IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 6, Ver. I (Nov - Dec. 2014), PP 77-83
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Optimization of Conventional FIR Filter Structures

Rahul Mhatre, Suraj Malpani, Alok Dadlani, Sanchay Srivastava
Department of Electronics and Telecommunication, University of Mumbai, India

***Abstract**: FIR Filter Technology has evolved as an eminent technology which has revolutionized the world of Digital Signal Processing (DSP). In addition, the characteristics of a digital filter can be easily changed under software control making them versatile. The research done in this paper focuses on the drawbacks of the given conventional structures and proposes a method to improvise them. The method given in this paper combines the architecture of linear and concurrent structure to achieve a semi concurrent filter with half the number of multipliers conventionally required. To authenticate this research, simulation of this FIR Filter was done and compared with traditional linear and concurrent structures in terms of sampling rates and chip area utilization.*
***Key Terms:** Digital FIR Filter, Reduced Multiplier Concurrent Filter, Linear Filter, Concurrent Filter, FPGA.*

I. Introduction

Finite Impulse Response (FIR) filters are an important class of digital filters used in digital communication and digital signal processing. This research paper provides a comparative study of the existing conventional structures which are linear and concurrent filters and proposes a new near-optimal filter design imbibing both area and speed design considerations. It focuses on the drawbacks of the given conventional structures and proposes a new method to improvise them. The method combines the architecture of linear and concurrent structure to achieve a semi concurrent filter with half the number of multipliers conventionally required. To authenticate this research, simulation of this FIR Filter was done and compared with the conventional structures. The proposed structure is compared with traditional approaches in terms of maximum clock frequency and slice utilization. The traditional and proposed approach was simulated on Xilinx 14.2 software.

II. Structural Form Of Fir Filter

The FIR filter is described as the weighted sum of current inputs and past inputs. It does not depend on past outputs and hence has no feedback. It can be expressed in equation form [1] as shown in (1):

$$y(n) = \sum_{k=0}^{N} b(k)x(n-k)$$

Equation 1

N is number of past inputs, x(n) and b(k) are set of weights applied to the inputs. The number of past inputs to be considered in computation of any output is called Taps of the filter.

FIR filters are basically linear phase filters described by principal of superposition [2] offering constant delay through all frequencies therefore avoiding phase distortion or delay distortion. Another property is time-invariance, meaning any shift or delay in input sequence causes a corresponding shift in output sequence.

2.1 Direct form

The direct form is the most common form and is derived directly from the filter equation given by (1). This is illustrated in Fig. 1. The inputs to the filter are stored in series of registers each delayed by one cycle. Each delayed input is multiplied by one of the filter coefficients and the partial products are summed to give the filter output.

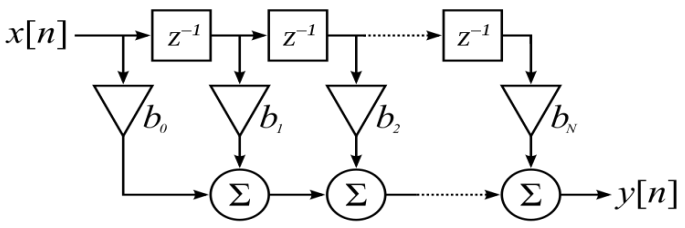

Fig.1: Direct Form Structure

2.2 Transposed Direct form

Similar to direct form, however here the current input is multiplied simultaneously by filter coefficients and partial sums are stored. The sums are then delays and reversed so that products will be combined correctly.

