A Flexible Design for Optimization of Hardware Architecture in Distributed Arithmetic based FIR Filters

Fazel Sharifi, Saba Amanollahi, Mohammad Amin Taherkhani and Omid Hashemipour
Faculty of Electrical and Computer Engineering, Shahid Beheshti University, G.C., Tehran, Iran
{f_sharifi, s_amanollahi, m_taherkhani, hashemipour}@sbu.ac.ir

Abstract
FIR filters are used in many performance/power critical applications such as mobile communication devices, analogue to digital converters and digital signal processing applications. Design of appropriate FIR filters usually causes the order of filter to be increased. Synthesis and tape-out of high-order FIR filters with reasonable delay, area and power has become an important challenge for hardware designers. In many cases the complexity of high-order filters causes the constraints of the total design could not be satisfied. In this paper, efficient hardware architecture is proposed for distributed arithmetic (DA) based FIR filters. The architecture is based on optimized combination of Look-up Tables (LUTs) and compressors. The optimized system level solution is obtained from a set of dynamic-programming optimization algorithms. The experiments show the proposed design reduced the delay cost between 16%-62.5% in comparison of previous optimized structures for DA-based architectures.

1. Introduction
Nowadays, FIR filters, regarding to their superior properties such as stability and high reliability in digital signal processing, have had many important and widespread applications. This kind of digital filters are applied to an extensive form in many areas such as image processing, radio communication and high technology devices. One of the important applications of FIR filters is in analog to digital converters [1][2]. Also FIR filters are applied in read channel of disc drives known as PRML [3] in addition to wideband receivers in wireless communication devices [4].

Generally in an N-order FIR filter with \( \text{coef} [i] \forall i \in [0, N-1] \), the output \( y[t] \) is calculated according to the current input \( x[t] \) and previous inputs \( x[t-i] \forall i \in [1, N-1] \) by equation (1):

\[
y[t] = \sum_{i=0}^{N-1} \text{coef}[i]x[t-i]
\]

As it is shown in equation (1), in each step for calculation of \( y[t] \) as the output of N-order filter, \( N \) additions and \( N \) multiplications are required.

Although the FIR filters are more stable compared to IIR filters, their circuits are very complicated due to necessity of designing high order FIR filters in real application. By increasing in order of filter, the circuit is become more complicated and this is one of the most important challenges in front of designing these types of filters. This complexity sometimes causes desired design does not meet area, timing and power constraints.

In this paper a compound architecture for FIR filters is proposed. Considered modules in the proposed architecture are optimized by efficient algorithms and final architecture will be extracted.

The rest of this paper includes following sections. In section 2, architectures of FIR filter with related works are reviewed. In section 3 the proposed architecture is introduced. Optimization algorithms for construction of efficient hardware architecture are introduced in section 4. Experimental results are shown in 5 and finally section 6 concludes the paper and future works are considered.

2. A review of related architectures
For implementation of FIR filter structures, hardware architectures based on multiply and add (MAC) and distributed arithmetic (DA) are known as the main classes of FIR filter architectures.

In MAC-based architectures, computation of the desired output is done directly and it is based on multiplication and addition. To improve the performance of MAC-based architectures, some researches focused on design of filters based on Residue Number system (RNS) [5]-[7]. In these designs, processing overhead and hardware cost of binary to RNS conversion should be considered.

Complexity of classic multiplication and addition causes a lot of researches tend to change in filter architecture and be done based on DA, so that without using multipliers, the idea of pre-computing and storing the required values has been exploited.

