AN AREA EFFICIENT VLSI ARCHITECTURE FOR DA-BASED RECONFIGURABLE FIR FILTER

K. AMRUTHA VARSHINI, K. PADMA VASAVI

Abstract: Digital Signal processing is one of the most dynamic areas of research in today's digital world. Filtering and conditioning the real time analog signal is one of the major challenges in the area of DSP. This is justified because the need for a real time digital audio processing which allows the modification of audio based on the judgment by hearing while it is being processed. Furthermore the implementation of such filter on a processor should have high speed utilizing minimum number of resources. In this paper an efficient implementation of Reconfigurable Finite Impulse Response (FIR) Filters whose filter coefficients change during runtime using Distributed Arithmetic (DA) architecture is proposed. Here, the multipliers in FIR filter are replaced with multiplier less DA based technique. The proposed architecture provides an efficient less area-time when compared with the existing structure of FIR Filter.

Keywords: Distributed Arithmetic, Finite Impulse Response, Partial Product, Reconfigurable Implementation.

Introduction: Digital Signal processing is one of the most dynamic areas of research in today's digital world. Filtering and conditioning the real time analog signal is one of the major challenges in the area of DSP. This is justified because the need for a real time digital audio processing which allows the modification of audio based on the judgment by hearing while it is being processed. Furthermore the implementation of such filter on a processor should have high speed utilizing minimum number of resources. This has motivated to deal with the development of a reconfigurable FIR filter in this paper. Real time signals in mobile Communications do not have fixed coefficients. To overcomenoise in such systems, filters with variable coefficients known as reconfigurable filters are required. In reconfigurable filters the filter coefficients change dynamically during runtime. They are used in many real time applications such as digital up/down convertors, multi channel filters etc.

The general FIR filter is a multiplier based structure which occupies more area with less speed. So to overcome this Distributed Arithmetic(DA)–based technique is used which has more popularity for their high throughput, cost effective and area-time efficient computing structure. The DA-based computations are performed using lookup-table access followed by shift-accumulation operations of the LUT output.

There are many VLSI architectures available for the implementation of FIR filters. The conventional DA implementation used for the implementation of FIR filter assumes that impulse response coefficients are fixed and this behavior makes to use ROM-based LUTs. The memory required for DA-based FIR filters increases with the filter order. To overcome such problems systolic decomposition techniques are proposed [4] [5]. For reconfigurable DA-based Filter whose filter coefficients change dynamically, we need to use rewritable RAM-based LUT [7] instead of using ROM based LUT. Another way is to store the filter coefficients in analog domain by using serial digital to analog converters resulting in mixed signal architecture [2]. We can also find some works on DA implementation of adaptive filters where the coefficients change in every cycle [3] [6]. In this paper, VLSI architecture of DA based reconfigurable FIR filter is proposed which aims at using minimum resources in the FPGA processing. In the next section we discuss the decomposition for DA implementation of a generic FIR filter. In section III we describe the proposed reconfigurable DA-based FIR filter for removing impulse noise in the speech signal. We provide the synthesis results for the proposed designs in section IV.

Preliminaries: Distributed Arithmetic (DA) technique is bit serial in nature. It is actually a bit-level rearrangement of the multiply and accumulation operation. It is bit serial operation used to compute inner product of a coefficient vector and a variable input vector.

The output of general FIR filter of length N can be obtained as an inner product of the impulse response vector {h(k), for k=0,1,2,....,N-1} and the input

vector{x(n-k), for k=0,1,2,....,N-1}, given by$y = \sum_{k=0}^{N-1} h(k)x(n-k)$ (1)

For simplification, the time index n as$y = \sum_{k=0}^{N-1} h(k)s(k)$ (2)

Where s(k)=x(n-k). Assuming L to be the wordlength, the input sample s(k) may be expressed in two's complement representation.$s(k) = -[s(k)]_0 + \sum_{l=0}^{L-1}[s(k)]_l\, 2^{-l}$ (3)

