Realization of Distributed Arithmetic Based Reconfigurable Digital FIR Filter

Mayur B Kachare **Dineshkumar U Adokar**

Abstract— **This paper presents efficient distributed arithmetic (DA) based reconfigurable implementation of finite-impulse response (FIR) filters whose filter coefficients change during runtime. Conventionally, for reconfigurable DA based implementation of FIR filter, the lookup tables (LUTs) are required to be implemented in RAM and the RAM-based LUT is found to be costly for ASIC implementation. Therefore, a shared LUT design is proposed to realize the DA computation.**

Instead of using separate registers to store the possible results of partial inner products for DA processing of different bit positions, registers are shared by the DA units for bit slices of different weightage. A distributed RAM based design is used for the field programmable gate array (FPGA) implementation of the reconfigurable FIR filter, which supports up to 91 MHz input sampling frequency.

Index Terms—**Distributed Arithmetic (DA), FIR filters, Look up table, FPGA.**

I. INTRODUCTION

A reconfigurable finite-impulse response (FIR) filter whose filter coefficients dynamically change during runtime plays an important role in the software defined radio systems [1,2], multichannel filters [3], and digital up/down converters [4]. However, the well known multiple constant multiplication based technique [5], which is widely used for the implementation of FIR filters, cannot be used when the filter coefficients dynamically change. On the other hand, a general multiplier based structure requires a large chip area and consequently enforces a limitation on the maximum possible order of the filter that can be realized for high throughput applications.

A distributed arithmetic (DA) based technique [6] has gained substantial popularity in recent years for its 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 are a sequence of lookup table (LUT) accesses followed by shift accumulation operations of the LUT output. The conventional DA implementation used for the implementation of an FIR filter assumes that impulse response coefficients are fixed, and this behaviour makes it possible to use ROM-based LUTs. The memory requirement for DA based implementation of FIR filters, however, exponentially increases with the filter order. To eliminate the problem of such a large memory requirement, systolic decomposition techniques are used for DA based implementation of long length convolutions and FIR filter of large orders [7, 8].

II. DIGITAL FILTERS

Digital filters are used extensively in all areas of electronic industry. This is because digital filters have the potential to attain much better signal to noise ratios than analog filters and at each intermediate stage the analog filter adds more noise to the signal, the digital filter performs noiseless mathematical operations at each intermediate step in the transform [5]. The digital filters have emerged as a strong option for removing noise, shaping spectrum, and minimizing inter symbol interference in communication architectures. These filters have become popular because their precise reproducibility allows design engineers to achieve performance levels that are difficult to obtain with analog filters. FIR and IIR filters are the two common filter forms. A drawback of IIR filters is that the closed form IIR designs are preliminary limited to low pass, band pass, and high pass filters, etc. Furthermore, these designs generally disregard the phase response of the filter. For example, with a relatively simple computational procedure one may obtain excellent amplitude response characteristics with an elliptic low pass filter while the phase response will be very nonlinear. In designing filters and other signal processing system that pass some portion of the frequency band undistorted, it is desirable to have approximately constant frequency response magnitude and zero phases in that band [2]. For casual systems, zero phases are not attainable, and consequently, some phase distortion must be allowed.

As the effect of linear phase with integer slope is a simple time shift. A nonlinear phase, on the other hand, can have a major effect on the shape of a signal, even when the frequency response magnitude is constant. Thus, in many situations it is particularly desirable to design systems to have exactly or approximately linear phase. Compare to IIR filers, FIR filters can have precise linear phase. Also, in the case of FIR filters, closed form design equations do not exist. While the window method can be applied in a straight forward manner, some iteration may be necessary to meet a prescribed specification. The window method and most algorithmic methods afford the possibility of approximating more arbitrary frequency response characteristics with little more difficulty than is encountered in the design of low pass filters [4]. Also, it appears that the design problem for FIR filters is much more under control than the IIR design problem because there is an optimality theorem for FIR filters that is meaningful in a wide range of practical situations. The magnitude and phase plots provide an estimate of how the filter will perform; however, to determine the true response, the filter must be simulated in a system model using either calculated or recorded input data. The creation and analysis of representative data can be a complex task. Most of the filter algorithms require multiplication and addition in real-time. The unit carrying out this function is called MAC (multiply accumulate). Depends on how good the MAC is the better MAC the better performance can be obtained. Once a correct filter response has been determined and a coefficient table has been generated, the second step is to design the hardware architecture. The hardware designer must choose between area, performance, quantization, architecture, and response.

Digital filter design problem involves the determination of a set of filter coefficients to meet a set design specifications. These specifications typically consist of the width of the pass band and the corresponding gain, the width of the stop band(s) and the attenuation therein; the band edge frequencies (which give an indication of the transition band) and the peak ripple tolerable in the pass band and stop band(s).There are many techniques for selecting coefficients.