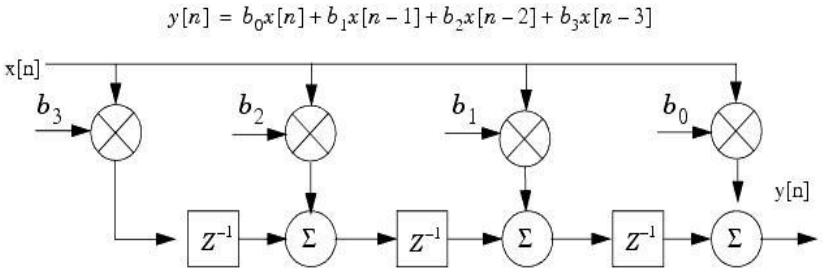

Fig.2: Transposed direct form

III. Implementation Approaches

3.1 Software implementation

Common practice of implementing FIR filters is using programmable Digital Signal Processing (DSP) chip. These chips are optimized for intensive arithmetic computation. The filter written in software is stored in form of memory and executed. Certainly this is a flexible approach as software are easy to modify but each instruction executes in sequential fashion on processor and hence the throughput is limited to clock speed of processor. Assuming 10 ns per instruction and assuming each multiplication and addition requires two instructions. A 100-tap filter will require approximately 100 multiplications and 100 additions and that will be (100multiplications*2instructions+100additions*2instructions)*10ns = 4000 ns.
Since larger filters will require more instructions this approach is limited to lower throughput applications.

3.2 Dedicated Hardware implementation

Application specific integrated Circuits (ASIC) are rigid tools offering no flexibility. Custom hardware however exploits the parallelism of FIR filter to maximize performance. However making custom made chips is time consuming and expensive. Also due to large non-reoccurring costs, design changes after chip production has been extremely expensive, only high production volumes are required for this approach to be economical.

3.3 FPGA implementation

Field Programmable Gate Arrays (FPGAs) offer powerful yet flexible alternative to ASICs and DSPs. The advantages include faster sampling rates compared to traditional DSP chips, lower costs than ASIC and flexible implementation. FPGAs are well suited to data path designs, like those in digital filtering applications.

IV. Traditional Approach

4.1 Concurrent FIR Filter

4.1.1 Structure

In Concurrent approach, each input is multiplied by respective coefficient simultaneously. Thus the number of multipliers required is equal to the number of taps of the filter. A subsequent adder tree adds the weighted inputs where each adder adds two weighted inputs. Thus a 3-Tap filter requires two adders as seen in Fig. 3. However as the number of taps increase the number of multipliers and adders required increases. For a hundred tap filter, it requires 100 multipliers and adder tree with 7 layers where subsequent layers have 50, 25, 13 (roughly), 7, 4, 2, and 1 adders respectively.

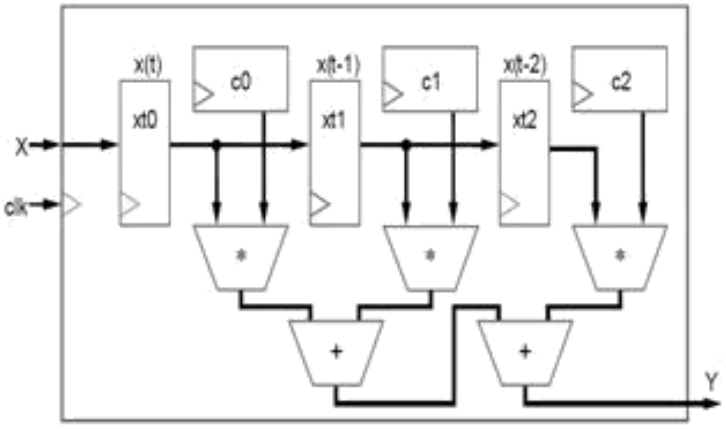

Fig. 3: Data path of Concurrent filter

4.1.2 Drawbacks

The structure has very high throughput however the number of multipliers required is equal to the number of taps of filter. Modern FPGA boards have on board multiplier but in a limited number. It becomes difficult to construct larger tap filters as each coefficient demands a separate multiplier thus increasing the chip utilization.

Moreover the extraneous hardware adds to increased power dissipation thus demanding better heat sinks.

4.2 Linear FIR Filter

4.2.1 Structure

In linear approach we use the same components and make recursive computation using a regulated data flow which is done with help of a control unit. The control unit is made of a 13 state Finite State Machine (FSM) as shown in Fig. 5. The first state loads all the registers each one with a delayed input. The next twelve states add weighted inputs and store them in the accumulator thus representing the 12 Taps of the filter. The last state loads the output register with the outputs and holds it till the next output is computed. The system now returns back to the first state.

