ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2009 (827-831)
VOLUME : 9
NUMBER : 1

A SINGLE/DOUBLE PRECISION FLOATING-POINT MULTIPLIER DESIGN FOR MULTIMEDIA APPLICATIONS

Metin Mete ÖZBİLEN[1] **Mustafa GÖK**[2]

[1]Mersin University, Engineering Faculty, Department of Computer Science, 33342, Mersin, Turkey.
[2]Çukurova University, Engineering Faculty, Department of Electric and Electronic, 01330, Adana, Turkey.

E-mail: mmozbilen@mersin.edu.tr E-mail: musgok@cu.edu.tr

ABSTRACT

Modern graphic processors, multimedia processors, and general-purpose processors with multimedia extensions provide SIMD floating-point instructions. SIMD floating-point multiplication is commonly used in 2D and 3D applications, which mostly use single precision floating-point operands. Consequently, efficient single precision multiplier units are crucial for high performance systems. This paper introduces a novel floating point multiplier that can perform either a double precision or two parallel single-precision floating-point multiplications. The proposed design uses approximately 10% more hardware and has 33% more delay compared to a conventional double precision floating-point multiplier.

Keywords: *Multimedia, floating-point numbers, multiply*

1. INTRODUCTION

Modern general purpose processors provide special instructions for multimedia applications [1]. New generations of the general purpose processors use larger sets of multimedia instructions than the ones offered by the previous generations, since the variety of the operations and the performance requirement for multimedia applications increase [1-6]. Consequently, providing efficient multimedia hardware has become an important design task. In the past, modifying only integer data path were enough to implement most multimedia instructions; however, today, the floating-point data path is modified as well, since many applications use floating-point operands [2-3].

Floating-point multiplication is one of the most supported operation in multimedia extensions offered by popular chip producers. For example, Intel's Pentium 3 has multimedia extensions SSE and SSE2 that support the multiplication of two double precision or four single precision floating-point numbers [5], similarly AMD's 3DNow multimedia extension contains an

Received Date: 12.08.2008
Accepted Date: 05.01.2009

instruction that can perform two simultaneous single precision floating-point multiplications [6]. The AltiVec extension of PowerPC also contains an instruction that executes four single-precision floating-point multiplications [7].

IEEE 754 describes single and double-precision floating-point number formats. A double-precision floating-point multiplier can be used to perform a single precision multiplication, however, this requires additional processing, since the operands should be translated to double precision, and after the multiplication, the result should be translated back to single precision format [2]. Furthermore, a double precision floating-point multiplier can perform a single multiplication per cycle. The Dual Mode IEEE multiplier design presented in [2] performs two single precision multiplications using a double precision multiplier. However, the proposed multiplier in [2] needs a delay cycle before starting a new multiplication.

This paper presents a new floating-point multiplier which can perform a double-precision floating-point multiplication or two simultaneous single precision floating-point multiplications. Since in single precision floating-point multiplication two results are generated in parallel, the multiplier's performance is almost doubled compared to a conventional floating-point multiplier.

2. PROPOSED FLOATING-POINT MULTIPLIER DESIGN

This section presents proposed design, which performs the following floating-point multiplications described in Figure 1.

Figure 1a shows the alignments of two double precision floating point numbers X, Y and their product Z that are placed in three 64-bit registers. Figure 2b shows the alignments of four single precision floating-point numbers A,B,C and D and the product of A and B, H, and the product of C and D, J that are placed in three 64-bit registers.

The multiplication of X and Y is performed as

$$E_z = E_x + E_y \quad (1)$$

$$M_z = M_x \times M_y \quad (2)$$

$$S_Z = S_x \otimes S_y \quad (3)$$

The multiplication of A and B, and the multiplication of C and D are performed as

$$E_h = E_a + E_b, E_j = E_c + E_d \quad (4)$$

$$M_h = M_a \times M_b, \; M_j = M_c \times M_d \quad (5)$$

$$S_{ab} = S_a \otimes S_b, \; S_{ab} = S_a \otimes S_b \quad (6)$$

The proposed design performs these two floating-point multiplications in parallel. In [9], a design method for the multiplication of two unsigned integer operands is presented. Figure 2 presents the adaptation of the technique in [9] to implement the proposed method. In this figure, the matrices generated for the two single precision floating-point multiplications are placed in the matrix generated for a double precision floating-point multiplication. All the bits are generated in double precision mode, the shaded areas Z1, Z2, Z3 are not generated when single precision multiplication is performed, the non-shaded regions designate the generated bits.

