International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

An Encoder Based Radix -16 Booth Multiplier for Improving Speed and Area Efficiency

Nutakki Vijaya Kumari & K. Sampath Singh

[1]PG Student,Dept.of ECE,Universal College of Engg & Tech. Perecherla, Guntur, A P, India 522438.

[2]Assistant Professor, ECE Dept. Universal College of Engg & Tech. Perecherla, Guntur, A P, India 522438.

ABSTRACT: *In this paper, we describe an optimization for binary radix-16 (modified) Booth recoded multipliers to reduce the maximum height of the partial product columns to [n/4] for n = 64-bit unsigned operands. This is in contrast to the conventional maximum height of [(n + 1)/4]. Therefore, a reduction of one unit in the maximum height is achieved. This reduction may add flexibility during the design of the pipelined multiplier to meet the design goals, it may allow further optimizations of the partial product array reduction stage in terms of area/delay/power and/or may allow additional addends to be included in the partial product array without increasing the delay. The method can be extended to Booth recoded radix-8 multipliers, signed multipliers, combined signed/unsigned multipliers, and other values of n. The proposed architecture of this paper analysis the delay and area using Xilinx 14.3.*

Index Terms—Binary multipliers, modified Booth recoding, radix-16.

I.INTRODUCTION

Binary multipliers are a widely used building block elementin the design of microprocessors and embeddedsystems, and therefore, they are an important target for implementation optimization. Current implementations ofbinary multiplication follow the steps:

1) Recoding of the multiplier in digits in a certain number system;

2) Digitmultiplication of each digit by the multiplicand, resulting in acertain number of partial products;

3) Reduction of the partialproduct array to two operands using multi-operand additiontechniques; and

4)Carry-propagate addition of the two operandsto obtain the final result.

The recoding type is a key issue, since it determines thenumber of partial products. The usual recoding process recodesa binary operand into a signed-digit operand with digitsin a minimally redundant digit set. Specifically, forradix-r ($r = 2m$), the binary operand is composed of non-redundantradix-r digits (by just making groups of m bits). Radix-4 modified Booth is a widely used recoding method that recodes a binary operand into radix-4 signed digits in theset $\{-2, -1, 0, 1, 2\}$. This is a popular recoding since the digitmultiplication step to generate the partial products only requiressimple shifts and complementation. The resulting number ofpartial products is about $n/2$.

Higher radix signed recoding is less popular because thegeneration of the partial products requires odd multiples ofthe multiplicand which cannot be achieved by means ofsimple shifts, but require carry-propagate additions. However, the advantage of the high radix is that the numberof partial products is further reduced. For instance,

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

for radix-16and *n*-bit operands, about *n*/4 partial products are generated.Although less popular than radix-4, there exist industrial instances of radix-8 and radix-16 multipliers inmicroprocessors implementations.

II.EXISTED SYSTEM

Fig. 1 shows a possible implementation of the partialproduct generation. Five bits of the multiplier Y are used toobtain the recoded digit (four bits of one digit and one bit ofthe previous digit to determine the transfer digit to be added).

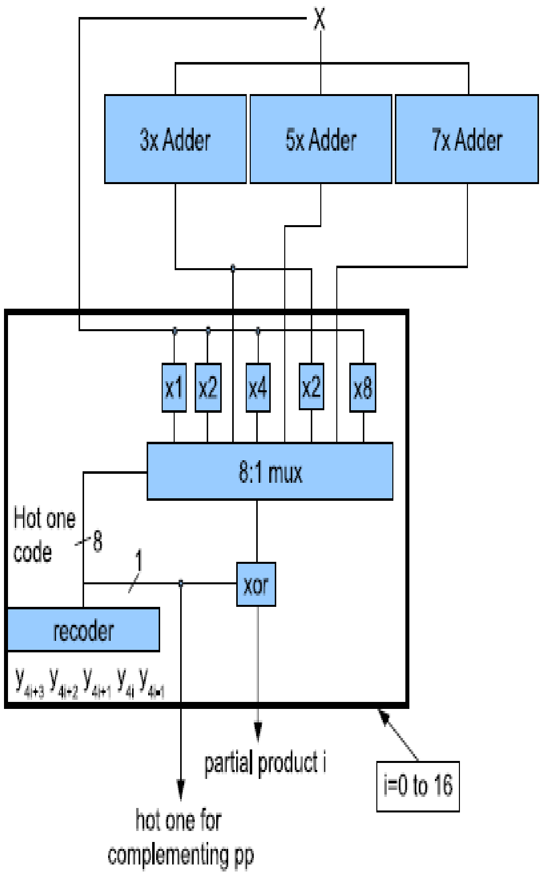


Fig 1. Existed system.

