International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

FPGA Implementation of Low Power and High Speed 64-Bit Multiply Accumulate Unit for Wireless Applications

R. Malleshwari[1], E. Srinivas[2]

[1]M. Tech-Student (VLSI System Design), CVSR College of Engineering

[2]Assistant Professor, CVSR College of Engineering

Abstract: ***The MAC operation is the main computational kernel in Digital Signal Processing (DSP) architectures. The MAC unit is considered as one of the fundamental operations in DSP and it becomes a basic component in Application-Specific-Integrated-Circuits (ASIC).The MAC unit determines the speed of the overall system; it always lies in the critical path. Developing a high speed MAC is crucial for real time DSP applications. In this project 64-bit MAC unit is designed and it performs multiplication and addition to accumulator at a time in a summation network and here two architectures are developed, i.e. merged and proposed architechtures.The merged architecture is based on fully utilizing the summation tree. Feeding the accumulated data bits into the unused inputs of 4:2 compressors. This will save the cost of the additional accumulator by merging the accumulation operation with the multiplication circuit. In proposed 64-bit multiply and accumulate unit architecture, saving the area is achieved by fully utilizing the compressors instead of putting zeros in free inputs by using 5:2 compressor technique. And increasing the speed is achieved by reducing the critical path by changing the MAC structure. The proposed 64-bit MAC unit saves 21.9% of area, 75.7% of power and reduces the delay by 12% compared to the regular merged 64-bit MAC unit.64-bit MAC unit is designed by using verilog HDL in Xilinx ISE14.3i and implemented on Zynq FPGA development board.***

Keywords: Power, Delay, Area, Multiply accumulate unit (MAC), digital signal processing, compressors, and 4:2compressor, 5:2compressor.

1. Introduction

The multiply accumulate unit represents one of the essential building blocks that is frequently used in digital signal processing applications. Applications such as Digital filtering, speech processing, video coding and CDMA require high processing speed while maintaining a low power consumption [1], [2] that allows embedding powerful processors in applications that are using portable power supplies. In order to improve the speed of this unit, there are two major bottlenecks that need to be considered.

The first one is the partial products reduction network that is used in the multiplication block and the second one is the accumulator. Both of these stages require addition of large operands that involve long paths for carry propagation. The classical straightforward approach to build multiply accumulate units implements each part separately using individual functional blocks [1] and then cascade them to realize the complete operation. Another approach to speed up the operation implements both the multiplication and the accumulation operations within the same functional block by merging the accumulator with the computational platform for hardware and software components to interact with the environment and other nodes. the multiplication circuit. Using tree architectures for the partial products reduction network represent an attractive solution that is frequently applied to speed up the multiplication process. Our contribution in this paper is introducing a merging technique for further speeding up the accumulation operation by merging it within the partial products reduction tree used for multiplication.

Each sensor node has a processing unit. Many different types of processing units can be integrated into a sensor node. There are a large number of commercially available microcontrollers, DSPs, and field programmable gate arrays (FPGAs), which allows a big flexibility for processing unit implementations. The sensor nodes available in the market depend on an 8-bit or 16-bit microcontroller. FPGA is not used in the current sensor nodes because its power consumption is not as low as sensor nodes should be. Moreover the FPGA is not compatible with traditional programming methodologies. Currently, DSP is being a challenge for node demanding, such as a gateway or a robust sensor node, which can be the head of hierarchical cluster in a wireless sensor network. The future WSN needs DSP for more computational capabilities in order to engage in signal processing operations of the complex applications.

Signal processing in wireless sensor network has a huge range of applications. Infinite Impulse Response filtering (IIR), Finite Impulse Response filtering (FIR), and Kalman Filter (KF) find applications in object tracking, environmental monitoring, surveillance, and many other applications. These tasks are very computationally intensive and they could easily strain the energy resources of any single computational node in a wireless sensor network. In other words, most sensor nodes do not have the computational resources to complete many of these signal-processing tasks repeatedly. Since MAC unit is the main computational kernel in DSP architectures. Therefore, saving power at the MAC unit level will have a significant impact on the energy consumption of each node.

