Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.536 – 541

RESEARCH ARTICLE

High Performance Pipeline Signed 64*64 bit Multiplier using Radix-32 Modified Booths Algorithm and Wallace Structure

Monika Mane
E&TC DEPT
Savtribai phule
university india
manemonika1993@gmail.com

Supriya Mokashi
E&TC DEPT
Savtribai phule
university india
mokashi.supriya12@gmail.com

Archana Bharude
E&TC DEPT
Savtribai phule
university india
archanabharude@gmail.com

Kanchan Maske
E&TC DEPT
Savtribai phule
university india
kanchanmtech@gmail.com

Abstract— This paper mainly emphasis on improving speed performance of signed multiplication using radix 32 modified Booths algorithm and Wallace Structure. It is intended for fixed length 64x64 bit operands. 3:2and 4:2 Compressors used in Wallace tree structure gather partial products. Using both compressor, Number of levels has been compact that also origins increasing the speed of multiplier. An effective VHDL code has been written and effectively produced and replicated using Xilinx ISE 9.2i and Model Sim PE Student Edition 10.2c. Proposed pipelined signed 64*64 bit multiplier using radix 32 Booths algorithm and Wallace tree structure delivers fewer delay 1.4 ns and required 87% minimum number of levels in Wallace tree structure, 76% less total number of Compressors, 70% less generated partial products as compared to conventional multipliers.

Keywords— Xilinx8.1, VLSI, FPGA, Wallace tree, Booth Algorithm

I. INTRODUCTION

Multiplication is a one of the most significant process in processing of digital signal and DSP system. Speed performance of the multiplication influents overall performance of the digital computer systems as well as in digital signal processing. Many high enactment algorithms and architectures have been suggested to improve and accelerate multiplication operations. In the recommended algorithm, issues of designing multiplier are formation of partial products plus addition of partial products. The speed performance of multiplication can be increased by reducing the no. of partial products in radix-32 Booth Algorithm. For decreasing the delay of addition of partial products of Wallace Tree Structure has been used. This paper presents Pipelined Signed 64*64 bit multiplier. Radix 32 algorithm generates less number of partial products than that in the radix 16 algorithm hence we are using radix 32 algorithm. Instead of array structure, Wallace Tree Structure has been used since in Wallace Tree structure, number of levels required is less in comparison of array structure.

II. MULTIPLIER STRUCTURES

A. Wallace Structure

Wallace tree is famous for their optimum computation time, while adding multiple operands to 2 outputs by using 3:2 or 4:2 compressors or both. Wallace tree guarantees the lowest overall delay Figure1 shows nine operands Wallace structure, where 3:2 compressor compress the data having three multiple bit inputs and 2 multiple bit outputs. 4:2 compressor compress the data having 4 multiple bit inputs and 2 multi-bit outputs. Using both compressor, No. of levels has been reduced that also origins improving the speed of multiplier. For 64X64 operands using radix-32, total 7 compressors component and 4 levels has been used in Wallace tree that is effectively less in comparison to radix-16, in which 14 compressors and 6 levels have been used in Wallace tree structure so the total performance of multiplication has been effectively improved because of less total gate delay in high performance pipelined signed 64*64 multiplier using Wallace structure and radix 32 Booths Algorithm.

