Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(2): 55-59

Research Article ISSN: 2394 - 658X

Design of Fast Floating Point Multiply Accumulate Unit using Ancient Mathematics for DSP Applications

Meenu S Ravi1, R H Khade and Ajit Saraf

Department of Electronics Engg, MES Pillai Institute of Information Technology, Navi-Mumbai, India
meenusravi@gmail.com

ABSTRACT

Digital Signal processing became an application to create high speed data processing systems like 3D rendering, 4G mobile internet, etc., we need best processors with high performance data path units and there is a growing need for research on alternative methods for signal processing hardware implementation. In most systems using digital signal processing Multiply-Accumulate (MAC) is one of the main functions. The performance of the whole system depends on the performance of the MAC units in place. Regardless of that these days' real time signal processing systems require high throughput and high performance MAC units. A MAC unit is simply one of the main units in all Digital Signal Processors which performs the multiplication of two numbers of any radix and accumulates the by-products in order. In this paper, a floating point multiply and accumulate unit is designed using ancient mathematics that reduces the number of partial products to be added as well as increases the speed of accumulation of partial products by reducing the number of stages of partial products that needs to be added thereby making it a high performance unit. The output of this unit is simulated using simulation software ModelSim SE plus 6.2C and the language used is VHDL.

Key words: 3D rendering, 4G mobile internet, MAC and ModelSim

INTRODUCTION

Because of the high-accuracy, impressing dynamic range and use of easily operable rules, floating-point operations have found applications in many fields that require the above mentioned qualities. At present even in computers, floating- point arithmetic operations are mainly performed by the co- processors. In systems without floating-point hardware, the CPU emulates it with a series of simpler fixed-point arithmetic operations that run on the integer arithmetic and logical unit. This saves the added cost of a floating- point unit (FPU) but is too slower compared to the traditional systems. Coprocessors cannot fetch instructions from the main-memory; perform I/O, manage-memory and so on. These processors require the host main processor to fetch the coprocessor instructions from the memory and handle all operations aside from the coprocessor functions. High processor speeds demand high coprocessor operation speed [1-[2]. This saves the added cost of a floating- point unit but is significantly slower. Coprocessors cannot fetch instructions from the main-memory; perform I/O, manage-memory and so on. These processors require the host main processor to fetch the coprocessor instructions from the memory and handle all operations aside from the coprocessor functions. High processor speeds demand high coprocessor operation speed.

PROBLEM DEFINITION

A MAC unit is simply one of the main units in all Digital Signal Processors which performs the multiplication of two numbers of any radix and accumulates the by-products in order. In this paper, a floating point multiply and accumulate unit is designed using ancient mathematics that reduces the number of partial products to be added as well as increases the speed of accumulation of partial products by reducing the number of stages of partial products that needs to be added thereby making it a high performance unit.

Vedic Mathematics

It is one of the fastest growing technologies in this century. Faster additions and multiplications are having extreme importance in DSP applications such as convolution, discrete Fourier transforms digital filters, etc. The main computing process is always a multiplication routine. Vedic mathematics is the name given to the ancient system of mathematics, which was rediscovered, from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirthaji. The whole of Vedic mathematics is based on 16 sutras (word formulae) and manifests a unified structure of mathematics. As such, the methods are complementary, direct and easy. Amongst the sutras for various arithmetic operations Urdhav-Triyak sutra is the sutra for multiplication operations [10].

URDHVA – TIRYAGBHYAM SUTRA

Urdhva – tiryagbhyam is the general formula applicable to all cases of multiplication and also in the division of a large number by another large number [4].

Algebraic Proof

a) Let the two 2 digit numbers be (ax+b) and (cx+d). Note that x = 10. Now consider the product (ax + b) (cx + d) $= ac.x^2 + adx + bcx + b.d = ac.x^2 + (ad + bc)x + b.d$

- The first term i.e., the coefficient of x^2 (i.e., 100, hence the digit in the100th place) is obtained by vertical multiplication of a and c i.e., the digits in10th place (coefficient of x) of both the numbers;
- The middle term, i.e., the coefficient of x (i.e., digit in the 10th place) is obtained by cross wise multiplication of a and d; and of b and c; and the addition of the two products;
- The last (independent of x) term is obtained by vertical multiplication of the independent terms b and d.

