

Implementation of Power Efficient Vedic Multiplier

Sree Nivas A
Amrita Vishwa Vidyapeethem
Coimbatore, India

Kayalvizhi N
Amrita Vishwa Vidyapeethem
Coimbatore, India

ABSTRACT

Vedic multiplier is based on the ancient algorithms (sutras) followed in INDIA for multiplication. This work is based on one of the sutras called “Nikhilam Sutra”. These sutras are meant for faster mental calculation. Though faster when implemented in hardware, it consumes more power than the conventional ones. This paper presents a technique to modify the architecture of the Vedic multiplier by using some existing methods in order to reduce power. The 32 X 32 Vedic multiplier is coded in Verilog HDL and Synthesized using Synopsys Design Compiler. The performance is compared in terms of area, data arrival time and power with earlier existing architecture of Vedic multiplier. The proposed design shows very good improvements in terms of power.

General Terms

Algorithms.

Keywords

Vedic Multiplier, Low power multiplier, Nikhilam Sutra.

1. INTRODUCTION

Multipliers play a vital role in any DSP processors or in any DSP applications. Multipliers are classified based on how the data is been processed. They are broadly classified as serial multipliers and parallel multipliers.

The add and shift multiplier [1] will work as normal array multiplier. The way of multiplying in this type of multiplier is similar to normal manual calculation. The parallel multipliers are mainly classified as array based and tree based multipliers. C.S Wallace proposed a tree based multiplier called Wallace tree multiplier [2] which operates at high speed. The irregularity in structure is the main disadvantage in this multiplier. Braun Multiplier is one of the parallel array based multiplier which requires n^2 gates and $(n-1)$ adders. So the area complexity increases (almost quadratically) with the number of bits. Baugh-Wooley multiplier [3] [4] is an array multiplier that can perform signed multiplication but again it has a drawback on area. A.D.Booth [5] introduced Booth multiplier for signed binary numbers. They are also called as radix-2 multiplier. This multiplier will not work when it has alternate zeros and ones, and it requires more number of additions and subtractions. This problem is overcome by Modified booth multiplier [6] or radix-4 multiplier, in which number of partial product rows is reduced by half. It will perform very well in terms of speed and power consumption. The main disadvantage of this type of multiplier is, it will not work for negative numbers.

Ancient Indian mathematics is called as Vedic Mathematics [7]. Vedic mathematics from Vedas was first proposed by Sri Bharati Krsna Tirtha, after his survey on Vedas [8]. Vedic

mathematics reduces the complexity in calculations that exist in conventional mathematics. Generally there are sixteen sutras available in Vedic mathematics. Among them only two sutras are applicable for multiplication operation. They are Urdhva Triyakbhyam sutra (literally means vertically and cross wise) and Nikhilam Sutra (literally means All from 9 and last from 10). The logic behind Urdhva Triyakbhyam sutra is very much similar to the ordinary array multiplier.

In the next section, the basic working of Vedic multiplier is explained with its architecture. In section 3, the circuit is modified for better speed and area. In section 4, the proposed method for improving the power is explained. In section 5, the implementation details are, and the result analysis are made.

2. VEDIC MULTIPLIER

This work mainly focuses on the implementation of Nikhilam Sutra. The operation behind this type of sutra is very simple [9].

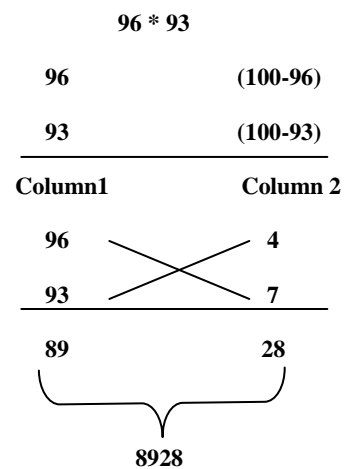


Fig 1. Example of multiplying two numbers using Nikhilam Sutra

The nearest base is chosen first. The multiplicand and the multiplier will be subtracted from the nearest base, which is equivalent to taking two's complement. Then the product of the two's complement and the common difference will give the final result.

Fig.1 shows an example [9] for multiplying two numbers $(96*93)$. In this case the common base is 100. Both 96 and 93 are subtracted from 100 which will give 4 and 7 respectively. The product of 4 and 7 is 28 and the common difference $(96 - 7)$ and $(93 - 4)$ is 89. By concatenating 89 and 28 the final result is obtained.

With this logic Manoranjan Pradhan et.al. [9] had proposed an architecture for Nikhilam sutra. Choosing a nearest base and subtracting the multiplicand and multiplier is equivalent to taking two's complement of the numbers. Fig.2 shows the architecture of the Nikhilam Sutra multiplier.

