

# Design and FPGA Implementation of Compositional Microprogram FIR Filter

Kamran Javed, Naveed Khan Baloch, Fawad Hussain, Dr. Muhammad Iram Baig  
University of Engineering & Technology, Taxila, Pakistan

## Abstract

**F**IR Filters on Field Programmable Gate Array (FPGA) are designed by different methods of Digital Design. Microprogramming based FIR filters are vastly used in Video and Image Processing application. Purpose technique is Compositional Microprogram Control Unit (CMCU) FIR Filter. CMCU is both time and area optimized filter than that of microprogram FIR Filter. Parallel architecture is used in Data path of design. Verilog Hardware Descriptive Language (HDL) is used to implement design. Results are evaluated on ModelSim SE Plus 6.1f and hardware optimization results are evaluated on Xilinx ISE web pack 10.1. As an example of synthesis, Compositional Microprogram Control Unit (CMCU) FIR Filter designed in this paper is also tested for real time Audio Filtering. Code is tested on FPGA XC3S700AN [14] using stereo audio codec (AKM AK4551) [13] on 50MHz clock frequency. Proposed filter is tested for third order but it can be extended for higher order which can be used for high speed applications like DSP applications e.g., Noise Cancellation, Video and Image Processing.

**Index Terms**— FPGA, Compositional Microprogram, Parallel Architecture, Audio Codec.

## 1. INTRODUCTION

Digital signal processing is very important process in many image and video applications. Finite impulse response (FIR) is a commonly used digital filter in many digital signals processing (DSP) [5]. FIR Filters are widely used because they have linear phase characteristics and guaranteed stability. Digital filters are mainly used for removing the undesirable parts of the input signal such as random noise or components of a given frequency content. FIR filters are commonly used in spectral shaping, motion estimation, noise reduction, channel equalization among many other applications. The simplest realization of an FIR filter is derived from.

In direct form mentioned above,  $y(n)$  are the Outputs,  $h(k)$  are Tap Coefficients,  $x(n)$  are the Inputs and  $x(n-k)$  are the delayed samples by time unit ' $k$ '.

There are two type of implementation FIR Filters.

- (i) Software
- (ii) Hardware

In software implementation we used Matlab and Java to implement FIR Filter. In hardware implementation we use programmable Digital Signal Processors (DSPs) which are program according to FIR filter instructions which are write in programming language like C [15]. Another hardware implementation of FIR filter is by configuring hardware like Complex Programmable Logic Device (CPLD) or Field Programmable Gate Array (FPGA).

In software implementation we use general purpose computer for computing which is slow as compare to hardware implementations where we use dedicated hardware which provide fast computation as compare to general purpose computer [15]. Hardware implementation itself has two type in processor based implementation hardware is programed according to filter requirements which Fetch, Decode and Executes the instructions while configuration of FPGA for FIR filter is more faster implementation even as compare to processor based implementation. In FPGAs actually we design hardware as compare to processor based technique where we only program pre design hardware. This paper presents hardware implementation on FPGA. The architecture of FIR Filter is Compositional Microprogram.

