Design and FPGA implementation of sequential digital 7-tap FIR filter using microprogrammed controller

Adarsh Kumar Singh, Neeraj Kumar Dangwal, Mohan Singh, Manoranjan Kumar, Garima Singh, Anand Singh

Abstract —This paper describes the design and FPGA implementation of FIR filter using a microprogrammed controller based design approach.The controller controls the series of operation of the filter.To reveal the technique ,design of a sequential 7-tap digital FIR filter based on the microprogrammed controller is presented.The projected FIR filter is coded in VHDL using sructural design approach.Performance assessment is done based on the implementation results obtained through Xilinx ISE 8.1i tool.

Index Terms— FIR Filter, FPGA, Microprogrammed controller, Xilinx ISE, Simulation.

1 Introduction

Digital filters are classified either as finite duration unit impulse response (FIR) filters or infinite duration unit impulse response (IIR) filters depending on the unit pulse response of the system. Digital finite impulse response (FIR) filters are the basic building block of many digital signal processing (DSP) systems. FIR filters are used due their stability, easy to accomplish linear phase and can be realized efficiently in hardware.The main objectives of digital FIR filter are to filter out undesirable parts of the signal, shape of the spectrum of signals in communication channels, signal detection or analysis in radar applications. Adders, multipliers and delay elements are the main components used in the implementation of digital FIR filters [1]. FIR filter performs a linear convolution on a window of N data samples which can be mathematically expressed as follows with input x(n) and output y(n) :

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

bk is the set of filter coefficients ,also known as tap weights, that make up the impulse response.
A direct form implementation of an FIR filter can be readily developed from the convolution sum as shown in fig.1.

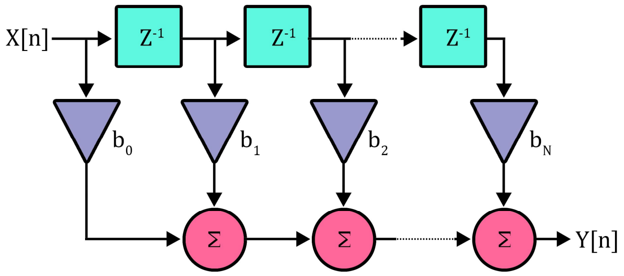


Fig (1)

Direct form FIR filters are also known as tapped delay line or transversal filters. N-tap FIR filter consists of N delay elements, N multipliers and N-1 adders or accumulators. The impulse response of the FIR filter can be directly incidental from the tap coefficients b_k..
Several techniques for the implementation of digital FIR using Field Programmable Gate Array (FPGA) are available [2-4]. The objective of this paper is to demonstrate a novel microprogrammed controller [5-7] based technique using an example of sequential 7-tap digital FIR filter.

2 Design Methodology

2.1 Datapath Architecture

The proposed FIR filter architecture comprises of two main building blocks which are datapath unit and control unit. The block diagram of the 7-tap sequential FIR filter with the integrated datapath and control unit is shown in fig. 2.

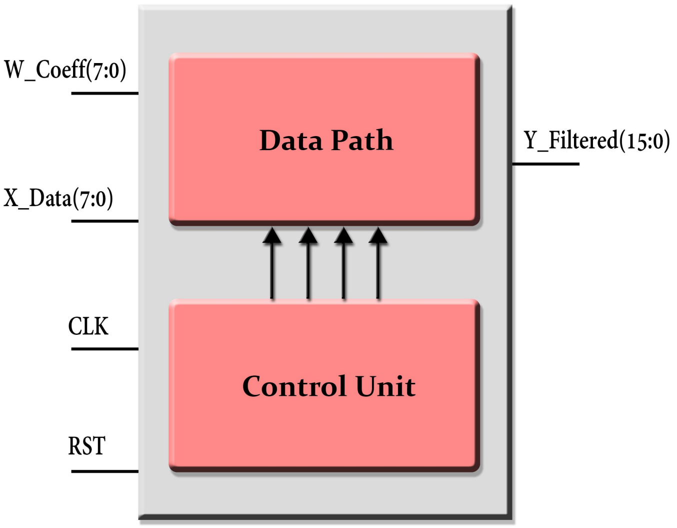


Fig (2)

Fig. 3 illustrates the datapath architecture of 7-tap FIR filter.