b) Consider the multiplication of two 3 digit numbers. Let the two numbers be $(ax^2 + bx + c)$ and $(dx^2 + ex + f)$. Note that x=10 Now the product is

$$\begin{array}{c} ax^2 + bx + c \\ * \\ dx^2 + ex + f \\ \hline ad.x^4+bd.x^3+cd.x^2+ae.x^3+be.x^2+ce.x+af.x^2+bf.x+cf \\ = ad.x^4 + (bd + ae).\,x^3 + (cd + be + af).x^2 + (ce + bf)x + cf \end{array}$$

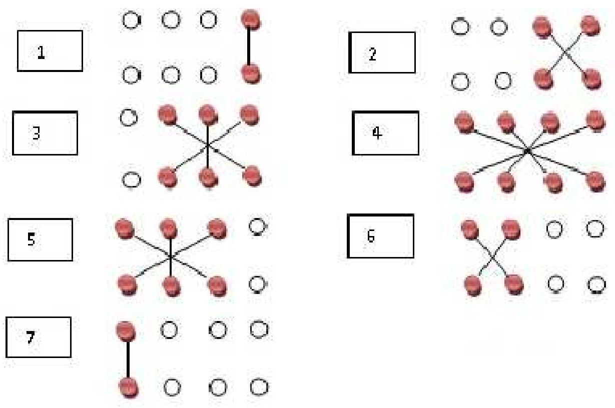


Fig. 1 4 bit Vedic multiplication

The same operation will be carried out even when the numbers are binary: Now let's assume the number consists of four bits then there will be seven iterations and the iterations are mentioned simply by using the diagram represented below. This is the basic which I have used to implement the Vedic multiplier.

IMPLEMENTATION OF FLOATING POINT MAC

The floating point MAC generally comprised of 2 two main sections a multiplier and an accumulator. The multiplier used is a Vedic multiplier which is based on ancient mathematics. The operands are binary floating point numbers having 40 bits [5]. The 40 bits are divided into three sections sign, exponent and mantissa [Fig. 3].each will be processed separately. The one bit sign (1 bit) will be obtained by performing XOR operation to the sign of multiplier and multiplicand. The exponent (7 bits) is obtained as the sum of exponents of the inputs. Finally the mantissa of the result is obtained by the 32 bit Vedic multiplication [6] [7]. The flow diagram of the main module is as follows [Fig. 2].

KOGGE-STONE ADDER

KSA is a parallel prefix-carry look ahead adder. In this work it has been useful to add the by-products of multiplication through carries. It generates carry in zero processing time and is widely considered as the fastest adder [9], Due to these characteristics it is widely used in the industry for high performance arithmetic circuits. In KSA, carries are computed fast by computing them in parallel at the cost of increased area. The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts [1]:

Part 1: Pre processing
This step involves computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

$$p_i = A_i \textit{ xor } B_i$$

$$g_i = A_i \textit{ and } B_i \quad (1)$$

Part 2: Carry Looks Ahead Network
This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

$$P_{i:j} = P_{i:k+1} \text{ and } P_{k:j}$$

$$G_{i:j} = G_{i:k+1} \text{ or } (P_{i:k+1} \text{ and } G_{k:j}) \quad (2)$$

Part 3: Post processing
This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic given below:

