HIGH SPEED FIR FILTER USING CSLA AND BOOTH MULTIPLIER

1V.Swathi, 2G.T.Nikhitha, 3B.Krishna Chaitanya, 4Lokesh Kumar, 5N.Pallavi
1,2,3,4 Student, 5 Assistant Professor
Department of ECE,
MLR Institute of Technology, Hyderabad, INDIA

Abstract— High speed finite impulse response (FIR) filter designs are accomplished using the conviction of rounded truncated multipliers. We consider both the optimization of bit width and hardware resources without immolating the frequency response and output signal precision. Total area cost is reduced by using a non uniform coefficient quantization with proper filter order. Multiple constant multiplication/accumulation in a direct FIR structure is designed using an improved version of truncated multipliers. Instead of normal multiplier we are using booth multipliers to reduce the partial products and carry select adder to reduce the delay. Comparisons with previous FIR design approaches show that the proposed design attain the best area and power results.

IndexTerms— Finite impulse response, Multipliers, Bit width, Adders, Radix-4, Accumulation, Optimization, BEC.

I. INTRODUCTION

Finite impulse response (FIR) digital filter is one of the elementary components in shedload digital signal processing (DSP) and communication systems. It is also universally used in many portable applications with limited area and power budget.

A general FIR filter of order M can be expressed as

\[ y[n] = \sum_{i=0}^{M-1} a_i x[n-i]. \]  

(1)

In case of linear phase, the coefficients are either symmetric or anti-symmetric with \( a_i = a_{M-i} \) or \( a_i = -a \). There are two basic FIR structures, direct form and transposed form, as shown in Fig. 1 for a linear-phase even-order FIR filter. In the direct form in Fig. 1(a), the multiple constant multiplication accumulation (MCM) module performs the concurrent multiplications of solitary delayed signals and relevant filter coefficients, followed by accumulation of all the products. Thus, the operands of the multipliers in MCM are delayed input signals \( x[n-i] \) and coefficients \( a_i \).

In the transposed form in Fig. 1(b), the operands of the multipliers in the MCM module are the current input signal \( x[n] \) and coefficients. The repercussion of solitary constant multiplications go through structure adders (SAs) and delay elements. In the past decades, there are many papers on the designs and implementation of low cost or high speed FIR filters [1]-[13],[15]-[19]. In order to avoid costly multipliers, most prior hardware implementations of digital FIR filters can be divided into two categories: multiplier less based and memory based. Multiplier less-based designs discern MCM with shift-and add operations and share the common sub operations using canonical signed digit (CSD) recoding and common sub expression eradication (CSE) to cut down the adder cost of MCM [1]-[10]. In [18] and [19], more area savings are seized by jointly keeping in mind the optimization of coefficient quantization and CSE. Majority multiplier less MCM-deployed FIR filter designs employ the transposed composition to accredit for cross-coefficient apportion and tend to be faster, notably when the filter order is large. However, the area of delay elements is larger compared with that of the direct form due to the range expansion of the constant multiplications and the subsequent additions in the SAs. In [17], Blad and Gustafsson entrusted high-throughput (TP) FIR filter designs by pipelining the carry-save adder trees in the constant multiplications using integer linear programming to cut down the area cost of full adders (FAs), half adders (HAs), and registers (algorithmic and pipelined registers).

Memory-based FIR designs consist of two types of approaches: lookup table (LUT) methods and distributed arithmetic (DA) methods [11]-[13]. The LUT-based design stores in ROMs odd multiples of the input signal to realize the constant multiplications in MCM [11]. The DA-based approaches recursively accumulate the bit-level partial results for the inner product computation in FIR filtering [12],[13].

An important design issue of FIR filter implementation is the optimization of the bit widths for filter coefficients, which has direct impact on the area cost of arithmetic units and registers. Forbye, since the bit widths after multiplications swell, myraid DSP applications do not need full-precision outputs. Instead, it is expedient to generate loyally rounded outputs where the total error instigated in quantization and rounding is no more than one unit of the last place (ulp) defined as the weighting of the
least significant bit (LSB) of the outputs. In this brief, we present low-cost implementations of FIR filters based on the direct structure in Fig. 1(a) with faithfully rounded truncated multipliers.

**Fig.1.** Structures of linear-phase even-order FIR Filters (a) Direct form and (b) transposed form.

The MCMA module is realized by accumulating all the partial products (PPs) where unnecessary PP bits (PPBs) are removed without affecting the final precision of the outputs. The bit widths of all the filter coefficients are trivialized using non-uniform quantization with biased word lengths in order to reduce the hardware cost while still persuasively the specification of the frequency response. This brief is organized as follows. Section II discusses the non-uniform quantization and optimization of filter coefficients. Section III and IV describes the PP generation and compression in the Booth Multiplier. Section V describes the experimental results.

