A High Speed Design of 32 Bit Multiplier Using Modified CSLA

1.Vijaya kumar vadladi 2.David Solomon Raju. Y
1.M. Tech. Student (VLSID), Holy Mary Institute of Technology and Science, Keesara, R.R. Dt.
2.Associate Professor, Dept. of ECE, Holy Mary Institute of Technology and Science, Keesara, R.R. Dt.
1.vijay20052009@gmail.com, 2.davidsolomonraju131@gmail.com

Abstract: This project deals with the comparison of the VLSI design of the carry look-ahead adder (CSLA) based 32-bit Unsigned integer multiplier and the VLSI design of the modified carry select adder (CSLA) based 32-bit unsigned integer multiplier. Both the VLSI design of multiplier multiplies two 32-bit unsigned integer values and gives a product term of 64-bit values. The CSLA based multiplier uses the delay time for performing multiplication operation where as in modified CSLA based multiplier also uses nearly the same delay time for multiplication operation. But the area needed for CSLA multiplier is reduced to 40% by the modified CSLA based multiplier to complete the multiplication operation. These multipliers are implemented using modelsim and Xilinx ISE.

Keywords: CSLA; modified CSLA; delay; Area; Array Multiplier; Verilog Modelling & Simulation

I. INTRODUCTION:

MULTIPLICATION is one of the most area consuming arithmetic operations in high-performance circuits. As a consequence many research works deal with low power design of high speed multipliers. Multiplication involves two basic operations, the generation of the partial products and their sum, performed using two kinds of multiplication algorithms, serial and parallel. Serial multiplication algorithms use sequential circuits with feedbacks: inner products are sequentially produced and computed. Parallel multiplication algorithms often use combinational circuits and do not contain feedback structures. Multiplication of two bits produces an output which is twice that of the original bit. It is usually needed to truncate the partial product bits to the required precision to reduce area cost. Fixed-width multipliers, a subset of truncated multipliers, compute only n most significant bits (MSBs) of the 2n-bit product for n × n multiplication and use extra correction/compensation circuits to reduce truncation errors. In previous related papers, to reduce the truncation error by adding error compensation circuits.So that the output will be précised.

Basic building block

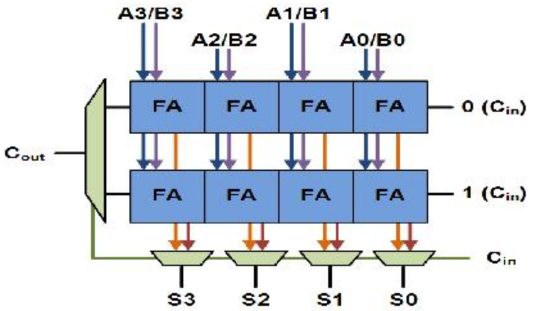

Figure 1: Basic carry Select adder.

Above is the basic building block of a carry-select adder, where the block size is 4. Two 4-bit ripple carry adders are multiplexed together, where the resulting carry and sum bits are selected by the carry-in. Since one ripple carry adder assumes a carry-in of 0, and the other assumes a carry-in of 1, selecting which adder had the correct assumption via the actual carry-in yields the desired result.

Uniform-sized carry select adder

A 16-bit carry-select adder with a uniform block size of 4 can be created with three of these blocks and a 4-bit ripple carry adder. Since carry-in is known at the beginning of computation, a carry select block is not needed for the first four bits. The delay of this adder will be four full adder delays, plus three MUX delays.

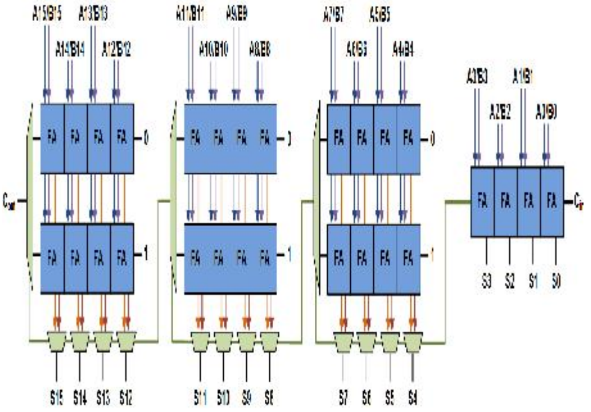

Figure 2: Regular Fixed Size CSLA

Variable-sized carry select adder

A 16-bit carry-select adder with variable size can be similarly created. Here we show an adder with block sizes of 2-2-3-4-5. This break-up is ideal when the full-adder delay is equal to the MUX delay, which is unlikely. The total delay is two full adder delays, and four mux delays.

