International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 1, Issue 4 (sept-oct 2013), PP. 86-90

DESIGN AND IMPLEMENTATION OF 32 BIT HIGH LEVEL WALLACE TREE MUTIPLIER

M.RAVINDRA KUMAR[1], G.PARAMESWARA RAO[2]
[1]Dept. of ECE S S C E, Chilakapalem Srikakulam, India.
[2]Assoc. Prof. Dept. of E C E S S C E, Chilakapalem Srikakulam, India.

Abstract --- Designing multipliers that are of speedy, low power, and standard in layout are of generous research interest. Wallace tree multiplier is one of the multiplier, which is used to accomplish high speed and low power multiplier and to condense the number of partial products generated in multiplication process. To trim down the time Wallace Tree adder structures have been used to sum the partial products .In general Wallace tree multipliers are existing in 8*8 bit multipliers and 16 bit multipliers. In this paper we put forward a 32 bit high level Wallace tree multiplier structure is investigated and assessed. A 32 bit Wallace high-speed multiplier uses full adders and half adders, 4:2 compressor, 3:2 compressor in their declining phase. The Wallace tree multiplier is consisting of 64 registers, 64 flip flops and 3 half adders as well as 960 full adders with logic time delay is 53.198 ns and off set time delay is 7.6 ns, having total memory usage is 191.132MB. The design is efficiently mapped to Hardware Resources in SPARTAN-3 FPGA. The design is implemented, simulated and synthesized by using Verilog.

Index terms---Multipliers, Wallace tree adder (WT), Compressor, Partial Products.

I. INTRODUCTION

In the mainstream of digital signal processing (DSP) applications the crucial operations habitually engage many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput Multiplication and Accumulators always a key to pull off a high performance digital signal processing system. In the last few years, the main contemplation of MAC design is to improve its speediness. For the reason that, battery energy accessible for these portable products confines the power consumption of the system. To accomplish high speed MAC, it is indispensable to intend high speed multipliers.

To attain high speed, the renowned Wallace high-speed multiplier utilizes Wallace tree adders to diminish an N-row bit product matrix to a corresponding two row matrix so as to summed by means of a carry propagating adder to provide the product. The Wallace tree adders are predictable full adders whose carries are not connected, accordingly that three words are taken in and two words are output. The Wallace multiplier also uses full adders and half adders, 4:2 compressor, 3:2 compressor in their reduction phase. This paper presents a high level wallace tree multiplier design that greatly reduces the number of half adders for a progressive speed.

II. WALLACE TREE MULTIPLIER

A high-speed procedure for multiplication of two numbers was developed by Wallace. It executes the accumulation operations in similar, consequential in less delay. In Wallace tree multiplier , a diverse way of comparable addition of the partial product bits by a tree of carry save adders, which is well-known as Wallace tree. A Wallace tree is a competent hardware execution of a digital circuit that multipliers two integers .In order to execute the multiplication of two numbers with the Wallace method, partial product matrix is compacted to a two-row matrix by using a carry save adder and the left over two rows are summed via a fast carry-propagate adder to form the product. These improvements turn into more prominent is now. The assistance of the Wallace tree is that there are only O(log(n)) decline layers, and each layer O(1) has transmission delay. As making the partial products is O (1) and the ultimate addition is O (log n) the multiplication is only O(log n) , not much slower than addition . Honestly adding partial products with standard adders would need O(log2 n) time. From a complexity theoretic perspective, the Wallace tree algorithm places multiplication .These multiplications only considered gate delays and don't deal with wire delays, which can also considerable. The beneath figure shows that the common block diagram of Wallace tree multiplier. It consists of Half adders and full adders.

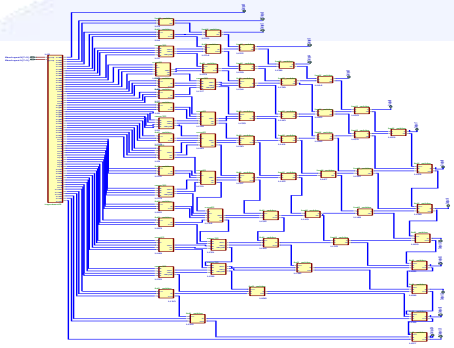

Fig 2.1 Block Diagram of Wallace tree multiplier.

