Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 5, Issue 3, June 2021
Part of the Proceedings of the 1st All India Paper writing Competition on Emerging Research - PaCER 2020

Design of Multipliers Using Various Methods

Pragnya Patil
M.Tech [VLSI & ES], Department of ECE, BNM Institute of Technology, Bengaluru, India,
patilpragnya96@gmail.com

Dr. Subodh Kumar Panda
Professor, Department of ECE, BNM Institute of Technology, Bengaluru, India,
subodhpanda2013@gmail.com

Abstract: Multipliers are highly on demand as they are used in tremendous areas such as digital signal processing applications, image processing application , and in various Microprocessors such as ARM, NVIDIA, Intel, DSP Processors such as DM270,DM320 etc. Many researchers are urging to improve the performance of the multipliers by adopting various methods. Most of the existing papers includes the work on Array multipliers, Wallace tree multiplier, Dadda multiplier and Booth multipliers. And few recent papers include Vedic Mathematics that have implemented most commonly used Urdhva Sutra. In this paper, various methods are implemented to design a multiplier such as Booth multiplier, Modified Booth multiplier, Urdhva multiplier and Nikhilam multiplier. All the methods are used to design for 8bit signed numbers by coding very efficiently in Verilog. This paper focuses on the design of multipliers that are very simple in structure such that there is no circuit complexity. The design is simulated using Xilinx ISE or ModelSim SE.

Keywords: Booth; Modified Booth; Vedic Mathematics; Urdhva Sutra; Nikhilam Sutra; Multipliers

I. INTRODUCTION

The various fundamental arithmetic operations such addition, subtraction and multiplication are used in VLSI system designs such as Arithmetic Logic Units. Among the arithmetic operations , Multiplication is the one which consumes more time to compute and hence researchers are exploring to implement various different multipliers to improve the performance parameters. In this digital world many applications rely on processors which are most important in any smart devices. Thus, these processors require arithmetic logic unit and hence there is a need to design multipliers with different structures and methods that are suitable for respective applications in any fields such as DSP applications, Image Processing applications, IOT applications which uses various ARM processors etc.

Multipliers generate various partial products and these partial products keep increasing with increased number of bits. In this paper Booth multiplier and Modified booth multipliers are designed for 8 bit signed numbers. And it is observed that Modified Booth Multiplier requires less number of iterations to perform the operation than Booth multiplier. From the Vedic mathematics two sutras has been used for multiplication of largest numbers with easiest method. In this paper Urdhva and Nikhilam sutra based multipliers were also designed for 8 bit signed numbers.

II. RELATED WORK

The survey on existing papers based on Multipliers is done. The survey papers consist of various methods such as Wallace tree, Dadda and Urdhva methods. Array multiplier is designed which requires large number of gates and hence area is more and hardware complexity is high [1]. Wallace tree multiplier used various full and half adders. By using 4:2 , 5:2 and 6:2 compressors the performance of the Wallace tree can be improved such that complexity of the circuit and delay can be reduced [2]. The Dadda tree multiplier uses carry propagation adder, many full adders and half adders. But area occupied by Dadda tree is less when compared to Wallace tree[2]. A multiplier is designed using Vedic mathematics sutra called Urdhva. Vedic multiplier is designed using carry save adder[3]. From the related work we can know that various methods used over here used many internal blocks such as carry save adder , carry propagation adder and so on and the circuit complexity is high.

III. METHODOLOGY

In this paper multipliers are designed based on Booth algorithm, Modified Booth algorithm and Urdhva and Nikhilam sutra. All the four methods are designed for 8 bit signed numbers without circuit complexity. The algorithms are explained and output waveforms are added in the result section.

A. Booth Algorithm

Booth multiplication algorithm is an algorithm that multiplies signed binary numbers in two’s complement notation. Where ‘A’ is multiplicand and ‘B’ is multiplier. ’X’ is the output.

