Novel Architecture of High Speed Parallel MAC using Carry Select Adder

Deepika Setia
Post graduate (M.Tech)
UIET, Panjab University, Chandigarh

Charu Madhu
Assistant Professor
UIET, Panjab University, Chandigarh

ABSTRACT
In this paper, new hardware architecture of multiplier and accumulator (MAC) for high speed arithmetic was designed. The performance was improved by merging multiplication with accumulation and organize a hybrid type carry save adder (CSA).The proposed CSA tree uses 1's complement based radix-4 and radix-8 modified booth algorithm(MBA). The CSA propagates the carries to the least significant bits of the partial products and generates the least significant bits in advance to reduce the number of the input bits of the final adder. This MAC add the intermediate results in the form of sum and carry bits instead of the final adder output, which made it possible to optimize the pipeline system to improve the performance. The final addition was carried out by high speed carry select adder (CSLA) with binary to excess convertor using CLA. Based on the theoretical and experimental estimation, we analyzed the results in terms of delay. The design is implemented using VHDL language and simulated using Xilinx ISE10.1 Simulator.

Keywords
Modified Booth algorithm (MBA), Multiplier and Accumulate (MAC),Carry Look Ahead Adder(CLA),Carry Select adder(CSLA)

1. INTRODUCTION
With the rapid advances in multimedia and communication systems like signal processing, image processing, high capacity data processing is in great demand. Since most DSP functions like filtering, convolution etc. are accomplished by repetitive applications of multiplication and addition arithmetic, so a high speed MAC[1] is essential to improve the performance of a signal processing system. Thus the execution time of DSP function depends largely on the speed of multiplication and addition arithmetic i.e. MAC unit. So a high speed MAC is crucial for enhancing the performance of a signal processing system. Parallel implementation of MAC is widely used due to higher efficiency. In parallel MAC implementation, accumulator stage that provides the largest critical path delay in MAC is merged with multiplication stage to enhance speed and decrease the hardware architecture. The performance of MAC is mainly governed by two factors: i) Efficiency of MBA [2], which generates the partial product matrix. ii) Area and speed efficiency of final Adder which combines the results of accumulator and multiplication. MAC uses booth encoder[3] followed by Wallace tree[4] instead of using array of FA[5],[6].As in the Wallace tree and compressors ,partial product addition carried out as parallel as possible and operational time is approximate $O(\log_2 N)$ where N is the number of inputs. Many hardware implementations of MAC have been proposed which employ different methodologies for partial product reduction. In order to improve the efficiency of MAC, MBA algorithm has been used as the number of partial product rows reduces as per radix.

Many parallel multiplication architecture have been researched [7],[8]. One of the advanced types of MAC [19] architecture for digital signal processing has been proposed by Elguibaly [12]. It is an architecture where accumulation has been combined with the Carry Save Adder (CSA) tree that compresses the partial products and provides fast possible implementation. The critical path delay is removed by eliminating the separate adder block and decreasing the number of input bit in the final adder. In order to avoid irregular wiring, CSA favored to connect the neighbor interconnect structure. In recent, Bayoumi [18] proposed a high speed and area-efficient merged MAC architecture is based on binary trees constructed using a modified 4:2 compressor circuits. And *Seo, Kim* [22] proposed a high speed MAC with radix-2 that uses CSA tree architecture for reduction. The primary objective of this paper is to present a new type of Parallel MAC using modified Carry Select Adder, to reduce the implementation to practice and to demonstrate through simulation and design that this algorithm is competitive with other more commonly used algorithms when used for high performance implementations.

This paper is organized as follows. In Section 2, introduction of a conventional MAC will be given. In section 3, a high speed parallel MAC using radix-4 and radix-8 will be discussed. Results and implementation are detailed in section 4.Finally Section 5 presents a conclusion to this paper followed by references.

2. INTRODUCTION OF MAC

Firstly the basic operation of MAC is introduced. A multiplier unit can be split into three basic steps. For high speed multiplication, Modified Booth Algorithm (MBA) [3] is most commonly used, in which partial product is generated from Multiplicand (X) and Multiplier (Y). Although the partial product rows are reduced by using higher radix (4, 8, 16, 32) Booth Encoder but it increases complexity and improves the performance. The second step consists of partial product reduction process which may be carried out by CSA. The last step is the final addition of sum and carries which becomes the final multiplication result. If the process of accumulation is introduced then it consists of four steps.