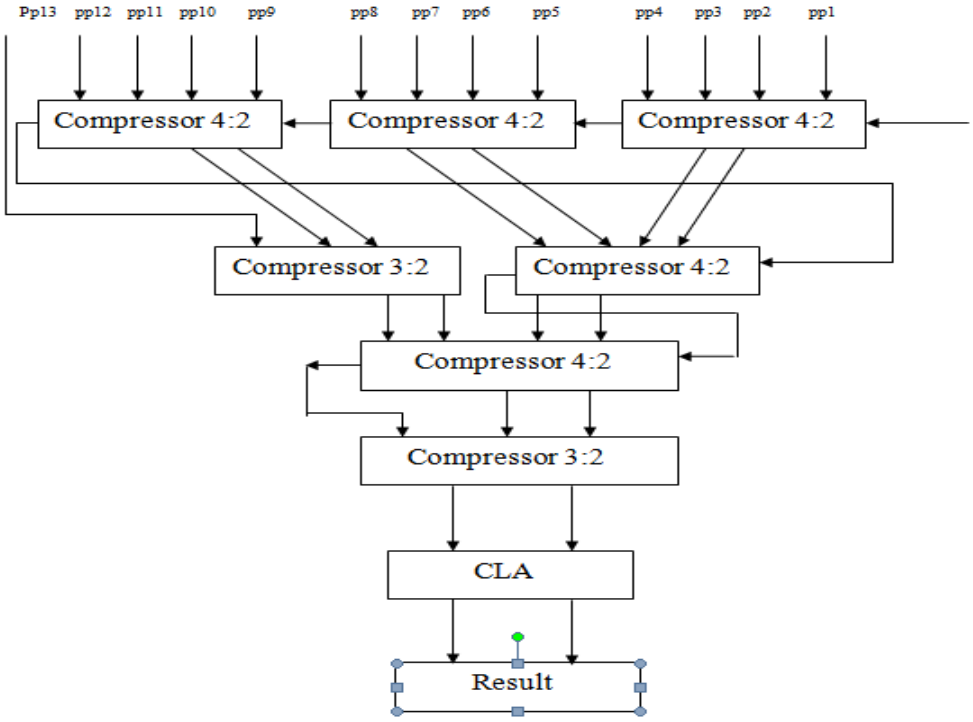

Fig 1: wallace tree structure multiplier

B. Array multiplier

Array is a straight forward way to accumulate partial products using a number of compressors. The n-operand array consists of n-2 compressor. Fig2 shows an array for 6-operand, producing 2 outputs, where compressor compress the data having three multi-bit inputs and two multi-bit outputs.

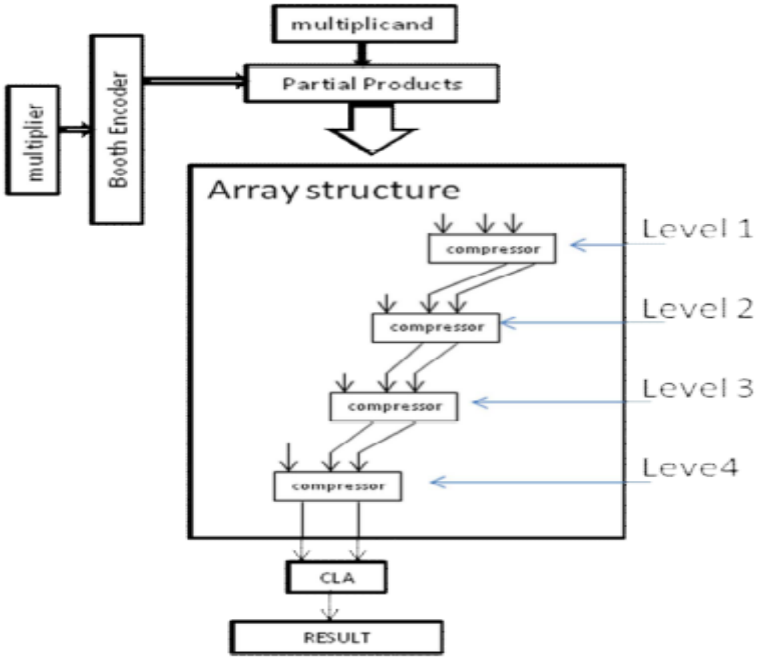

Fig 2: Array Multiplier

III. DESIGN OF HIGH PERFORMANCE MULTIPLIER

A. Radix-32 Proposed Modified Booth Algorithm

In radix 32 booth algorithm the multiplier is divided into the 13 group each group having 6 bits.in 1st group 1st bit is consider '0' and other bits are least significant five bit of multiplier operand. In 2nd group 1st bit is most significant bit of1st group and other bits are next five bit of multiplier operand.
In 3rd group, 1st bit is most significant bit of 2nd group and other bits are next five bit of multiplier operand. This process is carried on for all 13 groups.

