A Novel High Speed FPGA Architecture for FIR Filter Design

Sachin B. Jadhav, Nikhil N. Mane
Department of Electronics and Telecommunication Engineering,
Bharti Vidhyapeeth's College of Engineering, Kolhapur.
e-mail - sachin30_2k6@ rediffmail.com, mane.nikhil@rediffmail.com

Article Info

Article history:

Received Feb 10, 2012
Revised Mar 9, 2012
Accepted Mar 13, 2012

Keyword:

application specific integration circuits (ASCIs)
digital signal processing (DSP)
field programmable array (FPGA)
multiply and accumulate (MAC)
partial product generator (PPG)

ABSTRACT

This paper presents the details of hardware implementation of linear phase FIR filter using merged MAC architecture. Speed of convolution operation of FIR filter is improved using merged MAC architecture. By exploiting the reduced complexity made possible by the use of sparse powers of two partial products terms coefficients, an FIR filter tap can be implemented with 2B full adders, and 2B latches, where B is intermediate wordlegnth. Word and bit level parallelism allows high sampling rates, limited only by the full adder delay. The proposed architecture is based on binary tree constructed using modified 4:2 and 5:2 compressor circuits. Increasing the speed of operation is achieved by using higher modified compressors in critical path. Our objective of work is, to increase the speed of multiplication and accumulation operation by minimizing the number of combinational gates using higher n: 2 compressors, which is required more for Array multiplier at the time of implementation of array architecture. This novel architecture allows the implementation of high sampling rate filters of significant length on FPGA Spartan-3 device (XC3S400 PQ-208). The simulation result shows convolution output of digital FIR filter which is done using Questa Sim 6.4c Mentor Graphics tool. The experimental test of the proposed digital FIR filter is done using Spartan-3 device (XC3S400 PQ-208).

Corresponding Author:

Sachin B. Jadhav
Department of Electronics and Telecommunication,
Bharti Vidhyapeeth's College of Engineering,
Near Chitranagri, Morewadi, Kolhapur - 416013.
Maharashtra, India.
Email: sachin30_2k6@ rediffmail.com

1. INTRODUCTION

FPGAs are widely used for variety of computationally intensive applications, mainly in Digital Signal Processing (DSP) and communication. The multiplication and addition performed through multiply and accumulate unit (MAC) which is the main computational kernel in Digital Signal Processing (DSP) architectures. A variety of approaches to high speed implementation of FIR filters have been pursued [1, 2, 3, 6]. In order to attain high performance, parallel improvement strategies such as, binary tree methods have been applied. Such a bit parallel processing techniques has gives improvements in implementation technology and increase demands for high performance.

This paper presents a new Parallel FIR filter architecture suited for increasing the speed multiplication and addition operation using 4:2 and 5:2 higher compressors. This novel technique is used to implement FIR convolution operation, where generated result is a sum of several powers of two partial products terms. This architecture allows high speed of FIR convolution operation of substantial length to be implemented on the new generation of FPGA. The high speed of FIR convolution operation is obtained through this architecture is due to use of higher n: 2 compressors at the stage of coefficient multiplication and addition operation which is a main computational operation. The implementation efficiency is a result of improving overall propagation delay in the adder stage operation which mainly cause for speed up the operation as compare to array multiplier architecture.

In binary arithmetic, multiplication by power- of- two is simply a shift operation. Implementation of systems with multiplications may be simplified by using only limited number of power- of- two terms, so that only a small number of shift and add operations are required. The improvements in speed and saving extra adder area are, however achieved at stage of summation tree (multiplication and addition operation) in the merged MAC architecture shown in figure 1. Where multiplication and accumulation take place in the same space. The extent of response of filter is depends on the number of power of-two terms used in approximating each partial product coefficient value, the architecture of the filter, and optimization of delay technique used to derive the discrete space coefficient values.

The FIR filter architecture which is used for realization of convolution operation using two power-of- two terms for each coefficient value, given that the filter is in cascade form coefficient values are derived using unsigned integer linear programming.

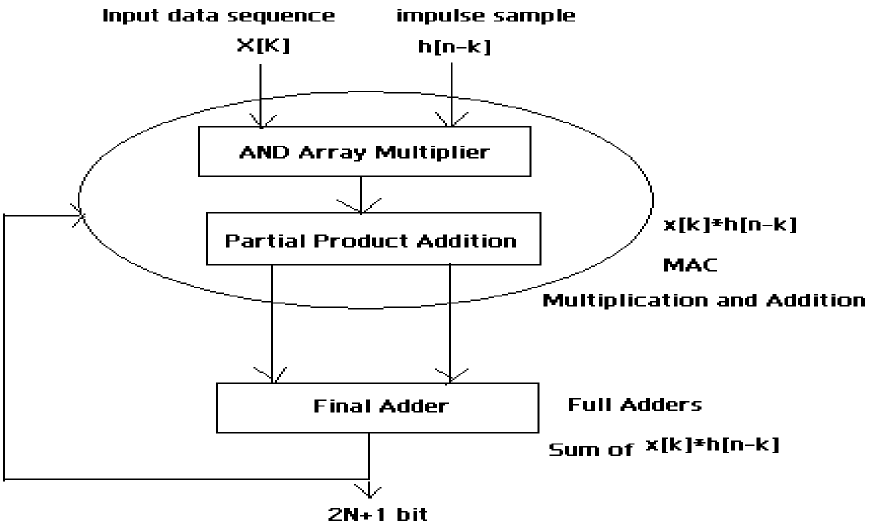


Figure 1. Merged MAC Architecture.