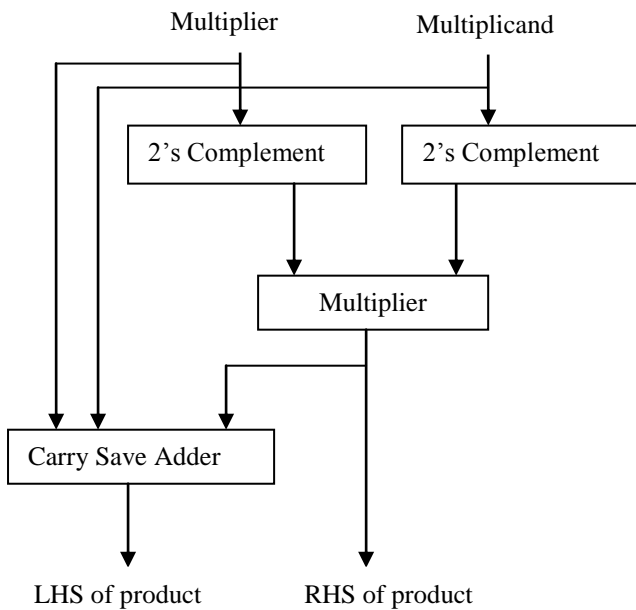


Fig 2: Architecture of Nikhilam Sutra

The two's complement is taken for the Multiplier and the Multiplicand. The output of the two's complement is then multiplied. If it is a $N \times N$ bit multiplication this multiplier will have $2N$ number of bits as output. The lower half i.e. the last (N) bits will be the result of the multiplier and the first N bits will be the result of the carry save adder.

Nikhilam sutra multiplier will operate at high speed [9], but the power consumed by this multiplier will also be high. The power can be reduced by making modifications in the architecture shown in Fig.2. The architecture mainly consists of two two's complement block, a multiplier block and an adder block. The modifications can be done at the two's complement block and the multiplier block in order to reduce the power consumption.

3. TWO'S COMPLEMENT BLOCK MODIFICATION

Jung-Yup Kang et.al. [10] have proposed a new way of finding a two's complement. In conventional way the binary

number is complemented and one is added to the complemented number. In their proposed work the first step was to find the right most one of the binary number. Then the bits before the right most one is complemented and the remaining bits are left unchanged. The advantage of finding two's complement by this method is that the addition stage will not be necessary here. So the power consumed by the adder will be reduced.

4. MULTIPLIER BLOCK MODIFICATION

V.S.Dimitrov et.al. have proposed [11][12] a multiplier based on multiple radix representation which consumes very less power. The multiple radix multiplier is based on Double Base Number System (DBNS).

Definition : Given p, q are two different prime numbers, the double base number system is a representation scheme in which every positive integer n is represented as the sum or difference of $\{p, q\}$ – integers.

$$n = \sum_{i=1}^l s_i p^{a_i} q^{b_i} \quad (1)$$

where $s_i \in \{-1, 1\}$ and a_i, b_i are nonnegative integers. Equation (1) gives the general representation of a double base number system (DBNS). The term l represents size or the length of the DBNS. In general $\{2, 3\}$ are used as bases. There are many ways to represent a particular number in DBNS. This type of representation is called canonic representation. The representation among many possibilities is said to be good, when it contains minimum number of $\{2, 3\}$.

Any one either the multiplier or the multiplicand is broken into a segment of seven bits for encoding. Here the 7-bit (for example A) is encoded into 11-bits. These 11-bits will act as the control bits in the architecture of the multiple radix multiplier. Fig. 3 shows how the 7-bits is encoded into 11-bits.

	Digit Selection	Binary exponent	Digit Selection	Sign	Binary exponent
A (7-bits Number) →	2 bits	3 bits	2 bits	1 bit	3 bits

Fig 3: Encoding scheme for 7-bit number

The first two bits represents the digit selection bits, next three represent binary exponent, next two represent the digit selection, next one bit represent sign bit and the last three bits represent binary exponent. From Table 1 we can see that 89 is encoded as (01 101 11 0 000). As it is a seven bit encoding, the odd numbers till 7 is taken as {1, 3, 5, 7}.

This multiple radix multiplier will work on LUT based approach. Larger the LUT's are, fewer the number of additions/subtractions will be used. To perform 32 bit multiplication, numbers from 0 to 127 is converted into canonical representation with minimum number of {2,3} and it should be stored in the LUT. The conversion is made from 0 to 127 because as it is a 7-bit encoding the maximum possibility will be 2^7 (128 data i.e. from 0 to 127). Table 5 of [11] is the LUT [4] for a seven bit encoding.