A. Wallace tree Reduction Procedure

In Wallace tree proposal, all the bits of all of the partial products in each column are added simultaneously among a set of counters in parallel not including propagating any carries. An additional set of counters then reduce this new matrix and so on, awaiting a two-row matrix is generated.

Wallace tree multiplier uses three -steps procedure for the multiplication.

Step 1: Formation of bit products.

Step 2: The bit product matrix is reduced to a 2-row matrix by using a Wallace tree adder.

Step 3: The remaining two rows are summed using a fast carry – propagate adder to Produce the product.

The beneath figure demonstrate about the procedure of Wallace tree multiplier.

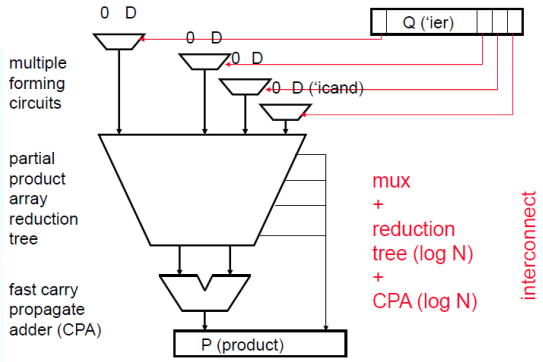

Fig.2.2: Wallace tree multiplication reduction procedure

B. WALLACE TREE ADDER

Wallace tree has been used in this project in order to speed up the multiplication by constricting the number of partial products. This design is done using half adders, full adders Wallace tree adders speed up the multiplication. Let us consider a 4 bit wallace tree adder as shown in the figure below there are four sign extension values generated namely sign 1E, 2E, 3E and 4E for the partial product PP1, PP2, PP3 and PP4 correspondingly. The Partial product line arrangement of total four partial products is shown in figure2.3.

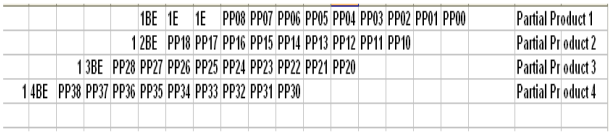

Fig 2.3: Partial Product Initial Arrangement

The second partial product had to be shifted left by two bits before adding to the first partial product. Hence the third will be shifted left by four where as for fourth it will be shifted left by fifth and so on. Hence after proper arrangement all the four partial products will be added along with the sign extension.

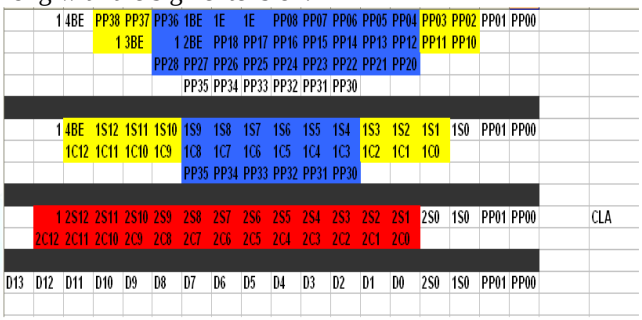

Fig 2.4: Wallace Tree Multiplication Method

First of all, the partial product initial arrangement is rearranged into first stage as shown in figure 2.4. It can be seen like a tree shape here. The stage from PP36 till 1 from the 4th partial product is stimulated to the first row and 3BE together with 1 is moved up to the row partial product 2. Behind rearrangement, the first three rows will be added using half adder and Wallace tree adder. The fourth partial product will not be added first but will be sent directly to the second stage. Hence, there total up to nine Wallace tree adders and four half adders. For second stage, the summation of the first half adders in right hand side of the first stage is examined. After the summation is done to add up PP02 and PP10, The SUM (1S0) will be generated in the same column as the second stage shows where as the CARRY (1C0) will be shift left into next level of summation. In this stage, the bit PP30-PP35 is ultimately being added using Wallace tree adder. At this stage, bit 4BE is also being added by using half adder. Hence, there are total six Wallace tree adders and seven half adders needed in this stage. In third stage, it is a final stage adder and since there are only remaining two inputs to be added instead of three, thus addition operation is used to perform the final summation based on the Sum and Cout signal in which had been propagated by the second stage. The bit PP00 and PP01 are directly sent to the output without going through any gate level. Hence, Wallace tree adder will have a 17 bit length output including the carry from the final bit. Like in this way the same process was performed by the wallace tree adder for 8 bit wallace tree multiplier and 16 bit wallace tree multiplier. And the same concept is used in high level 32 bit wallace tree multiplier also.

