IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

DESIGN OF A HIGH PERFORMANCE DA BASED FIR FILTER FOR SDR APPLICATIONS

[1]C.Swarnalatha, [2]M.Kalpana, [3]R.Bindurani, [4]T.Saraswathi, [5]Mr.S. Hari Krishnan(Ph.D)
[1,2,3,4] Students, [5]Assistent Professor in Electronics &Communication Engineering
[1,2,3,4,5]Department of Electronics & Communication Engineering,
[1,2,3,4,5]Sanskrithi School of Engineering, Puttaparthi, Andhra Pradesh, India

Abstract: We have analyzed the register complexity of direct-form and transpose-form structures of FIR filter and explored the possibility of register reuse. We find that direct-form structure involves significantly less registers than the transpose-form structure, and it allows register reuse in parallel implementation. We analyze further the LUT consumption and other resources of DA-based parallel FIR filter structures, and find that the input delay unit, coefficient storage unit and partial product generation unit are also shared besides LUT words when multiple filter outputs are computed in parallel. Based on these finding, we propose a design approach, and used that to derive a DA-based architecture for reconfigurable block-based FIR filter, which is scalable for larger block-sizes and higher filter-lengths. Interestingly, the number of registers of the proposed structure does not increase proportionately with the block-size. This is a major advantage for area-delay and energy efficient high-throughput implementation of reconfigurable FIR filters of higher block-sizes. Theoretical comparison shows that the proposed structure for block-size 8 and filter-length 64 involves 60% more flip-flops, 6.2 times more adders, 3.5 times more AND-OR gates, and offers 8 times higher throughput. ASIC synthesis result shows that the proposed structure for block-size 8 and filter-length 64 involves 1.8 times less area-delay product (ADP) and energy per sample (EPS) than the existing design, and it can support 8 times higher throughput. At common throughput, the proposed structure for block sizes 4 and 8, respectively, consumes 38% and 50% less power than the exiting structure on average for different supply voltages.

Index Terms - Software : MATLAB, Modelsim , Xilinx ISE

Hardware : Xilinx or Altera(Intel) FPGA

I. Introduction

Software defined radio (SDR) technology enables for digital implementation of wide band trans-receivers of multi-standard wireless communications. In SDR, a channelizer is used to extract narrowband channels from the wide band signal. Channelization is usually performed by a bank of finite impulse response (FIR) filters. The channelizer is required to operate at highest sampling rate and requires FIR filters of large order to extract narrowband channels with stringent adjacent channel attenuation specification. The channelizer is the most computation intensive part of SDR. On the other hand, channelizer needs to be implemented in a reconfigurable hardware to support multi-standard wireless communication.

Reconfigurability, high-sampling rate and low-power are the three mutually conflicting design features of channelizer to make it suitable for next generation wireless application. Large order FIR filters where the coefficients having different sets of filter coefficients is highly resource consuming. Besides, there is no redundant computation in FIR filter algorithm. The derivation of hardware designs with less area-delay-product (ADP) for the implementing filter bank for SDR channelizer is a challenging task. Several designs have been suggested in the last decade to improve its efficiency in reconfigurable architectures. We briefly discuss here the key developments in this area. A programmable multiply-accumulate based processor is proposed in for FIR filtering. In a programmable canonical signed digit (CSD) based architecture was proposed using Booth encoding to generate partial products and Wallace tree adder for addition of partial products. Chen et al have proposed a CSD based reconfigurable FIR filter, where the non-zero CSD values are modified to reduce the precision of filter coefficients without a significant impact on filter behavior. But, the reconfiguration overhead is significantly large and does not provide an area-delay efficient structure. A few multiplier-based designs are also proposed for the realization of a reconfigurable FIR filter. Few multiplier-less design also have been proposed using common sub-expression elimination (CSE) method. A reconfigurable multiplier design was proposed using CSE. Mahesh et al. have proposed a constant shift method (CSM) and a programmable shift method (PSM) to construct reconfigurable architecture for large order FIR filters. They have used binary CSE (BCSE) method to reduce complexity of the FIR filter structure. The reconfigurable architectures of are efficient than the previously proposed structures.

