



SECTIONAL RECONFIGURABLE FIR FILTER DESIGN USING SYSTOLIC DISTRIBUTED ARITHMETIC (DA) ARCHITECTURE ON FPGA

Sk. Firoz

Department of Electronics and Communication Engineering
Narayana Engineering College, Nellore-524003, A.P, India

S.Girish Gandhi

Department of Electronics and Communication Engineering
Narayana Engineering College, Nellore-524003, A.P, India

C.Leela Mohan

Department of Electronics and Communication Engineering
Narayana Engineering College, Nellore-524003, A.P, India

**M.Swarna
Lakshmi**

Department of Electronics and Communication Engineering
Narayana Engineering College, Nellore-524003, A.P, India

ABSTRACT

This work proposes an efficient architecture that employs a two dimensional fully pipelined structure to realize computationally efficient, low power, high speed Finite Impulse Response (FIR) filter. In distributed arithmetic, an architecture is defined for the LUT's (look up table) to reduce the partial reconfiguration time and this is stored in RAM. By changing the filter coefficients in the partial reconfigurable module, the fir filter is robustly reconfigured to achieve low pass and high pass filter characteristics. The proposed architecture shows improvement in reconfiguration time and efficiency. The design is implemented on Spartan3E FPGA kit.

KEYWORDS : Active Partial Reconfiguration, FIR, Distributed Arithmetic, Systolic architecture, FPGA.

INTRODUCTION

In Digital Signal Processing, FIR (Finite Impulse Response) filters are the most essential components which are generally implemented for high speed calculations. Based on power or resource considerations, to implement new functionality, it is necessary to design configurable architecture in various embedded applications to obtain the low power high speed FIR filters. In this literature [1,2,3,4] the time varying requirements in power, resources or performance and to maintain a good speed of operations adopted in hardware resources can be easily done by using FPGA. Numerous well-organized architectures are developed by the usage of reconfigurable architectures and non-reconfigurable architectures. Reconfigurable overhead is the major challenge took place in partial reconfigurable architecture, with the increase in filter order and the type of arithmetic used which tends to the increase computational complexity and reconfiguration time in FIR filters. A distributed arithmetic method is introduced [2,5,6,7] which reduces the power computation of multipliers in MAC operation several multiplier less scheme, by introducing the multiplier less technique which uses memories or LUTs for the storage of pre-computed values of coefficient operations. The speed reduces significantly by introducing a DA algorithm which shows a good set of characteristics with respect to speed and chip area. As the number of filter coefficients increases, results increase in LUT complexity which drastically reduces the speed. The DA architecture which uses the concept of systolic arrays evades the output adder network by introducing the pipelining as the output adder network will not support pipelining concept, this systolic array scheme was developed by H.T. Kung [9]. In static region of FPGA the design uses output adder network for the architecture. To achieve reconfiguration the reconfigurable FIR filter [6,7] uses Xilinx reconfiguration tools on Spartan FPGA. The reconfiguration time depends on the size of bit stream file (file size in kb). The coefficients in the LUT are dynamically reconfigured using modular reconfigurable scheme as the systolic DA architecture is introduced in this work, the LUT's use small dynamic reconfiguration area to store the coefficients by uploading a '.BIT file' [4]. This paper proposed in five sections, section II shows the basics of DA with systolic architecture, section III shows proposed systolic DA architecture, section IV and section V shows the results and conclusion.

II. DISTRIBUTED ARITHMETIC ARCHITECTURE COMPUTATION

A decomposition scheme for flexible DA based systolic FIR filters is derived by computing the inner products by considering briefly the conventional distributed architecture.

$$y[n] = \sum_{i=0}^{n-1} x[n-i]c[i] \dots \dots \dots (1)$$

The above equation describes an FIR filter, this shows nothing but the inner product deferred by the explicit value with filter coefficients.

