Literature Review on Multiplier Accumulation Unit by Using Hybrid Adder

Amiya Prakash
M.E. Scholar, Department of (ECE)
NITTTR Chandigarh, Punjab

Dr. Kanika Sharma
Assistant Prof. Department of (ECE)
NITTTR Chandigarh, Punjab

Abstract – In present day reversible logic gates is generally used instead of conventional logic gates. Because conventional logic transmit numbers of garbage bits in the circuits. The reversible logic gate algorithm will reduce the garbage bits and logical components during arithmetic manipulations. Fast adders and multipliers are the essential part of the digital signal processing system. The speed of multiplier and adder operation is of great importance in digital signal processing as well as area is also great importance to design a MAC unit in digital signal processing. The proposed algorithm maximally decreases function complexity during synthesis steps.

Keywords – Reversible logic gates; hybrid adders; multipliers and accumulator.

I. INTRODUCTION (MAC)

In today's world, the majority of digital signal system design applications, the critical operations usually involve many multiplication and accumulations. For the real time data processing applications such as Real Time Gross settlement (RTGS) system in banking for transfer money from one bank to another, a high speed and high throughput multiplier accumulation is always a key to achieve high performance digital signal processing system. The main goal of the MAC design is to enhance its speed; this is because the real time machine is always concern with speed and throughput rate of the digital system. The speed of the multiplication and addition determines the execution speed and performance of the entire calculation. Many of the digital signals processing application are accomplished by repetitive multiplication and addition operations. Therefore multiplier-and-accumulator (MAC) unit is the essential element of the digital signal processor. In order to increase the speed of a multiplier, the number of the partial products generated must be reduced. If N-bit data are multiplied, the number of the generated partial products is propositional to N, thus the execution time. The accumulation operation has the largest delay in MAC, an architecture that uses modified booth algorithm and hybrid carry save adder is proposed. A brief description of MAC unit and its operation is introduced. In general, MAC unit consists of multiplier and an accumulation unit. Multiplier performs multiplication operation between multiplicand and multiplier where adder adds the Multiplier result to the contents of the accumulator. This process of multiplication and accumulation continues to operate until generation of final result, that itself stored in the accumulator. The number of clock cycles of the operation depends on the number of partial products generated during the operation. In most of computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of optional representation. The speed of the multiplication and addition determines the execution speed and performance of the entire calculation. Many of the digital signals processing application are accomplished by repetitive multiplication and addition operations. Therefore multiplier-and-accumulator (MAC) unit is the essential element of the digital signal processor. In order to increase the speed of the multiplier and adders, the number of the partial products generated must be reduced. If N-bit data are multiplied, the number of the generated partial products is propositional to N, thus the execution time. The accumulation operation has the largest delay in MAC, an architecture that uses reversible logic gates algorithm and a reversible hybrid adder is proposed [1].

II. OVERVIEW OF MAC

The research on multiplier accumulation unit using hybrid adder is being pursued towards both design and synthesis. In the synthesis of multiplier accumulation unit, reversible logic gates have an important role for reducing the circuit complexity of the MAC unit. The input value A and B applied at the multiplier unit and the result of the multiplier is used for one input of the adder. The final result consists of higher order bits Z [2N-1: N] that are generated by adding Sum S and Carry C in the final adder and lower order bits Z [N-1: 0] that are already generated. The hardware architecture of this MAC is shown in figure 1. It executes the multiplication operation by multiplying the input multiplier A and multiplicand B. This is added to the previous multiplication result Y and another input X as the accumulation steps. One of the most advanced types of