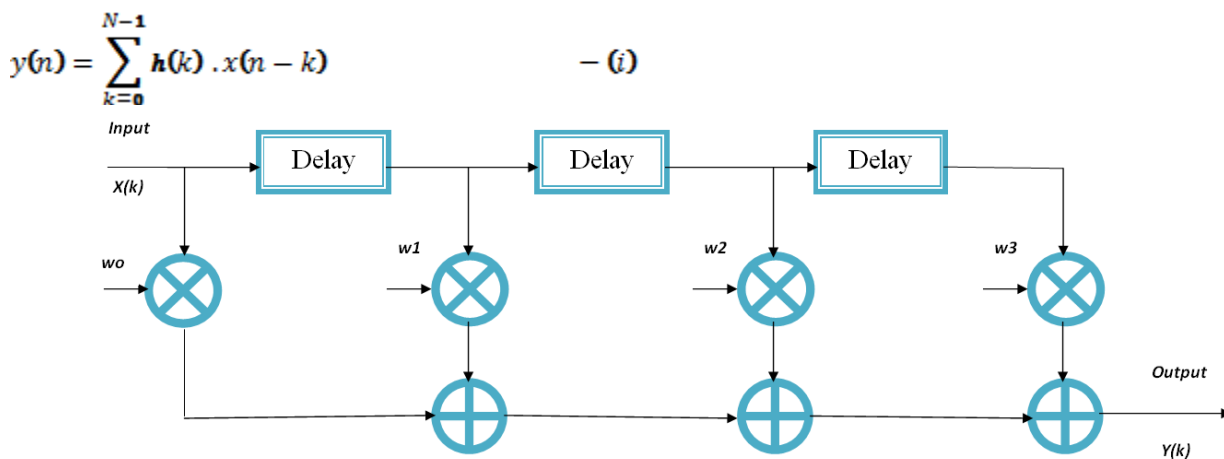


Fig.1 FIR Filter

## 2. DESIGN ARCHITECTURE OF FIR FILTER

The architecture of proposed FIR Filter is divided into two parts:

- I. Control Logic (Control Unit)
- II. Components that actually execute the Logic (Datapath)

Control Unit is controlling part of FIR Filter it undergoes different states, each state generates commands to Datapath which are executed as per direction of Control Unit in the Datapath of FIR Filter. Control Unit just think what are the control sequences and don't know how the design will operate on data, Datapath gets the signals from Control Unit and don't think what next, and execute the current control signals. So, the fig.2 clearly states that Control Unit is what which generates control signals and decides what to do, and Datapath is what which gets control signals from the Control Unit and executes the job.

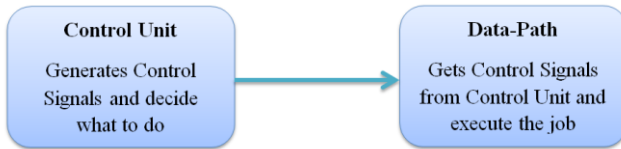


Fig 2. FIR Filter Design Partitioning

### 2.1 CONTROL UNIT

Control Unit takes decisions and produces control signals to Datapath. Control Unit doesn't have artificial intelligence to command operations. It goes through predefined sequence of operations. There are different ways of designing a Control Unit like Microprogram Control Unit and Hardwired Control Unit.

Flip-flops, decoders, gates and other digital circuits are used to implement the control logic in the hardwired architecture. One of benefits of hardwire organization is that it can be advanced to generate a fast mode of operation. On the other hand, Control memory is used to save the control information in the microprogram architecture [16]. The desired arrangement of micro-operations is programed in the control memory. Hardwired control is not beneficial since if there is a need to modify the design then modifying wiring among the various components is necessary [16]. Microprogram control is preferable because design can be modified easily by reprogramming the microprogram in the control memory.

Microprogram Control Unit consists of Control Memory which has microinstructions. Microinstructions in the control memory are addressed with the help of address register, which defines the address of corresponding microinstruction and as a result, control signals are produced. One of the most popular reasons to implement Control Unit by microprogramming is that it translates the hardware problems into programming problem, which makes it easy to control by a wider range of designers.

There is another way of designing Microprogram Control Unit i.e., Compositional Microprogram Control Unit

(CMCU). In CMCU, Mealy machine is implemented. Program Counter is used to address microinstructions in the Control Memory [1], [2]. The advantage of proposed technique is that it permits to calculate the next address of control memory in one clock cycle of Control Unit operation. Because of which CMCU is efficient than MCU [1], [2].

The proposed design of CMCU is shown in Fig 3 and Algorithm State Machine of filter is shown in Fig 4.

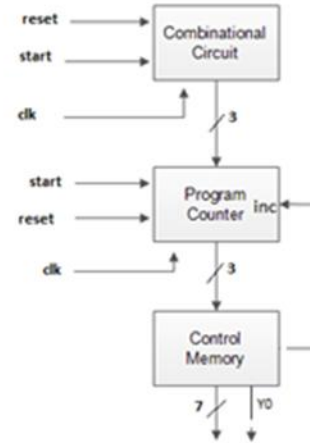


Fig. 3. FIR Filter Compositional Microprogram Control Unit

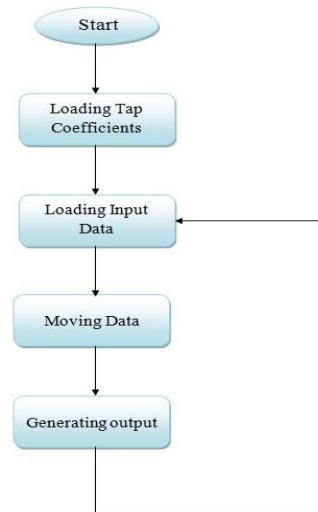


Fig. 4 Algorithm State Machine of Filter

The size of Control Memory is 8x8 having 8 microinstructions each of 8 bits. LSB 7 bit field of microinstruction includes the control signals for the Datapath where remaining single bit is used to increment or load the Program Counter. The Program Counter is of 3 bits to address 8 different microinstructions in the control memory. The Combinational Circuit is responsible for branching of Control Unit to capture new upcoming data.