The hardware architecture of conventional MAC is shown in Figure 1.It depicts the inputs X and Y for multiplication and Z as the previous accumulated value.

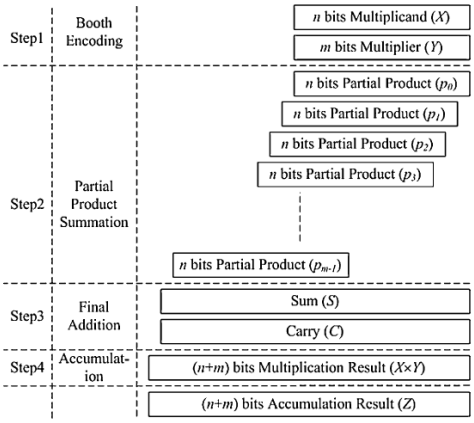

Figure 1: Operation Arithmetic of conventional MAC [22]

Mathematically, MAC can be expressed as:

$$P = X \times Y + Z = \sum_{i=0}^{\frac{N}{2}-1} d_i 2^{2i} Y + \sum_{i=0}^{2N-1} z_i 2^i \quad \text{for Radix-4} \quad (1)$$

$$P = X \times Y + Z = \sum_{i=0}^{\left(\left(N+\frac{1}{3}\right)-2\right)} d_i 2^{3i} Y + \sum_{i=0}^{2N-1} z_i 2^i \quad \text{for Radix-8} \quad (2)$$

Where $d_i = -2x_{2i+1} + x_{2i} + x_{2i-1}$

The two terms on the right hand side of (1) and (2) can be calculated separately and the final output is the addition of two terms.

3. HIGH SPEED MAC USING RADIX-4 AND RADIX-8 MBA

This section basically demonstrates the complete experimental methodology employed to design the MAC architecture. Implementation details of the techniques used for designing are illustrated. The expression for new arithmetic is derived from the standard equation. VLSI architecture for the new MAC design is implemented. In addition, a hybrid-type of CSA Tree architecture that can fulfill the operation of parallel MAC is formed.

3.1 Basic Concept

The basic approach is to design high speed performance MAC. The first stage performs the multiplication in order to generate the partial products. By using Modified Booth algorithm technique, the partial products formed will reduce to half in case of Radix-4 and thrice in case of Radix-8. The critical path can be reduced by adding the partial product terms and accumulate value together, in short forming a merged architecture of MAC. The delay of last accumulator stage must be further reduced in order to enhance the performance of MAC by applying the pipeline scheme to the standard design.

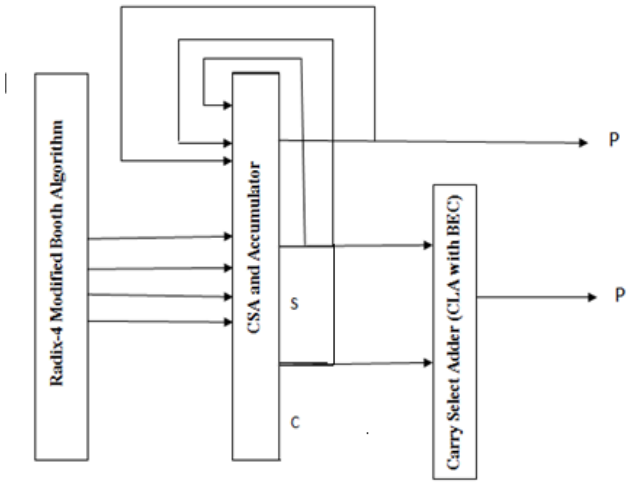

Figure 2: Hardware Architecture of MAC

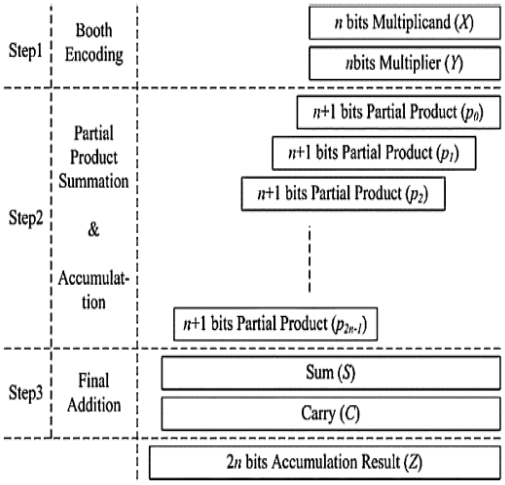

Figure 3: Operation arithmetic of Merged MAC [22]