MAC for general-purpose digital signal processing is to be proposed. While it has a better performance because of the reduce circuit complexity as well as power consumption as compared to the conventional logic gates MAC [2]. The computations of multiplication and accumulation are combined and a hybrid-type multiplier and adder are proposed to reduce the delays of logical operations. In the majority of digital signal processing applications the critical operations usually involved many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput multiplier-accumulator (MAC) is always a key to achieve a high performance digital signal processing system. In the last few years, the main consideration of MAC design is to enhance its speed. This is because; speed and throughput rate is always concern of digital signal processing system. But for the epoch of personal communication, low power design also becomes another main design consideration. This is because; battery energy available for these portable products limits the power consumption of the system. Therefore the main motivation of this work is to investigate various pipelined multiplier and accumulator architecture and circuit design techniques which are suitable for implementing high throughput signal processing algorithm and at the same time achieve low power consumption [3]. A conventional MAC unit consists of (fast multiplier) multiplier and an accumulator that contains the sum of the previous consecutive
product. On the other hand, using reversible logic the implementation of digital circuits is gaining popularity with the arrival of quantum computing and reversible logic. In this paper, a novel reversible multiply accumulate unit is proposed. The comparison of various possible implementations of the reversible multiply accumulate unit in terms of gate count, quantum cost, constant inputs and number of garbage outputs is carried out.

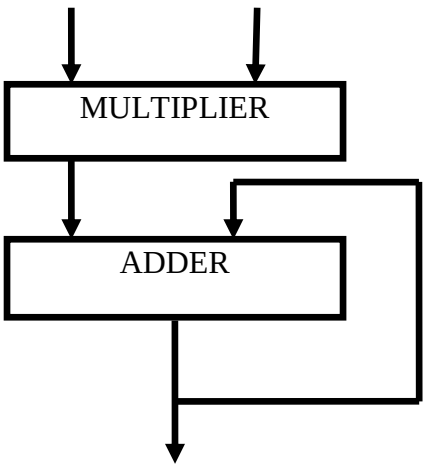

Figure 1. Architecture of MAC

I.
II.

III. FUNDAMENTALS OF REVERSIBLE LOGIC GATES

Arithmetic logic unit must be able to produce a variety of logical outputs such as AND, NAND, OR, and XOR based on outputs determined by the programmer for implementation in an instruction set architecture. Therefore the reversible logic gates used for this purpose must be able to maximize the types of logical operations, it can calculate while minimizing the number of select lines, cost and delay. A programmable reversible logic gate is defined here as a logic structure which possesses a bi-jection between input and output states and an equal number of inputs and outputs wherein a subject if the inputs are fixed select lines and a fixed subset of the outputs produce guaranteed logical calculations .Proposed programmable logic device must also consider which values to propagate to the output. In some instance, it may be beneficial to produce a copy of the input data values, whereas other designer may wish to propagate the input signal to the output signals. The following two algorithm are presented in order to ensure reversibility is maintained in the design of these logic gates .in these algorithm, the total number of inputs z, the number of data input is x, the fixed input select lines is y, the quantity of chosen propagated values to the output is p and the fixed select output is m [4].

Theorem 1. An ideal programmable reversible logic gate with z input and outputs has a quantity of fixed select inputs y, fixed select outputs m, data inputs x and propagated outputs p such that the magnitude of x – p = y – m. Reversible logic gates must have the same number of inputs and outputs. The number of fixed select inputs is the difference between the total inputs and the data inputs such that z = x + m. The number of the fixed logical outputs m, may be any value between 1 and z - p. When y > m, a number of garbage outputs g are incurred to maintain reversibility such that y - m = p - x + g. Therefore, in order to eliminate garbage outputs, the values for y - m and p - x must be identical. When m > y, a number of ancillary inputs 'a' are incurred such that m - y + a = x - p. Therefore in order to eliminate ancillary inputs. The value of m – y and x - p must be identical. Therefore, in order to maintain reversibility and eliminate ancillary inputs and garbage outputs, such that the magnitude of x – p = y – m [5].

Theorem 2. A programmable reversible logic gate with 'm' select inputs may produce at maximum $n * 2^m$ logical calculation on the 'n' logical outputs. The 'm' select inputs represent an input signal from the programmer and allow for up to 2^m unique input combinations. For each unique input signal combination, there may be one logical calculation per output. Since there are n outputs, the maximum number of logical outputs is $n * 2^m$. The simplest reversible gate is NOT gate and is a 1*1 gate. Controlled NOT (CNOT) gate is an example for a 2*2 gate.

