International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 10 (2017) pp. 2209-2220
© Research India Publications. http://www.ripublication.com

An Efficient Multichannel FIR Filter Architecture for FPGA and ASIC Realizations

J.Britto Pari[1] and D.Vaithiyanathan[2]

[1]Assistant Professor, [2]Teaching Fellow

[1]Department of Electronics and Communication Engineering, Sri Sairam Engineering College, Chennai, India.

[2]Department of Electronics and Communication Engineering, College of Engineering Guindy, Anna University, Chennai, India.

Abstract

In this paper, a Time division multiplexing (TDM) based multichannel FIR filter architecture is proposed using a single multiplier and adder irrespective of number of channels and taps using the concept of resource sharing principle. For efficient optimization of the resources Output Product Coding (OPC) and Dual port schematic is used, which are based on Look-Up-Table (LUT). The proposed 16-tap multichannel architecture is implemented using Verilog Hardware Description Language (HDL) and synthesized in Xilinx Vertex Field Programmable Gate Array (FPGA). The results obtained from the single channel FIR filter architecture, the frequency of the system supports up to 480 MHz with reduced area. The cell level performance is also obtained using Cadence RC compiler with TSMC 180 nm CMOS technology.

Keywords: Multichannel FIR filter, Look-up Table, Output Product Coding (OPC)

INTRODUCTION

Digital Signal Processing (DSP) opens the gateway of solutions for many complex real time applications in various fields like wireless communication, speech processing, multimedia compression which is comprised of voluminous data. With the advancement of recent DSP technology, large sampling array size is mandatory for variety of applications such as communication and multimedia, in which the information from single channel may be erroneous and time consuming and hence multi-channel signal processing is advisable for reliable and efficient processing of these signals. The sampled multi-channel data are processed effectively in FIR filter through time multiplexed mechanism to accomplish resource optimization [1-2].

Recently with the advent of Software Defined Radio (SDR), the research has been focused more on reconfigurable realization of FIR filters mainly due to the necessity of high flexibility and low complexity [2-19]. The digit-based reconfigurable architecture presented in [3] provides a flexible and low power solution with a wide range of precision and variable tap length of FIR filters. Conventionally, the reconfigurable FIR filters are designed based on programmable Multiply-Accumulate [MAC] architecture [6], systolic architecture [7] and Programmable Shift Method (PSM) [5]. The performances of various architectures are analyzed in terms of hardware complexity, power consumption and throughput. The programmable MAC architectures consume low power with reduced supply voltage, but it requires large area [6]. Even though systolic based architecture reduces the complexity, it increases the latency when the order of the filter gets increases [7]. The PSM based reconfigurable architecture provides less complexity and significant speed due to the presence of programmable shifters [5] but the area increases with the rise in number of filter taps. Therefore many researchers are concentrated on FIR filter architecture in order to reduce the hardware and power consumption.

The direct implementation of N-tap FIR filter requires N number of MAC operations, which are too expensive to implement in hardware due to its logic complexity and area constraint. Therefore an architecture has to be designed which overcomes the above constraints. In this paper, a multi-channel FIR filter is implemented through Time Division Multiplexing(TDM) mechanism using single multiplier and adder irrespective number of taps and number of channels by applying the principle of resource sharing and increasing the filter operating frequency. For optimizing the resource complexity of multiplier, the two schemes, namely output product coding (OPC) and Dual port schematic architectures are implemented. The main advantages of OPC and Dual port schematic architecture schemes are to minimize design complexity by manipulating the odd multiples of the fixed coefficient in LUT design. The performance is analyzed in terms of area and speed by varying the number of taps.

The rest of the paper is organized as follows. In section 2, Single Channel FIR filter is described. The section 3 deals with the architecture design and implementation of multi-channel FIR filter. The performance of the design is analyzed and discussed in Section 4. Finally section 5 concludes the paper in brief.

EFFICIENT SINGLE CHANNEL FIR FILTER ARCHITECTURE

The structure of the FIR filter has the multipliers in the form of MAC structure and delay blocks as the main building blocks. The performance of the FIR filter entirely depends

upon the speed of multiplier which determines the critical path in the filter structure.

Let X(n) and Y(n) be the input and output sequences of the FIR filter respectively. The general form of an N-tap FIR filter can be formulated as

$$Y(n) = \sum_{k=0}^{N-1} h_k X(n-k) \qquad (1)$$

where h_k is the k_{th} coefficient of the filter impulse response.

The complexity of the FIR filter is dominated by multipliers. The increase in the number of taps increases the number of multipliers. In order to overcome these difficulties, an efficient architecture is proposed in which a single multiplier and adder is used to operate the entire filter, by increasing the output sampling rate of the FIR filter. The N-tap FIR filter requires N clock cycles are required to complete the filter operation. Suppose For a 4- tap FIR filter, the input sampling rate is 1 Mega samples per second (MSPS), the sampling rate of output filter is increased to 4 MSPS. For each clock cycle, data is injected through delay blocks to the multiplier to do filter operation and the output is obtained within four clock cycles. The proposed architecture is shown in Figure 1. The Multiplexer is used to select the data across the registers and perform the multiplier operation. A 256x8 bits RAM is used to store the filter coefficients (which are used to reduce the switching activities instead of storing the coefficients in registers) of which only 4x8 bits is used for 4-tap FIR filter. Accumulator block is used to accumulate the multiplier outputs and is set to zero after 4 clock cycles. Multiplexer select lines, coefficient memory address array and accumulator operation are selected by only one generic counter. Similarly N number of taps can be implemented using single multiplier and adder by inserting registers which increases the operating frequency of FIR filter. Hence an 8-tap FIR filter takes 8 MSPS with the sampling rate of 1MSPS.Hence for a N-tap FIR filter,N clock cycles are required to obtain the filter output.

The performance of the FIR filter is further enhanced by introducing OPC and Dual port memory based architecture in the Multichannel FIR filter, which reduces the complexity and critical path and is discussed below.

