International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Design of Efficient Braun Multiplier for Arithmetic Applications

Telagamalla Gopi

Assistant Professor, Department ECE, Annamacharya Institute of Technology and Science, Hyderabad, Telangana State, India

Abstract: ***Multiplier plays key role in Digital Signal Processing and FPGA based applications, as it consumes more power and area compared other devices. In DSP and FPGA based applications power, speed and area all are important parameters and all depend on multiplier which in turn depends on adders. So, by implementing adders we can reduce the delay. Pyramidal adders are used which uses half-adder and full-adder to increase the speed and to reduce the number of gates used in the multiplier, but delay is not decreased significantly. If we modify the Pyramidal adder with XNOR's and MUX instead of normal half-adder and full-adder, such pyramidal adder uses less gates and delay is reduced compared normal 16-bit adder. The use of XNOR's and MUX in Pyramidal adder reduces delay, as the MUX function is only select the output among inputs. The use such pyramidal adder in multiplier delay can be decreased greatly. Modified full adder sum and carry values are same as normal full adder but the transistor topology is different.***

Keywords: multiplexer (MUX), half-adder (HA), full-adder (FA), field programmable gate array (FPGA), digital signal processing (DSP)

1. Introduction

In matrix multipliers, summation is carried by a matrix of adder, which consist successive rows of one bit adders with transfer of saving, as the data moves transfer saving adds another partial product to the sum of partial products. High performance can be achieved with their regularity, which is important when implementing such multipliers in integrated circuits. On the other hand due to increase in bit-capacity large area is required and another disadvantage is that matrix multipliers have low level equipment utilization. Matrix multipliers like Braun multiplier uses horizontal spreading of transfer where as Baugh-Wooley's multipliers uses complementary codes for binary multiplication.

Tree-like structured multipliers like Wallace's and Dadda'D multipliers, delay can be greatly decreased. Tree-structured multipliers are faster than matrix multipliers however additional links are required which occupies more area in the crystal circuit compared to matrix multipliers. In multilayer multipliers summation is carried out by tree-like structured multiplier to compress partial products with same weight using one-bit half-adders and full-adders. Multilayer multipliers consume less power and less area, so very useful to DSP and high performance systems. Booth Algorithm is used to reduce the number of partial products but increases the number of shifts and intermediate sums will increase, it causes to occupy more area and decreases the speed of operation. Binary half-adder is hardware formed complex circuit with five logic elements; if we use such half-adder in multi-combinational adders it gives more complex circuit. For example, for 1024-bit DSP processor, and also speed is reduced due to serial connecting logic elements. For n*n bit multiplier, n*n AND gates and n(n-1) OR gates are required, in terms of adders n number of half-adders and n(n-1) number of full-adders are required.

2. Literature Survey

Implementation of ALU and arithmetic devices depends on functionality of small bit processor (32-bit), accelerating methods are used for multi-bit processors (1024, 2048…). For DSP and FPGA based multipliers accelerating methods must be used to reach the required speed.

3. Existing Method

3.1 Matrix and Tree like Structures

Figure below shows the 4-bit matrix Braun multiplier, in which each column corresponds to the diagonal of the multiplier. Partial products formed with bits a_j and b_j using AND gates, to add partial products half adder and full adder are used. For Braun m matrix nxn multiplier, n^2 products and n^2-1 operations of 1-bit additions required where n is the bit capacity of the input data, speed of this multiplier is determined by longest route of signal processing i.e. –(3n-4). Tree-like structure Wallace's tree multiplier for nxn multiplier, n^2 products and n^2-n operations of 1-bit additions required, so for full binary adding n^2-n operations needed and for not full binary adding n operations required. speed is limited by (2n-2) one bit adders. For small bit capacity multiplication Dadda's tree is widely used, nxn multiplier n^2 products and n^2-n operations of 1-bit additions required where n is the bit capacity

3.2 Proposed Method

Improving the performance of digital adder is needed because execution of binary operation completely depends on adders, there are so many adders are implemented such that to meet the requirements of FPGA based VLSI environment and DSP Processor operations.

- **Ripple carry adder:** It is simplest adder among all adders but slowest adder, it requires O (n) and delay of O (n), here n represent the operand size.
- **Carry Look Ahead Adder:** It has good area O (n log) and good delay of 0(log n), but suffers from irregular layout design.
- **Carry select adder:** It has area of O (n) and delay with O ($n^{l+2/l+1}$), and it is the best adder in terms of area and delay.