There are many 3*3 reversible gates such as Fredkin (FG), Toffoli (TG), Peres (PG) and TR gate. The quantum cost of 1*1 reversible gates is zero, and quantum cost of 2*2 reversible gates is one. Any reversible gate can be realized by using 1*1 NOT and 2*2 CNOT reversible gates, such as V, V+ (V is square root of NOT gate and V+ is its Hermitian) and Feynman gate which is also known as CNOT gate. The V and V+ quantum gates have the property given in the Equations 1, 2 and 3.

$$V * V = NOT \quad (1)$$

$$V * V^{+} = V^{+} * V = I \quad (2)$$

$$V^{+} * V^{+} = NOT \quad (3)$$

The quantum cost of a design using reversible logic is calculated by counting the number of V, V + and CNOT gates [6].

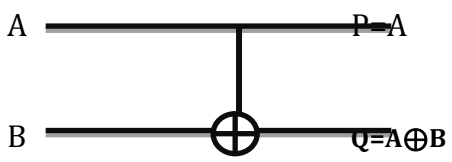

Figure 2. Quantum representation of

Feynman gate

The second type of fundamental 2*2 reversible- logic gate is the Feynman gate, or the controlled-not gate. It is configured such that its output states correlate to the input states in the following manner: P = A and Q = A⊕B. the resulting value of the second output corresponds to the result of a conventional XOR gate. Since fan-out is expressively forbidden in reversible logic. Since fan-out has one input and two outputs, the Feynman gate may be used to duplicate a signal when B is equal to 0. Its quantum configuration is shown in figure 2.

IV. RELEVANT WORKS AND LITERATURE REVIEW

R. Landauer et. al. in [7], have demonstrated Irreversibility and Heat Generations computing machines inevitably involve devices which perform logical function that do not have a single-valued inverse. This logical irreversibility is associated with physical irreversibility and requires a minimal heat generation, per machine cycle, typically of the order of kT for each irreversible function. This dissipation serves the purpose of standardizing signals and making them independent of their exact logical history. Two simple, but representative models of bi-stable devices are subjected to a more detailed analysis of switching kinetics to yield the relationship between speed and energy dissipation, and to estimate the effects of errors induced by thermal fluctuations.

A. Barenco, C.H. Bennet et.al in [8], have proposed a set of gates that consist of all one-bit quantum gates (U(2)) and the two bit exclusive-or gate (that maps Boolean values (x, y) to (x, x⊕y) is universal in the sense that all unitary operation on arbitrarily many bits ($U(2^n)$) can be expressed as composition of these gates. They investigate the number of the above gates required to implement other gates, such as generalized Deutsch-Toffoli gates, that apply a specific U(2) transformation to one input bit if and only if the logical AND of all remaining input bits is satisfied. These gates play a central role in many proposed constructions of quantum computational networks. They derive upper and lower bounds on the exact number of elementary gates required to build up a variety of two and three bit quantum gates, the asymptotic number required for n- bit Deustsch - Toffoli gates, and make some observations about the number required for arbitrary n-bit unitary operations.

Matthew Morrison et. al. in [9], reviewed reversible logic gate is gaining significant consideration as the potential logic design style for implementation in modern nanotechnology and quantum computing with minimal impact on physical entropy. Recent advances in reversible logic allow schemes for computer architectures using improved quantum computer algorithms. Significant contributions have been made in the literature towards the design of reversible logic gate structures and arithmetic units, however, there are not many efforts directed towards the design of reversible ALUs. In this work, a novel programmable reversible logic gate is presented and verified, and its implementation in the design of a reversible Arithmetic Logic Unit is demonstrated. Then, reversible implementations of ripple-carry, carry-select and Kogge-Stone carry look-ahead adders are analyzed and compared. Next, implementations of the Kogge-Stone adder with sparsity 4, 8 and 16 were designed, verified and compared. The enhanced sparsity-4 Kogge-Stone adder with ripple-carry adders was selected as the best design, and its implemented in the design of a 32-bit arithmetic logic unit is demonstrated.

