Design and Analysis of Multiplier Accumulation Unit by using Hybrid Adder

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

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

Abstract - A new architecture of Multiplier accumulation unit (MAC) by using reversible logic gates Algorithm for reducing circuit complexity, power consumption and delays, have been proposed and implemented on Xilinx FPGA device. By combining a reversible multiplier and reversible adder, design a hybrid type Multiplier accumulation unit by using reversible logic gates. 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 important 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)

A Multiplier Accumulation (MAC) unit operates on two sequences of complex numbers, \( \{X_i\} \) and \( \{Y_i\} \). The Multiplier accumulation unit multiplies corresponding elements of the sequences and accumulates the sum of the products. Each complex number is represented in Cartesian form, consisting of a real and an imaginary part. Our MAC calculates its result by taking successive pairs of complex numbers, one each from the two input sequences, forming their complex product and adding it to an accumulator register. The accumulator is initially cleared to zero and reset after each pair of sequences has been processed. The multiplication operation is generally performed by multiplying. It is possible to decompose the multipliers into two steps. The first step is to the generation of partial products, and the second one collects and adds them. 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](image)

**III. Fundamental 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 bijection 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 \( n * 2^{2m} \) logical calculation on the ‘n’ logical outputs. The ‘m’ select inputs represent an input signal from the programmer and allow for up to \( 2^{2m} \) 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 \( n * 2^{2m} \). 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](image)
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. Modified MAC 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

\[ S = A\oplus B\oplus C_{\text{in}} \]
\[ C_{\text{out}} = (A\oplus B)C_{\text{in}} \oplus AB \]

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 [7].

A. Design of Reversible Adders

Inputs of the complete 4 bits adder are three input vectors (4 bits) and a single bit C_{\text{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_{\text{out}} (Carry out) bit. Unfortunately, as can be seen, the garbage cost to realize this system is very high [8].

B. Schematic Diagram of Peres Reversible full Adder

In this MAC 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 that we had this, and we 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 [9].
V. Reversible Adder Simulation

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, some of the most used universal quantum gates and their 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. The implementation of Reversible adder is mainly done for the purpose of reducing the area and circuit complexity of the Adder [10]. 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 gates) 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 compared to the other adder. The purpose of reducing 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 [11].

A. Multiplier

Proposed work have designed the $8 \times 8$ bit Multiplier which provides better result with respect to the available reversible multiplier or other Multiplier and as known dynamic power is proportional to the frequency (clock speed capacity) consumed to perform any operation. So, it can say that proposed design also reduces the dynamic power indirectly. This makes efficient multiplier and very useful for optimized designing circuit complexity of MAC unit. Now we have successfully designed $8$ 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 [12].

![Figure 5. Simulation of Proposed Adder](image-url)
VI. Multiplier Accumulation Unit Simulation

The proposed architecture is defined in VHDL and simulated using Xilinx ISE tool. Values are taken in a 16 bit Multiplicand (num1) and Multiplier (num2 operands). A 32-bit MAC out operand is defined which displays the result. A 32-bit output operands are also defined which displays multiplier accumulation result is shown in figure 5. The implementation of MAC unit is mainly done for the purpose of reduces the area and complexity of the multiplier accumulation unit. Applied 16 bit Multiplier and 32 bit reversible Adder. The area calculation is done through the number required by the slices and LUTs and to give the better result. The reversible multiplier accumulation has been synthesized and implemented on Spartan-3E based “XC3S500E” has been chosen and the package as “FG320” with the device speed such as “-4”. The design of MAC is synthesized and its result was analyzed as follows. The speed calculation is done through the time period required by the click cycle and clock period should always less than or equal to Q delay of a flip-flop plus combinational delay plus setup time. The complete simulation of MAC unit by using different logic families and the results can be shown in the form of table.

![Image of figure 6. Simulation of Proposed MAC](image)

VII. Result Comparison

The comparison for above techniques of multiplier accumulation unit is described below. Table [1] shows the resource comparison and figure [7] shows the comparative analysis of the logic utilization, available and source utilization. The proposed MAC can operate at maximum operating frequency of 86.987 MHZ with minimum period of 11.496 ns here.

Table 1. Resource Comparison between Existing work and Proposed Work

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Available</th>
<th>Existing Work Utilization</th>
<th>Proposed Work Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of slices</td>
<td>4656</td>
<td>8.6%</td>
<td>4%</td>
</tr>
<tr>
<td>Number of input LUTs</td>
<td>9312</td>
<td>14.31%</td>
<td>3%</td>
</tr>
<tr>
<td>Number of bonded IOBs</td>
<td>220</td>
<td>10%</td>
<td>41%</td>
</tr>
<tr>
<td>Number of slice flip-flops</td>
<td>9312</td>
<td>7.3%</td>
<td>3.9%</td>
</tr>
</tbody>
</table>
The proposed MAC unit has consumed only 199 352 number of LUTs and 365 slice flip-flops as compared to 404 slices, 1333 number of LUTs and 673 slice flip-flops, and the developed design number of slices, LUTs and number of slice flip flops. Table [1] shows the comparison of resource utilization between Existing work and proposed work, in which the proposed Multiplier Accumulation unit utilizing less resource than existing work. Figure [6] shows the design parameter chart comparison of resource utilization between Existing work and proposed work. The proposed MAC unit has consumed only 4% slices, 3% number of LUTs and 3.9% slice flip-flops as compared to 8.6% slices, 14.31% number of LUTs and 7.3% slice flip-flops, and the developed design number of slices, LUTs and number of slice flip flops. Comparative analysis is shown in figure [7].

Figure 7. Comparative Analysis

VI. Conclusion

The Multiplier Accumulation unit is commonly used in digital signal processing applications. MAC unit is a basic arithmetic cell in computer processing units. Furthermore, reversible implementation of this unit is necessary for quantum computers. Targeting this purpose, various designs can be found in the literature. We designed a novel 16x16 bit multiplier and 32x32 bit reversible circuit using Peres gates. The proposed reversible adder circuit is better than the existing designs in terms of hardware complexity, number of gates, garbage outputs and constant inputs. Furthermore, the restrictions of reversible circuits were highly avoided.

Hybrid adder designed using reversible logic gates is used to give the better result on its performance. Peres reversible logic gates are used to implement hybrid adder instead of conventional logic. As it has been seen from the implementation table that the proposed Multiplier accumulation unit consumed only 199 slices as compared to 404 slices, 352 LUTs as compared to 1333 LUTs and 365 number of slice flip flop as compared to 673 numbers of slices flip flop. Implementation of Multiplier accumulation unit by using hybrid adder improved the area efficiency and also reduces the circuit complexity to provide high performance cost effective solution.

REFERENCES


