MAC Implementation using Vedic Multiplication Algorithm

Manoranjan Pradhan
VSS University of Technology
Burla, Sambalpur
Orissa, India-768018

Rutuparna Panda
VSS University of Technology
Burla, Sambalpur
Orissa, India-768018

Sushanta Kumar Sahu
VSS University of Technology
Burla, Sambalpur
Orissa, India-768018

ABSTRACT
The paper presents the implementation of MAC (multiplier-accumulator) unit using Vedic multiplier. The speed of MAC depends on the speed of the multiplier. The Vedic multiplier uses "Urdhva Tiryagbhyam" algorithm. The proposed MAC unit is coded in VHDL, synthesized and simulated using Xilinx ISE 10.1 software. The MAC is implemented on a FPGA device XC2S200-6PQ208 using Xilinx ISE10.1 tool. The proposed design shows improvement of speed over the design presented in [1].

General Terms
Algorithms.

Keywords
MAC, Vedic multiplier, VHDL, Carry Save Adder.

1. INTRODUCTION
A conventional MAC unit consists of multiplier and an accumulator that contains the sum of the previous consecutive products. The main goal of a DSP processor design is to enhance the speed of the MAC unit.

A high speed energy efficient ALU design using Vedic mathematics is discussed in [1]. They have implemented ALU using adder, subtractor, Vedic multiplier, and MAC unit. They have implemented MAC using Vedic multiplier. Their Vedic multiplier architecture shows speed improvements over conventional shift and add algorithm.

In [2], authors have compared implementation of normal multiplication and Vedic multiplication. They claim that same number of multiplication and addition operations is required in both normal multiplier and Vedic multiplier. They have tested and compared various multiplier implementations such as Array multiplier, Multiplier macro, Vedic multiplier with full partitioning, Vedic multiplier using 4 bit macro, fully Recursive Vedic multiplier, Vedic multiplier using 8 bit macro for optimum speed.

Dhillon and Mitra [3] Proposed a multiplier using 'Urdhva Tiryagbhyam' algorithm, which is optimized by 'Nikhilam' algorithm. They have suggested a reduced bit multiplication algorithm using 'Urdhva Tiryagbhyam' and 'Nikhilam' Sutra. Their multiplier architecture is very similar to the array multiplier.

2. THE PROPOSED MAC UNIT
The multiply-accumulate unit computes the product of two numbers and adds that product to an accumulator. The MAC unit, consisting of a multiplier followed by an adder and an accumulator register which stores the result when clocked [4-5]. The output of the register is fed back to one input of the adder, so that on each clock the output of the multiplier is added to the register. Combinational multipliers require a large amount of logic, but can compute a product much more quickly than the method of shifting and adding typical of earlier computers.

The MAC circuit must check for overflow, which might happen when the number of MAC operations is large. Overflow in a signed adder occurs when two operands with the same sign produce a result with a different sign.

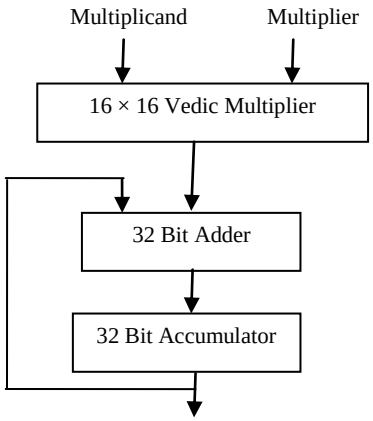

Fig 1: Architecture of MAC unit.

Fig. 1 shows the architecture of the proposed MAC unit. The two input 16 bit operand to the MAC unit are A[15:0] and B[15:0]. The 32 bit output from MAC unit is Q[31:0]. The proposed design uses one 16x16 Vedic multiplier using 'Urdhva Tiryagbhyam' algorithm [6-10], 32 bit accumulator using carry save adder, and one 32 bit register. Vedic multiplier can increase the MAC unit design speed. Carry save adder is used as an accumulator in this design. The Vedic multiplier and carry save adder in the MAC unit design enhance the MAC unit speed so as to gain better system performance. The product of Ai X Bi is fed back into the 32-bit Carry Save Adder and then added again with the next product Ai x Bi. This MAC unit is capable of multiplying and adding with previous product consecutively (Output = Σ Ai Bi).