Consequently, energy efficient system extends the WSN‟s lifetime and increases its computational capabilities. In this

work we propose a fast and low power MAC Unit.

This paper is organized as follows. Section 2 provides a brief background in the general construction of the MAC unit. Section 3 explains the merging architecture. Section 4 introduces the proposed architecture. The results are given in section 5. Section 6 concludes the work.

2. General Construction of MAC

In computing, especially digital signal processing, the multiply–accumulate operation is a common step that computes the product of two numbers and adds that product to an accumulator. The hardware unit that performs the operation is known as a multiplier–accumulator (MAC, or MAC unit); the operation itself is also often called a MAC or a MAC operation. The MAC operation modifies an accumulator *a*:The MAC operation is the main computational kernel in Digital Signal Processing (DSP) architectures. The MAC unit is considered as one of the fundamental operations in DSP and it becomes a basic component in Application-Specific-Integrated-Circuits (ASIC).The MAC unit determines the speed of the overall system; it always lies in the critical path. Developing a high speed MAC is crucial for real time DSP applications. Moreover, with the ever-increasing demand for WSN, a MAC unit with low power consumption would surely lead the market.

In order to improve the speed of the MAC unit, there are two major bottlenecks that need to be considered. The first one is the partial products reduction network that is used in the multiplication block and the second one is the accumulator.

Multiply-Accumulate is a common operation that computes the product of two numbers and adds that product to an accumulator. The multiplier A and multiplicand B are assumed to have n bits each and the addend Z has (2n+1) bits.

$$Z \leftarrow (AxB) + Z$$

The MAC Unit is made up of a multiplier and an accumulator as shown in Fig. 1.

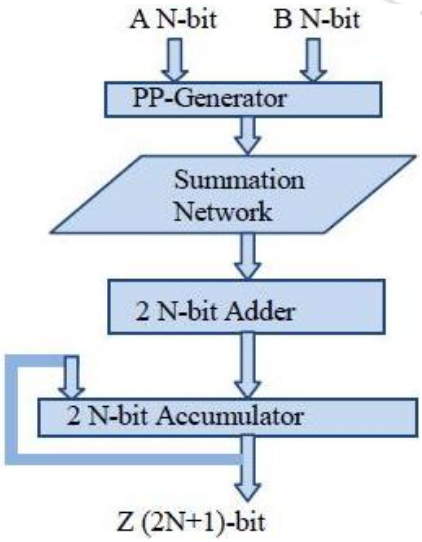

Figure 1: Basic Mac Unit

The multiplier is divided into the partial products generator, summation network, and final adder. The summation network represents the core of the MAC unit; it reduces the number of partial products into two operands representing a sum and a carry. The summation network occupies most of the area and consumes most of the circuit area and delay. Several algorithms and architectures are proposed in an attempt to optimize the implementation of the summation network. The final adder is then used to generate the multiplication result out of these two operands. The accumulator is used to perform a double precision addition operation between the accumulated operand and multiplication result. Due to the large operand size, the accumulator required a very large adder.

3. The Merging Architecture

The merged MAC architecture is based on fully utilizing the summation tree. Feeding the accumulated data bits into the unused inputs of the 4:2 compressors as shown in Fig. 2 can do this. This will save the cost of the additional accumulator by merging the accumulation (1) operation with the multiplication circuit. This can directly result in increasing the overall speed of the MAC operation. Power consumption and circuit area are saved as well.

The 8-bit MAC unit architecture is based on taking advantage of the free input lines of the available 4:2 compressors to implement the accumulation operation by feeding the bits of the accumulated operand into the summation tree, as shown in Fig. 3. The decision of where to insert the bit Z7 determines the number of required modified 4:2 compressors. The 8-bit MAC unit architecture consists of two compression stages. Stage one has two levels of compression. Each level has a combination of half adder, full adder and a modified 4:2 compressor [9].