C. Half adder:

In the reduction process of wallace tree multiplier the half adders are used frequently. The half adder adds two single

binary bits A and B. It has two outputs, sum (*S*) and carry (*C*). The carry signal signifies an overflow into the next digit of a multi-bit addition. The value of the sum is 2C + S. The simplest half-adder design is shown in below figure. Integrate an XOR gate for Sum and an AND gate for Carry. The half-adder adds two input bits and generate carry and sum which are the two outputs of half adder as shown in below figure 2.5.

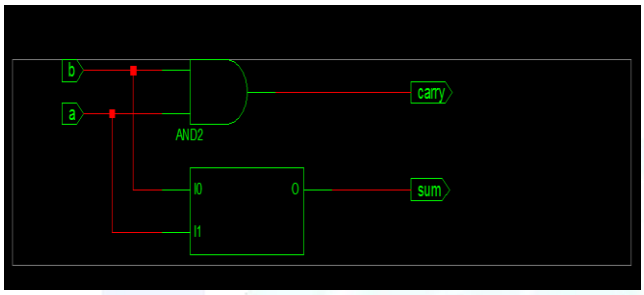

Fig 2.5: Half adder logic diagram

D. Full adder

In the process of wallace tree multiplier reduction full adders are also used frequently .A full adder adds binary numbers and accounts for values carried in as well as outas shown in below figure 2.6. A single bit full adder adds three one-bit numbers, renowned as A, B, and C_{in} ; A and B are the operands, and C_{in} is a bit carried in from the next less significant stage. The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit wide binary numbers. The circuit produces a two-bit output, output carry and sum typically represented by the signals C_{out} and *S*.

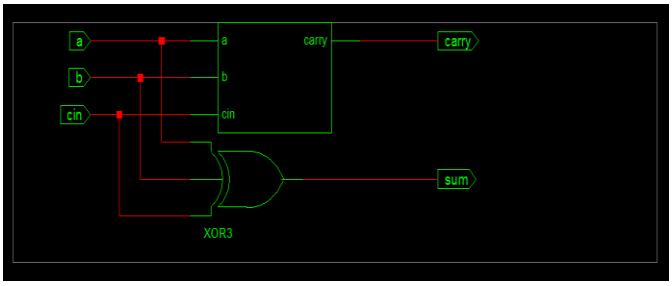

Fig 2.6: Full-adder logic diagram

III. DESIGN OF 32 BIT HIGH LEVEL WALLACCE TREE MULYTIPLIIER

The multiplication of two numbers X and Y each one having 32 bits and producing its result as Product of 64bits explains the method of addition of different intermediate terms. The different intermediate terms formed after the multiplication of two 32-bit numbers are. Two intermediate terms in one column are added using a half adder and more than two terms in one column are added using full adder as explained above in Wallace tree reduction procedure.. The sum obtained after each addition is denoted by where varies from 1 to 991. Similarly carries are denoted by where varies from 1to 991 and denotes next carries, where varies from 1 to 32. bit Wallace high level multiplier , the product output is shown. 3:2 compressors takes four inputs and generates two outputs with carry given to a next stage.

Full adders takes three inputs to generate two outputs which contains sum and carry while half adders takes two inputs and produces two outputs containing sum and carry. As half adder does not contribute much more in diminution phase, it is essential to decrease the number of half adders. The customized Wallace tree construction reduces the number of half adders therefore difficulty reduces. 32*32 bit multiplication presents 1024 partial products which are reduced using Wallace tree by using above components.