B $_{i+1}$	**B$_i$**	**Cases**
0	0	Arithmetic shift right (ASR) X by 1
1	1	Arithmetic shift right (ASR) X by 1
0	1	+A and ASR by 1
1	0	-A and ASR by 1

Table 1. Booth multiplier logic

A zero is appended to the LSB of two's complement of Multiplier 'B'. Consider B_i and B_{i+1} from LSB to MSB pair wise and follow the Table I i.e. if the bit pair is 00 and 11 then do arithmetic right shift by 1. If the bit pair is 01 do +A and ASR by 1 and if 10 then do –A and ASR by 1. Continue the procedure till last iteration.

B. *Modified Booth Multiplier*

Modified Booth algorithm also multiplies the signed binary numbers in two's complement notation as Booth multiplier. But, in Modified Booth Multiplier pair of 3-bits is considered from LSB to MSB.

B_{i+2}	B_{i+1}	B_i	Cases
0	0	0	Arithmetic shift right by 2
0	0	1	+A and Arithmetic shift right by 2
0	1	0	+2A and Arithmetic shift right by 2
0	1	1	-2A and Arithmetic shift right by 2
1	0	0	-A and Arithmetic shift right by 2
1	0	1	-A and Arithmetic shift right by 2
1	1	0	-A and Arithmetic shift right by 2
1	1	1	Arithmetic shift right by 2

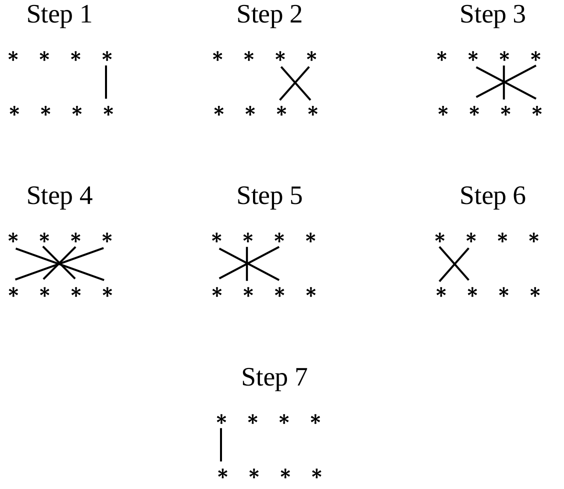
Table 2. Modified Booth multiplier logic

A zero is appended to the LSB of two's complement of Multiplier 'B'. Consider Bi, B i+1 and Bi+2 from LSB to MSB pair wise and follow the Table I i.e if the bit pair is 000 and 111 then do arithmetic right shift by 2. If the bit pair is 001 do +A and ASR by 2 and if 010 then do +2A and ASR by 2 and so on as per the Table II.

C. *Urdhva Multiplier*

Urdhva Tiryagbhyam is the ancient Sutra of Vedic Mathematics and is the easiest method for multiplication. Urdhva Tiryagbhyam means "Vertically and Crosswise" method.

The steps for Urdhva method are shown below:



The above steps are followed to multiply between two numbers.

For example, 672 x 451

6 7 2

4 5 1

1. First multiply column of RHS,

   2 X 1= 2, write 2, carryover 0.

2. Multiply diagonally crosswise,

   0 + 2 x 5 + 7 x 1 = 0 + 10 +7 = 17, write 7, carryover 1.

3. 1 + 2 x 4 + 6 x 1 + 7 x 5 = 1 + 8 + 6 + 35 = 50, write 0, carryover = 5.

4. 5 + 7 x 4 + 6 x 5 = 5 + 28 + 30 = 63, write 3, carryover 6.

5. 6 + 6 x 4 = 6 + 24= 30

   Therefore, 672 x 451 = 303072.

D. *Nikhilam Multiplier*

Nikhilam is the second sutra of 16 sutras of Vedic Mathematics and is the easiest and shortcut method for multiplying any number of digits.

Nikhilam sutra is "all from 9 and last from 10". This is applied to multiply 2 numbers which are near to the same decimal bases (10,100,1000 etc).

The method is:

1. First find out that the given is nearer to which base like 10, 100, 1000 etc.
2. The difference of the given numbers from their base is multiplied.
3. Add or subtract the deviations (based on whether the number is more or less than the base) from the base.