II THEORY:-

Proposed design:

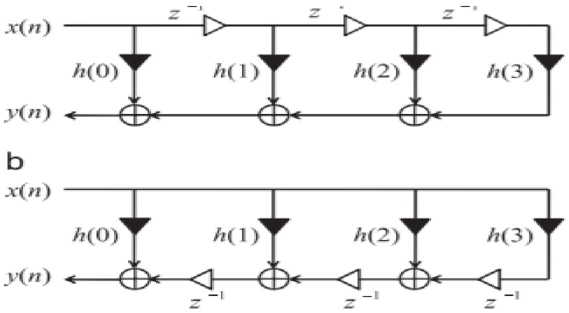
The system function of an FIR filter is given by

$$H(z)= \sum_{i=0}^{N-1} h(i)z^{-i}$$

Computations of Eq. (1) can be performed using a direct-form or transposed-form structure as shown in Fig. 1 for N¼4. The two structures re

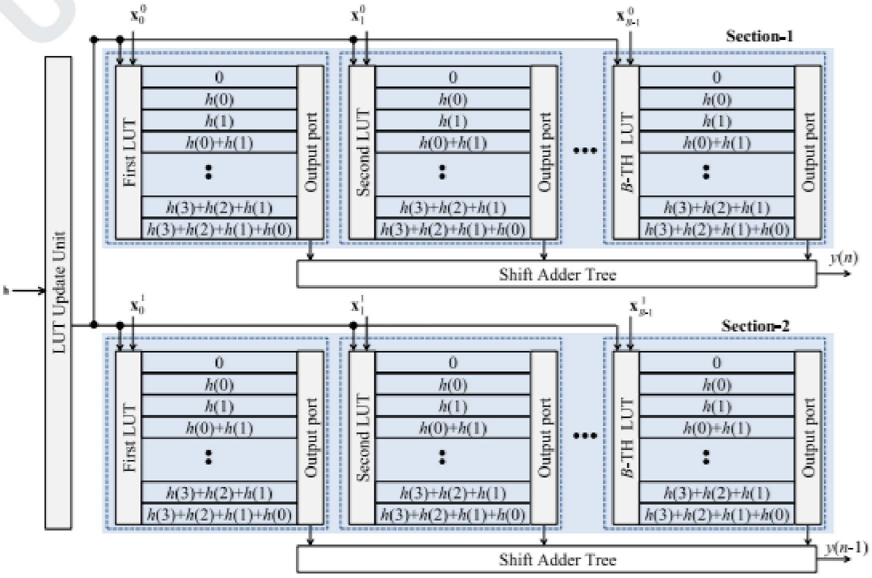
quire the same number of arithmetic components (multiplier and adders) and delay elements (z 1 corresponding to registers) except for their locations in the data-path. Due to the growth of bit-width after arithmetic operations, bit-width of input and intermediate signals are different. Therefore, the number of register bits required by the direct-form and transpose-form structures are different (when multiplier results are not truncated), although both have the same number of delay elements in the data-path. Assuming the bit-width of the input signal and the coefficients to be 8, the bit-width of the intermediate/output signal after truncation could be 16 or 20. Accordingly, we have estimated the register-complexity of the direct and transpose-form structures for filter length (N=128 and N=256), and the estimated values are listed in Table 1. We find that direct-form structure requires less register-complexity than the corresponding transpose-form structure. Interestingly, the register complexity of the direct-form structure is independent of the word-length of the intermediate signals, since all the delay elements of this structure are placed on the input path only. This is a very useful feature which could be further exploited for register sharing of parallel FIR filter implementation.

Exploring register sharing: To explore the register sharing, let us consider the input dataflow of the direct-form structure for the computation of four successive outputs {y(n), y(n-1), y(n-2), y(n-3)}as shown in Fig. 2 for filter length N=4. Samples required to compute four successive outputs are shown in a 4x4 matrix, where each row of samples involved to compute a filter output is shown at its right. Each down-arrow shows the source (register) of the sample. Four samples {x(n), x(n-1), x(n-2), x(n-3)} are required to compute the output y(n), where x(n) is the current input sample and others are the immediate past samples. The three past samples are obtained from the register unit comprised of (N-1) =3 registers.