These 3 basic functional blocks are implemented by using;

1. PPG- Consist multiple AND gates.
2. Summation N/W- Consist partial product reduction N/W in to two operands representing SUM and CARRY. This is implemented by using 4-2 counter/compressor for improving the speed of the partial product addition.
3. Final adder- used to generate the multiplication result out of these 2 operand; we use CSA; for improving speed. Here Accumulator is used to perform double precision addition operation between multiplication result & the accumulated operand which is merged in to summation network.

Mathematically the MAC is represented by the multiplication of two matrices, a matrix or scalar which implies each element of the vector is multiplied by scalar. Two matrices A and B can be multiplied if the number of columns **A** equals to the number of rows in **B**. Then the product of two matrices P=AB, is matrix of order (mxn) with elements.

$$\textbf{Product } P = A \bullet B_i$$

Where the multiplier, **Bi** is an unsigned integer number in binary number **n**-bit wide and the multiplicand, **A** is an **m**-bit wide, and **i** represent the bit placement.

$$B = \sum_{i=0}^{K-1} A \bullet B_i \bullet 2^i \qquad \textbf{(1)}$$

$$P = A \bullet B_i = \sum_{i=0}^{K-1} A \bullet B_i \bullet 2^i \qquad \textbf{(2)}$$

$$P = B_0 \bullet A \bullet 2^0 + B_1 \bullet A \bullet 2^1 + B_2 \bullet A \bullet 2^2 + B_3 \bullet A \bullet 2^3$$

The above expressions operation is illustrated in the logic given below,

A Multiplicand				A3	A2	A1	A0	
B Multiplier			×	B3	B2	B1	B0	
				A3 B0	A2 B0	A1 B0	A0 B0	$B_0 \bullet A \bullet 2^0$
+			A3 B1	A2 B1	A1 B1	A0 B1		$B_1 \bullet A \bullet 2^1$
		A3 B2	A2 B2	A1 B2	A0 B2			$B_2 \bullet A \bullet 2^2$
	A3 B3	A2 B3	A1 B3	A0 B3				$B_3 \bullet A \bullet 2^3$
	P6	P5	P4	P3	P2	P1	P0	**P Product**

It can be seen that Equation 1.2 is nothing more than a series of binary AND operations (multiplication) to determine add or no add decision followed by a shift and ADD function (accumulation). Equation II is a representation of fundamental MAC. Checking the bits of the multiplier one at a time and summing partial products is a sequential operation that require add and shift sub-operations.

The basic structure of an FIR filter is illustrated in Figure 2.For FIR convolution operation the coefficient values are in the sum of two powers –of –two, the multipliers can be replaced by bit serial shifters, as depicted in Figure .Since the coefficient values will be fixed for this type of filter architecture, the coefficient values can be realized by approximating routing the inputs to the full adder and 4:2, 5:2 compressors [4] in the filter structure. That is moving the adder inputs **i** places to the left achieves same effect as would a coefficient value of $2^{\mathbf{i}}$**.**

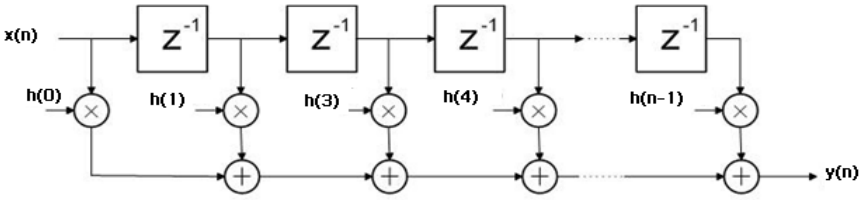


Figure 2. Basic Structure of FIR filter.

2. IMPLEMENTATION OF MERGED MAC ARCHITECTURE.

The tree basic functional blocks of architecture are Partial product Generator, Summation network and Final adder. The implementation details of each functional block are as follows.

2.1 Partial product Generator

It consist multiple AND gates, for generation of partial products. Array multipliers are probably most common. The multiplication of two binary number can be done with one sub operation by means of a combinational circuit that forms the product bits all at a once . This a faster way of multiplying two numbers since all it takes is time for the signal to propagate through the gates that construct the multiplication array. However the array multiplier requires a large number of gates for this reason it is not economical for MAC intensive applications. Figure 3, shows Generation of partial product using array of AND gates.

