ISSN: 2456-043X

International Journal of Engineering Science and Generic Research (IJESAR)
Available Online at www.ijesar.in
Volume 3; Issue 5; September-October; 2017; Page No. 07-10

Design of modified Radix-10 parallel multiplier

[1]K.Prasanthi, [2] Mrs.P.AsiyaThapaswin

[1] PG Scholar, Vaagdevi Institute of Technology and Science, Kadapa (dist), AP. India.

[2]Assistant Professor, Vaagdevi Institute of Technology and Science, Kadapa (dist), AP. India.

Abstract

The importance of multiplication in various applications necessitates improvement in its design so as to obtain the multiplication result efficiently. Multiplication operation can be improved by reducing the number of partial products to be added and by enhancing the adder unit for obtaining sum. The number of partial products can be reduced by using higher radix multiplication. For better speed applications a radix-10 multiplier is proposed which uses recoded multiplier digits as in conventional parallel multiplier design. The multiplier digits are encoded using Signed Digit (SD) radix-10 method which converts the digit set to {-5 to 5} from {0 to 9} and also generate a sign bit. This recoding leads to minimized calculations as only five multiples are required to be calculated and the negative multiples are obtained using 2's complement approach. The modified architecture eliminates the extra recoding logic thereby reducing the area of overall architecture. This paper delivers the design and implementation of 16-Bit multiplication unit. The design entry is done in Verilog Hardware Description Language (HDL) and simulated using ISIM Simulator. It is synthesized and implemented using Xilinx ISE 14.7.

Keywords: Radix-10 parallel multiplication; Recoded multiplier; BCD multiplier.

INTRODUCTION

High speed computing has become an expected norm for the average user. This has led to increased research efforts in enhancing computing capabilities of a digital circuit. Multipliers are the key components of many high performance systems such as microprocessors, digital signal processors and many countless applications. Multiplication comprises of two stages: Evaluation of partial products and the addition of evaluated products to obtain the final product. These two stages require an efficient adder in order to implement multipliers.

Multiplication can be done in sequential [1], [2] or in a parallel manner [3], [4], [5]. Parallel computation involves generation of multiplicand multiples in parallel fashion which improves the computation speed. Certain architectures proposed for parallel implementation of multipliers requires recoding of multiplicand and multiplier [8] in redundant BCD codes which requires extra recoding logic. To overcome the extra circuitry of recoding and complexity of multiplier design, a methodology is put forth for obtaining multiples and final output in an easy manner. This paper delivers the design of a multiplication method which uses three main components for obtaining the correct multiplication product. The three components utilized for computing are:

- Multiples evaluating unit.
- Multiplier decoding unit.
- Multiple accumulation unit.

Multiples evaluation unit calculates the multiples of the multiplicand. Multiplier decoding unit decodes the multiplier digit in order to select appropriate multiples of multiplicand. Multiplication accumulation unit gather the partial products obtained and provide the final product of multiplication. There are various decoding approaches, some partition each multiplier digit into two parts and then the intermediate products thus obtained are added by taking appropriate values from digit set defined [6] [7]. Some algorithms require subtraction to evaluate the partial product from the intermediate products [3]. In the proposed model, as shown in figure 1, the multiples generated are {A, 2A, 3A, 4A, 5A}, considering A as the multiplicand and B as the multiplier. The multiplier digit is recoded using signed digit (SD) radix-10 recoder [8].

Corresponding author: K.Prasanthi

A modified BCD adder [9] is used while computing the multiples of the multiplicand and during their accumulation. Multiplexer is used to select the appropriate multiples according to the recoded value of multiplier digit. Figure 2 shows the various architectural components of the modified radix-10 parallel multiplication unit. The inputs to the structure are BCD digits depicted by A and B of length 4n. Here n represents the number of digits and 4nrepresents the number of bits, since each digit corresponds to four bits.

RADIX-10 PARALLEL DECIMAL MULTIPLIER

As stated earlier the importance of efficient implementation of multiplication unit, there are various measures taken to improve the methods of partial product generation and their accumulation. In this sequence an improved version of parallel decimal multiplication was proposed [8], [10] which dealt with the BCD--4221 and BCD--5211 codes to calculate partial products and for their accumulation. The input digits A and B in radix-10 parallel decimal multiplier [10] are assumed to be decimal values having d digits and 4d bits. The architecture is shown in figure 1.

