International Journal on Applications in Information and Communication Engineering
Volume 1: Issue 5: May 2015, pp 23-25. www.aetsjournal.com ISSN (Online) : 2394-6237

Efficient FIR Filter Design for Sound Amplification

Edwin Meshach

Abstract— In this paper, a constant coefficient finite impulse response (FIR) filter design has been proposed for hearing aid application. The major change in the proposed architecture is the use of 4:2 compressors instead of using adders. The 17 order filter for hearing aid has been realized at gate level using Verilog HDL. The architectures have been implemented in UMC 90nm technology by the use of Cadence RTL compiler. Synthesis results of the proposed architecture show an improvement of 39.4% and 11.34% in speed and area respectively as compared to the recently published architecture. The proposed architecture provides sig-nificant gain of 46.3% in area-delay product (ADP) and 23.7% in power-delay product (PDP). Finally, the functionality of the architecture has been verified by Altera DSP Builder tool.

Keywords— FIR filter; Hearing aid; CSD; HCS CSD; 4:2 Compressor

I. INTRODUCTION

Hearing aids are sound amplifying devices that help an individual suffering from hearing loss to hear well. A major challenge in hearing aids is poor speech recognition capability in a noisy environment [1]. In this paper, the focus has been given on optimizing the filter, which forms an important block in the hearing aid system. The hearing aid device to be planted in the patient′ s ear must be small in size, fast in operation and should have a long battery life. This makes it important to design a fast area efficient filter with low power dissipation. FIR filters are inherently stable along with the advantage of a linear phase response. This makes FIR filters a better choice over infinite impulse response (IIR) filters [2].

The basic functionality for the hearing aid system to suppress the noise signal from the desired sound signal was proposed in [3]. This algorithm required the use of 17 order FIR filter block for avoiding the effects of noise. Various tech-niques been proposed in literature to reduce the computation effort and design an efficient filter for hearing aid application. One such technique has been presented in [4] to optimize the multiplier block, used in each tap of the filter. A power efficient architecture has been achieved by the use of canonical signed digit (CSD) representation. The CSD representation is defined as a redundant number system that represents numbers with no adjacent non-zero digits [4]. Every number has a unique CSD representation. Recoding of a binary number in CSD format results in fewer non-zero bits in its representation. As a result, the number of additions to be performed on the partial products in the multiplier block are reduced greatly by replacing the conventional 2′ s complement binary notation by the CSD notation. Another architecture has been published in [5] based on the Computation sharing multiplier (CSHM) approach. In this method, small bit sequences or patterns are chosen from the bit patterns of the coefficients to be multiplied with the input sequence. The product of the input with this specific set of bit patterns is computed instead of carrying out multiplication of the input with the entire filter coeffi-cient. These intermediate partial products are then combined using multiplexers and adders to generate the final result [5]. The Hierarchical Computation Sharing CSD (HCS CSD) architecture proposed in [6], which combined the features of CSD notation and CSHM, has been taken as the reference architecture. The number of additions required in the pre -computation block has been further reduced as compared to CSHM, due to the use of CSD notation. In this technique, the coefficients have been represented in CSD format. Further, the coefficients have been generated purely by the process of shift-and-add, thus avoiding the use of any multiplier blocks in each of the filter taps.

II. PROPOSED ARCHITECURE

At first, the specifications of the filter for the hearing aid application were finalized. The sampling frequency was 22.05 kHz as mentioned in [3]. Also, the cutoff frequency was 9 kHz, which is used in modern hearing aids [7]. The coefficients of the filter were fixed in nature. The values of the coefficients for the 17 order filter have been taken from [6].

A. *Filter architecture*

The architecture of the FIR filter has been illustrated in Fig.1. The multiplier used in the basic FIR filter in the direct transposed form has been replaced by the precomputation bank in our architecture. The precomputation bank gives the product of the input signal x with the coefficients (C0 to C17) of the 17 order filter simply by the process of shift -and-add. While implementing the HCS CSD architecture proposed in [6], carry-look-ahead (CLA) adder had been used as the structural adder in the filter. To achieve a faster response combined with an increase in area efficiency, the CLA adder blocks have been replaced by 4:2 compressor blocks. The 4:2 compressor accounts for a 3 xor gate delay for sum computation [8]. The number of gates required in the compressor block were found to be relatively less as compared to CLA. Therefore, the struc-tural CLA adder blocks in the filter have been substituted by the basic 4:2 compressor blocks

in the proposed architecture. The resulting sum and carry from the output of the compressor in each tap of the filter have been propagated from one tap to the next by use of delay elements, which have been realized by the basic D flip-Flops (DFF). The sum and carry outputs from