For our proposed MAC architecture we have considered Multiplicand and Multiplier bits are 8-bit each. The first partial product is formed by multiplying B0 by A7, A6, A5, A4, A3,A2,A1,A0; similarly last partial product is formed by multiplying B7 by A7, A6, A5, A4, A3,A2,A1,A0 respectively. For this example first partial product is formed by means of eight AND gates. Second partial product is formed by multiplying B1 A7, A6, A5, A4, A3, A2, A1, and A0 and is binary shifted one bit to the left. Similarly remaining each bit of multiplier will be multiplied with each bit of multiplicand and forms partial products. For 8-bit MAC 64 partial products were created, these partial products are added with Half adder, Full Adder, 4:2 Compressors, 5:2 compressors.

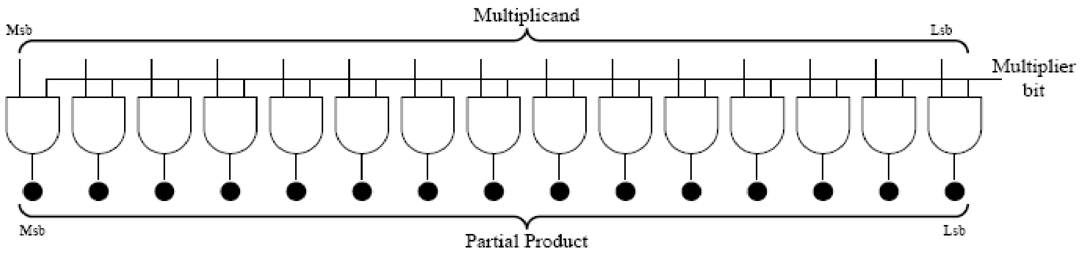


Figure 3. Generation of partial product using array of AND gates.

2.2 Summation network

Summation network consist of partial product reduction network into two operands representing SUM and CARRY. This is implemented by using 4-2 counter/compressor for improving the speed of the partial product addition. Figure 4, Shows 8 x 8 partial product Array reductions using 4:2 and 5:2 compressors where multiplicand multiplier are 8-bits wide.

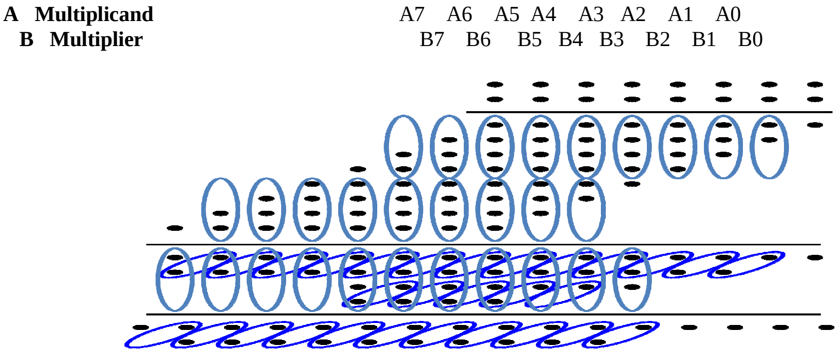


Figure 4. 8 x 8 partial product Array reductions.

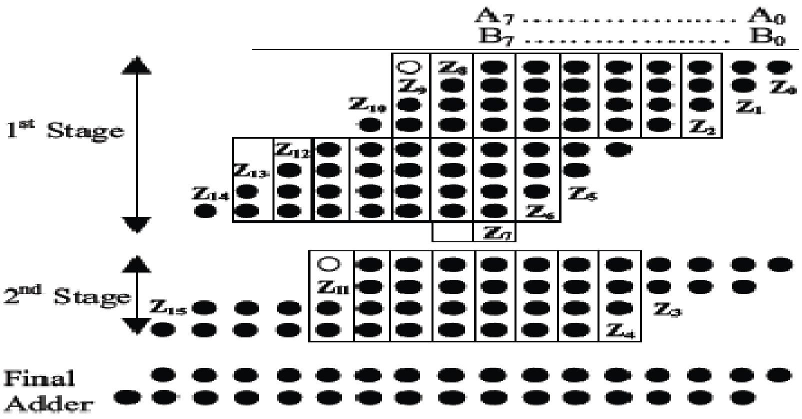


Figure 5. Data bit distribution in the proposed Merged MAC unit.

Figure 5, shows logic used for data bit distribution in the proposed MAC unit [2]. Figure 6, shows that while implementation of partial products addition how the data bit distribution in proposed MAC architecture takes place; the partial product addition is done in to two different stages. The first stage is addition of partial products using half adder, full adder and 4; 2 compressors, similarly at the second stage of partial product addition 4; 2 compressor and 5:2 compressors are used.

