FPGA BASED PARTIAL RECONFIGURABLE FIR FILTER DESIGN

J. Sheeba Rani, Member IEEE,
Indian Institute of Space Science and Technology,
Trivandrum.
sheeba@iist.ac.in

Sai Phalghun.C
Indian Institute of Space Science and Technology,
Trivandrum
saiphalghun@gmail.com

Abstract— **This paper proposes partial reconfigurable FIR filter design using systolic Distributed Arithmetic (DA) architecture optimized for FPGAs. To implement computationally efficient, low power, high speed Finite Impulse Response (FIR) filter a two dimensional fully pipelined structure is used. To reduce the partial reconfiguration time a new architecture for the Look-Up Table (LUT) in distributed arithmetic is proposed. The FIR filter is dynamically reconfigured to realize low pass and high pass filter characteristics by changing the filter coefficients in the partial reconfiguration module. The design is implemented using XUP Virtex 5 LX110T FPGA kit. The FIR filter design shows improvement in configuration time and efficiency.**

Keywords-Dynamic Partial Reconfiguration; FIR, Distributed Arithmetic; Systolic architecture; FPGA.

I. INTRODUCTION

Finite Impulse Response (FIR) filters, most fundamental components in digital signal processing are generally implemented on dedicated hardware rather than software for high speed computation. Because of the requirement of low power high speed implementation of FIR filter design in various embedded applications it is necessary to design reconfigurable filter architectures based on power or resources considerations, or simply to implement new functionality on the run.

FPGA is one such platform which allows adapting hardware resources to meet time-varying requirements in power, resources, or performance and at the same time maintaining a good speed of operation. In the literature [1,4,8,9,12,13] several efficient hardware architectures are developed using non reconfigurable/reconfigurable architectures. One of the challenges in partial reconfigurable architectures is the reconfigurable overhead, which is the time spent for reconfiguration on the fly because the computational complexity and reconfiguration time of the FIR filter increases with the increase in filter order and type of arithmetic used. As multipliers consume more power in Multiply and Accumulate (MAC) operation several multiplier less schemes have been proposed in the literature. [2, 3.4,7]. Distributed arithmetic method is one of the multiplier less technique which uses memories (RAMs, ROMs) or LUTs to store pre-computed values of coefficient operations. Although DA algorithm show a good set of characteristics with respect to speed and chip area the LUT complexity increases with respect to number of filter coefficients thus the memory requirement increases which greatly reduces the speed. All these designs have a common feature that they consist of output network adders which will not support pipelining. The architecture in [8] proposed a DA architecture which uses the concept of systolic arrays which was developed by H.T.Kung [2] and avoids the output adder network by introducing pipelining.

A reconfigurable FIR filter has been proposed earlier in [7,8] uses Xilinx reconfiguration tools on Virtex FPGA to achieve reconfiguration. But the design uses output adder network for the architecture in the static region of the FPGA. But the reconfiguration head (analogy to the size of the partial bit stream file in KB) is almost same in both cases. It is also shown in [8] that reconfiguration time depends on the size of the bit stream file. It utilizes the approach of dynamically reconfiguring the coefficients via LUTs using modular reconfiguration scheme. Self reconfigurable adaptive FIR filter in partial re-configurable platform is also proposed in the literature [8].

In this paper a systolic based DA architecture with dynamic reconfigurable module to reconfigure the filter coefficients is proposed. It is based on dynamically reconfiguring at the finest possible level, the LUTs that store the coefficients, with a small dynamic reconfiguration area. Here, we propose a new architecture for a LUT. in DA. By using this architecture the size of the *.BIT* file needed to upload is reduced to a great extent as compared to [9]. This paper is organized as follows. Section II describes the basics of DA and systolic Architecture for FIR filter. Section III gives the proposed Partial Reconfiguration Module followed by Implementation results and discussion in Section IV. Section V presents the conclusion.

II. SYSTOLIC DA ARCHITECTURE

We briefly outline here the conventional distributed arithmetic approach for inner-product computation, and thereafter derive a decomposition scheme for flexible DA-based systolic FIR filters.

