INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume II, Issue 1 SEP 2014

IMPLEMENTATION OF HIGH PERFORMANCE 64-BIT MAC

NALLAGONDA RAMESH(1),RAJASEKHAR TURAKA(2)
M.Tech, Vlsi And Embedded System Design, Chalapathi Institute Of Engineering And Technology(1)
M.Tech, Asst.Professor, , Chalapathi Institute Of Engineering And Technology(2)

ABSTRACT

This paper present Multiply and Accumulate (MAC) unit design using Vedic Multiplier, which is based on Urdhva Tiryagbhyam Sutra. The paper emphasizes an efficient 32 -bit MAC architecture along with 8 - bit and 16 -bit versions and results are presented in comparison with conventional architectures. A design of high performance 64 bit Multiplier and-Accumulator (MAC) is implemented in this paper. MAC unit performs important operation in many of the digital signal processing (DSP) applications. The multiplier is designed using modified Wallace multiplier and the adder is done with carry save adder. The total design is coded with verilog-HDL and the synthesis is done using Cadence RTL complier The efficiency in terms of area and speed of proposed MAC unit architecture is observed through reduced area, low critical delay and low hardware complexity. The proposed MAC unit reduces the area by reducing the number of multiplication and addition in the multiplier unit. Increase in the speed of operation is achieved by the hierarchical nature of the Vedic multiplier unit. The proposed MAC unit is implemented on a field programmable gate array (FPGA) device. The performance evolution results in terms of speed and device utilization are compared to earlier MAC architecture. Though the use of Vedic mathematics methods for multiplication is reported in literature, it has been observed that our proposed method of 32-bit MAC unit implementation is using multiplication unit and shows improvements in the delay and area.

Index Terms-MAC
Multiplier,Nikhilam Sutra, Urdhva Tiryagbhyam Sutra, Vedic Mathematics.

I.INTRODUCTION MAC

unit is an inevitable component in many digital signal processing (DSP) applications involving multiplications and/or accumulations. MAC unit is used for high performance digital signal processing systems. The DSP applications include filtering, convolution, and inner products. Most of digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transforms (DWT). Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition arithmetic determines the execution speed and performance of the entire calculation [1]. Multiplication-and accumulate operations are typical for digital filters. Therefore, the functionality of the MAC unit enables high-speed filtering and other processing typical for DSP applications. Since the MAC unit operates completely independent of the CPU, it can process data separately and thereby reduce CPU load. The application like optical communication systems which is based on DSP , require extremely fast processing of huge amount of digital data. The Fast Fourier Transform (FFT) also requires addition and multiplication. 64 bit can handle larger bits and have more memory. A MAC unit consists of a multiplier and an accumulator containing the sum of the previous successive products. The MAC inputs are obtained from the memory location and given to the multiplier. block. The design consists of 64 bit modified Wallace multiplier, 128 bit carry save adder and a register.

1.1. MAC OPERATION

The Multiplier-Accumulator (MAC) operation is the key operation not only in DSP applications but also in multimedia information processing and various other applications. As mentioned above, MAC unit consist of multiplier, adder and register/accumulator. In this paper, we used 64 bit modified Wallace multiplier. The MAC inputs are obtained from the memory location and given to the multiplier block. This will be useful in 64 bit digital signal processor. The input which is being fed from the memory location is 64 bit. When the input is given to the multiplier it starts computing value for the given 64 bit input and hence the output will be 128 bits. The multiplier output is given as the input to carry save adder which performs addition.

The function of the MAC unit is given by the following equation [4]:

$F= \sum P_j Q_j$ (1)

The output of carry save adder is 129 bit i.e. one bit is for the carry (128bits+ 1 bit). Then, the output is given to the accumulator register. The accumulator register used in this design is Parallel In Parallel Out (PIPO). Since the bits are huge and also carry save adder produces all the output values in parallel, PIPO register is used where the input bits are taken in parallel and output is taken in parallel. The output of the accumulator register is taken out or fed back as one of the input to the carry save adder. The figure 1 shows the basic architecture of MAC unit.