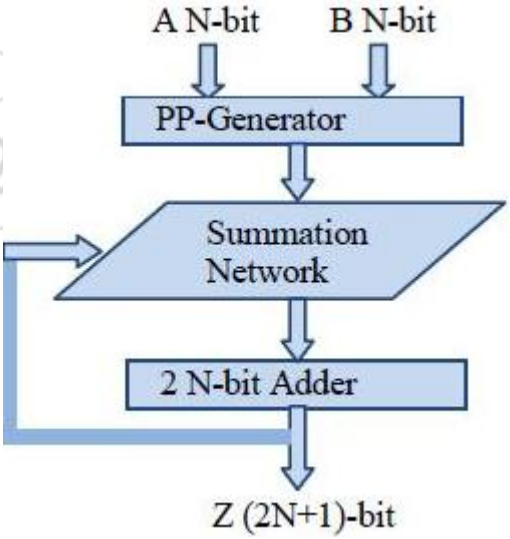

Figure 2: Merged Architecture

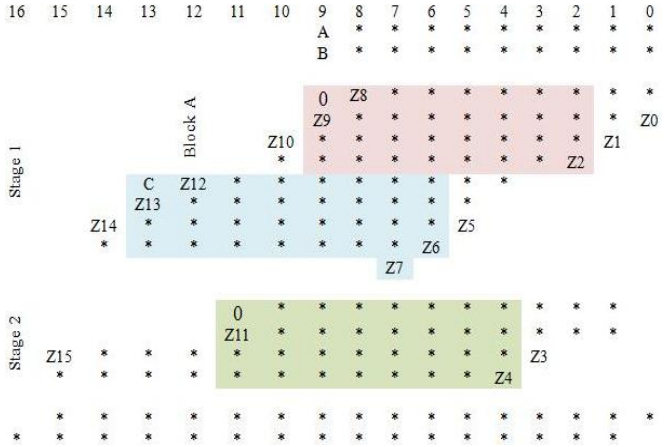

Figure 3: Data bit Distribution in the Merged MAC Unit

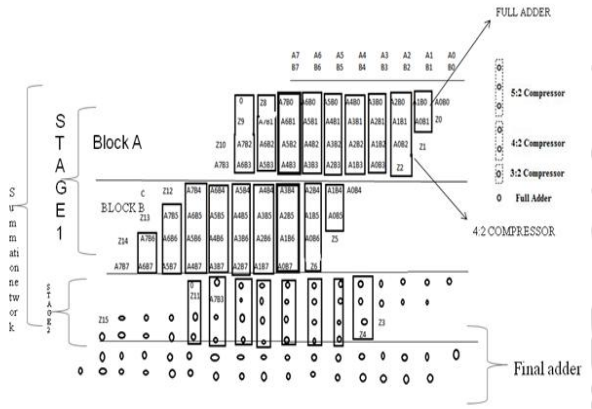

Figure 3: Example for 8-bit Data bit Distribution in the Merged MAC Unit

The best place to insert this bit is in the 8th column in block "A" of the first stage and use a modified 4:2 compressor at this location to accommodate the extra bit [9]. The modified 4:2 compressor used will generate an additional carry out bit that is supposed to be fed to the next column. This in turn will require an additional modified 4:2 compressor in the 9th column at the first stage, which will also have an additional carry out.

