AN EFFICIENT AND IMPROVED RADIX-16 BASED BOOTH MULTIPLIER FOR 64-BIT BASED ON THE REDUCED HEIGHT OF PARTIAL PRODUCT ARRAY

N. Shehanaz[1] , B.V. Sowjanya[2] , T.C Sanjeeva Rayudu[3]
[1] Adhoc Assistant Professor, [2] Adhoc Assistant Professor, [3] Adhoc Assistant Professor,
[1,2,3] JNTUA College of Engineering (Autonomous), Pulivendula.

Abstract : The reduction of partial product columns height has made to 16 bit using a modified booth multiplier for 64 bit unsigned operations in this proposed method, where as the conventional system has more than 16-bit partial product size. So the height of the array has reduced by one unit in this modified technique. This reduction of one unit adds some properties like increased speed of operation, low power consumption and reduced area. This work can be extended by increasing bit size & fusion techniques.

***Index Terms*: Booth Multiplier (BM), Partial Product Array (PPA)**

I. Introduction

Fast Adders and Multipliers are most important aspects of DSP Systems. The important part of multiplication is partial product generator, which takes more time to generate the result. The multiplication process was ordinarily applied through a sequence of Addition, Subtraction & Shift operations. Multiplication will also be considered as continuous procedure of repeated additions. The number to be delivered is the multiplicand, the number of times that it's introduced is the multiplier, and the effect is the product. The generation of partial products is the resultant of step by step addition. Generally the size of multiplicand & multiplier is same in most of the computers.

Higher radix signed recoding is less popular because the generation of the partial products requires odd multiples of the multiplicand which can't be achieved by means of simple shifts, but require carry-propagate additions. For instance, for radix-16 signed digit recoding the digit set is $\{-8, -7, ..., 0, ..., 7, 8\}$, so that some odd multiples of the multiplicand have to be generated. Specifically, it is required to generate ×3, ×5, and ×7 multiples (×6 is obtained by simple shift of ×3). The generation of each of these odd multiplies requires a two term addition or subtraction, yielding a total of three carry-propagate additions.

II EXISTING DESIGN

The architecture of the basic radix-16 Booth multiplier is shown in below figure. For sake of simplicity, but without loss of generality, an unsigned operands with $n = 64$ bit has explained.

The multiplicand operand 'X' with bit components x_i ($i = 0$ to $n - 1$, with the least-significant bit, LSB, at position 0) and with 'Y' the multiplier operand and bit components y_i. The first step is the recoding of the multiplier operand [8]: groups of four bits with relative values in the set $\{0, 1, ..., 14, 15\}$ are recoded to digits in the set $\{-8, -7, ..., 0, ..., 7, 8\}$ (minimally redundant radix-16 digit set to reduce the number of multiples). This recoding is done with the help of a transfer digit t_i and an interim digit w_i. The recoded digit z_i is the sum of the interim and transfer digits $z_i = w_i + t_i$.

For the set of digits $\{-8, -7, ..., 0, ..., 7, 8\}$, the multiples 1X, 2X, 4X, and 8X are easy to compute, since they are obtained by simple logic shifts. The negative versions of these multiples are obtained by bit inversion and addition of a 1 in the corresponding position in the bit array of the partial products. The generations of 3X, 5X, and 7X (odd multiples) requires carry-propagate adders (the negative versions of these multiples are obtained as before). Finally, 6X is obtained by a simple one bit left shift of 3X.

