High Speed Programmable FIR Filters for FPGA

Shahid Hassan[1, 2], Farhat Abbas Shah[1, 2], Umar Farooq[1]

Abstract ----- This paper presents high speed programmable FIR filters specifically designed for FPGA. Vendor provided components are used in Filter's MAC unit. FIR filters are programmable in terms of new coefficients.

Both UDF & FDF of FIR filters are analyzed. Results are presented for 16bit-20taps and 8bit-20taps on 2s100tq144-6 of Xilinx Spartan-II FPGA. Maximum speed improvement of about 64.83% for 16bit-20taps, 49.70% for 8bit-20taps filter in UDF FIR filters and 48.3% for 16bit-20taps, 21.47% for 8bit-20taps in FDF FIR filters have been achieved utilizing a small variation of area in some cores.

Index Terms ---- Digital Signal Processing (DSP), Multiply Accumulate (MAC), Finite Impulse Response (FIR) filters, Application Specific Integrated circuits(ASIC) , Field Programmable Gate Arrays(FPGA), System-on-Chip (SoC) Design. UDF (Unfolded Direct Form), FDF (Folded Direct Form).

I: INTRODUCTION

Present era of mobile computing and multimedia technology demands high performance and low power VLSI digital signal processing (DSP) systems. The availability of larger FPGA devices has started a shift of SoC designs towards reprogrammable FPGAs, thereby starting a new era of System-on-a-Reprogrammable-Chip (SoRC). Parameterized IP cores remain a standard way to utilize the improvement in FPGA technology and contend with time to market pressure through reuse. [2]

One of the most widely used operations in DSP is finite-impulse (FIR) filtering which performs the weighted summations of input sequences. There are two main types of FIR filter implementations namely sequential and parallel [3]. Former is selected for its low complexity and area over head as compared with the later. Sequential implementation requires a single multiplier as compared to multiple adders and multipliers required in parallel implementation.

Due to increasing complexity of DSP systems and large computations, filtering operations at times become slow. This makes high speed design an important area of research in the field of digital design.

Most of the previous work has been limited to the design of FIR filters with fixed coefficients [6]. FIR filter with programmable coefficients are used in many applications like adaptive pulse shaping and signal equalization on the received data in real time. So filter coefficients have been programmable in the design.

In past lots of work has been done for high speed FIR filters but most of them have used user defined components. So decision was made to use the components provided by the vendor itself. Proper configuration of components is required before its use. Idea of this work evolved from the fact that vendor provided components are the most suitable for FPGA implementation.

Organization of the paper is as follows: Section-II describes the implementation of reference core, vendor provided components that can be use in FIR filter cores, and their proper implementation. Results of the reference core and the new implemented cores are presented in Section-III. Finally, conclusion and future work has been shown in the end.

II: IMPLEMENTATION

a) Reference FIR Filter Core Implementation

FIR Filtering is one of the most widely used operations in Digital Signal Processing (DSP) devices. The basic equation of the UDF FIR Filter is given as

$$y(n) = \sum_{m=0}^{M} h(m)x(n-m) \qquad (1)$$

h_m's are the filter coefficients and xn-m ‘s are the filter input sample values and yn is the output.

1Department of Electrical Engineering, University of Engineering and Technology Taxila, Pakistan
2National Engineering and Scientific Commission, Islamabad, Pakistan

shahid_uet38@yahoo.com , fas77pk@hotmail.com , umarfarooq@uettaxila.edu.pk

Equation can also be written in the form shown below.

$Yn= h_0X_n + h_1X_{n-1} + h_2X_{n-2} + \ldots. + b_{m-1}X_{n-(m-1)}$ (2)
Implementation of equation is called direct form FIR Filter as shown in Fig.1.

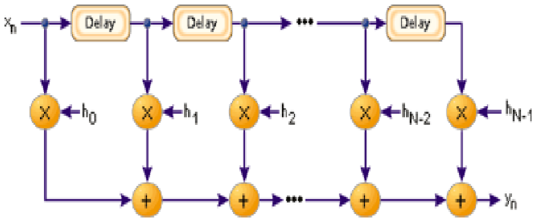


Fig.1. Unfolded Direct form FIR filter

Specification of the given filter core are: 64 taps, 16-bit data and coefficient width and single MAC implementation. Basic block diagram of the direct form programmable FIR Filter is shown in Fig.2.