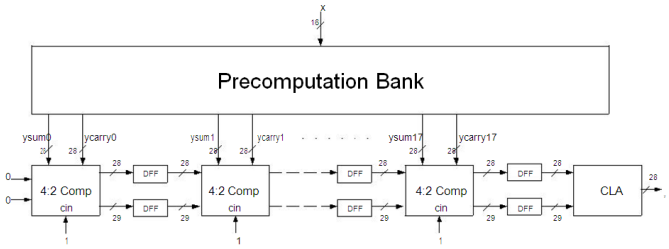

Fig 1: FIR filter

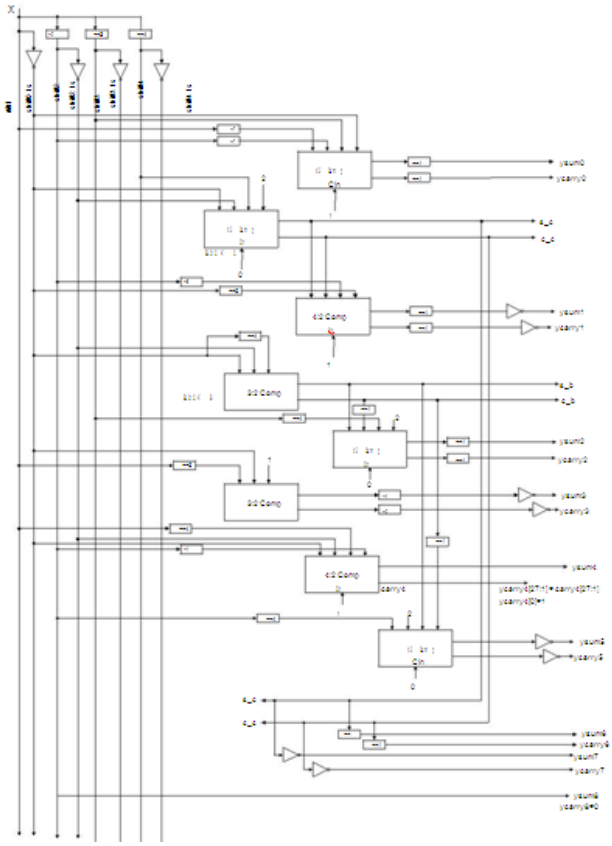

Figure 2. Precomputation Bank

the final tap of the filter have been accumulated to generate the ultimate result (y) by use of CLA adder.

B. Precomputation Bank

In the architecture [6], a total of 10 adders had been used in the precomputation part of the filter which have been realized using CLA adders. However, due to better performance provided by 4:2 compressors, the 10 adders in precomputation part have been substituted by a combination of six 4:2 compressors and two 3:2 compressors in the proposed design. The 4:2 compressor is implemented using the architecture in [8]. Thus, the proposed architecture for precomputation bank resulted in a faster and area optimized design, at the cost of increased power dissipation within tolerable limits. The proposed architecture for the precomputation bank has been depicted in Fig.2. The filter coefficients $_0$ to $_{17}$ are symmetrical in nature and have been given in Table I.