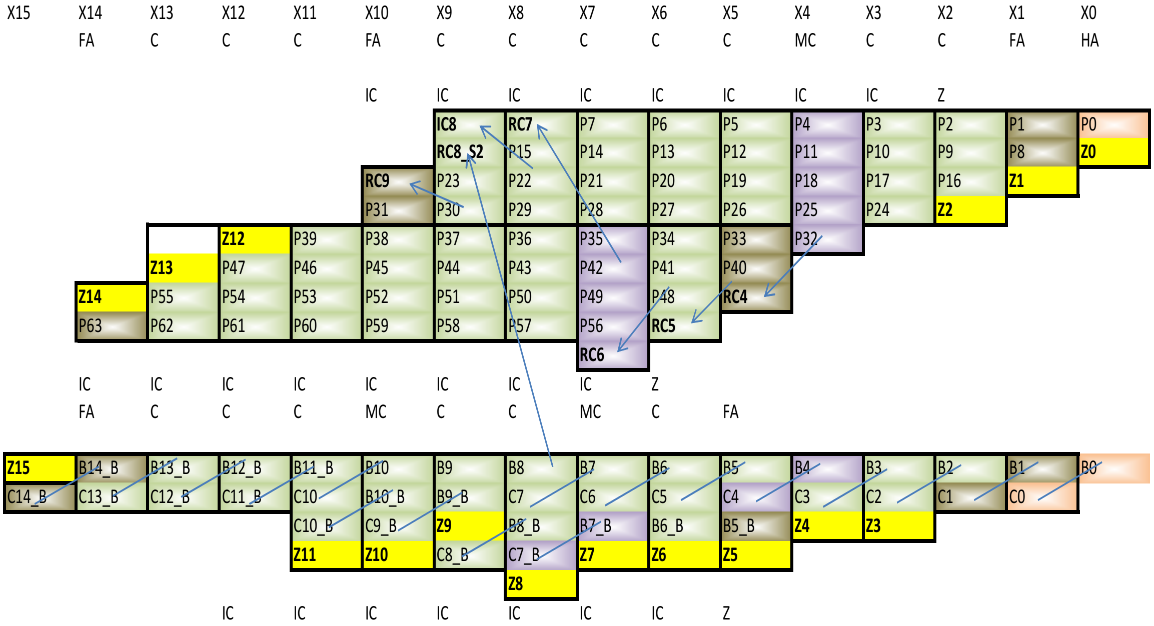
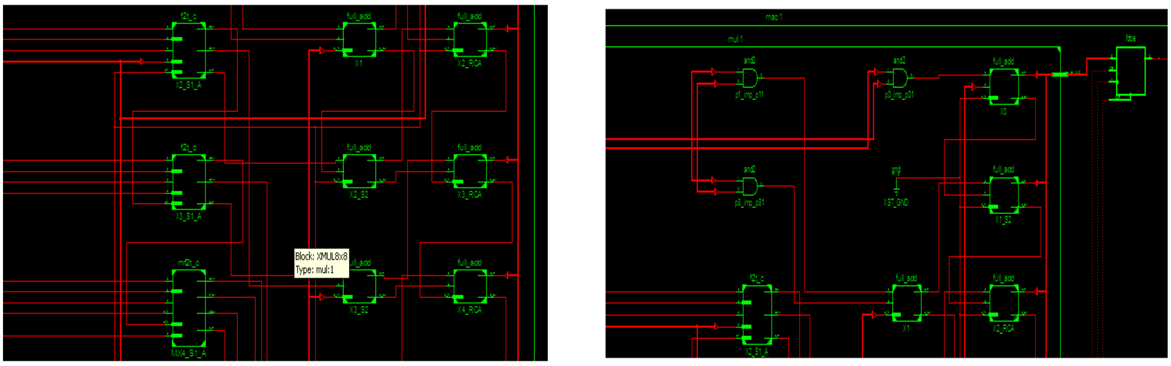

X15 X14 X13 X12 X11 X10 X9 X8 X7 X6 X5 X4 X3 X2 X1 X0
FA C C C FA C C C C C MC C C FA HA
IC IC IC IC IC IC IC IC Z
IC8 RC7 P7 P6 P5 P4 P3 P2 P1 P0
RC8_S2 P15 P14 P13 P12 P11 P10 P9 P8 Z0
RC9 P23 P22 P21 P20 P19 P18 P17 P16 Z1
P31 P30 P29 P28 P27 P26 P25 P24 Z2
Z12 P39 P38 P37 P36 P35 P34 P33 P32
Z13 P47 P46 P45 P44 P43 P42 P41 P40
Z14 P55 P54 P53 P52 P51 P50 P49 P48 RC4
P63 P62 P61 P60 P59 P58 P57 P56 RC5
RC6
IC IC IC IC IC IC IC IC Z
FA C C C MC C C MC C FA
Z15 B14_B B13_B B12_B B11_B B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
C14_B C13_B C12_B C11_B C10 B10_B B9_B C7 C6 C5 C4 C3 C2 C1 C0
C10_B C9_B Z9 B8_B B7_B B6_B B5_B Z4 Z3
Z11 Z10 C8_B C7_B Z7 Z6 Z5
Z8
IC IC IC IC IC IC IC Z