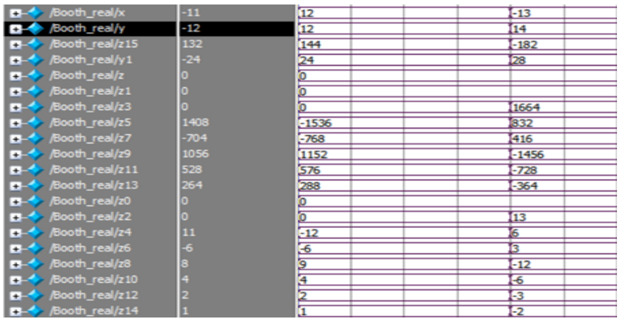
   For example, 94 x 95

- 94 and 95 are nearer to the base 100.
- 100 – 94 = 6 & 100 – 95 = 5, multiply 6 x 5 = 30
- Subtract 6 & 5 from 100 i.e. 100 – 6 – 5 =89
- Combine 30 and 89 i.e. from previous stage output, 8930.
- Thus, multiplication of 94 and 95 from Nikhilam method is 8930 which is absolutely correct result for given numbers and within very few steps we have obtained the answer.

  The similar approach is chosen was multiplying numbers bitwise, each decimal number is replaced with the bit value of the numbers considered. For, Nikhilam, instead of powers of 10, powers of 2 is considered.

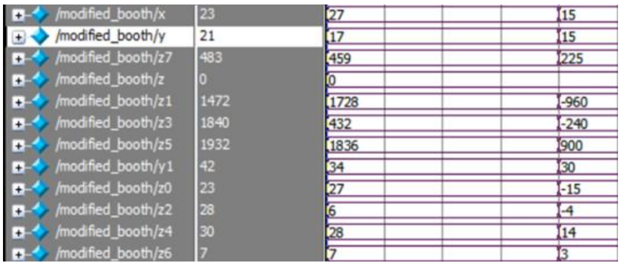
IV. RESULTS

Results obtained from the simulation window of Xilinx are found in below figures. Fig 1 is 8 bit signed Booth multiplier and Fig 2 is of 8 bit signed Modified Booth multiplier. Also, Vedic multipliers such as Urdhva and Nikhilam results for 8 bit signed numbers are shown in Fig 4 and Fig 5.



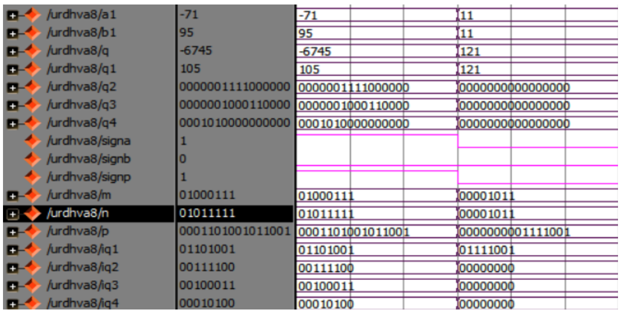
```
/Booth_real/x     -11    12      -13
/Booth_real/y     -12    12      14
/Booth_real/z15   132    144     -182
/Booth_real/y1    -24    24      28
/Booth_real/z     0      0
/Booth_real/z1    0      0
/Booth_real/z3    0      0       1664
/Booth_real/z5    1408   -1536   832
/Booth_real/z7    -704   -768    416
/Booth_real/z9    1056   1152    -1456
/Booth_real/z11   528    576     -728
/Booth_real/z13   264    288     -364
/Booth_real/z0    0      0
/Booth_real/z2    0      0       13
/Booth_real/z4    11     -12     6
/Booth_real/z6    -6     -6      3
/Booth_real/z8    8      9       -12
/Booth_real/z10   4      4       -6
/Booth_real/z12   2      2       -3
/Booth_real/z14   1      1       -2
```

Fig 1. Simulation results of 8 bit Booth Multiplier.



