IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 7, Issue 2 (Jul. - Aug. 2013), PP 13-18
www.iosrjournals.org

A Novel Approach of Area-Efficient FIR Filter Design Using Distributed Arithmetic with Decomposed LUT

[1]P.Sravanthi, [2]CH.Srinivasa Rao, [3]S.Madhava Rao,
[1]M.TECH student, MLE College of Engineering, S.Konda, prakasam(DT), A.P,
[2,3]ASSOC.Professor, MLECollege, singarayakonda, prakasam, A.P, INDIA,

Abstract: *In this paper, a highly area-efficient multiplier-less FIR filter is presented. Distributed Arithmetic (DA) has been used to implement a bit-serial scheme of a general asymmetric version of an FIR filter, taking optimal advantage of the 3-input LUT-based structure of FPGAs. The implementation of FIR filters on FPGA based on traditional arithmetic method costs considerable hardware resources, which goes against the decrease of circuit scale and the increase of system speed. This paper presents the realization of area efficient architectures using Distributed Arithmetic (DA) for implementation of Finite Impulse Response (FIR) filter. The performance of the bit-serial and bit parallel DA along with pipelining architecture with different quantized versions are analyzed for FIR filter Design. Distributed Arithmetic structure is used to increase the resource usage while pipeline structure is also used to increase the system speed. In addition, the divided LUT method is also used to decrease the required memory units. However, according to Distributed Arithmetic, we can make a Look-Up-Table (LUT) to conserve the MAC values and callout the values according to the input data if necessary. Therefore, LUT can be created to take the place of MAC units so as to save the hardware resources. The simulation results indicate that FIR filters using Distributed Arithmetic can work stable with high speed and can save almost 50 percent hardware resources to decrease the circuit scale, and can be applied to a variety of areas for its great flexibility and high reliability. This method not only reduces the LUT size, but also modifies the structure of the filter to achieve high speed performance.*

Keywords: *DSP, Digital Filters, FIR , FPGA, MAC, Distributed Arithmetic(DA),Divided LUT, pipeline*

I. Introduction

In the recent years, there has been a growing trend to implement digital signal processing functions in Field Programmable Gate Array (FPGA). In this sense, we need to put great effort in designing efficient architectures for digital signal processing functions such as FIR filters, which are widely used in video and audio signal processing, telecommunications and etc.

Traditionally, direct implementation of a K-tap FIR filter requires K multiply-and-accumulate (MAC) blocks, which are expensive to implement in FPGA due to logic complexity and resource usage. To resolve this issue, we first present DA, which is a multiplier-less architecture.

Implementing multipliers using the logic fabric of the FPGA is costly due to logic complexity and area usage, especially when the filter size is large. Modern FPGAs have dedicated DSP blocks that alleviate this problem, however for very large filter sizes the challenge of reducing area and complexity still remains. An alternative to computing the multiplication is to decompose the MAC operations into a series of lookup table (LUT) accesses and summations. This approach is termed distributed arithmetic (DA), a bit serial method of computing the inner product of two vectors with a fixed number of cycles.

The original DA architecture stores all the possible binary combinations of the coefficients w[k] of equation (1) in a memory or lookup table[5,7,9]. It is evident that for large values of L, the size of the memory containing the pre computed terms grows exponentially too large to be practical. The memory size can be reduced by dividing the single large memory (2Lwords) into m multiple smaller sized memories each of size 2k where $L = m \times k$. The memory size can be further reduced to $2L-1$ and $2L-2$ by applying offset binary coding and exploiting resultant symmetries found in the contents of the memories[8,9].

This technique is based on using 2's complement binary representation of data, and the data can be pre-computed and stored in LUT. As DA is a very efficient solution especially suited for LUT-based FPGA architectures, many researchers put great effort in using DA to implement FIR filters in FPGA. Patrick Longa introduced the structure of the FIR filter using DA algorithm and the functions of each part. Sangyun Hwang analyzed the power consumption of the filter using DA algorithm[3,7]. Heejong Yoo proposed a modified DA architecture that gradually replaces LUT requirements with multiplexer/adder pairs. But the main problem of DA is that the requirement of LUT capacity increases exponentially with the order of the filter, given that DA implementations need 2Kwords (K is the number of taps of the filter)[4,5,10]. And if K is a prime, the hardware resource consumption will cost even higher. To overcome these problems, this paper presents a hardware-efficient DA architecture.