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

The input value $x(n)$ can be expressed in the form of corresponding bits in the second equation. If we include the equation (2) into equation (1) we get

$$y = \sum_{n=0}^{N-1} c[n] \times \left\{ \sum_{b=0}^{B-1} (x_b[n] \times 2^b) \dots \dots \dots (3) \right.$$

$$y = \sum_{b=0}^{B-1} 2^b \times \left\{ \sum_{n=0}^{N-1} (c[n] \times x_b[n]) \dots \dots \dots (4) \right.$$

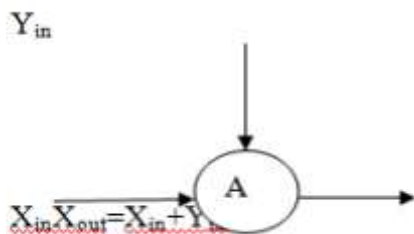
Equation (4) is re arranged from the equation (3), the sum of the product of the filter coefficients with the bit of inputs is shown in the equation (4). As the number of inputs is N , then the sum of product contains 2^N values, and all the values obtained are stored in the LUT's, and the inputs added to the LUT's are the corresponding bit vector from the input.

$$y = \sum_{b=0}^{B-1} 2^b \times \{f(x_b[0], x_b[1], \dots, x_b[n])\} \dots (5)$$

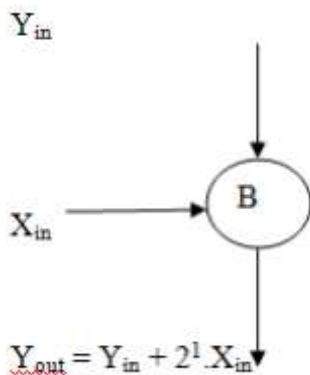
The outputs of the LUT's are shift added to get the output by taking all the b bits as shown in the above equation (5). The term $f(x_b[.])$ shown is the value from LUT, according to input bits. As to hold the result values temporarily we use the LUT's instead of the multipliers adders and registers, LUT's have a memory unit (ROM), and a set of shift adders equal to the number of bits that are used for representation. The LUT should be the size of $2N$ where N is the order of the filter as the computation of MAC operation are very fast to do. LUT size increases exponentially as the order of the filter increases which results to the increase in the time to fetch memory which effects the operational frequency of the FIR filter, to overcome this problem as to reduce the size of the LUT'S with multiple LUT's we introduce the systolic decomposition technique.

$$y = \sum_{b=0}^{B-1} 2^b \times \left\{ \sum_{p=0}^{P-1} \left[\sum_{m=0}^{M-1} c[n] \times x_b[n] \right] \right\} \dots (6)$$

If the order of filter is N is a composite number which obtained by the product of two numbers P and M , then its is stale the 2^N LUT into P LUTs of size 2^M this shows in the above equation (6). For efficient mapping of the equation (6) into the FPGA a systolic array implementation is introduced. This systolic arrays are used to implement computational expensive algorithm in hardware[5] and these are the examples of VLSI special processor. The processing elements are systematically arranged which as small cells combined shows the systolic arrays which performs simple task like multiply addition and memory fetching and some other operations and passes the data. And this processing elements perform a simple task which does not consume more number of clock cycles[7]. This processing element in DA architecture consist of an LUT with a memory fetch unit. Here we have two different types of processing elements that are shown below.



