INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 5, Issue 2 JAN 2016

DESIGN AND IMPLEMENTATION OF DA-BASED RECONFIGURABLE FIR DIGITAL FILTER USING VERILOGHDL

[1]J.SOUJANYA,P.G.SCHOLAR, KSHATRIYA COLLEGE OF ENGINEERING,NIZAMABAD
[2]MR. DEVENDHER KANOOR,M.TECH,ASSISTANT PROFESSOR,KSHATRIYA COLLEGE OF ENGINEERING,NIZAMABAD
[3]MR. KONDRA GOPI,M.TECH,ASSISTANT PROFESSOR,KSHATRIYA COLLEGE OF ENGINEERING,NIZAMABAD

ABSTRACT:

In this paper, we present the design optimization of one- and two-dimensional fully-pipelined computing structures for area-delay-power-efficient implementation of finite impulse response (FIR) filter by systolic decomposition of distributed arithmetic (DA)-based inner-product computation. The systolic decomposition scheme is found to offer a flexible choice of the address length of the look-up-tables (LUT) for DA-based computation to decide on suitable area-time trade-off. It is observed that by using smaller address-lengths for DA-based computing units, it is possible to reduce the memory-size but on the other hand that leads to increase of adder complexity and the latency. For efficient DA-based realization of FIR filters of different orders, the flexible linear systolic design is implemented on a Xilinx Virtex-E XCV2000E FPGA using a hybrid combination of Handel-C and parameterizable VHDL cores. Various key performance metrics such as number of slices, maximum usable frequency, dynamic power consumption, energy density and energy throughput are estimated for different filter orders and address-lengths. Analysis of the results obtained indicate that performance metrics of the proposed implementation is broadly in line with theoretical expectations. It is found that the choice of address-length M=4 yields the best of area-delaypower-efficient realizations of the FIR filter for various filter orders. Moreover, the proposed FPGA implementation is found to involve significantly less area-delay complexity compared with the existing DA-based implementations of FIR filter.

Keywords: Finite impulse response (FIR) filter, linear convolution, systolic array, field programmable gate arrays (FPGA),distributed arithmetic

I. INTRODUCTION

Finite impulse response (FIR) digital filters are extensively used due to their key role in various digital signal processing (DSP) pplications [1], [2]. Along with the advancement in very large scale integration (VLSI) technology as the DSP has become increasingly popular over the years, the highspeed realization of FIR filters with less power consumption has become much more demanding. Since the complexity of implementation grows with the filter order and the precision of computation, real-time realization of these filters with desired level of accuracy is a challenging task. Several attempts have, therefore, been made to develop dedicated and reconfigurable architectures for realization of FIR filters in application specific integrated circuits (ASIC) and field programmable gate arrays (FPGA) platforms. Systolic designs represent an attractive architectural paradigm for efficient hardware implementation of computation-intensive DSP applications, being supported by the features like simplicity, regularity and modularity of structure. Additionally, they also possess significant potential to yield high-throughput rate by exploiting high-level of concurrency using pipelining or parallel processing or both [3]. To utilize the advantages of systolic processing, several algorithms and architectures have been suggested for systolization of FIR filters [4]–[7]. However, the multipliers in these structures require a large portion of the chip-area, and consequently enforce limitation on the maximum possible number of processing elements (PEs) that can be accommodated and the highest order of the filter that can be realized. Multiplierless distributed arithmetic

(DA)-based technique, has gained substantial popularity, in recent years, for their high-throughput processing capability, and increased regularity which results in cost-effective and area-time efficient computing structures. The main operations required for DA-based computation of inner-product are a sequence of look-up-table (LUT)-accesses followed by shift accumulation operations of the LUT output. DA-based computation is well-suited for FPGA realization, because the LUT as well as the shift-add operations can be efficiently mapped to the LUT-based FPGA logic structures.

