LOW POWER 16X16 BIT MULTIPLIER DESIGN USING DADDA ALGORITHM

J.I. CHAKRAVARTHY, B. Tech, M. Tech, BANOTH GANESH [2], SAVILLA ADARSH [3], PADALA MANIMALA [4].

[1] ASSOCIATE PROFESSOR OF ECE IN MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE, MAISAMMAGUDA, MEDCHAL (M), HYDERABAD-500100, T. S.

[2, 3, 4.] FINALYEAR STUDENTS FROM DEPT OF ECE IN MALLA REDDY INSTITUTE OF TECHNOLOGY & SCIENCE, MAISAMMAGUDA, MEDCHAL (M), HYDERABAD-500100, T. S.

ABSTRACT: In this research paper, The model of 16-bit multiplier having low power and high speed using Algorithm named Dadda and the basic building block used is optimized Full adder having low power dissipation and minimum propagation delay the technique for the design of a fast multiplier is proposed by using two different techniques. A 4:2 compressor is used for power-efficient row compression in the proposed model, whereas for faster final summation, a carry-select adder has been used. Based on the above technique, a 16-bit Dadda-multiplier is proposed and its performance is analysed by comparing it with the standard 16- bit Dadda-multiplier which normally uses carry-look ahead adder. The time delay of carry-select adder is improved when compared to carry propagate adder. The simulation of the proposed multiplier is carried out using Verilog HDL in Xilinx ISE Design Suite 14.7.

Keywords— ALU, Dadda-multiplier, 4:2 compressor, carry select adder, carry propagate adder, Wallace-multipliers, half adder, full adders

I. INTRODUCTION

Multiplier is one of the most important circuits for designing computers and computing devices. The Dadda technique for partial product reduction is based on the idea of ‘avoid use of full adder.’ But the use of full adder is more regular in other Wallace tree multipliers. This paper presents a modified Dadda technique based on the idea of ‘prefer the use of full adder over the use of half adder.’ Only the last stage that is ‘three to two reductions’ is the exception. This idea used in the modified technique makes it more regular and simple. Hence, based on the idea, it is named as ‘Full-Dadda.’ The fact behind saying this technique an alternative approach is that this technique results in same number of full adders, half adders, same size of final carry propagation adder and same number of compressors (adders) at each stage as required in the Dadda technique. Therefore the proposed multiplier can be used in place of Dadda multiplier in all its applications. This paper presents a comparative performance analysis of the proposed multiplier with the Dadda multiplier. For this comparison each multiplier with different operand sizes is taken. The main disadvantages of the Dadda multiplier are: (i) it is less regular, (ii) more complex and (iii) it reduces less number of bits at early stages of reductions. The proposed ‘Full-Dadda’ multiplier is more regular, simple and reduces more number of bits at early stages of summand reduction. In section 2, there is brief overview of literature and in section 3; there are general rules and equations for reduction scheme and number of hardware components respectively. The comparison between multipliers is arranged in five sub-sections under section.

II. Literature Survey

A. Some Important points

The parallel multipliers are faster and more fancied [1]. There are many ways and schemes available for multiplication [1], such as Array multiplication scheme, Booth multiplication and Vedic multiplication etc. Among these schemes, the tree multiplication scheme is one of the most popular schemes. [1,2,3,4,5,6]. The most popular multiplier among tree multipliers is the Dadda multiplier [2]. A Reduced Area multiplier is also the best optimized multiplier in terms of Area if interconnects are properly managed [3]. The basic steps involved in a tree multiplication scheme, first used by Wallace [6] in 1963 are:

1) Generation of the partial products [3] (or summands [5, 6] or matrix of partial products [1]).