Fig(1). Processing element A

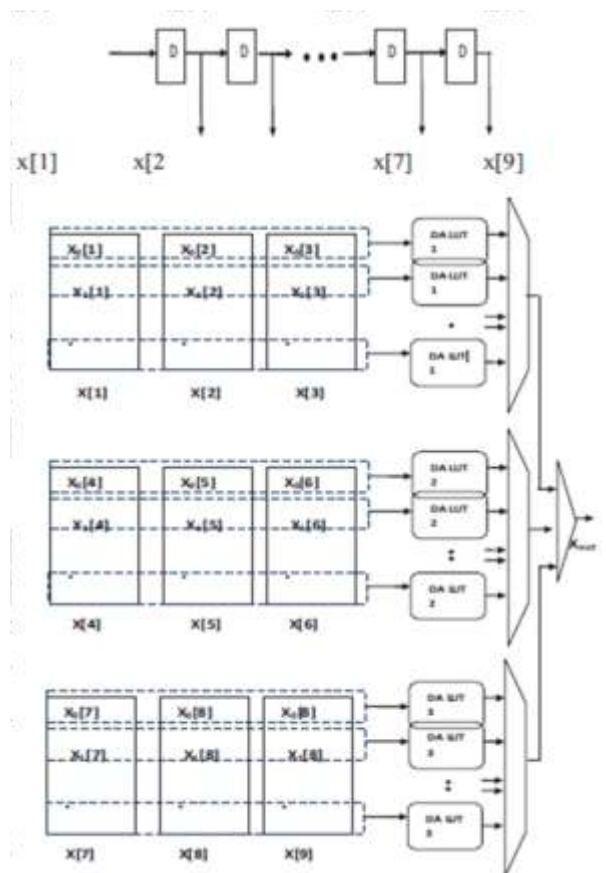


Fig(2). Processing element B

These two processing elements consists of an LUT and an adder to perform different operations and processing element A do the memory fetching and addition of fetch to the input value and processing element B do the shift addition of input with the another input. These processing elements have different LUT's and all processing elements A are connected in horizontal and one processing element B is connected in horizontal and all the other B elements are connected in column manner.

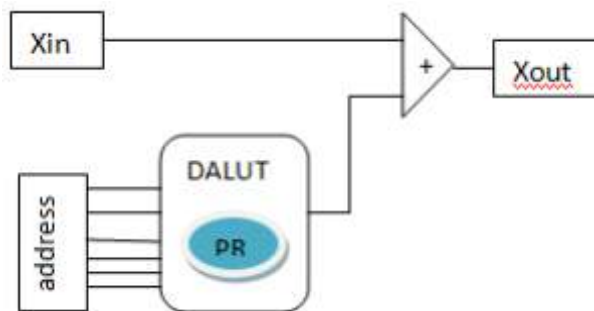
III PROPOSED SYSTOLIC DA ARCHITECTURE

Using systolic DA architecture the sectional reconfigurable FIR filter is designed and it is shown below in a block diagram.



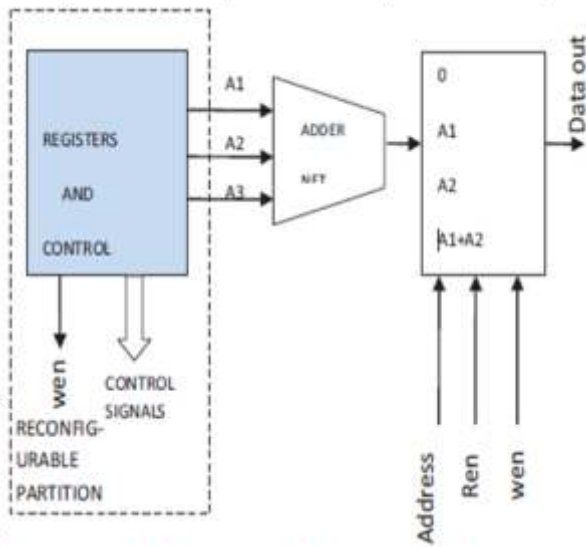
Fig(3). Systolic DA architecture

The block diagram shows 9 tap filter where the triangle show the adder and the trapezium shows the shift adder with the DA LUT's. the reconfiguration partition and LUT's are present in the DA architecture. As the reconfiguration part only consists of the filter coefficients not all the LUT part so the area of the reconfiguration partition is reduced[8]. The computation of LUT is taken from FPGA is taken further which take less time in computation than the reconfiguration time.



Fig(4). Processing element A architecture

The above figure shows the processing element of 'A' where it contains an LUT which is referred as DA LUT. In this DA LUT there is a part reconfigurable part shaded in blue. The LUT is modified which contains registers, adder and a memory as shown in the below figure(5).



Fig(5). DALUT Architecture

The memory is RAM which is placed in the reconfiguration part and it configures the DA coefficients[9]. In this work the DA coefficients are not considered and the lut coefficients are taken into consideration. The values are updated into LUT RAM which are calculated by the network of adders, where RAM contains the input addresses and write enable. The control signals and write enable are generated whenever the reconfigurable area is reconfigured and the sum variables are allocated to their primary values. The values in the LUT RAM are changed as the write enable signal enables writing data into the RAM due to this the system works as a different filter. In this work the reconfigurable area is gradually decreased and there is an great extant of decrease in size of partial BIT file, and the time taken for the reconfigured filter is highly reduced.

