

A Peer Reviewed Open Access International Journal

An area optimized FIR Digital filter using DA Algorithm based on FPGA

B.Chaitanya

Student, M.Tech (VLSI DESIGN), Department of Electronics and communication/VLSI Vidya Jyothi Institute of Technology, JNTU University Hyderabad, India

Abstract: The VLSI design industry has grown rapidly during the last few decades. The complexity of the applications increases day by day due to which the area utilization increases. The tradeoff between area and speed is an important factor. The main focus of continued research has been to increase the operating speed by keeping the area and memory utilization of the design as low as possible. In this paper we have presented a DA based approach which uses the logic of shift and add operations which reduce the area occupied by the multiplication logic and enhance the speed. The results of the proposed work have been observed by XILINX ISE and the design has been targeted for SPARTAN 3E-XC3S250E device. We observe up to 50% reduction in the number of slices and up to 75% reduction in the number of LUTs for fully parallel implementations. Our design performs significantly faster than the MAC filters, which uses embedded multipliers.

Keywords:- Verilog, Multiple Constant Multiplication(MCM), finite impulse response, Low complexity, Distributive arithmetic.

I.INTRODUCTION

In digital signal processing (DSP) systems finite impulse response (FIR) filters have very much importance since their characteristics in linear-phase and feed-forward implementations make them very useful for building stable high performance filter architectures. The direct and transposed form FIR filter logic diagrams are illustrated in Fig. 1(a) and 1(b). As shown in figure both architectures have similar Mrs. A. Jayalakshmi

Associate Professor, Department of Electronics and communication/VLSI Vidya Jyothi Institute of Technology, JNTU University Hyderabad, India

complexity in hardware, and the performance and power efficiency.

The architecture of a multiplier of the digital FIR filter in its transposed form is shown in [Fig. 1(b)], where the multiplication of filter inputs with the filter coefficients is realized, due to the significant impact on the complexity and performance of the design because a large number of constant multiplications are required. This is generally know as the multiple constant multiplications (MCM) operation and is also a central operation and performance bottleneck in many other DSP systems such as fast Fourier transforms, discrete cosine transforms (DCT's) and ECC codes.



Transposed form with generic multipliers.

Multiple constant multiplications is involved to generate constant multiplication in Digital Signal Processing, ECC codes, MIMO system applications.

Volume No: 2 (2015), Issue No: 7 (July) www.iimetmr.com

Technology, Management and Resea A Peer Reviewed Open Access International Journal

In most of the applications complete use of multipliers are not needed. As the coefficients are constant then constant multiplication can be used. So depending upon the constant coefficients MCM architecture can be constructed, the designed architecture can be called as many times it required.

There are two methods to implement Constant multiplication either by digit serial design or digit parallel design. Digit-parallel design of constant multiplier needs external wire for shifting. Hence it occupies more area while implementation takes place in FPGA or any other ASIC. So digit serial design overcomes area constrain with acceptable delay timing.

Multiplication done with constants is known as constant multiplication. This process is used mostly in filter functions. There are two different types of constant multiplications such as Single Constant Multiplication (SCM) and Multiple Constant Multiplication. The input sample is multiplied with single specific coefficient to produce output is called SCM. Canonical Signed Digit (CSD) number system is used to implement SCM multipliers.

Input samples are multiplied with multiple coefficients to produce multiple outputs is known as MCM. Multiplication is a process of shift and adds operation. Constant multiplier design includes the number of adders, subtractors and shifters according to the coefficient pair.

FIR filter output can be obtained by multiplication of input sample and impulse response. Direct form and transposed form implementations are two forms of FIR filter implementations. Instead of direct form, transposed form is most effective. The Multiplication process takes place in multiplier block. Thus transposed form multiplier blocks in FIR filter will replace by MCM architecture also known as shift and add architecture.