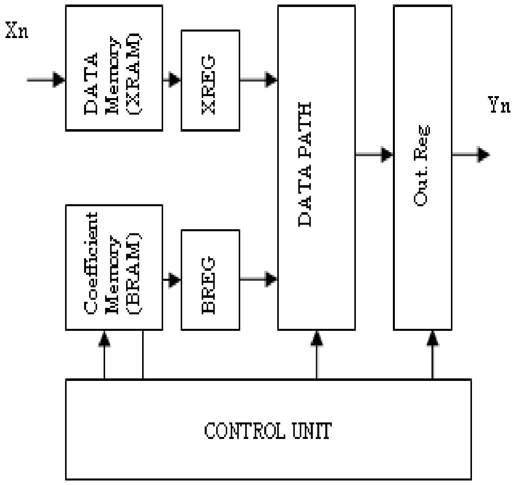


Fig.2. Basic Block Diagram of UDF FIR Filter

Data width of all components is 16 bit. XRAM and BRAM both have size 16x64. XRAM stores the input data while coefficients are already stored in the BRAM. Controller is responsible for sequencing and control of each logic function. It generates addresses, read and write signals for both memories. Data path is responsible for performing data manipulations. As for as operation is concerned, the present and

N-1 previous data samples of input Xn comes to the data path through XREG & are multiplied by corresponding N-tap coefficients one by one at each clock through BREG. Summation is also done at each clock by adding current and previous results of multiplications to form the filter output Yn after N cycles. Data path consists of a single 16 bit MAC unit and a round off module to get a 16-bit output of the filter after rounding the 32-bit output of MAC unit [1].

The basic block diagram of FDF FIR filter is shown in Fig.3.

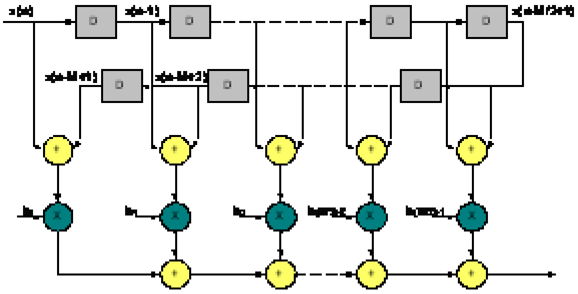


Fig.3. Folded Direct form FIR filter

Both UDF and FDF FIR Filter’s implementation are run-time programmable and uses user defined multipliers and adders. Upper limit for number of taps is 64, data and coefficient width is 16-bit. Area, and speed results of this core are considered as reference vto study the effect of vendor proided components of Xilinx on the performance of FIR Filters.

b) Vendor Provided Components

Speed and Power of FIR Filter is mostly affected by MAC unit of the filter. So decision has been made to use the vendor provided component available in the Xilinx library. These components are the most suitable for FPGA implementation.
Three main components are available in Xilinx library.

- RAM(random access memory)
- Multipliers
- Adders.

For MAC unit, only Multiplier and adder has been used.

Multipliers:
16- bit and 8-bit data width multipliers have been used in the design. Multiplier core is generated after proper configuration using CoreGen method.

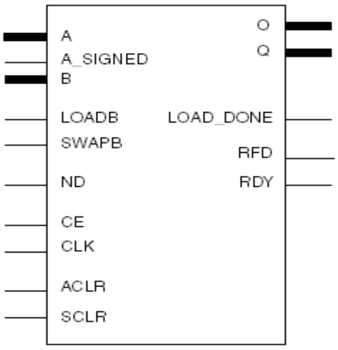


Fig.4. Multiplier Core Schematic symbol

Input data width can be changed from 1-64 bits. Corresponding outputs may very from 1-129 width. It supports two's complement signed / unsigned modes. It generates purely combinational and fully pipelined implementations. It has also optional registered output with optional clock enable and asynchronous and synchronous clears. It also has optional handshaking signals. This core is most suitable for FPGA implementation.

Adder:

32-bit and 16bit adders have been used in the design. It has been also generated using CoreGen method after proper configuration of the core.

This core supports both signed and unsigned data. The input data width may change from 1-256 bits while output result can be changed from 1-258 bits. Most suitable for FPGA implementation. It also has synchronous and asynchronous control options. Multiplier core three types of multipliers. It includes parallel, constant coefficient and sequential multipliers. Schematic symbol of Adder core is shown in Fig.5.