This method not only reduces the LUT size, but also modifies the structure of the filter to achieve high speed performance. The proposed filter has been designed and synthesized with ISE 7.1, and implemented with a 4VLX40FF668 FPGA device. [2] Our results show that the proposed DA architecture can implement FIR filters with high speed and smaller resource usage in comparison to the previous DA architecture.

II. Design Methodology

2.1 Distributed Arithmetic FIR filter architecture

Distributed Arithmetic is one of the most well-known methods of implementing FIR filters. The DA solves the computation of the inner product equation when the coefficients are pre knowledge, as happens in FIR filters.

An FIR filter of length K is described as:

$$y[n] = \sum_{k=0}^{K-1} h[k]x[n-k] \text{ --(1)}$$

Where h[k] is the filter coefficient and x[k] is the input data. For the convenience of analysis, x'[k] =x [n - k] is used for modifying the equation (1) and we have:

$$y = \sum_{k=0}^{K-1} h[k].x'[k] \text{ .....(2)}$$

Then we use B-bit two's complement binary numbers to represent the input data:

$$x'[k] = -2^B \cdot x_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^b \text{ -(3)}$$

Where $x_b[k]$ denotes the b'th bit of x[k], $x_b[k] \in$ {0, 1}.Substitution of (3) into (2) yields:

$$y = \sum_{k=0}^{K-1} h[k] \cdot (-2^B \cdot x_B[k] + \sum_{b=0}^{B-1} x_b[k] \cdot 2^b)$$

$$= -2^B \cdot \sum_{k=0}^{K-1} h[k] \cdot x_B[k] + \sum_{b=0}^{B-1} 2^b \cdot \sum_{k=0}^{K-1} h[k] \cdot x_b[k]$$

$$= -2^B \cdot f(h[k], x_B[k]) + \sum_{b=0}^{B-1} 2^b \cdot f(h[k], x_b[k]) \text{ - (4)}$$

We have

$$f(h[k], x_b[k]) = \sum_{k=0}^{K-1} h[k] \cdot x_b[k]$$

In equation (4), we observe that the filter coefficients can be pre-stored in LUT, and addressed by xb = $[x_b[0], x_b[1], \ldots x_b[K-1]]$.This way, the MAC blocks of FIR filters are reduced to access and summation with LUT.

The implementation of digital filters using this arithmetic is done by using registers, memory resources and a scaling accumulator.

Original LUT-based DA implementation of a 4-tap (K=4) FIR filter is shown in Figure 2.1. The DA architecture includes three units: the shift register unit, the DA-LUT unit, and the adder/shifter unit.

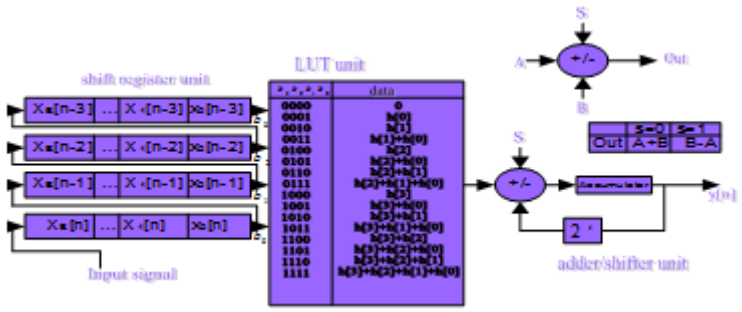


Figure 2.1: Original LUT-based DA implementation of a 4-tap filter

2.2 Distributed Arithmetic Architecture With Out Lut:

In Fig. 2.1, we can see that the lower half of LUT (locations where b_3=1) is the same with the sum of the upper half of LUT (locations where b_3=0) and h [3]. Hence, LUT size can be reduced 1/2 with an additional **2x1 multiplexer and a full adder, as shown in Figure 2.2.2(a).**

By the same LUT reduction procedure, we can have the final LUT-less DA architectures, as shown in Figure 2.2.2(b). On other side, for the use of combination logic circuit, the filter performance will be affected. But when the taps of the filter is a prime, we can use 4-input LUT units with additional multiplexers and full adders to get the trade off between filter performance and small resource usage.