The resultant digit is obtained as a one-hot code to directlydrive a 8 to 1 multiplexer with an implicit zero output (outputequal to zero when all the control signals of the multiplexer arezero).

The recoding requires the implementation of simple logicequations that are not in the critical path due to the generationin parallel of the odd multiples (carry-propagate addition). TheXOR at the output of the multiplexer is for bit complementation(part of the computation of the two’s complement when themultiplier digit is negative).

In general, each partial product has *n* + 4 bits including thesign in two’s complement representation. The extra four bits arerequired to host a digit multiplication by up to 8 and a sign bitdue to the possible multiplication by negative multiplier digits.Since the partial products are left-shifted four bit positionswith respect to each other, a costly sign extension would benecessary. However, the sign extension is simplified by concatenationof some bits to each partial product.

After the generation of the partial product bit array, the reduction(multi-operand addition) from a maximum height of 17(for *n* = 64) to 2 is performed. The methods for multioperandaddition are well known, with a common solution consisting ofusing 3 to 2 bit reduction with full adders (or 3:2 carry-saveadders) or 4 to 2 bit reduction with 4:2 carry-save adders.The delay and design effort of this stage are highly dependenton the maximum height of the bit array.

III.PROPOSED SYSTEM

BOOTH ALGORITHM: Booth multiplication algorithm or Booth algorithm was named after the inventor Andrew Donald Booth. It can be defined as an algorithm or method of multiplying binary numbers in two's complement notation. It is a simple method to multiply binary numbers in which multiplication is performed with repeated addition operations by following the booth algorithm. Again this booth algorithm for multiplication operation is further modified and hence, named as modified booth algorithm.

A multiplier generator that creates a smaller number of partial products will allow the partial product summation to be efficient and use less hardware. The simple multiplication generator can be extended toreduce the number of partial products by groupingthe bits of the multiplier into pairs, and selecting thepartial products from the set of 0, M, 2M or theircomplements, where M is the multiplicand. Thisreduces the number of partial products, by a factortwo but also generates some extra-bits for the signextension and the 2's complementation. All partial products set can be producedusing simple shifting and complementing.

Themultiplier is partitioned into overlapping groups of 3bits, and each group is decoded to select a singlepartial product as per the selection table 1 shownbelow. Each partial product is shifted 2 bit positionswith respect to its neighbors. The number of partialproducts has been reduced to half of total number ofmultiplier bits. In general there will be n/2 products,where n is the operand length. The multiply by 2 canbe obtained by a simple left shift of the multiplicandand negative of number obtained from its two'scomplement form. Following table shows booth encoding table.

According to that partial products are generated andadded to get final result.

MODIFIED BOOTH ALGORITHM: Booth multiplication algorithm consists of three major steps as shown in the structure of booth algorithm figure that includes generation of partial product called as recoding, reducing the partial product in two rows, and addition that gives final product.

Modified Booth Algorithm Encoder: This modified booth multiplier is used to perform high-speed multiplications using modified booth algorithm.

