LUT based FIR Filter Design & implementation on FPGA using Faithfully Rounded Truncated Multiple Constant Multiplication/Accumulation

U. Sudha Rani, S.P. Suresh Naik
E.C.E Department (VLSI & ESD)
GATES Institute of Technology, Gooty
sudha.u.rani@gmail.com, sureshnaik_sp@yahoo.co.in

Abstract—Low-cost finite impulse response (FIR) designs are presented using the concept of faithfully rounded truncated multipliers. We jointly consider the optimization of bit width and hardware resources without sacrificing the frequency response and output signal precision. Non-uniform coefficient quantization with proper filter order is proposed to minimize total area cost. Multiple constant multiplication/accumulation in a direct FIR structure is implemented using an improved version of truncated multipliers. Comparisons with previous FIR design approaches show that the proposed designs achieve the best area and power results.

I. INTRODUCTION

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

A general FIR filter of order M can be expressed as

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

In case of linear phase, the coefficients are either symmetric or antisymmetric with \( a_i = a_{M-i} \) or \( a_i = -a_{M-i} \). 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 (MCM)/accumulation (MCMA) module performs the concurrent multiplications of individual delayed signals and respective filter coefficients, followed by accumulation of all the products. Thus, the operands of the multipliers in MCMA are delayed input signals \( x[n-i] \) and coefficients.

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 results of individual constant multiplications go through structure adders (SAs) and delay elements. In the past decades, there are many papers on the designs and implementations of low-cost or high-speed FIR filters. In order to avoid costly multipliers, most prior hardware implementations of digital FIR filters can be divided into two categories: multiplierless based and memory based.

Multiplierless-based designs realize MCM with shift-and-add operations and share the common suboperations using canonical signed digit (CSD) recoding and common subexpression elimination (CSE) to minimize the adder cost of MCM. More area savings are achieved by jointly considering the optimization of coefficient quantization and CSE. Most multiplierless MCM-based FIR filter designs use the transposed structure to allow for cross-coefficient sharing and tend to be faster, particularly 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. Blad and Gustafsson presented high-throughput (TP) FIR filter designs by pipelining the carry-save adder trees in the constant multiplications using integer linear programming to minimize the area cost of full adders (FAs), half adders (HAs), and registers (algorithmic and pipelined registers).
Specification of frequency response

- Finding filter order an coefficients
- Coefficient quantization
- Hardware optimization

Fig.2. Three stages in digital FIR filter design and implementation.

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

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. Moreover, since the bit widths after multiplications grow, many DSP applications do not need full-precision outputs. Instead, it is desirable to generate faithfully rounded outputs where the total error introduced 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. 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 minimized using nonuniform quantization with unequal word lengths in order to reduce the hardware cost while still satisfying the specification of the frequency response.

This brief is organized as follows. Section II discusses the nonuniform quantization and optimization of filter coefficients. Section III describes the PP generation and compression in the faithfully rounded MCMA module. Section IV compares the experimental results.

II. COEFFICIENT QUANTIZATION AND OPTIMIZATION

A generic flow of FIR filter design and implementation can be divided into three stages: finding filter order and coefficients, coefficient quantization, and hardware optimization, as shown in Fig. 2. In the first stage, the filter order and the corresponding coefficients of infinite precision are determined to satisfy the specification of the frequency response. Then, the coefficients are quantized to finite bit accuracy. Finally, various optimization approaches such as CSE are used to minimize the area cost of hardware implementations. Most prior FIR filter implementations focus on the hardware optimization stage.

After FIR filter operations, the output signals have larger bit width due to bit width expansion after multiplications. In many practical situations, only partial bits of the full-precision outputs are needed. For example, assuming that the input signals of the FIR filter have 12 bits and the filter coefficients are quantized to 10 bits, the bit width of the resultant FIR filter output signals is at least 22 bits, but we might need only the 12 most significant bits for subsequent processing.