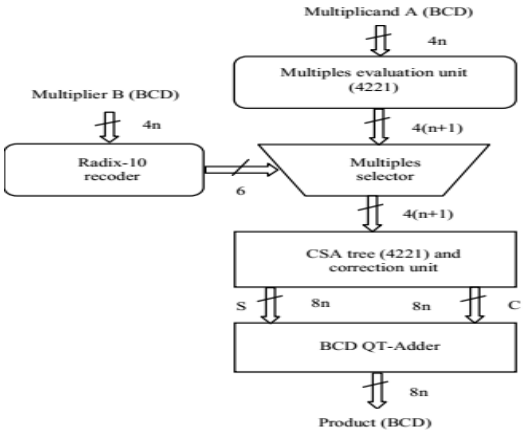

Figure 1: Radix-10 Parallel Multiplier

The architecture comprises of following levels:

- Evaluation of decimal partial products.
- Reduction of evaluated products.
- BCD carry-propagate addition.

Partial products are generated from SD radix-10 encoded multiplier and multiples are calculated in 4221 codes. Each encoded d digit selects the appropriate multiple using the SD radix-10 encoded bits. The radix-10 recoding recodes the digit set in B from {0 to 9} to {-4 to 5} and also provide a sign bit variable that depicts if B isin the range of {5 to 9}. The recoded signals from multiplier digit B are given by following equations:

Bsi = bi3 | bi2 & (bi1 | bi0) (1)

B5i =bi2 & ~bi1 & (bi0 ^ bsi-1) (2)

B4i = bsi-1 & bi0 & (bi2 ^ bi1) | ~bi-1 & bi2 & ~bi0 (3)

B3i = bi1 & (bi0 ^ bsi-1) (4)

B2i = ~bsi-1 & ~bi0 & (bi3 | ~bi2 & bi1) | bsi-1 & ~bi3& bi0 &~ (bi2 ^ bi1) (5)

B1i = ~(bi2 |bi1) & (bi0 ^ bsi-1) (6)

For each digit a 5 bit code along with a sign bit is generated. In the equations "bij",j denotes the jth bit of ith digit and the value of i, j starts from 0. Similarly bsi denotes the sign bit of ithdigit , which depicts whether the digit Bi of multiplier B is equivalent or more than 5. {B5i B4i B3i B2i B1i } in the equations set represent the 5 bit encodings obtained from a 4 bit BCD input. Characters ~ ,& , | and ^ are the symbolic representations for logical operation NOT, AND , OR and XOR respectively.

The recoded digits select the appropriate multiple using the multiplexer. The multiples are XORed so that the multiples for negative recoded digits can be obtained. The partial product are then reduced to two digits by utilizing CSA tree. For final product calculation S is obtained by recoding of 4221 coded digits to 5211 then 1 left shift and C is converted to BCD excess-6. Finally S and C are added using 2-d digit BCD Quaternary Tree(QT) Adder.

MODIFIED RADIX-10 PARALLEL MULTIPLIER

Decimal arithmetic is supported due to its ease in usage and also because in some representation the conversion from decimal to binary is unacceptable. As a result BCD being hardware oriented, emerges as an effective way of decimal representation .The proposed multiplier uses radix-10 architecture [10] but it eliminates extra recoding logic. The modified architecture is proposed to eliminate the complexity of partial products generation and their accumulation in the previous model.

The new architecture deals with easier calculation of multiples and also their addition for final product. However the multiplier recoding of the modified architecture is similar to that of the conventional model. The utilization of the sign bit in proposed method is done in a different manner. The elimination of extra recoding blocks in modified architecture leads to reduction in area. Another advantage of the proposed architecture is that it uses BCD codes instead of 4221 or 5211 codes which are redundant in nature. The redundancy of these codescauses a digit to have more than one corresponding coding that again puts the user in

dilemma which coding out of the two to choose from. These recoding being time consuming also put an extra area overhead on the architecture of the multiplication hardware model due to recoding logic blocks.