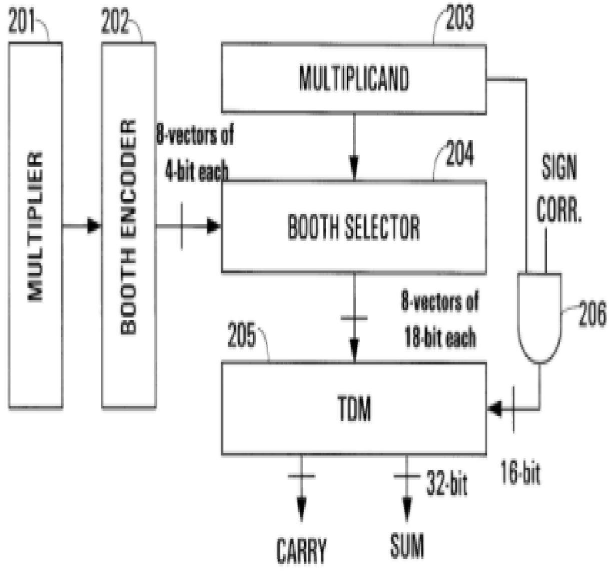


Fig 2. Proposed system.

This modified booth multiplier's computation time and the logarithm of the word length of operands are proportional to each other. We can

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

reduce half the number of partial product. Radix-4 booth algorithm used here increases the speed of multiplier and reduces the area of multiplier circuit. In this algorithm, every second column is taken and multiplied by 0 or +1 or +2 or -1 or -2 instead of multiplying with 0 or 1 after shifting and adding of every column of the booth multiplier. Thus, half of the partial product can be reduced using this booth algorithm. Based on the multiplier bits, the process of encoding the multiplicand is performed by radix-4 booth encoder.

The overlapping is used for comparing three bits at a time. This grouping is started from least significant bit (LSB), in which only two bits of the booth multiplier are used by the first block and a zero is assumed as third bit.

IV.SIMULATIONRESULTS

The below figures shows the simulation results of an encoder based radix-16booth multiplier for improving speed and area efficiency.The proposed is designed an encoder based radix-16 booth multiplier for improvingspeed and area efficiency in XILINX 14.7 Using VERILOG HDL code andsimulated using Modelsim 6.5e To evaluate the efficiency of the proposedarchitecture.

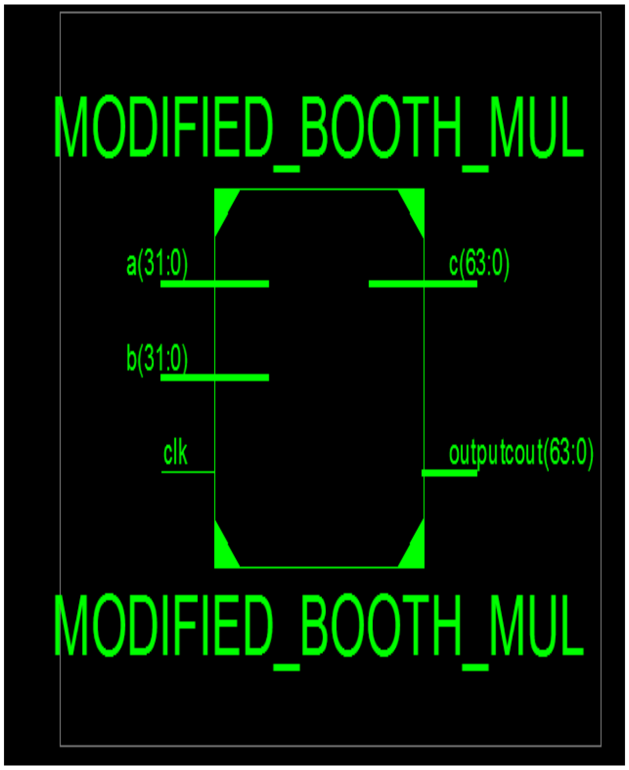
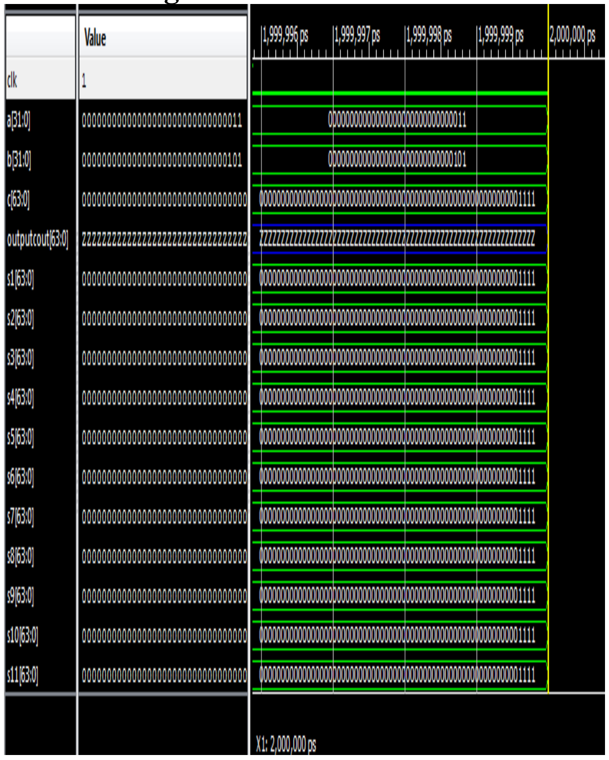