	63	62…52	51…0
X	S_x	E_x	M_x
Y	S_y	E_y	M_y
Z	S_z	E_z	M_z

(a) The alignment of two double precision floating numbers and their product.

	63	62…55	54…32		31	30…23	22…0
A	S_a	E_a	M_a	C	S_c	E_c	M_c
B	S_b	E_b	M_b	D	S_d	E_d	M_d
H	S_h	E_h	M_h	J	S_j	E_j	M_j

(b) The alignment of four single precision floating point numbers and their products.

Figure 1. The alignments for double and single precision numbers.

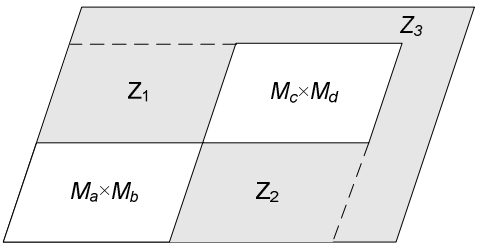

Figure 2. The multiplication matrix for single and double precision mantissas.

The partial products within the regions Z1, Z2 and Z3 are generated using following equations

$$\hat{b}_j = s \cdot b_j \text{ and } p_{ij} = a_i \cdot \hat{b}_j \quad (7)$$

,the rest of the partial products are produced with

$$p_{ij} = a_i \cdot b_j \quad (8)$$

's' is used as control signal. When s = '0', only the bits in no shaded regions are generated. When s = '1', all bits are generated.

High-speed multipliers reduce the partial product matrix to two vectors using a reduction method. Then, these two vectors are added to produce the result with carry-propagate adder. The reduction method and the type of the carry-propagate adder are not important for the proposed design, since it only modifies the generation of the partial products. also means that the reduction algorithm and the carry-propagate adder is not modified for the implementation of the proposed method.

Figure 3 presents a block diagram for a standard floating-point multiplier. This multiplier directly implements Eqns. (1)-(3). Figure 4 presents the proposed single/dual floating-point multiplier which is designed by slightly modifying the standard floating-point multiplier. The modifications can be used on every type of double precision floating-point multiplier.
The data flow and the functionality of each unit in the proposed design are explained as follows:

- The control signal determines the mode of execution; when s='0' double precision floating-point multiplication is performed, otherwise two single precision multiplications are performed.
- The 11-bit adder is used for double precision exponent addition and two 8-bit adders are used for single precision exponent additions.
- The exponent updaters remove extra bias values form the exponent additions.
- The mantissa modifier implements the logic equations given in Eqn 7.
- The mantissa multiplier generates carry-save vectors.
- The add, normalize and round unit generates normalized and rounded result.
- The signs of the products are obtain by XOR gates.