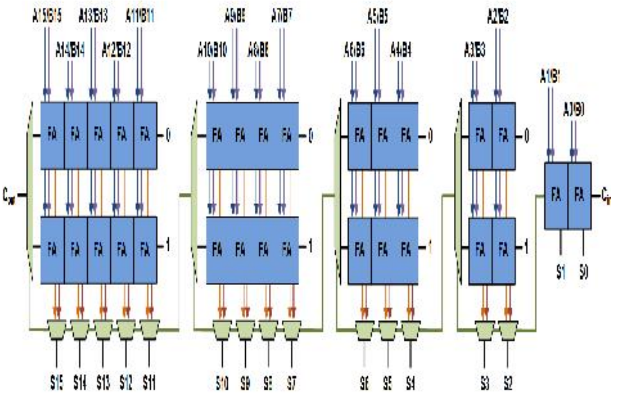

Figure 3: Variable Sized CSLA.

The basic idea of this work is to use Binary to Excess-1 Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption [2]–[4]. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. The details of the BEC logic are discussed in Section III.

This brief is structured as follows. Section II deals with the delay and area evaluation methodology of the basic adder blocks. Section III & IV presents the detailed structure and the function of the BEC logic. The CSLA has been chosen for comparison with the proposed design as it has a more balanced delay, and requires lower power and area. The delay and area evaluation methodology of the regular and modified CSLA are presented in Sections V and VI, respectively. The ASIC implementation details and results are analyzed in Section VII. Finally, the work is concluded in Section VIII.

II. Basic Function and Structure of BEC Logic

The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. We then add up the number of gates in the longest path of a logic block that contributes to the maximum delay. The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA), and FA are evaluated and listed in Table I.

The basic work is to use Binary to Excess-1 Converter (BEC) instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure As stated above the main idea of this work is to use BEC instead of the RCA with Cin=1 in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1-bit BEC is required. A structure and the function table of a 4-bit BEC are shown in Figure.2 and Table .2, respectively.

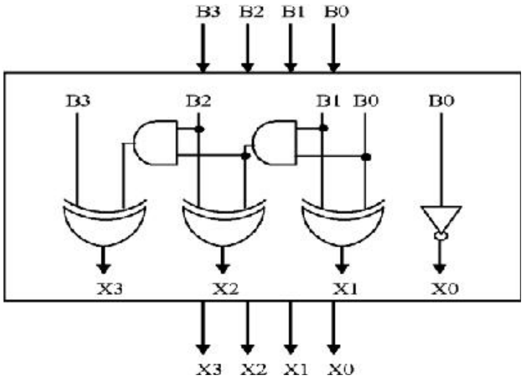

Figure 4: 4 Bit BEC.

The Boolean expressions of the 4-bit BEC is listed as (note the functional symbols ~ NOT, & AND, ^XOR)

X0 = ~B0

X1 = B0^B1

X2 = B2^ (B0 & B1)

X3 = B3^ (B0 & B1 & B2)

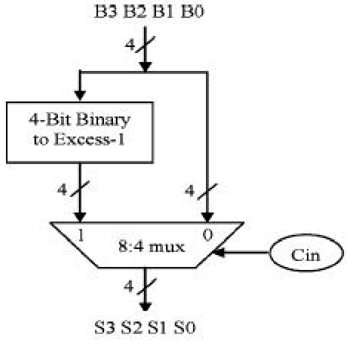

Figure 5: 4-Bit BEC with 8:4 mux.

III. Basic Structure of Regular 16-Bit CSLA

A 16-bit carry select has two types of block size namely uniform block size and variable block size. A 16-bit carry select adder with a uniform block size has the delay of four full adder delays and three MUX delays. While a 16-bit carry select adder with variable block size has the delay of two full adder delays, and four mux delays. Therefore we use 16-bit carry select adder with variable block size. Ripple-carry adders are the simplest and most compact full adders, but their performance is limited by a carry that must ripple from the least-significant to the most-significant bit. A carry-select adder achieves speeds 40% to 90% faster by performing additions in parallel and reducing the maximum carry path.