The datapath architecture consists of the following sub modules: two 8 bit 4:1 multiplexers which are used as data selectors, eight 8-bit coefficient registers (wo, $w1$, $w2$, $w3$, X_{n-0}, X_{n-1}, X_{n-2}, X_{n-3}) one 16 bit multiplier, one 16 bit adder and one register [8] to store the data. Each and every sub modules are coded in VHDL and finally integrated to achieve the complete datapath. The control signals generated by the microprogrammed controller for this datapath are fed to different sub modules for appropriate operation of the FIR filter.

2.2 Microprogrammed Controller

There are numerous methods to design the controller, such as hardwired controller and microprogrammed controller. In this paper, we used microprogrammed controller to systematize the operation of FIR filter. The leading advantage of the microprogrammed controller is its flexibility to append or alter by simply changing the microprogram in the ROM based control memory [9]. This makes the design of larger tap FIR filter much simpler. The tap coefficients (w0, w1, w2, w3)are loaded with data based on load enable (LE) signal. Once loading the input data in the first register, the input data is multiplied and accumulated based on the select signals (S1 and S0), product select (Ps) and load accumulator (lacc) signals.

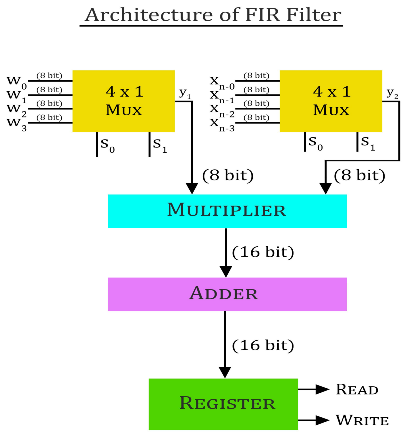


Fig (3)

2.3 Software used for Simulation

2.3.1 Project navigator Application Version ISE 8.1i of Xilinx Company

Xilinx has been a semiconductor manufacturing industry leader at the head of technology, market and business achievement. It is a tool to design the IC and to view their RTL (Register Transfer Logic) schematic .It is a tool to test the code on FPGA atmosphere and we can get the all parameters details required to implement the Chip.

2.3.2 ModelSim SE 6.5 of Mentor Graphics Company

Mentor Graphics was the first to unite single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS performance with the most excellent integrated debug and analysis environment build ModelSim the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it simple to accept in the majority of process and tool flows.

2.2.3 Simulation and Design Steps

The following diagram shows the basic steps for simulating a design in ModelSim.

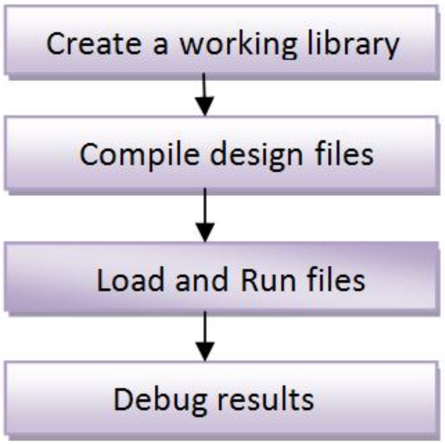


Fig (4)

- Creation of Working Library: In ModelSim, the entire designs are compiled into a library. Usually start a new simulation in ModelSim by creating a working library called "work," which is the default library name used by the compiler as the defaulting destination for compiled design units.
- Compilation Design: once creating the working library, design is being compiled into it. The ModelSim library format is well-matched across all supported platforms.
- Loading and Running the Simulator with the Design: Once the design compiled, we load the simulator with design by invoking the simulator on a configuration or entity/architecture pair (VHDL). Assuming the design loads