From Table 5 of [11] we can see that 89 is encoded as (01 101 11 0 000). As it is a seven bit encoding, the odd numbers till 7 is taken as {1, 3, 5, 7}.

Fig.4 shows an example how the encoding is done for 89. The first block represents 01, so the first value from the set i.e. 3 is taken. The second block is 101 which is equivalent to 5.

So 3 is multiplied with 2^{101} which will give 96. The fourth block is 0 which represents negative sign and if it is 1 then it is positive. The third block is 11, so the fourth value from the set i.e. 7 is taken. The fifth block is 000 which is equivalent to zero. So 7 is multiplied with 2^{000} which will give 7. Finally (96-7) will give us 89.

$$\begin{array}{r} 3 * 2^{101} = 3 * 32 = 96 \\ 7 * 2^{000} = 7 * 1 = 7 \\ \hline 89 \end{array}$$

Fig 4: Example of encoding a number 89

These 11 bits will act as a control bits for the architecture [11] shown in Fig 8. The multiplicand A is broken into segments of 7 bits. After splitting, every 7 bits are encoded into 11 bits. The 32 bit multiplier B is made used to find the values of 3B, 5B and 7B. These values are found by using shift and addition/subtraction method. The main advantage of this method is the number of adders will get reduced when compared to conventional way of multiplying.

The first block (2 bits) of the encoded output will act as a control signal t_1 for a 4X1 multiplexer. The second block (3 bits) will be the control signal a_1 for a barrel shifter. The third block (2 bits) of the encoded output will act as a control signal t_2 for a 4X1 multiplexer. The fourth block (1 bit) will be the sign bit control signal s_2 for a addition/subtraction block. The fifth block (3 bits) will be the control signal a_2 for a barrel shifter.

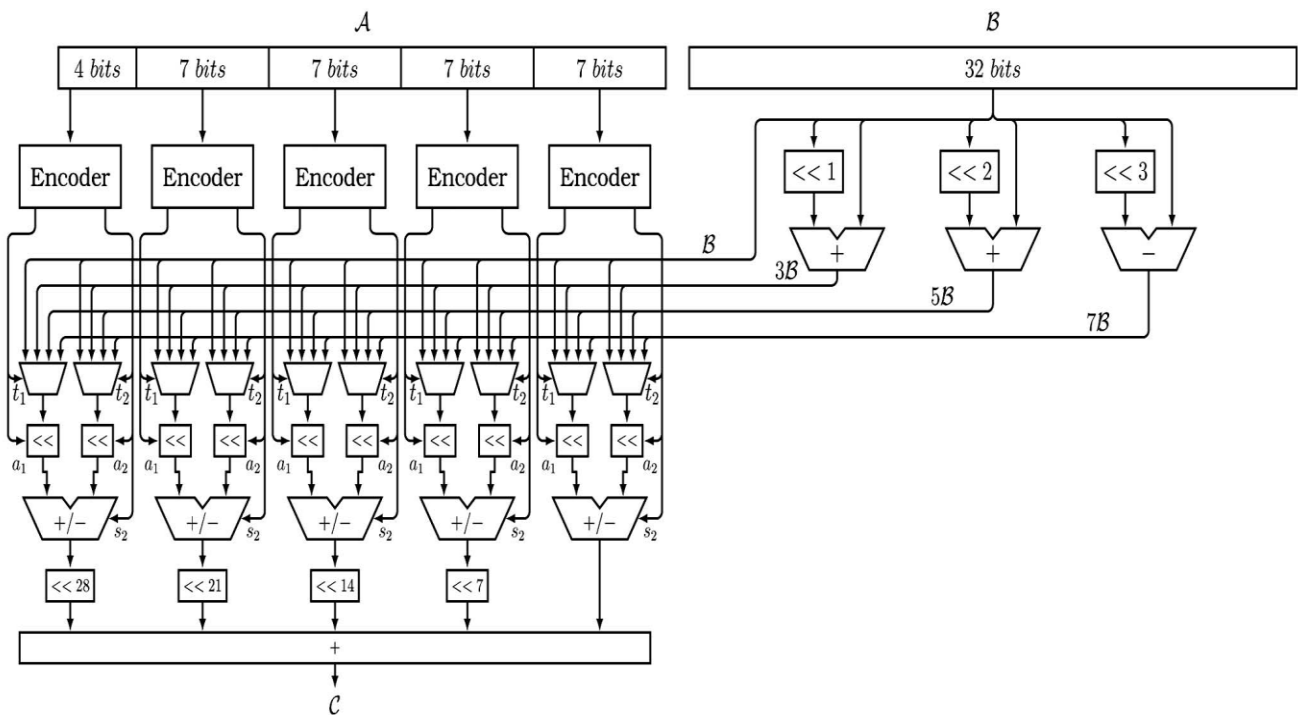


Fig 5: Architecture of Multiple radix multiplier using 7-bit encoding

The performance of the multiplier after such modifications is compared with the unmodified Vedic multiplier.