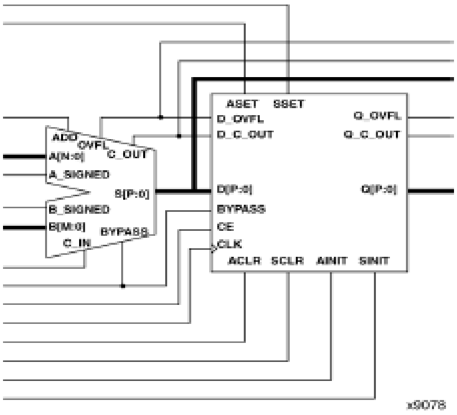


Fig.5. Adder Core Schematic Symbol

Eight cores were implemented. They include:

1. Ufdf_UDC (Unfolded direct form FIR Filter using User defined components)
2. Ufdf_Add_Inst (Unfolded direct form FIR Filter using Vendor's Adder instantiated in Ufdf_UDC)
3. Ufdf_Mul_Inst (Unfolded direct form FIR Filter using Vendor's Multiplier instantiated in Ufdf_UDC)
4. Ufdf_Add_Mul_Inst (Unfolded direct form FIR Filter using both Vendor's Adder & multiplier instantiated in a single core in Ufdf_UDC)
5. Fdf_UDC (Folded direct form FIR Filter using User defined components)
6. Fdf_Add_Inst (Folded direct form FIR Filter using Vendor's Adder instantiated in Fdf_UDC)
7. Fdf_Mul_Inst (Folded direct form FIR Filter using Vendor's Multiplier instantiated in Fdf_UDC)
8. Fdf_Add_Mul_Inst (Folded direct form FIR Filter using both Vendor's Adder & multiplier instantiated in a single core in Fdf_UDC)

III: SIMULATIONS AND RESULTS

Effects of all the implemented FIR filter cores have been observed. All the above fir filter cores were implemented and checked for functionality.

Results were taken in terms of area and speed for 16bits-20 taps and 8bits-20 taps. FIR filter cores were designed in Verilog HDL and were implemented using Xilinx 8.1i tool. Simulations were performed using Modelsim 5.7g.

a). Timing Results

Timing results for speed analysis are shown in table-I&II respectively. Results show a maximum speed improvement of 64.83% with Ufdf_Add_Inst for 16bit-20 taps, 49.70% with Ufdf_Add_Mul_Inst for 8bit-20 taps, 47.27% with Fdf_Add_Mul_Inst for 16bit-0 taps and 21.47% with Fdf_Mul_Inst 8bit-20taps. This is because the components instantiated are specifically designed for FPGA use and these are compatible with the FPGA internal architecture. Similarly the remaining filter combinations also show good results for both 16bit-20taps and 8bit-20taps.

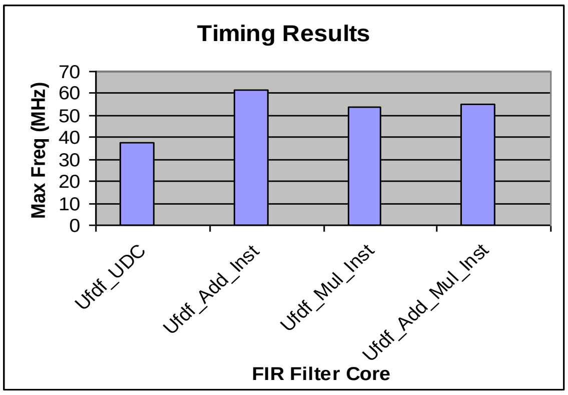
UNFOLDED DIRECT FORM FIR FILTER
FOR 16BIT-20TAPS

TABLE-I: TIMING RESULTS (16BIT-20TAPS)

Filter Cores	**Ufdf _ UDC**	**Ufdf_ Add_ Inst**	**Ufdf_ Mul_ Inst**	**Ufdf_Add _Mul_Inst**
Min Period	26.79 nS	16.285 nS	18.690 nS	18.161 nS
Input Arrival Time	9.574 nS	9.574 nS	9.574 nS	9.574 nS
Output Req Time	15.842 nS	12.647 nS	14.627 nS	16.526 nS
Max Freq	37.32 MHz	61.508 MHz	53.502 MHz	55.063 MHz
Speed Improvement	------	**64.83%**	**43.4 %**	**47.56%**