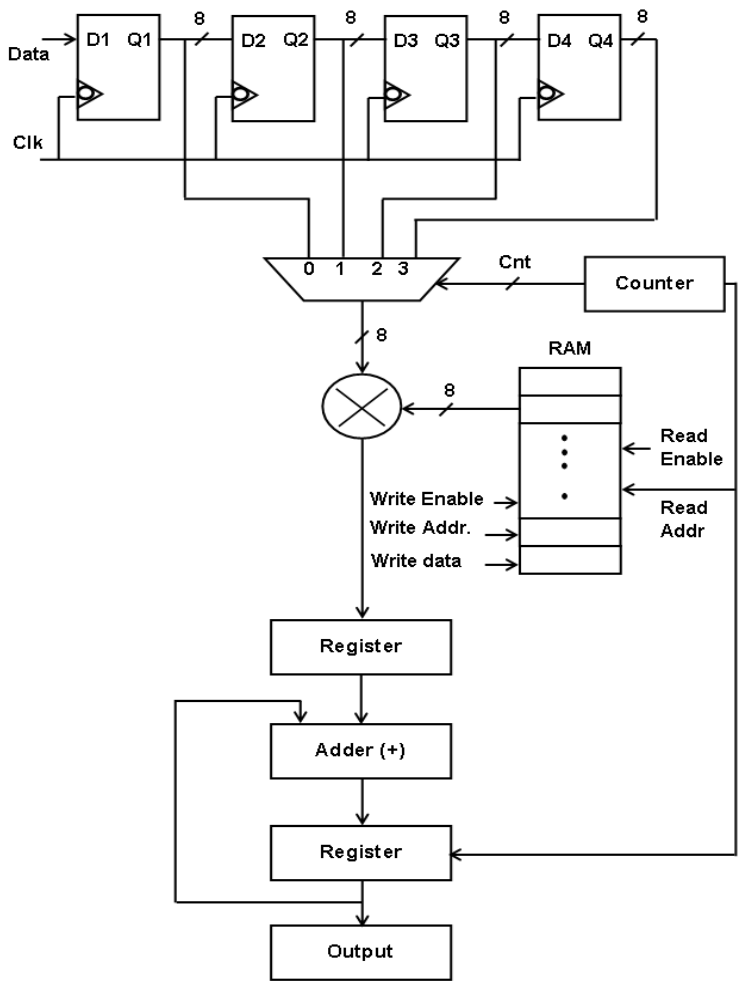

Figure 1: Proposed single channel FIR filter

OPC Scheme

The odd multiple storage scheme proposed by [4] stores the product of odd multiple of co-efficient and the input. Even multiples of co-efficient are obtained by left shifting the odd multiples. For any input vector of L bits the number of locations in the LUT is identified by 2^L. However, the odd multiple storage scheme reduces the number of locations by a factor of 2. But this method operates when the input is an unsigned number. So, this method fails in case of both negative and positive samples of the signal exist. This disadvantage can be overcome by OPC scheme proposed by [4]. The method of OPC considers both the positive and negative samples.

Let us assume the input and the co-efficient to be in L-bit sign magnitude format. The most significant bit of both input and co-efficient corresponds to the sign value while the remaining (L-1) bits correspond to the magnitude. In order to obtain the complete product value in the sign magnitude representation, the magnitude and the sign bits are processed separately and finally appended. Table 1 shows a tabular representation of the LUT needed for OPC scheme with the input length of 8 bits wherein the most significant bit (MSB) corresponds to the sign bit and the remaining 7 bits corresponds to the magnitude.

In Table 1, the 7-bit magnitude is converted to the 6-bit address of LUT. Hence only 64 product values are stored in LUT instead of 128. The OPC representation in table indicates how the final output from LUT is obtained. Let the product values on the ith and the $(128+2-i)^{th}$ row be u and v respectively. The relation between 'u' and 'v' is given as u= [$\frac{u+v}{2} - \frac{v-u}{2}$] and v=[$\frac{u+v}{2} + \frac{v-u}{2}$] where (u+v) is the constant value, 128A. It is observed that the address value of ith row is the two's complement of that on $(128+2-i)^{th}$ row for $2 \leq i \leq 64$. The sum of product values on these two rows is 64A. Hence the table shows that three considerations namely, a constant value, value accessed from the look up table and also whether an addition or a subtraction is to be performed. The constant value is nothing but the value obtained after the computation of (u+v) which is given by 128A. The value accessed from the memory array depends upon the magnitude portion of the input vector, i.e., for an input of L bits, only (L-2) bits are considered for memory access.

Figure 2 depicts the architecture for computing the inner product using the method of OPC for the input length of 8-bits (inclusive of the sign bit) and is depicted as X_7-X_0 and the co-efficient to be represented as A_7-A_0, where A_7 and X_7 correspond to the sign bits of the co-efficient and the input respectively.

Table 1: Look up table for OPC scheme

ADDRESS $X_6X_5X_4X_3X_2X_1X_0$	PRODUCT VALUE	OPC REPRESENTAION	STORED VALUE IN LUT
0000000	0	64A-64A	64A
0000001	A	64A-63A	63A
0000010	2A	64A-62A	62A
0000011	3A	64A-61A	61A
0000100	4A	64A-60A	60A
. . .	. . .	. . .	. . .
0111100	60A	64A-4A	4A
0111101	61A	64A-3A	3A
0111110	62A	64A-2A	2A
0111111	63A	64A-A	A
01000000	64A	64A-0	-
01000001	65A	64A+A	-
01000010	66A	64A+2A	-
. . .	. . .	. . .	. . .
1111100	124A	64A+60A	-
1111101	125A	64A+61A	-
1111110	126A	64A+62A	-
1111111	127A	64A+63A	-