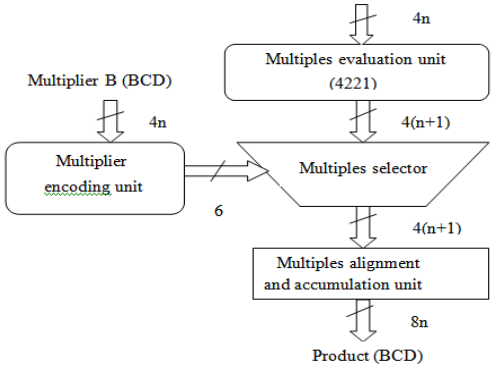

Figure 2: Modified Radix-10 Parallel Multiplier

A. Multiples Evaluation Unit

While computing the multiplication product, the essential requirement is of pre computation of the multiplicand multiples. In the proposed multiplication method the multiples {A, 2A, 3A, 4A, 5A} are calculated by using modified BCD adder. 3A is calculated by BCD addition of A and 2A, similarly all the multiples are calculated. Modified BCD adder [9] is utilized so as to eliminate the post corrections required in conventional BCD adder.

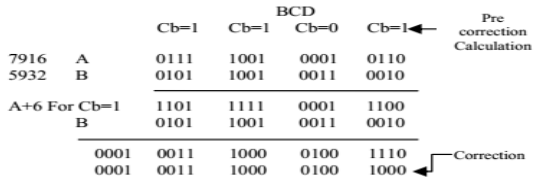

Figure 3: Multiples calculation using modified BCD adder

The conventional BCD addition is calculated by binary carry propagation addition. This addition requires post sum correction due to which the multi operand BCD addition becomes more complex and cumbersome. For more number of digits the BCD carry chain adder is used and to skip the unused combinations a correction factor is added. Although sometimes due to correction, carry may be generated that requires correction. In modified BCD adder pre-correction stage [9] is used instead of post correction stage as depicted by figure 3 In pre-correction step, first it is determined whether the digits A and B added up to greater than-equal to eight or not. If their summation is greater than eight only then the addition of 0110 is done for correcting sum. In modified adder the invalid digit combinations are also used. The unused digits 1110, 1111 are considered 8 and 9 respectively which arises when correction is added in excess due to carry output [9].

B. Multiplier Decoding Unit

In order to obtain partial products, the multiplier digit is recoded if it is greater than five. Digits lying within the range 6 to 9 are encoded to -4 to -1. Thus the encoded digit set lies between -5 to 5, thereby named as radix-10 recoding. Each digit generates a five bit code and a sign bit to select one multiple out of five which were generated by multiple evaluation stage. Sign bit determines whether the digit is greater than or equal to five. Table I shows the decoding unit and an example of the decoding five, then the digit at adjacent most significant bit (MSB) is increased by one. As depicted in table I the digit eight is encoded as minus two, represented by digit with a bar and the digit next to it is incremented to eight from seven. Similarly for all the digits the sign bit is taken into consideration while encoding. Multiplexer acts as the selection unit which determine the correct multiple in accordance with the encoded bit set. It selects the positive multiples when the sign bit is zero whereas when sign bit is one then the negative multiple is chosen. In case when sign bit is one then the value of sign bit for other digits is also considered while determining the correct partial product.

C. Multiple Accumulation Unit

The Partial products obtained after the use of multiple evaluation unit and encoding unit are aligned accordingly and added using modified BCD adders. For multiplication of two n digit numbers the partial products obtained are of each n+1 digit length and the final product may be of length 2n. Thus the partial products are left shifted and then added to obtain the desired product.

RESULTS:

In this chapter we have to discuss the simulation results for Radix-10 Parallel Multiplier and Modified Radix-10 parallel multiplier