V. MODIFIED ALU DESIGN WITH PERES GATE

There are different ways to implement a reversible adder. These different implementations depend on a balance between gates count, garbage outputs, ancillary bits and quantum cost. Being universal, these previously presented gates can implement any logical function and therefore they can also implement the well-known functions for a full-adder

$$S = A \oplus B \oplus Cin$$

$C_{out} = (A \oplus B)C_{in} \oplus AB$

A. DESIGN OF REVERSIBLE ADDERS

For this implementation, we are using the Peres gate because of the lower quantum cost of 4. This PFA (Peres Full Adder) can be taken as a block in order to facilitate the notation of its expansion. The inputs order was also changed to better fit in an expansion diagram. The Peres full adder is shown in figure 3 [10].

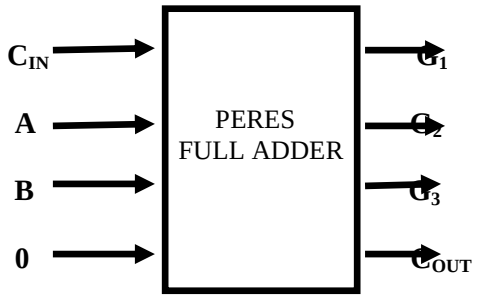

Figure 3. Peres Full Adder

Inputs of the complete 4 bits adder are three input vectors (4 bits) and a single bit C_{IN} (Carry in). Two of the three input vectors are the desired added 4-bits values. The remaining vector could be called the ancillary vector which is filled with zeros and outputs of the system are one garbage vector of 8 bits, one sum vector of 4 bits and a C_{OUT} (Carry out) bit. Unfortunately, as can be seen, the garbage cost to realize this system is very high [11].

B. SCHEMATIC DIAGRAME OF PERES REVERSIBLE FULL ADDER

In this adder we make the VHDL code as simple as possible, so we recurred to the Tops down design technique. First we implemented the main architecture in a general schematic involving each Peres Full Adder as a black box. This design is suited for 4 plus 4 bits. Once we

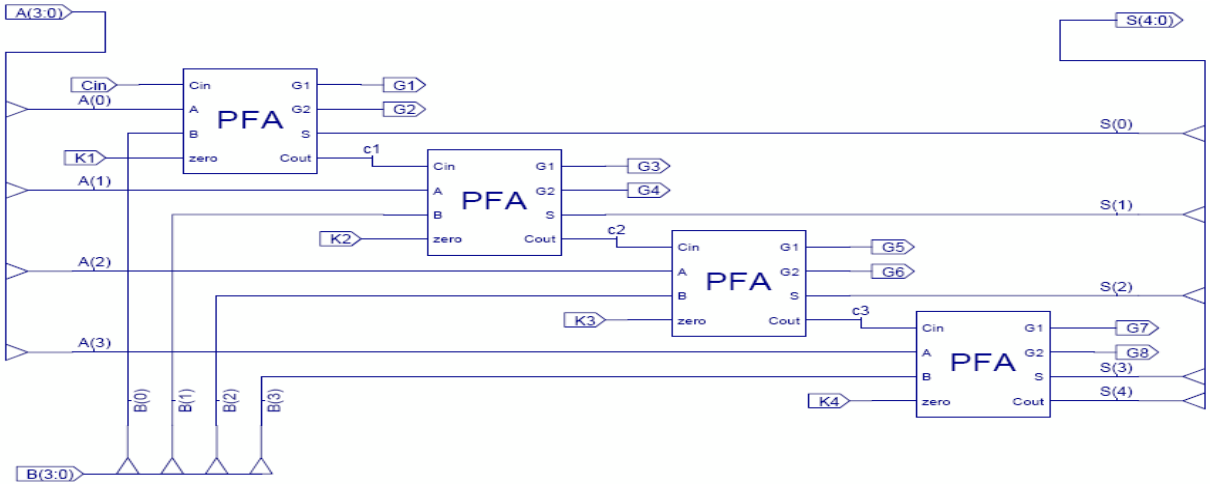

Figure 4. Schematic representation of the Peres full adder architecture

Proceeded to design the PFA block as depicted in the following figure 4. First of all, we need to know that in order to build a reversible circuit we must use reversible gates. The reader can learn more about the history of reversible logic by referring to [11]. The implementation of Reversible adder is mainly done for the purpose of reduce the area and circuit complexity of the Adder. The area calculation is done through the number required by the slices and gives the better output. Adding two bit numbers with a hybrid adder is done with two adders (using reversible logic gate) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The propagation delay is less for hybrid adder and at the same time it occupies less area compare to the other adder. The purpose of reduce the area and circuit complexity of the Adder. This makes efficient multiplier and very useful for optimized designing circuit complexity of MAC unit. Now we have successfully design 8 bit reversible multiplier. This multiplier is very useful for designing the delay and circuit complexity optimized for FFT, FIR, IIR, and DFT whose performance is dependent on the speed of MAC unit.