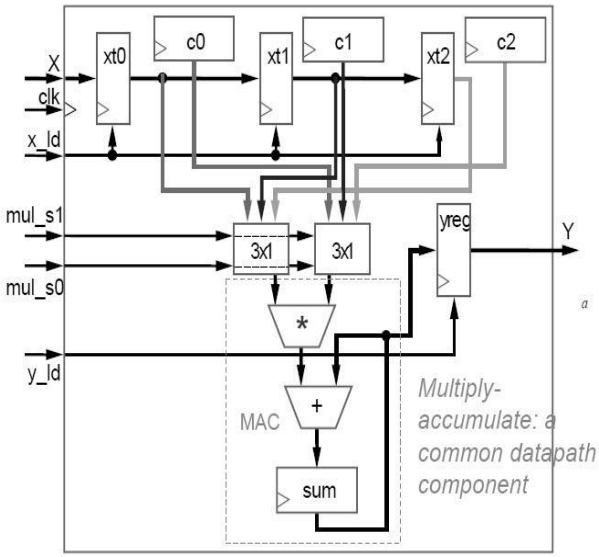

Fig. 4: Data Path of linear Filter

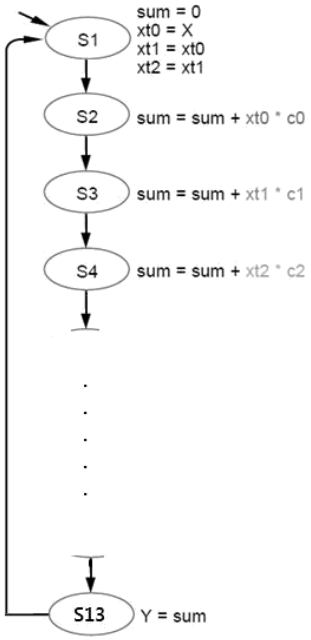

Fig.5: FSM for 12 Tap filter

4.2.2 Drawbacks

Though the structure demands a single multiplier the FSM becomes more complex and longer with addition of additional states. Larger FSM results in more clock cycles needed for computation of single output thus drastically reducing throughput for higher tap filters.

V. Proposed Approach

The method proposed here relies on generation of symmetric co-efficient for FIR filter using Hamming-Window method. FIR filter design by using hamming is stable as compared to rectangular and hanning windows techniques. Ripples in pass band are less in hamming as compare to other two techniques [4].

The structural form used to implement the structure is shown in Fig.6

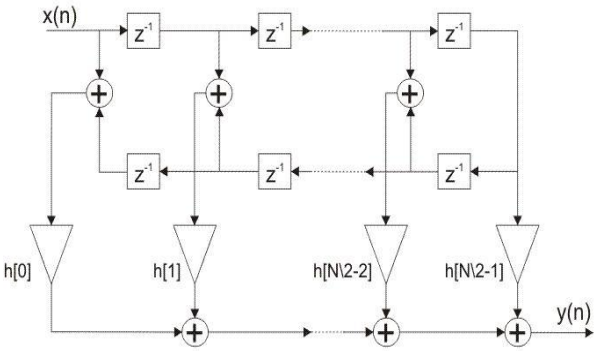

Fig.6: Symmetric coefficient structure

5.1 Structure

All the past inputs are now divided in two sections called as "Odd" and "Even". All odd inputs are multiplied with respective coefficients and the weighted inputs are then added through an adder tree and stored in an accumulator.

The Even inputs are then selected and are multiplied with the same coefficients using the same multipliers as in the earlier case thus reusing the multipliers. Similar to earlier case, all weighted inputs are added and then the final value is added to the existing value in accumulator to get the final output y[n].