In FIR filtering, one of the convolving sequences is derived from the input samples while the other sequence is derived from the fixed impulse response coefficients of the filter. This behavior of FIR filter makes it possible to use DA-based technique for memory-based realization. It yields faster output compared with the multiplier-accumulator-based designs because it stores the pre-computed partial results in the memory elements [8], which can be read out and accumulated to obtain the desired result. The memory requirement of DA-based implementation for FIR filters, however, increases exponentially with the filter order. DA was first introduced by Croisieret al [9]; and further developed by Peled and Lui [10] for efficient implementation of digital filters. Attempts are made to use offset-binary coding [11] to reduce the ROM size by a factor of 2. An LUT-less adder-based DA approach has been suggested by Yoo and Anderson, where memory-space is reduced at the cost of additional adders [12]. Memory-partitioning and multiple memory-bank approach along with flexible multi-bit data-access mechanisms are suggested for FIR filtering and inner-product computation in order to reduce the memorysize of DA-based implementation [13]–[17]. Allredet alhave suggested an efficient DA-based implementation of least mean square (LMS) adaptive filter using a decomposition of DAbased FIR computation and subsequent memory decomposition [18]. All these structures, however, are not suitable for implementation of the FIR filters in systolic hardware since the partial products available from the partitioned memory modules are summed together by a network of output adders. A new tool for the automatic generation of highly parallelized FIR filters based on PARO design methodology is presented in [19], where the authors have performed hierarchical partitioning in order to balance the amount of local memory with external communication, and they have achieved higher throughput and smaller latencies by partial localization. A systolic decomposition technique is suggested in a recent paper for memory-efficient DA-based implementation of linear and circular convolutions [20]. In this paper we have extended further the work of [20] to obtain an area-delay-power-efficient implementation of FIR filter in FPGA platform.

II. ADAPTIVE ALGORITHMS

There are numerous methods for the performing weight update of an adaptive filter. There is the wiener filter, which is the optimum liner filter in terms of mean squared error, and several algorithms that attempt to approximate it, such as the method of steepest descent. There is also least-mean square algorithm, developed by Windrow and Hoff originally for use in artificial neural networks. Finally, there are other techniques such as the recursive-least square algorithm and the kalman filter. The choice of algorithm is highly dependent on the signals of interest and the operating environment, as well as the convergence time required and computation power available.

PROBLEM STATEMENT

Due to the high performance requirements and increasing complexity of DSP and multimedia communication applications, filters with large number of taps are required to increase the performance in terms of high sampling rate. As a result the filtering operations are computationally intensive and more complex in terms of hardware requirements. The FIR filters perform the weighted summations of input sequences with constant coefficients in most of the signal processing and multimedia applications. These filters are widely used in video convolutions functions, signal preconditioning, and other communication applications. The decrease in computational complexity causes the increase in the performance, in terms of speed, area and power. High speed, low area and power efficient

conscious design techniques in SoC include efforts at all level of abstraction. One way to efficiently incorporate high performance design technique is toimplement IP cores [4].
These cores have following major advantages.

- Reusability across designs
- Reduction of the design effort
- Shorter time to market.

The disadvantage of FIR filters is that they require high order. The high order demands more hardware, area and power consumption. To minimize these parameters, our goal is to implement an efficient high order filterin digital systems. By the reduction of arithmetic in terms of multipliers, our goal is to reduce the parameters namely, hardware, area and power. This is ultimate goal of the implementation of an efficient FIR filter and hence DA algorithm is used for implementation of high order FIR filter. FIR filter is incorporated with a MAC unit. The purpose of MAC unit is to multiply the input with constant coefficients, to shift and then to add them. This process is repeated until all partial products produce the output after accumulation. It increases the hardware complexity because a simple multiplier circuitry is used. The idea is to somehow bypass or replace the multiply and shift operations with less complex operations. Distributed Arithmetic (DA) Algorithm can be used to replace MAC unit. The DA Algorithm actually uses lookup table for storing constant coefficients. So the use of lookup tables reduces the hardware complexity and hence the new design is more efficient in terms of less area, more speed and low power consumption. FIR filter reference core uses a simple MAC unit. We have replaced MAC unit in FIR filter reference core with DA Algorithm. In this study, performance of Reference Core with Simple MAC and reference core with DA is compared.