The classic method of distributed computing works based on changing the form of required computations in equation (1) and rewriting it in new computation forms. For this purpose it assumes, the states \( x[k] \forall k \in [t-N+1, t] \) in binary standard format (2’s complement) and has been
scaled \( \{1_{\text{LUT}}(k) \leq 1 \} \) and it can be shown by the following equation:
\[
x[k] = -x_{B+1}[k] + \sum_{j=0}^{R-1} v_j[k] 2^{-j}
\]  
(2)

Substituting equation (2) in equation (1) results to:
\[
y[i] = \sum_{j=0}^{k} \text{coeff}[j] \cdot x[i-1] + \sum_{j=0}^{R-1} 2^j \sum_{j=0}^{k} \text{coeff}[j] x[i-1]
\]  
(3)

A look up table could be used with \( N \)-bit input address to preserve the value of \( \sum_{j=0}^{k} \text{coeff}[j] x[i-1] \). In this condition, size of LUT size (without applying optimization) would be \( 2^N \times (C + \log_2(N)) \), where \( N \) is the order of filter \( C \) is the length of coefficients in binary form. As the equation 3 shows, a combination of sum of coefficients can be calculated in advance and be stored in look up tables.

The main problem of look-up tables is the complexity of table size which grows exponentially by increasing of filter order. A method have been proposed in [9] to reduce (and ultimately eliminate) the size of required look-up table. But in the proposed LUT-less architecture, the delay of adders and multiplexers (MUX) is not considered and therefore the solution is not efficient for performance-critical applications. In [10] another DA-based FIR filter has been presented that is suitable for FPGA platforms with 4-input look-up tables.

3. Proposed architecture

As it has been shown in previous section, distributed arithmetic based architectures face off with exponential complexity problem for the size of look up tables. In this section an efficient architecture is presented in order to increase performance of computation part of digital filters. As it is illustrated in Figure 1, main components of proposed architecture are formed based on a compound model with two main layers including look up tables and compressors. As it can be seen in Figure 1, \( N \) bits are used in first layer of shift register and \( k = \sum_{j=0}^{k} k_j \) bits out of these \( N \) bits are assigned to \( M \) look up tables, with \( k_j \) ... \( k_m \) inputs. In the next section, an optimization algorithm for finding efficient structure for look up tables set with \( k \) bits input address is introduced.

Remaining bits (\( N-k \)) are used as selectors in 2-1 multiplexers (for every bit) and totally \( C \times (N-k) \) multiplexers are required. If the selector of \( i \)-th multiplexer becomes 1 the related coefficient will be added to compressor part.

\( M \) outputs of look up tables with \( N-k \) outputs of multiplexers are used as inputs of \( N-k+m:2 \) compressor. The functionality of the compressor is shown in Figure 2. Extraction of an efficient \( N-k+m:2 \) compressor is described in the next section. After compression, a CLA adder is used for final summation.

4. Architecture extraction algorithms

As was noted in the previous section, the proposed approach provides possibility of choosing suitable hardware architecture based on compound structure sets in a flexible way. For finding efficient architecture the following parameters should be determined precisely:

- Optimized structure of LUT set based on input parameter (\( k \)): In this section \( k \) and \( m \) values are chosen somehow LUT size with \( k \) bit input address is optimized. The proposed algorithm is presented in section 4.1.
- Optimized compressor structure based on input bits set (\( h \)): In this section optimized compressor structure is extracted based on small-optimized compressors. The proposed algorithm is presented in section 4.2.
- Final optimized architecture: In this section, according to parameter values for LUTs and compressors, number of input addresses for the LUT layer and number of selector bits for compressor layer are chosen in a way which the final architecture is...
Optimization could be done based on following parameters:

- Gate latency: in all optimizations process delay of XOR gate is considered as the delay unit. This parameter is shown by $(2\Delta G)$.
- Power consumption: optimization for power consumption is based on minimum number of resources and could be determined by power consumption unit of XOR gate.
- Power-delay product (PDP): the proposed algorithms could be extended to optimize PDP parameter based on previous parameters.

4.1. Optimization of LUT Layer

In this section, technique has been proposed for partitioning of LUTs. In this work the LUT layer with $k$ bits for input addresses and $m$ outputs is partitioned into $m$ basic LUTs based on computation of delay/power and power delay product (PDP) parameters in gate level. LUT models based on Decoder-Memory are exploited for description of architecture in the LUT layer [14].