GRAPHICAL REPRESENTATION OF THE ABOVE RESULTS IS

GRAPH-I: TIMING RESULTS (16BITS-20TAPS)



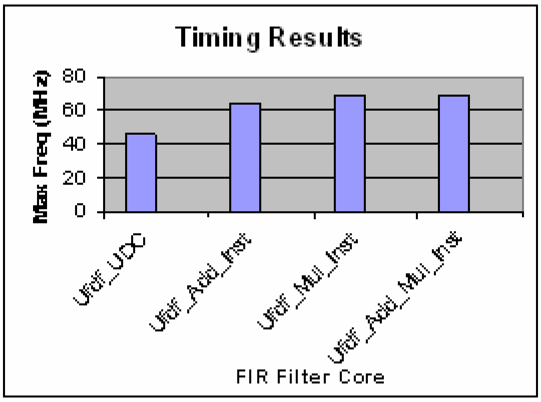
FOR 8BIT-20TAPS

TABLE-II: TIMING RESULTS (8BIT-20TAPS)

Filter Cores	**Ufdf_ UDC**	**Ufdf_ Add_ Inst**	**Ufdf_ Mul_ Inst**	**Ufdf_Add _ Mul_Inst**
Min Period	21.68 nS	15.781 nS	14.482 nS	14.482 nS
Input Arrival Time	9.484 nS	9.484 nS	9.484 nS	9.484 nS
Output Req Time	13.17 nS	13.124 nS	11.585 nS	11.585 nS
Max Freq	46.13 MHz	63.367 MHz	69.051 MHz	69.051 MHz
Speed Improvment	------	**37.37 %**	**49.70 %**	**49.70 %**

GRAPHICAL REPRESENTATION OF THE ABOVE RESULTS IS

GRAPH-II: TIMING RESULTS (8BIT-20TAPS)



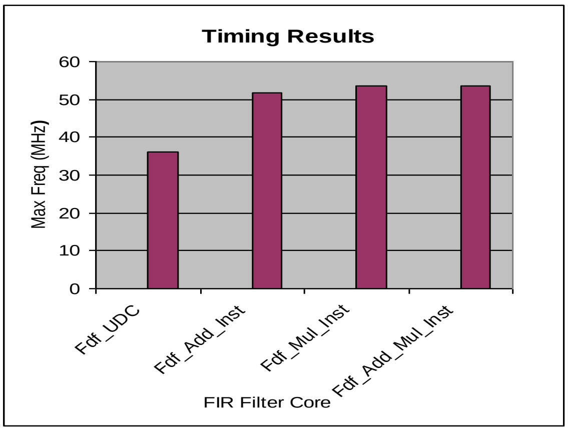
FOLDED DIRECT FORM FIR FILTER
FOR 16BIT-20TAPS

TABLE-III - TIMING RESULTS (16BIT-20TAPS)

Filter Cores	**Fdf_ UDC**	**Fdf_ Add_ Inst**	**Fdf_ Mul_ Inst**	**Fdf_Add_ Mul_Inst**
Min Period	27.69 nS	19.331 nS	18.65 nS	18.651 nS
Input Arrival Time	5.049 nS	5.049 nS	5.049 nS	5.049 nS
Output Req Time	11.87 nS	8.543 nS	10.703 nS	8.54 nS
Max Freq	36.16 MHz	51.73 MHz	53.62 MHz	53.616 MHz
Speed Improvement	------	**43.06%**	**48.3%**	**47.27%**

GRAPHICAL REPRESENTATION OF THE ABOVE RESULTS IS

GRAPH-III: TIMING RESULTS (16BITS-20TAPS)



