Design and Implementation of Multiply Accumulate Unit for Large Arithmetic Unit Operations

M. Saipriya Sharanya[1] Venu Adepu[2]
[1]M.Tech. Student [2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Jyothishmathi Institute of Technological Sciences, Ramakrishna Colony, Nustulapur, Karimnagar, Telangana

Abstract— The proposed paper consisting the design and implementation of the 64 bit MAC (Multiply Accumulate Unit) for large number of arithmetic operations. The proposed design has multiplier and accumulator units, multiplier block perform normal multiplication operation. The proposed multiplication can be designed by using Wallace tree algorithm. And accumulator it performs two operations; one is storing and another one addition of products. The proposed MAC technique can be used in several ALUs and many types of Digital Signal Processing apllications. The proposed design was implemented with Verilog HDL and simulated by XILINX ISE 14.5 synthesis tool.

Key words: Digital Signal Processing, ALU, Wallace Tree, carry save adder, Accumulator

I. INTRODUCTION

MAC unit is associate inevitable part in several digital signal process (DSP) applications involving multiplications and/or accumulations. MAC unit is employed for prime performance digital signal processing systems. The DSP applications embrace filtering, convolution, and inner product. Most of digital signal process strategies use nonlinear functions like separate trigonometric function remodel (DCT) or discrete rippling transforms (DWT). As a result of they're basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition arithmetic determines the execution speed and performance of the whole calculation. Multiplication-and-accumulate operations are typical for digital filters.

Therefore, the functionality of the macintosh unit permits high-speed filtering and different process typical for DSP applications. Since the macintosh unit operates completely freelance of the central processor, it will method data on an individual basis and their by scale back central processor load. The application like optical communication systems which is predicated on DSP, need very quick processing of giant quantity of digital information.

The Fast Fourier remodel (FFT) conjointly needs addition and multiplication. sixty four bit will handle larger bits and have more memory. A macintosh unit consists of a multiplier factor associated an accumulator containing the total of the previous successive product. The macintosh inputs are obtained from the memory location and given to the multiplier factor block. the look consists of sixty four bit changed Wallace multiplier, 128 bit carry save adder and a register.

II. ARITHMETIC OPERATIONS

Arithmetic Logical unit performs two types of operations those are arithmetic operations and logical operations. Arithmetic operations such as addition, multiplication, substracton, accumation, Mac. This paper represents Mac design by using high performance adders. In this paper third session represents about the Accumulator operation, fourth session about the high performance adders; in fifth session Wallace tree algorithm in sixth session MAC operations are obtained.

III. ACCUMULATOR

The Multiplier-Accumulator (MAC) operation is signal processor. The input that is fed from the memory location of sixty four bit once the input is given to the multiplier fa tor it starts computing worth for the given sixty four bit input and thus the output are going to be 128 bits. The multiplier factor output is given because the input to hold saves ad-der that performs addition.

IV. HIGH PERFORMANCE ADDER

The binary adder is that the crucial part in most digital circuit styles together with digital signal processors (DSP) and microprocessor knowledge path units. As such, in depth analysis continues to be targeted on up the ability delay performance of the adder.

In VLSI implementations, parallelprefix adders square measure best-known to possess the simplest performance. Parallel-prefix adders (also referred to as carry-tree adders) square measure known to possess the simplest performance in VLSI styles as shown in Fig.3.

However, this performance advantage will not translate directly into FPGA implementations attributable to constraints on logic block configurations and routing overhead. This paper investigates 3 kinds of carry-tree adders (the Kogge-Stone, distributed Kogge-Stone, and spanning tree adder). It consist some stages these are following in below.