Figure 6. Data bit distribution in the implemented Merged MAC unit

Figure 6, shows the data bit distribution of implemented merged MAC architecture and types of circuitries required at each bit weight positions of partial products. Arrowhead shows carry propagation through summation circuit (4:2 compressors).Two types of carry propagation is takes place: 1] Horizontal carries propagation. (**i-1 column**) to (**i+1**). 2] Vertical carries used for addition whenever data bits is exceed than 4.In this (**Cin** is independent of **Cout**). Notations: **HA**- Half Adder, **FA**-Full Adder, **C**- 4:2 Compressors and **MC**- Modified 4:2 Compressors

2.3 Final adder

Final adder required large size because of addition of 2n+1 bit, which is main critical path in the depicted MAC architecture. This is a bottleneck in implementing the adder, since it requires large time delay for propagation carry from one stage to another reduction of this delay is measure task of implementation. Many researchers has implemented several techniques for reduction of propagation delay using CPA carry propagation, CSA carry save adder tree which is having some advantages and disadvantages over each other [1,3].



a

b

Figure 7. a) RTL schematic of Partial product addition using 4:2 and 5:2 compressors. b) RTL schematic of addition of SUM and Carry using full adders.

The objective of our work is, by using higher compressor minimize the propagation delay of the gates which is also called ripple carry propagation of the gates at the summation stage. Using compressor circuits as the basic cell to construct the addition tree for partial products reduction in parallel multipliers has a great advantage since they introduce parallelism in adding the available operands instead of having them added sequentially which gives them a speed advantage. Figure 7 (a) shows, RTL schematic of addition of partial products using higher compressor called as partial product reduction tree.

The result of it is, in the form of SUM and Carry which accumulated in to the free lines of 4:2 and 5:2 compressor causes high speed addition operation as compared to the CSA carry save adder tree as shown in Figure 7(b), RTL schematic of CSA.

3. FPGA IMPLEMENTATION

In order to attain high speed using conventional FPGA, bit level parallelism is exploited. The overall filter architecture is shown in figure 8 where operands are stored in appropriate synchronous register banks. The implementation of the programmable multiplexed filter is based on data flow architecture using only one multiplier and adder (MAC) unit for arithmetic operations. The adder is required to resolve the carries that are generated and propagated through the pipeline. Figure 10, shows RTL schematic of, input data sequence **X [K]** and sample data sequence **h (n-k)** for MAC input port A [7:0], Port B [7:0].

The full adders, 4:2 compressors, and 5:2 compressors are necessary for partial product coefficients that are a sum of two unsigned power of two are implemented as two block states of n: 2 compressor reduction tree called as binary tree structure, whose inputs are configured with the appropriate shift for the given coefficients. The sum and carry signals from the carry signals from the full adders are pipelined using a carry-save addition (CSA) techniques in order to increase sampling rate and alleviate potential routing delays in the target FPGA device.

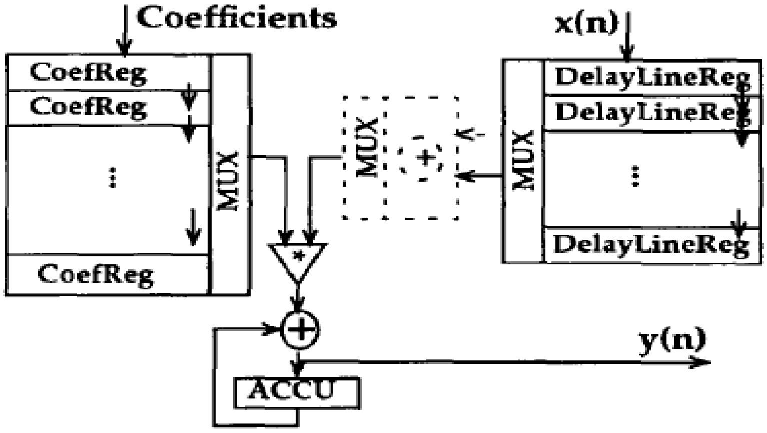


Figure 8. Multiplexed data flow FIR filter using one MAC.

