Performance Analysis of DA FIR filter on Various FPGA Family

Sejal D. Patel
Department of Instrumentation & Control Engineering
L.D. College of Engineering
Ahmedabad

Prof. M.C. Patel
Assistant Professor
Department of Instrumentation & Control Engineering
L.D. College of Engineering
Ahmedabad

***Abstract* – Now a day, Implementation of FIR filter is one of the Challenging Task and growing fields in the area of Very Large Scale Integrated Circuits (VLSI).As getting tight transition width order of filter higher and require more hardware resources. DA algorithm is popular for saving hardware resources because it replace multiplier with LUTs. This paper deals with implement modified DA algorithm for 8-tap FIR filter .Simulation is done on xlink 14.2 and check it performance indices for various devices in terms of speed and area.**

Keywords – FIR Filter, MAC operation, DA algorithm, Area, Speed, Time delay

I. INTRODUCTION

Digital filtering is one of the most versatile tools of DSP.As advantage over the analog filter that it consist of error associated with passive component such as temperature, drift etc. In addition, the characteristics of a digital filter can be easily changed under software Control. So, they are widely used in adaptive filtering applications in communications such as echo cancellation in modems, noise cancellation, and Speech recognition. Digital filter are basically characterized by two types that is Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). There are lots of advantage such as linearity and stability are most basic property of FIR filter that is popular over IIR filter.[11] But one drawback of FIR filter is that for getting desired frequency response means narrow transition width as we know that this width is inversely proportional to order of filter so achieving this criteria order of filter is higher and ultimately it required more Area for implementation on embedded platform. So in many application we require phase linearity such as speech recognition then it must to use FIR filter and keep in mind area constraint. If phase linearity is not issue then it is better recommended to use IIR filter because it achieve desired frequency response with less order so no area optimization issue. FPGA implementation include higher sampling rates than are available from traditional DSP chips, lower costs than an ASIC for moderate volume applications, and more flexibility than the alternate approaches. In particular, several multiply-accumulate (MAC) units may be implemented on a single . As there is challenging task to implement FIR filter on embedded platform with area constraint. There are lots of research technique available for implementing FIR filter on embedded platform. Most of research are available on Field Programming Gate Array (FPGA). The advantages of the FPGA approach to digital filter implementation include higher sampling rates than are available from traditional DSP chips, lower costs than an ASIC for moderate volume applications, and more flexibility than the alternate approaches. In particular, multiple multiply-accumulate (MAC) units may be implemented on a single FPGA, which provides comparable performance to general-purpose architectures which have a single MAC unit. The design of FIR Filter involves multipliers and adders (MAC) which consumes an efficient technique for calculation of sum of products or vector dot product or inner product or multiply and accumulate (MAC) MAC operation is very common in all Digital Signal Processing Algorithms. Distributed Arithmetic was first brought up by Croisier , and was extended to cover the signed data system by Liu , and then was introduced into FPGA design to save MAC blocks with the development of FPGA technology.[10][13].

This paper is organized as follows. Section II presents Basic DA algorithm Section III modified DA algorithm for FIR filter. Section IV Simulation result. Section V Conclusion

II. BASIC DA ALGORITHM

If the length of the impulse response is finite, the filter is an FIR (finite impulse response) filter. Otherwise, the filter is an IIR (infinite impulse response) filter.For a FIR filter of order *N*, each value of the output sequence is a weighted sum of the most recent input values**:** [2]

$$Y[n] = \sum_{i=0}^{N} h_i . x[n-i] \quad \ldots\ldots\ldots(1)$$

All Rights Reserved © 2017 IJSETR

Where X[n] = input signal Y[n] = output signal. Where h[n] is the filter coefficient and x[n] is the input sequence to be processed. The FIR structure consists of a series of Multiplication and addition units, and consume N MAC blocks of FPGA, which are expensive in high speed system. Compared with traditional direct arithmetic, Distributed Arithmetic can save considerable hardware resources through using LUT to take the place of MAC units [2]. Another virtue of this method is that it can avoid system speed decrease with the increase of the input data bit width or the filter coefficient bit width, which can occur in traditional direct method and consume considerable hardware resources [2].

