International Journal of Advanced Computer Technology (IJACT)
ISSN:2319-7900

HIGH PERFORMANCE PIPELINED SIGNED 64X64-BIT MULTIPLIER USING RADIX-32 MODIFIED BOOTH ALGORITHM AND WALLACE STRUCTURE

First A. K.DAYANAND, M.TECH 1; Second B.D.SUDHEER, B.TECH 2; Third C. M.SATEESH, B.TECH 3; Four D.V.NAGESWARA RAO,B.TECH 4; Five E.V.AVINASH,B.TECH 5; Six F.P.VAMSIKRISHNNA,B.TECH.

Abstract

This project mainly focuses on designing of 64-bit signed multiplier using radix-32 modified Booth algorithm and Wallace Structure. It is designed for fixed length 64x64 bit operands. 3:2 and 4:2 Compressors used in Wallace tree structure accumulate partial products very quickly. Using both compressors, No. of levels has been reduced that also causes enhancing the speed of multiplier. Because of the use of radix-32 modified booth algorithm, the number of partial products generated is very less here when compared to conventional multipliers. An efficient VHDL code has been written and successfully synthesized and simulated using Xilinx ISE 9.2i and ModelSim PE Student Edition 10.2c.Proposed signed 64x64 bit multiplier using radix -32.

Index terms:-Radix-32, boothencoder, 3:2 compressor, 4:2 compressor, Wallace Tree.

Introduction

Multiplication is one of the most important operations in digital signal processing and DSP systems. A variety of computer arithmetic techniques can be used to implement a digital multiplier. Most techniques involve computing a set of partial products, and then summing the partial products together.

In the proposed algorithm, factors of designing multiplier are generation of partial products and summation of partial products. To improve the speed performance of multiplication, number of partial products has been reduced by using radix-32 Booth Algorithm and for reducing the delay of summation of partial products Wallace Tree Structure has been used. This project presents Signed 64x64 bit multiplier. The decision of using Radix-32 rather than Radix-16 is that it generates less number of partial products and Wallace Tree Structure has been used rather than array structure because in Wallace Tree, number of levels required are less in comparison to radix-16.

Multiplier structures

The multiplier is one of the key hardware blocks in most of the digital and high performance systems such as digital signal processors and microprocessors. The following are the different structures of the multipliers that can be employed

1) Wallace structure
2) Array structure

A. Wallace multiplier

The Wallace tree basically multiplies two unsigned integers. The conventional Wallace tree multiplier architecture comprises of an AND array for computing the partial products, a carry save adder for adding the partial products so obtained and a carry propagate adder in the final stage of addition.

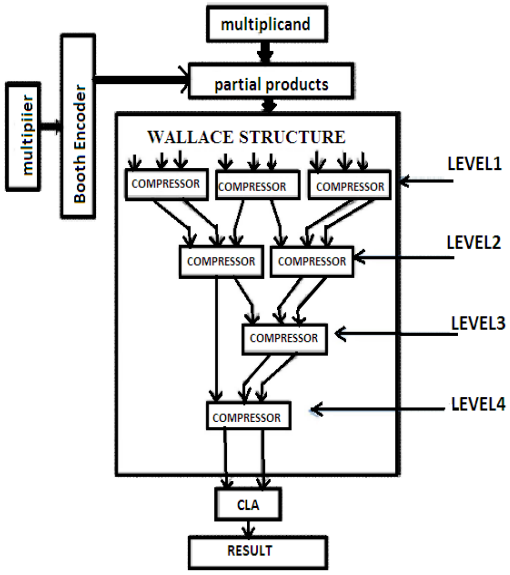

Figure 1: Structure of Wallace multiplier

B. Array multiplier

Array is a straightforward way to accumulate partial products using a number of compressors. The n-operand array consists of (n-2) compressors. Figure below shows an array structure for 6-operands, producing 2 outputs, where

compressor compresses the data having three multi-bit inputs to two multi-bit outputs.

As the number of operands increases, the number of compressors to be used and the number of levels in the array structure increases at a rate larger than that in a Wallace structure.