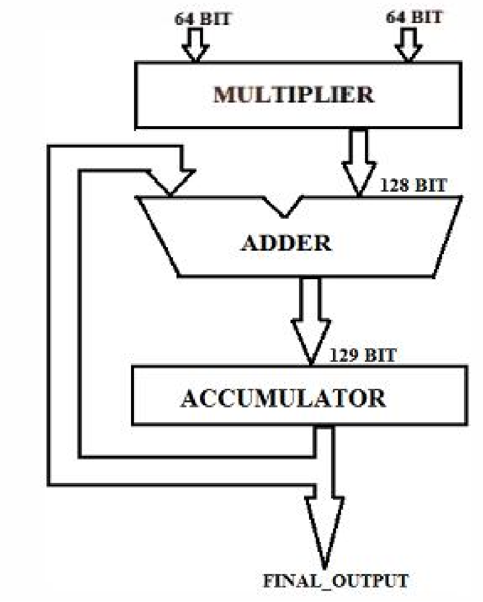


FIG1. Basic architecture of MAC unit

II.MODIFIED WALLANCE MULTIPLIER

A modified Wall ace multiplier is an efficient hardware implementation of digital circuit multiplying two integers. Generally in conventional Wallace multipliers many full adders and half adders are used in their reduction phase. Half adders do not reduce the number of partial product bits. Therefore, minirnizing the number of half adders used in a multiplier reduction will reduce the complexity [2]. Hence, a modification to the Wallace reduction is done in which the delay is the same as for the conventional Wallace reduction. The modified reduction method greatly reduces the number of half adders with a very slight increase in the number of full adders [2].

Reduced complexity Wallace multiplier reduction consists of three stages [2]. First stage the N x N product matrix is formed and before the passing on to the second phase the product matrix is rearranged to take the shape of inverted pyramid. During the second phase the rearranged product matrix is grouped into non-overlapping group of three as shown in the figure 2, single bit and two bits in the group will be passed on to the next stage and three bits are given to a full adder. The number of rows in the in each stage of the reduction phase is calculated by the formula

$r_{j+1}= 2[ri/3]+r_j mod3$ (2)

If r_j mod3 = 0, then $r_j+ 1 = 2r/3$ (3)

If the value calculated from the above equation for number of rows in each stage in the second phase and the number of row that are formed in each stage of the second phase does not match, only then the half adder will be used. The final product of the second stage will be in the height of two bits and passed on to the third stage. During the third stage the output of the second stage is given to the carry propagation adder to generate the final output. Thus 64 bit modified Wallace multiplier is constructed and the total number of stages in the second phase is 10. As per the equation the number of row in each of the 10 stages was calculated and the use of half adders was restricted only to the 10th stage. The total number of half adders used in the second phase is 8 and the total number of full adders that was used during the second phase is slightly increased that in the conventional Wallace multiplier.

Since the 64 bit modified Wallace multiplier is difficult to represent, a typical lO-bit by 10-bit reduction shown in figure 2 for understanding. The modified Wallace tree shows better performance when carry save adder is used in final stage instead of ripple carry adder. The carry save adder which is used is considered to be the critical part in the multiplier because it is responsible for the largest amount of computation.