II. DISTRIBUTED ARITHMETIC (DA)

Filters are usually frequency selective networks, which is capable of modifying an input signal in order to facilitate further processing. Thus digital filters are more preferred than analog due to its high signal integrity [2]. An FIR filters are widely applied for variety of DSP areas because providing virtues of linear phase and system stability. The basic convolution equations of filter representation are shown as follows:

$$Y(n) = b_0 x(n) + b_1 x(n-1) + \dots + b_N x(n-N)$$
(1)
= $\sum_{i=0}^{N} bi x(n-i)$ (2)

Distributed Arithmetic is termed so because the multiplications that appear in signal processing are reordered and combined such that the arithmetic becomes distributed completely through the structure rather than being lumped. Multipliers are replaced by combinational Look Up Tables (LUT) [3]. Since LUTs are considerably larger in size, the quality of implementing FIR filter mainly relies upon the efficiency of logic synthesis algorithm mapping to FPGA.

DA provides bit serial operations that implement a series of fixed point MAC operations in a known number of steps, regardless of number of terms to be calculated. The main operations required for DA based computation of inner product are sequence of lookup table access followed by shift accumulation operations of LUT output [9].

According to DA, we can make look up tables (LUT) to store MAC values and call out values accordingly to the input data if necessary. Therefore, LUT's are utilized to facilitate the operations of MAC units so as to save hardware resources. This technique also facilitates DA computation suitable for FPGA realization, because the LUT along with shift and add operations can be directly mapped to LUT base FPGA logic structure.

The advantage of DA is its efficiency of mechanization. It turns out well when the number of

A Peer Reviewed Open Access International Journal

elements in a vector is nearly same as word size then DA is quite fast.

III PROPOSED WORK

Distributed arithmetic is well known method of implementing FIR filters without the use of multipliers. In DA the task of summing product terms is replaced by table lookup procedures that are easily implemented on FPGA.

In FIR filtering, one to one mapping of the convolving sequences are from two sources, one obtained from input samples while the other sequences are obtained from fixed impulse response coefficients of filter [6]. This behavior of FIR filter makes it suitable for DA based technique for memory realization. It yields faster output, it stores pre-computed partial results in memory components that can be read out and accumulated to obtain desired result [8].



Fig 2: Proposed structure of the DA-based FIR filter for FPGA implementation.

The Distributed Arithmetic technique of FIR filter consists of Look-Up Table (LUT), Shifters and accumulator with adder tree. In this technique all the partial product outputs are pre-computed and placed in a Look Up Table (LUT). These table entries are addressed by the addresses generated by input

> Volume No: 2 (2015), Issue No: 7 (July) www.ijmetmr.com

multiplier bit data. All inputs are fed simultaneously. From the input data, address are generate and allowed to access the LUT; its outcome is added to the accumulated partial products.



Fig 3: Block diagram of FIR filter using DA

The Basic block diagram for FIR filter implementation using DA [8] is shown in Figure 3. The complete dot product calculations takes L clocks where L is the size of input data, and it is not depended on input data size. During the initial cycle of operation, the Least-Significant Bits of input i.e., $X_0(n)$, $X_0(n-1)$..., of the K input samples are arranged to form K-bit addresses. These addresses are allowed to access Look-Up Table and that table outcome becomes the initial value of the accumulator. During the very next cycle of operation, the next-to-least significant bits $X_1(n)$, $X_1(n-1)$, ..., $X_1(n-K+1)$ of the K input samples are arranged to form K-bit address which will be allowed access lookup, and the adder sums the Look up Table output is shifted by one bit and summed to the contents of the accumulator.



Fig.4: Shift-add's implementation of 29x and 43x (a) Without partial product sharing (b) with partial product sharing (C) Modified shift add

July 2015

A Peer Reviewed Open Access International Journal



Fig 5: Output for an FIR filter with existing algorithm



Fig 6: Output for an FIR filter with proposed algorithm



Fig 7: The RTL schematic view of proposed 16-tap FIR filter

Modeling and simulation are important in research. Representing the real systems either via physical reproductions at smaller scale, or via mathematical models that allow representing the dynamics of the system via simulation, allows exploring system behavior in an articulated way which is often either not possible, or too risky in the real world. Simulation results have been observed by using Modelsim simulator and synthesis results observed by XILINX ISE tool.