When three or more operands are added simultaneously using two operand adders, the time consuming carry propagation is repeated several times. If the number of operands is 'n', then carries have to propagate (n-1) time's .In the carry save addition, the carry propagate only in the last step, while in all the other steps the partial sum and sequence of carries are generated separately.

The register module of MAC unit is implemented by using a 32 bit register. The 32 bit output from accumulator becomes input to 32 bit register. It produces 32 bit output.

3. RESULT AND DISCUSSION

Table-1 displays the comparison of combinational path delay of the proposed MAC module with optimized Vedic multiplier discussed in [1].

Table 1. Comparison of Combinational Path Delay

Device Spartan 2s200pq208: -6 Size	Delay of Optimized Vedic multiplier in ns described in [1]	Delay of Proposed MAC in ns
16-bit	22.604	6.884
32-bit	35.76	7.556

It is observed that for 16x16,and 32x32 proposed MAC module, the gate delay are 6.884ns,and 7.556ns while it is 22.604 ns, and 35.76 ns for the corresponding optimized Vedic multiplier described in [1].The total number of additions required in different bit size MAC are less compared to corresponding optimized Vedic multiplier due to the carry save adder used in MAC architecture. So MAC module uses less number of slices compared to optimized Vedic multiplier. Hence, the proposed MAC module implementation found to be most efficient in terms of speed as compared to the scheme presented in [1].

Table 2. Comparison of Synthesis Results

Algorithm Device spartan2s200pq208: -6	Proposed 16-bit MAC	Proposed 32-bit MAC
Delay	6.884 ns	7.556 ns
Number of Slices	121 out of 2352(5%)	481 out of 2352(20%)
Number of 4 input LUTs	214 out of 4704(4%)	850 out of 4704(18%)
Number of bonded IOBs	35 out of 140(25%)	67 out of 140(47%)

TABLE 2 shows the comparison of synthesis results of 16x16, 32x32, MAC architecture. For 16x16 MAC, the gate delay in the architecture is 6.884 ns with nearly 4% device utilization (number of slices: 121 out of 2352 (5%) and number of 4_input LUTs: 214 out of 4704(4%)). For 32x32 MAC, the gate delay in the architecture is 7.556 ns with nearly 18% device utilization (number of slices: 481 out of 2352 (20%) and number of 4_input LUTs: 850 out of 4704(18%)).