The basic approach to improve the performance of the final adder is to decrease the number of input bits. In order to shorten this number of input bits, the multiple partial product rows formed by MBA are compresses into a carry and sum using CSA tree [15]. The bits to be transferred to the final adder is reduced by adding the lower bits of sum and carry in advance at each row within the well defined range such that overall performance due to large bit adder will not degraded. 2-bit CLA is used to add the lower bits of sum and carry in the CSA[16] tree in case of radix-4 MBA.3-bit and 2-bit CLA are used to add the lower bits of sum and carry in the CSA tree in case of radix-8 MBA [23] which also adds the sign bit .This cause the partial product matrix reduce to N½

instead of N(½ +1) rows in case of radix-4 MBA and N(1/3) instead of N(1/3+1) in case of radix-8 MBA [23].
In addition, output rate when pipelining is applied, the sums and carries from the CSA tree are added instead of the outputs from the final adder in the manner that the sum and carry from the CSA in the previous cycle are inputted to CSA. Due to the feedback of both sum and carry, the number of inputs to CSA increments compared to the standard design and [12]. The output from sum and carry are accumulated to determine the final adder result that must be very fast adder to avoid critical path delay of MAC [18]. Carry Select Adder (CSLA) is one of the fastest adders used in many data/audio-processing processors to perform fast arithmetic functions. This design employs a simple and efficient gate –level adjustment to reduce the area and power significantly of the CSLA. This adder is commonly used in many arithmetic and computational system to alter the problem of carry propagation delay by independently getting multiple carries and then select a exact carry to determine the sum [15].In the meanwhile, CSLA uses multiple pairs of independent adders to generate partial sum and carry at each stage by considering carry input Cin='0' and '1', then the final sum and carry are chosen by the multiplexers.
The basic concept of this design is to implement Binary to Excess-1 Convertor (BEC) instead of pairs of RCA with Cin ='1' in the regular CSLA to accomplish lower area and power consumption. The main feature of this BEC logic comes from the lesser number of logic gates than the N-bit Full Adder (FA) structure. The area and speed efficiency can be improved by using CLA at each stage instead of RCA.

3.2 Equation Derivation

The above mentioned concept is applied to basic equation of design to express the proposed MAC architecture. Then, the multiplication would be transferred to a hardware architecture that complies with the proposed concept, where the feedback value for accumulation will be modified and expanded for the new MAC. First, if the multiplication for Radix-8 is disintegrate and rearranged, it becomes

$$XxY=s_0 2Y+s_1 2^3 Y+s_2 2^6 Y+\ldots+s_{(N+1/3)-1} 2^{N-2} Y \quad (3)$$

This equation (3) can be further divided into first partial product, sum of middle terms and final partial product terms. This separation of terms help to feedback the input to accumulator in terms of sum, carry and pre-added results of the sum and carry from lower bits.

$$XxY=s_0 2Y+\sum_{i=1}^{\left(N+\frac{1}{3}\right)-2} s_i 2^{2i} Y+s_{(N+1/3)-1} 2^{N-2} Y \quad (4)$$

This concept of separation of equation (4) is also applied to accumulated value Z.Z is divided into upper and lower bits. As the lower values of Z are calculated in advance by 2- bit CLA including the sign bits.

$$Z=\sum_{i=0}^{N-1} z(i)2^i+\sum_{i=N}^{2N-1} z(i)\,2^i \quad (5)$$

The first term of above equation(5) on the right hand side corresponds to the lower bits that is fed back as sum and carry ;and second term on the left hand side corresponds to the upper bits that is fed back as additional output of sum and carry.

$$\sum_{i=N}^{2N-1} z(i)2^i=\sum_{i=0}^{N-1} z(N+i)2^i\,2^N=\sum_{i=0}^{N-2}(c(i)+s(i))2^i 2^N \quad (6)$$