$$S_i = p_i \text{ xor } C_{i-1} \quad (3)$$

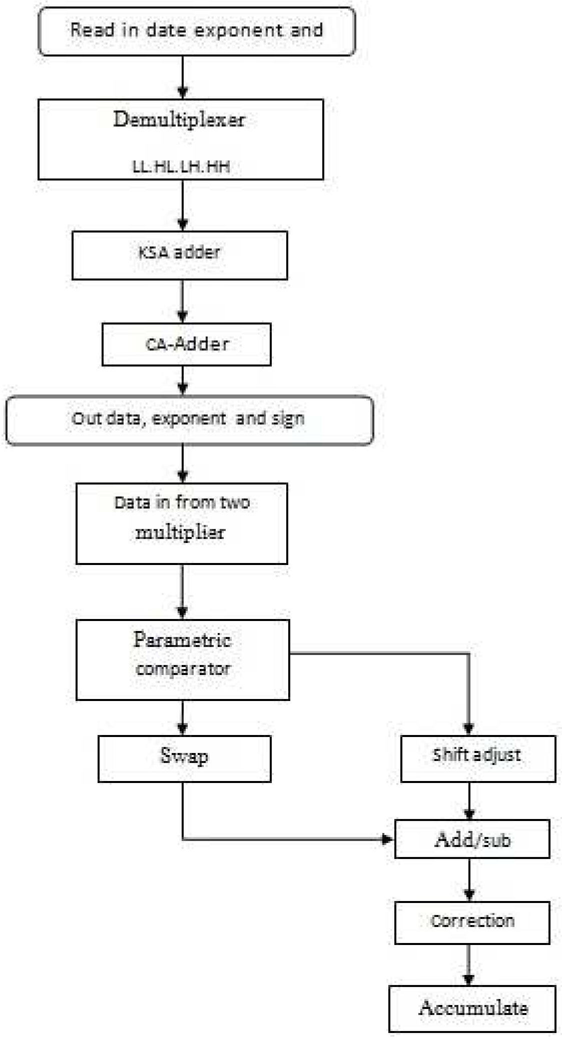


Fig. 2 Flow chart of implementation

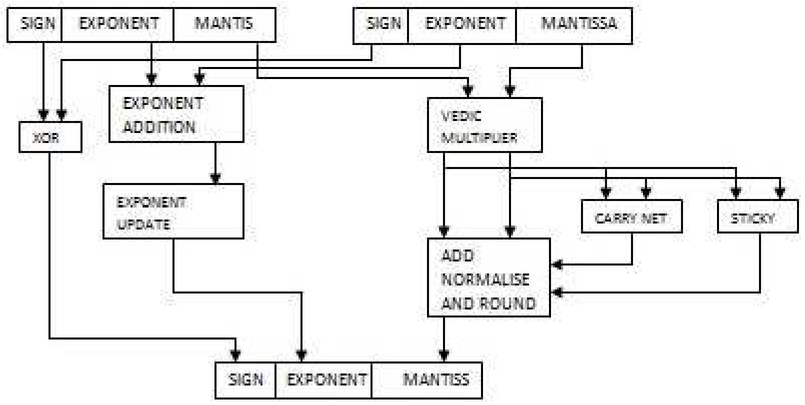


Fig. 3 Block diagram representation of floating point multiplier

