

DA based Efficient Parallel Digital FIR Filter Implementation for DDC and ERT Applications

E. Chitra¹, T. Vigneswaran²

¹Asst. Prof.,

SRM University, Dept. of Electronics and Communication Engineering,
SRM University, Chennai, INDIA

²Professor,

Dept. of Electronics and Communication Engineering,
VIT University, Chennai
INDIA

Abstract – This paper discusses FPGA implementation of finite impulse response (FIR) filters using their application in Digital Down-Converters (DDCs) for software radio and in (Electrical Resistance Tomography) ERT. The implementation is based on distributed arithmetic (DA) which substitutes multiply and accumulate operations with a series of look-up-table (LUT) accesses. Distributed arithmetic provides a multiplication-free method for calculating inner products of fixed-point data, based on table lookups of pre-calculated partial products. The implementation results are provided to demonstrate a high-speed and low power proposed architecture. The proposed DDC is implemented in VHDL and verified via simulation. The proposed method offers average reductions of 30% in the number of LUT, 42% reduction in occupied slices and 38% reduction in the number of gates needed for low pass FIR filter implementation method. The proposed DA based FIR filter can be used in electrical resistance tomography (ERT) system: it is the time delay of the filter that affects the real-time performance of the conventional ERT system. The proposed design shows 14% reduction in delay as compared to conventional logic based DA architecture. Though there is a power trade-off but there is significant improvement in area and delay parameters.

Keywords: Digital down converters, Distributed arithmetic, LUT, Software radio, Finite impulse response and Electrical resistance tomography system.

I. Introduction

Finite impulse response (FIR) digital filters are common components in many digital signal processing (DSP) systems and are used to perform signal preconditioning, anti-aliasing, band selection, decimation/interpolation, low-pass filtering, and video convolution functions [1-3]. In FIR filter applications, arithmetic elements for operations such as addition, multiplication and delay (storage) are commonly required. Digital signal processing algorithms rely heavily on the efficient computation of inner products. Very efficient methods have been developed for implementation of digital filters in FPGAs or custom ICs. Digital filtering is the main task in IF processing. The computational complexity of finite impulse response (FIR) filters used in the IF processing block is dominated by the number of adders (subtractors) employed in the multipliers. The use of SDR technology is predicted to replace many of the traditional methods of implementing transmitters and receivers while offering a wide range of advantages including adaptability, reconfigurability, and multifunctionality encompassing modes of operation, radio frequency bands, air interfaces, and waveforms [4]. Research in this field is mainly directed towards improving the architecture and the computational efficiency of SDR systems. The most computationally intensive part of an SDR receiver is the channelizer since it operates at the highest sampling rate [5]. The key functional units in a digital filter are delay, adder, and multiplier – out of which the multiplier dominates the hardware complexity. The complexity of the FIR multiplier is dominated by the number of adders (subtractors) employed in the coefficient multipliers. The contributions of this paper can be summarized as follows: An efficient scheme using DA based implementation for FIR filters in DDC and ERT is proposed. By employing this technique, it is shown that the delay, area and power consumption of the filters can be minimized.

This paper is organized as follows: In section II, a brief background on DA and parallel FIR filters. In section III, the DDC example system and FIR filters for ERT are explained. The DA for implementation of FIR filters is discussed in section IV. In section V, The multiplexer based DA scheme is presented. The results are illustrated in section VI. Section VII provides our conclusions.

II. Background study

A. Distributed arithmetic

Distributed arithmetic is a multiplication free method applicable to fixed-point data, and is based on table lookups of pre-calculated partial products [6]. Distributed Arithmetic (DA) [7] is a method often preferred since it eliminates the need for hardware multipliers and is capable of implementing large filters with very high throughput. Also, DA filters achieve these advantages while retaining full precision, unlike filters using reduced sums and differences of powers of two. Fig. 1 illustrates basic concept of DA. DA provides multiplier free multiplication by using bit serial computation by storing all possible combination sums of filter weights in LUT. Distributed arithmetic a possible candidate for low power applications because it allows replacement of costly multiplies with shifts and table lookups [6]. The battery lifetime of portable electronics has become a major design concern as more functionality is incorporated into these devices. Therefore, the shrinking power budget of modern portable devices requires the use of low-power circuits for signal processing applications. The signal processing functions employed in these devices include finite-impulse response (FIR) filters, discrete cosine transforms (DCTs), and discrete Fourier transforms (DFTs). The common feature of these functions is that they are all based on the inner product. Digital signal processing (DSP) implementations typically make use of multiply-and-accumulate (MAC) units for the calculation of these operations, and the computation time increases linearly as the length of the input vector grows.