A carry-select adder is divided into sectors, each of which, except for the least significant performs two additions in parallel, one assuming a carry-in of zero, the other a carry-in of one within the sector, there are two 4-bit ripple- carry adders receiving the same data inputs but different Cin. The upper adder has a carry-in of zero, the lower adder a carry-in of one. The actual Cin from the preceding sector selects one of the two adders. If the carry-in is zero, the sum and carry-out of the upper adder are selected. If the carry-in is one, the sum and carry-out of the lower adder are selected. Logically, the result is not different if a single ripple-carry adder were used.

First the coding for full adder and different multiplexers of 6:3, 8:4, 10:5, and 12:6 was done. Then 2, 3, 4, 5-bit ripple carry adder was done by calling the full adder. The regular 64-bit CSLA was created by calling the ripple carry adders and all multiplexers based on circuit. Finally, regular 128-bit was implemented in the FIR filter design (section5).

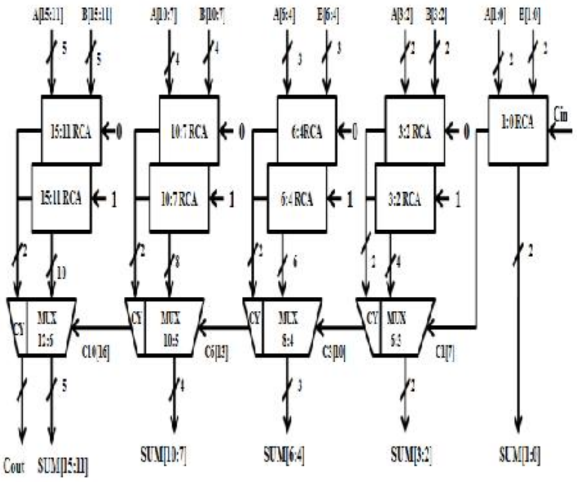

Figure 6: Regular 16-bit CSLA

IV Basic Structure of Modified 16-Bit CSLA

It is similar to regular 16-bit SQRT CSLA. Only change is that in basic blocks having two ripple-carry adders, one ripple carry adder fed with a constant 1 carry-in is replaced by BEC. The area estimation of each group is calculated.

Based on the consideration of delay values, the arrival time of selection input C1 [time (T) =7] of 6:3 mux is earlier than the s3 [t =9] and c3 [t =7] and later than the s2 [t =4]. Thus, the sum3 and final c3 (output from mux) are depending on s3 and mux and partial c3 (input to mux) and mux, respectively. The sum2 depends on c1 and mux. For the remaining parts the arrival time of mux selection input is always greater than the arrival time of data inputs from the BEC's. Thus, the delay of the remaining MUX depends on the arrival time of mux selection input and the mux delay.

First the coding for full adder and multiplexers of 6:3, 8:4, 10:5, and 12:6 was done. The BEC program was design by using NOT, XOR and AND gates. Then 2, 3, 4, 5-bit ripple carry adder was done by calling the full adder. The modified 16-bit CSLA was created by calling the ripple carry adders, BEC and all multiplexers based upon the circuit. Finally, modified 64-bit was implemented in the FIR filter design (section 5).

The arrival time of selection input of 6:3 mux is earlier. Thus, the sum3 and final c3 (output from mux) depends on s3 and mux and partial c3 (input to mux) and mux, respectively. The sum2 depends on c1and mux. For the remaining group's the arrival time of mux selection input is always greater than the arrival time of data inputs from the BEC's. Thus, the delay of the remaining groups depends on the arrival time of mux selection input and the mux delay.

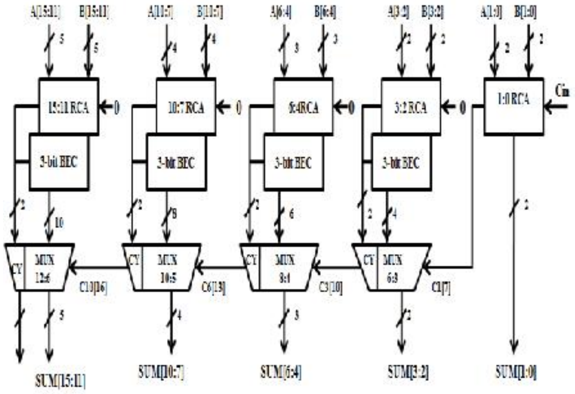

Figure. 7. Modified 16-bit CSLA.

V. Multiplier For Unsigned Data

Multiplication involves the generation of partial products, one for each digit in the multiplier, as in Figure3.These partial products are then summed to produce the final product. The multiplication of two n-bit binary integers results in a product of up to 2n bits in length.