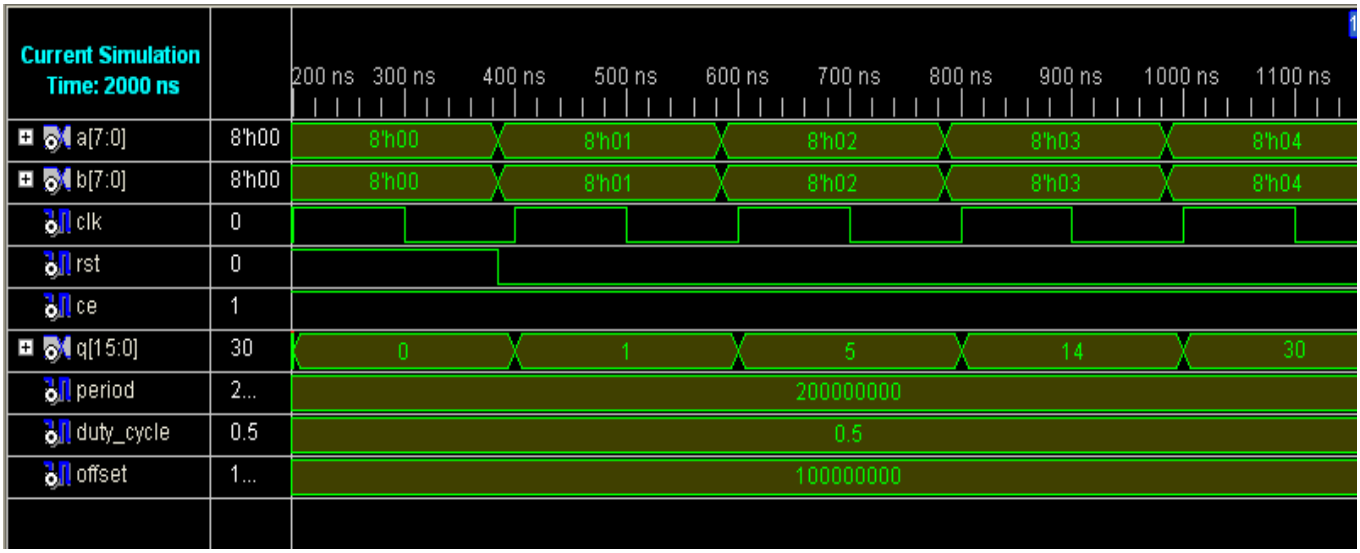

Fig 2: Simulation result of MAC unit.

Fig. 2 displays the simulation result of proposed MAC unit. The MAC unit, consisting of 16x16 Vedic multiplier followed by 32 bit carries adder and 32 bit accumulator register which stores the result when clocked. The output of the register is fed back to one input of the adder, so that on each clock the output of the multiplier is added to the register. The two input 16 bit operand to the MAC unit are A15:0] and B[15:0].The 32 bit output from MAC unit is Q[31:0]. Initially '0' decimal equivalent of 16 bit operand 'A' is multiplied with '0' decimal equivalent of 16 bit operand 'B' using 16x16 Vedic multiplier to produce result '0',which 32 bit equivalent of 'Q'. The result is stored in 32 bit accumulator register. Then the decimal equivalent of next two

16 bit numbers '1' and '1' are multiplied to get result '1', and added with previous result '0' to get the accumulated sum as '1'. Similarly '2' is multiplied with '2'to get result '4' , added with previous sum '1' to get accumulated sum '5' and so on.

4. CONCLUSION

The 16x16, and 32x32 bit proposed MAC module show improved speed improvements over optimized Vedic multiplier architecture presented in [1].Hence, the proposed MAC may be useful for high performance DSP processor.

5. REFERENCES

[1] M. Ramalatha , K. D. Dayalan , P. Dharani , and S. D. Priya , "High Speed Energy Efficient ALU Design using Vedic Multiplication Technique ," Lebanon , pp. 600-603, July 2009.

[2] P. Mehta, and D. Gawali, "Conventional versus Vedic Mathematical Method for Hardware Implementation of a Multiplier," International Conf. on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, India, pp. 640-642, 2009.

[3] H. S. Dhillon , and A. Mitra , "A Reduced-Bit Multiplication Algorithm for Digital Arithmetic ," International Journal of Computational and Mathematical Sciences, pp. 64-69, Spring 2008.

[4] V. A. Pedroni ," Circuit Design with VHDL ," 2008.

[5] M.Pradhan, R.Panda, "Design and Implementation of Vedic Multiplier", A.M.S.E. Journal,Series D,Computer Science and Statistics, France ,vol.15,Issue2,pp.1-19,July 2010.

[6] Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, "Vedic mathematics", Motilal Banarsidass Publishers Pvt. Ltd, Delhi, 2009.

[7] T.T Hoang, M. Sjalander, "Double Throughput Multiply Accumulate Unit for Flexcore Processor Enhancements", IEEE International Symposium on Parallel & Distributed Processing, pp 1-4, 2009.

[8] S.Kumarvel, Rmarimuthu, "VLSI Implementation of High Preference RSA Algorithm", International Conference on CIMA, pp 126-128,2007.

[9] Kihak Shin, Ik Kyun Oh, Sang Min, Beom Seom Ryu,Kie Young Lee and Tae Won Cho " A Multi-Level Approach to Low Power Mac Design" IEEE Trans.VLSI systems, vol 48 , pp 361- 763, 1999.

[10] S. Shanthala, C.P. Raj, "Design and VLSI Implementation of Pipelined Multiply Accumulate Unit", International Conference on ETET-09, pp 381-386, 2009.