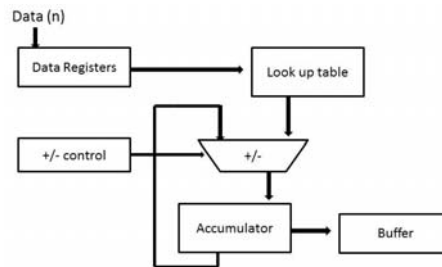


Fig. 1 Basic concept of distributed arithmetic

B. Parallel FIR filters

A FIR filter can be mathematically expressed by the equation (1) [8].

$$y[n] = \sum_{i=0}^{N-1} h[i]x[n-i] \tag{1}$$

where x represents the input signal, h the filter coefficients, y the output signal, $y[n]$ is the current output sample, and N is the number of taps of the filter. This is a convolution operation of the filter coefficients along with the signal. In the sequential implementation a set of multiply-and-accumulate (MAC) operations is performed for each sample of the input data signal, multiplying the N delayed input samples by coefficients and summing up the results together to generate the output signal. In parallel implementations, have two main architectures. The first one consists of unrolling of MAC loop where we have several delayed versions of the input signal entering in a fully parallel multiplier block, followed by a summation block. The other one consists of a multiplier block, which takes the same input signal and delivers each output to an input of a delayed summation block. Fig. 2 shows the basic block diagram of parallel FIR filtering.

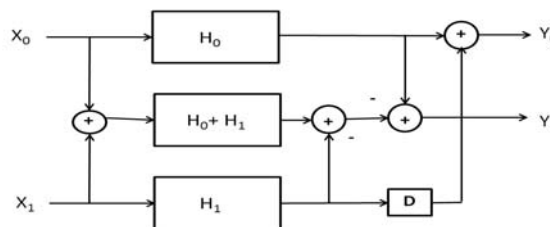


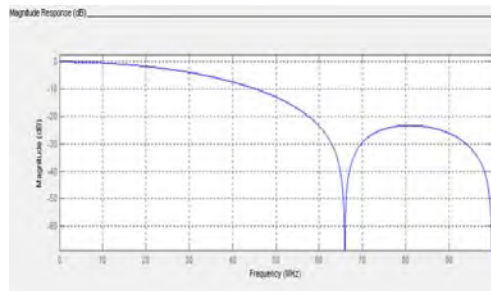
Fig. 2 Block diagram of parallel FIR filtering

III. Applications

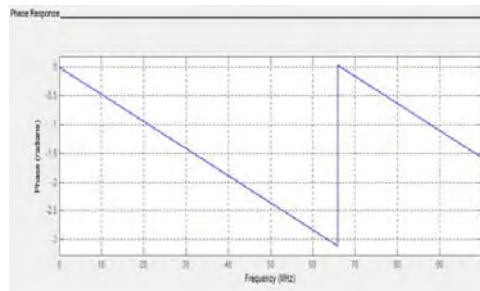
A. Digital down converter

Software radio receivers [9] require mixing, filtering and down sampling of received signals to allow data to be processed at a suitable rate. Part of this process can be achieved in FPGAs using a Digital Down-Converter (DDC).

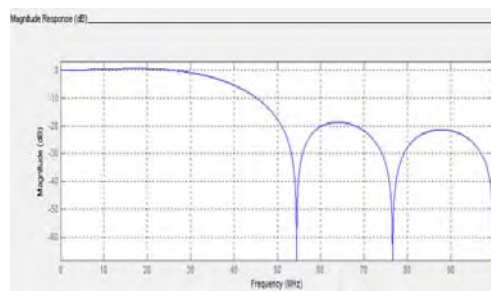
As well as mixing the incoming real signal from the ADC to extract the complex signal, a DDC must filter the complex signal to reject image components introduced by the mixing process and then down sample. For maximum software radio flexibility, the ADC, mixer and filters should sample as quickly as possible. Hence, if the DDC is implemented on an FPGA, full-parallel techniques can be used to reach the required sampling rates. The calculation of low pass filter coefficients for DDC specifications used in this paper are calculated using MATLAB, sampling frequency 200MHz with cutoff frequency of 40Mhz and attenuation band 6dB using Kaiser window. The phase and magnitude response of 4-tap and 8-tap filters are shown in Fig.3.