The transition table of CMCU is shown in Table 1. The table shows the control signals are generated for Datapath which execute the job depending upon the control signals. According to Table 1, first microinstruction loads first tap coefficient, second microinstruction loads second tap

coefficient, third microinstruction loads third tap coefficient, fourth not only loads fourth tap coefficient but also clears the data registers, 5<sup>th</sup> microinstruction loads input data, 6<sup>th</sup> microinstruction moves the input data, 7<sup>th</sup> microinstruction latches the output. The first 4 steps are executed once at the start while step number 5, 6 and 7 are repeated again and again for each data.

## 2.2 FIR FILTER DATAPATH

The Datapath architecture of third order FIR Filter consists of the following sub modules: four 8-bit data registers, one 2-to-4 decoder, four 8-bit coefficient registers (ho, h1, h2, h3), four multipliers, three 16-bit adders and one 16-bit register for latching the output. The complete Datapath is obtained after coding each sub module in Verilog. The complete Datapath of four tap FIR filter with parallel architecture [3] is shown in Fig 5.

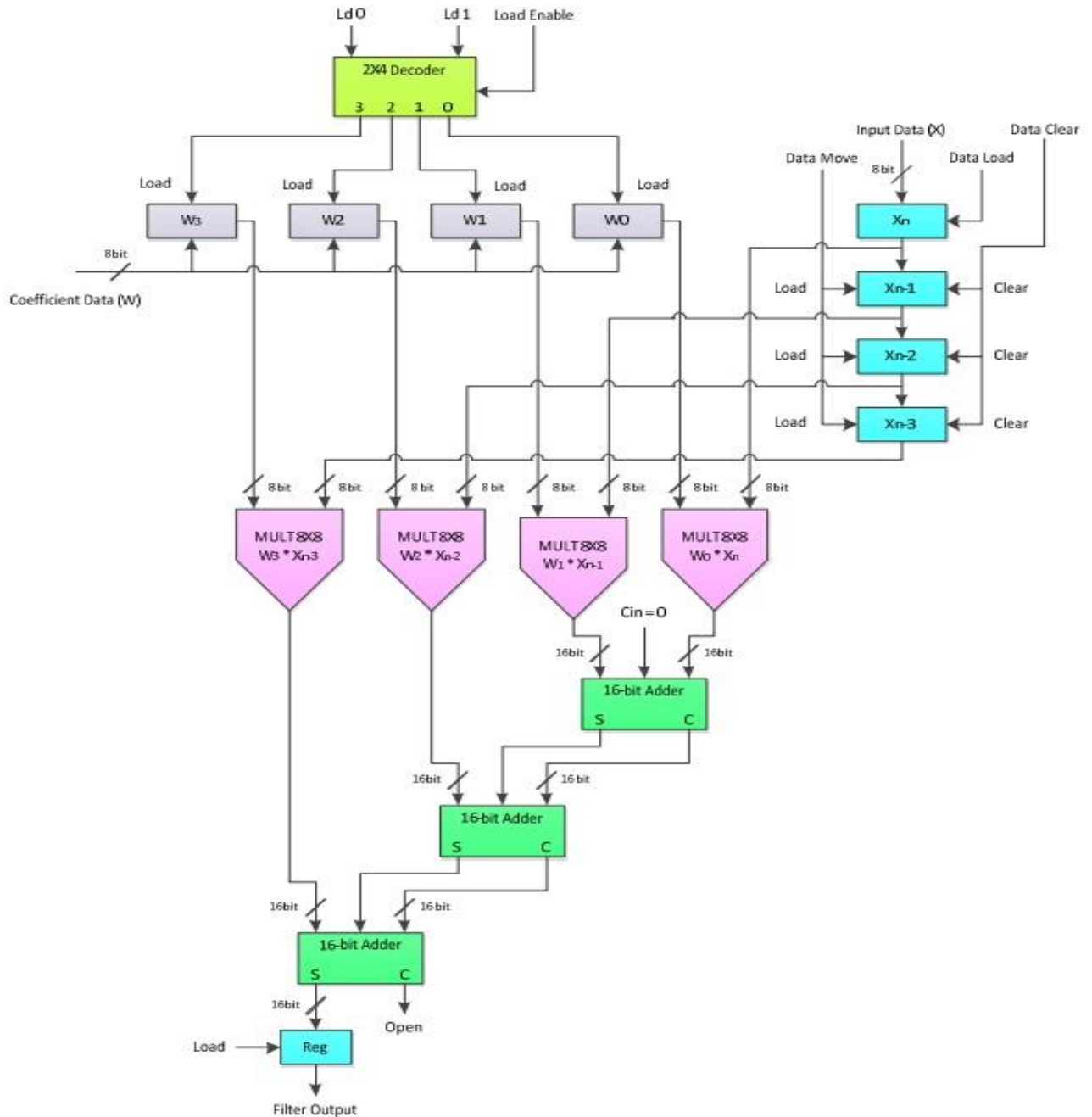


Fig.5 Datapath Architecture [3]

Sr #	Micro Operation	Increment	Load_en	Ld_1	Ld_0	D_clear	D_load	D_move	Y <sub>L</sub>
1	Loading Tap Coefficients	<i>h0</i>	0	1	0	0	0	0	0
2		<i>h1</i>	0	1	0	1	0	0	0
3		<i>h2</i>	0	1	1	0	0	0	0
4		<i>h3</i>	0	1	1	1	1	0	0
5	Loading Input Data <i>x[n]</i>	0	0	0	0	0	1	0	0
6	Moving Input Data	0	0	0	0	0	0	1	0
7	Generating Output <i>y[n]</i>	1	0	0	0	0	0	0	1

TABLE.1 CMCU Transition Table

### 3. FPGA IMPLEMENTATION

To implement the proposed architecture, the FPGA device used is Spartan-3AN (xc3s700AN-4fg484). Table 2 shows the design summary of Resource Utilization of the device

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	44	11,776	1%
Number of 4 input LUTs	219	11,776	1%
<b>Logic Distribution</b>			