- **Carry save adder:** Requires area O (n) and delay of O (log).
- Carry select adder is the fast adder as it reduces computation time for operation among all adders but suffers from fan out limitation. The sorting problem is defined as the re-arrangement of N input values so that they are in ascending order, merge sort method uses divide and conquer algorithm and uses recursion to perform sorting.
- **Pyramidal adder:** Hardware complexity of multipliers can be greatly reduced by using so many hardware structures and in those one of the structures is pyramidal adder.

It contains single input 2n bit bus carrying inputs (a_0b_0, a_1b_1 a_2b_2 a_3b_3…..a_nb_n), the pyramidal structures contains three single bit adders namely 2.1 block-to direct transfer of outputs, 2.2 block-to transfer inverse outputs, and 2.3block-to inverter the output bus of combinary adder (S_0 S_0, $S_1S_2S_3$…..S_n), here 2.1 and 2.2 blocks acts as half adders as compared to half adder with five logic gates, so there is a reduction of gate count in the multipliers with 2.1 and 2.2 blocks. If binary braun multiplier is implemented with pyramidal adder the gate count is reduced and speed of operation also increased.

3.3 Block (normal half-adder)

It uses NAND gate, AND gate and OR gate, its function is
$S_i=(a_i*b_i)x\ (a_i+b_i)= =a_i\ `*b_i+a_ib_i`$
$P_+= a_i*b_i$
Where S_i is the sum and P_+ is the carry

3.4 Block (normal full-adder)

It uses two NAND gates and one OR gate
It uses NAND gate, AND gate and OR gate, its function is
$S_i= (a_i*b_i)' * (a_i+b_i) = a_i\ `*b_i+a_ib_i`$
$P_+= (\ a_i*b_i)'$
The P_+ is given to inverter to get final carry P_+

For standard 4x4 bit Braun multiplier using normal half adders and full-adders consists there are 120 gates, if multiplier is implemented with pyramidal adder the gate count is 76, but we can get reduced delay compared to standard multiplier, then we will go for 16x16bit pyramidal adder.

4. Implementation of 16- Bit Pyramidal Adder

16-bit Pyramidal adder implemented with modified half-adder and full-adder instead of using normal half-adder (2.1 block) and full-adder (2.2 blocks), this time for 16-bit pyramidal is implemented with both XNOR's and MUX, in this output selected from available inputs i.e. just selecting input, because of this implementation both gate count and delay are reduced compared to standard 16-bit adder which uses normal half adders and full adders.

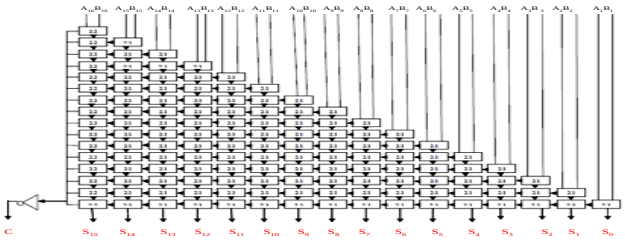

Figure 1: 16x16-bit Pyramidal adder with modified half adder (2.1 block) and full adder (2.2 block)

This time for 16-bit pyramidal is implemented with both XNOR's and MUX, in this output selected from available inputs i.e. just selecting input, because of this implementation both gate count and delay are reduced compared to standard 16-bit adder which uses normal half adders and full adders.

16-bit pyramidal adder consists two 16 bit inputs a and b respectively and they generate 16 sum output and 1 carry output, here 2.1 block and 2.2 blocks acts as half-adders hence each block generates one sum and one carry bit.