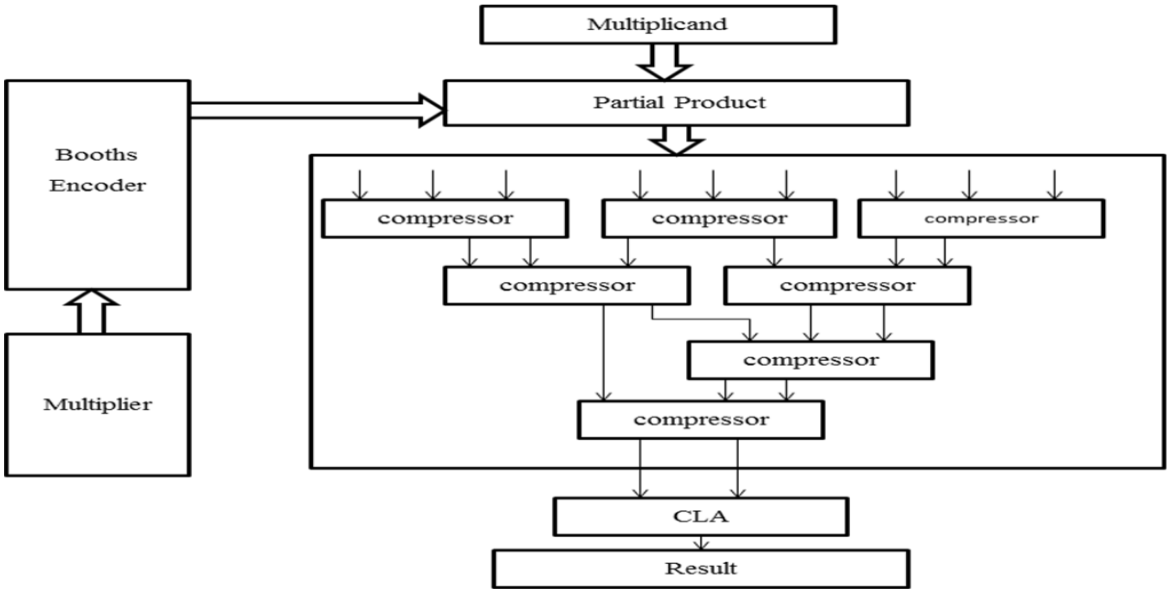

Fig 3: Architecture of proposed piplined signed 64*64 bit multiplier

For n bit multiplier there is n/5 or [n/5 + 1] groups and partial products is proposed modified Booth Algorithm radix-32. Table I is for Proposed radix-32 modified Booth algorithm. It has been planned and radix-32 Register Transfer Logic level view of modified Booth encoder also has been designed that is shown in fig 3.

