

# Low Cost High Speed FIR Filter Design using MCMA Truncation Method

Pankaj Prakash Saha<sup>1\*</sup>, Rajendra Bhadur Singh<sup>2</sup>, Rohit Saini<sup>3</sup>

<sup>1</sup>Department of ECE, Gautam Buddha University, Greater Noida

<sup>2,3</sup>School of ICTI, Gautam Buddha University, Greater Noida

E-mail: <sup>1</sup>pan12saha@gmail.com, <sup>2</sup>rajendra@gbu.ac.in, <sup>3</sup>rohitsaini671@gmail.com

**Abstract**—This paper discussed to implement FIR filter using truncated method. This paper states the behavior of filter by using MCMA technique. The FIR architecture describes by using pipelining form and direct form of the filter. The non-pipeline and pipeline FIR filter have been designed using Hardware Description Language (HDL) and a comparative study of both the filter designs. The method of using pipelining technique is for the reduction of delay, power dissipation and reduces the area of the filter. The proposed filter uses truncation multiplier and designed using VHDL HDL language and synthesis using Xilinx simulator and simulate it using MODELSIM ALTERA6.4a (Quartus II 9.2 SP2). It is better than the previous design in an area, power and delay.

**Keywords:** FIR filter, Pipeline, MCMA technique.

## 1. INTRODUCTION

The systems discussed in this paper are finite impulse response (FIR) digital filters. The value is depend on the past, present and future value of the output. The basic FIR filter equation is

$$y(n) = \sum_{i=0}^{m-1} a_i x(n - i) \quad \dots 1$$

Where 'm' is the level of the filter order,  $a_i$  is the coefficient of the filter,  $x(n)$  is the input of the filter and  $y(n)$  is the output of the filter. A FIR filter is based on no feedback method, that's why it is more stable than IIR filter. There are two basic FIR structures, direct form and pipelined form as shown in Fig. 1 for a FIR filter. In the direct form in Fig. 1(a) the multiple constant multiplication (MCM) accumulation (MCMA) modules performs the multiplications of individual delayed inputs and respective filter coefficients, and store it to the next adder as accumulation method. By accumulation of all the products the final output is now obtained. The operand of the multipliers in multiple constant multiplications is delayed through the delay elements like flip flop or connecting wires. Input signals  $x(n - i)$  and coefficients  $a_i$  both get multiply and store in the data. In the pipelined form Fig. 1(b), it have current input signal  $x(n)$  and the coefficient  $a_i$  both get multiply by a parallel multiplier. Then the multiplied input and coefficient then sum by the SA (structural Adder) and delay elements. For avoiding expensive multipliers, most

prefer hardware implementations of digital FIR filters can be manage as two categories: multiplier less based and memory based.

Multiplier less-based designs realize MCM have shift and add operations and share the common sub operations like CSD recoding and like CSE to minimize the cost of adder of MCM. Memory-based Finite impulse response (FIR) designs consist of two types of approaches: lookup table methods and distributed arithmetic methods.