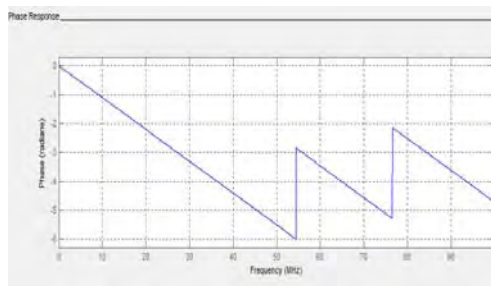
(a)



(b)



(c)



(d)

Fig. 3 FIR filter responses for DDC (a) 4-tap low pass FIR filter magnitude response (b) 4-tap low pass FIR filter phase response (c) 8-tap low pass FIR filter magnitude response (d) 8-tap low pass FIR filter phase response

B. FIR filters for Electrical resistance tomography system

ERT is used to achieve visual detection through boundary sensors array to obtain the real-time distribution of the sensing field. For the use of the sinusoidal signal as the inject current, the demodulation and low-pass filter are needed in the data acquisition system, which were always implemented by analog devices. This not only complicates the structure but also weakens the real-time performance [10]. The time delay of the analog filter and demodulation is the main problem that affects the data acquisition speed. As the development of the integrated circuit, digital technology has become the main method for signal processing. Nowadays the digital FIR filter is widely used in electronic instruments, for it can solve the problem caused by the time delay with well dynamic response. Fig. 4 describes the magnitude and phase response of the low pass FIR filter used in ERT. Hence, in this system, the FIR filter and the demodulation can also be implemented in FPGA digitally. For this the simulation is done using Spartan 3 FPGA device.

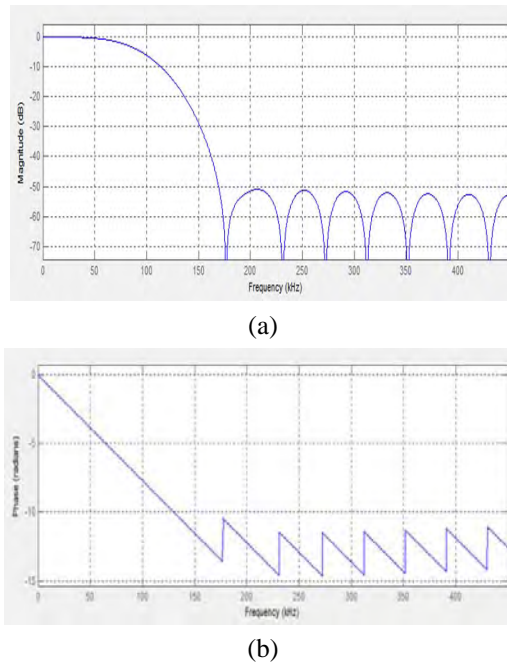


Fig.4 FIR filter response for ERT system (a)Magnitude response of low pass FIR filter (b) Phase response of low pass FIR filter

III. Distributed arithmetic based filtering scheme

Distributed Arithmetic was first brought up by Croisier [11], and was extended to cover the signed data system by Liu, and then was introduced into FPGA design to save MAC blocks with the development of FPGA technology. Fig. 5 illustrates the concept of distributed arithmetic.

If $h[n]$ is the filter coefficient and $x[n]$ is the input sequence to be processed, the N-length FIR filter can be described as:

$$y = \langle h, x \rangle = \sum_{n=0}^{N-1} h[n]x[n] \tag{2}$$

Distributed Arithmetic is introduced into the design of FIR filters as follows. In the two's complement system, $x[n]$ can be described as:

$$x[n] = -2^B x_B[n] + \sum_{b=0}^{B-1} 2^b x_b[n] \tag{3}$$

Substitute eq.(3) into eq.(2) yields:

$$y = -2^B x_B[n]h[n] + \sum_{b=0}^{B-1} h[n] \sum_{n=0}^{N-1} 2^b x_b[n] \tag{4}$$

The (5) can be changed into another form:

$$\sum_{b=0}^{B-1} h[n] \sum_{n=0}^{N-1} 2^b x_b[n] = \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} h[n] x_b[n] \tag{5}$$

Substituting (6) into (5) yields to the final form of Distributed Arithmetic:

$$y = -2^B x_B[n]h[n] + \sum_{b=0}^{B-1} 2^b \sum_{N=0}^{N-1} h[n] x_b[n] \tag{6}$$

It is conserve that the values of $\sum_{n=0}^{N-1} h[n] x_b[n]$ into a LUT unit and then callout the relevant value according to the input data to save MAC blocks. And then the weighted sum of $\sum_{n=0}^{N-1} h[n] x_b[n]$ is calculated through shift registers, the result is $\sum_{b=0}^{B-1} 2^b \sum_{N=0}^{N-1} h[n] x_b[n]$. In signed system, the signed bit should be taken into consideration so $-2^B x_B[n]h[n]$ is also added. As a result, the final form of Distributed Arithmetic is defined as (6) and the implementation can be achieved on FPGA through LUT units.