Number of occupied Slices	132	5,888	2%
Number of Slices containing only related logic	132	132	100%
Number of Slices containing unrelated logic	0	132	0%
<b>Total Number of 4 input LUTs</b>	219	11,776	1%
Number of bonded IOBs	35	372	9%
Number of BUFGMUXs	1	24	4%
Number of MULT18X18SIOs	4	20	20%

TABLE.2 Device Resource Utilization

### 4. RESULTS AND SIMULATIONS

The Compositional Microprogram FIR Filter code is tested for three different input vectors as described in the Table 3. The Tap Coefficients for a particular test are fixed while the input data is changed continuously. The output generated by the third order FIR Filter for each input vector is shown in output vector. The result of all three different tests is shown in Table 3.

Test Case	Tap Coefficients (W)	Input Data (X)	Output Data (Y)
1	{5,4,4,1}	{3,9,7,7}	{15,57,83,102}
2	{3,6,6,5}	{2,10,3,3}	{6,42,81,97}
3	{1,2,2,1}	{1,2,3,3}	{1,4,9,14}

TABLE. 3. CMCU Transition Table

### 5. CONCLUSION

In Micro-program Controller based Parallel Digital FIR Filter, each memory location was of 12 bits in order to save the control signals [3] while in proposed technique 8 bits are used in Compositional Micro-programmed Controller based Parallel Digital FIR Filter. So, memory width is reduced from 12 bits to 8 bits. Number of memory locations is also reduced from 16 to 8. Memory size is reduced from 16x12 (192 bits) [3] to 8x8 (64 bits). It has not only increased the access time but also the overall speed is increased. Moreover, branching instruction for each pair of data is reduced. Now, each pair of data require 12 clock cycles instead of 16 clock cycles which were required by Micro-programmed Controller based Parallel Digital FIR Filter. So, overall speed is increased. Filter is tested on FPGA XC3S700AN using stereo audio codec (AKM AK4551) [13] on 50MHz clock frequency. As a future work, this FIR Filter can be optimized by using Xilinx IP Core and implementing Control Memory on dedicated FPGA BRAM.