By substituting this s(k) in the above equation (3),

$y=-\sum_{k=0}^{N-1} h(k)[s(k)]_0 + \sum_{k=0}^{N-1} h(k)\{\sum_{l=1}^{L-1}[s(k)]_l 2^{-l}\}$ (4)

To convert the sum of products form of inner product of (2) into the distributed form, the order of summations over the indices k and l in (4)can be interchanged to have

$y=-\sum_{k=0}^{N-1} h(k)[s(k)]_0 + \sum_{l=1}^{L-1} 2^{-l}\{\sum_{k=0}^{N-1} h(k)[s(k)]_l\}$ (5)

and the inner product given by the equation(5) can be computed as$y = \sum_{l=1}^{L-1} 2^{-l}C_l - C_0$ (6a)

Where $C_l = \sum_{k=0}^{N-1} h(k)\,[s(k)]_l$(6b)

Without loss of generality, for simplicity of discussion, we may assume the signal samples to be unsigned words of size L, we can always obtain unsigned input signal by adding fixed offset when the original input signal is signed. The inner product of the equation (6a) can then be expressed in a simple form,

$y = \sum_{l=1}^{L-1} 2^{-l}C_l$(7)We can use the above equation (7) for implementing the DA-based FIR filter using the LUT containing 2^N possible values of C_l. For large values of N the LUT size becomes large and the access time also becomes large. So the composite number N can be decomposed as *N=PM* (where P and M are two positive integers) one can map the index k into (m+pM) for m=0,1,...M-1 and p=0,1,....P-1. So the above equation becomes $y = \sum_{l=1}^{L-1} 2^{-l}(\sum_{p=0}^{P-1} S_{l,p})$(8a)

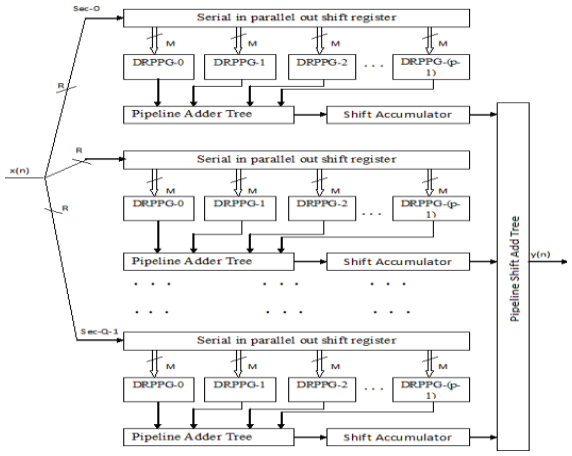
Where $S_{l,p}$ is sum of partial product of M samples and represented as

$S_{l,p} = \sum_{m=0}^{M-1} h(m+pM)[\,S(m+pM)]_l$(8b)

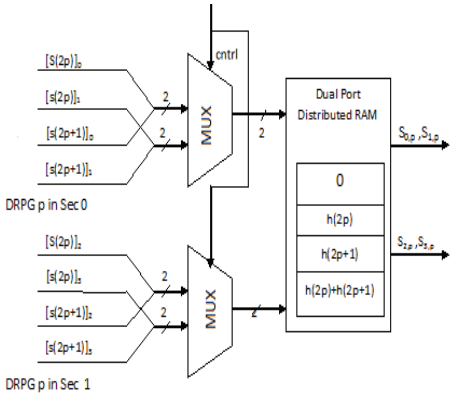
For l=0,1,2,.....,L-1 and p=0,1,2,....,P-1.

