ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 2, Issue 11, November 2013

Floating point multiplier using Canonical Signed Digit

D. Harini Sharma, Addanki Purna Ramesh

Abstract— **Digital signal and image processing applications require a large number of floating point multiplications. For such applications fast multiplication techniques are required to improve the overall system speed. Fast multiplication can be achieved by reducing number of partial products. Canonical signed digit is a recoding technique, which recodes a number with minimum number of non-zero digits. As the number of partial products depends on the number of non-zero digits, by using Canonical recoding, the number of non-zero digits will be reduced, thereby reducing the number of partial products. In this paper, Double-precision floating point multiplication using canonical signed digit is proposed and is compared with Conventional multiplication technique. The design is implemented in Verilog and simulated using Xilinx 9.2 ISE. The results showed that CSD outperforms other algorithm when delay is primary concern.**

.

Index Terms— Canonical signed digit (CSD), Double precision floating point number, Modified Booths Algorithm (MBA), Recoding.

I .INTRODUCTION

Operations involving floating point have taken on greater significance in recent years as 3D games, sound, image and video editing have become more and more a part of everyday computing. The most fundamental operation for such applications is multiplication. The multiplication operation involves two major steps, Partial product generation and Accumulation. The speed of multiplication can be improved by reducing number of partial products or accelerating the accumulation [4]. Number of partial products depends on the number of non-zero digits. More number of non-zero digits gives more number of partial products [1]. Many researchers have proposed several techniques to reduce the number of partial products by coding the digits in such a way that it has minimum number of non-zero digits [1]-[4].M-ary segmentation technique reduces number of partial products by decomposing the bits of multiplier into words of length d=$\log_2 m$, but as m grows larger, the probability to perform additions will increase [6]. Modified Booths algorithm generates partial products by scanning two or more bits at a time based on the radix; the partial products will be reduced to Half [3].A BCSD multiplier uses a binary number system which allows representing any CSD number using the same Word length used in twos complement representation [5]. CSD recoding reduces number of non-zero digits to n/3[1].

Manuscript received Oct, 2013.

D. Harini Sharma*, PG student, ECE, SVEC, Tadepalligudem. Andhra Pradesh, INDIA.*
Addanki Purna Ramesh*, Associate Professor, ECE, SVEC, Tadepalligudem, Andhra Pradesh, India*

II. FLOATING POINT MULTIPLICATION

A scientific notation with a sign, exponent, mantissa called floating point representation is used for the real numbers. Its format is

$$significant \times base^{exponent} \tag{1}$$

To improve the portability of floating point numbers, IEEE created a standard called IEEE-754 standard for binary floating point numbers. The two basic formats are single and double precision. The IEEE-754 floating point standard Representation of floating point number [1] is

$$F = (-1)^{sign} * (2)^{exponent\ -bias} * 1.fraction \tag{2}$$

Floating point multiplication involves three basic steps [1].

1) Multiplication of mantissa of two numbers.

Because the floating point numbers are stored in signed magnitude form, the multiplier need only deal with unsigned numbers. After multiplication rounding of result is done. If the mantissa of the two numbers contains *p* bits then the product can have as many as 2*p* bits and must be rounded to *p* bits. Finally normalization is done.

2) Computing the new exponent.

Because the exponents are represented with a bias, this involves subtracting the bias from the sum of the biased exponents. The bias is 127 for single and 1023 for double precision.

3) Computing the sign bit.

The new sign bit is calculated by performing Ex-or operation on two sign bits.

The mantissa multiplication plays an important role in improving the overall speed [1].

A) CONVENTIONAL MULTIPLIER

In this method, starting from LSB, based on the bits of the multiplier either 1 or 0, the corresponding partial products are generated and necessary shifts are performed based on the bit positions. When the multiplier bit is ‘1’ then the partial product is the multiplicand, if the bit is ‘0’ then the partial product is 0. The final result is obtained by adding the partial products.
In this approach each multiplier bit generates a partial product. When the multiplier is large, then a large number of partial products have to be added.

B) MODIFIED BOOTHS ALGORITHM

Booth multiplication is a technique that recodes the multiplicand in order to reduce the number of partial products. In radix-4 booth recoding, the number of partial products are reduced by half [3].
Each digit in the recoded multiplier is obtained independently by examining three bits at a time with overlapping. Initially a zero is added to the right most bit then the three bits are recoded using the table1. To ensure that the number is even, the sign bit is extended if necessary.
Suppose, the multiplier is 010110101, a zero is placed to the right most bit of multiplier which gives 0101101010. The three bits are selected at a time with overlapping the left most bit as follows [3].

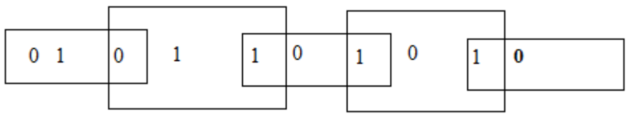

Fig.1.Grouping of bits from the multiplier

Table.1. Recoding of bits

Block	Recoded value	Operation on X
000	0	0X
001	+1	+1X
010	+1	+1X
011	+2	+2X
100	-2	-2X
101	-1	-1X
110	-1	-1X
111	0	0X

After recoding the multiplier, the partial products are obtained by using table1.
In table 1, -2 * X is actually the 2s complement of the multiplicand with an equivalent left shift by one bit position. Also, +2* X is the multiplicand shifted left by one bit position. 1* X is the multiplicand itself. -1* X is twos complement of multiplicand. After generating all the partial products, final accumulation is done.
MBA reduces the number of partial products to half.