## REFERENCES

- [12] Alexander Barkalov, Larysa Titarenko "Logic Synthesis for FSM-Based Control Units," vol. 53, Springer-Verlag, Berlin, 2009
- [13] Alexander Barkalov, Larysa Titarenko "Logic Synthesis for Compositional Microprogram Control Units," vol. 53, Springer-Verlag, Berlin, 2008
- [14] Mohammed S. BenSaleh, Syed Manzoor Qasim, M. Bahaidarah, H. AlObaisi, T. AlSharif, M. AlZahrani, and H. AlOnazi. "Field Programmable Gate Array Realization of Microprogrammed Controller based Parallel Digital FIR Filter Architecture " Proceedings of the World Congress on Engineering and Computer Science 2012 Vol II WCECS 2012, October 24-26, 2012, San Francisco, USA
- [15] Bruce W. Bomar, Senior Member, IEEE "Implementation of Microprogrammed Control in FPGAs.", IEEE Transactions On Industrial Electronics, Vol 49, No. 2, April 2002
- [16] Yajun Zhou, Pingzheng Shi. 'Distributed Arithmetic for FIR Filter implementation on FPGA.', 978-1-61284-774-0/11 ©2011 IEEE
- [17] Remigiusz Wiśniewski, Monika Wiśniewska, Marek Węgrzyn, Norian Marranghello. 'Design of Microprogrammed Controllers with Address Converter implemented on Programmable Systems with Embedded Memories', 978-1-4577-1958-5/11 ©2011 IEEE
- [18] Monika Wiśniewska, Remigiusz Wiśniewski, Marek Węgrzyn, Norian Marranghello. 'Reduction of the Memory Size in the Microprogrammed Controllers', 978-1-4577-1958-5/11 ©2011 IEEE
- [19] Syed Manzoor Qasim, Mohammed S. BenSaleh, Mazen Bahaidarah, Hesham AlObaisi And Tariq AlSharif, Mosab AlZahrani and Hani AlOnazi. "Design and FPGA Implementation of Sequential Digital FIR Filter using Microprogrammed Controller.", 978-1-4673-2015-3/12 ©2012 IEEE
- [20] Shoab Ahmed Khan. ' Digital Design Of Signal Processing Systems A Practical Approach', John Wiley and Sons, United Kingdom, 2011.
- [21] Dr. Shoab A. Khan And Hamid M. Kamboh. 'An Algorithmic Transformation for FPGA Implementation of High Throughput Filters', 978-1-4577-0768-1/11 ©2011 IEEE
- [22] Remigiusz Winiewski. 'Synthesis of Compositional Microprogram Control Units for Programmable Devices.', Ph.D. Thesis University of Zielona Góra Zielona Góra, Poland, 2008
- [23] Ms. Aye Thi Ri Wai and Ms. Phyu Phyu Tar "Translating A Microprogram To Hardwire Control" Proceedings of ECTI-CON 2008
- [24] Dave Vandebout. "Stereo loopback circuit" available at:  
<http://www.xess.com/static/media/projects/loopbk.zip>
- [25] Xilinx Development Team. "Spartan-3AN Documentation." available at:  
[http://www.xilinx.com/support/index.html/content/xilinx/en/supportNav/silicon\\_devices/fpga/spartan-3an.html](http://www.xilinx.com/support/index.html/content/xilinx/en/supportNav/silicon_devices/fpga/spartan-3an.html)
- [26] Pieter Abbeel Assistant Professor UC Berkeley "Signals and Systems- Implementation of FIR filters" available at:  
<http://ptolemy.eecs.berkeley.edu/eecs20/week12/implementation.html>
- [27] CADENTI "Hardwired control" available at:  
<http://www.cadenti.com/hardwired.html>
- [28] Shih-Lien lu and Hubert Stier " Design of Pipelined FIR Filter with MSB-First Multiplier" Dept. of Electrical and Computer Engineering , Oregon State University ,Corvallis, Or 97331 USA
- [29] Joseph B. Evans. "Efficient FIR Filter Architectures Suitable for FPGA Implementation," , ISCAS '93 in Chicago, Illinois.
- [30] Remigiusz Wiśniewski , Alexander Barkalov , Larisa Titarenko Wolfgang A. Halanl: "Design Of Microprogrammed Controllers To Be Implemented In FPGAs.", Int. J. Appl. Math. Comput. Sci., 2011, Vol. 21, No. 2, 401-412 DOI: 10.2478/v10006-011-0030-1
- [20] Alexander Barkalov, Larysa Titarenko "Logic Synthesis for Compositional Microprogram Control Units" Donetsk National Technical University, Poland

\*\*\*\*