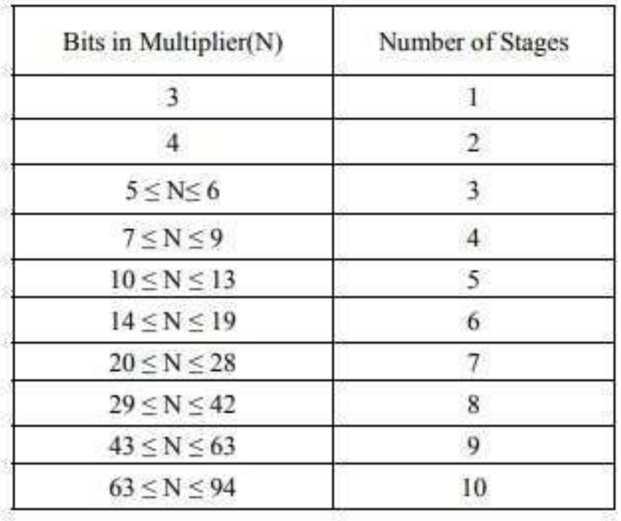
2) Partial products reduction: reduction in column height by using pseudo adders (there are many parallel addition schemes [8] and partial product reduction schemes [9] implemented by using pseudo additions such as conditional sum addition [7] etc.).

3) Use of final CPA (carry propagation adder): After the last stage of column-height reduction (remaining bits in a column are only two), there is a need of final addition of two rows to obtain final product of multiplied operands. The size and type of final CPA used plays an important role in determining overall performance such as delay, area and power consumption [2, 3, 5].

B. Dadda Multipliers

Dadda method for partial product reduction uses only necessary reduction determined by the Wallace table shown in table 1 [2, 3, 4]. It uses an approach to increase the number of half adders and to decrease the number of full adders required for overall reduction of partial products. The detail of Dadda technique is given in the references [2, 3, 4, 5, 9]. The Dadda method keeps as these are all right columns having height less than or equal to the necessary bits required at a stage after reduction. This is repeated at every stage.

Table1: Number of reduction stages for Dadda multiplier



Bits in Multiplier(N)	Number of Stages
3	1
4	2
$5 \leq N \leq 6$	3
$7 \leq N \leq 9$	4
$10 \leq N \leq 13$	5
$14 \leq N \leq 19$	6
$20 \leq N \leq 28$	7
$29 \leq N \leq 42$	8
$43 \leq N \leq 63$	9
$63 \leq N \leq 94$	10

III. PROPOSED METHOD

Two of the most well-known column compression multipliers have been presented by Wallace [5] and Dadda [6]. Both architectures are similar with the difference occurring in the procedure of reduction of the partial products and the size of the final adder. In Wallace's scheme, the partial products are reduced as soon as possible. On the other hand, Dadda's method does minimum reduction necessary at each level and requires thesame number of levels as Wallace multiplier. As a result, final adder in Wallace multiplier is slightly smaller in size as compared to the final adder in Dadda multiplier. The Block diagram of proposed energy efficient column compression multipliers (Wallace/Dadda) is shown in Fig