RESULTS AND DISCUSSION

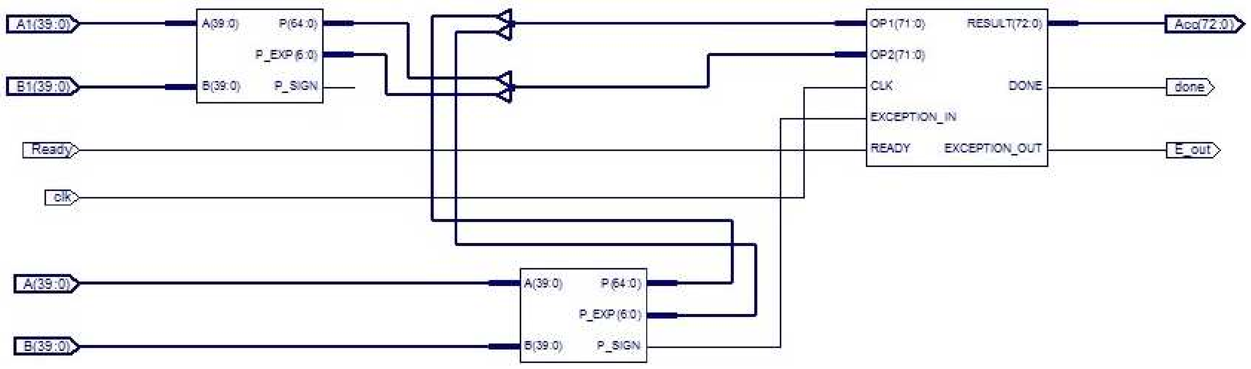


Fig. 4 RTL schematic of floating point MAC

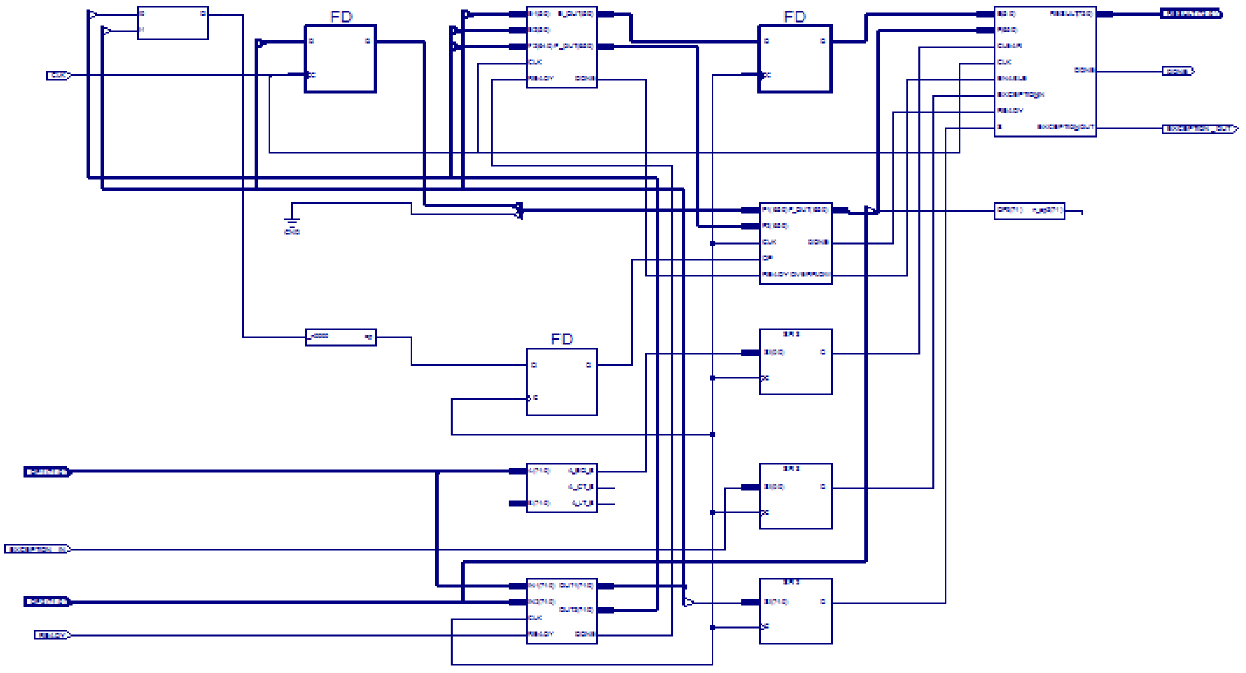
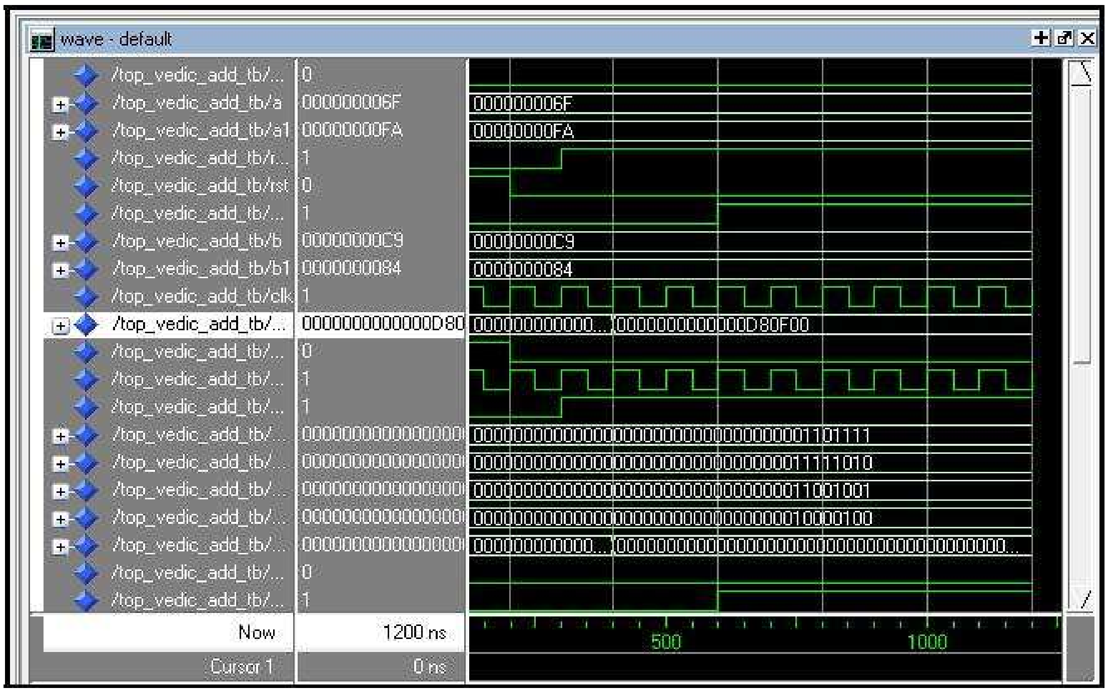