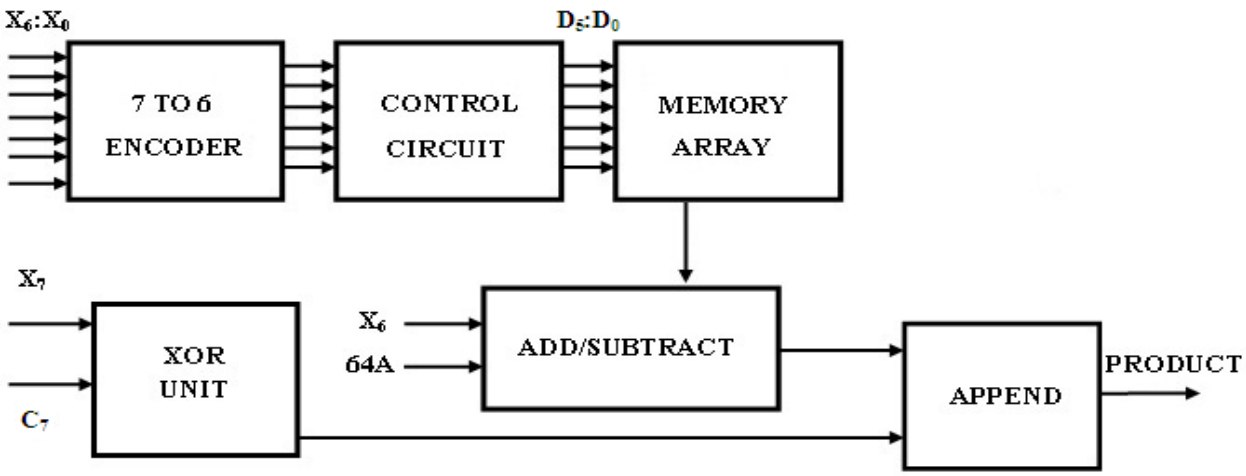

Figure 2: Proposed architecture for OPC scheme

The first block under discussion is the encoder that converts the 7-bit magnitude portion of the input (X_6: X0) into 6-bit value which is processed to formulate the address values of D_5:D_0 of memory array. The conversion is controlled by a control circuit using logical relations expressed in the equations (2a) to (2f).

$$D_5 = \overline{X_6}.X_5 \quad (2a)$$

$$D_4 = \overline{X_6}.X_4 \quad (2b)$$

$$D_3 = \overline{X_6}.X_3 \quad (2c)$$

$$D_2 = \overline{X_6}.X_2 \quad (2d)$$

$$D_1 = \overline{X_6}.X_1 \quad (2e)$$

$$D_0 = \overline{X_6}.X_0 \quad (2f)$$

This becomes the final address to access the value from the memory array consisting of stored value in the LUT as tabulated in Table 1. The next unit to be considered is the add/subtract unit which decides whether to add or subtract the value accessed from the memory array with the constant value of $\frac{u+v}{2}$ based on the X_6^{th} bit. Since the input is represented in sign magnitude form, the final product should also be represented using the sign magnitude form. The sign bit of the final product is computed by performing the XOR operation between MSB (sign bit) of both input as well as the co-efficient. The corresponding logical relation is given by $s = X_7 \oplus A_7$. Finally, the process of concatenation is done to obtain the final product. Therefore, for an input of L-bits, this method makes use of 2^{L-2} memory locations thereby reducing the number of memory locations further.

Dual port Memory based LUT Multiplier

Consider a L-bit unsigned binary input X multiplied with a fixed coefficient A. The conventional implementation of memory-based multiplication requires memory unit of 2^L words. The dual port memory based LUT multiplier proposed by [4] shows that ($2^{L/2-1}$) words are adequate to the odd multiples of A in the LUT. One of the possible product words is zero, while the rest of ($2^{L-1} - 1$) words are even multiples of A, derived by left-shift operations from odd multiples of A. Address correspond to (0000) can be obtained by resetting the LUT output. The approach behind memory based multiplication is given in Table 2.

This multiplier consists of memory with the size of eight words of (w+4) bit width and a 3-to-8 line address decoder, a NOR-cell, a shifter, a 4-to-3 bit encoder to map the 4-bit input operand to 3-bit LUT-address and a control circuit for generating the control word and RESET signal for the shifter and the NOR-cell respectively.

The 8-bit input binary number [x_7,x_6,x_5,……x_0] is split into two
4-bit numbers and these are given to two separate 4 to 3-bit encoder which produces two three address bits [d_2 d_1 d_0] and [d_5 d_4 d_3] for dual port memory as per the relation.

$$d_0 = \overline{(\overline{x_0.x_1}).(\overline{x_1.x_2}).(x_0 + \overline{x_2 + x_3})} \quad (3a)$$

$$d_1 = \overline{(\overline{x_0.x_2}).(x_0 + \overline{(x_1 + x_3)})} \quad (3b)$$

$$d_2 = x_0.x_3 \quad (3c)$$

Similar equations are evaluated for the address bits d_5 d_4 d_3 from x_4, x_5, x_6 and x_7. These generated address bits are fed to two different decoders that convert them into eight word-select signals {w_i, $0 \le i \le 7$}. Hence the dual port memory can be accessed through two ports (decoder) of eight word select signals each. The even multiples are obtained by doing shifts on the memory output, since the values from memory are

multiples of bit-inverted co-efficient. Three signals s_{00}, s_{01} and RESET are generated according to the relation.

$$s_{00} = \overline{(x_0 + \overline{(x_1 + \overline{x_2})})} \quad (4a)$$

$$s_{01} = \overline{(x_0 + x_1)} \quad (4b)$$

$$RESET = \overline{(x_0 + x_1)}.\overline{(x_2 + x_3)}$$

$$RESET = \overline{(x0 + x1)} \quad (4c)$$

When the input combinations are (0010), (0110), (1010) and (1110), the output of LUT is shifted once. Inputs of (0100) and (1100) requires two shifts, whereas (1000) requires three shifts. If the input word is (0000), the reset operation is performed by setting the RESET bit as given in the logical relation (4c). The schematic representation of dual port memory based LUT multiplier is shown in Fig.3. It is obtained by two 4-bit multipliers along with a shift-adder. The most significant half of the input is left-shifted by four bits and added with the outputs of other shifter.