Distributed Arithmetic is introduced into the design of FIR filters as follows.

In the two's complement system, x[n] can be described as:

$$x[n] = -2^B x_B[n] + \sum_{b=0}^{B-1} 2^b x_b[n] \ldots (2)$$

Substitute this eq (2) in eq(1)

$$y[n] = -2^B x_B[n]h[n] + \sum_{b=0}^{B-1} h[n] \sum_{n=0}^{N-1} 2^b x_b[n] .. (3)$$

Final Form of Distributed arithmetic is :

$$y[n] = -2^B x_B[n]h[n] + \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} h[n]x_b[n] .. (4)$$

Now as observing right part of equation (4) we can store the possible combination of h[n] and every bit of input signal x[n] that is indicated as xb[n] is stored to LUT unit and call out relevant data according to input to save MAC blocks[10] And then the weighted sum of is calculated through shift Registers. In signed system, the signed bit should be taken into consideration so is also added. As a result, the final form of Distributed Arithmetic is define as equ. (4) And the implementation as shown in Fig 1[2] can be achieved on FPGA through LUT units.

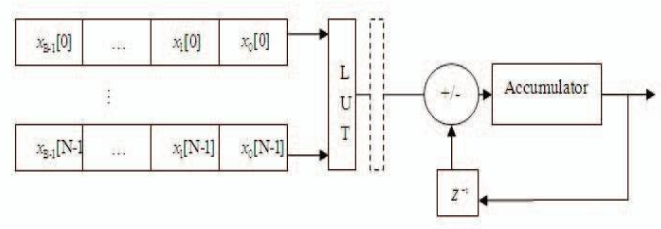

Fig: 1 DA basic structure

So according to combination of different input signal and filter coefficient value precalculated sum stored to LUT and depending upon combination of different input signal value is fetched and passed to accumulator until all bit are processed.

III MODIFIED DA ALGORITHM

As from section II we seen that basic DA algorithm which is serial in nature means each output bit processed in once clock cycle so if only area is prime concert then it's good to use serial DA algorithm.In basic DA algorithm computations are bit-serial in nature, i.e., each bit of the input samples must be indexed in turn before a new output sample becomes available. When the input samples are represented with ***B*** bits require ***B*** clock cycles are required to complete an inner-product calculation [1].but in most of VLSI design time delay play an important roleIn the delay of design is arrived by design delay and routing delay. Earlier design delay play dominant but now a days routing delay dominates more.A parallel realization of distributed arithmetic corresponds to allowing multiple input from LUT to be processed in one clock cycle.In sequential manner they require for an example 4 bit input 4 clock cycle where as in parallel require only 1 clock cycle. By above modification done in basic DA we can balance between time delay and Area.so in this paper focus on parallel architecture.

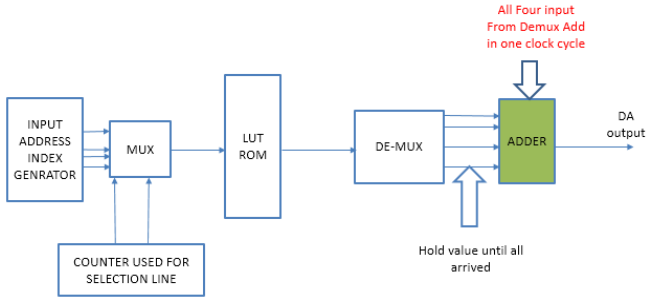

Fig 2: Block diagram of modified parallel DA technique

As shown in Fig 2 parallel DA technique in which MUX select the address generation from input sample and counter is used for selection of address line. LUT RAM or LUT ROM used for storing the value of partial product according to combination of address of input sample. Depending upon combination of input sample value from LUT is fetched and passed to demux that hold the value until all not arrived then add all value in one clock cycle. For example if 4 bit input sample take four clock cycle in serial DA where as in proposed DA algorithm 4 bit input sample whatever the tap of FIR filter there are 4 values from LUT that added in one clock cycle. In this paper 8-tap DA simulation result and synthesis report shown. For 8-tap DA LUT used as RAM in which coefficient of FIR are continuously changed so LUT is updated and store new value its dynamic in nature. For this type of structure performance is check for various device.As shown in Fig 3 shows simulation result of 8-tap DA for proposed method .