The structure of a basic LUT with $k_i$ input address (for preserving the summation of $k_i$ coefficients) contains a $k_i$ to $2^{k_i}$ decoder and a $2^{k_i}$ words memory with $C + \lceil \log_2 k_i \rceil$ length. Suppose, delay of this LUT is notated as $d_{lut}^{k_i}$, and its power consumption and power delay product (PDP) are respectively notated as $p_{lut}^{k_i}$ and $p_{d}^{p_{d}}$. Therefore for a LUT layer with the $k$ inputs and $m$ outputs, the structure with optimized delay ($D(LUT_{k,m})$) or optimized power ($P(LUT_{k,m})$) or optimized PDP ($PD(LUT_{k,m})$) could be obtained from the dynamic programming approach with polynomial complexity.

4.2. Optimization of Compressor Layer

In this section a method is proposed for auto construction of the $h:2$ optimized compressor based on delay, power and PDP parameters. Creating large input compressors are carried out by using of optimized conventional and unconventional compressors [11]-[13].

Therefore, for creating an optimized large input compressor ($h:2$), set of $F$ basic compressors $Comp_1...Comp_F$ with compression levels $\left( I_{Comp_{k,2}} : O_{Comp_{k,2}} \right)$ $\forall k \in \left\{ 1...F \right\}$ are used. Unconventional compressors have some carry in and carry out bits. However, these carry bits are created in such a way in compressor which there are not carry propagation. Suppose delay, power and PDP of basic compressor $k$ are $d_{Comp^k}$, $P_{Comp^k}$, $p_{d}^{p_{d}}_{Comp^k}$ respectively.

4.3. Extraction of final solution

In this step, based on optimization results of the LUT layer and the compressor layer, the final architecture of the proposed DA unit is extracted. In other word the value for $m$ and $k$ is determined by using the Algorithm 3.

Based on the main criteria for the designer, the algorithm could present separately the optimized solution for delay, power or PDP parameters. As shown in Algorithm 3, the cost function could be any arbitrary parameters Delay, Power or PDP returned from OptimizedLUT and OptimizedComp Algorithms.

\begin{algorithm}
\caption{Proposed Algorithm for Optimization of LUT layer} \label{algorithm1}
\begin{algorithmic}
1. $D(LUT_{1}) \rightarrow d_{lut}^{[j]} \quad \forall i \in \left\{ 1...k \right\}$
2. $P(LUT_{1}) \rightarrow p_{lut}^{[j]} \quad \forall i \in \left\{ 1...k \right\}$
3. $PD(LUT_{1}) \rightarrow p_{d}^{p_{d}}_{lut}^{[j]} \quad \forall i \in \left\{ 1...k \right\}$
4. \textbf{for} $(i=2; i < k; i++)$
5. \quad \textbf{for} $(j=2; j < m; j++)$
6. $D(LUT_{i}) \rightarrow \max\{d_{lut}^{[j]}, D(LUT_{i-1})\}$
7. $P(LUT_{i}) \rightarrow \min\{P_{lut}^{[j]}, P(\text{Optimized}LUT_{i-1})\}$
8. $PD(LUT_{i}) \rightarrow \min\{PD(LUT_{i-1}), PD(\text{Optimized}LUT_{i-1})\}$
9. \textbf{end for}
10. \textbf{end for}
11. return $D(LUT_{x_{\text{out}}})$, $P(LUT_{x_{\text{out}}})$, $PD(LUT_{x_{\text{out}}})$
\end{algorithmic}
\end{algorithm}

Figure 3. A configuration of $h:2$ compressor exploiting the basic compressor $\text{Comp}_h$.
Algorithms 2. Optimization of Compressor Layer