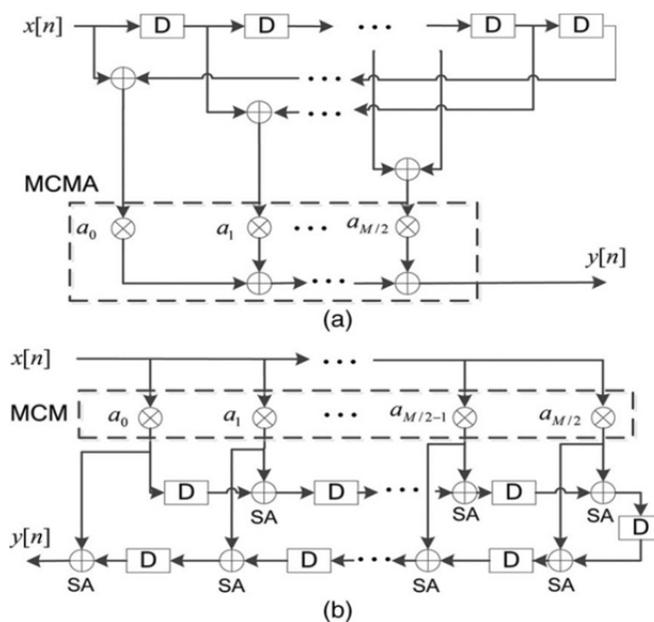


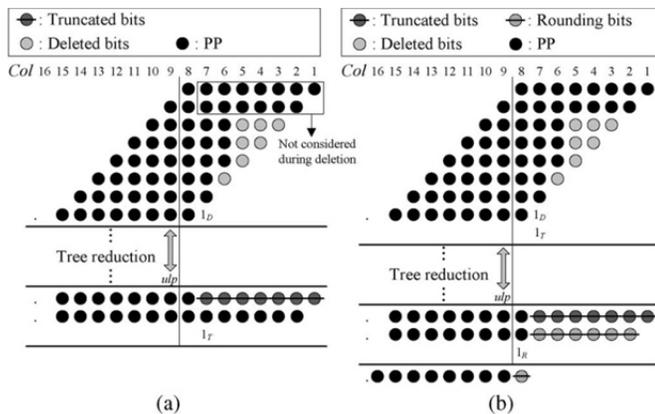
Fig. 1 (a) Direct form fir filter design (b) Indirect form of filter design

The look up table based design stores in secondary memory odd multiples of the inputs to realize the constant multiplications in MCM. The distributed arithmetic based approaches recursively stores the bit-level partial results for the inner product processing in finite impulse response (FIR) filtering.

We knew that in mathematical process the bit width is increase when multiply to number, many digital communication applications do not need full-precision outputs. However, it is acceptable to generate faithfully rounded outputs where the whole error introduced in quantization process and error will not more than one unit of the last place (ulp).

**2. TREE REDUCTION OF PARALLEL MULTIPLIERS**

Tree depended multiplier diagram is consists of following process Partial product generation, Partial product reduction, partial product deletion, rounding, truncation and last carry adder propagator. Partial Product generation means that it is the process of multiply two numbers and generates the partial product. The role of partial product deletion and reduction is to reduce the partial product and compress the bits. Because of these methods, it is really helpful for the reducing the area of the partial product. The reduced area can be achieved approx. 20% to 30% because of these methods. There are two famous tree reduction methods are Wallace tree and Dadda tree reduction methods.



**Fig.2. (a) Deletion, reduction, and truncation of PP bits. (b) Deletion, reduction, truncation, and rounding plus final addition. The above Fig. 2 is discussed below.**

**3. DELETION, REDUCTION AND TRUNCATION**

First step, we implement the deletion that decrease all the partial product bits that don't need to be generated, as shown as the gray dots in Fig. 2(a) for an example of 8 × 8 multiplication with eight product its truncated. In this step, we delete as many partial product bits as possible, for as long as deletion error  $E'_D$  is bounded by  $-\frac{1}{2}ulp \leq E'_D \leq 0$ . After giving 1/4 ulp, as shown in Fig. 2(a), the deletion error after the bias adjustment is

$$-\frac{1}{4}ulp \leq E_D \leq \frac{1}{4}ulp \quad \dots\dots 2$$

After the deletion of partial product bits, we implemented the per-column reduction and generate two rows of partial product

bits. After reduction, we implemented the truncation that further removes the first row of  $n - 1$  bits from column 1, as shown by the crossed gray dots in Fig.2. This step of truncation has truncation error as

$$\frac{1}{2}ulp \leq E_T \leq 0 \quad \dots\dots 3$$

in Fig.2, the adjusted truncation error is bounded

$$\text{by} \\ -\frac{1}{4}ulp \leq E_D \leq \frac{1}{4}ulp \quad \dots\dots 2$$

**4. ROUNDING AND FINAL ADDITION**

After reduction, truncation and deletion, the partial product bits are summed by using a CPA to generate the final product of partial bits, as in Fig. 2(b). Note that the bits in column 2 to column  $n - 1$  as crossed spotted dot in Fig. 2(b) can be easily removed before CPA because these bits are the only bits left behind in the columns after the truncation and deletion processes and thus, they don't affect the carry bit to column  $n + 1$  the column with a weighting of 1 unit last place (ulp) during the significant rounding process. we add a bias constant of 1/2 unit last place (ulp), which is shown as 1R in Fig. 2, in order to achieve nearest rounding with the rounding error.

$$-\frac{1}{2}ulp < E_R \leq \frac{1}{2}ulp \quad \dots\dots 4$$

The bits at column  $n$  after the final CPA is also deleted during the rounding process. hence, the total error for the designed system of the faithfully rounded truncated multiplier is bounded by

$$-ulp < E = (E_D + E_T + E_R) \leq ulp \quad \dots\dots 5$$

Note that the given truncated multiplier design achieves faithfully rounding because total error is not greater than 1 unit last place (ulp). The three bias constants, i.e., 1D, 1T, and 1R, can be combined as a single constant bit to be added at column  $n + 1$ , without increasing the height of the partial product matrix.