All Rights Reserved © 2017 IJSETR

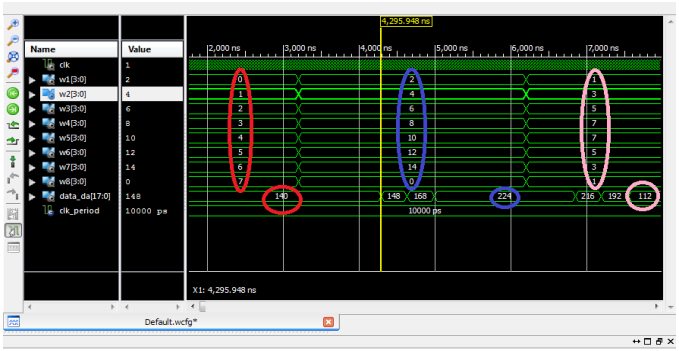

Fig 3 Simulation Result of 8-tap DA

Fig 4 shows that we used LUT as RAM so for that one clock cycle is used LUT update and one clock cycle for LUT output addition hence require two clock cycle where as in basic DA algorithm output are available after 5 clock cycle.

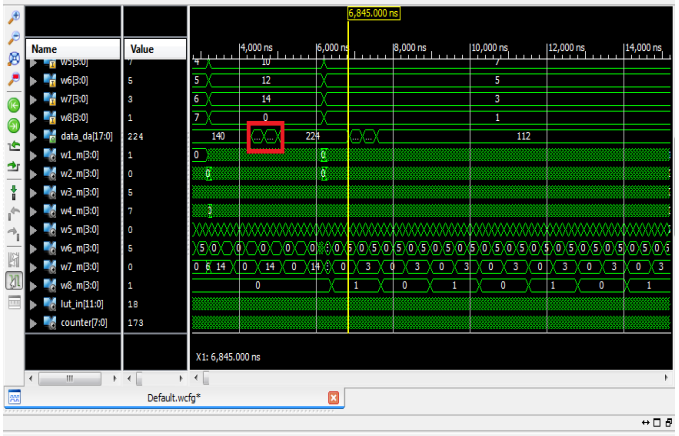

Fig 4: Output of 8-tap DA available after two clock cycle

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	42	12480	0%
Number of Slice LUTs	111	12480	0%
Number of fully used LUT-FF pairs	26	127	20%
Number of bonded IOBs	51	172	29%
Number of Block RAM/FIFO	2	26	7%
Number of BUFG/BUFGCTRLs	1	32	3%

Fig 5: Virtex 5 device utilization summry for proposed method

```
Minimum period: 5.482ns (Maximum Frequency: 182.414MHz)
Minimum input arrival time before clock: 6.020ns
Maximum output required time after clock: 2.826ns
Maximum combinational path delay: No path found
```

Fig 6: Virtex 5 Timing analysis for propose method

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	39	4800	0%
Number of Slice LUTs	99	2400	4%
Number of fully used LUT-FF pairs	29	109	26%
Number of bonded IOBs	51	102	50%
Number of Block RAM/FIFO	2	12	16%
Number of BUFG/BUFGCTRLs	1	16	6%

Fig 7: Spartan 6 deviceutilization summry for proposed method

```
Minimum period: 5.321ns (Maximum Frequency: 187.936MHz)
Minimum input arrival time before clock: 6.879ns
Maximum output required time after clock: 3.597ns
Maximum combinational path delay: No path found
```