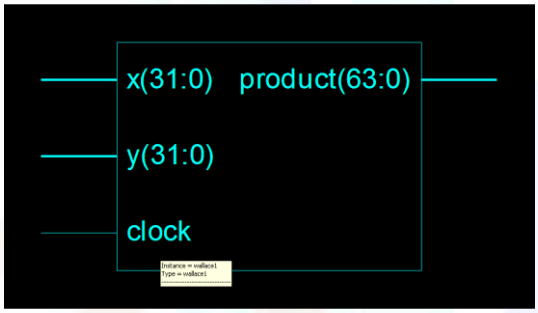

Fig. 3.1 32 × 32 bit Wallace tree multiplier

A. RTL SCHEMATIC VIEW OF HIGH LEVEL WALLACE TREE MULTIPLIER

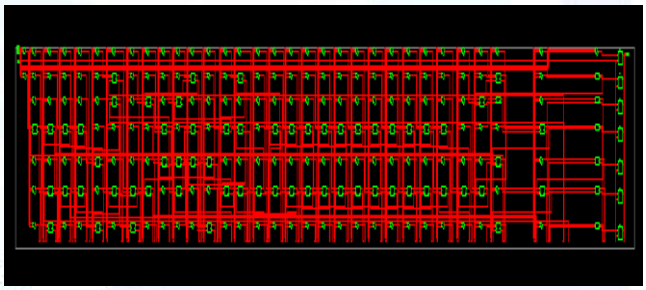

Fig: Full RTL Schematic View of 32 bit wallace tree multiplier

B. LOOK UP TABLES

A look-up table is an array that replaces runtime computation with a simpler array indexing operation. The savings in terms of processing time can be significant, since retrieving a value from memory is often faster than undergoing an expensive computation or input/output operation. The tables may be precalculated and stored in static program storage.The

below figure shows that the look up tables which are existing in high level wallace tree multiplier.

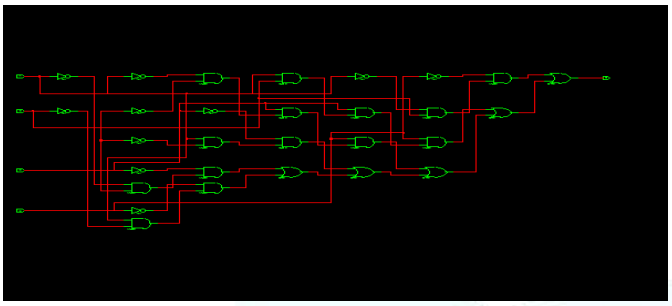

Fig 3.3: LUT in the wallace tree multiplier

IV. IMPLEMENTATION OF 32 BIT WALLACE TREE MULTIPLIER IN VERILOG HDL

Verilog Hardware Description Language is a enormous low level language. Structural models are trouble-free to design and Behavioural RTL code is appealing good. The syntax is standard and simple to remember. It is the best HDL language to learn and use. Nevertheless Verilog requires user defined data types and requires the interface-object separation of the VHDL's entity-architecture model. This paper is implemented in Verilog HDL in behavioural RTL code.

V. SYNTHESIZED RESULTS

The 32 bit high level Wallace tree multiplier is implemented on SPARTAN 3 FPGA device and it is synthesized on Xilinx 8.2i Version by using Verilog hard ware description language code. The Synthesized report and Device summary Utilization report as shown below. The Wallace tree multiplier is consisting of 64 registers, 64 flip flops and 3 half adders as well as 960 full adders with logic time delay is 53.198 ns and off set time delay is 7.6 ns, having total memory usage is 191.132MB.

```
--------------------------------

=====================================================================
*                        Final Report                               *
=====================================================================
Final Results
RTL Top Level Output File Name     : wallace1.ngr
Top Level Output File Name         : wallace1
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : NO

Design Statistics
# IOs                              : 129

Cell Usage :
# BELS                             : 2061
#      GND                         : 1
#      LUT2                        : 98
#      LUT3                        : 97
#      LUT4                        : 1839
#      MUXF5                       : 26
# FlipFlops/Latches                : 64
#      FD                          : 64
# Clock Buffers                    : 1
#      BUFGP                       : 1
# IO Buffers                       : 128
#      IBUF                        : 64
#      OBUF                        : 64
=====================================================================

Device utilization summary:
---------------------------

Selected Device : 3s200ft256-4

 Number of Slices:                     1167  out of   1920    60%
 Number of 4 input LUTs:               2034  out of   3840    52%
 Number of IOs:                         129
 Number of bonded IOBs:                 129  out of    173    74%
    IOB Flip Flops:                      64
 Number of GCLKs:                        1  out of      8    12%


=====================================================================
```