III. INTRODUCTION OF DISTRIBUTED ALGORITHM

Distributed arithmetic is a bit level rearrangement of a multiply accumulate to hide the multiplications. It is a powerful technique for reducing the size of a parallel hardware multiply-accumulate that is well suited to FPGA designs. It can also be extended to other sum functions such as complex multiplies, Fourier transforms and so on. In most of the multiply accumulate applications in signal processing, one of the multiplicands for each product is a constant. The DA targets the products of sums which cover all filtering application and frequency transfer functions. DA uses Look-Up Table (LUT) which stores the constant coefficients of FIR Filter. The size of Look-Up Table (LUT) in DA algorithm is 2, where k is the number of filter taps. When number of taps increases, LUT grows exponentially. By using offset Binary Code (OBC), the size of the LUT can be reduced. This is very efficient in terms of less hardware and more speed. Many DSP applications required FIR which having MAC (Unit multiplier and add accumulator), replacing MAC with LUT-Based DA algorithm having power, efficiency and less area usage. Proposed DA algorithm is hardware efficient for VLSI and FPGA, but LUT-Less OBC is efficient only for custom VLSI [5].We have used DA for multiplier less architecture in FPGA. For DA based on look-up table having constant coefficient and changing variable, one needs to design a highly efficient FIR in digital signal processing.

DA can be used for high order filter. There are two techniques used in DA algorithm, one of which is parallel distributed and the other is serial distributed. [6].The DSP FIR filter functions are used in telecommunications (e.g. Telecomm in Biomedical Signal Processing Communication, Wireless satellite and Image processing) which are performed efficiently. The multipliers in MAC unit of many DSP functions have more power and area requirements. There are two techniques in this respect which are multiplier less. One of them is Conversion based, in which coefficients of filters are converted into numeric representation. The second is based on LUT which stores pre- computed coefficients values of FIR filters. The LUT in DA algorithm uses more memory. [7].

IV.BUILDING BLOCK OF FIR REFERENCE CORE

The existing core may be implemented using the main components which are given below:

- Counter
- Controller
- X sample value memory (X-RAM)
- B coefficient memory (B-Rom)
- Beta and gamma register
- Multiply-accumulator (MAC)
- Rounding
- Output sample (Y-Register)

The main components of the cores can be easily recognized from the top of module of the core. The block diagram of the existing direct form of FIR filter core is [8]