Fig 8: Spartan 6 Timing analysis for propose method

Device Utilization Summary (estimated values)			
ic Utilization	**Used**	**Available**	**Utilization**
ber of Slices	61	4656	1%
ber of Slice Flip Flops	42	9312	0%
ber of 4 input LUTs	113	9312	1%
ber of bonded IOBs	51	232	21%
ber of BRAMs	4	20	20%
ber of GCLKs	1	24	4%

Fig 9: spartan 3E deviceutilization summry for proposed method

```
Minimum period: 13.795ns (Maximum Frequency: 72.490MHz)
Minimum input arrival time before clock: 14.100ns
Maximum output required time after clock: 4.283ns
Maximum combinational path delay: No path found
```

Fig 10 : Spartan 3E Timing analysis for propose method

Table 1: Synthesis report summary for 8-tap DA FIR filter on various FPGA device.

Parameter	Virtex5	Spartan 6	Spartan 3E
No of Slices	42/12480	39/4800	61/4656
Minimum period in ns	5.482	5.321	13.79
Maximum clock frequency in MHz	182.414	187.936	72.490

Table 1: Summery for synthesis report of various device

IV. CONCLUSION

MAC operation is common for implementing FIR filter. But due to its implementation require more Area and more challenging task is when implement on FPGA because of limited resources in it. Where DA algorithm is efficient technique for implementing FIR filter on FPGA. In DA algorithm Main issue regarding to growth of LUT size as order of filter increases so using different modification done in DA algorithm get advantage over MAC operation in terms of Area. Time delay is also issued in basic DA because of Accumulator So by using Parallel structure of DA we will get balance structure of FIR filter in terms of Area and time delay.8-tap DA filter using proposed method get good performance in terms of area and speed for various FPGA family.It is very good for Spartan 6.

V. FUTURE SCOPE

Proposed method can be implemented on any FPGA device and also test for higher order filter.

All Rights Reserved © 2017 IJSETR

REFERENCES

[1] T Pitchaiah Dhanalakshmi M PV shreedevi,' FPGA Implementation of Low Area and Delay Efficient Adaptive Filter Using Distributed Arithmatic' IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), August 01-02, 2014, Dr. Virendra Swarup Group of Institutions, Unnao, India

[2] Yajun Zhou, Pingzheng Shi,'. Distributed Arithmetic for FIR Filter implementation on FPGA '978-1-61284-774-0/11/$26.00 ©2011 IEEE

[3] V.Sudhakar, N.S.Murthy and L.Anjaneyulu, "Area efficient pipelined Architecutre for realization of FIR filter using Distributed Arithmetic", ICIII-2012, pp 169-173.

[4] M. A. M. Eshtawie and M. Othman, On-line DA-LUT architecture for high-speed high-order digital FIR filters, in: Proceedings of the IEEE International Conference on Communication Systems (ICCS), Singapore, November. 2006,

[5] D.J.Allred,H.Yoo,V.Krishnan,W.Huang and D.V.Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," IEEE Transcations on circuits and systems,2005,vol.52,pp.1327-1337.

[6] C.S. Avinash1 and John Sahaya Rani Alex2 'FPGA Implementation of Discrete Wavelet Transform using Distributed Arithmetic Architecture 2015' International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, T.N., India. 6 - 8 May 2015. pp.326-330

[7] S. M. Badave and A. S. Bhalchandra,'Multiplier less FIR filter Implementation on FPGA'*International Journal of Information and Electronics Engineering, Vol. 2, No. 2, March 2012*.

[8] A.cervin, M.ohlin and D.Henori Ksson, Simulation of networked control systems using true time, In proc 3rd International workshop on networked control systems, tolerant to faults,2007.

[9] R. Guo and L. S. DeBrunner, "Two high-performance adaptive filter Implementation scheme using distributed arithmetic,"IEEE TRANS Circuits system II*Exp. Briefs*, vol. 58, no. 9 pp. 600–604, Sep. 2011.

[10] Uwe Mayer - Baese. "Digital Signal Processing with field Programming Gate Array,"springer,2007

[11] Clive Max Filed, The Design.Warriors Guide to FPGA,Elsevier,2004

All Rights Reserved © 2017 IJSETR