Fig 5.1: Synthesized report of Wallace tree multiplier

A.DEVICE UTILISATION SUMMARY

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	1167	1920	60%
Number of 4 input LUTs	2034	3840	52%
Number of bonded IOBs	129	173	74%
Number of GCLKs	1	8	12%

VI. COMPARISIONS

TABLE I. Differentiating the Results

NAME OF THE MULTIPLIER	POWER VALUES
ARRAY MULTIPLIER	140mw
BOOTH MULTIPLIER	125mw
PROPOSED WALLACE TREE MULTIPLIER	94mw

If we evaluates the above values along with each other we can examine that the Array Multiplier is the worst case multiplier overwhelming highest amount of power. Then comes the Radix – 2 booth multiplier which consumes lesser power than array multiplier. The Wallace Tree multiplier consuming lesser power than the other. Hence we reach to a conclusion that the Wallace tree multiplier is one of the best for situations requiring Low power Applications.

VII. CONCLUSION

The 32 bit high level Wallace tree multipliers can be resolved & analyzed using method of Wallace tree construction. The high level trees has a somewhat less significant critical path and some extent larger wiring overhead but it entails low power and high speed. There is no requirement of logic registers, registers, pins memory bits and PLLs. This high level wallace tree multiplier, which consist of full adders and reduced no. of half adder and reduces the complexity and save the power.

ACKNOWLEDGMENT

The author would like to thank MR.G. PARAMESWARA RAO, Assoc. Professor, Electronics & Communication Engineering (Speclz‟n in VLSI D e s i g n),Sri Sivani College Of Engineering, JNTUK, for his continuous support and encouragement for this work. The authors would also like to acknowledge the support provided by the technical staff of Elect. & Comm. Engg., Sri Sivani College Of Engineering, JNTUK providing him with the ample amenities, ways & means through which he was capable to inclusive this task.

REFERENCES

[1] S. Ravi Chandra Kishore , K.V. Ramana Rao " Implementation of carry-save adders in FPGA"IJEAT ISSN: 2249 – 8958, Volume-1, Issue-6, August 2012.

[2] MS. V.N. CHAUDHARY,PROF. DR. P.R. DESHMUKH "ANALYSIS AND IMPLEMENTATION OF LOW POWER WALLACE TREE MULTIPLIER" JIKRECE NOV 10 TO OCT 11 | VOLUME – 01 ,ISSUE-02.

[3] Simran Kaur and Mr. Mansul bansar " FPGA IMPLEMENTATION OF EFFICIENT MODIFIED BOOTH WALLACE MULTIPLIER "Thaipur University,Pune,June-2011.

[4] J.-L. Beuchat and J.-M. Muller, "Automatic generation of modular mul-tipliers for fpga applications," IEEE Transactions on Computers, vol. 57, no. 12, pp. 1600–1613, December,2008.

[5] J. Detrey, F. de Dinechin, and X. Pujol, "Return of the hardware floating-point elementary function," in Proceedings of the 18th IEEE Symposium on Computer Arithmetic (Montpellier, France), Kornerup and Muller, Eds. Los Alamitos, CA: IEEE Computer Society Press, June 2007, pp. 161–168.

[6] H. Eberle, G. N., S. Shantz, V. G upta, L. Rarick, and S. Sundaram, "A public-key cryptographic processor for RSA and ECC," in Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors (ASAP2004), September 2004.

[7] H. R. Ismail, R.C., "High performance complex number multiplier using booth-wallace algorithm," in IEEE International Conference on Semiconductor Electronics ICSE, November 2006.

[8] K. Manochehri and S. Pourmozafari, "Modified radix-2 montgomery modular multiplication to make it faster and simpler," in IEEE International Conference on Information Technology: Coding and Computing, ITCC 2005April.