Fig. 5 Detailed RTL Schematics of floating point MAC



```
wave - default
/top_vedic_add_tb/...   0
/top_vedic_add_tb/a     000000006F          000000006F
/top_vedic_add_tb/a1    00000000FA          00000000FA
/top_vedic_add_tb/r...  1
/top_vedic_add_tb/rst   0
/top_vedic_add_tb/...   1
/top_vedic_add_tb/b     00000000C9          00000000C9
/top_vedic_add_tb/b1    0000000084          0000000084
/top_vedic_add_tb/clk   1
/top_vedic_add_tb/...   000000000000D80     000000000000... 000000000000D80F00
/top_vedic_add_tb/...   0
/top_vedic_add_tb/...   1
/top_vedic_add_tb/...   1
/top_vedic_add_tb/...   0000000000000000    0000000000000000000000000000000001101111
/top_vedic_add_tb/...   0000000000000000    0000000000000000000000000000000011111010
/top_vedic_add_tb/...   0000000000000000    0000000000000000000000000000000011001001
/top_vedic_add_tb/...   0000000000000000    0000000000000000000000000000000010000100
/top_vedic_add_tb/...   0000000000000000    000000000000... 00000000000000000000000000000000000000000...
/top_vedic_add_tb/...   0
/top_vedic_add_tb/...   1
Now                     1200 ns             500   1000
Cursor 1                0 ns
```

Fig. 6 Simulation Results radix-hexadecimal

CONCLUSION

Vedic mathematical methods are extracted from ancient systems of computations, now made globally available to everyone through the great work done by Swami Sri Bharati Krishna Tirthaji Maharaja, who has published a book on Vedic mathematics in 1965. Compared to other conventional mathematical methods, these are faster and easy to compute. It has been one of the fastest algorithms in computations due to the reduction in number of intermediate products (cross products). Therefore, such approaches have been extremely beneficial in digital signal processing applications. The engineers can develop such fast and low cost devices if it is included in engineering education.

REFERENCES

[1] Dhananjaya A and Deepali Koppad, Design of High Speed Floating Point MAC using Vedic Multiplier and Parallel Prefix Adder , *International Journal of Engineering Research & Technology*, **2012**, Vol. 2 Issue 6, p. 3359-3363.
[2] Dinesh Kumar and Girish Chander Lal, Simulation And Synthesis Of 32-Bit Multiplier Using Configurable Devices, *International Journal of Engineering Research & Technology*, Vol. 2 Issue 6, **2013**, p. 216-223.
[3] Anitha R, Alekhya Nelapati, Lincy Jesima W and V Bagyaveereswaran, Comparative Study of High Performance Braun’s Multiplier using FPGAs, *IOSR Journal of Electronics and Communication Engineering* ISSN, **2012,** Vol 1, Issue 4, p. 33-37.
[4] Purushottam, D Chidgupkar and Mangesh T Karad, The Implementation of Vedic Algorithms in Digital Signal Processing, Global Journal of Engineering. Education, **2004**, Vol 8, No 2, p. 153-158.
[5] Gong Renxi, Zhang Shangjun, Zhang Hainan, Meng Xiaobi, Gong Wenying, Xie Lingling and Huang Yang, Hardware Implementation of a High Speed Floating Point Multiplier Based on FPGA, *Proceedings of 4th International Conference on Computer Science & Education*, **2009**, p. 1902-1906.
[6] IEEE 754-2008, *IEEE Standard for Floating-Point Arithmetic*, **2008**.
[7] Aniruddha Kanhe, Shishir Kumar Das and Ankit Kumar Singh, Design and Implementation of Floating Point Multiplier based on Vedic Multiplication Technique, *International Conference on Communication, Information & Computing Technology (ICCICT)*, **2012**.
[8] K Ramakrishnan, T Ravi and V Kannan, Transistor Level Design and Analysis of Vedic Algorithm Based Low Power MAC, *International Journal of Engineering Research & Technology*, **2013,** Vol 2, Issue 2.
[9] Mohammed Hasmat Ali and Anil Kumar Sahani, Study, Implementation and Comparison of Different Multipliers based on Array: KCM and Vedic Mathematics Using EDA Tools, *International Journal of Scientific and Research Publications*, **2013**, Vol 3, Issue 6.
[10] Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaja, *Vedic Mathematics, Sixteen simple Mathematical Formulae from the Veda*, Delhi, **1965.**