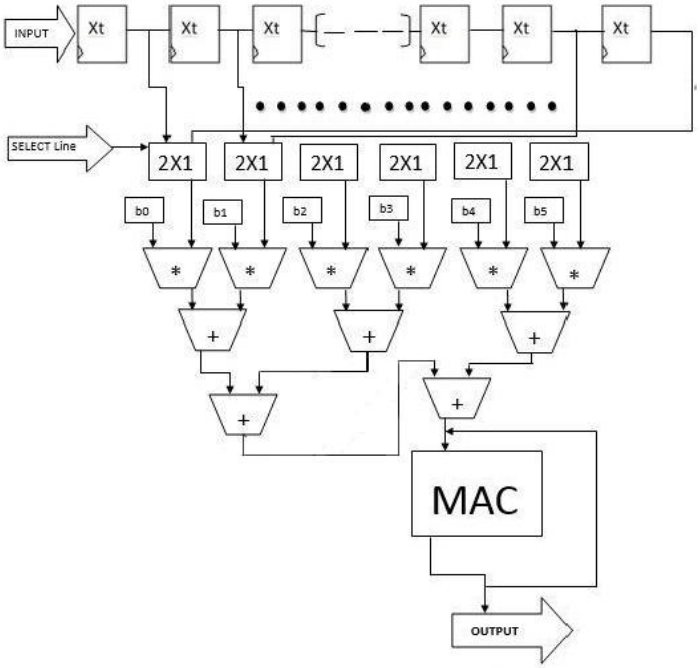

Fig.7. Data path of 12 Tap Proposed filter

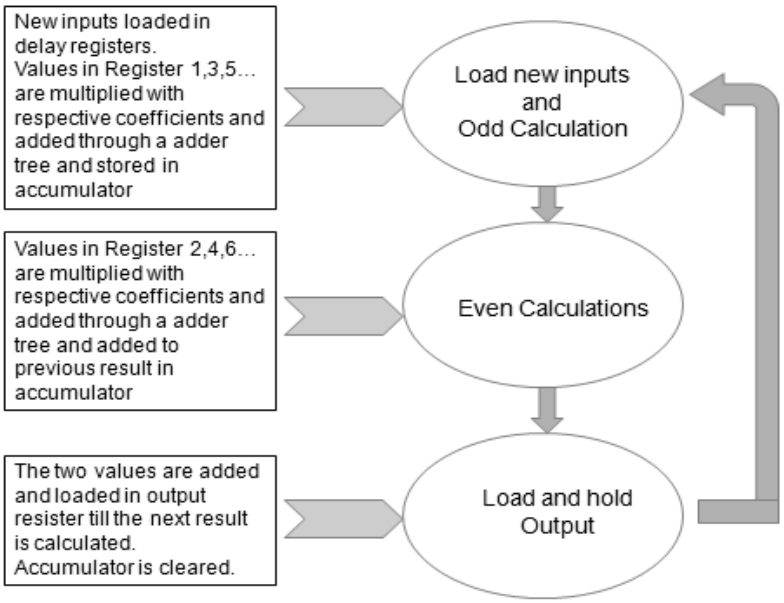

Fig.8. FSM for Proposed 12 Tap RMC filter

5.2 Advantages over earlier structures

The number of multipliers is reduced to half compared to concurrent filter thus resulting in substantial reduction in chip area utilization and also power dissipation.

Also unlike linear structure the FSM of the structure is independent of number of taps and the output is calculated in fixed 3 states.

VI. Simulation Results

All three structures were implemented in Xilinx ISE 14.2 and simulated using ModelSim. The coefficients for the filter were generated using MATLAB for a 1 KHz low pass filter using Hamming Window method. The target device used for simulation was Spartan 3s250epq208-4.

6.1 Timing summary

6.1.1 Concurrent FIR Filter

With each multiplier computing independently and simultaneously, the entire filter is a combinational circuit and does not require any FSM. The output is calculated within one clock cycle which can be verified as seen in simulation results shown in Fig.9. Simulation results yield a minimum time delay of 23.569 ns.

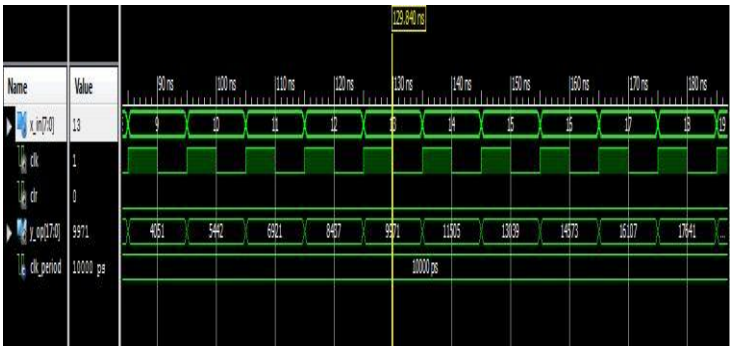

Fig.9: Simulation waveform for 12 Tap concurrent FIR filter

6.1.2 Linear Filter

With only one multiplier present, the same multiplier needs to be used for computing each tap of the filter. Thus the number of times the multiplier is reused is equal to the taps of the filter. Each multiplication requires one clock cycle. A 12 tap filter was simulated and thus it required 12 clock cycles to calculate one output as seen in Fig.10. Simulation results yield a minimum delay of 13.149ns. Thus a new output is calculated after every 157.788 ns for a 12 tap filter.