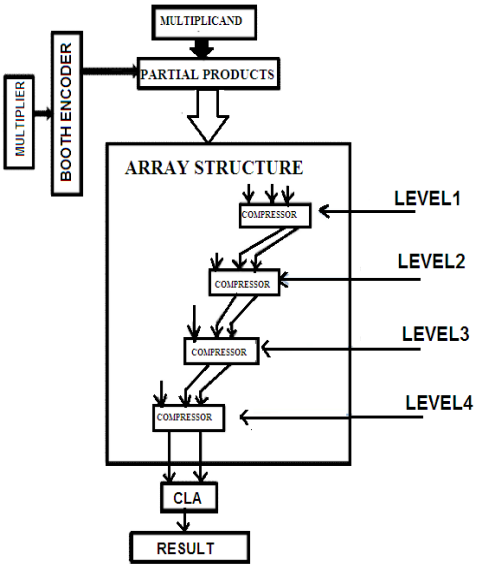

Figure 2: Structure of array multiplier

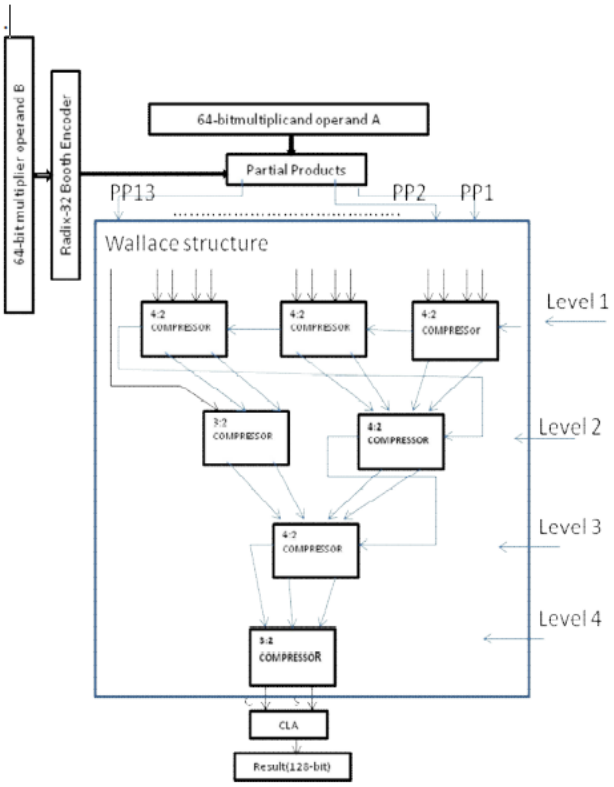

Figure 3: Architecture of proposed pipelined signed 64x64 bit Multiplier.

Design of high performance multiplier.

A Radix-32 Proposed Modified Booth Algorithm

To generate and reduce the number of partial products of multiplier, proposed modified Booth Algorithm has been used, In the proposed modified Booth Algorithm, multiplier has been divided in groups of 6 bits each and each group of 6 bits have been operational according to modified Booth Algorithm for generation of partial products 0, ±1A, ±2A, ±3A, ±4A, ±5A, ±6A, ±7A, ,±8A, ±9A, ±10A, ±11A,±12A, ±13A, ±14A, ,±15A, ±16A. These partial products are summed using compressors in structure of Wallace Tree.

In radix-32 Booth Algorithm, multiplier operand B is partitioned into groups having each group of 6 bits. In first group, first bit is taken zero and other bits are least significant five bit of multiplier operand. In second group, first bit is most significant bit of first group and other bits are next five bit of multiplier operand. In third group, first bit is most significant bit of second group and other bits are next five bit of multiplier operand. This process is carried on. For each group, Partial product is generated using multiplicand operand A. For n bit multiplier there is n/5 or[n/5 + 1] groups and partial products in proposed modified Booth Algorithm radix-32.