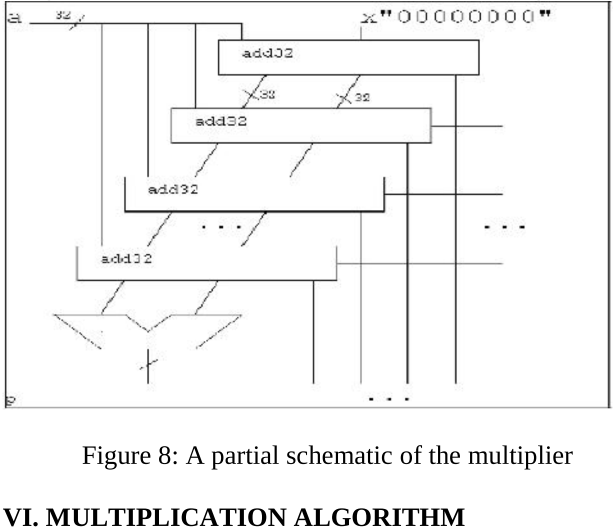

Figure 8: A partial schematic of the multiplier

VI. MULTIPLICATION ALGORITHM

Let the product register size be 64 bits. Let the multiplicand registers size be 32 bits. Store the multiplier in the least significant half of the product register. Clear the most significant half of the product register.Repeat the following steps for 32 times:1. If the least significant bit of the product register is "1"then add the multiplicand to the

most significant half of theproduct register. 2. Shift the content of the product register one bit to the right (ignore the shifted-out bit.) 3. Shift-in the carry bit into the most significant bit of the
product register. Figure 8.Shows a block diagram for such a multiplier.

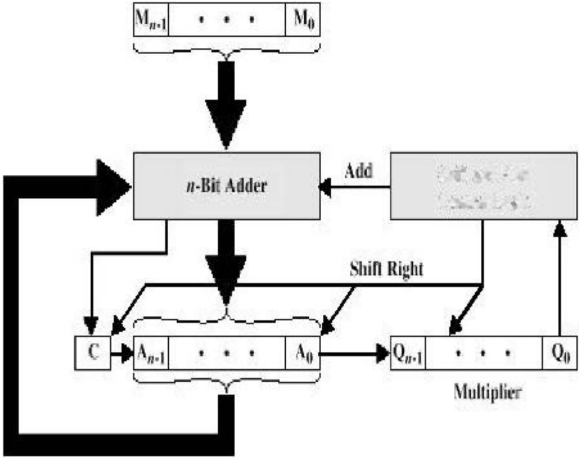

Figure 9: proposed multiplier diagram

VII. IMPLEMENTATION RESULTS

The design proposed in this paper has been developed using Verilog-HDL and synthesized in Xilinx ISE 9.1i. For each word size of the adder, the same value changed dump (VCD) file is generated for all possible input conditions and imported the same to Xilinx ISE 9.1i Power Analysis to perform the power simulations. The similar design flow is followed for both the regular and modified CSLA. the simulation results of both the CSLA structures in terms of delay, area and power.

The multiplier's power and delay product of the proposed 32-bit is lesser than that of the regular CSLA by 20% and the area-delay product is lower by 30%.

The simulation Results of regular and Modified Carry Select adder based multiplier results are shown figure 10 and 11. comapre to regular CSLA modified CSLA architecture is take less amount of delay it is approximately reduced by 30% of regular method design