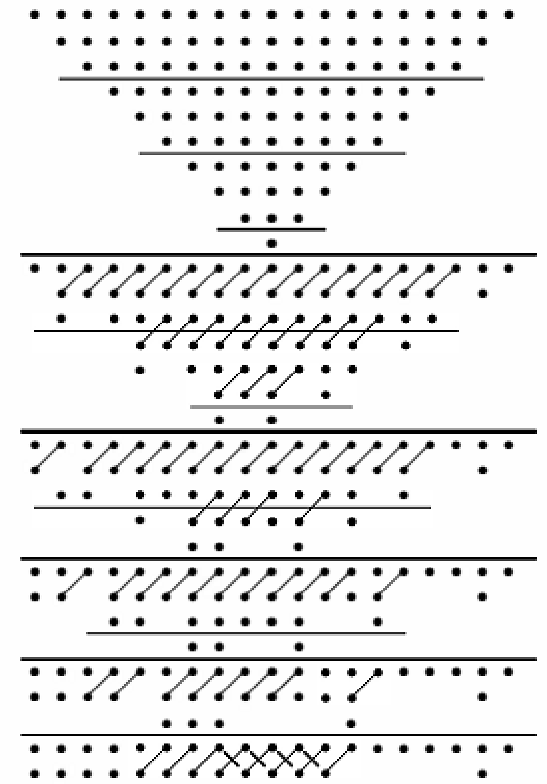


FIG 2.Modified Wallace IO-bit by IO-bit reduction

III. CARRY SAVE ADDER

In this design 128 bit carry save adder [6] is used since the output of the multiplier is 128 bits (2N). The carry save adder minirnize the addition from 3 numbers to 2 numbers. The propagation delay is 3 gates despite of the number of bits. The carry save adder contains n full adders, computing a single sum and carries bit based mainly on the respective bits of the three input numbers. The entire sum can be calculated by shifting the carry sequence left by one place and then appending a 0 to most significant bit of the partial sum sequence. Now the partial sum sequence is added with ripple carry unit resulting in n + 1 bit value. The ripple carry unit refers to the process where the carryout of one stage is fed directly to the carry in of the next stage. This process is continued without adding any intermediate carry propagation.

If x and y are supposed to be two 128 bit numbers then it produces the partial products and carry as S and C respectively.

Si = xi ^ yi (4)

Ci = xi & yi (5)

During the addition of two numbers using a half adder, two ripple carry adder is used. This is due the fact that ripple carry adder cannot compute a sum bit without waiting for the previous carry bit to be produced, and hence the delay will be equal to that of n full adders. However a carry-save adder produces all the output values in parallel, resulting in the total computation time less than ripple carry adders. So, Parallel In Parallel Out (PIPO) is used as an accumulator in the final stage.

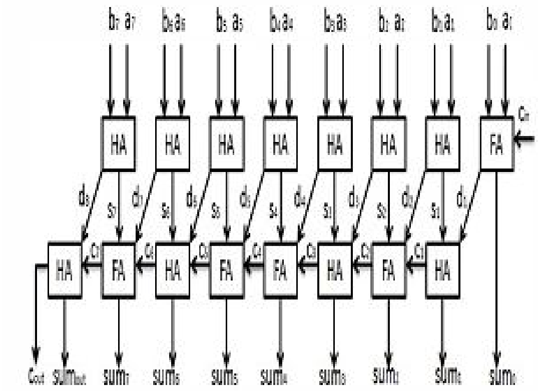
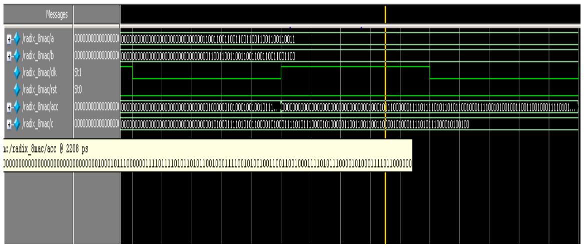


FIG 3. 8 bit carry save adder

Results:

As a previous work, 8 bit MAC unit is designed using different multipliers and adders. The multipliers used for comparative study are: (i) Modified Booth Aigorithm (ii) wallance Multiplier (iii) vedic multiplier. The different adders used in the study are: (i) Carry Look Ahead (ii) Carry Select Adder (iii) Carry Save adder.



```
/radix_8mac/a    0000000000000000  00000000000000000000000000000011001100110011001100110011001100110011
/radix_8mac/b    0000000000000000  0000000000000000000000000000000110011001100110011001100110011001100
/radix_8mac/clk  St1
/radix_8mac/rst  St0
/radix_8mac/acc  0000000000000000  000000000000000000000000000000010000001010001001001011...  00000000000000000000000000000000100010111000000111101111010110101100100011110010100100110011001000111101011...
/radix_8mac/c    0000000000000000  000000000000000000000000000000001010001111010101100001010001111010111000010100001100110011001100001010001111010111000010100100
m:/radix_8mac/acc @ 2208 ps
0000000000000000000000000000001000101110000001111011110101101011001000111100101001001100110010001111010111000010100011110110000000
```