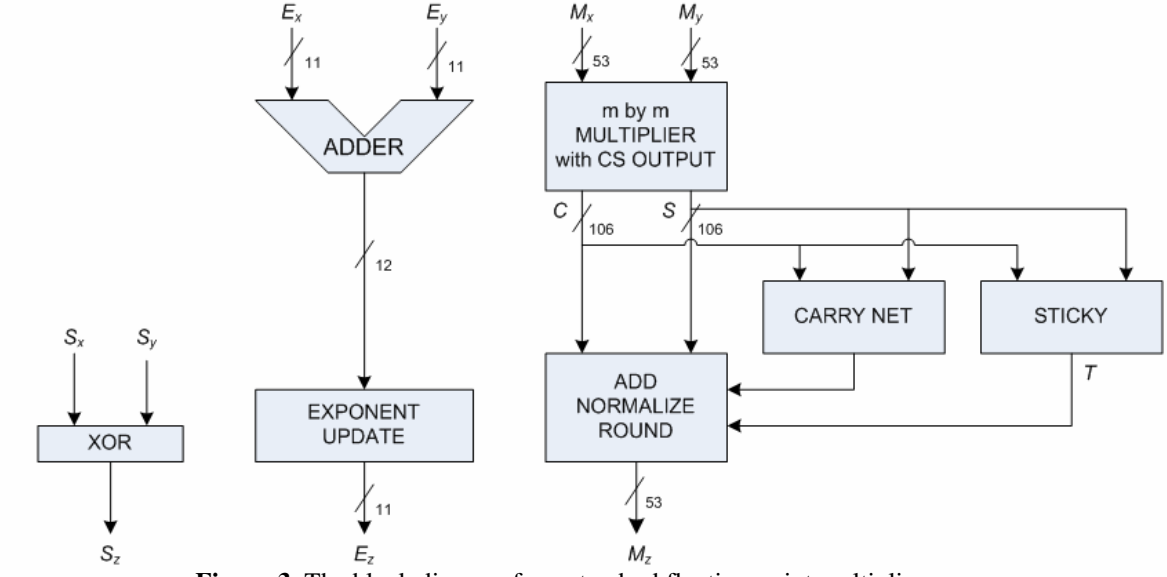

Figure 3. The block diagram for a standard floating-point multiplier.

This

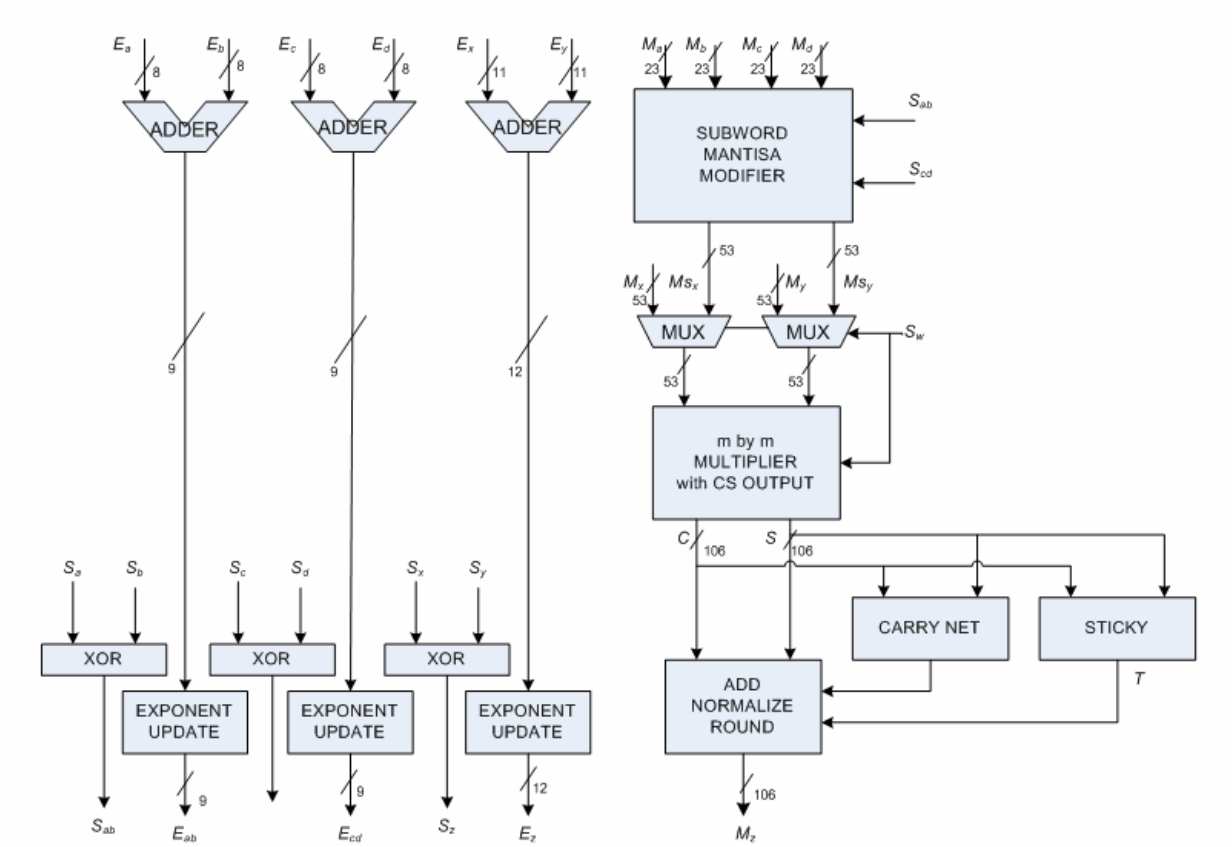

Figure 4. The block diagram for the proposed floating-point multiplier.