5. RESULT AND COMPARISION

All implementations are done using Verilog HDL and the performances of the multipliers are compared on the basis of area, data arrival time and power consumed. The 32X32 multipliers are synthesized with Synopsys Design Compiler using lsi_10k.db library file. Table.1 shows the comparison of the performance of Vedic multiplier in three cases.

Table 1. Comparison of synthesis report of 32 X 32 Vedic multiplier

Case	Area μm^2	Data Arrival Time ns	Power mW
1. Vedic Multiplier without any Modification	8364	99.43	2.3077
2. Vedic Multiplier with modified Two's complement block	8274	97.69	2.2966
3. Vedic Multiplier with modified Two's complement and multiplier blocks	12167	158.71	1.2215

The first case in Table 1 shows the actual 32X32 Vedic multiplier without any modification as shown in Fig.2. We can observe that the power consumed by the multiplier is very high. The result of case 2 is obtained by modifying only the two's complement block in the Fig.2 by the method mentioned in section 3. From this we can observe that there is little bit improvement in all three parameters. The result of case 3 is obtained by modifying both the two's complement block and the multiplier block in the Fig.2 by the method mentioned in section 3 and section 4 respectively. From this case we can see that the power is reduced to an extent but we should compensate in terms of area and delay.

6. CONCLUSION

In this paper, a new architecture for a power efficient Vedic multiplier is presented. Simulation and synthesis are carried out for three different cases. The multipliers can be chosen depending on what the performance it is required. The proposed Vedic multiplier architecture shows good

improvement in power consumption in third case. This multiplier can be efficiently used in any DSP application or in any DSP processors that requires low power consumption.

7. ACKNOWLEDGEMENTS

We thank Dr. S. Krishna Kumar of Amrita Vishwa Vidyapeetham for his constant support in performing the hardware implementation.

8. REFERENCES

- [1] C.N.Marimuthu, P.Thangaraj, Aswathy Ramesan, "Low power Shift and add multiplier design", International Journal of computer science and information technology, vol.2, no.3, June 2010.
- [2] Wallace, C S. "A suggestion for a fast multiplier," IEEE Transactions on Electronic Computers, vol EC-13, pp 14-17, Feb 1964.
- [3] C.R Baugh and B.A Wooley," A Two's Complement Parallel Array Multiplication Algorithm," IEEE Transactions on Computers, Vol. 22, No 12, pp.1045-1047, December 1973.
- [4] Aswathy Sudhakar , D. Gokila , "Proposal for an Efficient Reconfigurable Fixed-Width Multiplier", ICNVS'10 Proceedings of the 12th international conference on Networking, VLSI and signal processing, ISBN: 978-960-474-162-5,2010
- [5] A.D. Booth, "A signed Binary Multiplication Technique," Quart. J.Mech. Appl. Math.v, 4 part2, pp 236-240, 1951.
- [6] P.E.Madrid, B.Miller and E.E.Swartzlander," Modified Booth Algorithm for High Radix Fixed -Point Multiplication," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol .1, No. 2, pp 118-121 June 1993.
- [7] Honey Durga Tiwari, Ganzorig Gankhuyag, Chan Mo Kim, Yong Beom Cho, "Multiplier design based on ancient Indian Vedic Mathematics," in IEEE International SoC Design Conference, pp. II-65 - II-68, November 2008.
- [8] Jagadguru Swami, Sri Bharati Krishnsna Tirthji Maharaja," Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.
- [9] Manoranjan Pradhan, Rutuparna Panda, Sushanta Kumar Sahu, "Speed Comparison of 16x16 Vedic Multipliers," International Journal of Computer Applications (0975 – 8887), vol 21– No.6, May 2011.
- [10] Jung-Yup Kang and Jean-Luc Gaudiot, "A Simple High - Speed Multiplier Design," IEEE trans. on computers, vol. 55, no. 10, pp. 1253-1258, October 2006.
- [11] V. Dimitrov, K. Jarvinen, and J. Adikari, "Area-efficient multipliers based on multiple-radix representations," IEEE Transactions on Computers, vol. 60, no. 2, pp. 189 -201, February 2011.
- [12] A. Edirisuriya and A. Madanayake, J. Adikari, v. S. Dimitrov, "An Architecture For A 7 x 7-bit Multiple-Radix Multiplier Building Block," in IEEE 54th International Midwest symposium , pp 1- 4, September 2011.