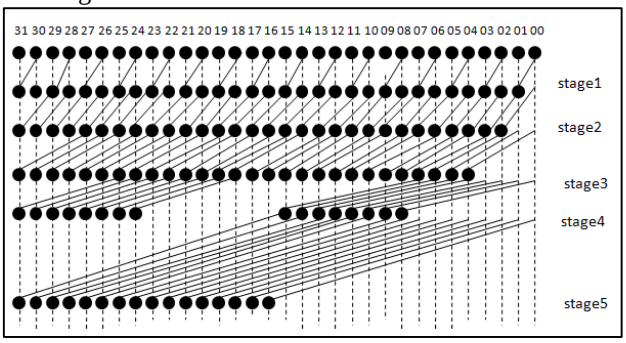


Fig. 1: High performance adder

A. Pre-Processing Stage

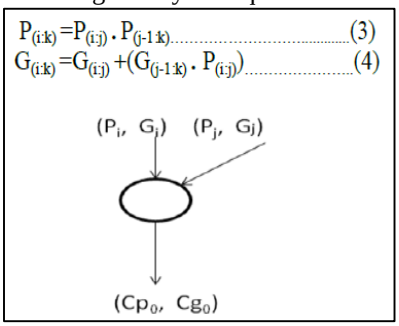
In this stage we tend to cypher, the generate and propagate signals square measure accustomed generate carry input of every adder. A and B square measure inputs. These signals square measure given by the equation 1&2.

$$P_i=A_i \oplus B_i \quad \text{.....(1)}$$
$$G_I=A_i \,.B_i \quad \text{.....(2)}$$

B. Carry Generation Network

In this stage we tend to cypher carries similar to every bit. Execution is finished in parallel kind [4].After the computation of carries in parallel they're divided into smaller items. Carry operator contain 2 AND gates, one gate. It uses propagate and generate as intermediate signals that square measure given by the equations 3&4.

$$P_{(i:k)}=P_{(i:j)} \,.\, P_{(j-1:k)} \quad \text{.....(3)}$$
$$G_{(i:k)}=G_{(i:j)} +(G_{(j-1:k)} \,.\, P_{(i:j)}) \quad \text{.....(4)}$$



C. Post Processing Stage

The operations concerned during this figure square measure given as. Post process stage this is often the ultimate stage to cypher the summation of input bits. it's same for all adders and add bit equation given

$$S_i= P_i \oplus C_i \quad \text{.....(5)}$$
$$C_{i+1}=(P_i \bullet C0) + G_i \quad \text{.....(6)}$$

V. WALLACE TREE MULTIPLIER

A changed Wall ace number is AN economical hardware implementation of digital circuit multiplying 2 integers. usually in typical Wallace multipliers several full adders and half adders are employed in their reduction section. half adders don't reduce the quantity of partial product bits. Therefore, minirnizing the quantity of half] adders employed in a multiplier reduction can cut back the quality. Hence, a modification to the Wallace reduction is done in that the delay is that the same as for the conventional Wallace reduction. The changed reduction methodology greatly reduces the quantity of half adders with a really slight increase within the range of full adders.

Processing Re-write Suggestions Done (Unique Article) Reduced complexness Wall ace number reduction consists of 3 stages [2]. initial stage the N x N product matrix is created and before the passing on to the second section the merchandise matrix is rearranged to take the form of inverted pyramid. During the second section the rearranged product matrix is grouped into non-overlapping cluster of 3 as shown within the figure a pair of, single bit and 2 bits within the group are going to be passed on to future stage and 3 bits area unit given to a full adder. the quantity of rows in the in every stage of the reduction section is calculated by the formula

$$r_{i+1}= 2[r_i/3]+r_i \bmod 3 \quad (2)$$
$$\text{If } r_i \bmod 3 = 0, \text{ then } r_{i+1} = 2r_i/3 \quad (3)$$