All the filter coefficients have been considered of 12 bits and the input signal of 16 bits, thus generating a 28 bit result corresponding to each coefficient. Since 4:2 compressors have been used in the architecture of the precomputation bank, the product of the input signal with each coefficient was available in the form of 2 terms − ysum and ycarry, each of 28 bits. The final output of the FIR filter is also of 28 bits which is long enough to account for all possible combinations of the input. The CSD notation in [6] has been used to generate the product of the input with the filter coefficients. As shown in Fig.2, 4:2 compressors have been used in the computation of ysum0, ycarry0, ysum1, ycarry1, ysum2, ycarry2, ysum4, ycarry4, ysum5 and ycarry5. These terms are the product of input signal x with the coefficients $_0$, $_1$, $_2$, $_4$ and $_5$. The 3:2 compressor, which is an array of full adders, have been used in the case of $_3$ to generate ysum3 and ycarry3. Here, the 3:2 compressor was used instead of 4:2 compressor, as only 3 terms were required to be added to compute the desired result. This helped in reduction of the redundant hardware resources, which would have been utilized in the case of 4:2 compressor block. As shown in Fig. 2, Block B − a 3:2 compressor and Block C − a 4:2 compressor have been used to generate the intermediate results , and , respectively. These signals act as common input operands to the compressor blocks, later used in the computation of ysum1, ycarry1, ysum2, ycarry2, ysum5, ycarry5, ysum6, ycarry6, and ysum7, ycarry7. These terms are the product of input signal x with the coefficients $_1$, $_2$, $_5$, $_6$ and $_7$. No additional compressor blocks were needed to generate ysum8 and ycarry8. Also, due to the symmetric nature of coefficients, ysumi and ycarryi for i = 9 to 17 need not be calculated again by separate set of hardware.

The two outputs of each compressor from the precomputation bank were available as inputs to the structural compressors, corresponding to each of the filter coefficients as shown in Fig.1. Another important thing to be considered in the proposed architecture was the computation of the negative numbers. The input signal x has been considered positive in nature. Also, the coefficients $_1$, $_3$, $_5$, $_7$, $_{10}$, $_{12}$, $_{14}$, $_{16}$ as mentioned in Table I are negative. The product of the input with these coefficients yields a negative result, which needs to be represented in 2′s complement form. 2′s complement form for a number can be generated by adding ′1′ to the 1′s complement form of that number. Using this logic, the ysum and ycarry terms corresponding to the negative valued coeficients been computed in their 1′s complement form. The additional 1s to be added to ysum and ycarry for the computation of the desired 2′s complement form, have been adjusted in the Cin bit of the structural 4:2 compressor, as shown in Fig.1. Also, such additional 1s needed for generating the 2′s complement form of various intermediate terms in the precomputation bank, have been smartly accommodated at suitable places as shown in Fig.2.

III. Implementation and Results

The filter specifications and values of the filter coefficients have been directly taken from the architecture in [6]. The input signal considered positive in nature, has been represented by 16 bits. The delay element in the proposed work has been realized by basic D Flip-Flop. Carry-look-ahead adder [8] has been used in place of all the required adder blocks in both the architectures to achieve a fair comparison. All the blocks in the architecture have been finally implemented in Verilog HDL [9]. The synthesis results of the architecture in [6] and the proposed architecture have been tabulated in Table II.

Both the architectures have been implemented by RTL compiler tool from Cadence in 90nm technology. The synthesis results have been illustrated in Table II.The proposed architecture provides an improvement of 46.3% in ADP as well as an improvement of 23.7% in PDP compared to the architecture proposed in [6]. The reduction in area combined with the fact of speed improvement in the proposed method is due to the use of 4:2 compressors instead of CLA adder blocks. The area, delay and power reports of CLA adder and compressor for 2 29-bit inputs, as required in each tap of the FIR filter have been tabulated in Table III. Thus, the compressor works much better with respect to factors of area and speed, at the expense of little increment in power. Also, two delay blocks − one for propagating sum and the other for carry, are required in the proposed filter architecture as compared to that of only one delay block in the based filter architecture. This fact combined with the extra power needed in the compressor justifies for the overall power increment in the presented work.

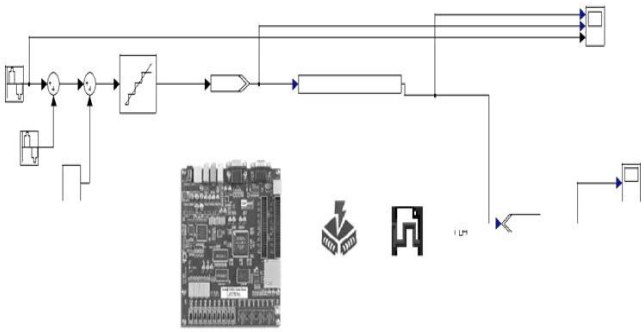

Figure 3. DSP Builder Implementation of the Proposed filter

Functionality Verification

In this section, the functionality of the proposed filters is discussed. The implementation the proposed filter was done using Altera DSP builder. The Fig. 3 shows the sources