Fig 3: RTL schematic view.



```
                Value
clk             1
a[31:0]         00000000000000000000000000000011
b[31:0]         00000000000000000000000000000101
c[63:0]         0000000000000000000000000000000000
outputcout[63:0] ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
s1[63:0]        0000000000000000000000000000000000
s2[63:0]        0000000000000000000000000000000000
s3[63:0]        0000000000000000000000000000000000
s4[63:0]        0000000000000000000000000000000000
s5[63:0]        0000000000000000000000000000000000
s6[63:0]        0000000000000000000000000000000000
s7[63:0]        0000000000000000000000000000000000
s8[63:0]        0000000000000000000000000000000000
s9[63:0]        0000000000000000000000000000000000
s10[63:0]       0000000000000000000000000000000000
s11[63:0]       0000000000000000000000000000000000

1,999,996 ps  1,999,997 ps  1,999,998 ps  1,999,999 ps  2,000,000 ps
X1: 2,000,000 ps
```

Fig 4: Simulation results.

V.CONCLUSION

Pipelined large wordlength digital multipliers are difficult to design under the constraints of core cycle time,pipeline depth, power and energy consumption and area.Low level optimizations might be required to meet these constraints. we have presented a method 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.

VI. FUTURE SCOPE

As an attempt to develop arithmetic algorithm and architecture level optimization techniques for low-power multiplier design, the research presented in this dissertation has achieved good results and demonstrated the efficiency of high level optimization techniques. However, there are limitations in our work and several future research directions are possible. Higher-radix recoding further reduces the number of PPs and thus has the potential of power saving. Another possible direction can be representation of Arguments such as in sign-magnitude or 2's compliment form which in any case would prove better according to situation and require less power and consume less time.

VII.REFERENCES

[1] S. Kuang, J. Wang, and C. Guo, "Modified booth multipliers with aregular partial product array," *IEEE Trans. Circuits Syst. II, Exp. Briefs*,vol. 56, no. 5, pp. 404–408, May 2009.

[2] 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.

[3] N. Petra *et al.*, "Design of fixed-width multipliers with linear compensationfunction," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 5,pp. 947–960, May 2011.

[4] S. Galal*et al.*, "FPU generator for design space exploration," in *Proc. 21st IEEE Symp. Comput. Arithmetic (ARITH)*, Apr. 2013, pp. 25–34.

[5] K. Tsoumanis*et al.*, "An optimized modified booth recoder for efficientdesign of the add-multiply operator," *IEEE Trans. Circuits Syst. I, Reg.Papers*, vol. 61, no. 4, pp. 1133–1143, Apr. 2014.

[6] A. Cilardo*et al.*, "High speed speculative multipliers based on speculativecarry-save tree," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 12,pp. 3426–3435, Dec. 2014.

[7] M. Ercegovac and T. Lang, *Digital Arithmetic*. Burlington, MA, USA:Morgan Kaufmann, 2004.

[8] S. Vassiliadis, E. Schwarz, and D. Hanrahan, "A general proof foroverlapped multiple-bit scanning multiplications," *IEEE Trans. Comput.*,vol. 38, no. 2, pp. 172–183, Feb. 1989.