FIR_EXISTING_X Partition Summary					
No partition information was found. Device Utilization Summary (estimated values)					
					Logic Utilization
Number of Slices	110	2448	4%		
Number of Slice Flip Flops	105	4896	2%		
Number of 4 input LUTs	161	4896	3%		
Number of bonded IOBs	26	108	24%		
Number of MULT18X18SIOs	7	12	58%		
Number of GCLKs	1	24	4%		

Fig 8: Device utilization report of an Existing system

FIR_PROPOSED_X Partition Summary No partition information was found.					
Logic Utilization	Used	Available	Utilization		
Number of Slices	47	2448	1%		
Number of Slice Flip Flops	17	4896	0%		
Number of 4 input LUTs	79	4896	1%		
Number of bonded IOBs	14	108	12%		
Number of GCLKs	1	24	4%		

Fig 9: Device utilization report of an proposed system

IV. CONCLUSION

The proposed architecture shows the area efficient implementation of Digital filter using DA based on FPGA. This architecture replaces the complicated multiplication-accumulation operation with simple shifting and adding operations based DA algorithm which is directly applied to realize FIR filter. The simulation results have been observed by Modelsim simulator. Also synthesized by XILINX XST tool and targeted for SPARTAN 3E FPGA device XC3S250E. Device utilization values of both algorithms are compared. We have observed up to 50% reduction in the number of slices and up to 75% reduction in the number of LUTs for fully parallel implementations.

REFERENCES

[1] Kenny Johansson, Oscar Gustafsson, Andrew G. D., and Lars Wanhammar, "Algorithm to reduce the number of shifts and additions in multiplier blocks using serial arithmetic" IEEE MELECON, pp. 197-200, 2004.



[2] Levent Aksoy, Cristiano Lazzari, Eduardo Costa, Paulo Flores, José Monteiro., "Efficient shift-adds design of digit-serial multiple constant multiplications" GLSVLSI'11,2011.

[3] Sang Yoon Park Member, IEEE —Efficient FPGA and ASIC Realizations of DA-Based Reconfigurable FIR Digital Filter IEEE transactions 2014 on circuits and systems-ii: express briefs.

[4] P. Karthikeyan, Dr. Saravanan.R — FPGA Design of Parallel Linear-Phase FIR Digital Filter Using Distributed Arithmetic Algorithm IJCSMC, Vol. 2, Issue. 4, April 2013

[5] Ramesh .R , Nathiya .R —Realization of fir filter using Modified distributed arithmetic architecture An International Journal (SIPIJ) Vol.3, No.1, February 2012

[6] M. Kumm, K. Moller, and P. Zipf, —Dynamically reconfigurable FIR filter architectures with fast reconfiguration, in Proc. 2013 8th Int. Workshop on Reconfigurable and Communication-Centric Systems-on-Chips (ReCoSoC), Jul. 2013.

[7] P. K. Meher and S. Y. Park, —High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic, I in Proc. 2011 IEEE/IFIP 19th Int. Conf. VLSI, System-on-Chip, (VLSI-SOC'11), Oct. 2011, pp. 428–433.

[8] Damarla Ramya & V.C. Madavi, FPGA Controlled Rover through Android Smartphone, IJMETMR, http://www.ijmetmr.com/olmay2015/DamarlaRamya-

VCMadavi-24.pdf, Volume No: 2 (2015), Issue No: 5 (May)

 [9] S. M. Badave and A .S. Bhalchandra, International Journal of Information and Electronics Engineering —Multiplierless FIR Filter Implementation on FPGAI Vol. 2, No. 3, May 2012

[10] Md.Zameeruddin, Sangeetha Singh ∥ Efficient Method for Look-Up-Table Design in Memory Based Fir Filters∥ International Journal of Computer Applications (0975 – 8887) Volume 78 – No.16, September 2013

[11] M. Yazhiniand R. Ramesh —FIR Filter Implementation using Modified Distributed Arithmetic ArchitectureIIndian Journal of Science and TechnologyISSN: 0974-6846, May 2013

Volume No: 2 (2015), Issue No: 7 (July) www.ijmetmr.com

July 2015