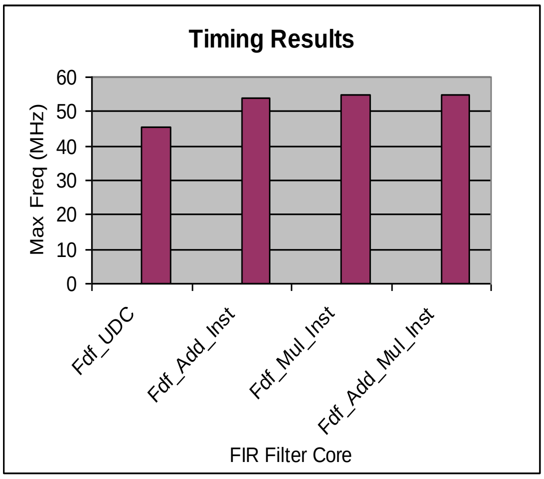
FOR 8BIT-20TAPS

TABLE-IV - TIMING RESULTS (8BIT-20TAPS)

Filter Cores	Fdf_ UDC	Fdf_ Add_ Inst	Fdf_ Mul_ Inst	Fdf_Add_ Mul_Inst
Min Period	22.06 nS	18.47 nS	18.161 nS	18.16 nS
Input Arrival Time	5.049 nS	5.049 nS	5.049 nS	5.049 nS
Output Req Time	9.76 nS	9.263 nS	9.26 nS	9.263 nS
Max Freq	45.33 MHz	54.139 MHz	55.063 MHz	55.063 MHz
Speed Impro vment	------	**19.43 %**	**21.47 %**	**21.47 %**

GRAPHICAL REPRESENTATION OF THE ABOVE RESULTS IS

GRAPH-IV: TIMING RESULTS (8BITS-20TAPS)



b). Area Results

Area results, for both 16bit-20taps and 8bits-20taps, are shown in Table-V, VI, VII & VIII respectively. Results show small variations in different fir filter cores. Both area improvement and loss has been observed. In UDF FIR filter, for 16bit-20taps, Ufdf_Add_Inst requires 5% more area for implementation while Ufdf_Mul_Inst and Ufdf_Add_Mul_Inst require 3.54% and 0.3% less area respectively for implementation as compared to reference core. Similarly, for 8bit-20taps, Ufdf_Add_Inst and Ufdf_Add_Mul_Inst require 9% and 1.6% more area while Ufdf_Mul_Inst requires 0.5% less area for implementation. While in case of FDF FIR filter using 16bit-20 taps, all the three new cores show improvement in area utilization of 23.1%, 8.5% and 30% respectively but in case of 8bit-20taps, a small loss of area occurred. Hence efficiency in terms of area consumed has also been observed.

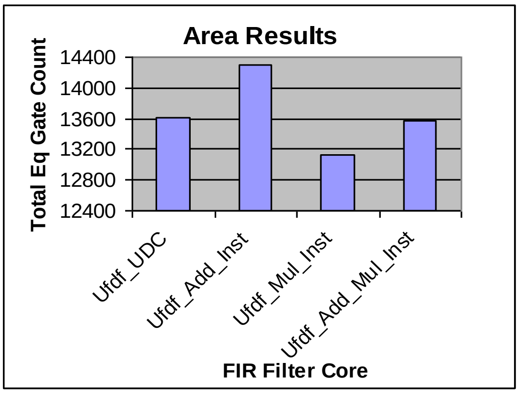
UNFOLDED DIRECT FORM FIR FILTER FOR 16BIT-20TAPS

TABLE-V: AREA RESULTS (16BIT-20TAPS)

Filter Cores	Ufdf_ UDC	Ufdf_ Add_ Inst	Ufdf_ Mul_ Inst	Ufdf_Add_ Mul_Inst
No. of Slices	70%	72%	74%	75%
Slice FF	32%	35%	34%	37%
Input LUTs	38%	35%	42%	37%
Bonded IOBs	60%	60%	60%	60%
Total Eq Gate Count	13610	14294	13128	13573
Area Expense	_____	**5%**	**-3.5%**	**-0.3%**

GRAPHICAL REPRESENTATION OF THE ABOVE RESULTS IS

GRAPH-V: AREA RESULTS (16BIT-20TAPS)



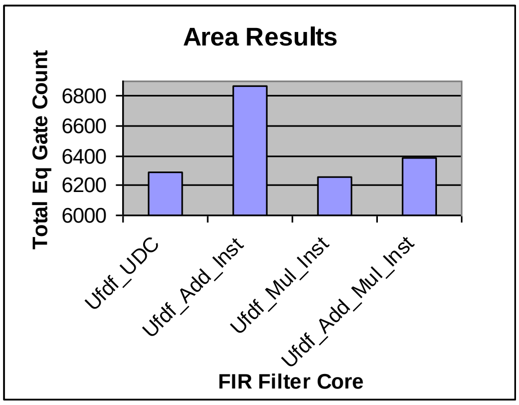
FOR 8BIT-20TAPS

TABLE-VI: AREA RESULTS (8BIT-20TAPS)