\textbf{OptimizeComp (CompLevel: h):}

1. \textit{F}→ Number of basic compressors
2. \(D_i(0)\leftarrow 0, P_i(0)\leftarrow 0, PDP_i(0)\leftarrow 0;\) \(\forall i \in [1..h], j < i\)
3. \(D_i(k)\leftarrow 0, P_i(k)\leftarrow 0, PDP_i(k)\leftarrow 0;\) \(\forall k \in [1..f], i \in (1,2), j < i\)
4. \textbf{while} \((k < f)\)
5. \textbf{for} \((i=3; i<h; i++)\)
6. \textbf{for} \((j=1; j<i; j++)\)
7. \(U(j) = (l_{comp}\cup O_{comp}(j))\)
8. \(D_i(k) \leftarrow D_{comp}(k)\)
9. \(P_i(k) \leftarrow P_{comp}(k)\)
10. \(PDP_i(k) \leftarrow PDP_{comp}(k)\)
11. \textbf{else}
12. \(D_{min} \leftarrow \min\{\max(D_{i-Comp}(k), D_{comp}(k)) + D_{i-Comp}(k)+c(k)\}\)
13. \(P_{min} \leftarrow \min\{P_{i-Comp}(k) + P_{comp}(k) + P_{comp}(k)\}\)
14. \(PDP_{min} \leftarrow \min\{D_{comp}(k), P_{comp}(k), P_{comp}(k)\}\)
15. \(D_i(k) \leftarrow \min(D_i(k-1), D_{min})\)
16. \(P_i(k) \leftarrow \min(P_i(k-1), P_{min})\)
17. \(PDP_i(k) \leftarrow \min(D_i(k-1), P_i(k-1), D_{min}, TP, P_{min}, TD)\)
18. \textbf{end for}\n19. \textbf{end for}\n20. \(k = k + 1\)
21. \textbf{end while}\n22. \textbf{return} \(D_i(F), P_i(F), PDP_i(F)\)

\textbf{Algorithm 3. Extraction of optimized architecture}

5. Implementation and Experiments

Implementation has performed in two sections. In first section, hardware description of all basic components has been implemented. Verilog implementation of basic optimized compressors includes 3:2, 4:2, 5:2, 6:2, 7:2 and 9:2 compressors [11]-[13]. The other regular compressors such as 7:3 or 15:4 could be constructed from these basic components. The implementation of basic LUTs was based on memory model in CACTI 5.1[14]. In the second section of this phase, the optimized architecture algorithms have been implemented in 9 source and header files to present the optimized structure of DA unit.

The real coefficients have been extracted from Filter Design and Analysis Tool (FDATool) [15]. The coefficients are produced for implementation of the sample filters listed in Table 1. As shown in the table based on the supplied criteria, the order of filter is determined from 8 to 143. According to our requirements in application of designing digital part of ADCs, the frequency of sampling \((F_s)\) and length of inputs \((B)\) have been set to a 40MHz and 3 bits respectively. In addition, in all design C (length of coefficients) is set to 16 bits.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Filter Order} & \textbf{F_s (MHz)} & \textbf{F_{pass} (MHz)} & \textbf{A_{pass} (dB)} \\
\hline
8 & 40 & 1.2/3.8 & 3/20 \\
18 & 40 & 1.2/3.8 & 3/40 \\
31 & 40 & 1.6/3.0 & 3/40 \\
72 & 40 & 2.2/2.8 & 3/40 \\
108 & 40 & 2.2/2.8 & 3/60 \\
143 & 40 & 2.2/2.8 & 3/80 \\
\hline
\end{tabular}
\caption{Specification of analyzed FIR filters}
\end{table}

The estimated delay based on (Gate delay-AG) for Distributed Arithmetic unit of LUT, LUT-less [9] and proposed architectures is shown in Figure 4. LUT-less architecture in [9] is implemented by full compressors instead of regular adders. As shown in the figure delay of the proposed architecture is 16% (for 8-order filter) to 62.5% (for 143-order filter) less delay in comparison of LUT-less architecture.