Proposed Reconfigurable Distributed Arithmetic Based Fir Filter: For FPGA implementation of any VLSI design the registers are the limited resources. So the LUTs are required to be implemented by Distributed RAM(DRAM) for FPGA implementation. Using DRAM to implement LUT for each bit slice will lead to very high resource consumption. Thus we decompose the partial inner product generator to Q parallel sections, and each section has R time multiplexed operations corresponding to R bit slices. Where L is a composite number given by *L=RQ* (R and Q are two positive integers), the index l can be mapped into (r+qR) for r=0,1,2,....R-1 and q=0,1,2,....Q-1 and the equation becomes$y = \sum_{q=0}^{Q-1} 2^{-Rq}\left[\sum_{r=0}^{R-1} 2^{-r}\left(\sum_{p=0}^{P-1} S_{r+qR,p}\right)\right]$ (9)

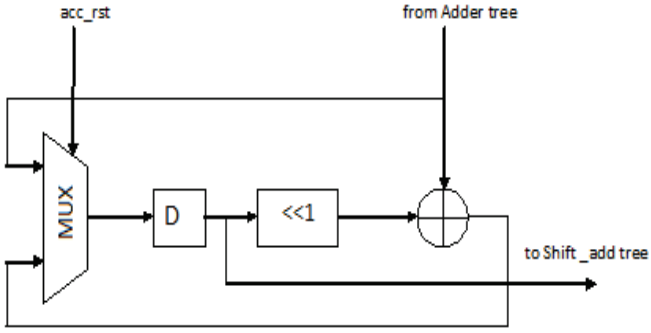
We have R time slots of the same duration as the operating clock period. To implement the equation (9), the proposed structure has Q sections, and each section has *P* DRAM-based Reconfigurable Partial Product Generators (DRPG) and the Pipeline Adder Tree (PAT) to calculate the right most Summation followed by shift accumulator that that performs over R cycles according to the second summation. The proposed structure of the DA-based FIR filter for FPGA implementation is shown in the Fig.1.



(a)



(b)



(c)

Fig.1: (a) Proposed Structure of the DA-Based FIR Filter.
(b) The Structure of Distributed RAM Based Reconfigurable Partial Product Generator (DRPPG).
(c) The Structure of Shift Accumulator.

The size of the LUT is reduced to half since that two DRPPGs from two different sections can share the single DRAM. Finally the Pipeline Shift Add Tree (PSAT) produces the filter output using the output from each section every R cycles. The structure of the DRAM based Reconfigurable partial product generator is shown in Fig. 1(b). Here the DRPPG shown for two sections. Q=2 and for M=4. The Shift Accumulator is (b) n in Fig. 1(c). The accumulated value is reset every R cycles by the control signal acc_rst. To keep the accumulator register ready for the calculation of the next filter output, this acc_rst is used.

Experiments and Results: The input samples {x(n)} at every instant, is given to the Serial In Parallel Out Shift Register (SIPOSR). This SIPOSR decomposes the recent most samples to *P* vectors of length *M* for p=0, 1, 2,...., P-1. The output of the serial in parallel out shift register is fed into *P* Distributed RAM (DRAM) based Reconfigurable Partial Product Generators (DRPPG).In this we considered that the input given is sixteen bit and as we are having two sections the first eight bits are given to the section 0 DRPPGs and the remaining eight bits are given to the section 1 DRPPGs. The simulation results obtained for DRPPG is shown in the Fig.2. Depending on the input signal x(n) $b_{0\text{-}p}$ $b_{1\text{-}p}$ are obtained as $b_{0\text{-}p}$ is {x[0],x[2]} and $b_{1\text{-}p}$ is {x[1],x[3]}. Depending in the impulse response the output is assigned to w.