Given specification of frequency response;
M=Parks_McClellan(); //Find FIR filter order M
//Step 1: Uniform Quantization
Area_min=MAX;
While(){
  Coeff=remez(M); //Find filter coefficients for given M
  B<=0;
  While(freq_resp_satisfied(B)=0)
    B<-B+1;
    //Quantize coefficients to B bits
    Select CSD or radix-4 Booth recoding with fewer nonzero digits;
  Area=area_cost_estimate(B);//estimate total area cost of FIR filter
  If(area<=area_min){
    Area_min=area;
    M<-M+1;
  }
  else{
    M<-M-1;
    Break;
  }
}
//Step 1: Non-uniform Quantization
N=ceil(M/2); M is even //N non-redundant coeffs. for linear phase FIR
B<0..1,...,N-1;
while(){
  reduction = 0;
  for(i=0 to N-1)
    if(freq_resp_satisfied(Bi)=1){
      Bi<-Bi-1;
      reduction = 1;
  }
  if(reduction=0)break;//no further bit-width reduction is possible
}
//Step 3: Coefficient Fine-tune
while(){
  reduction=0;
  for(i=0 to N-1)
    a_i<a_i+2^-Bi; //increase each coefficient value by 1 ulp
    if(freq_resp_satisfied(Bi+1)=1){
      Bi<-Bi-1;
      reduction = 1;
      continue; }
  if(reduction=0)break; //no bit-width reduction is possible
}

Fig. 3. Proposed algorithm of coefficient quantization and fine tuning.
In this brief, we adopt the direct FIR structure with MCMA because the area cost of the flip-flops in the delay elements is smaller compared with that of the transposed form. Furthermore, we jointly 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 nonuniform quantization technique with possibly different bit widths is adopted in this brief. Fig. 3 shows the pseudocode of the proposed quantization scheme.

Initially, subroutine Parks_McClellan() is used to find the filter order M for the given frequency response. Step 1 of uniform quantization starts with calling the MATLAB built-in function remez() to find the coefficients for the FIR filter of order M. Then, we quantize the coefficients with enough bits and generate the set of uniformly quantized coefficients ai with equal bit width B. The subroutine freq_resp_satisfied() checks if the frequency response is still satisfied after quantization.

After coefficient quantization, we perform recoding to minimize the number of nonzero digits. In this brief, we consider CSD recoding with digit set of {0,1,−1} and radix-4 modified Booth recoding with digit set of {0,1,−1,2,−2} and select the one that results in smaller area cost.

While most FIR filter designs use minimum filter order, we observe that it is possible to minimize the total area by slightly increasing the filter order. Therefore, the total area of the FIR filter is estimated using the subroutine area_cost_estimate() using the approach in [20]. Indeed, the total number of PPBs in the MCMA is directly proportional to the number of FA cells required in the PPB compression because a FA reduces one PPB.

After Step 1 of uniform quantization and filter order optimization, the nonuniform quantization in Step 2 gradually reduces the bit width of each coefficient until the frequency response is no longer satisfied.

Finally, we fine-tune the nonuniformly quantized coefficients by adding or subtracting the weighting of LSB of each coefficient and check if further bit width reduction is possible. Using the algorithm in Fig. 3, we can find the filter order M and the nonuniformly quantized coefficients that lead to minimized area cost in the FIR filter implementation.

III. PP TRUNCATION AND COMPRESSION
The FIR filter design in this brief adopts the direct form in Fig. 1(a) where the MCMA module sums up all the products ai×x[n−i]. Instead of accumulating individual multiplication for each product, it is more efficient to collect all the PPBs into a single PPB matrix with carry-save addition to reduce the height of the matrix to two, followed by a final carry propagation adder. Fig. 4 illustrates the difference of individual multiplications and combined multiplication for A×B+C×D.

In order to avoid the sign extension bits, we complement the sign bit of each PP row and add some bias constant using the property s=1−s, where sis the sign bit of a PP row, as shown in Fig. 5. All the bias constants are collected into the last row in the PPB matrix. The complements of PPBs are denoted by white circles with over bars.

Fig. 4. Multiplication/accumulation using (a) individual PP compression and (b) combined PP compression.

Fig. 5. Generation of PPBs considering sign extension and negation.

Fig. 6. Truncated multiplier designs using (a) the normal approach (b) the improved version.

The removal of unnecessary PPBs is composed of three processes: deletion, truncation, and rounding. Two rows of PPBs are set undeletable because they will be removed at the subsequent truncation and rounding. The error ranges of deletion, truncation, and rounding before and after adding the offset constants are given as follows:

\[\text{error}_{\text{deletion}} = \text{error}_{\text{truncation}} + \text{error}_{\text{rounding}}\]
Fig. 6(a) shows an example of the approach where the gray circles, crossed green circles, and crossed red circles represent respectively the deleted bits, truncated bits, and rounded bits.