The output of MAC design can be expressed as:

$$P=X \times Y+Z$$

By the putting the values in above equation from (4) and (6) then

$$P=(s_0 2Y+\sum_{i=1}^{(N+1/3)-2} s_i 2^{3i} Y+s_{(N+1/3)-1} 2^{N-2} Y) + (\sum_{i=0}^{N-1} z_i 2^i+\sum_{i=0}^{N-2}(c_i+s_i)2^i 2^N) \quad (7)$$

Similarly for Radix-4,output of MAC design can be expressed as:

$$P=(s_0 2Y+\sum_{i=1}^{\frac{N}{2}-2} s_i 2^{2i} Y+s_{(N/2-1)} 2^{N-2} Y) + (\sum_{i=0}^{N-1} z_i 2^i+\sum_{i=0}^{N-2}(c_i+s_i)2^i 2^N) \quad (8)$$

This equations (7) and (8) can be rearranged and by matching the bit positions, they can be expressed further as three parts and becomes the final equations (9) and (10) for the proposed MAC design.
The first parenthesis on the right expresses the operation to accumulate the first partial product with the added result of the sum and the carry. The second parenthesis expresses the one to accumulate the middle partial products with the sum of the CSA that was fed back. Lastly, the third parenthesis expresses the operation to accumulate the last partial product with the carry of the CSA.

For Radix-8

$$P=(s_0 2Y+\sum_{i=0}^{N-1} z_i 2^i)+(\sum_{i=1}^{(N+1/3)-2} s_i 2^{3i} Y+\sum_{i=0}^{N-2}(c_i 2^i 2^N)+(s_{(N+1/3)-1} 2^{N-2} Y+\sum_{i=0}^{N-2} s_i 2^i 2^N) \quad (9)$$

For Radix-4

$$P=(s_0 2Y+\sum_{i=0}^{N-1} z_i 2^i)+(\sum_{i=1}^{\frac{N}{2}-2} s_i 2^{2i} Y+\sum_{i=0}^{N-2} c_i 2^i 2^N)+(s(N/2-1)2^{N-2} Y+\sum_{i=0}^{N-2} s\,(i)2^i 2^N). \quad (10)$$

3.3 CSA Architecture

The desired architecture of the hybrid–type CSA tree that is applied on the partial products rows generated by 8x8 bit radix-4 and 8 MBA operation is shown in Figures 4 and 5 respectively. It was basically designed using equation (9) and (10).In figure 4 and 5, S_i designates the sign extension and N_i is to compensate 1'scomplement over 2's complement number. *C[i]* and *S[i]* represents the *i*th bit of the feedback sum and carry. *Z[i]* represents ith bit of the sum of the lower bits for each partial product row that were added in advance. Z'[i] is the previous results. In case of radix-4, total four partial products (P_0[7:0]-P_3[7:0] are generated and the CSA needs at least four rows of Full Adder In total five FA rows are necessary since one more stage of rows needed for accumulation. $S_0 2Y$ and $s_{N/2-1} 2^{N-2} Y$ corresponds to P_0[7:0] and P_3[7:0] respectively for radix-4 .