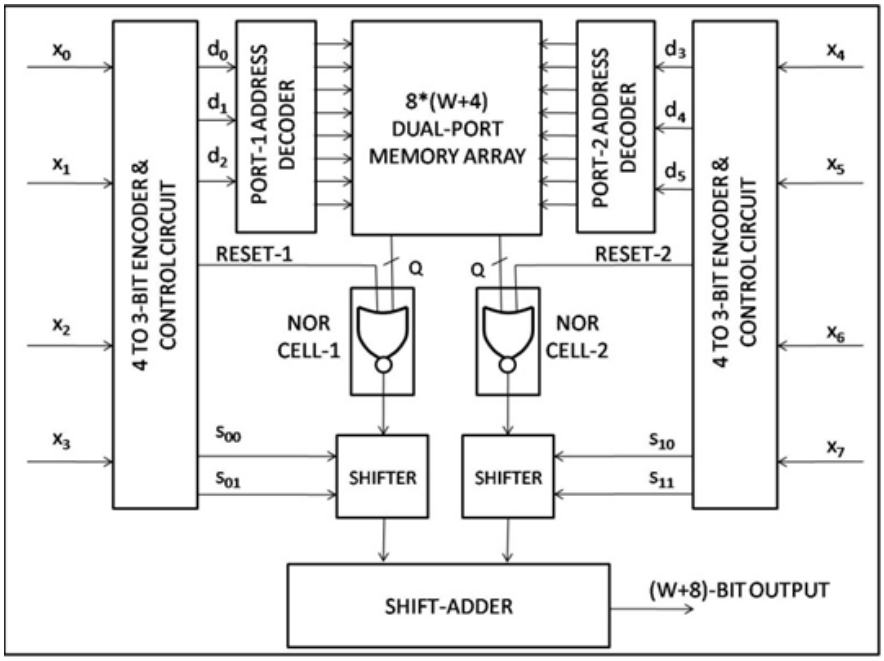

Figure 3: Dual port Memory based LUT Multiplier

Table 2: LUT Input and Product Values for Word Length L=4

Address $d_2d_1d_0$	Stored value	Input $x_3x_2x_1x_0$	Product value	No. of shifts	Control s_1s_0
000	A	0001	A	0	00
		0010	2^1 x A	1	01
		0100	2^2 x A	2	10
		1000	2^3 x A	3	11
001	3A	0011	3A	0	00
		0110	2^1 x 3A	1	01
		1100	2^2 x 3A	2	10
010	5A	0101	5A	0	00
		1010	2^1 x 5A	1	01
011	7A	0111	7A	0	00
		1110	2^1 x 7A	1	01
100	9A	1001	9A	0	00
101	11A	1011	11A	0	00
110	13A	1101	13A	0	00
111	15A	1111	15A	0	00

PROPOSED MULTICHANNEL FIR FILTER

With the development of digital signal processing technology, the sampling array size is becoming increasingly large in a variety of communication applications, such as acoustic and Software Defined Radio (SDR). Therefore multichannel signal processing is essential as far as reliability and efficient processing of SDR technology is concerned. For optimizing the logic resources a much more hardware efficient implementation would be time division multiplexing with single set of hardware resources across the data streams of several different channels [1-2].

In multichannel FIR filter structure shares the logic resources between multiple sample streams through time division mechanism for the efficient utilization of the hardware resources. The conventional TDM based multichannel FIR filter is discussed in [2]. For M channel N-tap FIR filter, N multipliers and N-1 adders are required and if the sampling frequency for the one channel is f_s, then for a M-channel filter, processing M sample streams require a sampling frequency of f_s/M. Consider a conventional 2-channel, 4-tap FIR filter as shown in Figure 4 having channel 1 and 2 data fetched simultaneously. Each channel1 and channel2 data are injected to the 2:1 Mux. The multiplexed single channel data are multiplied with coefficients which are stored in the registers. The products are added to get the filter output. For this filter computation 4-multipliers and 3 adders are required. Increasing the number of channels and taps, the complexity increases and power consumption is also increased.