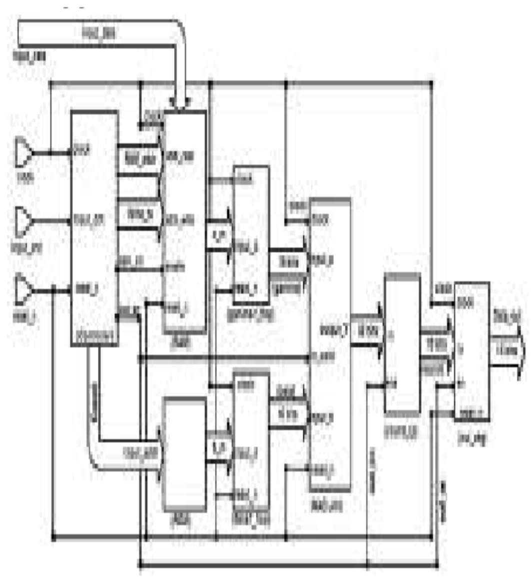


Fig.1. Block diagram of TOP module of the FIR Filter

V.IMPLEMENTING THE FIR FILTER USING DA

FIR filter has 16-taps. Each tap has 16-bit input data width and 16 filter coefficients. In designing FIR filterusing DA (Distributed Arithmetic), these coefficients are placed in a look-up table. This is because these coefficients are constants. The look-up table grows exponentially when the filter coefficients are increased, so the break-up in the design is necessary and one must place four-coefficients in each look-up table. The width of each coefficient may be 8-bits or 16-bits depending upon the design. The width of the inputs data also vary to 8-bits and 16-bits, each LSB bits of input data combined in parallel to form the address of the look-up table. Distributed arithmetic Algorithm replaces "AND" and"add" operation as compared with MAC unit. The fourlook-up table store 16 coefficients of FIR filter. More than four look-up tables are used for storing more coefficients for the better response of the FIR filter. The LUTsin DA algorithm uses the multiplier less technique.The LUTs used less CLB (configuration logic blocks) in the FPGA to increase the throughput and data rates. The FPGA has no multiplier and can be used SRAM based DA algorithm.Single FPGA chip can be used instead of using multiple DSP devices for better performance in terms of speed area and power, due to SRAM present in FPGA, FPGA is more efficient for the implementation of signal processing applications. DA became best algorithm relating to filtering operation, because SRAM based FPGA stored look-up table values which are pre-computed and also FPGA gives surrounding logic in a single chip. Distributed arithmetic algorithm gives best performance when we used in filtering operation because hardware complexity less in DA as compared to conventional MAC.

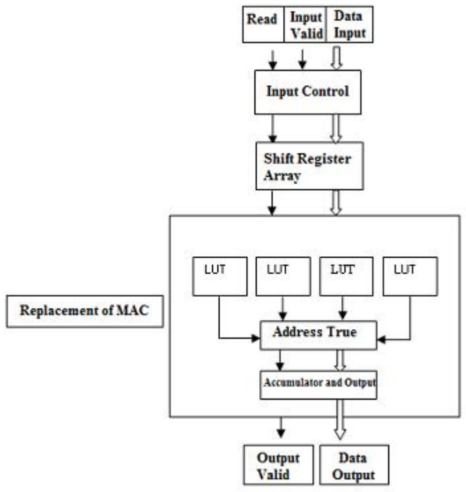
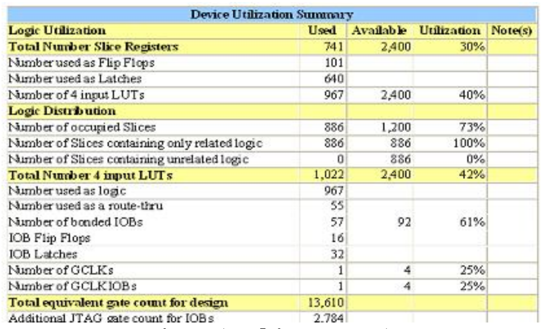


Fig.4.Implementation of FIR filter using DA

VI.SIMULATION AND RESULTS

Implementation of FIR filter cores has been observed and we can see that fir filter cores have been implemented with both reference and DA structure. Results have been taken in terms of area utilized, power dissipated and speed performance for 16bits-20 taps and 8bits-20 taps. FIR filter cores have been designed in Verilog HDL and implemented using Xilinx 10.1i tool. Simulations were performed using Modelsim6.4b.



Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Total Number Slice Registers	741	2,400	30%	
Number used as Flip Flops	101			
Number used as Latches	640			
Number of 4 input LUTs	967	2,400	40%	
Logic Distribution				
Number of occupied Slices	886	1,200	73%	
Number of Slices containing only related logic	886	886	100%	
Number of Slices containing unrelated logic	0	886	0%	
Total Number 4 input LUTs	1,022	2,400	42%	
Number used as logic	967			
Number used as a route-thru	55			
Number of bonded IOBs	57	92	61%	
IOB Flip Flops	16			
IOB Latches	32			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	13,610			
Additional JTAG gate count for IOBs	2,784			

Area Comparison (16 bit 16-taps)

Table 1 show that the area of Conv.UDF FIR Filter is less as compared with same core implemented with DA algorithm

Table 1. Area Comparison of 16 bit 16-taps

Filter Cores	Conv.UDF FIR Filter Core	Conv.UDF FIR Filter Core with DA Algorithm
No. of Slices	70%	74%
Slice Flip Flops	32%	37%
Input LUTs	38%	37%
Bonded IOBs	60%	60%
Total Eq Gate Count	13610	13573

Speed comparison (16 bit 16-taps)

Table 2 and Fig. 7 show results comparison of conv.UDF FIR Filter Core and conv.UDF FIR Core with DA Algorithm for speed. About 47% increase in speed has been found when using DA algorithm.