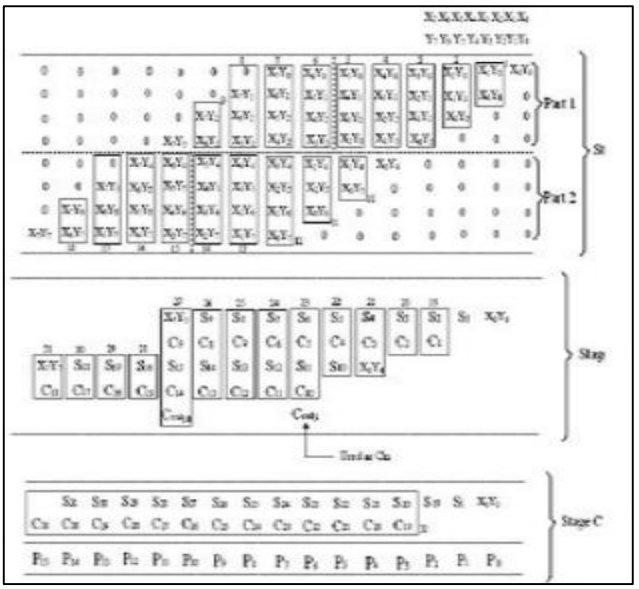


Fig. 2: Wallace tree Algorithm

If the worth calculated from the on top of equation for number of rows in every stage within the second part and the number of row that square measure fonned in every stage of the second part doesn't match, solely then the half adder are used. the ultimate product of the second stage are within the height of 2 bits and passed on to the third stage. throughout the third stage the output of the second stage is given to the carry propagation adder to get the ultimate output.

IThus sixty four bit changed Wallace multiplier factor is constructed and therefore the total variety of stages within the second part is ten. As per the equation the amount of row in every of the ten stages was calculated and therefore the use of 0.5 adders was restricted solely to the tenth stage. the whole variety of 0.5 adders utilized in the second part is eight and therefore the total variety of full adders that was used throughout the second part is slightly increased that within the typical Wallace multiplier factor.Since the sixty four bit changed Wallace multiplier factor is difficult to represent, a typical lO-bit by 10-bit reduction shown in figure a pair of for understanding. The modified Wallace tree shows higher performance when carry save adder is employed in finish rather than ripple carry adder. The carry save adder that is used is taken into account to be the important half within the multiplier as a result of it's accountable for the most important amount of computation.

VI. MULTIPLY ACCUMULATE UNIT

The Multiplier-Accumulator (MAC) operation is the key operation not solely in DSP applications however also in multimedia system IP and various different applications. As mentioned on top of, MAC unit encompass multiplier factor, adder and register/accumulator. During this paper, we tend to used sixty four bit modified Wallace multiplier factor. The mackintosh inputs square measure obtained from the memory location and given to the multiplier block. This can be helpful in sixty four bit digital signal processor. The input that is being fed from the memory location is sixty four bit. Once the input is given to the multiplier factor it starts computing price for the given sixty four bit input and thus the output are going to be 128 bits. The multiplier factor

output is given because the input to carry save adder that performs addition.

The output of carry save adder is 129 bit i.e. one bit is for the carry (128bits+ one bit). Then, the output is given to the register. The accumulator register utilized in this style is Parallel In Parallel Out (PIPO). Since the bits square measure immense and conjointly carry save adder produces all the output values in parallel, PIPO register is employed wherever the input bits square measure taken in parallel and output is taken in parallel. The output of the register is taken out or fed back as one of the input to the carry save adder. The figure one shows the fundamental design of mackintosh unit.

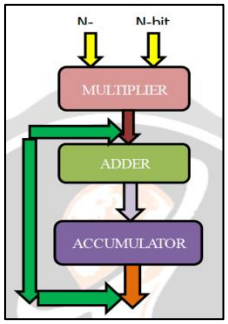


Fig. 3: MAC architecture

VII. IMPLEMENTATION RESULTS

The proposed MAC design was designed by using Verilog HDL then synthesis and simulation done by using XILINX ISE 14.5 v software. The proposed system RTL schematic and synthesis and simulation results are shown in the below figures.