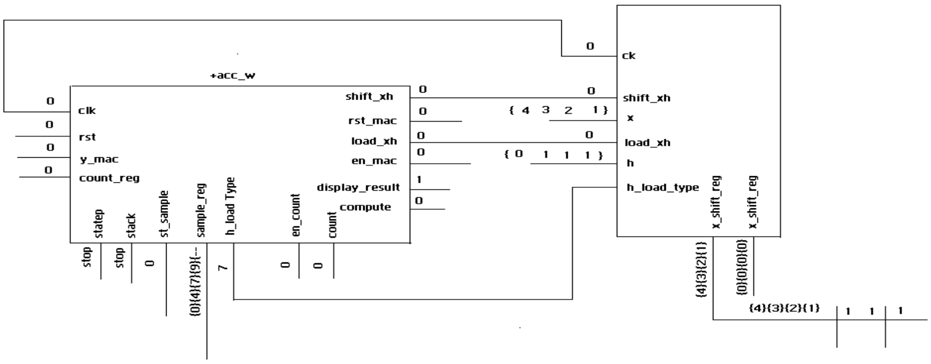


Figure 9. Data flow in FIR through merged MAC.

The input data bus passes through the bit slice array to provide short interconnection distances to the first row of 4:2 and full adders. The hardware requirements for tap with $\mathbf{B_d}$ input data path bits and $\mathbf{B_i}$ intermediate accumulation path bits then $\mathbf{2B_i}$ full adder and minimum of $\mathbf{2B_i}$ flip flops. Figure 9 shows data flow in FIR through merged MAC.

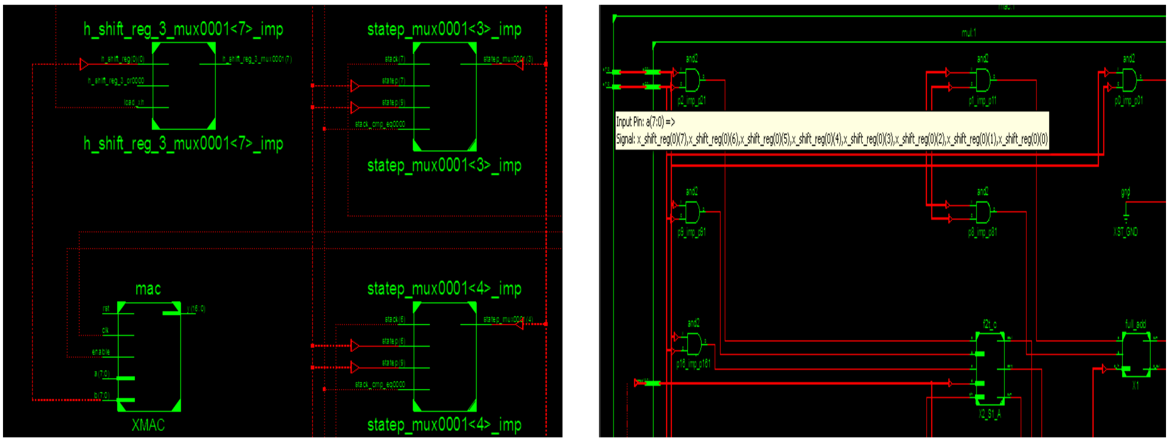


Figure 10. RTL schematic of data sequence to MAC.

The architecture proposed here is well suited to FPGA implementation only minor modifications need to be considered to map it to the array. The filter taps of FIR is implemented in array columns of Xilinx XC3S400 Spartan FPGA device. The architecture of an FIR filter tap with two powers of two coefficients is identical that shown in Figure 7(b) pipelined FIR structure. Each of the bit slices for the tap requires two combinational logic blocks (CLBs) in the array of 18x12 is required for implementation that is 18 rows and 12 columns. The extensive local routing capability of typical FPGAs can be used for majority of signals within and between the taps. Figure 11 illustrates the local routing required between CLBs, where column “1” maps to the first set of 4:2 compressors and column “2” maps to the second set of full adders. The globally routed input data signals are distributed using the horizontal and vertical nets running the length and width of the chip between the rows and columns of CLBs.

The primary concern is with routing of shift lines. In most accumulation realization, the accumulation path will have a wider word width than the input data from shifter, in order to account for the overflow and round off problems that are inherent in a design of this type. A tap with input data bits of width B d and Bi accumulation path bits can thus be implemented using 2Bi logic blocks. The final adder required by the filter can be implemented on FPGA. Typical FIR convolution operation have been implemented on a XC3S400 using this architecture. XC3S400 has an array of CLBs. An input data word size is 8-bits was used, all of the column were required for the fifteen taps. The maximum sampling rate for this particular design is 66.91 MHz.