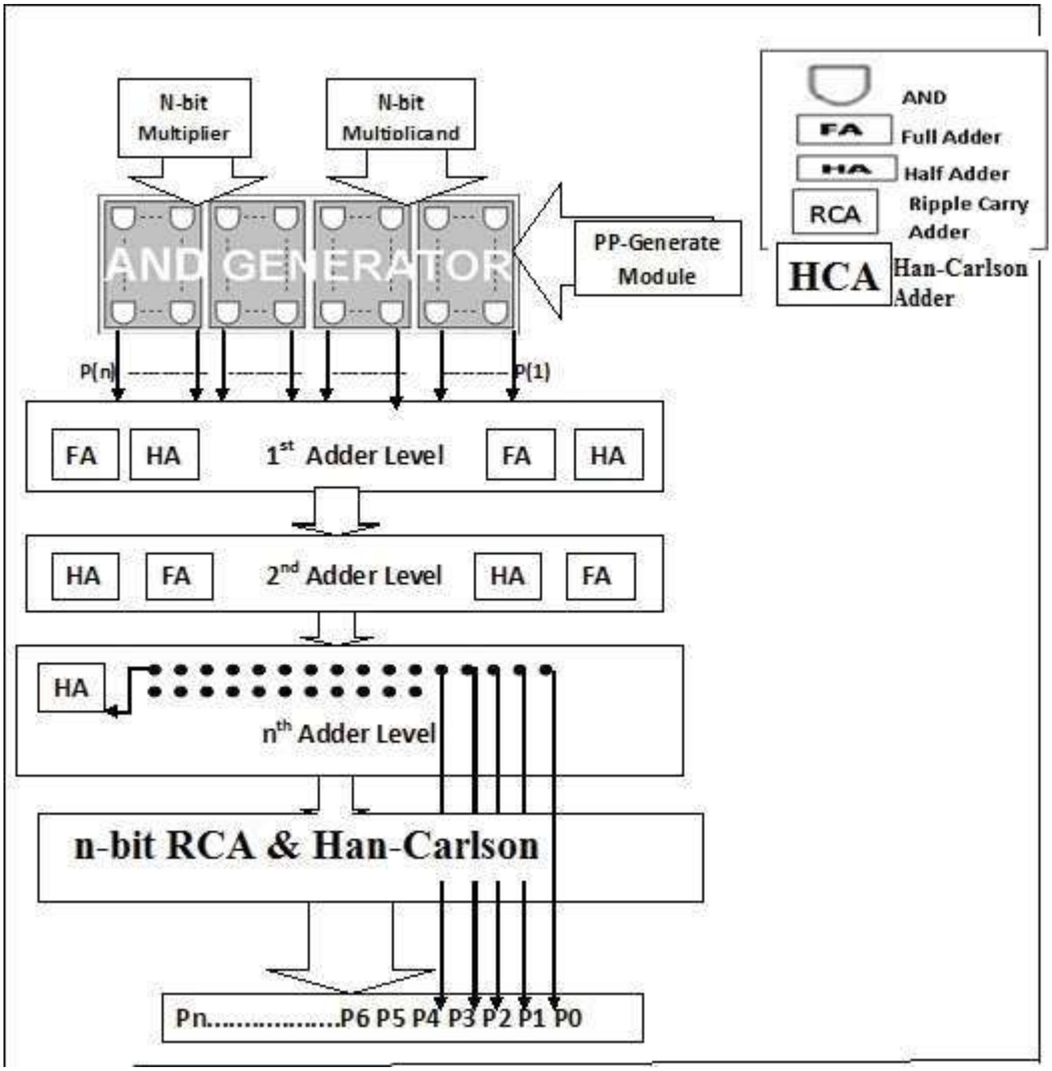


Fig. 1. Block diagram of proposed energy efficient n-bit column compression multiplier

A Wallace/Dadda multiplier is usually composed of threeparts (or modules)

- Partial product generate module

A Half Adder (HA) and Full Adder (FA) tree toreduce the partial products matrix to anaddition of only two Operands.

Han-Carlson and Ripple Carry Adder (RCA) for thefinal computation of the binary result.

In the proposed energy efficient multipliers, the reduction partuses half-adders, full-adders and ripple carry adders; eachpartial product bit is represented by a dot as shown in Fig 3.1.Reduction is performed depending on the number of elementsin that particular column of the group. Three dot products issued to represent a FA, whereas only two dot products is used To represent a HA. If a column has only one element then that ispassed on to the next stage without any reduction. If the lastgroup of a stage contains less than three rows then no reductionis performed on that group. The last part performs the functionto add the remaining two rows using an exact RCA & exactHan-Carlson adder to compute the final binary result in sub thresholdregime. All the standard cells are same in both 8x8column compression Wallace & Dadda multipliers, StaticCMOS logic family is used for AND gate, HA, FA and partialproduct generation module in sub-threshold regime.

ASIC IMPLEMENTATION

The complete ASIC implementation of the proposed design isalso done which follows the cadence design flow. Theproposed design has been developed using Verilog-HDLand synthesized in Encounter RTL compiler using typical libraries of 45 nm technology. The test bench is created for simulation and logic verification by Model-Sim simulator. The CadenceSoC Encounter is used for Placement & Routing (P&R).Parasitic extraction is performed using Encounter Native RC extraction tool. The extracted parasitic RC (SPEF format) is back annotated to Common Timing Engine in Encounter Platform for static timing analysis. Results obtained from both tools are analyzed and compared for semi-custom design verification shown.