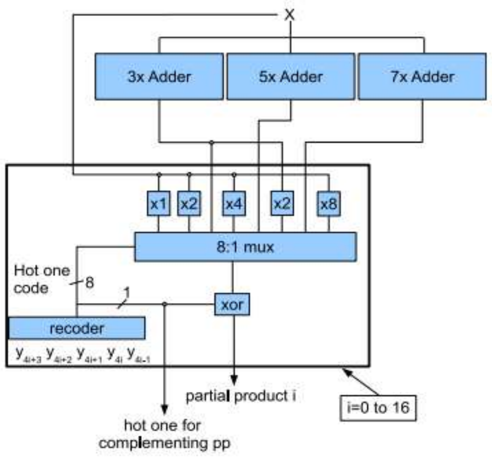

FIGURE 1: Existing system of Partial Product Generation

Figure 1 shows possible implementation of the partial product generation. Five bits of the multiplier Y are used to obtain the recoded digit (four bits of one digit and one bit of the previous digit to determine the transfer digit to be added). The resultant digit is obtained as a one-hot code to directly drive a 8 to 1 multiplexer with an implicit zero output (output equal to zero when all the control signals of the multiplexer are zero). The recoding requires the implementation of simple logic equations that are not in the critical path due to the generation in parallel of the odd multiples (carry-propagate addition). The XOR at the output of the multiplexer is for bit complementation (part of the computation of the two's complement when the multiplier digit is negative).

Figure 2(a) illustrates part of the resultant bit array for n = 64 bit size after the simplification of the sign extension. In general, each partial product has n + 4 bits including the sign in two's complement representation. The extra four bits are required to host a digit multiplication by up to 8 and a sign bit due to the possible multiplication by negative multiplier digits.

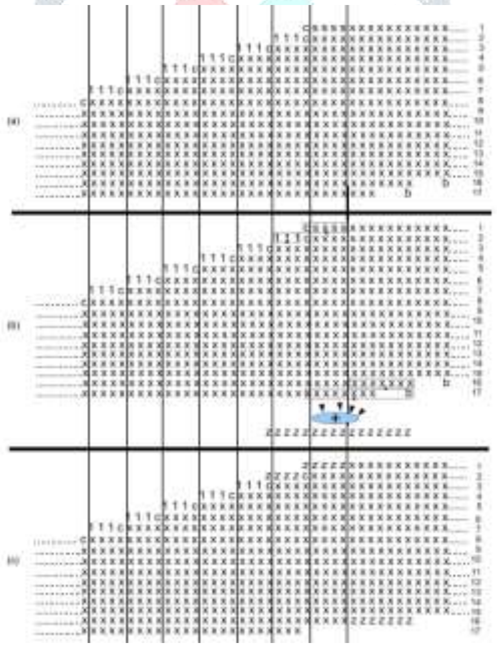

FIGURE 2: Partial Product reduction array of Radix-16 system

Since the partial products are left-shifted four bit positions with respect to each other, a costly sign extension would be necessary. However, the sign extension is simplified by concatenation of some bits to each partial product (S is the sign bit of the partial product and C is S complemented): CSSS for the first partial product and 111C for the rest of partial products (except the partial product at the bottom that is non negative since the corresponding multiplier digit is 0 or 1). The bits denoted by b in Fig. 2 corresponds to the logic 1 that is added for the two's complement for negative partial products.

III PROPOSED DESIGN

Figure 3 shows the specific elements of the bit array to be added by the short carry-propagate addition. In this figure, $p_{i,j}$ corresponds to the bit j of partial product i, s_0 is the sign bit of partial product 0, c_0 = NOT(s_0), bi is the bit for the two's complement of partial product i and z_i is the ith bit of the result of the short addition.

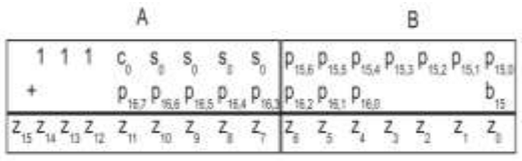
A
B
1 1 1 c_0 s_0 s_0 s_0 s_0
+ $p_{16,7}$ $p_{16,6}$ $p_{16,5}$ $p_{16,4}$ $p_{16,3}$
$p_{15,6}$ $p_{15,5}$ $p_{15,4}$ $p_{15,3}$ $p_{15,2}$ $p_{15,1}$ $p_{15,0}$
$p_{16,2}$ $p_{16,1}$ $p_{16,0}$ b_{15}
z_{15} z_{14} z_{13} z_{12} z_{11} z_{10} z_9 z_8 z_7
z_6 z_5 z_4 z_3 z_2 z_1 z_0

FIGURE 3: Short hand addition procedure

The selection of these specific bits to be added is justified by the fact that, in this way, the short addition delay is hidden from the critical path that corresponds to a regular partial product generation. Here the computation is performed in two concurrent parts A and B. The elements of the part A are generated faster than the elements of part B. Specifically the elements of part A are obtained from, the sign of the first partial product: this is directly obtained from bit y3 since there is no transfer digit from a previous radix-16 digit & bits 3 to 7 of partial product 16: the recoded digit for partial product 16 can only be 0 or 1, since it is just a transfer digit. Therefore the bits of this partial product are generated by a simple AND operations of the bits of the multiplicand X and bit y63.