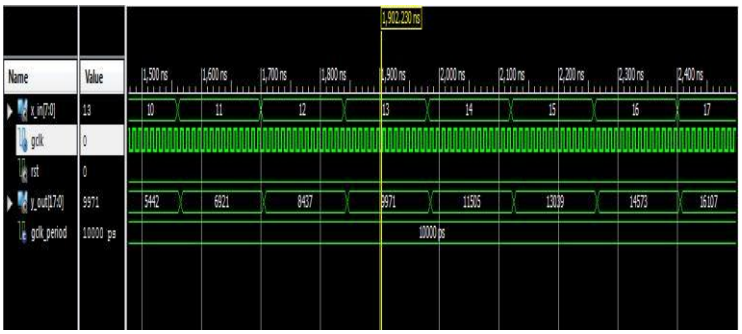

Fig.10: Simulation waveform for 12 Tap Linear FIR filter

6.1.3 Reduced Multiplier Concurrent FIR Filter

The proposed filter structure reused each multiplier twice the control of which is done by a 3 state FSM shown in Fig.8. Computation of each output thus required 3 clock cycles. Simulation results yield a minimum combinational delay of 20.103ns. Thus a new output is computed every 60.309ns.

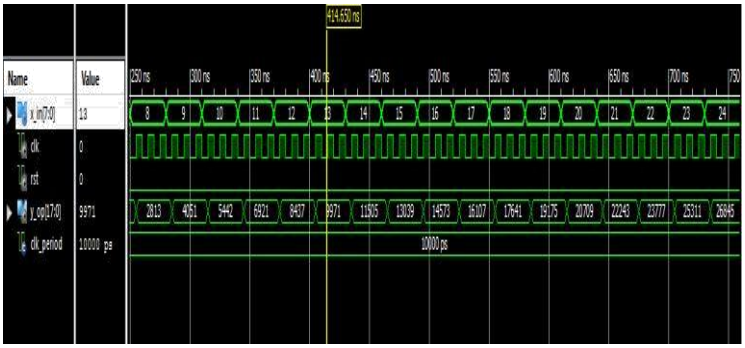

Fig.11: Simulation waveform for 12 Tap RMC FIR filter

6.2 Results

TABLE I: Comparison of Different Parameters for Concurrent, RMC and Linear Filter

Filter	**Concurrent FIR Filter**	**Reduced Multiplier Concurrent FIR Filter**	**Linear FIR Filter**
Logic path delay	23.569 ns	20.103 ns	13.149 ns
Number of clock pulses for one output	1	3	12 (Equal to number of taps)
Maximum Frequency	42.429MHz	16.5812MHz	6.337Mhz
Device Utilization	Slices:158 LUTs:268	Slices:130 LUTs:217	Slices:121 LUTs:109
Number of Multipliers	12	6	1

TABLE I. illustrates the improvement in speed in RMC filter compared to linear filter and improvement in device utilization and multiplier count in RMC filter compared to concurrent filter.

VII. Conclusion

Thus the proposed filter is optimized for better device utilization with a minimum deterioration of speed. Irrespective of number of taps, it can compute the output in 3 states unlike linear filters whose states depend on number of taps. Thus having high throughput and requiring half the number of multipliers compared to concurrent filter it saves precious chip area which can be later used for other peripheral purposes like building on chip ADC and DAC.

VIII. Acknowledgements

The authors thank Mrugendra Vasmatkar, V.E.S. Institute of Technology for providing the technical guidance for this research.

References

[1] Richard Lyons, Understanding Digital Signal Processing, vol. 1(Addison-Wesley Reading, MA, 1997)
[2] Alan Oppenheim and Ronald Schafer, Discrete-Time Signal Processing, vol. 1(Prentice Hall, EngleWood Cliffs, NJ, 1999).
[3] Vahid, Frank, Digital Design (Hobiken, NJ: Wiley, 2006. Print)
[4] Sonika Gupta, Aman Panghal, Performance Analysis of FIR Filter Design by Using Rectangular, Hanning and Hamming Windows Methods, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6, June 2012