ISSN: 2278-4632
Vol-13, Issue-11, November 2023

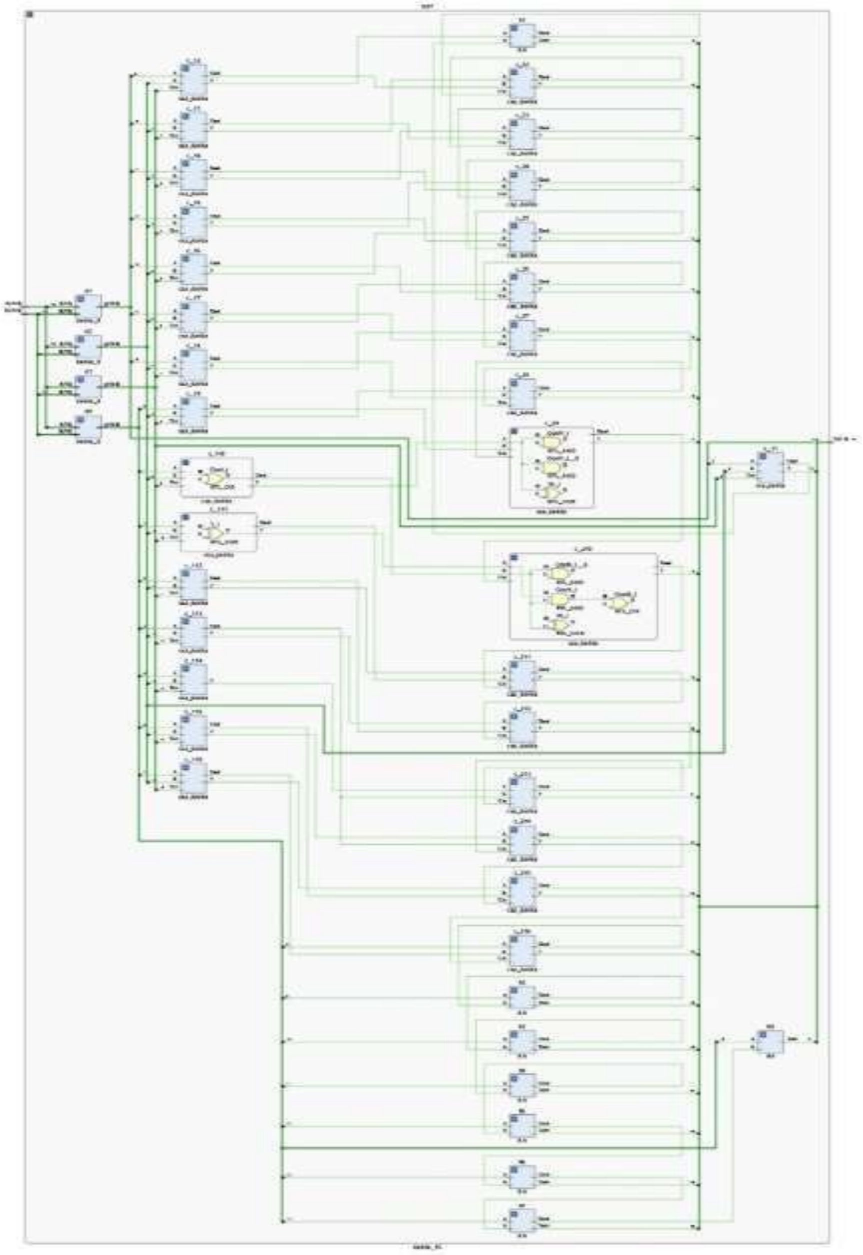


Fig 2 RTL Schematic 16*16 Dadda Multiplier

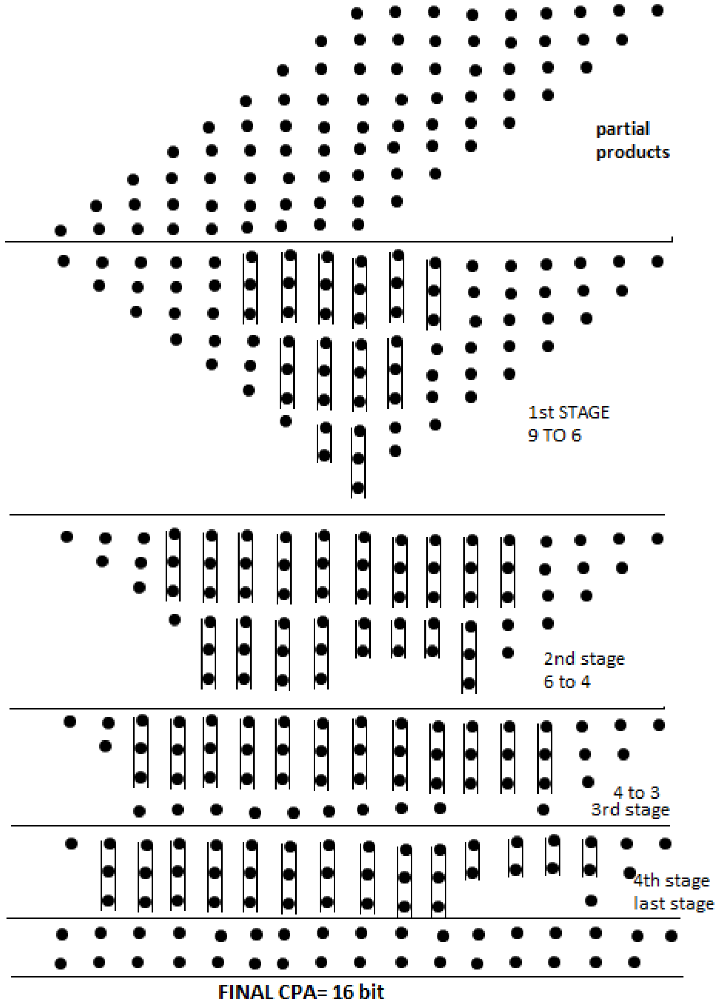


Fig 3: Proposed Multiplier Reduction Scheme

The ASIC Implementation is to be done to differentiate the performance evaluation of the Wallace and Dadda multipliers in sub-threshold and super-threshold regime. The stimulation results were carried out of two 8x8Wallace & Dadda multiplier designs using RCA and two 8x8Wallace & Dadda multiplier design using HCA in sub threshold regime. The designed energy efficient multiplies operates at 0.4V. In fact, in addition to normal transistors, circuits are tested in corner cases with fast and slow transistors and their combinations too. In each stage one of the components FF, SS, FS, and SF is replaced instead of normal transistors in circuit and is perused in each circuit function. Since the delay of multiplier circuit is proportional to the logarithm of the number of bits in the multiplier and also delay of used standard cells. To measure the critical path delay and to verify the functionality of proposed Wallace/Dadda multipliers, n-no. Of test patterns have been applied to the multiplier. The worst delay has been observed for the inputs from11111111x00001000 for 8x8 Wallace/Dadda multipliers.

IV. SIMULATION RESULTS

The proposed design has been developed using Verilog-HDL and synthesized in Xilinx vivado tool and the same tool was used for Placement & Routing (P&R).Results obtained from the tool are analyzed and the schematic is shown in the below figure.