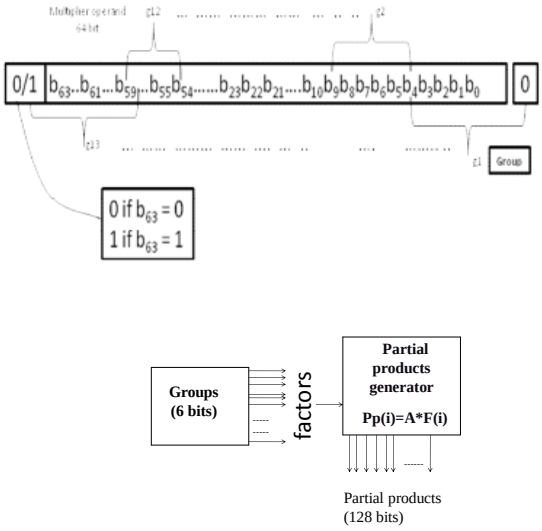

Figure 5: Generating groups and partial products

ISSN:2319-7900

Multiplier bits(g) bi+5,bi+4,bi+3, bi+2,bi+1,bi	Operation for group Fi	Multiplier bits bi+5,bi+4,bi+3, bi+2,bi+1,bi	Operation for Group Fi
000000	0	100000	-16A
000001	+1A	100001	-15A
000010	+1A	100010	-15A
000011	+2A	100011	-14A
000100	+2A	100100	-14A
000101	+3A	100101	-13A
000110	+3A	100110	-13A
000111	+4A	100111	-12A
001000	+4A	101000	-12A
001001	+5A	101001	-11A
001010	+5A	101010	-11A
001011	+6A	101011	-10A
001100	+6A	101100	-10A
001101	+7A	101101	-9A
001110	+7A	101110	-9A
001111	+8A	101111	-8A
010000	+8A	110000	-8A
010001	+9A	110001	-7A
010010	+9A	110010	-7A
010011	+10A	110011	-6A
010100	+10A	110100	-6A
010101	+11A	110101	-5A
010110	+11A	110110	-5A
010111	+12A	110111	-4A
011000	+12A	111000	-4A
011001	+13A	111001	-3A
011010	+13A	111010	-3A
011011	+14A	111011	-2A
011100	+14A	111100	-2A
011101	+15A	111101	-1A
011110	+15A	111110	-1A
011111	+16A	111111	0

B. Booth algorithum

Table for Proposed radix-32modified Booth algorithm has been designed. So it reduces the number of partial products in comparison to radix-16, improves the computational efficiency of multiplier, reduce the calculation delay. Computation of complex multiplier and re-encoding of multiplier can be executed in parallel.

Design process for Proposed modified multiplier

Process of proposed signed 64x64-bit operands multiplier has been described above. In the proposed multiplier, for increasing speed performance of multiplication, modified Booth Algorithm based on radix-32 has been used that produces less number of partial products. These partial products has been generated using 6 bits of multiplier operand that forms a group and these group match with the modified Booth Algorithm table and we find the factor according to these group of bits. After finding this factor, it has been multiplied with multiplicand operand to generate the corresponding partial product of the corresponding group .So in this way it generates less number of partial products usingradix-32 modified Booth Algorithm in comparison with radix-8, radix-16 etc. partials products have been generated for all the groups of multiplier operand. For further improving the speed performance of multiplication, Wallace tree structure has been used. Using 3:2 compressor and 4:2 compressor in Wallace tree structure, partial products have been summed in less delay. In Wallace tree structure summation delay is less because of using both 3:2 compressor and 4:2 compressor in summation operation ,there are less number of gates from initial bit to final bit rather than array structure. Computation of multiplier has been commutated through figures.

$$Fi = (k_1 + 2k_2 + 4k_3 + 8k_4 + 16k_5)$$

Where $k_1, k_2, k_3, k_4, k_5 = 0$ when 00 or 11

$k_1, k_2, k_3, k_4, k_5 = +1$ when 01

$k_1, k_2, k_3, k_4, k_5 = -1$ when 10

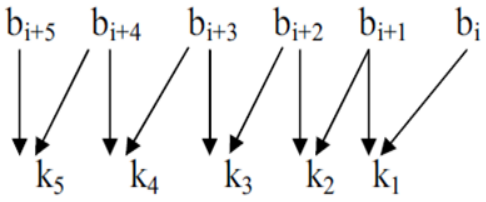

Figure 4: Finding value of k1, k2, k3, k4, k5

ISSN:2319-7900

Simulation and synthesis

For VHDL code of High performance pipelined signed 64x64 bit multiplier using radix-32, Xilinx ISE 9.2i tool, ModelSim PE Student Edition 10.0c. has been used for synthesizing and simulation. By using these tools, VHDLcode as been successfully synthesized and simulated..The result of simulation has been shown in figure 6.

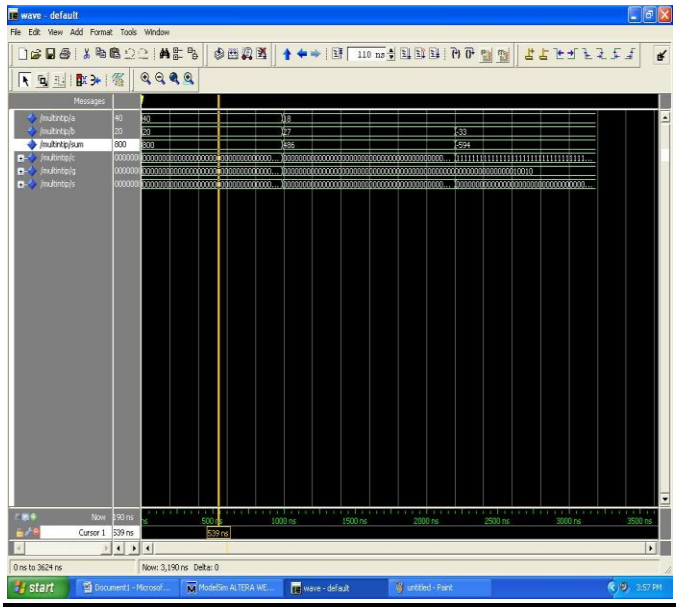

Figure 6: simulation result of our proposed multiplier when two input's are of same sign.