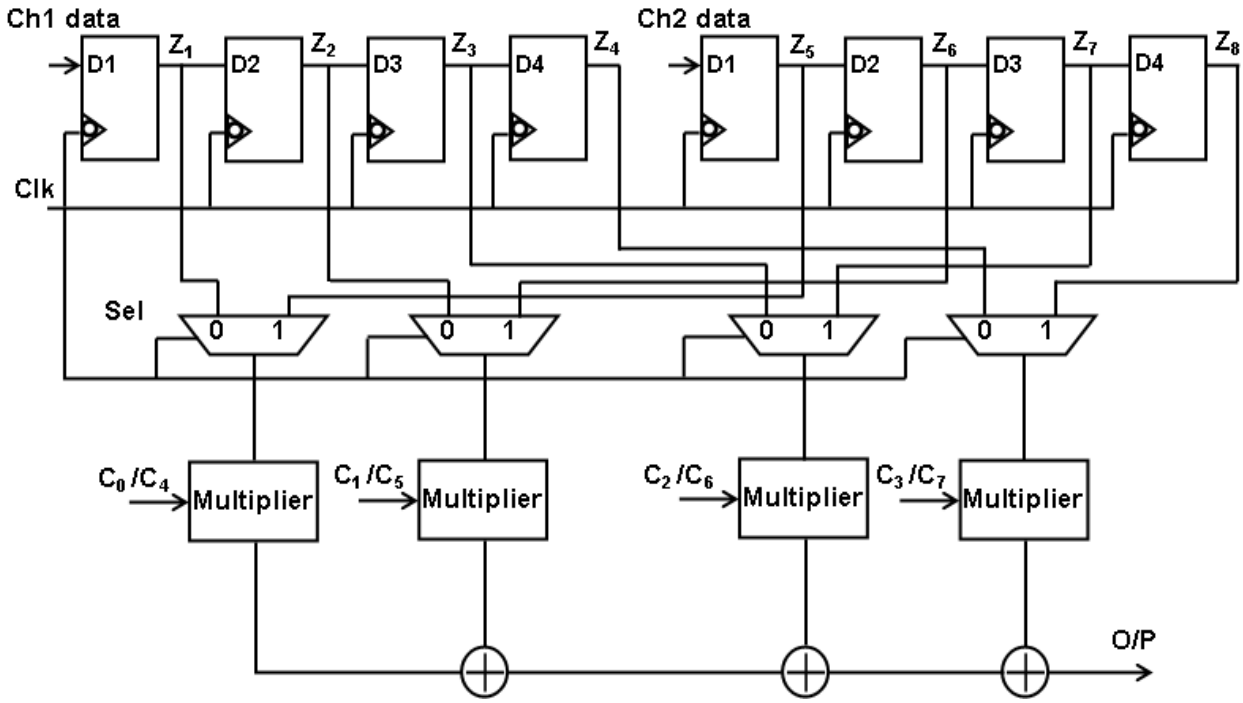

Figure 4 Conventional structure of two channel TDM based 4-tap FIR filter

In order to overcome these difficulties of conventional structure, the multichannel FIR filter architecture is proposed as shown in Figure 5. The proposed multichannel FIR filter architecture is implemented using single multiplier and adder irrespective of number of taps and channels. The filter operating frequency is increased by inserting Pipelined registers. For example M channel N-taps FIR filter requires (M * N) clock cycles for filter computation, where N clock cycles are required to implement each of the M channel N tap FIR filter. The proposed architecture are needed for a 2 channel 4 tap FIR filter is shown in Figure 5. Here 8 clock cycles for filter computation, of which the four clock cycles are required to implement each of the 2 channel 4 tap FIR filter. Initially channel 1 and channel 2 data are fetched simultaneously to the delay registers. For the first 4 clock cycles channel 1 data are injected to the multiplier to do filter operation and the output is obtained within 4 clock cycles. The 8:1 multiplexer is used to select the data across the 2 channels of 4 lines each. The multiplexed single channel data are multiplied and accumulated in the register. The next set of data from channel 2 is fetched and computed in the next 4 clock cycles. The MAC is then reset to accommodate the next set of channel data. A generic RAM memory of 256x8 is used to store the filter coefficients of which only 8x8 bits is used as determined by the number of channels and taps. The coefficients are stored in the RAM as 2 sets of 4 locations after the write enable is set, and the other locations of the RAM are set to 0 on resetting the write enable. A single generic counter is used to access the multiplexor select lines, coefficient memory address array and accumulator operation. Similarly for 8 tap 2 channel FIR filter, 16 clock cycles are required. Hence N number of taps with M number of channels uses single multiplier and adder, which can be implemented by increasing output operating frequency of FIR filter through inserting pipelined registers.

For example, consider a 2 channel 8 tap FIR filter:

The input sampling rate of fir filter =1MSPS

Output sampling rate of fir filter =16MSPS

TDM = output sampling rate/ input sampling rate

= 16/1 = 16

Let number of taps = T;