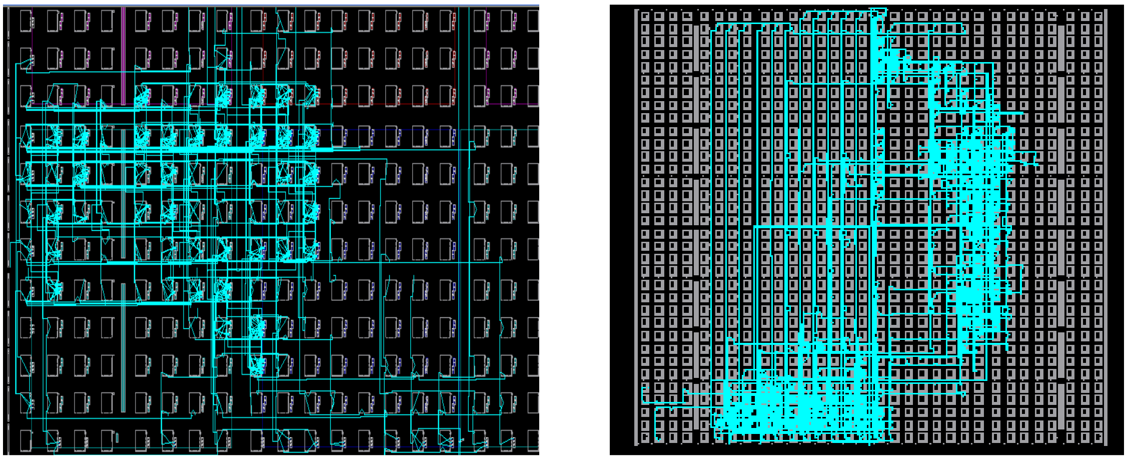


Figure 11. FPGA Filter Tap local Interconnection

The Fir filter structure used for implementation is the application of input data pipelining to reduce global signal distribution delays. Some details of implementation are given in table 1. For our Realization of the 8 by 8-bit multiplier and adder by using merged MAC architecture 17-bit accumulator requires 01 half and 06 full adders, 16 (4:2) compressors, 03 modified compressors that is (5:2) compressors used to speed up the multiplication process and 28 flip flops required. The configuration logic blocks (CLBs) required is 216 and its operating speed is about 95.76 MHz. The delay is determined by the combinational delay of an AND gate and a full adder which is for FPGAs typically 3.98 nsec.

Table 1. FIR Filter using XC3S400 FPGA

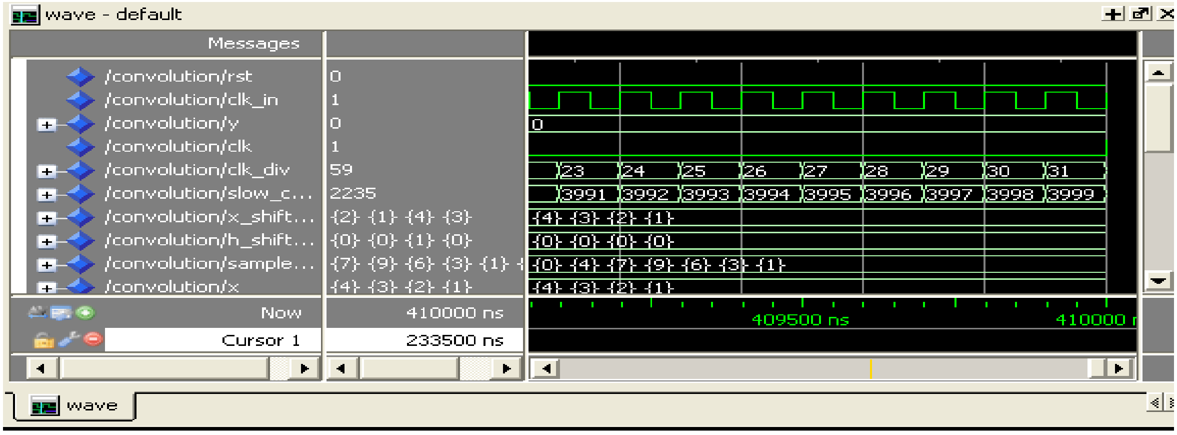
Specifications	Value
Input data	16-bits
Coefficient bits	16-bits
Intermediate data	16 bits
Filter Length	15 taps
Maximum sampling rate	66.91 MHz

4. RESULTS AND DISCUSSION

The described FIR filter model architecture shown in Figure 1 has been implemented in VHDL. The model supports generic parameters so that word length and filter length can be adjusted to the application requirements. Here, a signal word length of 16 bits, coefficient word length of 16 bits and a filter depth of 15 taps required. The VHDL code has been synthesized using Xilinx Foundation 11.1Series software and the experimental test of the proposed digital FIR filter is done using Spartan-3 (XC3S400 PQ-208).Figure 11 shows the physical area utilized by the spatan device which is 89%. Timing simulation shows that the maximum operating speed of the multiplier is 66.91MHz.