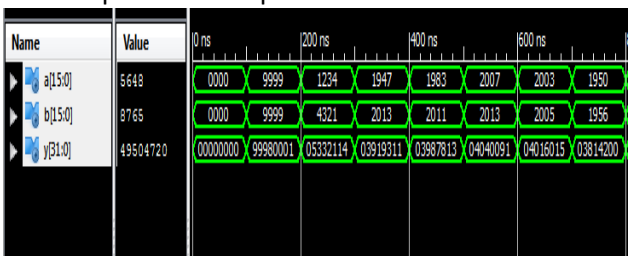

Fig Radix 3-16 bit simulation wave forms

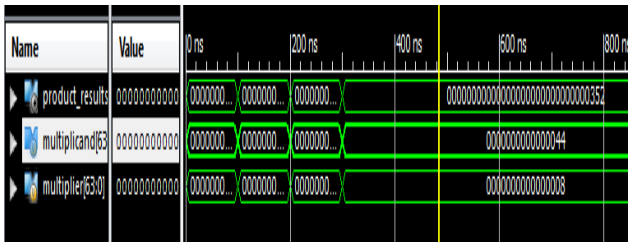

Fig. Radix 4-64 bit simulation wave forms

CONCLUSION

In this paper the design and implementation of radix-10 parallel decimal multiplier using modified BCD adder is shown for area-efficient applications. The design reduces the complexity of the previous designs which required extra recordings and also reduces the area by eliminating the recoding used for multiple generation.

By introducing modified BCD adder in multiplication architecture better performance is obtained in terms of speed and also it reduces various post correction steps for more number of partial products by implementing a precorrection step. This work can be extended for the implementation of 32 bit, 64 bit and 132 bit multiplication and there is a scope for VLSI application.

REFERENCES

1. F.Y. Busaba, C.A. Krygowski, W.H. Li, E.M. Schwarz, and S.R. Carlough, “The IBM z900 Decimal Arithmetic Unit,” Proc. Conf. Record of the Asilomar Conf. Signals, Systems and Computers, vol. 2, pp. 1335-1339, Nov. 2001.
2. F.Y. Busaba, T. Slegel, S. Carlough, C. Krygowski, and J.G. Rell, “The Design of the Fixed Point Unit for the z990 Microprocessor,” Proc. 14th ACM Great Lakes Symp.VLSI 2004, pp. 364-367, Apr.2004.
3. T. Lang and A. Nannarelli, “A Radix-10 Combinational Multiplier,” Proc. 40th Asilomar Conf. Signals, Systems, and Computers, pp. 313-317, Oct. 2006.
4. L. Dadda and A. Nannarelli, "A Varient of a Radix-10 Combinational Multiplier," in IEEE International Symposium on Circuits and Systems, ISCAS 2008, Seattle, Washington, USA, pp. 3370-3373, May 2008.
5. L. Dadda, “Multioperand parallel decimal adder: A mixed binary and BCD approach,” IEEE Trans. Comput., vol. 56, no. 10, pp. 1320–1328, Oct. 2007.
6. M.A. Erle, M.J. Schulte, and E.M. Schwarz, "Decimal Multiplication with Efficient Partial Product Generation," 17th IEEE Symposium on Computer Arithmetic, IEEE Computer Society, pp. 21–28, Jun. 2005.
7. M.J. Schulte and M.A. Erle, "Decimal Multiplication Via Carry-Save Addition," in IEEE International Conference on Applications-Specific Systems, Architectures, and Processors, pp. 348-358, 2003.
8. Vazquez, E. Antelo, and P. Montuschi, "A New Family of High-Performance Parallel Decimal Multipliers," in 18th IEEE Symposium on Computer Arithmetic, Montpellier, France, pp. 195-204, Jun. 2007.
9. Vazquez and E. Antelo "VLSI Design of Combined Binary/Decimal Multioperand Adders", INRIA Research Report, Nov. 2010.
10. A. Vazquez, E. Antelo, and P. Montuschi, "Improved Design of High Performance Parallel Decimal Multipliers," IEEE Transactions on Computers, vol. 59, no. 5, pp. 679-693, May 2010.
11. M. E. Ingle and T. Panse, "Radix-10 Parallel Decimal Multiplier," International Journal of Electronics Signals and Systems (IJESS) ISSN: 2231- 5969, vol.1 Iss. 3, 2012
12. R.D. Kenney and M.J. Schulte, “High-Speed Multioperand Decimal Adders,” IEEE Trans. Computers, vol. 54, no. 8, pp. 953- 963, Aug. 2005.

AUTHOURS DETAILS:

K.Prasanthi. is pursuing her M.Tech in VLSI System Design from Vaagdevi Institute of Technology and Science (VITS), J.N.T.U.A affiliated college.

Mrs.P.AsiyaThapaswin is an Associate Professor at Vaagdevi Institute of Technology and Science (VITS),