$$\mathbf{Fi = (k1 + 2\ k2 + 4\ k3 + 8\ k4 + 16\ k5\ )}$$

Where k1, k2, k3, k4, k5 = 0 when 00 or 11

k1, k2, k3, k4, k5 = +1 when 01

k1, k2, k3, k4, k5 = -1 when 10

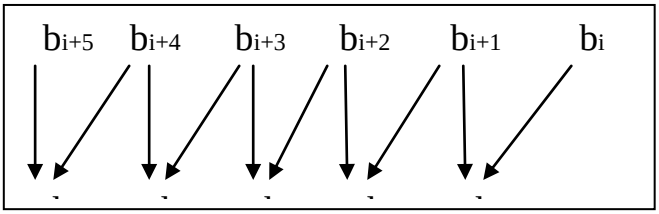

Fig 4: Finding value of k_1, k_2, k_3, k_4, k_5

TABLE I
PROPOSED RADIX-32 MODIFIED BOOTH ALGORITHM

Multiplier bits(g) b_{i+5},b_{i+4},b_{i+3}, b_{i+2},b_{i+1},b_i	Operation for group Fi	Multiplier bits b_{i+5},b_{i+4},b_{i+3}, b_{i+2},b_{i+1},b_i	Operation for Group
0 0 0 0 0 0	0	1 0 0 0 0 0	-16A
0 0 0 0 0 1	+1A	1 0 0 0 0 1	-15A
0 0 0 0 1 0	+1A	1 0 0 0 1 0	-15A
0 0 0 0 1 1	+2A	1 0 0 0 1 1	-14A
0 0 0 1 0 0	+2A	1 0 0 1 0 0	-14A
0 0 0 1 0 1	+3A	1 0 0 1 0 1	-13A
0 0 0 1 1 0	+3A	1 0 0 1 1 0	-13A
0 0 0 1 1 1	+4A	1 0 0 1 1 1	-12A
0 0 1 0 0 0	+4A	1 0 1 0 0 0	-12A
0 0 1 0 0 1	+5A	1 0 1 0 0 1	-11A
0 0 1 0 1 0	+5A	1 0 1 0 1 0	-11A
0 0 1 0 1 1	+6A	1 0 1 0 1 1	-10A
0 0 1 1 0 0	+6A	1 0 1 1 0 0	-10A
0 0 1 1 0 1	+7A	1 0 1 1 0 1	-9A
0 0 1 1 1 0	+7A	1 0 1 1 1 0	-9A
0 0 1 1 1 1	+8A	1 0 1 1 1 1	-8A
0 1 0 0 0 0	+8A	1 1 0 0 0 0	-8A
0 1 0 0 0 1	+9A	1 1 0 0 0 1	-7A
0 1 0 0 1 0	+9A	1 1 0 0 1 0	-7A
0 1 0 0 1 1	+10A	1 1 0 0 1 1	-6A
0 1 0 1 0 0	+10A	1 1 0 1 0 0	-6A
0 1 0 1 0 1	+11A	1 1 0 1 0 1	-5A
0 1 0 1 1 0	+11A	1 1 0 1 1 0	-5A
0 1 0 1 1 1	+12A	1 1 0 1 1 1	-4A
0 1 1 0 0 0	+12A	1 1 1 0 0 0	-4A
0 1 1 0 0 1	+13A	1 1 1 0 0 1	-3A
0 1 1 0 1 0	+13A	1 1 1 0 1 0	-3A
0 1 1 0 1 1	+14A	1 1 1 0 1 1	-2A
0 1 1 1 0 0	+14A	1 1 1 1 0 0	-2A
0 1 1 1 0 1	+15A	1 1 1 1 0 1	-1A
0 1 1 1 1 0	+15A	1 1 1 1 1 0	-1A
0 1 1 1 1 1	+16A	1 1 1 1 1 1	0

B. design process for proposed modified multiplier

In the proposed multiplier, modified Booths Algorithm based on radix-32 has been made for increasing speed performance of multiplication, which produces less number of partial products. Multiplier operand is divided into 13 groups of 6 bits. Then to form the partial products these 6 bits of group is then multiplied with the multiplicand operand that forms a group. To find the factor according to this group of bits we are using modified booths algorithm. Then the corresponding factor has been multiplied with multiplicand operand and generate analogous partial product of corresponding group, this computation is performed by Booth Encoder. Now the partial products have been generated for each group of multiplier operand.

So by using modified radix 32 Booths Algorithm we can generate less number of partial products. This tree

structure has been used for enhancing speedy multiplication. Wallace tree structure is a straightforward way to accumulate partial products using a number of compressors. Using 'n:2' compressors in Wallace tree structure, partial products have been summed in less delay. There is less number of gates from initial bit to final bit rather than array structure. Computation of multiplier has been commutated over fig 6 and fig 7.