4. Proposed Architecture

The proposed MAC architecture is based on increasing the overall speed and throughput of the MAC and reducing the overall area of it. Saving the area is achieved by fully utilizing the compressors instead of putting zeros in free inputs. Increasing the speed is achieved by reducing the critical path by changing the MAC structure. The 32-bit MAC unit architecture is very large, so the 8-bit MAC unit architecture is used to explain the contribution and the 32-bit MAC unit can apply the same principle.Fig. 4 shows the proposed distribution for an 8-bit MAC unit; the compression network consists of two stages. In the first stage, there are two compression levels A and B that are used to reduce the number of partial products from 8 to 4. The bits of the accumulated operand Z are inserted within the network, before the final adder in the second stage, in those empty locations at the corresponding columns such that those 4:2 compressors are fully utilized. The bits Z0, Z1, Z2, Z12, Z13, and Z14 inserted in the first stage, the bits Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z10, Z11, and Z15 are inserted in the second stage. The proposed structure reduces the number of compressors from 24 to 22 compressors. Two full adders are used instead of the saved two compressors. In the fifth column we used a modified compressor in stage 1 (block A), so we were capable of using a full adder in stage 2.We also use two modified compressors in the eighth, ninth and the eleventh column to save a compressor in the tenth column in stage 2 (block A), as shown in Figure 4.The feedback of the critical path of Z8 output is done in the stage 2 to speed up the design, instead of feeding it in stage 1, so we save the time, which was consumed in stage 1. When the number of input bits increases the number of the stages in the summation network increases. The numbers of stages for 8-bit, 16-bit and 32- bit MAC unit are 2, 4 and 9 stages respectively. So the time saving will increase for 32-bit MAC unit, because it has many stages.