**5. FIR DIGITAL FILTER USING PIPELINING**

The pipelining leads to a decrease in the critical path, which can be removed to either increasing the clock speed or sampling speed or to reducing power consumption at same speed. In the parallel processing, various outputs are processed in parallel in clock duration. Hence, the effective sampling speed is getting increased by the order of parallelism. As to the pipelining/transposed, parallel processing may also be used for reducing of power consumption.

Consider the three tap FIR digital filter

$$y(n) = ax(n) + bx(n - 1) + cx(n - 2) \quad \dots\dots 6$$

The critical path here is limited by one multiplier and two adders .i.e.,  $T_M$  is the time taken for multiply and  $T_A$  is time for sum operation then the “sample period” ( $T_{sample}$ ) is given by,

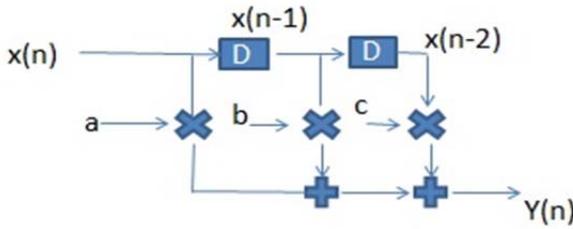


Fig. 3: Three tap FIR filter

$$T_{sample} \geq T_M + 2T_A \quad \dots\dots 7$$

Hence, the sample frequency ( $f_{sample}$ ) (also referred to as the throughput or the iteration rate) is given by

Notice that the direct-form structure shown in Fig. 3 may only

$$f_{sample} \leq \frac{1}{T_M+2T_A} \quad \dots\dots 8$$

be used when Fig. 4 will satisfy. But if some real time application demands a fast input rate (sample rate), then these structures may not be used. In that situation, the effective critical path can be removed by using either pipelining or parallel processing. Pipelining decreases the effective critical path by introducing pipelining flip flop (here we use D flip flop) along the data path. Pipelining has used in the context of architecture designs and compiler synthesis. Parallel processing increases the sample rate by duplicate hardware so that several inputs must be processed in parallel and several outputs can be obtained at the same time. Consider the simple structure in Fig. 4(a), where the computational time of the critical path is  $2T_A$ . Fig. 4(b) shows the second level pipelined structure, where one flip flop is placed between the 2 adders and hence the critical path is decreased by half. Its second level parallel processing structure is shown in Fig. 4(c), where the same design is duplicated so that 2 inputs can be processed at the concurrent time and 2 outputs are produced simultaneously. Hence, the sampling rate is increased by two.

Let consider the pipelined implementation of the three tap FIR filter of (Fig. 3) obtained by introducing two additional flip flops as shown in Fig. 5. The critical path is now decreased from  $T_M + 2T_A$  to  $T_M + T_A$ . In these connections while the left summer initiates the computation of the current iteration the right summer is processing the computation of the previous iteration result. The events for this system are shown in Table1. In this design, at any time, 2 simultaneously outputs are calculated in an interleaved manner.

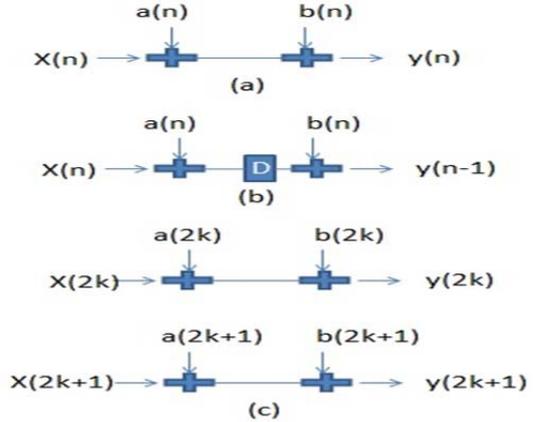


Fig. 4 (a) A data path. (b) The 2-level pipelined structure of (a). (c) The 2-level parallel processing structure of (a).

One must note that in an M-level pipelined system, the number of delay elements in any path from input to output is  $m - 1$  greater than that in the actual path in the original same clock circuit. While pipelining decreases the critical path, it leads to increase in the terms of latency. Latency is the difference between the availability of the first output data in the pipeline system and the same clock system. For example if latency is 1 clock cycle then the  $n^{th}$  output is available in  $(n + 1)^{th}$  clock period in a first stage pipelined system. The two drawbacks of the pipelining are increasing in the number of flip flop and in circuit latency.