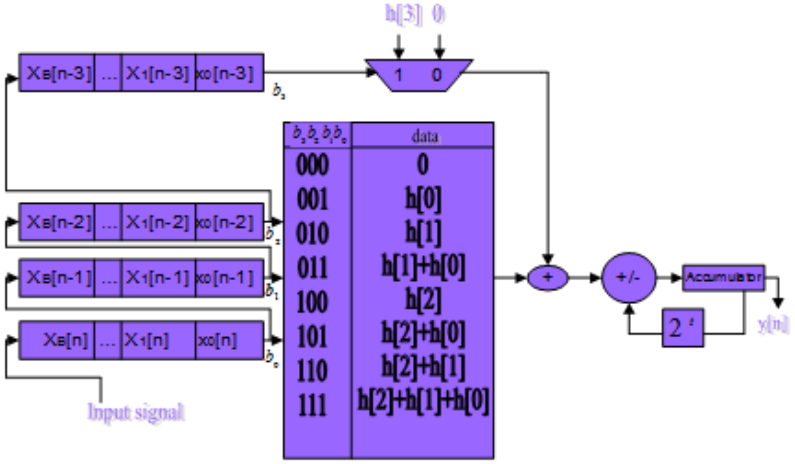


Figure 2.2.2(a): Proposed DA architecture for a 4-tap filter (2^3 word LUT implementation of DA)

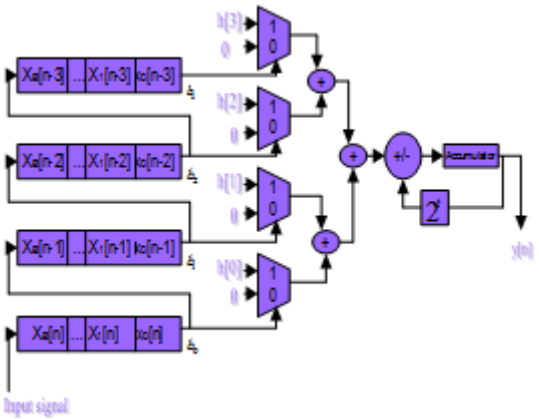


Figure 2.2.2(b): LUT-less DA architectures

III. Distributed Arithmetic With 4-Input Look Up Table Architecture

The main advantage of FIR filters is their linear-phase response. It is only achieved with symmetrical or ant symmetrical structures which, furthermore, offer hardware reductions.

As the filter order K increases, the LUT size (2^K words) grows exponentially. With this it is difficult to implement the architecture using lut less method as more and more combinational logic is needed. This resulted in a new architecture which is explained below.

3.1 Four input look up table architecture:

As the filter order K increases, the LUT size (2^K words) grows exponentially. We can divide the K-tap FIR filter into L groups of smaller filters (K = L *M). Hence the LUT size is reduced to $L \times 2^M$ words, and then we have:

$$y = \sum_{k=0}^{LM-1} h[k] \cdot x[k] = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} h[lM+m] \cdot x[lM+m]$$

We would better set M=4, because only under this condition can we take optimal advantage based on the 4-input LUT structure of FPGA, as shown in Fig 3.1. If we can not obtain M=4 after the calculation, we can use both the proposed DA-LUT unit and the original 4-input DA-LUT unit to achieve our goal.

