk} \quad n = 0,1,\dots,N-1$$

The DFT/IDFT requires N<sup>2</sup> complex multiplies and N(N-1) complex additions Due to this we go for Fast Fourier transforms (FFT). A FFT is an efficient algorithm to compute the discrete Fourier transform (DFT) and it's inverse. The FFT provides a fast method for computing the DFT by breaking N-point DFT into two N/2-Point FFTs plus N/2 butterfly operations. This process continues by recursively dividing the DFT Reduces complexity of O(N<sup>2</sup>) to O(Nlog2N)..

The computation of the N point DFT by the divide-and conquer approach provides an efficient algorithm. We split the N-point data sequence into two N/2-point data sequences of f1 (n) and f2 (n), corresponding to the even numbered and odd-numbered samples of x(n), respectively, that is,  
 f1(n) = x(2n), and  
 f2(n) = x(2n+1) n = 0,1,...,N/2-1.

Thus f1 (n) and f2 (n) are obtained by decimating x (n) by a factor of 2, and the resulting FFT algorithm is called a decimation-in-time algorithm.

In present days, high speed and power efficient FFT algorithms are required for high speed digital communication systems such as OFDM and DSL systems. In present days the power efficiency is becoming major concern since it determines the size of the system as well as economy. This approach to compute FFT will not only reduce the delay but also reduces the power consumption.

B. Multiplexer

A multiplexer (or mux) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2n inputs has n select lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth. A multiplexer is also called a data selector.

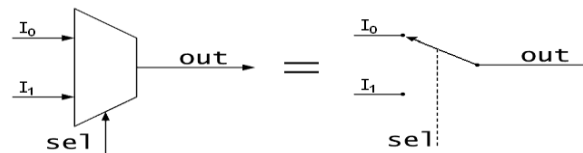


Fig.6 Basic MUX

An electronic multiplexer makes it possible for several signals to share one device or resource, for example one A/D converter or one communication line, instead of having one device per input signal.

C. Wallace tree

Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers, devised by Australian Computer Scientist Chris Wallace in 1964.

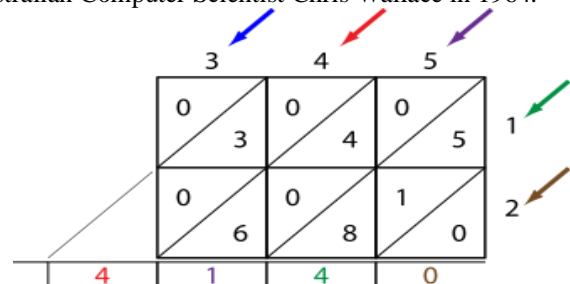


Fig.7 Basic principle of Wallace tree

The Wallace tree has three steps:

- Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding n<sup>2</sup> results. Depending on position of the multiplied bits, the wires carry different weights, for example wire of bit carrying result of a<sub>2</sub>b<sub>3</sub> is 32 (see explanation of weights below).
- Reduce the number of partial products to two by layers of full and half adders.
- Group the wires in two numbers, and add them with a conventional adder.

The second phase works as follows. As long as there are three or more wires with the same weight add a following layer:

- Take any three wires with the same weights and input them into a full adder. The result will be an output wire of the same weight and an output wire with a higher weight for each three input wires.
- If there are two wires of the same weight left, input them into a half adder.
- If there is just one wire left, connect it to the next layer.

II. PROPOSED METHODOLOGY

The Main Objective of the Project is to design an efficient multiplication which involves in implementation of several techniques and comparing its performance and choosing the efficient and emerging new concept for multiplication using multiplexer. The Implementation of Multiplication involves generation of partial products which can be easily designed by using multiplexer which consumes less power and area. The design description is given as follows consider A0A1 = 10 and B0B1 = 01 Using Multiplexer the partial products can be designed as in figure

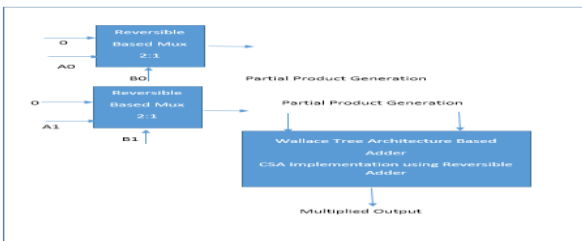


Figure 8. multiplexer and adder using reversible logic.

The required Functional Blocks are designed based on the reversible logic using reversible gates.

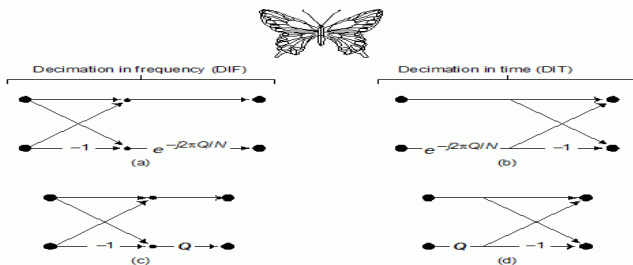


Figure 9. The twiddle factor computation.

After the generation of partial products the all partial products can be added by using Wallace tree multiplication. The Designed Multiplication is used for FFT Computation which involves in multiplication of twiddle factor which can be computed with floating point multiplier using multiplexer logic.

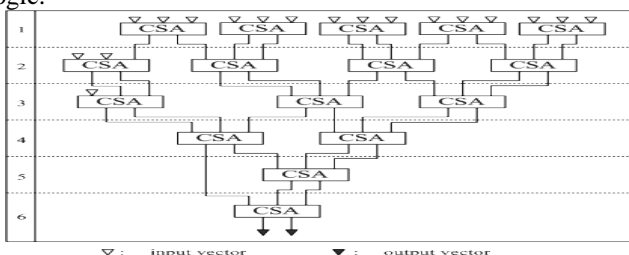


Figure 10. Wallace Tree based Addition

The Generated partial products using multiplexer can be added by using Wallace tree based to reduce the adding delay. The Multiplication algorithm can be designed by using the proposed methodology and to increase the efficiency with respect to power and speed reversible logic is used.

III. CONCLUSION

A faster and efficient FFT algorithm can be implemented using the reversible logic multiplexers which also reduce the power dissipation. CSA adder uses reversible logic which has low power dissipation and less delay because of computation of partial product is done in less clock cycle. The results of this FFT algorithm is can compared with conventional algorithms.

REFERENCE

- [1] R. Landauer, "Irreversibility and Heat Generation in the Com-putational Process", IBM Journal of Research and Development, 5, pp. 183-191, 1961.
- [2] C.H. Bennett, "Logical Reversibility of Computation", IBM J. Research and Development, pp. 525-532, November 1973.
- [3] R. Feynman, "Quantum Mechanical Computers", Optic News, Vol. 11, pp. 11-20, 1985
- [4] E. Fredkin, T. Toffoli, "Conservative Logic", Intl. Journal of The-oretical Physics, pp. 219-253, 1982.
- [5] Peres, "Reversible Logic and Quantum Computers", Physical review A, 32:3266- 3276, 1985
- [6] T. Toffoli, "Reversible Computing", Tech Memo MIT/LCS/TM-151, MIT Lab for Comp. Sci., 1980
- [7] International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013 Synthesis of Reversible Multiplexers Sumit Gugnani, Arvind Kumar
- [8] Synthesis of Reversible Multiplexers, Sumit Gugnani, Arvind Kumar, International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013 ISSN 2229-5518