Filter Cores	Ufdf_ UDC	Ufdf_ Add_ Inst	Ufdf_ Mul_ Inst	Ufdf_Add_ Mul_Inst
No. of Slices	34%	35%	35%	37%
Slice FF	16%	18%	18%	19%
Input LUTs	20%	23%	17%	17%
Bonded IOBs	35%	35%	35%	35%
Total Eq Gate Count	6285	6872	6256	6387
Area Expense	_____	**9%**	**-0.5%**	**1.6%**

Graphical representation of above results is

Graph-VI: Area Results (8bit-20taps)



Folded Direct Form FIR Filter

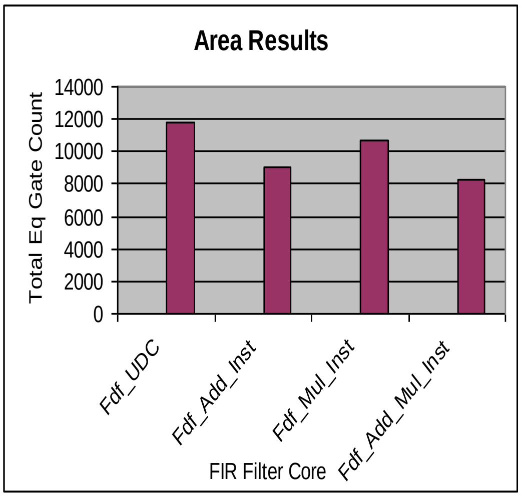
For 16bit-20taps

Table-VII: Area Results (16bit-20taps)

Filter Cores	Fdf_ UDC	Fdf_ Add_ Inst	Fdf_ Mul_ Inst	Fdf_Add_ Mul_Inst
No. of Slices	46%	43%	44%	44%
Slice FF	19%	19%	22%	22%
Input LUTs	40%	31%	35%	34%
Bonded IOBs	58%	59%	59%	59%
Total Eq Gate Count	11741	9029	10746	8223
Area Expense	_____	**-23.1%**	**-8.5%**	**-30%**

Graphical representation of above results is

Graph-VII: Area Results (16bit-20taps)



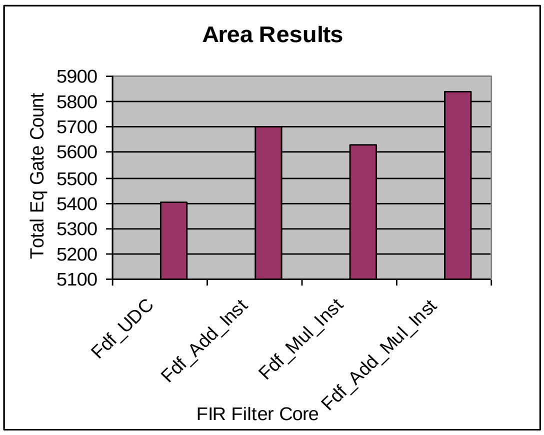
For 8bit-20taps

Table-VIII: Area Results (8bit-20taps)

Filter Cores	Fdf_ UDC	Fdf_ Add_ Inst	Fdf_ Mul_ Inst	Fdf_Add_ Mul_Inst
No. of Slices	23%	24%	33%	34%
Slice FF	10%	11%	13%	14%
Input LUTs	19%	21%	26%	27%
Bonded IOBs	33%	34%	34%	34%
Total Eq Gate Count	5401	5699	5632	5838
Area Expense	____	**5.52%**	**4.3%**	**8.1%**

Graphical representation of above results is

Graph-VIII: Area Results (8bit-20taps)



IV: CONCLUSION

Eight FIR filter cores have been implemented of both UDF and FDF FIR filters. Two of filter cores have all user defined components while the other six have adders and multipliers which are provided by the vendor Xilinx itself. Results show that vendor provided components give better performance especially in terms of speed. Area utilization in some cases has also been reduced upto 30%. Results are taken for 20taps and Device used is 2s100tq144-6 of Xilinx Spartan-II FPGA.