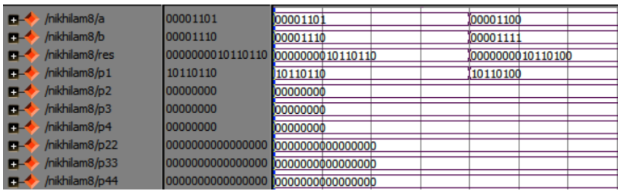
```
/modified_booth/x    23     27     15
/modified_booth/y    21     17     15
/modified_booth/z7   483    459    225
/modified_booth/z    0      0
/modified_booth/z1   1472   1728   -960
/modified_booth/z3   1840   432    -240
/modified_booth/z5   1932   1836   900
/modified_booth/y1   42     34     30
/modified_booth/z0   23     27     -15
/modified_booth/z2   28     6      -4
/modified_booth/z4   30     28     14
/modified_booth/z6   7      7      3
```

Fig 2. Simulation results 8-bit Modified Booth Multiplier.



```
/urdhva8/a1      -71                -71                11
/urdhva8/b1      95                 95                 11
/urdhva8/q       -6745              -6745              121
/urdhva8/q1      105                105                121
/urdhva8/q2      0000001111000000   0000001111000000   0000000000000000
/urdhva8/q3      0000001000110000   0000001000110000   0000000000000000
/urdhva8/q4      0001010000000000   0001010000000000   0000000000000000
/urdhva8/signa   1
/urdhva8/signb   0
/urdhva8/signp   1
/urdhva8/m       01000111           01000111           00001011
/urdhva8/n       01011111           01011111           00001011
/urdhva8/p       0001101001011001   0001101001011001   0000000001111001
/urdhva8/iq1     01101001           01101001           01111001
/urdhva8/iq2     00111100           00111100           00000000
/urdhva8/iq3     00100011           00100011           00000000
/urdhva8/iq4     00010100           00010100           00000000
```

Fig 3. Simulation results of 8-bit Urdhva Multiplier.



```
/nikhilam8/a     00001101           00001101           00001100
/nikhilam8/b     00001110           00001110           00001111
/nikhilam8/res   0000000010110110   0000000010110110   0000000010110100
/nikhilam8/p1    10110110           10110110           10110100
/nikhilam8/p2    00000000           00000000
/nikhilam8/p3    00000000           00000000
/nikhilam8/p4    00000000           00000000
/nikhilam8/p22   0000000000000000   0000000000000000
/nikhilam8/p33   0000000000000000   0000000000000000
/nikhilam8/p44   0000000000000000   0000000000000000
```

Fig 4. Simulation results of 8-bit Nikhilam Multiplier.

V. CONCLUSION

In this paper, Booth, Modified Booth and Vedic Mathematics based algorithms such as Urdhva and Nikhilam sutra based multipliers are designed for 8 bit signed numbers. The various multiplier designed over here does not contain complex blocks and thus there is no circuit complexity.

REFERENCES

[1] Sumit Vaidya and Deepak Dandekar, “Delay-Power performance comparison of multipliers in VLSI circuit design”, International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010 .

[2] R. Alwin, Dr. R. Naveen, “A Review on Multiplier Circuits based on Various Performance Parameters”, International Journal for Research in Applied Science & Engineering Technology, Volume 5 Issue V, May 2017.

[3] Yogita Bansal, Charu Madhu, Pardeep Kaur, “High speed vedic multiplier designs”, Proceedings of 2014 RAECS UIET Panjab University Chandigarh, 06 – 08 March, 2014.

[4] S. R. Rupanagudi et al., "Design of a low power Digital Down Converter for 802.16m - 4G WiMAX on FPGA," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 2303-2308.

[5] S. Rao Rupanagudi et al., "A Further Optimized Mix Column Architecture Design for the Advanced Encryption Standard," 2019 11th International Conference on Knowledge and Smart Technology (KST), Phuket, Thailand, 2019, pp. 181-185.

[6] K.Deergha Rao, Ch. Gangadhar, Praveen K Korrai, “FPGA Implementation of Complex Multiplier Using Minimum Delay Vedic Real Multiplier Architecture”, 2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON) Indian Institute of Technology (Banaras Hindu University) Varanasi, India, Dec 9-11, 2016.

[7] Sukhmeet Kaur , Suman and Manpreet Signh Manna, “Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)”, Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690 © Research India Publications.

[8] Prof S. B. Somani , Nikhil R. Mistri, “Study of Vedic Multiplier Algorithms using Nikhilam Method” , International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 5, Issue 5, May 2016