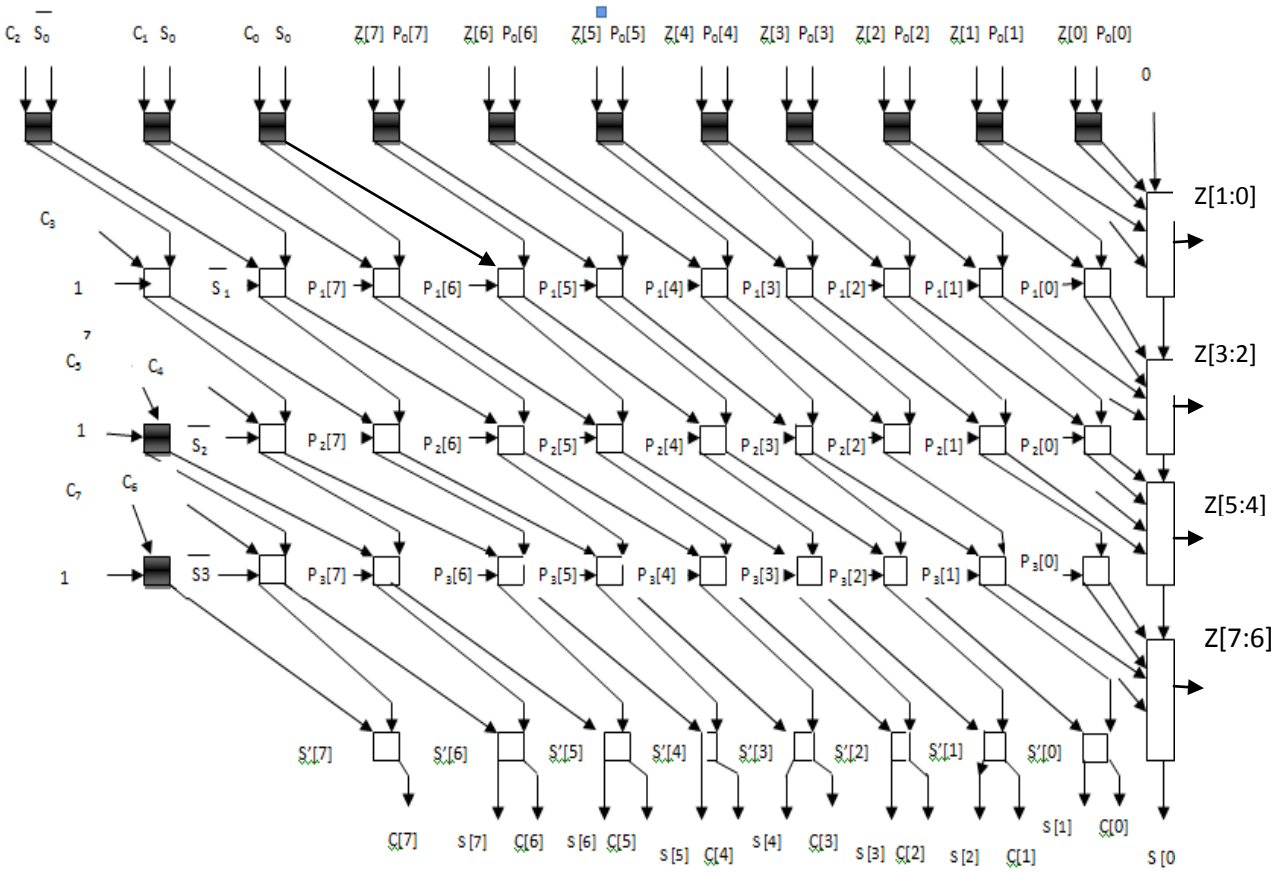

Figure 4: Architecture of CSA tree using radix-4 MBA

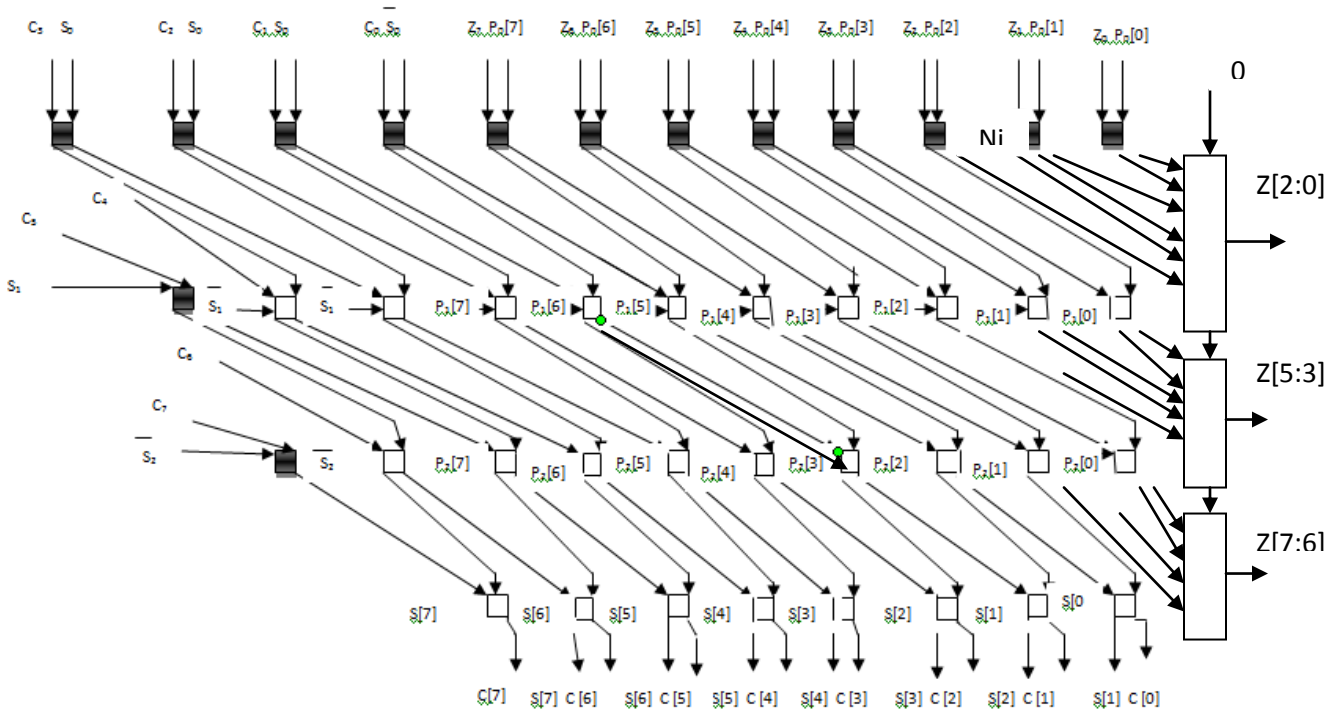

Figure 5: Architecture of CSA tree using radix-8 MBA

The white square in Figures 4 and 5 depicts Full Adder (FA) and the gray square is a half adder (HA).The rectangular block with four inputs represents 2-bit CLA with a carry input for radix-4MBA. The rectangular block with six inputs represents 3-bit CLA with a carry input for radix-8MBA.The number of bits for final adder step increases if the lower bits of the previously calculated partial products rows are not processed in advance by the CLA's.

3.4 Final Adder

Carry Select Adder [17] is the one of the fastest adder used in processors. Final addition is carried out between the upper bits of the CSA