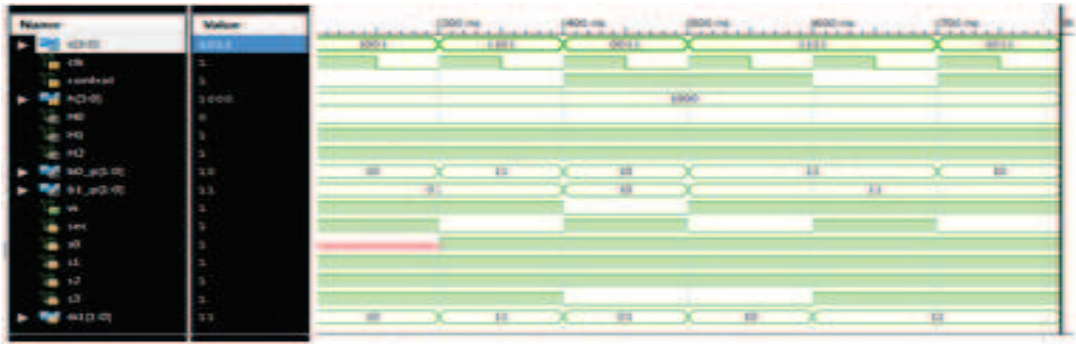


Fig.2: Simulation Results for DRAM Based partial product Generator

The output of the DRPG is given to the Pipeline Adder Tree (PAT). This pipeline adder tree simply adds the output of the DRAM based reconfigurable partial product generators. As there are two sections the two pipeline adder tree's performs the same function but with different inputs.

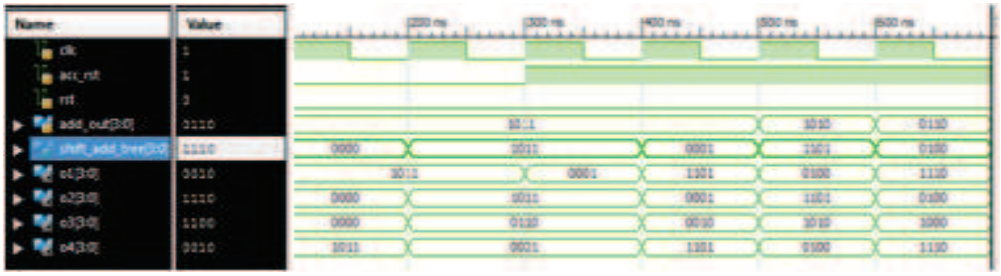


Fig. 3 Simulation Results of The shift Accumulator.

Now the output obtained from the pipeline adder tree is given as the input to the shift accumulator which performs shifting operation and adds that shifted result to the previously stored bits. The simulation results obtained for the shift accumulator is shown in Fig. 3.

Now finally the output obtained from the shift accumulator is given as input to the pipeline shift add tree. This pipeline shift add tree perform shift operation to the output obtained from the each section and adds all the shifted results finally. The simulation results obtained for DRAM based reconfigurable FIR filter are shown in Fig.4.

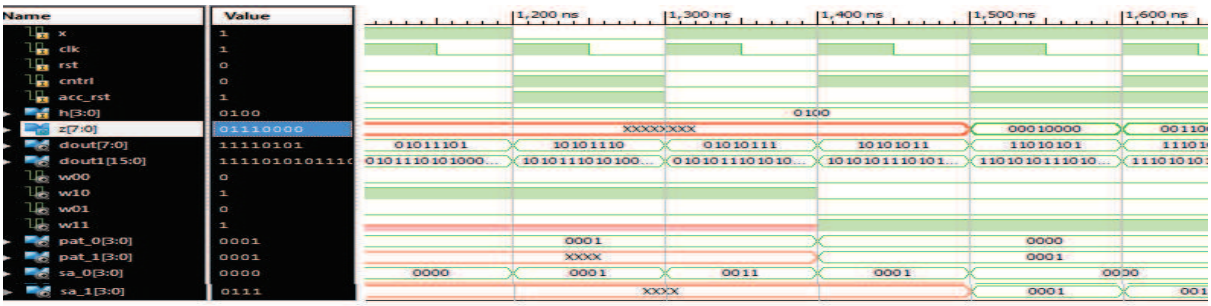


Fig. 4: Simulation Results for Reconfigurable FIR Filter

After HDL synthesis process, the schematic representation of synthesized source file. The detailed view of RTL schematic is shown in Fig. 5. RTL view consists of x as an input speech samples, reset, clock signal, accumulator reset, and the impulse series as inputs to the proposed system. Table I gives a summery on hardware resource consumption of reconfigurable DA-based FIR filter.