The implementation of part A as a speculative addition, by computing two results, a result with carry-in = 0 and a result with carry-in = 1. This can be computed efficiently with a compound adder as shown in below figure. Figure 4 shows the implementation of part A. The compound adder determines speculatively the two possible results. Once the carry-in is obtained (from part B), the correct result is selected by a multiplexer.

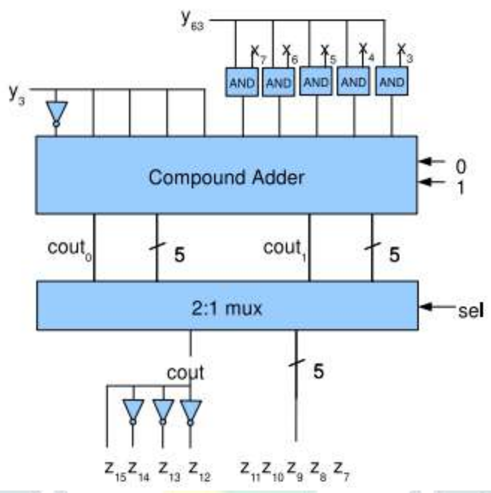
y_{63}
x_7 x_6 x_5 x_4 x_3
AND
y_3
Compound Adder
0
1
$cout_0$
5
$cout_1$
5
2:1 mux
sel
cout
5
z_{15} z_{14} z_{13} z_{12}
z_{11} z_{10} z_9 z_8 z_7

FIGURE 4: Proposed design of Speculative addition of Part A

The computation of part B is more complicated. The main issue is that we need the 7 least-significant bits of partial product 15. Of course waiting for the generation of partial product 15 is not an option since we want to hide the short addition delay out of the critical path. We decided to implement a specific circuit to embed the computation of the least-significant bits of partial product 15 in the computation of part B (and also the addition of the bit b15). Note that for the method to be correct the computation of the partial product embedded in part B should be consistent with the regular computation performed for the most significant bits of partial product 15.

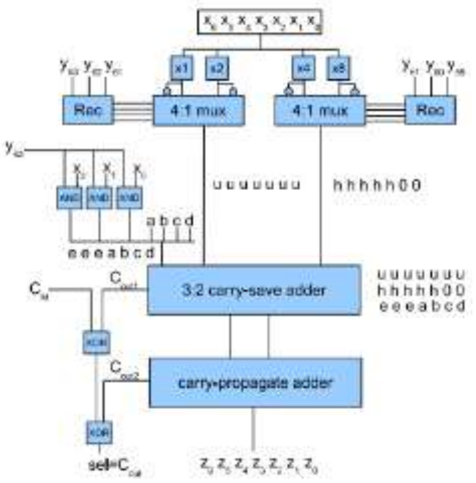

FIGURE 5: Proposed design of Computation of Part B

Figure 5 shows the computation of part B. We decided to compute part B as a three operand addition with a 3:2 carry save adder and a carry-propagate adder. Two of the operands correspond to the least-significant bits of the partial product 15 and the other operand corresponds to the three least-significant bits of partial product 16 (that are easily obtained by an AND operation).

Figure 6 shows the recoding and partial product generation stage including the high level view of the hardware scheme proposed. The way we compute part B may still lead to an inconsistency with the computation of the most significant part of partial product 15. Specifically, when partial product 15 is the result of an odd multiple, a possible carry from the 7 least-significant bits is already incorporated in the most significant part of the partial product. During the computation of part B we should not produce again this carry.