tree result as the lower bits are calculated earlier by CLA. This modified design of CSLA has power as well as area efficiency which is the main limitation of CSLA. Here instead of using two RCA for Cin='0'or'1', a single adder is used with BEC (Binary to Excess -1 Convertor) for Cin='1' in order to reduce area. So the basic design consists of CLA instead of RCA for each 2-bit input, one BEC and multiplexer to select output according to control signal Cin.

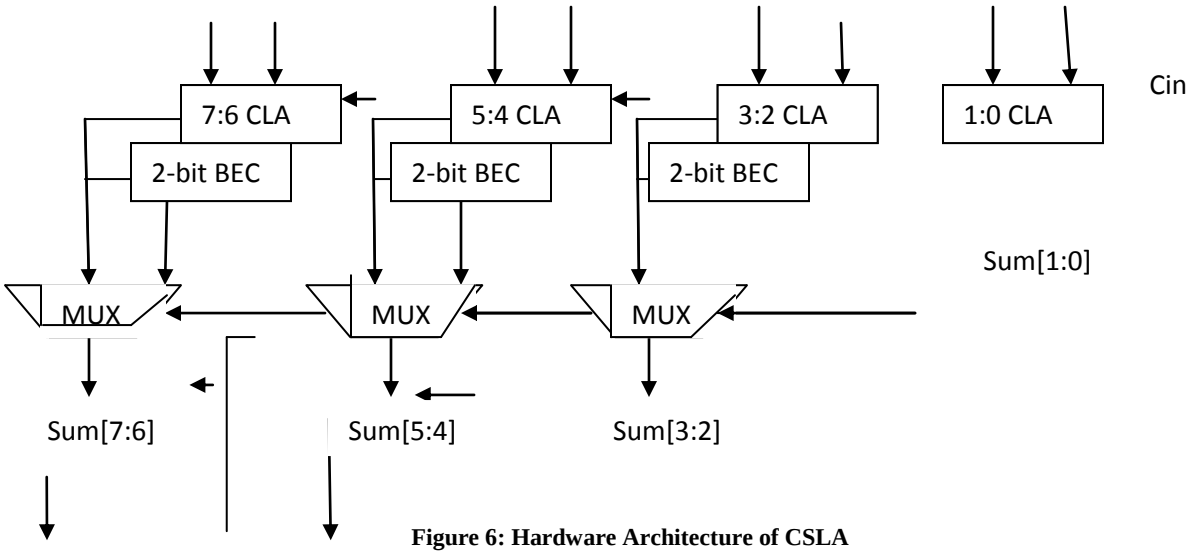

Figure 6: Hardware Architecture of CSLA

4. EXPERIMENTAL RESULT

The 8 x 8 bit MAC based on modified booth encoding (radix-4 and radix-8) and final adder (Carry Select adder using BEC) are designed and implemented in VHDL**.** The simulation waveforms for the design are shown in Figures 7 and 8.