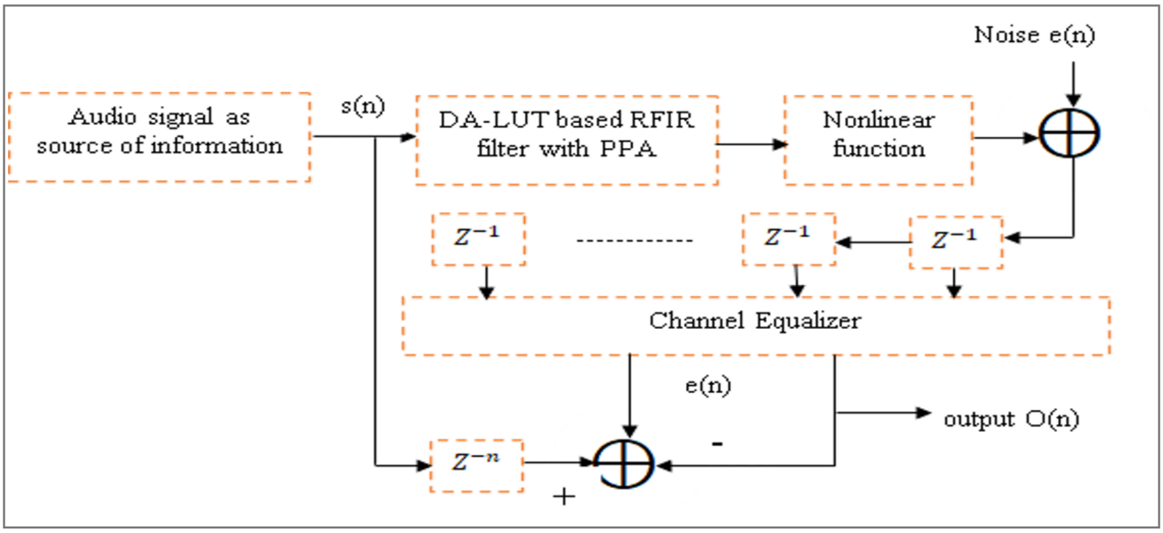
Therefore, the direct-form structure uses (N-1) memory words to compute each filter output

Exploring LUT-compaction in DA-based reconfigurable FIR filter



A full-parallel DA-based design for reconfigurable block FIR (RBFIR) filter is shown in Fig. 3 for N=4 and L=2. It consists of two L=2 sections where each section comprises of B identical LUTs and one shift adder tree (SAT). Apart from this the DA based reconfigurable FIR structure also involves an adder unit to compute LUT values corresponding to the coefficient vector. The first section receives bit-slices of the input-vector x0 and the second section receives the bit-slices of x1. B bit-slices of the input-vector are fed to the address line of B LUTs of a particular section and read B partial filter outputs in parallel. The partial filter outputs of each section are shift-added in the SAT to obtain the filter output. In general, the full-parallel conventional DA-based RBFIR design involves BL times the LUT words and adders of the conventional DA-based design, and gives BL times higher throughput rate. As shown in Fig. 3, all the (BL) LUTs of the bit-parallel structure store identical values. These identical values could be shared instead of being stored in different LUTs. All the words in the LUT need to be available in the same cycle for the sharing of the LUT words. Conventional RAM-based LUTs are not suitable for LUT sharing since in any given cycle they allow access to only one (or a few in case of multi-ported RAM) of the stored LUT values. A register multiplexer (REG-MUX) based LUT could be used instead for LUT sharing. A REG-MUX LUT takes only one cycle for updating the whole LUT content, where 2n cycles are required by conventional RAM-based LUT of 2n words. A full-parallel DA-based RBFIR design using REG-MUX LUT is shown in Fig. 4. It consists of a ð2N 1Þ-word register array which store the LUT words, L MUX-arrays, and L SATs, where L is the block size. Here we assume that zero is not required to be stored in the register array and that can be generated by a reset circuit. The REGMUX LUT is implemented using a ð2N 1Þ-word register array one MUX array. Each MUX array receives one N-point input-vector such that each MUX of the MUX-array receives one bit-slice of the input vector at its select line to select one out of(2n-1) LUT values available at input lines.