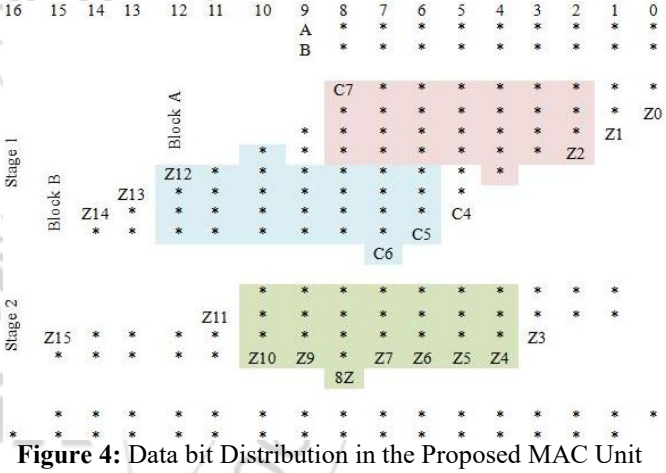

Figure 4: Data bit Distribution in the Proposed MAC Unit

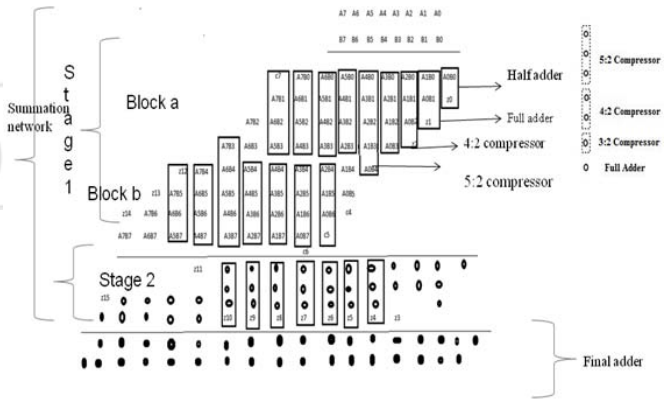

Fig5. Example for 8-bit Data bit Distribution in the Proposed MAC Unit

5. Simulation Results

Proposed_8-bits

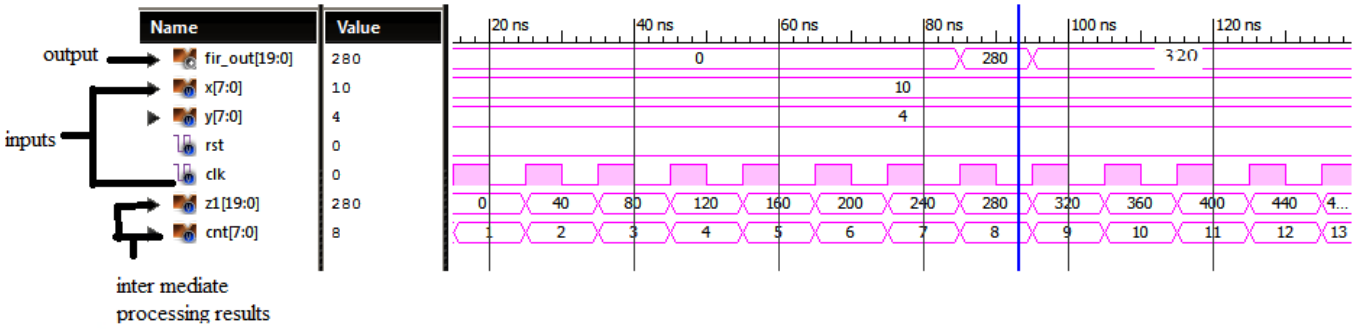

Figure 6: Simulation results for proposed 8-bit MAC Unit

Proposed_16-bits

Figure 7: Simulation results for proposed 16-bit MAC Unit

Proposed_32-bits

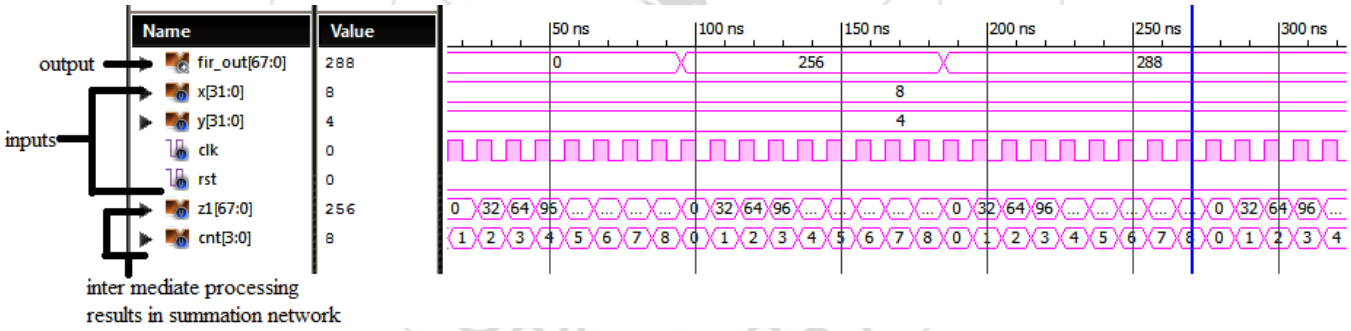

Figure 8: Simulation results for proposed 32-bit MAC Unit

Proposed_64-bits

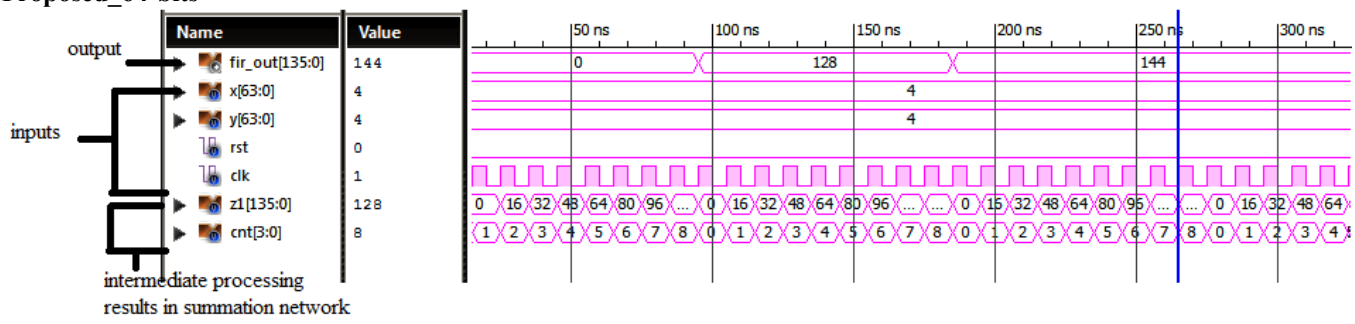

Figure 9: Simulation results for proposed 64-bit MAC Unit

Table 1: Comparison Table

	Performance Indicator	Merged architecture (4:2 compressor)	objective	Proposed architecture (modified compressor or 5:2 compressor)	DEVIATION
8-bit MAC UNIT	Area	No of slices 119 out of 4656		No of slices 189 out of 4656	----------
	Delay(ns)	31.140 ns	31.05 ns	31.076ns	1%
	Power(mw)	132 mw	110 mw	107 mw	18%
16-bit MAC UNIT	Area	No of slices 477 out of 4656	-	No of slices 401 out of 4656	---------
	Delay(ns)	54.988 ns	52.555 ns	52.861 ns	3.8%
	Power(mw)	151 mw	138 mw	130 mw	20%
32-bit MAC UNIT	Area	No of slices1859 out of4656		No of slices 119 out of 4656	--------
	Delay(ns)	103.771ns	90.043 ns	91.628ns	11%
	Power(mw)	275 mw	189 mw	180 mw	34%
64-bit MAC UNIT	Area	No of slices 10935 out of 28,800	-	No of slices 7840 out of 28800	----------
	Delay(ns)	37.771 ns	37.000 ns	37.362 ns	1%
	Power(mw)	6124 mw	1100 mw	1000 mw	75%

6. Conclusion

In this project, developed a high speed and low power Multiply Accumulate (MAC) Unit. Simulation result and delay analysis show that the proposed technique outperforms other merging techniques. The simulation results shows that the proposed technique for 8-bit, 16-bit,32-bit and 64-bit MAC unit reduces power by 18%,20%,30% and 75% respectively and increases the speed by 1%, 11%,3.8 and 12% respectively over the previous merged MAC unit technique.

References

[1] A. Abdelgawad and Magdy Bayoumi, "High Speed and Area-Efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications," in IEEE, pp. 3199-3202, 2007.