IV. Proposed DA based filtering scheme using multiplexer

Fig. 5 shows proposed multiplexer based DA filtering scheme. The basic LUT-DA scheme on an FPGA would consist of three main components: the input registers, the 4-input LUT unit and the shifter/accumulator unit. Additionally, it would require a control unit to manipulate the filter operation, and an adder tree unit to perform addition on partial filter results. Applying this approach in (4) the 4-input LUT unit will not be directly accessed instead 2-input LUT is used based on multiplexer select. The particular 2-input LUT is selected which represent all the possible sum combinations of filter coefficients. Though there is a power trade off but it implies about 50% reduction in the number of LUT used with increased speed. To evaluate the performance of the proposed scheme, 4-tap and 8-tap low pass FIR filters for DDC are implemented using VHDL and synthesis is carried out in XILINX-ISE8.1i.

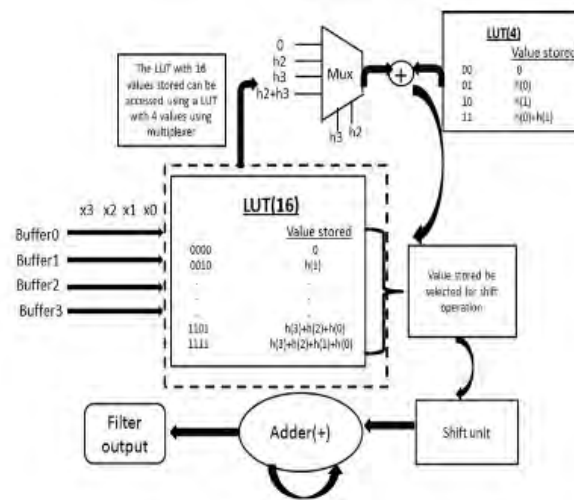


Fig.5 Multiplexer based DA filtering scheme

VI. Results and discussion

The simulation has been done using MODEL SIM 6.4 and XILINX Integrated Software Environment (ISE) is used for performing synthesis and implementation of designs using ‘Spartan-3’ device. The power analysis has been done using XILINX XPOWER tool. The filter coefficients for the DDC low pass filter application are calculated using MATLAB. The evaluation of device utilization using proposed DA architecture can be comprehended easily with the help of the results in Table I.

1) Table I shows the XILINX device utilization for 4-tap, 8-tap, 16-tap and 32-tap FIR implementation, it is observed that the proposed gate based architecture implies 30% reduced LUT, 45% reduced slices utilization and 40% reduced number of gates.

2) Fig. 6I represents the delay comparison for 4-tap, 8-tap, 16-tap and 32-tap filter designed using conventional DA and proposed gate based DA method. The proposed method outperforms by 15% speed improvement.

Compared with the traditional algorithm, distributed algorithm can greatly reduce the size of the hardware circuit, as well as it is easy to implement pipelining technology and improve the operation speed of the circuit. The key factor that affects the data acquisition rate of the conventional ERT system is the time delay of filter, which is reduced using proposed logic shown in Figure 6. Also compared with the analog filter, the time delay is reduced greatly by using the digital filter. As for the ERT system, the inject current has a frequency of 50k Hz and a sample frequency of 900k Hz. Hence, the cut-off frequency of the low-pass FIR filter would be 100k Hz, which could entirely meet the needs of the data acquisition system. And also it should have well frequency response and good cut-off capacity and performance improvement. The FPGA implementation of proposed DA based FIR filter using Spartan 3 device and the power consumption results are shown in Figure 7. The proposed method can be easily comprehended for the higher order filters.

Table I
Device utilization results for FIR filter (XILINX FPGA XC3S200-4FT256)

<i>4-tap Low pass FIR filter</i>			
	<i>Number of LUT</i>	<i>Occupied slices</i>	<i>Gates</i>
<i>Conventional parallel implementation</i>	267	190	2013
<i>Proposed parallel implementation</i>	225	169	1817
<i>8-tap Low pass FIR filter</i>			
	<i>Number of LUT</i>	<i>Occupied slices</i>	<i>Gates</i>
<i>Conventional parallel implementation</i>	358	248	2853
<i>Proposed parallel implementation</i>	319	210	2552
<i>16-tap Low pass FIR filter</i>			
	<i>Number of LUT</i>	<i>Occupied slices</i>	<i>Gates</i>
<i>Conventional parallel implementation</i>	443	335	3541
<i>Proposed parallel implementation</i>	403	303	3312
<i>32-tap Low pass FIR filter</i>			
	<i>Number of LUT</i>	<i>Occupied slices</i>	<i>Gates</i>
<i>Conventional parallel implementation</i>	535	410	4161
<i>Proposed parallel implementation</i>	489	378	3997