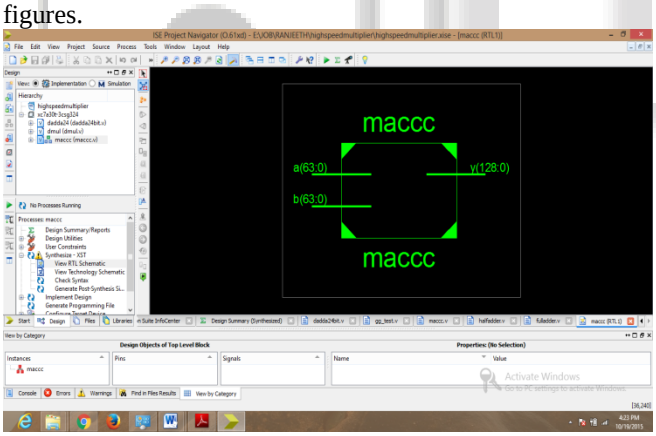


Fig. 4: Top Level RTL schematic diagram

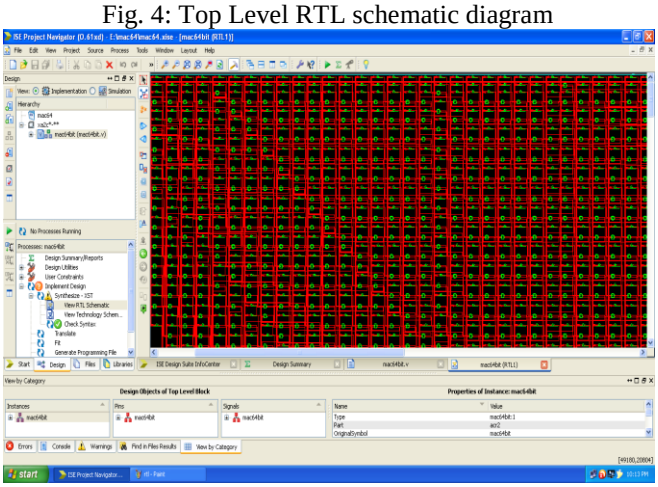


Fig. 5: Internal RTL Schematic Diagram

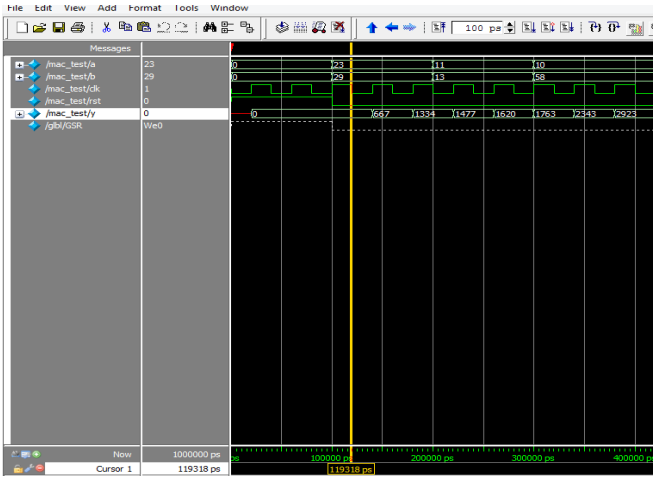


Fig. 6: Output simulation result

A. *Synthesis Results*

mac Project Status (04/13/2016 - 11:45:03)			
Project File:	batch12.xise	**Parser Errors:**	No Errors
Module Name:	mac	**Implementation State:**	Synthesized
Target Device:	xc7a30t-3csg324	**• Errors:**	No Errors
Product Version:	ISE 13.2	**• Warnings:**	1 Warning (1 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	129	42000	0%
Number of Slice LUTs	6453	21000	30%
Number of fully used LUT-FF pairs	108	6474	1%
Number of bonded IOBs	259	210	123%
Number of BUFG/BUFGCTRLs	1	32	3%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Wed Apr 13 11:45:03 2016	0	1 Warning (1 new)	65 Infos (65 new)