The chances of required clock cycles = 2*T

Maximum number of channels = TDM/T

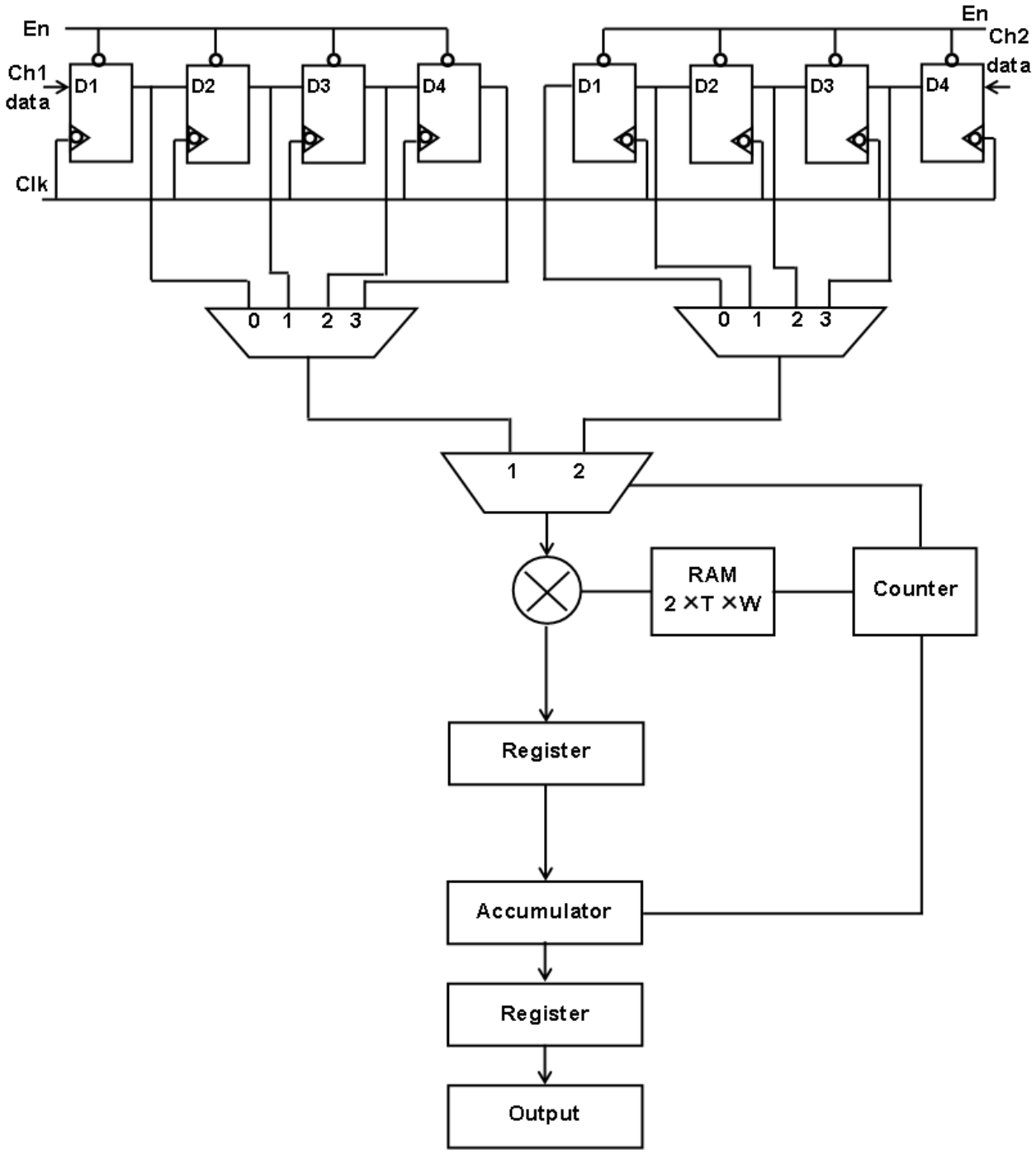

Figure 5: Proposed two channel TDM based 4-tap FIR filter

RESULTS AND DISCUSSION

The proposed architectures are designed by parameterizable Verilog cores. A key advantage of hardware description languages (HDLs) is that all the statements are executed in concurrent manner. The performances of both single channel and multichannel architectures were synthesized on Xilinx VIRTEX -5 FPGA Device and Cadence RC compiler.

Single channel FIR filter

The proposed FIR filter architectures are implemented using two multiplier schemes, namely, (i) OPC scheme and (ii) dual port memory based LUT Multiplier, and synthesized using Xilinx VIRTEX -5 XC5VSX95T-1FF1136 FPGA device. The performance results of FIR filter architectures are analyzed in Table 3. The proposed single channel FIR filter is implemented using single MAC core architecture with the concept of time sharing mechanism irrespective of number of taps. For higher number of filter taps, area is increased marginally due to the increasing complexity of multiplexer logic and registers. Further reduction in the complexity of the multiplier is achieved by two methods- Dual port memory and OPC based multiplier.

Table 3: Performance results of single channel FIR filter using XILINX VIRTEX -5 FPGA Device

Performance measures	Dual port Memory based LUT			OPC		
No of taps	4-tap	8-tap	16-tap	4-tap	8-tap	16-tap
Number Of Slice Registers	126	158	222	116	148	212
Number of Slice LUTs	436	444	476	1278	1286	1318
Number of fully used Slices	97	107	120	88	99	111
Number of used Slices	465	495	578	1015	1335	1419

Table 4: Comparison of proposed FIR Filter with other architectures

Design	MSP(ns)	MSF(MHz)	NOS	SREG	SLUT	Slice delay product	%Improvement in Slice delay product		%Improvement in frequency	
Meher (2008)	4.17	239	275	688	833	1147	OPC	Dual	OPC	Dual
							80	75	50	42
Meher (2011)	17.35	57	178	412	267	4632	OPC	Dual	OPC	Dual
							95	94	88	86
LogiCORE IP FIR Compiler Xilinx (2010)	3.96	252	368	970	806	1457	OPC	Dual	OPC	Dual
							84	80	48	39
Sang Yoon Park (2014) [R=2] **	5.11	195	205	671	517	1048	OPC	Dual	OPC	Dual
							78	72	59	53
Sang Yoon Park (2014) [R=4] **	10.91	91	126	397	277	1375	OPC	Dual	OPC	Dual
							85	79	81	78
Proposed Design with Dual port Memory	2.426	412.133	120	222	476	291	-		-	
Proposed Design with OPC	2.081	480.558	111	212	1318	231	-		-	

* Note: Device used: Xilinx VIRTEX -5 XC5VSX95T-1FF1136

**R represents the number of time division multiplexing slots

Table 4 compares the synthesis results of 16 tap FIR filter architectures with existing architectures. Both the proposed and existing architectures are synthesized using the Xilinx Virtex-5 FPGA device. The minimum sampling period (MSP), maximum sampling frequency (MSF), number of slices (NOS), number of slice registers (SREG), and number of slice LUTs (SLUT) are listed in Table 4. The speed performance of proposed single channel FIR filter with OPC architecture shows 48 to 88% improvement and dual port memory structure shows 39 to 86% improvement respectively when compared to existing architectures. The slice-delay product shows 78 to 95% and 72 to 94% improvement respectively when compared to existing architectures. The proposed architecture provides maximum sampling frequency by adding the pipelined registers in between multiplier and adder and also provides area efficiency by use of single MAC with resource sharing. This has been shown in Figure 6.