An FIR filter can be described by the following equation

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

This is nothing but the inner product of inputs delayed each by a specific value with filter coefficients.

The input value x(n) can be expressed in the form of corresponding bits .

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

Applying equation (1) in equation (2)

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

By rearranging the terms in equation (3) we get

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

The term inside the braces in equation (4) is the sum of products of filter coefficients with the bits of inputs. If the number of input bits is N, then the sum of products can have 2^N values. All the possible values are stored in a LUT (look up table). The corresponding bit vector from the input act as input to the LUT. The outputs from the LUT are taken for all the *b* bits and are shift added to get the output. This can be represented by equation (5)

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

where the term $f(x_b[.])$ in equation (5) is the value from LUT, according to input bits. So instead of adders and multipliers and registers (to hold temporary results) we use a LUT (a memory unit typically a ROM) and a set of shift adders equal to the number of bits that are used for representation. The computation of MAC operation is very fast.The LUT should be of size 2^N where N is the order of filter. As the order of filter increases the LUT size increases exponentially and the time taken for memory fetch also increases which affects the operational frequency of the FIR filter. So we used a systolic decomposition technique to reduce the size of LUT by using multiple LUTs.

Suppose if the order of the filter N is a composite number which can be obtained by product of two other numbers M and P, then we decompose the 2^N LUT into P LUTs of size 2^M . The equation (4) turns out to be

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

We have used systolic array implementation for efficient mapping of equation (6) onto FPGA hardware. Systolic arrays (SA) are examples of VLSI special purpose processor networks that directly implement computationally expensive algorithms in hardware[2]. Systolic arrays are a systematic arrangement of small cells called processing elements where each processing element performs a simple task such as addition, multiplication, memory fetch etc and passes the data. The processing elements are simple finite state machines which generally perform simple tasks that generally does not consume more than a few clock cycles [8]. For the DA FIR filter our PE should contain a LUT and a memory fetch unit. The LUT results as per concept are to be added with the LUT fetches from other LUTs. So adder comes into the requirement. And finally there is a shift addition operation on the results from different bit positions. Here we use two different kinds of processing elements. They are schematically shown in the Fig 1 and 2.

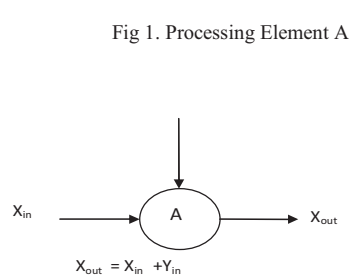

Fig 1. Processing Element A

The two processing elements used are shown in Fig 1 and 2. Processing element A performs the action of memory fetch and addition of memory fetch to the input value obtained from another PE A. PE B performs shift addition of input with another input.

This array consists of PE A arranged in a horizontal line. PE A consists of an LUT and a adder in actual hardware. There are 'P' PEs A and one PE *B* on a horizontal line. All

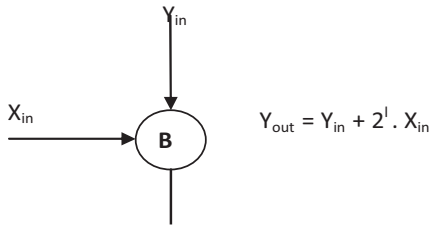

Fig 2. Processing Element B

these 'P' PE have different LUTs in them. This horizontal row of *'P'* PE and one PE *B* is replicated **B** (No. of Bits in a word) times. All the PE B^s are connected to one another in a columnar manner.

III. PROPOSED DA LUT ARCHITECTURE

The partially reconfigurable FIR filter is designed using systolic DA architecture. The block diagram of the proposed 9 tap FIR filter is shown Fig 3.The trapezium shapes indicate shift adder, while the triangle is adder and DA LUTs are present. The reconfiguration partition in [8] consists of LUT used in DA architecture.