In this brief, we propose an improved version of the faithfully rounded truncated multiplier design as shown in Fig. 6(b). Only Fig. 7 shows the illustrative architecture of MCMAT that removes unnecessary PPBs. The white circles in the L-shape block represent the undeletable PPBs. The deletion of the PPBs is represented by gray circles.

After PP compression, the rounding of the resultant bits is denoted by crossed circles. The last row of the PPB matrix represents all the offset and bias constants required including the sign bit modifications.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

We implemented three FIR filters with the specifications given in Table I. M is the original filter order while Mopt is the filter order with optimized total area using the method in Fig. 2. B denotes the number of fractional bits for uniformly quantized coefficients with filter order Mopt. EWL is the effective word length without counting the leading sign bits. Passband and stopband edge frequencies normalized to one, and Apass and Astop denote the corresponding peak-to-peak ripples.

Tables II–IV show the implementation results for the input and output signals of 12 fractional bits using Synopsys Design Compiler with the Taiwan Semiconductor Manufacturing Company 90-nm standard cell library. The total area includes that of arithmetic units (MCM and SAs) and D-type Flip-Flops.

TABLE I
SPECIFICATIONS OF THE THREE FIR FILTERS UNDER CONSIDERATION

<table>
<thead>
<tr>
<th>Filter</th>
<th>M</th>
<th>Mopt</th>
<th>B</th>
<th>EWL</th>
<th>fpass</th>
<th>fstop</th>
<th>Apass (dB)</th>
<th>Astop (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>LP</td>
<td>25</td>
<td>28</td>
<td>11</td>
<td>10</td>
<td>0.15</td>
<td>0.25</td>
<td>0.09</td>
</tr>
<tr>
<td>B</td>
<td>LP</td>
<td>59</td>
<td>64</td>
<td>15</td>
<td>12</td>
<td>0.02</td>
<td>0.07</td>
<td>0.20</td>
</tr>
<tr>
<td>C</td>
<td>HP</td>
<td>121</td>
<td>121</td>
<td>19</td>
<td>17</td>
<td>0.40</td>
<td>0.37</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Tables II–IV show the implementation results for the input and output signals of 12 fractional bits using Synopsys Design Compiler with the Taiwan Semiconductor Manufacturing Company 90-nm standard cell library. The total area includes that of arithmetic units (MCM and SAs) and D-type Flip-Flops.
TABLE I V
SYNTHESIS RESULTS OF FILTER C WITH 12-BIT INPUT AND OUTPUT SIGNALS

<table>
<thead>
<tr>
<th>Filter C (121-tap HP)</th>
<th>CSM/Br (um^2)</th>
<th>SAs (um^2)</th>
<th>DFFs (um^2)</th>
<th>Area (um^2)</th>
<th>Delay (ns)</th>
<th>TP (M)</th>
<th>Power (mW)</th>
<th>AP/TP</th>
<th>Structur</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSD/Br@800MHz</td>
<td>50,835</td>
<td>62,047</td>
<td>57,451</td>
<td>170,311</td>
<td>6.23</td>
<td>161</td>
<td>8.82</td>
<td>3.99</td>
<td>Trans</td>
</tr>
<tr>
<td>NRSCSEf@11</td>
<td>34,830</td>
<td>62,047</td>
<td>57,451</td>
<td>195,028</td>
<td>6.23</td>
<td>161</td>
<td>8.45</td>
<td>3.36</td>
<td>Trans</td>
</tr>
<tr>
<td>ID_D [12]</td>
<td>32,068</td>
<td>22,990</td>
<td>50,585</td>
<td>7.70</td>
<td>11</td>
<td>2.04</td>
<td>4.36</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td>MCMA</td>
<td>71,888</td>
<td>22,353</td>
<td>94,241</td>
<td>8.94</td>
<td>112</td>
<td>6.00</td>
<td>2.16</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td>MCMA_opt</td>
<td>69,024</td>
<td>22,353</td>
<td>91,777</td>
<td>8.99</td>
<td>111</td>
<td>5.65</td>
<td>2.00</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td>MCMAT_I</td>
<td>49,278</td>
<td>22,353</td>
<td>71,631</td>
<td>8.24</td>
<td>121</td>
<td>4.37</td>
<td>1.11</td>
<td>Direct</td>
<td></td>
</tr>
<tr>
<td>MCMAT_II</td>
<td>46,018</td>
<td>22,353</td>
<td>68,791</td>
<td>8.24</td>
<td>121</td>
<td>4.14</td>
<td>1</td>
<td>Direct</td>
<td></td>
</tr>
</tbody>
</table>