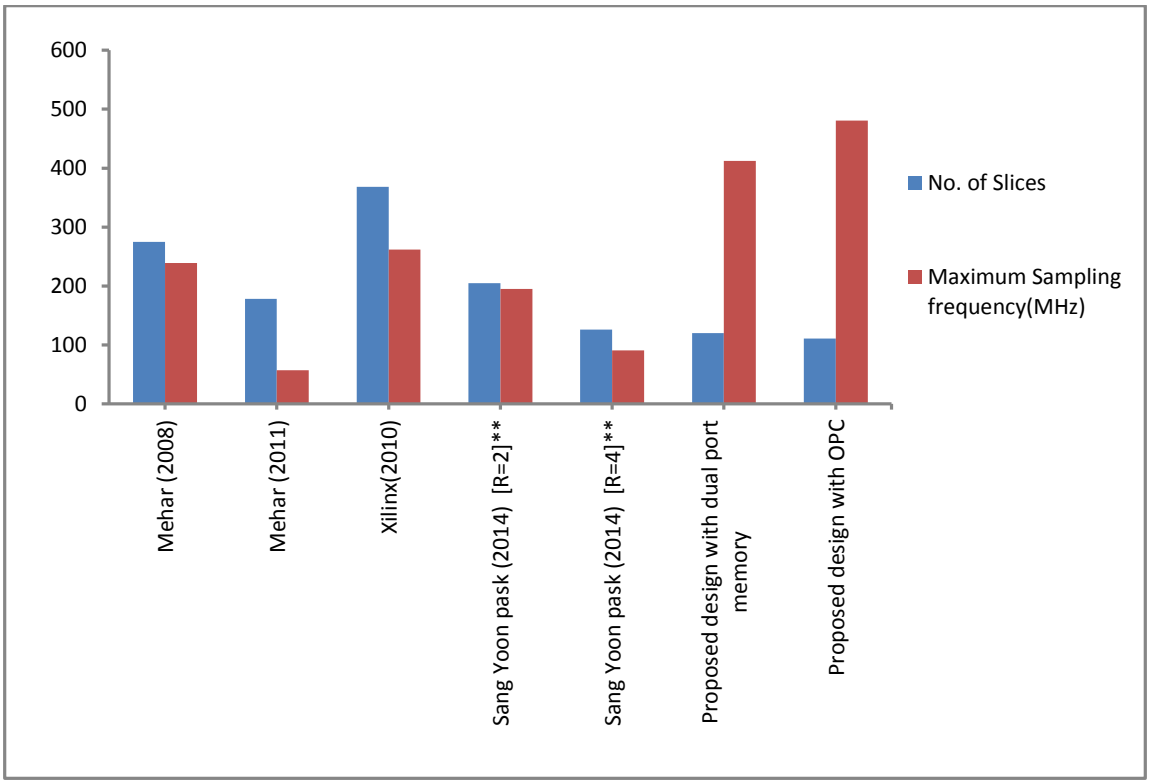

Figure 6: Performance results vs different filter architectures

Table 5: Hardware and Time complexities of proposed single channel FIR filter

Proposed single channel FIR filter	Complexities		
Number of taps	4	8	16
Number of multipliers	1	1	1
Number of registers	4+5=9	8+5=13	16+5=21
Number of adders	1	1	1
Latency	5	5	5

In Table 5 are listed the hardware and time complexities of single channel FIR filter design. The latency of the filter structure requires 5 clock cycles. For MAC operation, it takes 4 clock cycles in which two clock cycles required to register the inputs. one clock cycle for multiplication and another one clock cycle for accumulation. Finally one clock cycle is required for registering the output. The total number of registers required for this architecture is N+5 where N is the number of taps. Due to resource sharing principle, single multiplier and adder required for filter operation irrespective of number of filter taps.

Multichannel FIR Filter

Time division multiplexed multichannel reconfigurable FIR filter is synthesized using Xilinx VIRTEX -5 FPGA device and the performances are analyzed in Table 6. Due to Time Division Multiplexing (TDM), logic resources are optimized. From the Table 6, it can be analysed that area increases slightly for both the structures irrespective of the number of taps. The increasing area is due to the increased complexity of multiplexor logic and registers. Using dual port memory multiplier, area of the proposed architecture is minimized compared to the OPC based architecture. The speed of the OPC based multichannel based architecture is improved due to less number of decompositions is required compared to the dual port memory multiplier architecture.

Table 6: Performance analysis of Multichannel FIR filter

Performance measures	**Dual port Memory based LUT**						**OPC**					
No of taps	4-tap		8-tap		16-tap		4-tap		8-tap		16-tap	
Number of channels	2	4	2	4	2	4	2	4	2	4	2	4
MSP(ns)	2.426	2.430	2.430	2.528	2.528	2.839	2.089	2.081	2.081	2.269	2.311	2.823
MSF(MHz)	412.13	411.54	411.54	395.63	395.63	352.18	480	480.55	480.61	440.73	432.65	354.23
Number Of Slice Registers	158	222	222	352	352	594	148	212	212	342	342	584
Number of Slice LUTs	453	477	477	513	513	668	1303	1319	1319	1377	1377	1511
Number of fully used Slices	108	130	130	151	151	154	96	122	122	163	163	227
Number of Slices used	503	569	569	714	714	1108	1355	1409	1118	1556	1377	1868

* Note: Device used: Xilinx VIRTEX -5 XC5VSX95T-1FF1136

The proposed 16-tap multichannel FIR filter results are also compared with the conventional TDM based multichannel FIR filters using both Xilinx VIRTEX -5 FPGA and Cadence RC compiler with 0.18µm CMOS technology as given in Table 7 and 9 respectively. From Table 7 and 8, it is seen that TDM based multichannel FIR filter implementation is highly efficient in terms of utilization of logic resources. It is inferred from the Table 7 that the proposed multichannel FIR filter architecture occupies less NOS compare to conventional TDM based multichannel FIR filter for FPGA implementation. From Table 9, it is observed that drastic area reduction is achieved for the proposed multichannel filter architecture compared to the conventional multichannel architectures for the ASIC implementation.

Table 7: Synthesis results for multichannel 16-tap FIR filter

Parameter	**Conventional Design with Multichannel**				**Proposed design with multichannel**			
	Dual port Memory based LUT		**OPC**		**Dual port Memory based LUT**		**OPC**	
No of channels	2	4	2	4	2	4	2	4
Number Of Slice Registers	494	755	380	641	352	594	342	584
Number of Slice LUTs	2108	2168	3504	3537	513	668	1377	1511
Number of fully used Slices	350	506	215	300	151	154	163	227
Number of Slices used	2252	2314	3669	3878	714	1108	1377	1868

* Note: Device used: Xilinx VIRTEX -5 XC5VSX95T-1FF1136

Table 8: Comparison of proposed multichannel FIR Filter with other architecture

Design	MSP(ns)	MSF(MHz)	NOS	% Improvement in Slice delay product
Xilinx (2005)	2.202	454	216	-
Proposed Design with Dual port Memory	2.528	395.632	151	20
Proposed Design with OPC	2.339	427.460	163	20

Table 8 compares the synthesis results of 6-channel 8-tap FIR filter architectures with existing architecture (Xilinx 2005) using Xilinx Virtex-4 FPGA device. The Slice-delay product of proposed structures shows 20% improvement when compared to existing structure. The proposed multichannel FIR filter architectures achieve low complexity due to single MAC core architecture and the speed performance is reduced due to parallel MAC architecture used in existing architecture.