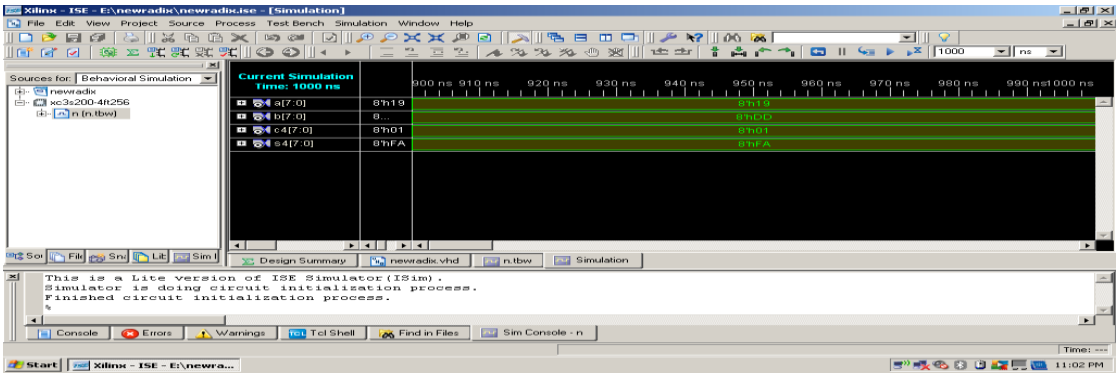

Figure 7: Simulation result of Pipelined 8-bit MAC based on radix-4 MBA

(I/Ps: X=0001101,Y=11011101 O/P :1111110010010101)

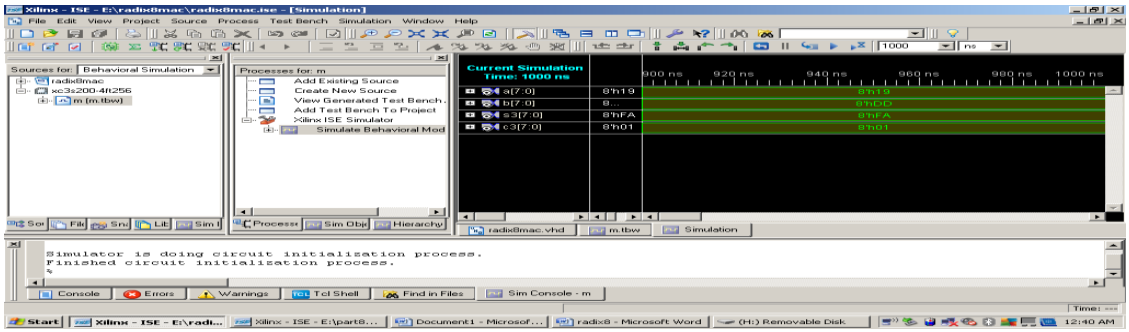

Figure 8: Simulation result of Pipelined 8-bit MAC based on radix-8 MBA

(I/Ps: X=0001101,Y=11011101 O/P :1111110010010101)

The maximum combinational delay for MAC using Radix-4 and Radix-8 MBA are 9.48ns and 12.60ns respectively. The high speed attained is on account of reducing partial product rows and carry select adder at final adder stage.

5. CONCLUSION

A new MAC architecture for the digital signal processing and multimedia information processing efficiently was designed. By eliminating the independent accumulation process that has the largest path delay and fusing it to the reduction process of the partial products, the overall performance of MAC has been improved to twice. This paper has shown that algorithms based upon the Radix-4 Booth partial product method are distinctly superior in performance when compared to Radix-8 MBA. The design can be further improved through architecture changes for better area and power requirements.

6. REFERENCES

[1] J. J. F. Cavanagh, *Digital Computer Arithmetic*. New York: McGraw-Hill, 1984.