4.1 Simulation Results

In order to compare between the merged architecture and parallel architecture the digital FIR filter is implemented on a Field Programmable Gate Array (FPGA).Two major CAD software tools were used; Mentor Graphics and Xilinx 11.1 ISE tools. ModelSim is used for simulation; Fig.12 shows the simulation result of the proposed digital FIR filter using merged MAC unit. The simulation result shows output of the proposed design with input sequence x (n) and impulse response h (n) will give convolution output y (n). For example consider input sequence x (n) =1, 2, 3, 4 and impulse response h (n) =1, 1, 1, 0 then after convolution will get output y (n) = 1,3,6,9,7,4,0 shown in Fig 5.



```
/convolution/rst             0
/convolution/clk_in          1
/convolution/y               0
/convolution/clk             1
/convolution/clk_div         59
/convolution/slow_c...       2235
/convolution/x_shift...      {2} {1} {4} {3}
/convolution/h_shift...      {0} {0} {1} {0}
/convolution/sample...       {7} {9} {6} {3} {1} {
/convolution/x               {4} {3} {2} {1}
Now                          410000 ns
Cursor 1                     233500 ns
```

Figure 12. Simulation result of digital FIR filter.

4.2 Experimental Results

The case of digital FIR filter unit is considered as a case of study for experimental test. Xilinx 11.1 ISE tool is used to synthesize the VHDL source code and Modelsim mentor graphics 6.4c tool is used for simulation. The bit map file is generated by Xilinx 11.1 ISE tools in which the bit stream is ready to be downloaded on the FPGA. The MX3MB0207-003-IM (Spartan kit) is used for testing. Figure 13 shows the test bed for the digital FIR filter, the inputs are internally forced and the outputs are shown on external LEDs as shown in Figure 13.

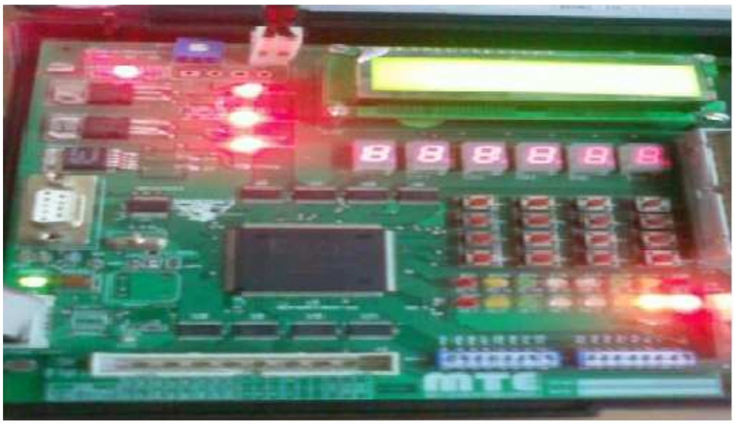


Figure 13. Test bed for digital FIR filter.

5. CONCLUSION

A new parallel FIR digital filter structure which is suitable for efficient implementation of filters whose coefficient values are sums of power of two terms were presented. By exploiting this architecture using higher compressor for computation addition operation, the constraints on the coefficient values, this architecture yields extremely efficient and high speed programmable and custom implementations. It has been shown that merged multipliers provide the best trade-off between speed and resource requirements. The synthesis results shows that the merged multiplier will be useful for high sampling rate applications > 1 MHZ. The experimental test shows that the results have been validated.

References

[1] A. Abedelgwad, “High speed and area efficient multiply Accumulate (MAC) Unit for Digital signal Processing applications,” *IEEE International Symposium on Circuits and Systems*, ISCAS 2007.

[2] Ayaman.A.Fayed, “A merged Multiplier Accumulator for High Speed signal processing Applications,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* 2002.

[3] P.Kolling, K.Abbot, “FPGA Implementation of High Performance FIR filter,” *IEEE International Symposium on Circuits and Systems,* ISCAS Proceedings of 1997.

[4] Ondrej Pirochta, “Hardware implementations of Digital fir filters in FPGA,” *Brno University of Technology Purkynova*

[5] Vesa savela, “Utilization of VHDL based ASIC –Reliable digital filter architecture library in DSP System design,”

[6] P.asadee, “A High-Speed Multiplication Algorithm Using Modified Partial Product Reduction Tree.

[7] Wayne Wolf, "Modern VLSI Designer", Prentice-Hall, Upper Saddle River, 2002.

[8] Douglas Perry, “VHDL programming by examples” McGraw-Hill Publication.

BIOGRAPHIES OF AUTHORS

Mr.Sachin.B.Jadhav has completed his B.E in Electronics & Telecommunication from Shivaji university kolhapur and M.E in Electronics from Shivaji university kolhapur.Currently he his working as an Assistant Professor in Department of Electronics & Telecommunication Engineering at Bharati vidyapeeth College of Engineering, Kolhapur. He has keen interest in the field of embedded system design and VLSI Design.

Mr. Nikhil Niwas Mane has completed M.Tech. Degree from Walchand College of Engineering, Sangli, Maharashtra, India in Electronics. He has 2 years industrial work experience. Presently he is an Assistant Professor in Bharati Vidyapeeth’s College of Engineering, Kolhapur. He has keen interest in the field of power electronics, embedded systems and VLSI Design.