FIG 4.Modified Booth Algorithm mac

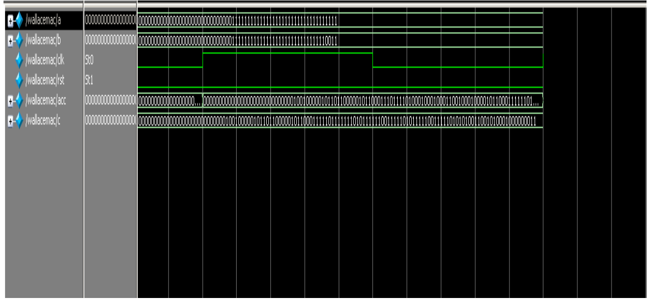


FIG 5.Wallance Mac waveforms

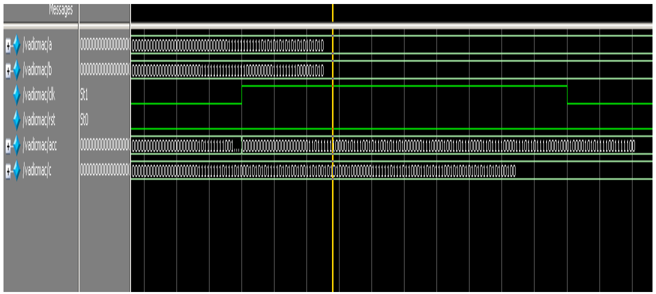


FIG 6.Vedic Mac wave forms.

IV. CONCLUSION

Hence a design of high performance 64 bit Multiplier-and-Accumulation (MAC) is implemented in this paper. The total MAC unit operates at a frequency of 217 MHz. The total power dissipated by 64 bit MAC unit is 177.732 mW. The total area occupied by it is 542177 11m2. Since the delay of 64 bit is less, this design can be used in the system which requires high performance in processors involving large number of bits of the operation. The MAC unit is designed using Verilog-HDL and synthesized in Cadence 180nm RTL Complier.

REFERENCES

[1].Young-Ho Seo and Dong-Wook Kim, "New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm," IEEE Transactions on very large scale integration (vlsi) systems, vol. 18, no. 2,february 20 10

(2). Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wall ace Multiplier Reduction, " IEEE Transactions On Computers, vol. 59, no. 8, Aug 20 10

[3]. C. S. Wallace, "A suggestion for a fast multiplier," iEEE Trans. ElectronComput., vol. EC-13, no. I, pp. 14-17, Feb. 1964

[4].Shanthala S, Cyril Prasanna Raj, Dr.S.Y.Kulkarni, "Design and VLST Implementation of Pipelined Multiply Accumulate Unit," IEEE International Conference on Emerging Trends in Engineering and Technology, ICETET-09

[5].B.Ramkumar, Harish M Kittur and P.Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder ", European Journal of Scientific Research, Vol. 42, Issue 1, 2010.

[6]. R.UMA, Vidya Vijayan, M. Mohanapriya and Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICSj Vo1.3, No.1, February 2012

[7]. V. G. Oklobdzija, "High-Speed VLSI Arithmetic Units: Adders and Multipliers", in "Design of High-Performance Microprocessor Circuits", Book edited by A.Chandrakasan,IEEE Press,2000

[8]. C.S. Wall ace "A Suggestion for a fast multipliers," IEEE Trans. Electronic Computers, vol. 13, no.l,pp 14-17, Feb. 1967

[9]. WJ. Townsend, E.E. Swartzlander Jr., and J.A. Abraham, "A Comparison of Dadda and Wall ace Multiplier Delays," Proc. SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations XIII, pp. 552-560, 2003

[10]. Fabrizio Lamberti and Nikos Andrikos, " Reducing the Computation Time in (Short Bit-Width) Two's Complement Multipliers", IEEE transactions on computers, Vol. 60, NO. 2, FEBRUARY 20 11