Result comparison

The proposed pipelined signed 64x64 bits using radix- 32 modified booth multiplier has been designed. Comparison of the proposed multiplier with the signed 64x64 bits multiplier using radix-16, Array structure multiplier and 32x32 b multiplier using radix-16 Booth Encoder is shown in table II. Performance of different multiplier is shown in figure9. It shows that proposed pipelined signed 64x64 bits multiplier using radix-32 modified Booth Algorithm and Wallace tree structure have 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 improved because it require less total number of steps that reduces total delay of multiplication.

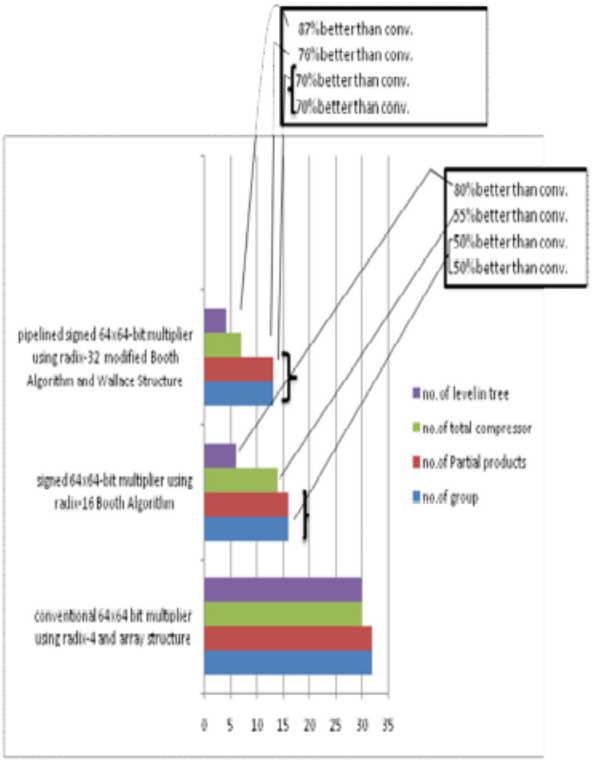

Figure7. Comparison of total number of components, levels, groups and partial products in multipliers

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%)

ISSN:2319-7900

Conclusion

In this project, we have designed a signed 64x64-bitmultiplier using radix-32 modified Booth Algorithm and Wallace structure. Wallace tree using 3:2 and 4:2 compressor, radix-32 modified Booth Algorithm improve the speed of the proposed multiplier because radix-32 reduces no. of partial products, both 3:2 and 4:2 compressor reduces no. of levels in Wallace structure. It provides a delay of 11.048 ns. we have obtained the synthesis reports in Xilinx using virtex-7 family. We have compared it with an array multiplier of same radix and it resulted in a delay of 12.207 ns and comparisons are listed in the table above.

References

[1] Chen ping-hue and ZHAO Juan, "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-hue and ZHAO Juan, 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, Shogun 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-bMultiplier 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.

Biographie

FIRST A. K.DAYANAND received the B.tech degree in Electrionics and communication Engineering from the University of jntu, kakinada, andhrapradesh, in 2008, the M.tech. degree in communication and engineering and singnal processing from the M.V.R.COLLEGE, Paritala, andhrapradesh, in 2012, currently he is an associate professor of electronics and communication engineering from the L.I.M.A.T,madhalavarigudem.email:mail4dayanand@gmail.com

SECOND B. D.SUDHEER BABU received the B.tech degree in Electronics and communication engineering form the gudlavalleru engineering college(junk), gudlavalleru, Andhra Pradesh in 2012,at present pursing M.tech degree in embedded systems from the L.I.M.A.T, madhalavarigudem. email:sudhir.devarapalli@gmail.com

Third c: MAHADEV SATHISH,at present pursing B.TECH degree in Electronics and communication engineering form the L.I.M.A.T, madhalavarigudem. email:sathishmahadev2014@gmail.com

Four d: V.NAGESWARA RAO at present pursing B.TECH degree in Electronics and communication engineering form the L.I.M.A.T, madhalavarigudem. email:naga.wills@gmail.com

Five e: V.AVINASH at present pursing B.TECH degree in Electronics and communication engineering form the L.I.M.A.T,madhalavarigudem.email:vammuavinash9@gmail.com

Six: f P.VAMSI KRISHNA
At present pursing B.TECH degree in Electronics and communication engineering form the L.I.M.A.T, madhalavarigudem. email:vamsikrishnap97@gmail.com