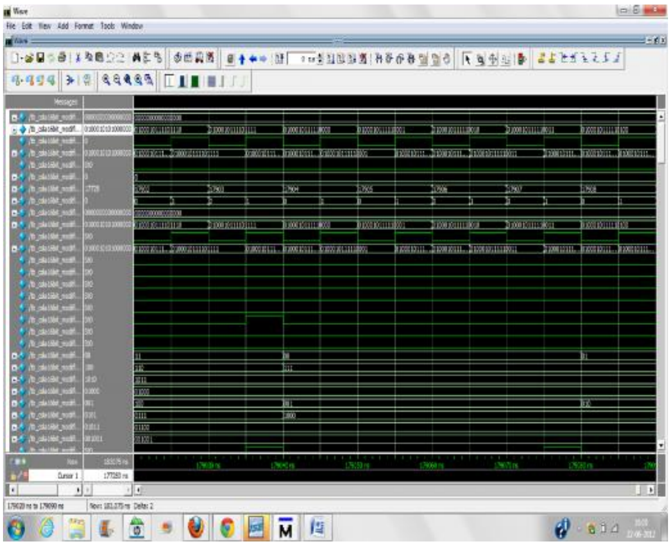

Figure. 10. Regular 16-bit CSLA

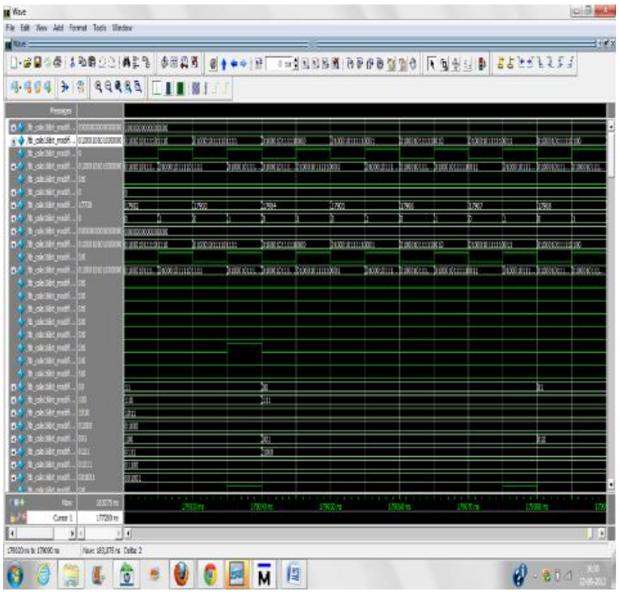

Figure. 11. Modified 16-bit CSLA

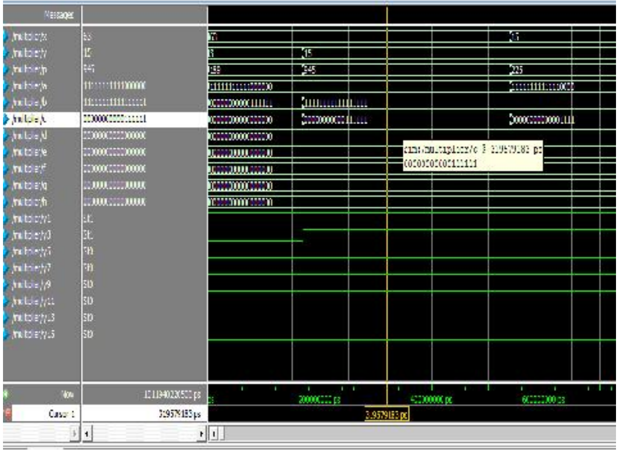

Figure. 11. Modified 32-bit CSLA based multiplier

VII. CONCLUSION

A design and implementation of a Verilog-based 32- bit unsigned multiplier with CSLA and modified CSLA was presented. Verilog Hardware Description Language, was used to model and simulate our multiplier. Using modified CSLA improves the overall performance of the multiplier.
Thus a 30% area delay product reduction is possible with the use of the CSLA based 32 bit unsigned parallel multiplier than CSLA based 32 bit unsigned parallel multiplier.

VIII. FUTURE WORK

This 32 bit multiplier can be further extended to 64 bit multiplier and 128 bit multiplier using the proposed method for multiplication operation can be done as future work.

REFERENCES

[1] Bedrij, O. J., (1962), "Carry-select adder," IRE Trans. Electron. Comput., pp.340–344 .

[2] Ceiang ,T. Y. and Hsiao,M. J. ,(Oct1998),"Carry-select adder using single ripple carry adder," Electron. Lett., vol. 34, no. 22, pp. 2101– 2103

[3] He, Y. , Chang ,C. H. and Gu, J. , (2005), “An A rea efficient 64-bit square root carry-select adder for low power application,” in Proc. IEEE Int. Symp.Circuits Syst. vol. 4, pp. 4082–4085.

[4] Ramkumar,B. , Kittur, H.M. and Kannan ,P. M. ,(2010),“ASIC implementation of modified faster carry save adder,” Eur. J. Sci. Res., vol. 42, no. 1,pp.53–58.

[5] Kim ,Y. and Kim ,L.-S.,(May2001), “64-bit carry-select adder with reduced area,”Electron Lett., vol. 37, no. 10, pp. 614–615.

[6] E. Abu-Shama and M. Bayoumi, “A New cell for low power adders,” in Proc.Int. Midwest Symp. Circuits and Systems, 1995, pp. 1014–1017

[7] www.xilinx.com.

[8] Verilog HDL- Digital Design and Synthesis, by Samir Palnitkar