- *Adarsh Kumar Singh is currently pursuing B.Tech in ECE in DIT University, Dehradun, India, PH-9808268341. E-mail: adarshk952@gmail.com.*
- *Neeraj Kumar Dangwal is currently pursuing B.Tech in ECE in DIT University, Dehradun, India, PH-9634041019. E-mail: neerajdangwal659@outlook.com*
- *Mohan Singh is currently pursuing B.Tech in ECE in DIT University, Dehradun, India, PH- 9997031622. E-mail: mohansinghpal91@gmail.com*
- *Manoranjan Kumar is currently pursuing B.Tech in ECE in DIT University, Dehradun, India, PH-8909076011. E-mail: manoranjan-kumar112@gmail.com*
- *Garima Singh is working as Assistant Professor in the ECE department at DIT University, Dehradun, India, PH-9634912942. Email: singh01garima01@gmail.com*
- *Anand Singh is working as Assistant Professor in the ECE department at DIT University, Dehradun, India, PH-8979058120. E-mail: singh01anand01@gmail.com*

successfully, the simulation time is set to zero, and enter a run command to start simulation.

- Results Debugging: ModelSim's vigorous debugging environment is used to hunt down the cause of the problem.

3 FPGA Implementation and Simulation Results

The sequential FIR filter is designed and simulated by VHDL. For the synthesis, translation, mapping and place-and-route processes, Xilinx ISE 8.1i webpack is used. To implement the projected architecture, Spartan-6E FPGA device is used. Various reports are generated by the tools. The RTL view generated by the tool is shown in fig. 5. Three different test data cases used for testing the designed FIR filter is shown in the table I.

TABLE I: SIMULATION TEST CASES FOR 7-TAP FIR FILTER

Test No.	Tap coefficients(W)	Input Data (X)	Output Data(Y)
1.	{1,2,2,1}	{1,2,3,3}	{1,4,9,14,14,9,3}
2.	{3,6,6,5}	{2,10,3,3}	{6,42,81,97,86,33,15}
3.	{5,4,4,1}	{3,9,7,7}	{15,57,83,102,65,35,7}

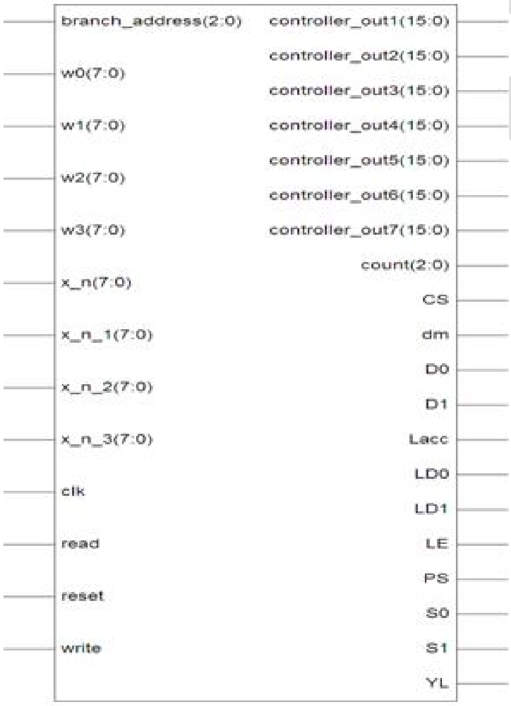
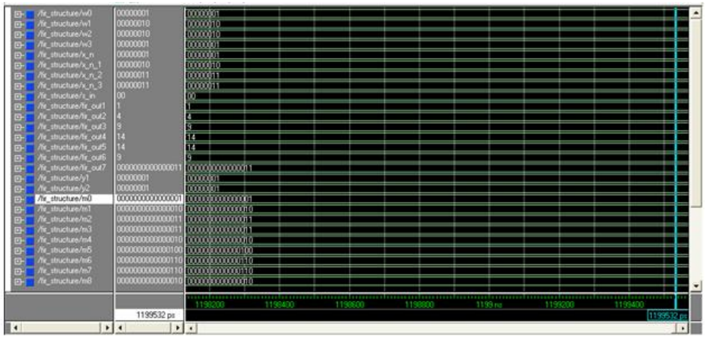