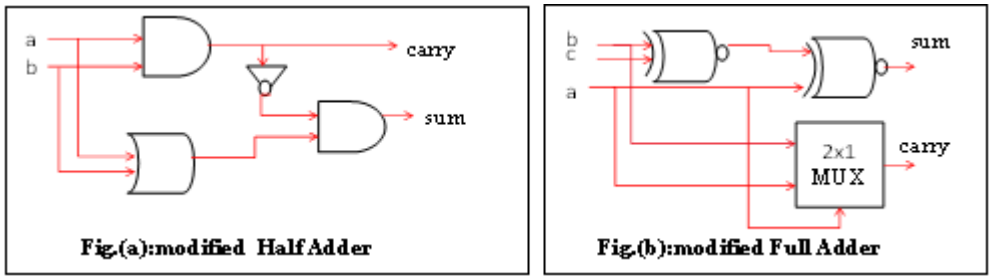

Figure 2: (a) Modified half adder (2.1 block) (b) Modified full adder (2.2 block)

5. Implementation of 16-Bit Multiplier using modified Pyramidal Adder

16-bit Braun multiplier is implemented with modified half-adder and full-adder, for nxn multiplier if input a[15:0] and b[15:0] is applied we will get 2n output sum values i.e S[31:0].In this Braun multiplier MUX selects the output among inputs, so delay is decreased significantly.

6. Simulation Results of 16-Bit Pyramidal Adder and 16-Bit Multiplier

Simulation and synthesis results of both 16 bit pyramidal adder and 16-bit multiplier are obtained using xilinx ISE 14.7 design software.

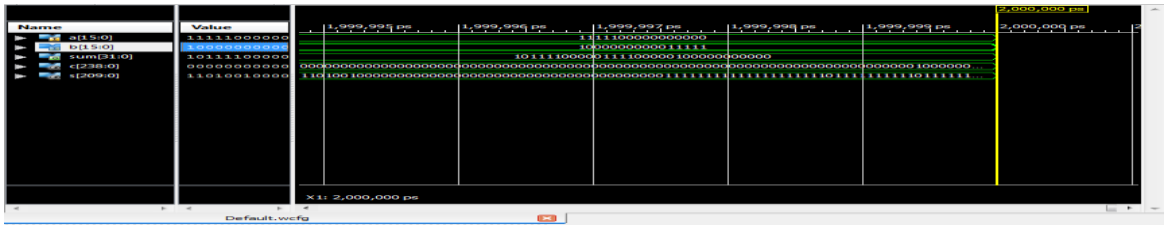

Figure 3: Graphical representation of 16x16 bit Braun multiplier

16-bit Braun multiplier implemented with Pyramidal adder, utilizing only 26% from available LUT's and 96% from available IOB's, in terms delay Braun multiplier consumes 38.48ns compared to normal Braun multiplier with 39.94ns.

7. Conclusion

Binary multipliers are widely used in DSP processors and FPGA based VLSI domain environment, where area, power, speed and delay are important parameters. Power, area, speed and delay depends on multipliers which in turn depends on adders, so by implementing adders we can reduce the required parameter values. There are so many adder structures are there for binary multipliers, pyramidal adder is the one structure to reduce hard ware complexity and delay. In this thesis normal half adder and full adders are modified with XNOR gates and MUX results minimum delay as the MUX function is to select output among inputs. By using of modified half-adder and full-adder both gate count and delay is reduced normal to 16-bit multiplier.

References

[1] Grega .V, Kolosov .I, Danyluk .O (2016) The development of a Fast Iterative Algorithm Structure of Cosine Transform in Proceedings of XIIIth International Conference TCSET2016—Lviv-Slavsko, Ukraine - pp.506-509.
[2] Pezaris .S(1971) A 40-ns 17b by 17b Array Multiplier, IEEE Transactions on Computers, c-20, pp.442- 447.
[3] Baugh .C, Bwooley (1973) A 2's Complement Parallel Array Multiplication Algorithm, IEEE Transactions on Computers, c-22, pp. 1045 – 1047.
[4] Dadda .L Some Schemes for Parallel Multipliers //Altra Freguenza-v .34 –p.349 356.
[5] Volodymyr Gyrga, Bogdan Dzunda, Ivan Dadaik, Yarslov Nykolaichuk (2018) Research and Implementation of hardware Algorithm for multiplying binary numbers, IEEE, PP. 1277 -1281.

Author Profile

Telagamalla Gopi received B-Tech degree from Scient Institute of Technology Hyderabad, completed M-Tech with VLSI system design from Sree Dattha Institute of Engineering and Science. Currently working as Assistant Professor in Annamacharya Institute of Technology and Sciences, Hyderabad. Has 8 years of teaching experience. Areas of interest include digital signal processing and VLSI.