The above work shows a series of high speed FIR filters suitable for FPGA. Above designed FIR filters will provide suitable platform for real time data processing systems or DSP applications. The work can be extended for transpose direct form of FIR filters. RAM mega cells provided by the vendor can also be instantiated which can save more area.

ACKNOWLEDGEMENT:

We are very thankful to Dr. Habibullah Jamal, (Professor Electrical Engineering Department, UET Taxila), Mr. Muhammad Akhtar Khan (General Manager, NESCOM) and Mr. Fawad Nawaz (Assistant Manager, NESCOM) for their valuable support in this research work.

REFERENCES

[1] "Reconfigurable Low Power FIR Filter based on Partitioned Multipliers", Farhat Abbas Shah, Habibullah Jamal, Muhammad Akhtar Khan, 16-19 December, KFUPM, Dhahran, KSA ICM2006.

[2] Xilinx Design Methodology for ASIC and FPGA Designers, 2004

[3] C.H.Wang, A.T.Erdogan and T.Arslan, "Algorithmic Implementation of Low-Power High Performance FIR Filtering IP Cores", 18th International Conference on VLSI Design (VLSID'05), 1063-9667/05©2005 IEEE

[4] "Algorithms for Low Power and High Speed FIR Filter Realization using Differential Coefficients", N.Sankarayya, Kaushik Roy, Debashis Bhattacharya, 1057-7130/97©1997 IEEE

[5] A.T. Erdogan and T.Arslan,"Low Power Block Based FIR Filtering Cores", ISCAS-2003

[6] Jongsun Park, "Computation Sharing Programmable FIR Filter for Low Power and High Performance Applications", IEEE journal of Solid State Circuits. Vol.39, no.2, February 2004

[7] K. Wiatr and E. Jamro, "Constant coefficient multiplication in FPGA structures", Euromicro Conference, 2000. Proceedings of the 26th, 2000.

[8] M. Yamada and A. Nishihara, " High Speed FIR digital filter with CSD coefficients implemented on FPGA", Design Automation Conference, 2001. Proceedings of the ASP-DAC 2001. Asia and South Pacific, 2001.

[9] C. J. Chou, S. Mohanakrishnan, and J. B. Evans. "FPGA implementation of digital filters". Proc. Int. Conf. Sig.Proc., Applcn. and Technology, pages 80–88, 1993

[10] J. B. Evans. "An efficient FIR filter architecture". IEEE Int. Symp. Circuits and Syst., pages 627–630,May 1993.

[11] J. B. Evans. 'Efficient FIR filter architectures suitable for FPGA implementation." IEEE Trans.Circuits and Syst., July 1994.

[12] "Area efficient FIR filters for high speed FPGA implementation" by Macpherson, K.N. Stewart, R.W. Vision, Image and Signal Processing, IEE Proceedings Dec. 2006, Dept. of Electron. & Electr. Eng., Univ. of Strathclyde, Glasgow;

[13] R. Hartley, P. Corbett, P. Jacob, and S. Karr. "A high speed FIR filter designed by compiler". IEEE Cust. IC Conf., pages 20.2.1–20.2.4, May 1989.

[14] A. Oppenheim and R. Schafer. Digital Signal Processing. Prentice-Hall, Inc., 1975.

[15] K. Azadet and C. J. Nicol, "Low-power equalizer architectures highspeed modems," IEEE Commun. Mag., vol. 36, pp. 118–126, Oct. 1998.

[16] R. Jain, P. T. Yang, and T. Yoshino, "FIRGEN: A computer-aided design system for high performance FIR filter integrated circuits," IEEE Trans. Signal Processing, vol. 39, pp. 1655–1668, July 1991.

[17] Jonksung Park et al., "High Performance and Low Power Filter Design Based on Sharing Multiplication", International Symposiumof Low Power Electronics and Design (ISLPED) 2002, Monterey, California, USA, August 12-14, 2002.

[18] S. Kim and G. E. Sobelman, "Digit-Serial Multiplier Design Using Skew-Tolerant Domino Circuits," IEEE International ASIC/SOC Conference, 2001.

[19] C. Lemonds, "A 500 MHz, One Volt 16 by 16 Bit Multiplier for DSP Cores," VLSI Signal Processing, IX, pp. 481-484, 1996