3. SYNTHESIS RESULTS

In this section we present the synthesis results for the proposed single/double precision floating point multiplier and the standard dual precision floating-point multiplier. Both circuits are modeled using structural VHDL code. Synthesis of the circuit is done using TSMC 0.18 micron ASIC library and Leonarda Spectrum program. Both circuits are optimized for delay. The values in Table 1 are in nanoseconds for time and in number of gate for area.

Table 1. The comparison of the standard double precision and proposed floating-point multiplier designs.

Design	# of Gates	Latency
Double Precision	25175	4.10 Ns
Single/Double Precision	27566	5.49 Ns

The single/double precision multiplier has approximately 9.49% more area and has about 34% more critical delay. The floating-point multipliers used in modern processors are usually pipelined designs. If the proposed method is applied to a pipelined multiplier the area increase is expected to fall down below 5% and also critical delay increase will be dissolved in pipeline stages.

4. CONCLUSION

In this study a floating point multiplier design which is capable of executing a double precision floating-point multiplication or two parallel single precision multiplications is presented. One of the important aspects of the presented design method is that it can be applicable to all kinds of floating-point multipliers. The presented design is compared with a standard floating point multiplier via synthesis. The synthesis results showed that proposed design is %10 larger than

conventional multiplier and critical path increment is only one or two gate delay. Since modern floating-point multiplier designs have significantly larger area than the standard floating-point multiplier, the percentage of the extra hardware will be less for those units. The methods presented in this study will be used on the design of floating-point multiplier-adder circuits. Also, the future work will enhance the proposed designs to support all IEEE754 rounding modes.

5. REFERENCES

[1] V. Lappalainen, T.D. Hämäläinen, P. Liuha, "Overview of Research Efforts on Media ISA Extensions and Their Usage in Video Coding", *IEEE Transactions on Circuits And Systems For Video Technology*, Vol: 12, No: 8, 2002.

[2] G. Even, S.M. Mueller, P.M. Seidel, "A dual mode IEEE multiplier Proceedings", *Second Annual IEEE International Conference*, pp. 282-289, 1997.

[3] R.B. Lee, "Multimedia extensions for general purpose processors, Signal Processing Systems, 1997", *SIPS 97 - Design and Implementation., IEEE Workshop*, pp. 9-23, 1997.

[4] ANSI-IEEE Standard 754-1985: "IEEE Standard for binary floating-point arithmetic", 1985.

[5] Intel, "IA-32 Intel™ Architecture Software Developer's Manual", http://download.intel.com/design/Pentium4/manuals/25366520.pdf, 2006.

[6] Advanced Micro Devices, "AMD64 Architecture Programmers Manual", White Paper, http://www.amd.com/usen/assets/contenttype/white_papers_and_tech_docs/26569.pdf, 2006.

[7] K. Diefendorff, P.K. Dubey, R. Hochsprung, H. Scale, "AltiVec extension to PowerPC accelerates media processing", *IEEE Micro*, Vol: 20 Iss: 2, pp. 85-95, 2000.

[8] M.D. Jennings, T.M. Conte, "Subword Extensions for Video Processing on Mobile Systems", *IEEE Concurrency*, Vol: 6, No: 3, pp. 13-16, 1998.

[9] M. Gök, S. Krithivasan, M.J. Schulte, "Designs for Subword-Parallel Multiplications and Dot Product Operations", *Workshop on Application Specific Processors*, pp. 27-31, 2004.

Metin Mete ÖZBİLEN received his B.Sc. degree in Electrics and Electronics from Gaziantep University in Turkey in 1996, and his M.Sc in Electrics and Electronics from Çukurova University in Adana, Turkey in 2001. He is pursuing Ph.D. degree in Electrics and Electronics at Çukurova University in Adana. He has been a research assistant in Computer Engineering Department in Mersin University since 2001. His current areas of research include computer arithmetic and the design of IEEE compliant floating-point units, multimedia acceleration algorithms.

Mustafa Gök received his B.Sc. degree in Electronics Engineering from the Istanbul University, Istanbul in 1995, the M.Sc. degree in Electrical Engineering from the Lehigh University, Bethlehem in 2000, and the Ph.D. degree in Electrical Engineering from the Lehigh University, Bethlehem in 2003. He is an Assistant Professor in Cukurova University. His research interests include computer arithmetic, computer architecture, and design of hardware accelerators.