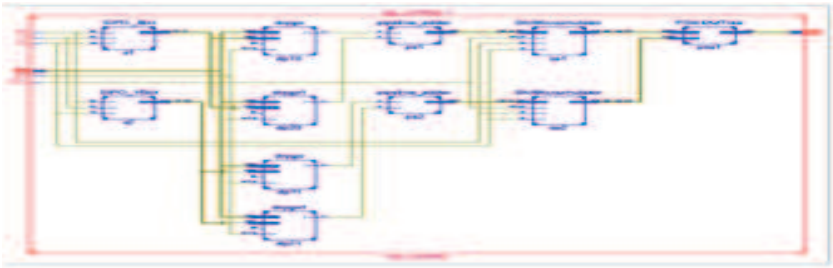


Fig. 5: Detailed View of RTL Schematic

And depending on the utilization of the resources a performance evaluation chart is drawn for both the DA-based reconfigurable FIR filter and the CSA based FIR filter.

TABLE I:Hardware Resource Utilization			
Logic Utilization	Used	Available	Utilization
No. of Slices	32	4656	1%
No. of Slice Flip	24	9312	1%

Flops			
No. of 4 input LUTs	48	9312	1%
No. of bonded IOBs	17	232	7%
No. of BUFGMUXs	1	24	4%

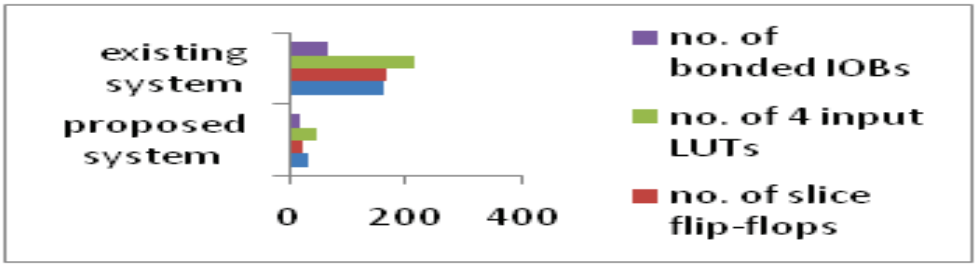


Fig.6: Performance Evaluation of Proposed Architecture

References:

1. Sang Yoon Park, Pramod Kumar Meher, "Efficient FPGA and ASIC realizations of DA-Based Reconfigurable FIR Digital Filter," IEEE Transactions, Aug.2014.
2. 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 Trans. Circuits Syst. I, vol. 55, no. 2, pp. 510–521, Mar. 2008.
3. P. K. Meher and S. Y. Park, "High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic," in Proc. 2011 IEEE/IFIP 19th Int. Conf. VLSI, System-on-Chip, (VLSI-SOC'11), Oct.2011, pp. 428–433.
4. P. K. Meher, "Hardware-efficient systolization of DA-based calculation of finite digital convolution," IEEE Trans. Circuits Syst. II, vol. 53, no. 8, pp. 707–711, Aug. 2006.
5. P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017, Jul. 2008.
6. D. J Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," *IEEE TRANS. Circuits Syst. I*, vol. 52, no. 7, pp. 1327-1337, Jul. 2005
7. M. Kumm, K. Moller, and P. Zipf, "Dynamically reconfigurable FIR filter architectures with fast reconfiguration," in *proc. 2013 8th international Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC),* Jul. 2013.
8. S. A. White, "Applications of the distributed arithmetic to digital signal processing: A tutorial review,:*IEEE ASSP Magazine*, vol.6, no. 3, pp.5-19, Jul. 1989.
9. M. Meyer and D. Agarwal, ""A high sampling rate delayed LMS filter architecture," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol.40, no.11, pp. 727-729, Nov. 1993.

***

K. AmruthaVarshini/M. Tech/ VLSID/ Shri Vishnu Engineering College for Women/ Bhimavaram/amrutha.varshini123@gmail.com
K. Padma Vasavi/ ECE Department/ Shri Vishnu Engineering College for Women/ Bhimavaram/padmavasaviece@svecw.edu.in