Abstract: Power dissipation is recognized as a critical parameter in modern VLSI design field. This paper presents the low power compressor based Multiply-Accumulate (MAC) architecture for DSP Applications. In VLSI, highly computed arithmetic cells including adders and multipliers are the most copiously used components. Efficient architecture of MAC using a modified Wallace tree multiplier is proposed. The proposed MAC uses multiplier with novel compressor designs and adders as carry save adder for fast low-power application. The proposed low power compressor architecture was applied to MAC unit and compared against the conventional compressor based MAC units and observed that the proposed architecture has reduced significant amount of Delay and power.

Keywords: Multiply Accumulate, Compressor, Wallace Tree Multiplier, CSA.

I. INTRODUCTION

The increasing demand for portable systems and the need to limit power consumption and heat dissipation in very-high-density chips have led to rapid developments in low-power design during the recent times. The battery lifetime is also a concern on the overall power consumption of the portable system. Hence, reducing the power dissipation of integrated circuits through design improvements is a major challenge in portable systems design. The need for low-power design is also an issue in high-performance digital systems, like microprocessors, digital signal processors (DSPs) and other applications. In digital VLSI circuits, computation is the critical part and it decides the power consumption and operating speed of the designs. For computations arithmetic circuits involves adders and multipliers; which are the most copiously used components. Digital signal processors performing filtering, convolution and etc, relies on the efficient implementation of these adder, multiplier and MAC arithmetic units. Low power compressor architecture is proposed in this brief to reduce the power consumption of the MAC architecture since the presence of more number of compressors. The impact of the circuit design level or the data path optimizations is addressed at the MAC level for DSP applications. In MAC, additionally the carry propagate addition involved in multiplier and accumulate stages are merged to exploit and increase the number of compressors in the MAC architectures. Designs were illustrated in ASIC and FPGA domains as per the standard design methodology.

II. CONVENTIONAL COMPRESSOR ALGORITHM

Multipliers require high amount of power and delay during the partial products addition. At this stage, most of the multipliers are designed with different kind of adders that are capable to add two/three or at most 4 bits by using 4-2 compressors. For higher order multiplications, a huge number of adders or compressors are used to perform the partial product addition. We have minimized the number of adders by introducing different compressors. The conventional 4-2 compressor structure actually compresses five partial product bits into three. The architecture can be implemented with two stages of full adder (FA) connected in series as shown in Fig. 1. The outputs of 4-2 compressor consist of one bit in position j and two bits in position (j + 1). This straightforward approach has four XOR gate delays.

![Fig. 1. Conventional 4-2 compressor.](image)

This implementation is better and the delay is that of three XOR gates delays. With the similar logic 5-2 compressor. The problems of this kind of conventional compressor are:

- The uneven delay profile of the outputs arriving from different input paths tends to generate a lot of glitches.
- Compressors do the simple operation of addition that adds more number of bits at a time. But the conventional 4-2 compressors require one more half adder of which two inputs are ‘COUT’ and ‘C’ (shown in Fig. 2), to produce the final addition result. Example: if \(X_1=X_2=X_3=X_4=1\) and \(CIN =0\) (in Fig. 1) then the addition result be four i.e. 100 but the conventional architecture produces \(COUT=1\), \(C=1\) and \(S=0\). Now if
COUT and C fed to a half adder then it produces the final result in exact form as shown in Fig. 2.

Fig. 2. Modified 4-2 compressor.

- For 4-2 compressor, a half adder is required but for 5-2 compressor a full adder is required because a 5-2 compressor is implemented by series connection of three full adders, that generates three carry output bits in position ‘j+1’ and one sum bit in position ‘j’, shown in Fig. 3. Thus this conventional logic not only increases the critical path delay but also increases the number of output bits.

Fig. 3. Conventional 5-2 compressor.

As the weightage of sum bit is ‘1’ and the weightage of carry bits is ‘2’ of conventional compressors, so the results that produced by those compressors are correct but not in proper binary form. When these conventional compressors are used in multiplier to achieve high speed then one half adder/full adder is required per compressor to process those carry bits. Thus it hampers the speed of operation. So the conventional compressors require one more half adder/full adder to get the final result and this eventually adds more delay and power to the reported results.