Here we have come with an architecture in which the reconfiguration part consists of only filter coefficients but not the entire LUT as in [8]. By this we reduce the area of reconfigurable partition. We use FPGA resources for further computation of LUT entries. The time taken for this computation is negligible compared to reconfiguration time since the clock cycle of FPGA is only few nanoseconds.

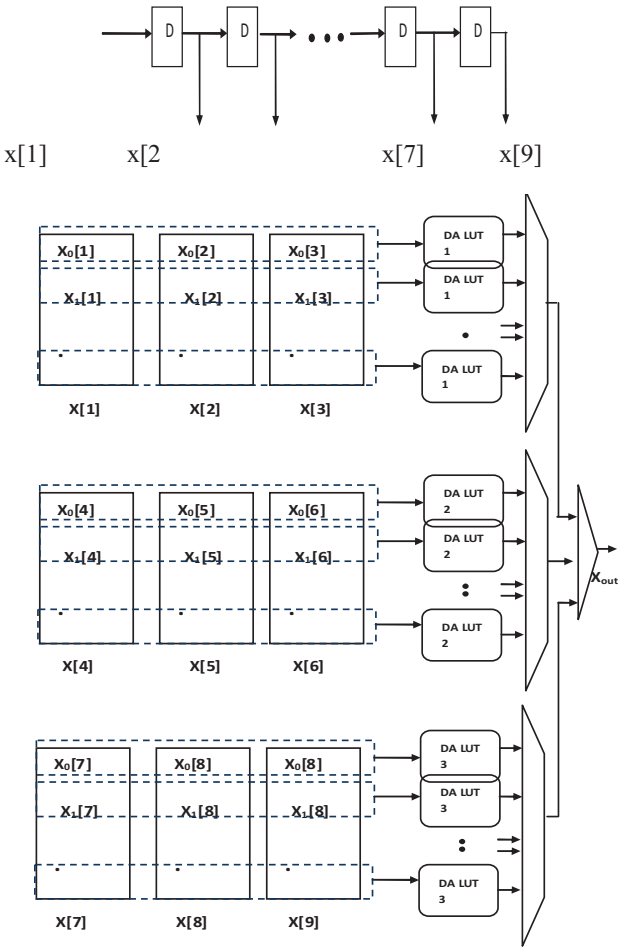

Fig 3. Systolic DA FIR Filter

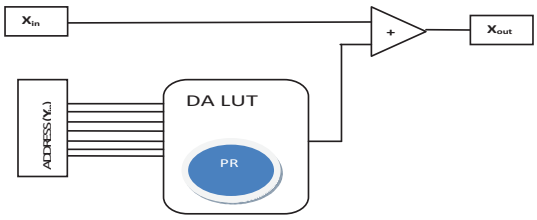

Fig 4. Processing Element 'A' architecture

Fig 4. shows the architecture of processing element A. It contains a LUT which is referred as DA LUT. The reconfigurable part is shaded in blue. The architecture of DA LUT is shown in Fig 5. This modified architecture of LUT which contains registers, adders and a memory. Memory is basically a RAM. In the design followed by D. [8] the RAM is placed in the reconfigurable partition which reconfigures the DA coefficients. In this proposed design the reconfigurable partition is modified to contain only filter coefficients rather than DA coefficients. The coefficients are obtained from the registers. The LUT values are then calculated by the network of adders and then are updated into the LUT RAM. The RAM has inputs address, write enable and read enable. This offers some advantages in speed of reconfiguration.