**II. COEFFICIENT QUANTIFICATION AND OPTIMIZATION**

A collective flow of FIR filter design and implementation can be cleaved into three stages: resolving filter order and coefficients, coefficient quantization, and hardware optimization, as shown in Fig. 2. In the first stage, the filter order and the consistent coefficients of infinite precision are unwavered to satisfy the specification of the frequency response. Then, the coefficients are quantized to finite bit reliability. Finally, sundry optimization approaches such as CSE are used to shrink the area cost of hardware implementations. Majority of FIR filter implementations focus on the hardware optimization stage.

Following FIR filter operations, the output signals have vast bit width due to bit width proliferation after multiplications. In many pragmatic situations, only partial bits of the full precision outputs are demanded. FIR filter have 12 bits and the filter coefficients are quantized to 10 bits, the bit width of the consequent FIR filter output signals is at least 22 bits, but we might need only the 12 most significant bits for successive processing. In this brief, we arrogate the direct FIR structure with MCMA because the area cost of the flip-flops in the delay elements is smaller collate with that of the transposed form. Forbye, we mutually
consider the three design stages in Fig. 2 in order to achieve more efficient hardware design with faithfully rounded output signals. Unlike conventional uniform quantization of filter coefficients with equal bit width, the non-uniform quantization technique with possibly different bit widths is adopted in this brief.

III. BOOTH MULTIPLIER AND SQRT CSLA

Radix-4 modified booth encoding has been used which allows for the reduction of partial product array by half \([n/2]\). The bit pair recoding table is shown in Table 3. In the implemented algorithm for each group of three bits \((y_{2i+1}, y_{2i}, y_{2i-1})\) of multiplier, one partial product row is generated according to the encoding in Table. Radix-4 modified booth encoding (MBE) signals and their respective partial products has been generated using figures 3 and 4. For each partial product row, figure 8 generates the one, two, and neg signals. These values are then given to the logic in figure 9 with the bits of the multiplicand, to produce the whole partial product array. To prevent the sign adjunct the obtained partial products are elongate as shown in figure 5 and the product has been calculated using carry select adder.

TABLE 1: Modified Booth Encoding Table

<table>
<thead>
<tr>
<th>Binary</th>
<th>MB Encoding</th>
<th>Input Carry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sign (y_0)</td>
<td>one (y_1)</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 1</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1</td>
<td>+2</td>
<td>0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

![Figure 3. MBE signal generation](image1)

![Figure 4. Partial Product Generation](image2)
Figure 5. Sign prevention extension of partial products.

IV. PP COMPRESSION

The FIR filter design in this brief embrace the direct form in Fig. 1(a) where the MC MA module sums up all the products $a_i \times x[n-i]$. Alternatively of assembling individual multiplication for each product, it is more systematic to collect all the PPs into a distinct PPB matrix with carry-save addition to reduce the height of the matrix to two, followed by a final carry propagation adder. Fig. 6 illustrates the contrast of individual multiplications and combined multiplication for $A \times B + C \times D$. In order to avoid the sign appendage bits, we complement the sign bit of each PP row and add some bias persistent using the property $\bar{s} = 1 - s$. When collected into the last row in the PPB matrix. The complements of PPBs are denoted by white circles with over bars.

In the dutiful rounded FIR filter implementation, it is entailed that the total error pioneered during the arithmetic operations is no stupendous than one ulp. We recast a neoteric truncated multiplier design in so that more PPBs can be expunged, leading to smaller area cost.

Fig. 6. Multiplication/accumulation using (a) individual PP compression and (b) combined PP compression.

Fig. 7. Modified 16-bit SQRT CSLA
V. BINARY TO EXCESS-I CONVERTER (BEC)

The main purpose of this work is to use BEC rather than RCA with carry in=1 in order to reduce the area of the regular linear CSLA as well as regular SQRT required. A anatomy and the function of 3-bit BEC are shown in fig. 7. The Boolean expressions for 3-bit BEC is exhibit below

\[ X_0 = \neg B_0 \quad (2) \]
\[ X_1 = B_0 \oplus B_1 \quad (3) \]
\[ X_2 = B_2 \oplus (B_1 \& B_0) \quad (4) \]

VI. SIMULATION RESULTS

Fig. 7. A 3-bit BEC

Fig. 8. Booth Multiplier results

Fig. 9. Carry Select Adder Results.
VII. CONCLUSION

This brief has presented high speed FIR filter designs by cooperatively considering the optimization of coefficient bit width and hardware resources in enactments. Although most prior designs are based on the transposed form, we observe that the direct FIR structure with Booth multiplier and CSLA leads to smallest Delay.

VIII. ACKNOWLEDGMENT

I would like to thanks our Assistant Professor, N. Pallavi for her guidance and for helping us to carry out our project successfully.

REFERENCES