Fig. 7: LUT or AREA result

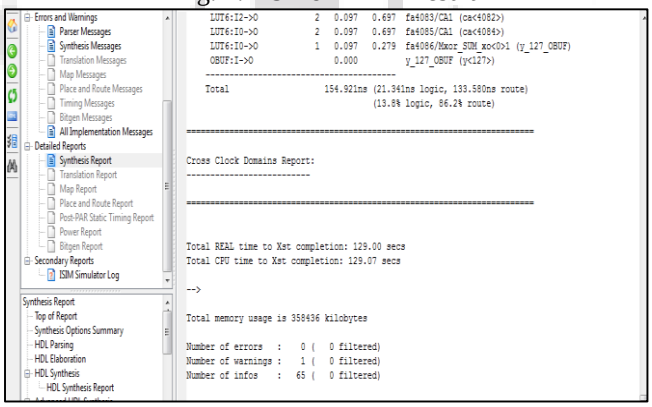


Fig. 8: Time Delay result

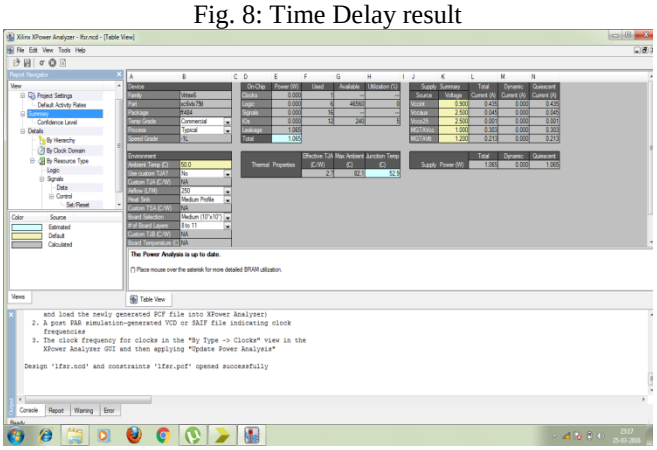


Fig. 9: Screenshot

VIII. CONCLUSION

In this paper I have worked on the design of MAC(Multiply Accumulate unit) by using wallace tree algorithm. To design this proposed worl I have used the wallace tree algorithm and high performance adders. The total design was implemented with the XILINX ISE 14.5 synthesis tool and Verilog HDL and I have checked the RTL achematic and synthesis results(Area, Delay,Power) and simulation in this project.

REFERNCES

[1] Young-Ho Seo and Dong-Wook Kim, "New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm," IEEE Transactions on very large scale integration (vlsi) systems, vol. 18, no. 2,february 20 10
[2] Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wall ace Multiplier Reduction, " IEEE Transactions On Computers, vol. 59, no. 8, Aug 20 10
[3] C. S. Wallace, "A suggestion for a fast multiplier," iEEE Trans. ElectronComput., vol. EC-13, no. I, pp. 14-17, Feb. 1964
[4] Shanthala S, Cyril Prasanna Raj, Dr.S.Y.Kulkarni, "Design and VLST Implementation of Pipelined Multiply Accumulate Unit," IEEE International Conference on Emerging Trends in Engineering and Technology, ICETET-09
[5] B.Ramkumar, Harish M Kittur and P.Mahesh Kannan, "ASIC Implementation of Modified Faster Carry Save Adder ", European Journal of Scientific Research, Vol. 42, Issue 1, 2010.
[6] R.UMA, Vidya Vijayan, M. Mohanapriya and Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICSj Vo1.3, No.1, February 2012
[7] V. G. Oklobdzija, "High-Speed VLSI Arithmetic Units: Adders and Multipliers", in "Design of High-Performance Microprocessor Circuits", Book edited by A.Chandrakasan,IEEE Press,2000
[8] Dadda, "Some Schemes for Parallel Multipliers," Alta Frequenza, vol. 34, pp. 349-356, 1965
[9] C.S. Wall ace "A Suggestion for a fast multipliers," IEEE Trans. Electronic Computers, vol. 13, no.l,pp 14-17, Feb. 1967 [10]. L.Dadda, "On Parallel Digital Multiplier", Alta Frequenza, vol. 45, pp. 574-580, 1976.