IV. IMPLEMENTATION AND RESULTS

The software used for simulation and implementation of the sectional reconfigurable FIR filter design is done using Xilinx system generator, Xilinx ISE 14.3[10] and spartan-3E FPGA kit. A filter of order 9 is designed such that it is represented in 12 bits. The coefficients are reconfigured in one module and the two other modules are used for testing, in one module we use coefficients of low pass filter and other module is implemented for high pass filter, as the coefficients of filter are obtained from MATLAB FDA tool. The design is implemented and run in two modules without switching off the device and outputs are verified as the image is uploaded into the ROM in FPGA provided by the code generator. The size of the bit file size is reduced as shown below

File Name	Size	File Type	Modified Date
PRO_RECONMODULE.gise	15 KB	GISE File	5/3/2013 1:52 PM
PRO_RECONMODULE.vise	37 KB	Xilinx ISE Project	5/3/2013 4:27 PM
rpb.dgn	8 KB	BDG File	5/3/2013 1:52 PM
p2a	5 KB	BIT File	5/3/2013 1:52 PM
RP.bit	1 KB	BID File	5/3/2013 1:50 PM
RP_and_log	2 KB	CHD LOG File	5/3/2013 1:51 PM
rpb.dtc	1 KB	Test Document	5/3/2013 1:52 PM
RP.iso	1 KB	ISO File	5/3/2013 1:08 AM

Fig (6) size of BIT file

A significant reduction in the BIT file is obtained compared to previous methods. The size of the filter is constant.



Fig(7). (A) Input image, (B) Low pass filtered image, (C) High pass filtered image

We test the functionality of the FIR filter by testing it on images shown in fig. the image with its low pass and high pass filtered image are also shown.

V.CONCLUSION

The SR FIR filter design using systolized DA architecture is successfully modeled using Xilinx spartan-3E. This design is optimized in terms of speed and maximum frequency of operation. The coefficients of the filter can be changed at any time during the operation with the help of SR. the size of the .BIT file is reduced to a great extent compared to the previous designs. The size of .BIT file is in direct relation to the reconfiguration time. Hence the overall time taken by this design is very much acceptable and satisfactory.

REFERENCES

- [1] R.Wyrzykowski and S. Ovrmenko, "Flexible systolic architecture for VLSI FIR filters," Proc. Inst. Elect. Eng.—Compute. Digit. Techniques, Vol. 139, No. 2, pp. 170–172, Mar. 1992.
- [2] S.-S. Jeng, H.C. Lin, and S.M. Chang, "FPGA implementation of FIR filter using M-bit parallel distributed arithmetic," in Proc. IEEE Int. Symp. Circuits Systems (ISCAS), May 2006, p. 4.
- [3] Daniel Llamocca, 1 Marios Pattichis, 1 and G. Alonzo Vera2, "Partial Reconfigurable FIR Filtering System Using Distributed Arithmetic" International Journal of Reconfigurable Computing Volume 2010 (2010).
- [4] Pramod Kumar Meher, Shrutisagar Chandrasekaran, Abbas Amira, "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic," IEEE Trans on Signal Processing, Vol. 56, No. 7, July 2008.
- [5] S. A. White, "Applications of the distributed arithmetic to digital signal processing: tutorial review," IEEE ASSP Mag., Vol. 6, No. 3, pp. 5–19, Jul. 1989.
- [6] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York: Wiley, 1999.
- [7] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 4, 1973.
- [8] A. Peled and B. Liu, "A new hardware realization of digital filters," IEEE Trans. Acoust. Speech, Signal Process., Vol. 22, no. 6, pp. 456–462, Dec. 1974.
- [9] H. T. Kung, "Why systolic architectures?," IEEE Computer, vol. 15, no. 1, pp. 37–45, Jan. 1982.
- [10] Partial Reconfiguration User Guide,