VI. PROPOSED METHODOLOGY

The designing of Multiplier Accumulation unit using reversible gates depends on delay introduced by the architecture of design digital circuits. In this literature survey we have studied various logic gates in order to

improve performance of delay as well as circuit complexity of the proposed MAC unit.

VII. CONCLUSION AND FUTURE SCOPE

The programmable reversible logic gates have been studied in reversible arithmetic logic units. These reversible logic gates, itself have many advantages in terms of reduce the complexity of circuit design and minimizes the delay. Since these reversible logic gates are designed by one to one mapping (equal number of inputs and outputs). Now we can design and implement the new multiplier accumulation unit by using reversible logic gates. In future we can design hybrid adder and multiplier individually and we can use it for designing and implementation of Multiplier Accumulation unit for digital signal processing application.

REFERENCES

[1] Haik Nasar and K Subbarao, "Design and Implementation of MAC Unit Using Reversible Logic," International Journal of Engineering Research and Applications, vol. 2, issue 5, pp. 1848-1855, 2012.

[2] Swaraj Raman M, Arun Kumar K and Srinivas Reddy K, "Reversible Implementation of Novel Multiply Accumulate (MAC) Unit," IEEE International conference on Communication information and Computing Technology, pp. 1-5, 2011.

[3] Shanbag N. R and Juneja P, "Parallel Implementation of a 4×4 – bit Multiplier using Modified Booth's Algorithm," IEEE J. solid state circuits, Vol. 23, No. 4, pp. 1010-1013, Aug.1988.

[4] Wille, D. Grobe, "Fast Exact Toffoli Network Synthesis of Reversible Logic," In Proceedings of International Conference on Computer-Aided Design, 2007, pp. 60-64.

[5] Y. Liao and D. B. Roberts, "A High Performance and Low Power 32-bit Multiply-Accumulate Unit with Single Instructions Multiple Data (SIMD) Feature", IEEE J. Solid State Circuits, vol. 37 no. 7, pp.926-931, July 2002.

[6] S. Islam, M. Rahman, Z. Begum and M Hafiz, Realization of a Novel Fault Tolerant Full Adder Circuit in Nanotechnology, vol. 7, issue 3, 2010, pp. 317-333.

[7] R Landauer, "Irreversibility and Heat Generation in the Computational Process," IBM Journal of Research and Development, Vol. 5, 1961, pp. 183-191.

[8] A. Barenco, C. H. Bennett, R. Cleve, D. P. Divincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, "Elementary Gates for Quantum Computation," Phys. Rev. A, Gen. Phys., vol. 52, no. 5, pp. 3457–3467, November 1995.

[9] Matthew Morrison, Matthew Lewandowski, Richard Meana and Nagarajan Ranganathan, "Design of a Moore Finite State Machine using a Novel Reversible Logic Gate, Decoder and Synchronous Up-Counter," 11th IEEE International Conference on Nanotechnology Portland Marriott, pp. 1445-1449, August 15-18, 2011. Journal of Research and Development, Vol. 5, 1961, pp. 183-191.

[10] J. Fadavi-Ardekani, "Booth Encoded Multiplier Generator using Optimized Wallace Trees," IEEE Transaction Very Large Scale Integration (VLSI) System, vol. 1, no. 2, pp. 120–125, June 1993.

[11] Nagapavani T, V. Rajmohan and P. Rajendaran, "Optimized Shift Register Design Using Reversible Logic," Proceedings of the third IEEE International Conference on Electronics Computer Technology, pp. 236-239, 2011.