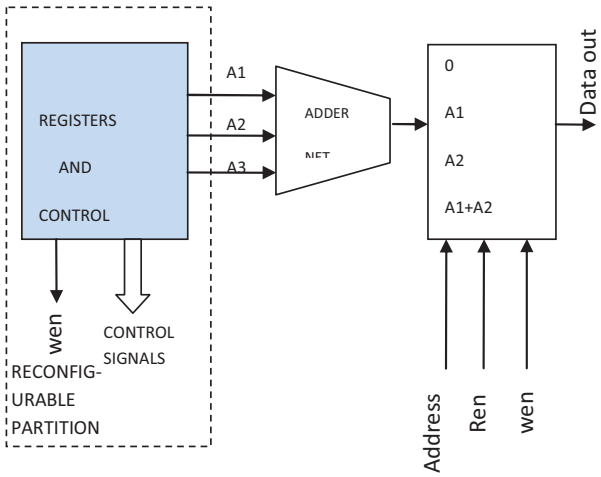

Fig 5. DA LUT architecture

Whenever the reconfigurable area is reconfigured, a set of control signals and write enable are generated. When reconfigured the some variables are assigned to their initial values**.** The control signals mainly consist of write enable and other enable signals. The write enable signal enables writing data into the RAM. Then the values in the LUT RAM are changed. The system functions now as a different filter.

The advantage of this architecture over the existing architecture is that the reconfigurable area has been exponentially decreased. So, the partial BIT file has its size reduced to a great extent. Therefore the time taken to reconfigure the FPGA is greatly reduced.

IV. IMPLEMENTATION AND RESULTS

The hardware used for developing a partial reconfigurable FIR filter is Virtex-5 FPGA on LX110T evaluation platform provided by Xilinx.inc and developed by Digilent.inc. The software used for developing and prototyping of the design are Xilinx System Generator, MODELSIM ,Xilinx ISE 13.2 , PlanAhead [10].

A filter of order 9 has been designed. All the numbers are represented in fixed point representation of 12 bits and binary point at 11. We followed the proposed reconfiguration scheme where only a module containing coefficients is reconfigured. Two reconfigurable modules have been created for testing. In one module we used the coefficients of a low pass filter and in the other module we have implemented a high pass filter. The filter coefficients are obtained from MATLAB FDA tool. The model is tested for an image. The Image is uploaded into a Read only Memory in FPGA provided by Core Generator and design is implemented and run with two different modules without switching off the device and the outputs are verified.

The size of the BIT file obtained was of 5KB, which is small when compared to the BIT file size obtained by [8]. This is shown in Fig 6.

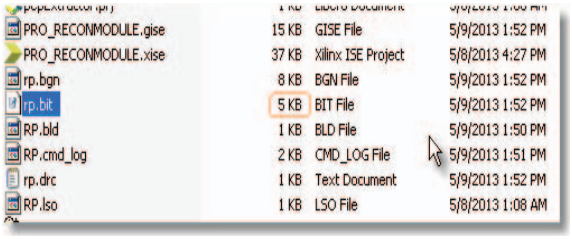

Fig 6: BIT file size

As the order of the filter increases the BIT file size obtained by [8] increases at high rate. But here the size of the file remains almost unchanged.

The functioning of the FIR filter is verified by testing the FIR filter on images. Fig 7. shows the original image, its low pass filtered image and high pass filtered image respectively.

a b c

Fig 7 (a) Input image (b) Low pass filtered image (c) High pass filtered image

V. CONCLUSION

A PR FIR Filter using systolized DA architecture has been implemented using Xilinx Virtex-5 FPGA. The FIR filter is optimized with respect to speed and maximum frequency of operation. The filter coefficients can be changed amid the operation using PR. The size of the .BIT file used for PR has been reduced from 43000 bytes to 5000 bytes. Since the size of .BIT is directly proportional to reconfiguration time, the time taken by PR is reduced to a great extent by using the proposed architecture.

REFERENCES

[1] R.Wyrzykowski and S. Ovramenko, "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. 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.

[3] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation.* New York: Wiley, 1999.

[4] 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.

[5] 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

[6] H. T. Kung, "Why systolic architectures?," *IEEE Computer, vol. 15,*no. 1, pp. 37–45, Jan. 1982.

[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] Daniel Llamocca,1 Marios Pattichis,1 and G. AlonzoVera2, "Partial Reconfigurable FIR Filtering System Using Distributed Arithmetic"International Journal of Reconfigurable Computing Volume 2010 (2010),.

[9] Pramod Kumar Meher, Shrutisagar Chandrasekaran, Abbes 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.

[10] Partial Reconfiguration User Guide, www.xilinx.com