The TP is in units of million output data samples per second (Mdata/s). The power is extracted from Synopsys PrimeTime (containing PrimePower) with a frequency of 50 MHz. The normalized area–power product divided by TP(AP/TP) is also included for easy comparison of various designs under the combined metric of area, power, and speed performance.

In multiplierless designs with transposed structure, CSE can effectively reduce the number of adders in MCM compared with CSD recoding. Nonrecursive signed CSE (NRSCSE) and multiroot binary partition graph (MBPG) belong to the category of CSE methods. Note that SAs are not optimized. The adder cost in CMC can be further reduced by joint optimization of coefficient quantization and CSE as suggested. The constant multiplication is realized by storing the odd multiples of the constant in LUT implemented with dual-port segmented memory sharing memory cells. This approach needs full-custom design of the LUT circuits. We realize the approach using ROM using Verilog codes. However, it is possible to reduce the area cost if customized ROM circuits are designed. 1-D and 2-D systolic arrays are presented to implement FIR filters using DA. We only implement the 1-D DA design due to cost consideration.

Fig. 8. (a) Area, (b) delay, and (c) power of the proposed designs for filter C.

Most of prior FIR filter designs are based on the transposed structure because the major goal is to minimize the cost of adders in MCM that takes less than 20% of the total area. Indeed, the MCM cost in transposed-form NRSCSE and MBPG (and with further coefficient optimization) can be effectively reduced. However, the SAs are not optimized, and the area of DFFs in the transposed forms is larger because of the range expansion of the results after MCM. Take as an example the 121-tap filter C with 19-bit coefficients and 12-bit signals. The direct form requires only 12*120 = 1440 flip-flops, while the transposed form calls for 3701 flip-flops, a significant increase because a resettable DFF cell has about the same area as a FA cell.

Our proposed FIR filter design has four versions. MCMA is the baseline implementation using combined PP compression [similar to Fig. 4(b)] with uniformly quantized coefficients. MCMA_opt is an improved version by adopting the nonuniform quantization in Fig. 3 for coefficient optimization. MCMAT_I and MCMAT_II faithfully truncate PPBs using the approaches in Figs. 6(a) and (b), respectively. Fig. 8 shows the relative comparison of the four designs for filter C. The total numbers of PPBs before compression in MCM, MCMA_opt, MCMAT_I, and MCMAT_II are 3407, 3175, 2124, and 1946, respectively. We observe that MCMA_opt, MCMAT_I, and MCMAT_II have area reduction rates of 3%, 24%, and 27%, respectively, compared with the baseline MCDA designs that are already smaller than prior designs of similar TP.

Although the area costs of the proposed designs are significantly reduced, the critical path delay is increased because all the operations in the MCMA are executed within one clock cycle. It is possible to reduce the delay by adding pipeline registers in the PP compression, where the major goal is to minimize the number of FAs, HAs, and registers (including arithmetic and pipelined registers) using integer linear programming. In this brief, we focus on low-cost FIR filter designs with moderate speed performance for mobile applications were area and power are important design considerations.

From Tables II–IV, we observe that the proposed designs have smaller area and power compared with prior MCM designs. For example, compared with that of MBPG, the total area saving rates are about 32%–54% with more saving for higher order filters. Although transposed FIR structures with CSE can reduce the adder cost in the MCM module, the paid price is more DFFs which occupy about 40% of the total area. Regarding memory-based designs, the 1-D DA requires 12 cycles to generate an output. The LUT designs have high area cost because the synthesized ROM does not support sharing of memory cell circuits. In terms of the combined metric of area, power, and TP, our proposed designs also have the best AP/TP with MCMAT_II normalized to one.

V. CONCLUSION

This brief has presented low-cost FIR filter designs by jointly considering the optimization of coefficient bit width and hardware resources in implementations. Although most prior designs are based on the transposed form, we observe that the direct FIR structure with faithfully rounded MCMAT leads to the smallest area cost and power consumption.

REFERENCES