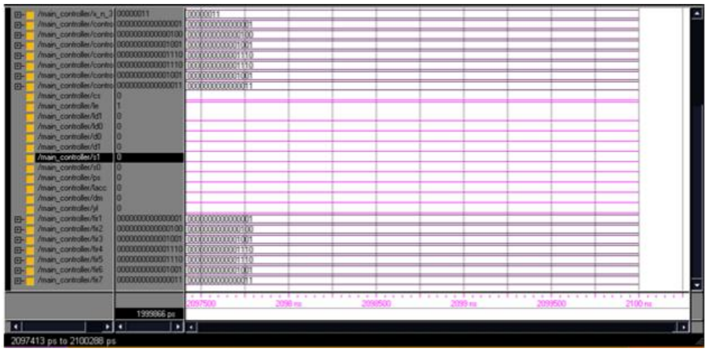
Fig (5)

Fig. 6 shows the simulation waveform of the datapath which verifies the first test case listed in table I .Finally, the datapath and microprogrammed controller are integrated together to show the simulation waveform of 7 tap FIR filter for first test case given in the table I. Fig. 7 and 8 shows the simulation waveforms of the FIR filter for first test case.



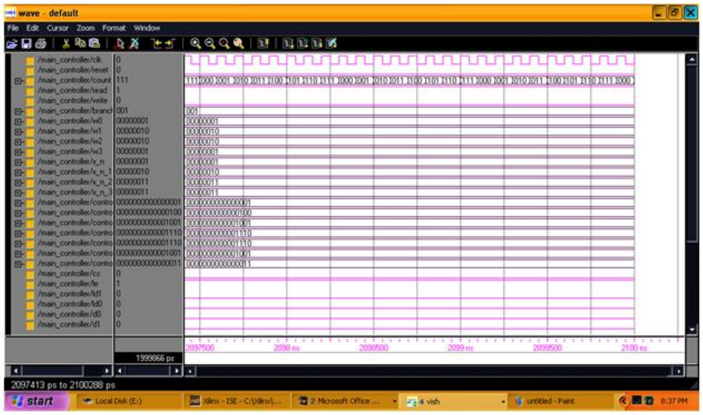
```
/fir_structure/w0       00000001
/fir_structure/w1       00000010
/fir_structure/w2       00000010
/fir_structure/w3       00000001
/fir_structure/x_n      00000001
/fir_structure/x_n_1    00000010
/fir_structure/x_n_2    00000011
/fir_structure/x_n_3    00000011
/fir_structure/c_in     00
/fir_structure/fir_out1 1
/fir_structure/fir_out2 4
/fir_structure/fir_out3 9
/fir_structure/fir_out4 14
/fir_structure/fir_out5 14
/fir_structure/fir_out6 9
/fir_structure/fir_out7 0000000000000011
/fir_structure/y1       00000001
/fir_structure/y2       00000001
/fir_structure/m0       0000000000000001
/fir_structure/m1       0000000000000010
/fir_structure/m2       0000000000000011
/fir_structure/m3       0000000000000011
/fir_structure/m4       0000000000000010
/fir_structure/m5       0000000000000100
/fir_structure/m6       0000000000000110
/fir_structure/m7       0000000000000110
/fir_structure/m8       0000000000000010
                        1199532 ps
```

Fig (6)



```
/main_controller/x_n_3   00000011
/main_controller/contro  0000000000000001
/main_controller/contro  0000000000000100
/main_controller/contro  0000000000001001
/main_controller/contro  0000000000001110
/main_controller/contro  0000000000001110
/main_controller/contro  0000000000001001
/main_controller/contro  0000000000000011
/main_controller/cs      0
/main_controller/le      1
/main_controller/ld1     0
/main_controller/ld0     0
/main_controller/d0      0
/main_controller/d1      0
/main_controller/s1      0
/main_controller/s0      0
/main_controller/ps      0
/main_controller/lacc    0
/main_controller/dm      0
/main_controller/yl      0
/main_controller/fir1    0000000000000001
/main_controller/fir2    0000000000000100
/main_controller/fir3    0000000000001001
/main_controller/fir4    0000000000001110
/main_controller/fir5    0000000000001110
/main_controller/fir6    0000000000001001
/main_controller/fir7    0000000000000011
                         1999866 ps
2097413 ps to 2100288 ps
```