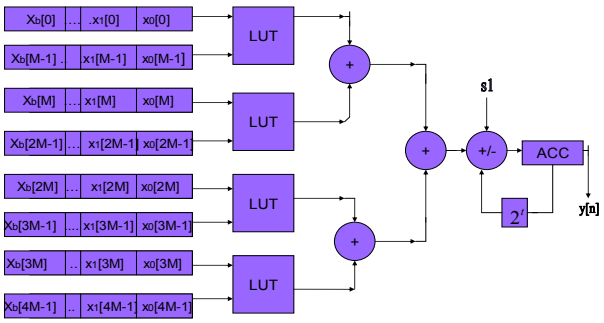
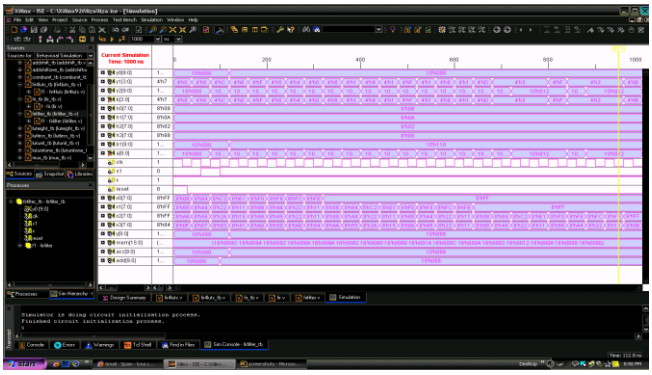


Figure 3.1 DISTRIBUTED ARITHMETIC WITH 4-INPUT LOOK UP TABLE

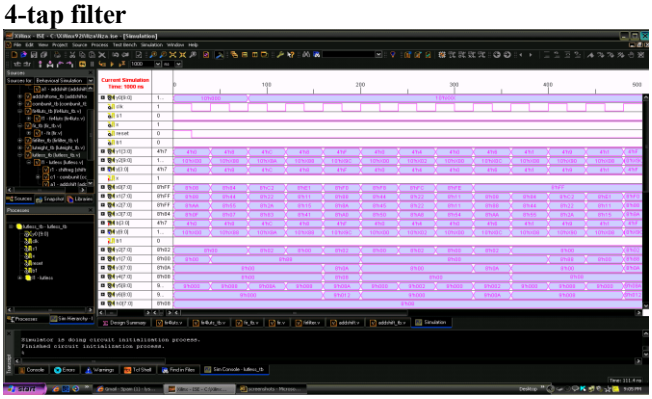
IV. Simulation Results:

Matlab and Modelsim are used as the simulation platforms. We can analysis the changes between the input wave and the output wave to observe the permanence of the designed filter through Matlab , while observing the real-time implementation performance of FPGA through Modelsim.

To observe the performance of the designed filter, a square signal of 1.6MHz is generated as the input by Matlab, and the sampling frequency is 32MHz. The data is changed into the two's complement form and stored in the file named fir_in.txt. And the data is called out later as the input of the FPGA filter through testbench language in Modelsim, the output data is shown in the waveform workstation and also stored into the file, which can be read by Matlab to make a contrast to analysis.



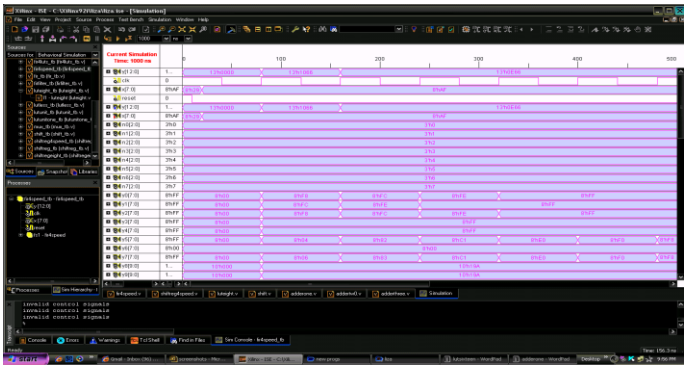
Original LUT-based DA implementation of a 4-tap filter