[2] Fayed, Ayman A., Bayoumi, Magdy A., "A Merged Multiplier-Accumulator for high speed signal processing applications", IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 3212 -3215, 2002.

[3] S. J. Jou, C. Y. Chen, E. C. Yang and C.C.Su, "A pipeline Multiplier-Accumulator using a high speed, low power static and dynamic full adder design", IEEE custom Integrated circuit conference, 1995, pp. 593-596.

[4] CMPEN 411VLSI Digital CircuitsSpring 2012Lecture 20: Multiplier Design [Adapted from Rabaey,s *Digital Integrated Circuits*, Second Edition, ©2003 J. Rabaey, A. Chandrakasan, B. Nikolic].

[5] C.N. Marimuthu and P. Thangaraj, "Low power high performance multiplier", Proc. ICGST-PDCS, vol. 8, pp.31–38, December 2008. International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.3, June 2012.

[6] P.C Anantha, and S.Y.Kulkarni (2009) : VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique, European Journal of Scientific Research,Vol 30,No.4, pp.620-630, 2009 .

[7] E. Musoll and J. Cortadella, "High-level synthesis techniques for reducing the activity of functional units," Proceedings of International Symposium on Low Power Design, pp. 99-104, Nov. 1995.

[8] L. Benini, P. Vuillod, G. D. Micheli, and C. Coelho, "Synthesis of low power selectively-clock systems from highlevelspecification," Proceedings of International Symposium on System Synthesis, pp. 57-63, Nov. 1996.

[9] D. Booth, "A Signed Binary Multiplication Technique", Quarterly J. Mechanical Applications in Math., vol. 4,part 2, pp. 236-240, 1951.

[10] L. Chen, T. Wang, C. Wang, "A multiplication accumulation computation unit with optimized compressors and minimized switching activities" 48th Mid west Symposium on Circuits and Systems, pp.1223

- 1226, 2005.
[11] V. Oklobdzija, D. Villeger, "Improving Multiplier Design by Using Improved Column Compression Tree and Optimized Final Adder in CMOS Technology," IEEE Transation. on VLSI, Vol. 3, No. 2, pp. 292-301, June 1995.
[12] H. Murakami, N. Yano, Y. Oo taguro, Y. Sugeno, M. Ueno, Y. Muroya, and T. Aramaki, "A multiplier-accumulator macro for a 45 MIPS embedded RISC pro cessor," IEEE Journal Solid-State Circuits, vol. 31, pp. 1067-1071, July 1996.