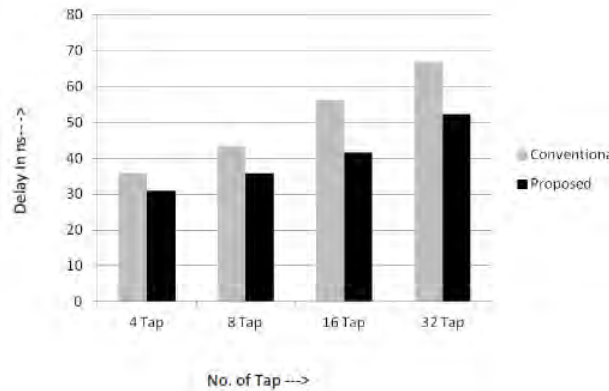


Fig. 6 Delay Results for low pass FIR filter (XILINX FPGA XC3S200-4FT256)

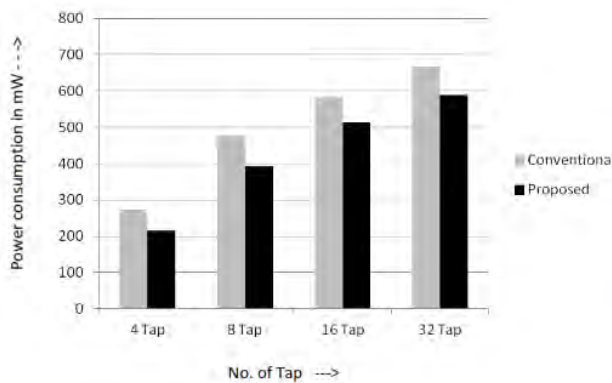


Fig. 7 Power Results for low pass FIR filter (XILINX FPGA XC3S200-4FT256)

VII. Conclusion

In this paper, presented an efficient DA based scheme which is used to implement FIR filters in DDC and ERT systems. The device utilization of the proposed architecture is relatively less since it used split LUT technique with multiplexer select logic. Our method is implemented for till 32 tap and can be even extended more. A high speed and less area implementation is achieved. The test results indicate that the designed filter using proposed distributed arithmetic can work stable with high speed and can save almost 40 percent hardware resources. The delay improvement turns out very useful for the ERT systems. Meanwhile, it is very easy to transplant the filter to other applications through modifying the order parameter and other parameters, and therefore have great practical applications in digit signal processing.

References

- [1]S. N. Merchant and B. V. Rao, Distributed arithmetic architecture for image coding, Proc. IEEE Int. Conf. TENCON'89,1989.
- [2]H. Q. Cao and W. Li, VLSI implementation of vector quantization using distributed arithmetic, Proc. IEEE Int. Symp. Circuits Syst., 1996.
- [3]S. A. White, Applications of Distributed Arithmetic to Digital Signal Processing, A Tutorial Review-IEEE ASSP Magazine, pp. 4-19, 1989.
- [4]W. H. W. Tuttlebee, Software Defined Radio: Enabling Technologies, New York, Wiley, 2002.
- [5]J. Mitola, Software Radio Architecture. New York: Wiley,2000
- [6]New, A distributed arithmetic approach to designing scalable DSP chips, EDN, pp. 107-114 , 1995.
- [7]W. P. Burlison, L. L. Scharf, A VLSI Design Method for Distributed Arithmetic, VLSI Sig. Proc., Vol. 2, pp. 235-252, 1991
- [8]Cheng and K. K. Parhi., Further complexity reduction of parallel FIR filters. Proc.IEEE Int. Symp. Circuits Syst., Kobe, Japan, 2005, pp. 1835-1838, 2005.
- [9]K. S. Yeung and S. C. Chan, The design and multiplier-less realization of software radio receivers with reduced system delay, IEEE Trans. Circuits Syst. I, vol. 51, no. 12, pp. 2444-2459, 2004.
- [10] Dickin, and M. Wang, Electrical Resistance Tomography for Process Applications, Measurement Science and Technology, vol.7, pp.247-260, January 1996.
- [11] Uwe Meyer-Baese,Digital signal processing with FPGA, Beijing:Tsinghua University Press,50, 51, 2006.