III. MULTIPLICATION LOGIC

Considering an example of 8 bit multiplication in which 8 bit input is $X_7X_6X_5X_4X_3X_2X_1X_0$ and multiplier is $Y_7Y_6Y_5Y_4Y_3Y_2Y_1Y_0$. The multiplication process is shown in fig. 4. There is the requirement of 64 AND logics. First $Y_0$ is multiplied with $X_7X_6X_5X_4X_3X_2X_1X_0$ and results $X_0Y_0$, $X_1Y_0$, $X_2Y_0$, $X_3Y_0$, $X_4Y_0$, $X_5Y_0$, $X_6Y_0$ and $X_7Y_0$. After it $Y_1$ is multiplied with $X_7X_6X_5X_4X_3X_2X_1X_0$ and results $X_0Y_1$, $X_1Y_1$, $X_2Y_1$, $X_3Y_1$, $X_4Y_1$, $X_5Y_1$, $X_6Y_1$ and $X_7Y_1$. Similarly all multiplications are taken place. In each step there is one binary shift in the resultant logic. All AND logics are represented by one bit representation starting from K0 to K63 sequentially as shown in fig. 5. The addition can be done using a tree formed itself. This is done using 3:2 compressor, 4:2 compressor and 5:2 compressor which are the optimized solutions instead of using 3:2 compressors only. This addition is possible using 3:2 compressors only but the implementation using 4:2 and 5:2 reduces the latency and increases the speed. In the process the sum output of intermediate compressors is the input for next compressors in the same column and the generated carry for the corresponding adders are propagated to next column adders. The result will be of 16 bits represented by $[P_{15}, ..., P_0]$. Several popular and well-known schemes, with the objective of improving the speed of the parallel multiplier, have been developed in past. Wallace introduced a very important iterative realization of parallel multiplier. This advantage becomes more pronounced for multipliers of bigger than 16 bits. In Wallace tree architecture, all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. The advantage of Wallace tree is speed because the addition of partial products is now $O(\log N)$.

Fig. 4. Multiplier (8 bits).

Fig. 5. Multiplication wallace Logic tree.
A. Carry Save Adder (CSA)

The Carry Save Adder (CSA) is a type of Digital Adder, used to compute the sum of three or more number of bits in binary form. CSA gives less propagation delay and the Glitching problem in RCA is also avoided. Since, the Representation of 8 bit CSA is shown in Fig.6. Here, we compute the sum of two 8 bit binary numbers so 8 half adders at the first stage is required instead of 8 full adders. Since, we add bits of two binary numbers only. If, P and Q are two 8 bit numbers then it produces the partial products and carry Si and Ci respectively. Where,

\[ S_i = P_i \cdot Q_i \]  
\[ C_i = P_i \cdot Q_i \]  

However, a CSA produces all the output values in parallel. So that, the computation time is reduced compared to RCA. Also, Parallel in Parallel out (PIPO) is used in Accumulator Stage.

IV. MAC UNIT

The Multiplier-Accumulator (MAC) operation is the key operation not only in DSP applications but also in multimedia information processing and various other applications. As mentioned above, MAC unit consist of multiplier, adder and register/accumulator. In this paper, we used 8 bit modified Wallace multiplier. The MAC Unit take inputs from the memory location such as RAM and given to the multiplier block. This is very useful in 8 bit digital signal processor. The inputs which is being fed from the memory location is 8 bit. When the input is given to the multiplier it starts computing value for the given 8 bit input and hence the output will be 16 bits. The multiplier output is given as the input to carry save adder (CSA) which performs addition. The function of the MAC unit is given by the following equation

\[ Y = \sum_{i=0}^{7} A_i \cdot B_i \]  

where, Ai & Bi are two 8 bit input Operands, Y is the output of MAC Unit and i is a 8 bit value. This Equation performs summation of partial products. The Carry Save Adder (CSA) produces 17 bit output. Since, one bit is for the carry (16 bits +1 bit). Then, the output of CSA is given to the accumulator register. The accumulator used is designed with Parallel in Parallel out (PIPO) Type. Because the CSA produces output in Parallel form and also the bits are huge. PIPO register is used where the input bits are given in parallel and output is taken in parallel. The output of the accumulator register is taken out or fed back as one of the input to the CSA. Fig. 7 shows the basic architecture of MAC unit.

V. RESULTS

Results of this paper is as shown in below Figs.8 to 11.
VI. CONCLUSION

Hence, a High Performance 8 bit MAC Unit is designed and implemented using compressor based Wallace Tree Multiplier and Carry Save Adder. When compared to all other MAC Units which are developed earlier using different combinations of multipliers and adders, the designed compressor based Wallace Tree Multiplier offers High Performance with Less Delay, Less Power Dissipation which further increases the overall speed of MAC Unit. This MAC Unit is designed using Verilog - HDL and Synthesized using Xilinx 14.3 ISE.

VII. REFERENCES


Author’s Profile:

NAGAMANOHAR TENALI is currently PG scholar of VLSI in Audisankara college of Engineering and Technology, Gudur (Autonomous), SPSR Nellore (Dist), Affiliated to JNTU Anantapur. He received B.TECH from Ramireddy Subaramireddy Engineering College, Kadanuthala, SPSR Nellore (Dist). His current research interest includes Analysis & Design of VLSI System Design.

MUKESH GANGALA received B.Tech from Narayana Engineering College, Gudur, SPSR Nellore (Dt), AP in the year of 2009. He pursued M.Tech from PBR VITS, Kavali, SPSR Nellore (Dist), AP in the year of 2013. He is having teaching experience of 3 years in Audisankara college of Engineering & Technology (Autonomous), Gudur. SPSR Nellore, AP. His interested areas are Embedded System and VLSI.