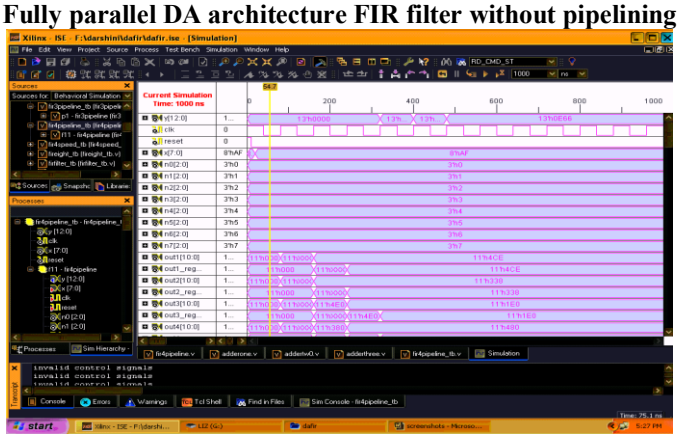
4-tap LUT-less DA architectures for a 4-tap FIR filter

4-input look up table architecture for high order filters Without pipelining

4-input look up table architecture with pipelining



Fully parallel DA architecture FIR filter without pipelining



Fully parallel DA architecture FIR filter with pipelining

V. Conclusion And Future Scope

This paper presents the proposed DA architectures for high-order filters. The architectures reduce the memory usage by half at every iteration of LUT reduction at the cost of the limited decrease of the system frequency. We also divide the high-order filters into several groups of small filters; hence we can reduce the LUT size also. As to get the high speed implementation of FIR filters, a full-parallel version of the DA architecture is adopted.

We have successfully implemented a high-efficient 70-tap full-parallel DA filter, using both an original DA architecture and a modified DA architecture on a 4VLX40FF668 FPGA device. It shows that the proposed DA architectures are hardware efficient for FPGA implementation. The limitation of this project is that by the use of pipelining speed is increased however, the resource usage increase more than B times of the bit-serial one's, since we need to use pipeline registers and additional adders. Also the number of LUT's required would be enormous. The future scope of this project is to improve the architecture of the Distributed arithmetic FIR filter such that it uses the hardware resources of the latest FPGA families. In vertex-5 and Vertex-6 family FPGA's, 6-input LUT's were introduced. Future work includes changing the architecture which uses 6-input

LUT's for storing coefficient sums and SRL(Shift register logic) macros to implement shift operations such that total number of slices used will be reduced.

References

[1]. Partrick Longa, Ali Miri, "Area-Efficient Fir Filter Design on FPGAs using Distributed Arithmetic" IEEE International Symposium on Signal Processing and Information Technology, pp:248-252,2006

[2]. Sangyun Hwang, Gunhee Han,Sungho Kang, Jaeseok Kim, "New Distributed Arithmetic Algorithm for Low-Power FIR Filter Implementation", IEEE Signal Processing Letters, Vol.11, No5, pp:463-466,May, 2004.

[3]. Heejong Yoo, David V.Anderson, "Hardware-Efficient Distributed Arithmetic Architecture For High-order Digital Filters", IEEE International Conference on Acoustics, Speech and Signal Processing, Vol.5,pp: 125-128,March,2005

[4]. Wangdian, Xingwang Zhuo "Digital Systems Applications and Design Based On Verilog HDL", Beijin: National Defence Industry press, 2006.

[5]. McClellan , J.H. Parks, T.W. Rabiner, L.R. "A computer program for designing optimum FIR linear phase digital filters":. IEEE Trans. Audio Electroacoust. Vol. 21, No.6, pp:506-526, 1973. http://en.wikipedia.org/wiki/instruction_pipeline

[6]. Httl://cse.stanford.edu/class/sophomore-college/projects-00/risc/pipelining/index.html we Meyer-Baese.Digital signal processing with FPGA[M].Beijing:Tsinghua University Press,2006:50~51

[7]. Tsao Y C and Choi K. Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2010,PP(99):1~5.

[8]. Chao Cheng and Keshab K Parhi. Low-Cost Parallel FIR Filter Structures With 2-Stage Parallelism[J].IEEE Transactions on Circuits and Systems I:Regular,2007,54(2):280~290.

[9]. Tearney G J and Bouma B E. Real-Time FPGA Processing for High-Speed Optical Frequency Domain Imaging [J].IEEE Transactions on Medical Imaging, 2009,28(9):1468~1472.