Table 2. Speed Comparison (16 bit 16-taps)

Filter Cores	Conv.UDF FIR Filter Core	Conv.UDF FIR Filter Core with DA Algorithm
Min Period	26.798 nS	18.161 nS
Input Arrival time	9.574 nS	9.574 nS
Output Req Time	15.842 nS	16.526 nS
Max Freq	37.316 Hz	55.063 Hz
Speed Improvement	________	47.56 %

Conclusion And Future Work

The results show that distributed arithmetic algorithm is better for FIR filters implementation on FPGAs. The efficiency in terms of area, speed and power has been analyzed. Comparison of results clearly shows that efficiency in terms of power dissipation and speed has been increased having almost same area consumption. The DA has two techniques, one of which is the serial DA and other one is the parallel DA. In this thesis, the serial distributed arithmetic is used to make the FIR Filter more efficient. In future, the parallel DA can be used to increase the efficiency of FIR Filter in terms of data rates. The implementation of DA based algorithm, serial distributed arithmetic algorithm and parallel distributed arithmetic use the look up table. The size of the look up table increases when the number of filter taps is increased. For better performance of FIR Filter, LUT less DA implementation uses MUX. Every shift register uses MUX that select0 or filter coefficients, and this technique can be used to increase the efficiency of the FIR Filter in future.

REFERENCES

[1]. Farhat Abbas Shah, Habibullah Jamal, Muhammad Akhter Khan, "Reconfigurable Low Power FIR Filter based on Partitioned Multipliers", 16-19 December, KFUPM, Dhahran, KSA, ICM, 2006.
[2]. Xilinx, "Design Reuse Methodology for ASIC and FPGA Designers", 2004. www.xilinx.com.
[3]. N. Sankarayya, K.Roy, D. Bhattacharya,

"Algorithms for Low Power and High Speed FIR filter Realization Using Differential Coefficients", Analog and Digital Signal Processing, IEEE Transactions, vol, 44(6), pp. 488-497, 1997.
[4]. A.T.Erdogan, M.Hasan and T.Arslan, "Algorithmic low power FIR cores", circuits and systems, IEEE proceedings, vol 150, pp. 155-169, 2003.
[5]. Heejongyoo and David V. Anderson, "Hardware efficient distributed arithmetic architecture for high order digital filters", Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE, vol 5, pp. 125-128, 2005.
[6]. Wang sen, Tang Bin and Zhu Jun, "Distributed arithmetic for FIR filter design on FPGA", Communications, Circuits and Systems, 2007. ICCCAS 2007. IEEE, pp. 620-623, 2007.
[7]. Patrick Longa and Ali Miri, "Area efficient FIR filter design on FPGAs using distributed arithmetic", Symposium on signal processing and information technology, IEEE processing, pp. 248-252, 2006.
[8]. Muhammad Akhtar khan and A.T .Rrdogan, "Parameterized and programmable low power soft FIR filtering IP cores", Proceedings of the 4th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, 2004.
[9]. P. K. Meher, S. Chandrasekaran and A. Amira, "FPGA realization of FIR filters by efficient and flexible systemization using distributed arithmetic", IEEE Transactions on Signal Processing, vol. 56, pp. 3009-3017, 2008.
[10]. D. J. Allred, H. Yoo, V. Krishnan, W. Huang and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," IEEE Transactions on Circuits and Systems, vol, 52, pp, 1327-1337, 2005.