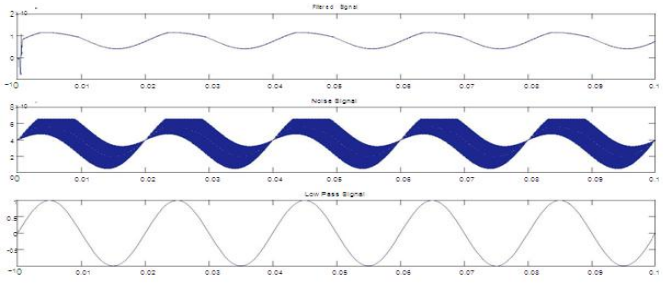

Figure 4. Waveforms

connected to the filter. The functionality verification of the filters is follows as: We considered two signal sources. One input signal is of 50 Hz and other is noise signal of 11 kHz. The sampling frequency is 22.05 kHz while the cutoff frequency is 9 kHz. Using HDL import block the filter is imported and synthesized. For a duration of 0.1 sec the signal is passed to the filter block. As per the specifications of the

TABLE II. Synthesis Results

Filter	Area(mm2)	Delay(ns)	Power(mW)	ADP	ADP Gain(%)	PDP	PDP Gain(%)
[6]	0.107085	10.063	8.8804	1.077596356		89.36346452	
Proposed	0.094938	6.097	11.188067	0.578836999	46.2844336	68.21364445	23.66719

TABLE III. Synthesis Results of CLA and 4:2 Compressor

	Area	Area Gain(%)	Delay (ns)	Delay Gain (%)	Power(mW)	Power Loss(%)
CLA	5219		5.266		0.1186	
4:2 COMPRESSOR	2680	48.64916651	1.142	78313710 6	0.1215	2.445193929

filter, the noise signal of 11 kHz was filtered out and only 50 Hz input signal was passed. The waveforms are shown in Fig.4

IV. Conclusion

An architecuture for the 17 order FIR filter in hearing aid application has been proposed in this paper. In place of adders, 4:2 compressors have been used in the proposed architecture. It is evident from the synthesis results that the proposed architecture is highly efficient in terms of area and speed. Although more power is incurred for the proposed architecture, the PDP has been improved by nearly 23.7% due to significant increase in speed (39.4%). As area utilization too has been decreased by about 11.34%, the ADP is upgraded by around 46.3% in the proposed architecture. Thus, the proposed architecture based on the use of 4:2 compressors is an efficient one for the design of FIR filter in hearing aid application

References

[1] F. Carbognani, F. Burgin, L. Henzen, H. Koch, H. Magdassian, C. Pedretti, H. Kaeslin, N. Felber, and W. Fichtner, "A 0.67-mm 2 45- w dsp vlsi implementation of an adaptive directional microphone for hearing aids," in *Circuit Theory and Design, 2005. Proceedings of the 2005 European Conference on*, vol. 3, pp. III–141, IEEE, 2005.

[2] S. K. Mitra and Y. Kuo, *Digital signal processing: a computer-based approach*, vol. 2. McGraw-Hill New York, 2006.

[3] H. Teutsch and G. W. Elko, "First-and second-order adaptive differential microphone arrays," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, pp. 35–38, 2001.

[4] C.-Y. Pai, A. J. Al-Khalili, and W. E. Lynch, "Low-power constant-coefficient multiplier generator," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 35, no. 2, pp. 187–194, 2003.

[5] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable fir filter for low-power and high-performance applications," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 2, pp. 348–357, 2004.

[6] F. Carbognani, F. Buergin, D. Kraehenbueh, F. Zuercher, N. Felber, H. Kaeslin, and W. Fichtner, "Low-power constant-coefficient fir filtering in a hearing aid application," in *Solid-State and Integrated Circuit Tech-nology, 2006. ICSICT '06. 8th International Conference on*, pp. 1637– 1639, 2006.

[7] T. A. Ricketts, A. B. Dittberner, and E. E. Johnson, "High-frequency amplification and sound quality in listeners with normal through moderate hearing loss," *Journal of Speech, Language, and Hearing Research*, vol. 51, no. 1, pp. 160–172, 2008.

[8] I. Koren, *Computer arithmetic algorithms*. Universities Press, 2002.