The below points may be noted:

- How faster the architecture is reduced by the longest path between any two flip flops or between an input and a flip flop or between a flip flops and an output or between the input and the output.
- This “critical path” may be decreased by smartly placing the pipelining flip flops in the design architecture.

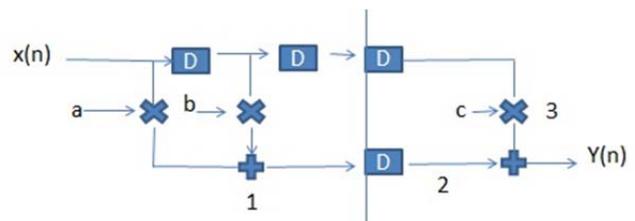


Fig. 5: Pipelined FIR filters. The dashed vertical line represents a feed-forward cutset.

C	Input	Node1	Node2	Node3	o/p
0	x(0)	ax(0)+bx(-1)	-	-	-
1	x(1)	ax(1)+bx(0)	ax(0)+bx(-1)	cx(-2)	y(0)
2	x(2)	ax(2)+bx(1)	ax(1)+bx(0)	cx(-1)	y(1)
3	x(3)	ax(3)+bx(2)	ax(2)+bx(1)	cx(0)	y(2)

Table 1.events occurs in pipelined FIR filter of Fig. 5

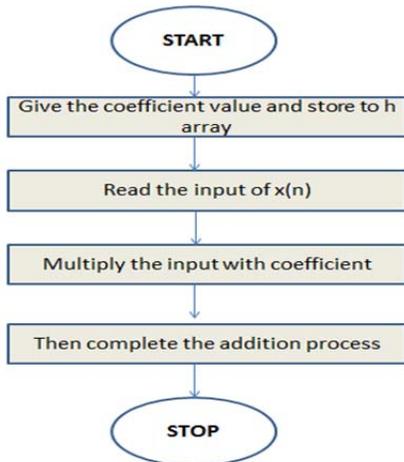


Fig. 6: Flow chart for designing the FIR Filter

6. EXPERIMENTAL RESULT

The given result has found by using MCMAT technique. The Multiple Constant Multiplication or Accumulation technique (MCMAT) is a type of technique which reduces the netlist of the design. This technique is really helpful for getting a high speed operational circuit. The technique is also helpful for reducing the dissipation power and delay of the circuit.

The following result is implemented on Stratix III and software is Quartus II 9.1 SP2 for finding power dissipation result and for delay, we are using Xilinx ISE software.

Table 2: Comparision of Power dissipation and delay

S · N o ·	Name (power dissipation)	4 tap filter (unit in mw)		5 tap filter (unit in mw)	
		Direct	Pipeline	Direct	Pipeline
1	Thermal power	399.37	397.94	399.23	398.76
2	Static power	371.10	370.15	370.96	370.49
3	I/O power	28.27	27.79	28.27	28.27
4	Delay(ns)	9.78	4.77	12.672	5.383

From the above table we can see that the power dissipation and delay comparison between tap 4 filter and tap 5 filter. As stated above the pipeline form is much better than the direct form of FIR filter circuit design.

Table 3: Synthesis report of tap 4 and tap 5

s · n	Logic Utilization	4 tap filter		5 tap filter	
		direct	pipeline	direct	pipeline
1	Slice FF	72	55	88	92
2	4 input LUTs	96	46	120	91
3	No. of Occupied Slices	77	33	97	48
4	Total No. of 4 input Slices	96	48	120	91
5	No. of IOBs	41	25	41	41
6	Avg.Fan out	1.91	1.63	1.76	1.50