The main advantage of the FIR filter structure is that it can achieve exactly linear phase frequency responses. That is why almost all design methods described in the literature deal with filters with this property. Since the phase response of linear phase filters is known, the design procedures are reduced to real valued approximation problems, where the coefficients have to be optimized with respect to the magnitude response only. A digital FIR filter is characterized by,

$$H(z) = \sum_{n=0}^{N} h(n)\, z^{-n}, n = 0, 1, \dots, N \qquad \dots(1)$$

Where, N is the order of the filter which has (N+1) number of filter’s impulse response coefficients, h(n). The values of h(n) will determine the type of the filter, e.g., low pass, high pass, band pass etc. The values of h(n) are to be determined in the design process and N represents the order of the polynomial function.

The even order FIR LP filter design with h(n) as positive even symmetric. Because the h(n) coefficients are symmetrical, the dimension of the problem is halved. Thus, (N/2+1) number of h(n) coefficients are actually optimized, which are finally concatenated to find the required (N+1) number of filter coefficients. An ideal filter has a magnitude of one in the pass band and a magnitude of zero in the stop band. Error fitness function is formed by the errors between the frequency responses of the ideal filter and the designed approximate filter. In each iteration of the optimization algorithm, error fitness values of particle vectors are calculated and used for updating the particle vectors with new coefficients h(n). The final particle vector obtained after a certain number of iterations or after the error fitness is below a certain limit is considered to be the optimal result, yielding an optimal filter. Various filter parameters which are responsible for the optimal filter design are stop band and pass band normalized frequencies (ω_s, and ω_p), pass band and stopband ripples (δ_p and δ_s), stop band attenuation and transition width. These parameters are decided by the filter coefficients. Several scholars have investigated and developed algorithms in which N, δ_p, and δ_s are fixed while the remaining parameters are optimized [16].

The FIR filter is one of the most fundamental components in digital signal processing. The block schematic is shown in Fig.1. Due to the high amount of MAC operations, the computational power of many real-time applications can only by realized by using the parallel nature of like FPGA. To reduce the performance gap between FPGAs, digital filters were one of the driving forces to push embedded multipliers or DSP blocks into the FPGA fabric. The price for those fixed coarse grained blocks is their inflexibility and limited quantity. However, in many applications like for digital filters, the multiplications have to be performed only with constants that may be only reconfigured from time to filters for decimation or interpolation like poly-phase FIR filters or frequency variable filters time which can be used to reduce the complexity [10]. Examples are multistage as needed in telecommunications, digital audio, medical, radar, sonar and instruments [2].

The conventional single-rate FIR version of the core computes the convolution sum as below:

$$y(k) = \sum_{n=0}^{N} a(n)x(k-n) \qquad k = 0,1,\dots \qquad \dots(2)$$

Where N is the number of filter coefficients. The conventional tapped delay line realization of this inner product is shown in Fig. 2. It is a useful conceptualization of the computation performed by the core; the actual FPGA realization is quite different. A distributed arithmetic realization [1, 2] is employed. This approach employs no explicit multipliers in the design, only lookup tables, shift registers, and a scaling accumulator.

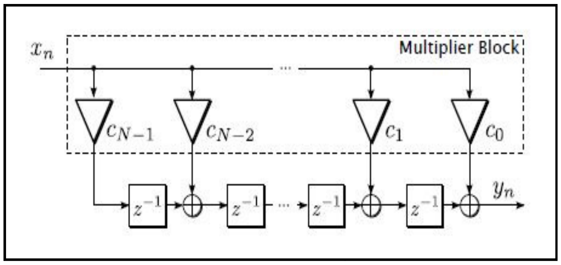

Fig. 1 FIR filter in transposed form [10]

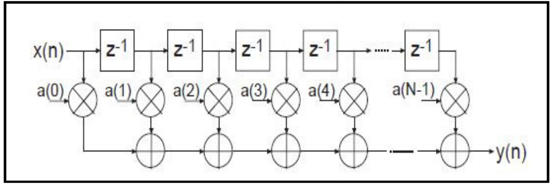

Fig. 2 Conventional Tapped-Delay Line FIR Filter [15]

III. SYSTEM DEVELOPMENT

FPGA technology has tremendously grown from a dedicated hardware to a heterogeneous system, which is considered to be a popular choice in communication base stations instead of being just a prototype platform.

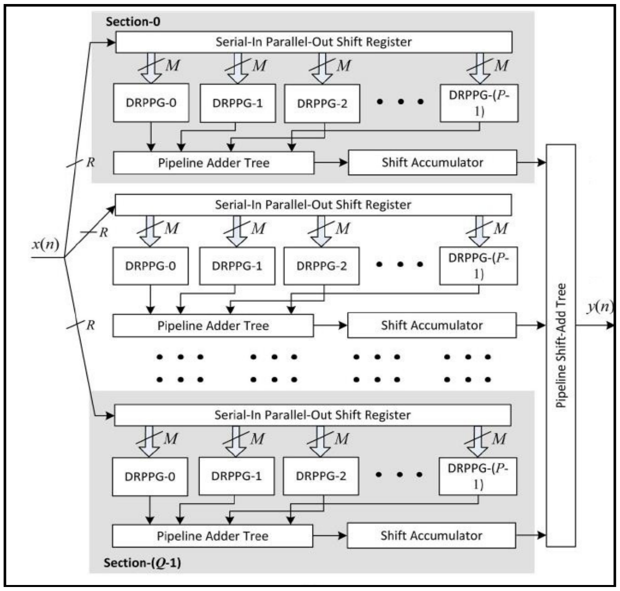

Fig. 3 Structure of the DA based FIR filter [1]

The proposed reconfigurable FIR filter may be also implemented as part for the complete system on FPGA. Therefore, here introduced a reconfigurable DA based FIR filter for FPGA implementation.