[2] O. L. MacSorley, "High speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pp. 67-91, Jan. 1961.

[3] A. D. Booth, "A signed binary multiplication technique," *Quart. J.Math.*, vol. IV, pp. 236–240, 1952.

[4] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron Comput.*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964

[5] A. R. Cooper, "Parallel architecture modified Booth multiplier," *Proc.Inst. Electr. Eng. G*, vol. 135, pp. 125–128, 1988 8968190248.

[6] N. R. Shanbag and P. Juneja, "Parallel implementation of a 4x4-bit multiplier using modified Booth's algorithm," *IEEE J. Solid-State Circuits*,vol. 23, no. 4, pp. 1010–1013, Aug. 1988.

[7] A. R. Cooper, "Parallel architecture modified Booth multiplier," *Proc.Inst. Electr. Eng. G*, vol. 135, pp. 125–128, 1988.

[8] N. R. Shanbag and P. Juneja, "Parallel implementation of a 4□4-bit multiplier using modified Booth's algorithm," *IEEE J. Solid-State Circuits*, vol. 23, no. 4, pp. 1010–1013, Aug. 1988.

[9] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54□54 regular structured tree multiplier," *IEEE J. Solid-State Circuits*, vol. 27, no. 9,pp. 1229–1236, Sep. 1992.

[10] J. Fadavi-Ardekani, "M□N Booth encoded multiplier generator using optimized Wallace trees," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 120–125, Jun. 1993.

[11] F. Elguibaly and A. Rayhan, "Overflow handling in inner-product processors,"in *Proc. IEEE Pacific Rim Conf. Commun., Comput., Signal Process.*, Aug. 1997, pp. 117–120.

[12] F. Elguibaly, "A fast parallel multiplier–accumulator using the modified Booth algorithm," *IEEE Trans. Circuits Syst.*, vol. 27, no. 9, pp. 902–908, Sep. 2000.

[13] Samiappa,Sakthikumaran,S. Salivahanan, V. S. Kanchana Bhaaskaran, V. Kavinilavu, B. Brindha and CVinoth, "A Very Fast and Low Power Carry Select Adder Circuit",978-1-4244-8679-3/11.2011 IEEE.

[14] P. Devi and A. Girdher, "Improved Carry Select Adder with Reduced Area and Low Power Consumption", International Journal of Computer Applications (0975–8887) vol. 3 – No.4, June, 2010..

[15] Raahemifar, K. and Ahmadi, M., "Fast carry look ahead adder", IEEE Canadian Conference on Electrical and Computer Engineering, 1999.

[16] Taewhan Kim and Junhyung Um,"A Practical Approach to the Synthesis of Arithmetic Circuits Using Carry-Save-Adders" IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 19, NO. 5, MAY 2000.

[17] Hiroyuki Morinaka, Hiroshi Makino, Yasunobu Nakase," A 64bit Carry Look-ahead CMOS Adder using Modified Carry Select" IEEE 1995 CUSTOM INTEGRATED CIRCUITS CONFERENCE.

[18] A. Abdelgawad , Magdy Bayoumi "High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications"IEEE 2007.

[19] Shanthala S, Cyril Prasanna Raj, Dr.S.Y.Kulkarni, "Design and VLSI Implementation of Pipelined Multiply Accumulate Unit", IEEE computer society, 2009

[20] P. Zicari, S. Perri, P. Corsonello, and G. Cocorullo, "An optimized adder accumulator for high speed MACs," *Proc. ASICON 2005*, vol.2, pp. 757–760, 2005.

[21] K. Babulu et al, G.Parasuram "FPGA Realization of Radix-4 Booth Multiplication Algorithm for High Speed Arithmetic Logics", (IJCSIT) Vol. 2 (5) , 2011, 2102-2107.

[22] Young-Ho Seo, Dong-Wook Kim, ''A New VLSI Architecture of Parallel Multiplier–Accumulator Based on Radix-2Modified Booth Algorithm'' IEEE transactions on very large scale integration (VLSI) systems, vol. 18, no. 2, February2010.

[23]*G.A. Ruiz1 and Mercedes Granda*,"Efficient hardware implementation of 3X for radix-8 encoding", Proc. of SPIE Vol. 6590 65901I-1