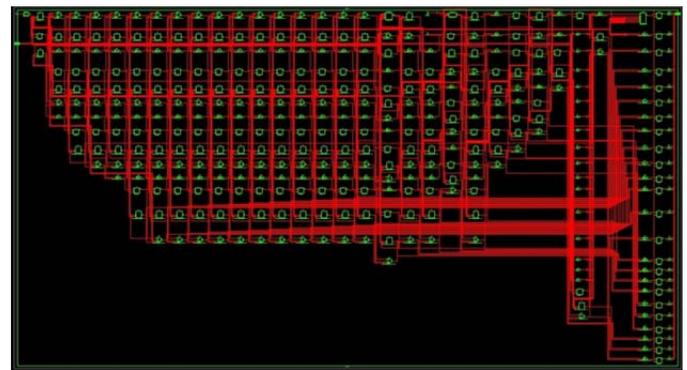


Fig. 7: Schematic of Pipeline FIR filter

In the above result is the comparison between direct form and pipelining form of the FIR circuit .Because of the use of the pipeline circuit, it reduces the delay in path and made a fast performance circuit .MCMA is a baseline implementation using partial product compression with uniformly quantized coefficient. However the area of the proposed circuit designs are significantly reduced, the critical path delay is increased because all the operations in the MCMA are executed within single clock cycle period. It is possible to reduce the delay by adding pipeline registers in the partial pressure compression.

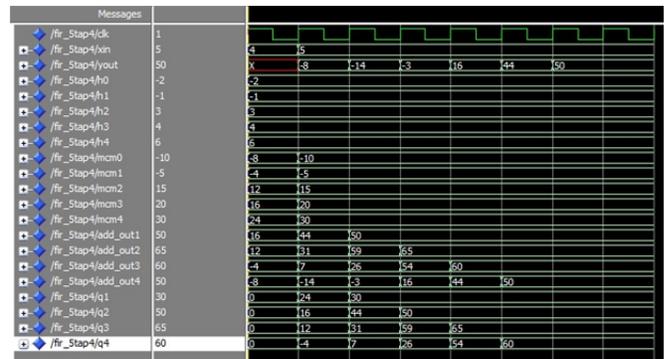


Fig. 8: Simulation Result of pipeline FIR filter

This simulation has been done by using Altera ModelSim 6.2a Starter edition.

## 7. CONCLUSION

The design of direct and pipelined FIR filter using MCMA technique has been accomplished via Hardware Description Language synthesized on XILINX ISE Software (Xilinx ISE 13.1 version) for Spartan 3E FPGA (Field Gate Programmable Array) family (XC3S1600E). Synthesis reports are shown in Table 3 and power dissipation analysis in Table 2. The Fig. 5 demonstrates that the pipelined technique enhances the speed and reduces delay from 12.672 ns to 5.383 ns in Tap 5 filter.

Simulation waveforms are shown in Fig. 4 – Fig. 8. The top RTL level symbol of proposed FIR filter is shown in Fig. 7. As all the structures are implemented using VHDL language, they can be ported to any FPGA family.

## 8. ACKNOWLEDGEMENT

The author would like to thank Mr. Rajendra Bahadur Singh for his complete guidance and entire VLSI department for their support.

## REFERENCES

- [1] F. Xu, C. H. Chang, and C. C. Jong, "Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common sub expressions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol.26, no. 10, pp. 1898–1907, Oct. 2007.
- [2] "Low-Cost FIR Filter Designs Based on Faithfully Rounded Truncated Multiple Constant multiplication/Accumulation" Shen-Fu Hsiao, Jun-Hong Zhang Jian, and Ming-Chih Chen.
- [3] D. Shi and Y. J. Yu, "Design of linear phase FIR filters with high probability of achieving minimum number of adders," *IEEE Trans.Circuits Syst. I, Reg. Papers*, vol. 58, no. 1, pp. 126–136, Jan. 2011.
- [4] A.Deepika, A.Bhuvaneswari," Low Power Fir Filter Design Using Truncated Multiplier," *International Journal of Engineering Trends and Technology (IJETT) – Volume 10 Number 1 - Apr 2014*.
- [5] Y. Voronenko and M. Puschel, "Multiplier less multiple constant multiplication," *ACM Trans. Algorithms*, vol. 3, no. 2, pp. 1–38, May 2007.
- [6] R. Huang, C.-H. H. Chang, M. Faust, N. Lotze, and Y. Manoli, "Sign extension avoidance and word-length optimization by positive-offset representation for FIR filter design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 12, pp. 916–920, Oct. 2011.
- [7] B. Rashidi, B. Rashidi and M. Pourormazd, "Design and Implementation of Low Power Digital FIR Filter based on low power multipliers and adders on Xilinx FPGA", *International Conference on Electronics Computer Technology*, 2011, p.18-22.
- [8] J.-A. Pineiro, S. F. Oberman, J. M. Muller, and J. D. Bruguera, "Highspeed function approximation using a inimax quadratic interpolator," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 304–318, Mar. 2005.
- [9] N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G.M. Strollo, "Truncated binary multipliers with variable correction and minimum mean square error," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 6,pp. 1312–1325, Jun. 2010.