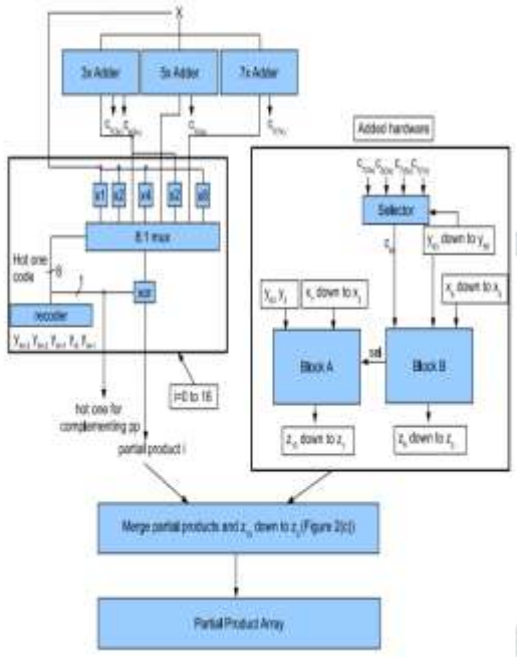

FIGURE 6: Proposed Scheme along with Part A and Part B

IV. SIMULATION RESLUT

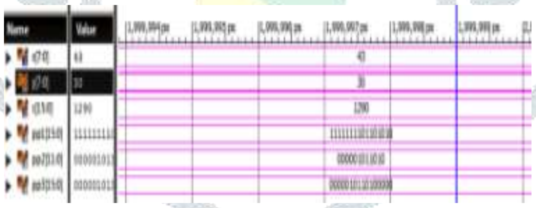

FIGURE 7: Simulation result of Proposed system

V.CONCLUSION

Finally the presented method effectively used to reduce by one the maximum height of the partial product array for 64-bit radix-16 Booth recoded magnitude multipliers. This reduction may allow more flexibility in the design of the reduction tree of the pipelined multiplier. We have shown that this reduction is achieved with no extra delay for n ≥ 32 for a cell-based design. The method can be extended to Booth recoded radix-8 multipliers, signed multipliers and combined signed/unsigned multipliers. Radix-8 and radix-16 Booth recoded multipliers are attractive for low power designs, mainly to the lower complexity and depth of the reduction tree.

VI. REFERENCES

[1] M. Daumas and D. W. Matula, "A Booth multiplier accepting both a redundant or a non redundant input with no additional delay," in Proc. IEEE Int. Conf. on Application-Specific Syst., Architectures, and Processors, pp. 205–214, 2000.
[2] S. Kuang, J. Wang, and C. Guo, "Modified booth multipliers with a regular partial product array," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 5, pp. 404–408, May 2009.
[3] S. Xydis, I. Triantafyllou, G. Economakos, and K. Pekmestzi, "Flexible datapath synthesis through arithmetically optimized operation chaining," in Proc. NASA/ESA Conf. Adaptive Hardware Syst., pp. 407–414, 2009.

[4] F. Lamberti et al., “Reducing the computation time in (short bit-width) twos complement multipliers,” IEEE Trans. Comput., vol. 60, no. 2, pp. 148–156, Feb. 2011.
[5] N. Petra et al., “Design of fixed-width multipliers with linear compensation function,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 5, pp. 947–960, May 2011.
[6] S. Galal et al., “FPU generator for design space exploration,” in Proc. 21st IEEE Symp. Comput. Arithmetic (ARITH), Apr. 2013, pp. 25–34.
[7] K. Tsoumanis et al., “An optimized modified booth recoder for efficient design of the add-multiply operator,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 1133–1143, Apr. 2014.

AUTHORS PROFILE

N. SHEHANAZ received B.Tech degree in ECE from Rajooli Veera Reddy Padmaja Engineering College for Women, C.K.Dinne, Kadapa in the year 2012 and M.Tech degree in VLSI System Design from Annamacharya Institute of Technology and Sciences, Kadapa in 2016. Her research interests include VLSI Chip Design, Communications and Robotics.

B.V. Sowjanya completed her B.Tech in ECE from Kuppam College of Engineering in the year 2011 and M.Tech degree in DECS from JNTUA college of Engineering, Pulivendula in 2014. Her research interests include Communications and Embedded Systems.

T.C Sanjeeva Rayudu did his B.Tech in ECE from PBR VITS, Kavali, Nellore in 2009 and M.Tech VLSI System Design from SVNE, Rangampeta, Tirupati in 2011. His area of research interest includes VLSI and Communications.