Figure 3 show the structure [1] of the time-multiplexed DA based FIR filter using DRAM. To implement [11], the distributed arithmetic structure has Q sections, and each section consists of P DRAM based RRPGs (DRPPGs) and the PAT to calculate the rightmost summation, followed by shift accumulator that performs over R cycles according to the second summation. However, one can use dual port DRAM to reduce the total size of LUTs by half since two DRPPGs from two different sections can share the single DRAM.

The hardware and time complexities of the proposed structures and the existing DA based structures of an FIR filter. A conventional DA based structure [6], DA based systolic structures [7,8], and a DA based structure using carry save adder [12] are compared with the DA based structures.

The DA FIR filter for FPGA implementation guarantees a higher throughput than the structures of [6] and [12] for $R < L$ and fewer adders and smaller LUT than the systolic structure. The structure shown in figure 3 involves fewer adders and registers, but marginally larger LUTs.

CONCLUSION

The efficient schemes for high-throughput reconfigurable DA-based implementation of FIR digital filters shows that the hardware cost could be substantially reduced by sharing the same registers by the DA units for different bit slices. The distributed arithmetic based reconfigurable FIR filter for FPGA implementation supports up to 91 MHz input sampling frequency.

REFERENCES

[1] Sang Yoon Park and Pramod Kumar Meher, “Efficient FPGA and ASIC Realizations of DA Based Reconfigurable FIR Digital Filter,” IEEE transactions on circuits and systems-ii, express briefs, vol. 61, no. 7, pp. 511-515, Jul. 2014.

[2] T. Hentschel, M. Henker, and G.Fettweis, “The digital front end of software radio terminals,” IEEE pers. commun. mag., vol. 6, no. 4, pp. 40 46, Aug. 1999.

[3] K. H. Chen and T. D. Chiueh, “A low power digit based reconfigurable FIR filter,” IEEE transactions on circuits systems-ii, express briefs, vol. 53, no. 8, pp. 617–621, Aug. 2006.

[4] L. Ming and Y. Chao, “The multiplexed structure of multichannel FIR filter and its resources evaluation,” in Proc. international conference CDCIEM, pp. 764–768, Mar. 2012.

[5] I. Hatai, I. Chakrabarti, and S. Banerjee, “Reconfigurable architecture of a RRC FIR interpolator for multi-standard digital up converter,” in Proc. IEEE 27th IPDPSW, pp. 247–251, May 2013.

[6] A. G. Dempster and M. D. Macleod, “Use of minimum-adder multiplier blocks in FIR digital filters,” IEEE transactions on circuits and systems-ii, analog digital signal processing, vol. 42, no. 9, pp. 569–577, Sep. 1995.

[7] S. A. White, “Applications of distributed arithmetic to digital signal processing: A tutorial review,” IEEE ASSP Mag., vol. 6, no. 3, pp. 4–19, Jul. 1989.

[8] P. K. Meher, “Hardware-efficient systolization of DA-based calculation of finite digital convolution,” IEEE transactions on circuits and systems-ii, express briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.

[9] P. K. Meher, S. Chandrasekaran, and A. Amira, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic,” IEEE transactions on signal processing, vol. 56, no. 7, pp. 3009–3017, Jul. 2008.

[10] M. Kumm, K. Moller, and P. Zipf, “Dynamically reconfigurable FIR filter architectures with fast reconfiguration,” in Proc. 8th international workshop ReCoSoC, pp. 1–8, Jul. 2013.

[11] E. Ozalevli, W. Huang, P. E. Hasler, and D. V. Anderson, “A reconfigurable mixed-signal VLSI implementation of distributed arithmetic used for finite-impulse response filtering,” IEEE transactions on circuits and systems-i, reg. papers, vol. 55, no. 2, pp. 510–521, Mar. 2008.

[12] 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-i, reg. papers, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.

[13] P. K. Meher and S. Y. Park, “High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic,” in Proc. IEEE/IFIP 19th international conference on VLSI-SOC, pp. 428–433, Oct. 2011.

[14] Design ware building block IP user guide, Synposys, Inc., Mountain View, CA, USA, 06-SP2, 2012.

[15] LogiCORE IP FIR Compiler v5.0, Xilinx, Inc., San Jose, CA, USA, 2010.

[16] S. Mandal, S.P. Ghoshal, R. Kar and D. Mandal, “Novel Particle Swarm Optimization for Low Pass FIR Filter Design”, WSEAS transactions on signal processing, Issue 3, Volume 8, pp.111-120, Jul. 2012.

[17] C. Bauer, (2011) “Interactive Digital Signage – An Innovative Service and Its Future Strategies”, Tirana, 2011 International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), 7-9 September 2011, pp 137-142.