Table 9: Synthesis results of Cadence RC compiler for multichannel 16-tap FIR filter

Parameter	**Conventional Design with Multichannel**				**Proposed design with multichannel**			
	Dual port Memory based LUT		**OPC**		**Dual port Memory based LUT**		**OPC**	
No of channels	2	4	2	4	2	4	2	4
Area(mm^2)	0.103	0.111	0.131	0.148	0.064	0.073	0.083	0.093
Power(mw)	45.32	45.92	62.82	63.61	6.18	6.43	6.90	6.96

CONCLUSIONS

In this paper, an area efficient scheme for single MAC based implementation of single and multichannel FIR digital filters are discussed. It is shown that the hardware cost could be substantially reduced by resource sharing multiplier and adder. The results of the proposed single channel FIR filter architectures (with OPC and dual port memory) were also compared with existing architectures and the proposed architectures show improved performance in terms of speed and area reduction. The speed performance of proposed structures shows 48 to 88% and 39 to 86% improvement respectively when compared to existing architectures. The slice-delay product of proposed OPC structure shows 78 to 95% improvement and dual port memory structure shows 72 to 94% improvement when compared to existing architectures. The proposed structures of single channel FIR filter for FPGA implementation supports up to 480 MHz input sampling frequency.

In multichannel FIR filters, dual port memory multiplier based architecture offers less area whereas OPC based architecture offers higher speed. The Slice-delay product of proposed structures achieves 20 % improvement when compared to existing structure. Thus, the proposed multichannel architectures achieves low complexity and efficient re-programmability which makes the architectures a viable alternative to the existing architectures for real time signal processing applications.

REFERENCES

[1]. Veranda Bhargav Alluri, J. Robert Heath, and Michael Lhamon, “A New Multichannel, Coherent Amplitude Modulated, Time-Division Multiplexed, Software-Defined Radio Receiver Architecture, and Field-Programmable-Gate-Array Technology Implementation” IEE Transactions On Signal Processing, Vol. 58, No. 10, October 2010.

[2]. Liu Ming, Yan Chao,” The Multiplexed Structure of Multi-channel FIR Filter and its Resources Evaluation,” International Conference on Computer Distributed Control and Intelligent Enviromental Monitoring, IEEE 2012.

[3]. Kuan-hung, and Tzi-Dar, “A low power digit based Reconfigurable FIR Filter,” IEEE Transactions on circuits and systems, Vol. 53, Aug 2006.

[4]. Pramod Kumar Meher, “New Approach to Look Up Table Design and Memory-Based Realization of FIR Digital Filter,” IEEE Transactions on circuits and systems irregular papers, Vol. 57, No. 3, March 2010.

[5]. R.Mahesh, and A.P Vinod, “New Reconfigurable Architectures for implementing FIR Filter with low complexity,” IEEE Transactions on computer aided design of integrated circuits and systems, Vol. 29, Feb 2010.

[6]. T. Solla, and O. Vainio, “Comparison of programmable FIR filter architectures for low power,” in Proc. of 28th European Solid State Circuits Conference, pp. 759-762, September 24 – 26, 2002

[7]. Pramod Kumar Meher, S. Chandrasekaran, and A. Amira, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic,”IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017, Jul.2008.

[8]. P. K. Meher and S. Y. Park, “High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic,” in Proc. IEEE/IFIP19th Int. Conf. VLSI-SOC, Oct.2011,pp. 428–433.

[9]. Sang Yoon park and promod kumar meher,”Efficient

FPGA and ASIC Realizations of a DA-based Reconfigurable FIR Digital Filter", IEEE Transactions on Circuits and systems-II: Express Briefs, VOL.61, No.7, July 2014.

[10]. LogiCORE IP FIR Compiler v5.0, Xilinx, Inc., San Jose, CA, USA, 2010.

[11]. H. Yoo and D. V. Anderson, "Hardware-efficient distributed arithmetic architecture for high-order digital filters," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Mar. 2005, vol. 5, pp. v/125–v/128.

[12]. Xilinx Incorporation, "The Role of Distributed Arithmetic in FPGA-based signal Processing," Xilinx application notes, San Jose, CA.

[13]. Asgar Abbaszadeh and khostov D.sadeghipour "A New Hardware Efficient Reconfigurable FIR Filter architecture suitable for FPGA applications" proc IEEE DSP 2011

[14]. J. Park, et al., "Computation Sharing Programmable FIR Filter for Low-Power and High-Performance Applications", IEEE J. Solid stateCir. Sys., vol.39, no.2, pp.348-357, Feb. 2004

[15]. Bahram Rashidi , "High performance and low-power finite impulse response filter based on ring topology with modified retiming serial multiplier on FPGA'', IET Signal Process., vol.7, no.8, pp. 743–753.Feb 2013.

[16]. Allred, DJ, Yoo, H, Krishnan, V, Huang, W & DVAnderson, ''LMS adaptive filters using distributed arithmetic for high throughput'', IEEE Trans. Circuits Syst. I, Reg. Papers, vol.52, no.7, pp. 1327–1337. 2005

[17]. DF.Chiper, MNS. Swamy, Ahmad, MO & Stouraitis, T, ''Systolic algorithms and a memory-based design approach for a unified architecture for the computation of CT/DST/IDCT/IDST'', IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 6, pp. 1125–1137.2005

[18]. Jeng, SS, Lin, HC & Chang, SM, ''FPGA implementation of FIR filter using M-bit parallel distributed arithmetic'', in Proc. 2006 IEEE Intl. Symp. Circuits Syst. ISCAS 2006, pp. 4.

[19]. Digital Signal Processing solution ,''Designing for Optimal Results High-Performance DSP using Virtex-FPGAs'', Xilinx corporation, pp. 99-103,2005