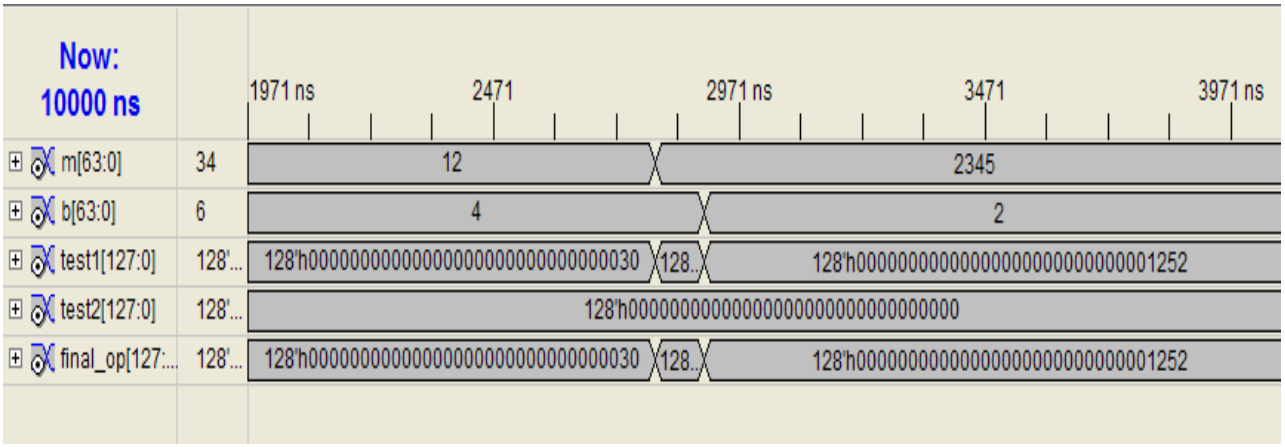

Fig 5: Simulation of radix-32, pipelined and signed 64x64 bit multiplier

32 modified booth multiplier has been designed. Comparison of the proposed multiplier with Array structure multiplier and 32x32 bits multiplier using radix-16, the signed 64x64 bits multiplier using radix-16. Booth Encoder is given in table II. It shows that proposed pipelined signed 64x64 bits multiplier using Wallace structure and radix-32 modified Booth Algorithm required 70% less number of groups of bits of multiplier operand, less number of partial products using proposed radix-32 booth algorithm, 76% less total number compressor also 89% less levels in Wallace tree structure in comparison with conventional Array structure multiplier and signed 64x64 bit multiplier using radix-16. So overall performance of proposed pipelined signed 64x64 bits multiplier using radix-32 modified Booth Algorithm has been increased because it require small total number of steps that decreases total delay of multiplication.

TABLE II COMPARISONS OF DIFFERENT MULTIPLIERS

Multiplier	**Delay in ns**	**Total count**	**Slices used**
64x64-bit multiplier using radix-4 and array structure	28.96	62	14271/69120 (20%)
High speed parallel 32x32-b Multiplier using radix-16 booth encoder[1]	7.55	-----	182/2352 (7%)
64x64-bit mult[3]	4.88	----	----
Signed 64x64 bit multiplier using radix-16 Booth Algorithm	1.8	22	3347/69120 (3%)
Proposed Signed 64x64-bit multiplier using radix-32 Booth Algorithm	1.4	17	3677/69120 (5.31%)

V. CONCLUSION

We have designed pipelined signed 64x64-bit multiplier using Wallace structure and radix-32 modified Booth Algorithm in this paper. Radix 32 decreases number of partial products and compressor like 3:2 and 4:2 decreases the level numbers in Wallace structure so Wallace tree using radix-32 modified Booth Algorithm , 3:2 and 4:2 compressor increases the speed of proposed multiplier. It gives small delay of 1.4 ns. It required 76% of less total number of Compressor, 70% less generated partial products as compared to conventional multiplier and 87% of numbers of levels of Wallace

REFERENCES

[1] Chen ping-hua and ZHAO Jaun, "*high-speed parallel 32x32-bit multiplier Using Radix-16 Booth Encoder*", 2009 IEEE proceeding of 2009 Third International Symposium on Intelligent Information Technology Application Workshop, IITAW 2009, 406-409.

[2] Chen ping-hua and ZHAO Jaun, XIE Guo-bo,LI Yi-jun, "*An improved 32-bit Carry-Look ahead Adder with Conditional Carry- selection*"[C].Proceeding of 2009 4th International conference on Computer Science & Education, ICCSE 2009:1911-1913.

[3] Weinan Ma, Shuguo Li, "*A New High Compression Compressor for Large Multiplier*", Institute of Microelectronics, Tsinghua University, Beijing 100084,P.R. China, 2008 IEEE.

[4] LIU Qiang, WANG Rongsheng, "*High-speed Parallel 32x32-b Multiplier Design Using Radix-16 Booth Encoder*". Computer Engineering [J], 2005, 31:200-202

[5] B.Parhami, "*Computer Arithmetic Algorithm and Hardware designs*" Oxford University Press, 2000.

[6] Ki-seon Cho, Jong-on Park,Jin-seok Hong, Goang seog Choi, '*54x54-bit Radix-4 Multiplier based on Modified Booth Algorithm*" proceeding GLSVLSI 03, April 28-29, Washington, DC, USA: 233-236

[7] Korean, "*Computer Arithmetic Algorithms*," 2nd Edition. Prentice Hall 2001.

[8] Lakshamanan, Masuri Othman and Mohamad Alaudin Mohd. Ali, Signal Processing Group, "*High Performance Multiplier using Wallace-Booth Algorithm*", 2002 IEEE proceeding ICSE2002 proc.2002, Penang Malaysia: 433-436.

[9] Dr. D.S.Dawoud, "*Modified Booth Algorithm for Higher Radix fixed- point multiplication*", 1997 IEEE.

[10] Ahmed Elhossini, Ali Rasid, and Mohammed K.Refai, "*A 16x16-bit Modified radix 16 Booth Encoder parallel Multiplier*", Al_azahar University Engineering Journal, AUEJ Vol.8,No.7, Jan.2005