Fig (7)



```
/main_controller/clk     0
/main_controller/reset   0
/main_controller/count   111
/main_controller/read    1
/main_controller/write   0
/main_controller/branch  001
/main_controller/w0      00000001
/main_controller/w1      00000010
/main_controller/w2      00000010
/main_controller/w3      00000001
/main_controller/x_n     00000001
/main_controller/x_n_1   00000010
/main_controller/x_n_2   00000011
/main_controller/x_n_3   00000011
/main_controller/contro  0000000000000001
/main_controller/contro  0000000000000100
/main_controller/contro  0000000000001001
/main_controller/contro  0000000000001110
/main_controller/contro  0000000000001110
/main_controller/contro  0000000000001001
/main_controller/contro  0000000000000011
/main_controller/cs      0
/main_controller/le      1
/main_controller/ld1     0
/main_controller/ld0     0
/main_controller/d0      0
/main_controller/d1      0
                         1999866 ps
2097413 ps to 2100288 ps
```

Fig (8)

4. Conclusion

The simulation of FIR controller is done and it is tested for different test cases we have presented a design of 7-tap FIR filter using microprogrammed controller and its FPGA implementation. The length of FIR filter is kept tap-7. The RAM controller is having 7 internal registers which are synchronised using reset and clock pulse.Performance evaluation is done by synthesizing and implementing the design in Spartan- 6E FPGA using Xilinx ISE tool. Since the size of the FIR filter presented in the paper is small, the results are not that significant, however, for larger tap FIR filters,these results would be significant.
Future research would focus on making an analogous parallel architecture of the FIR filter, increasing the tap size of the FIR filter architectures and comparing together the architectures for speed, area and power.

References

[1] S. A. Khan, Digital Design of Signal Processing Systems: A practical approach, John Wiley and Sons, United Kingdom, 2011.

[2] U. Meyer-Baese, G. Botella, D. E. T. Romero and Martin Kumm, “Optimization of high speed pipelining in FPGA-based FIR filter design using genetic algorithm,” Proc. of SPIE, Vol. 8401, 2012.

[3] Y. Zhou and P. Shi, “Distributed Arithmetic for FIR Filter implementation on FPGA,” Proc. of IEEE Intl. Conf. on Multimedia Technology (ICMT’2011), Hangzhou, China, pp. 294-297, July 2011.

[4] F. Nekoei, Y. S. Kavian and O. Strobel, “Some schemes of realization digital FIR filters on FPGA for communication applications,” Proc. Of 20th Intl. Crimean Conference on Microwave and Telecommunication Technology (CriMiCo’2010), Crimea, Ukraine, pp.616-619, Sept. 2010.

[5] B. W. Bomar, “Implementation of microprogrammed control in FPGAs,” IEEE Transaction on Industrial Electronics, Vol. 49, No. 2, pp. 415-422, Apr. 2002.

[6] A. Barkalov and L. Titarenko, Logic synthesis for Compositonal Microprogram Control Units, Springer, Berlin: Germany, 2008.

[7] R. Wiśniewski, A. Barkalov, L. Titarenko and W. Halang, “Design of microprogrammed controllers to be implemented in FPGAs,” Intl. Journal of Applied Mathematics and Computer Science, Vol. 21, No. 2, pp. 401-412, June 2011.

[8] M. Rafiquzzaman, Fundamentals of Digital Logic and Microcomputer Design, John Wiley and Sons Inc., 5th Edition, New Jersey: USA, 2005.

[9] A. A. Barkalov, L. A. Titarenko and K. N. Efimenko, “Optimization of circuits of compositional micro program control units implemented on FPGA,” Cybernetics and Systems Analysis,Vol. 47, No. 1, pp. 166-174, 2011.