III.BLOCK DIAGRAM:



We have estimated the LUT complexity of full-parallel DA based RBFIR filter using shared LUT and RAM LUT for different word-lengths, filter-lengths and block-sizes to find the saving of LUT words offered by the shared LUT design. From the estimated values shown in Table 2, we find that, the LUT sharing method provides nearly 94% saving of LUT words over the convectional method for different filter lengths, block-sizes and word-lengths. The saving of LUT words is more for larger filter-lengths, block sizes and longer word-lengths. This offers some saving in LUT resource when REG-MUX combination is used instead of RAM LUT. The register-complexity of the direct-form structure of the block FIR filter is independent of the input block-size and it does not change by increasing the parallel stages of the DA-design. Similarly, DA-based RBFIR filter design using REG-MUX LUT offers higher saving of LUT words in case of a bit-parallel DA-design with higher block-sizes. Besides, the adder unit computing LUT values is shared for parallel computation of filter output. Consequently, the area complexity of the DA-based design does not increase proportionately with increase in the number of parallel stages. To take advantage of this feature we propose here a design strategy to derive an area-delay-energy efficient DA-based structure for reconfigurable FIR filter.

Performance comparison: Hardware and time complexities of the proposed structure and structures of are listed in Table 5 for comparison. The structure of is the best available structure among all the existing structures. The proposed structure requires the same number of ROM words, (BB'(L-1)) more flip-flops, nearly L) times more adders, and nearly (5LBB'/2) more AND-OR gates than those of the structure of, and it offers L times more throughput. Compared with PSM-based structure of, the proposed structure requires 2 times less number of ROM words, nearly [30(B+B'/LB) less flip-flops, nearly (BL/12) times more adders and nearly times less AND-OR gates, and offers more than L times higher throughput due to smaller cycle period. It is interesting to find from Table 5 that in the proposed structure, the number of flip-flops used in delay-line is independent of the block-size. The total number of flip-flops and AND-OR gates of the proposed structure do not increase proportionately with L. However, the flip-flop (register) saving offers reduction of power consumption, which does not increase proportionately with the block size.

IV.CONCLUSION:-

In this paper, we have analyzed the register complexity of the direct-form and transpose-form structures of an FIR filter, and explored the scope of register reuse. We find that the direct-form structure involves significantly less registers than the transpose form structure, and offers extensive register sharing in parallel implementation. We analyze further the LUT consumption and other resources of DA-based parallel FIR filter structures, and find that the input delay unit, coefficient storage unit and partial product generation unit are also shared besides LUT words when multiple filter outputs are computed in parallel. We have made use of these observations to derive a DA-based architecture for reconfigurable block-based FIR filter, which is scalable for higher block-sizes, and larger filter-lengths. Interestingly, the number of registers of the proposed structure does not increase proportionately with the block-size. Theoretical comparison shows that the proposed structure for L=8 and N=64 involves 60% more flipflops, 6.2 times more adders, 3.5 times more AND-OR gates than the best available similar design, but offers 8 times higher throughput. ASIC synthesis results show that the proposed structure for block-size 8 involves 1.8 times less ADP and EPS than the existing design, and it can support 8 times higher sampling rate. At common throughput, the proposed structure for block sizes 4 and 8, respectively, consumes 38% and 50% less power than the structure of on average for different supply voltages.

V.REFERENCES:

T. Hentschel, G. Fettweis, Software radio receivers, CDMA Techniques for Third Generation Mobile Systems, Kluwer Academic, Dordrecht, The Netherlands (1999), p. 257–283.

J. Mitola, Object-oriented approaches to wireless systems engineering, Software Radio Architecture, Wiley, New York, 2000.

E. Buracchini, The software radio concept, IEEE Commun. Mag. 38 (September) (2000) 138–143.

R.I. Hartley, Subexpression sharing in filters using canonic signed digit multipliers, IEEE Trans. Circuits Syst. II 43 (October (10)) (1996) 677–688.

R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, D. Durackova, A new algorithm for elimination of common subexpressions, IEEE Trans. Comput. Aided Design Integr. Circuits Syst. 18 (January (1)) (1999) 58–68.

T. Solla, R. Makela, M. Liljeroos, O. Vainio, Application-specific filter processor for flexible receivers, in: Proceedings of 19th NORCHIP Conference on Kista, Sweden, Novemver 2001, pp. 53–58.

T. Solla, O. Vainio, Comparison of programmable FIR filter architectures for low power, in: Proceedings of 28th European Solid-State Circuits Conference on Fire