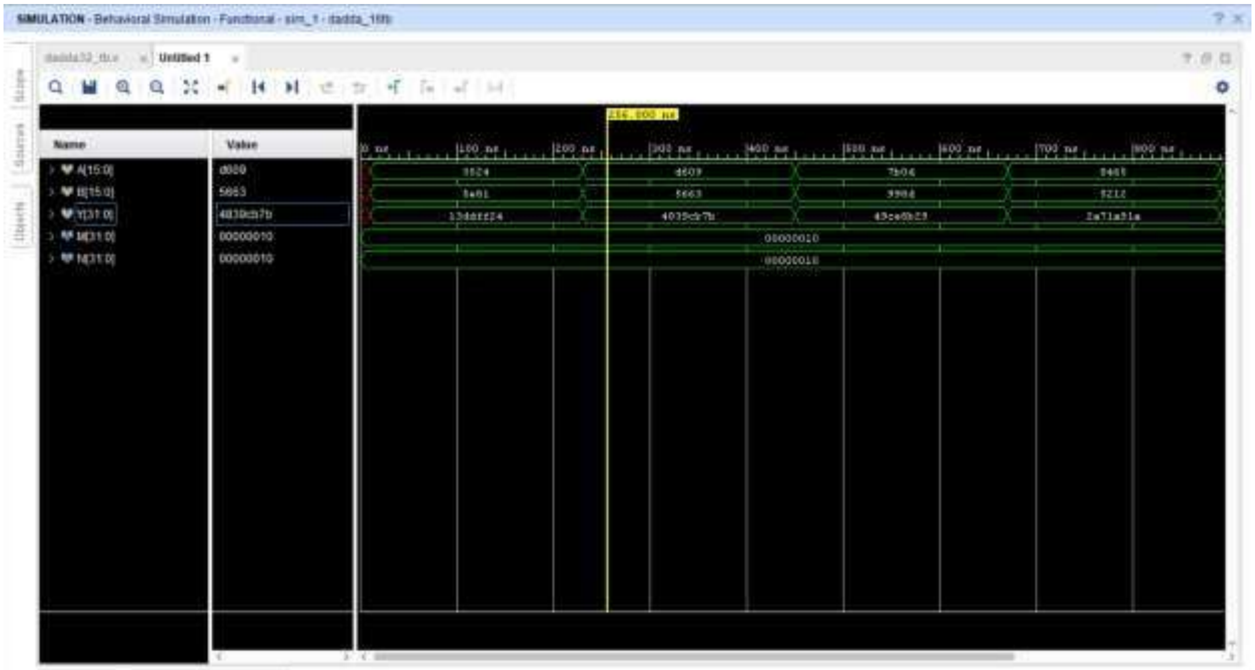


Fig 4 simulation results for 16*16 Dadda Multiplier

V CONCLUSION

In this paper, the 16x16 Dadda multipliers have been implemented at0.4 volt power supply. The main focus of this paper was to optimize overall power delay product. We have also optimized the no. of transistor in the design. The simulated result shows that the proposed Dadda have improved results as compared to existing published results. Thus the design of Dadda multiplier has improved power, performance.

VI. FURTHERSCOPE

As can be seen from the results obtained by Dadda multiplication scheme, this approach is further extended to perform the multiplication of higher bits (i.e., 32 bit × 32 bit, 64 bit × 64 bit and so on). The power consumption and area estimate are further reduced by implementing the final adder with look ahead carry generation logic (look ahead carry adder).

VII. REFERENCE

[1]. High-Speed Parallel Multiplication Scheme," IEEE TRANSACTIONS ON COMPUTERS, VOL. C-26, NO. 10, OCTOBER 1977.

R. S. Waters and E. E. Swartzlander, "A reduced complexity wallace multiplier reduction," IEEE Transactions on Computers, vol. 59, no. 8, pp. 1134–1137, 2010.

[2] K. C. Bickerstaff, M. J. Schulte, and E. E. Swartzlander, Jr., "Reduced Area Multipliers," in Proceedings of the 1993 International Conference on Application Specific Amy Processors, pp. 478-489, IEEE, 1993.*The Proposed Full-Dadda Multipliers (IJIRST/ Volume 4 / Issue 11 / 017)*

[3] K'Andrea C. Bickerstaff, Earl E. Swartzlander, Michael J. Schulte, "Analysis of Column Compression Multipliers",15th IEEE Symposium on Computer Arithmetic Proceedings, pp 33-39, 2001.

[4] A. Habibhi and P.A. Wintz "Fast Multipliers," IEEE Transactions on Computers, Vol. C-19, pp. 153-157, 1969

[5] C. S.Wallace, "A suggestion for a fast multiplier," IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14–17, 1964.

[6] J. Sklansky, "Conditional-sum addition logic," IRE Transactions on Electronic Computers, vol. EC-9, pp. 226–231, 1960

[7] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Transactions on Computers, vol. C-31, no. 3, pp. 260–264,

[8] Wesley Chu, Ali I. Unwala, Pohan Wu, and Earl E. Swartzlander, Jr., "Implementation of a High Speed Multiplier Using Carry Lookahead Adders," 978-1- 4799-2390-8/13, 2013, IEEE. K'Andrea C. Bickerstaff, Earl E. Swartzlander, Michael J. Schulte, "Analysis of Column Compression Multipliers",15th IEEE Symposium on Computer Arithmetic Proceedings, pp 33-39, 2001.

[9] A. Habibhi and P.A. Wintz "Fast Multipliers," IEEE Transactions on Computers, Vol. C-19, pp. 153-157, 1969

[10] C. S.Wallace, "A suggestion for a fast multiplier," IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14–17, 1964.

[11] J. Sklansky, "Conditional-sum addition logic," IRE Transactions on Electronic Computers, vol. EC-9, pp. 226–231, 1960

[12] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEE Transactions on Computers, vol. C-31, no. 3, pp. 260–264,

[13] Wesley Chu, Ali I. Unwala, Pohan Wu, and Earl E. Swartzlander, Jr., "Implementation of a High Speed Multiplier Using Carry Lookahead Adders," 978-1- 4799-2390-8/13, 2013, IEEE.