III. CANONICAL SIGNED DIGIT

The CSD format aims to reduce the number of additions during multiplication. The CSD format has a ternary set as opposed to a binary set in number representation. The symbols used in this format are {-1, 0, 1}. The goal is to group consecutive 1s and change them to a ternary representation from binary representation. This is done starting from the rightmost 1 and proceeding left until the last 1. By, doing so CSD representation never has adjacent non-zero digits. It is proved that the number of operations never exceeds n/2 and on an average it can be reduced to n/3 [2].

A) CSD CONVERSION ALGORITHM

Let S is the mantissa of multiplier, C is the CSD code and B is the auxiliary carry. The first auxiliary carry is set to 0.Starting from the LSB, scan two bits at a time and CSD code is obtained using the following equations

$$B_{i+1} = B_i + S_{i+1} + S_i \qquad (3)$$
$$C_i = B_i + S_i - 2(S_{i+1}) \qquad (4)$$

B_i is obtained according to the conventional arithmetic rule. It is equal to 1, when there are 2 or more ones among B_i, S_{i+1}, S_i [1].
The CSD code for 7 (0111) is 100-1.

B) MULTIPLICATION USING CSD

The CSD multiplication unit contains CSD recoding block, shift control unit, adder unit and twos complement unit. CSD recoding block, converts the given number to CSD representation.

Twos complement unit is used to obtain twos complement. Initially, the two operands i.e. multiplier and multiplicand are stored in registers. The mantissa of multiplier is given to CSD recoding unit to obtain CSD code of multiplier. The multiplicand is given to 2s complement unit to get its 2s complement.

The multiplicand and its twos complement are given to shift control unit and the partial products are generated by performing required number of shifts based on the sign and position of non-zero digits. The obtained partial products are added in the adder unit.

The result is normalized and rounded to get the final product.

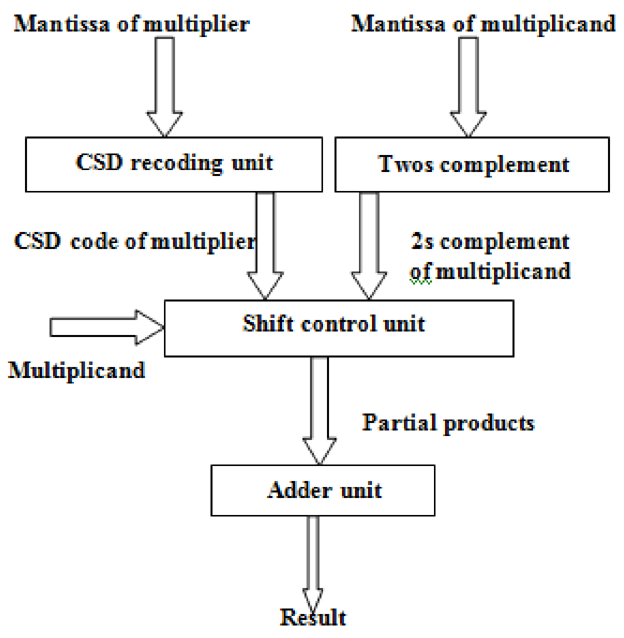

Fig.2. CSD multiplication unit

IV.RESULTS

The algorithms are implemented in verilog and are simulated, synthesized using Xilinx 9.2 ISE. The device used is Spartan3E. Table 2 gives comparison of results of CSD with conventional method for double precision floating point multiplication. It shows that CSD has less delay.

Table.2. Comparison for Double precision floating point multiplication

Parameter	conventional	CSD
Delay	100.206ns	93.193ns

V.CONCLUSION

In this paper, we have discussed algorithms for floating point multiplication. The results showed that CSD has less delay compared to conventional method for Double precision floating point multiplication.

REFERENCES

[1] Geetanjali Wasson, "IEEE-754 compliant algorithms for fast multiplication of double precision floating point numbers"
International Journal of Research in Computer Science, eISSN 2249-8265 Volume1 Issue 1(2011) pp.1-7

[2] Yunhua Wang, Linda S. DeBrunner, Dayong Zhou, Victor E.DeBrunner, "A novel multiplier less hardware implementation method for adaptive filter coefficients" *ICASSP 2007*

[3] S. Jagadeesh, S. Venkata chary "Design of parallel multiplier- accumulator based on Radix-4 modified Booth algorithm with SPST",
International Journal of Engineering research and applications, volume 2, issue 5, September- October 2012, pp.425-431.

[4] Yunhua Wang, "Iterative radix-8 multiplier structure based on a novel real-time CSD recoding" *ACSSC 2007, conference record of 41st Asilomar conference on signals, systems& computers*, 2007, pp. 977- 981.

[5] Mrs. Pushpawathi changlekar, Mrs. Sujatha, "Implementation of Binary Canonic Signed Digit multiplier using Application specific IC",
International journal of engineering research and applications, volume 3, Issue 1, Jan- Feb 2013, pp.1912-1915.

[6] Cetin K. KOC, Chin -Yu Hung, "Adaptive m-ary segmentation and Canonical recoding algorithms for multiplication of large binary numbers",
Computers & mathematics with applications, Vol 24, No.3, pp. 3-12, 1992.

D. Harini Sharma, M.Tech student, ECE department, Sri Vasavi engineering college, Tadepalligudem, Andhra Pradesh. India.

A. Purna Ramesh, Associate Professor, ECE department, Sri Vasavi engineering college, Tadepalligudem, Andhra Pradesh, India.