![Figure 4 Estimated Delay of FIR filters for DA-based architectures](image)

6. Conclusion and Future Works

According to design constraints in high-order FIR filters, in this work the following contributions were presented:

- Compound hardware architecture was proposed for distributed arithmetic based FIR filters. The architecture exploits the benefits of pre-served summation in optimized LUTs and improves the speed of addition by using high efficient compressors.
- A dynamic programming algorithm was proposed with polynomial complexity to find the optimized structure of compressors.
- A dynamic programming algorithm was proposed to find the best solution for LUT partitioning.
- The final optimized architecture could be extracted from third-proposed algorithm.

For the future works, we will attempt to extend the tool which is capable for automatic generation of HDL code for the optimized extracted architecture.
Acknowledgment

The authors would like to acknowledge members of Micro Electronic Lab in Faculty of Electrical and Computer Engineering in Shahid Beheshti University (SBU) specially, Professor K. Navi, S. Z. Reyhani and Adel Hosseiny.

7. References

[1] R. M. R. Koppula, S. Balagopal, V. Saxena, "Efficient design and synthesis of decimation filters for wideband delta-sigma ADCs," *In Proc. IEEE SOCC*, pp. 380-385, 2011.

[2] H. V. Sorensen P. M. Aziz and J. V. D. Spiegel, "An overview of sigma-delta converters," *IEEE Signal Processing Magazine*, 61–84, January 1996.

[3] M. Singh et al., "An Adaptively Pipelined Mixed Synchronous-Asynchronous Digital FIR Filter Chip Operating at 1.3 Gigahertz," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 7, pp. 1043-1056, 2010.

[4] R. Mahesh, A. P. Vinod, "New Reconfigurable Architectures for Implementing FIR Filters With Low Complexity," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 29, Issue 2, pp. 275-288, Feb. 2010

[5] P. Patronik, K. Berezowski; S. J. Piestrak; J. Bienrat; A. Shrivastava, "Fast and energy-efficient fir filters constant coefficient using residue number systems," *In Proc. Int. Symp on Low Power Electronics and Design (ISLPED’11)*, pp. 385-390, 2011.

[6] S. Pontarelli, G. C. Cardarilli, Marco Re, Adelio Salsano, "Optimized Implementation of RNS FIR Filters Based on FPGAs," *Journal of Signal Processing Systems*, Vol. 67, Num. 3, 2012.

[7] R. Conway, J. Nelson, "Improved RNS FIR Filter Architecture," *IEEE Transaction on Circuits and Systems II: Express Briefs*, Vo. 51, pp. 26-28, 2004.

[8] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoustics., Speech, Signal Processing*, Vol. 22, Issue. 6, pp. 456–462, Dec. 1974.

[9] H. Yoo and D. V. Anderson, "Hardware-Efficient Distributed Arithmetic Architecture for High-Order Digital Filters", *In Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, pp. 125-128, Mar. 2005.

[10] P. Longa and A. Miri, "Area-efficient FIR filter design on FPGAs using distributed arithmetic," *In Proc. IEEE Int. Symp. Signal Processing and Information Technology*, pp. 248-252, 2006.

[11] M. Rouholamini, O. Kavehie, A. P. Mirbaha, S. J. Jasbi, K. Navi, "A New Design for 7:2 Compressors," *In Proc. IEEE/ACS International Conference on Computer Systems and Applications, AICCSA ’07*, pp.474-478, May 2007.

[12] C. H. Chang; J. Gu; M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Transactions on Circuits and Systems*, vol.51, no.10, pp. 1985-1997, Oct. 2004.

[13] S. Veeramachaneni; K. M Krishna; L. Avinash; S. R. Puppala; M. B. Srinivas; , "Novel Architectures for High-Speed and Low-Power 3-2, 4-2 and 5-2 Compressors," *In Proc. 6th Int. Conf. on Embedded Systems*, pp.324-329, Jan. 2007.

[14] S. Thoziyoor , N. Muralimanohar , J. H. Ahn and N. P. Jouppi *CACTI 5.1*, 2008.

[15] Filter Design and Analysis Tool (FDA Tool), MathWorks Inc., [http://www.mathworks.com](http://www.mathworks.com).