Efficient Truncated Multiplier Design for FIR Filter

S.PRIYADHARSHINI¹, L.RAJA²

¹²Department of Electronics and Communication Engineering, Angel College of Engineering and Technology, Tirupur, Tamilnadu, India

¹meenatchipriya@gmail.com, ²kaushikraja2000@gmail.com

Multipliers have a significant impact in the performance of the DSP system. Some DSP applications require multipliers whose input operands and outputs need to have the same width. These multipliers are called truncated multipliers which eliminates the least significant columns. Thus there occurs truncation error when compared to the conventional multiplier, which is tolerable in some digital signal processing applications. To compensate the truncation error, compensation methods are introduced, where the coefficients in compensation function are quantized significantly. Thus the truncation error of the multiplier is not kept to a minimum. This work presents a quantization method which minimizes the truncation error and can be efficiently implemented. It is observed that the implementation of the proposed method in FIR filter provides nearer representation of the output that obtained through theoretical calculation and experimental results confirms low cost in terms of area and power.

Keywords: Multipliers, Digital Signal Processing, Truncated Multiplier, Truncation Error

1. INTRODUCTION

Multiplication operations are mandatory in digital signal processing. Multiplication involves two basic operations. First is about the generation of the partial products and second thing is about their sum. It can be performed using two kinds of multiplication algorithms, serial and parallel. In order to increase the speed, parallel multipliers can be used [1]. But it occupies large area and consumes much power. Multiplication of two operands results in product, which is of twice the word length of the input operands. There are some applications where full width of the multiplication output is not required and can accept error to a certain level. There truncated multipliers can be used to reduce area and power consumption [3]. Generally in truncated multipliers, the least significant parts are eliminated and keep only the most significant part of the multiplier. Thus the hardware complexity and power consumption can be reduced by removing the adders that is necessary to compute the least significant columns. Large truncation error occurs for the direct truncated multiplier. In order to reduce the truncation error some error compensation methods are introduced, so that the error can be kept to a minimum. This compensation function provides the good estimate of columns that are eliminated. They can be introduced by the following methods. They are constant correction method and variable correction method. The constant correction method [4] adds a constant to remaining columns that are kept, based on analytical evaluation of the bits, which are not formed. The drawback of this approach is that the correction value is independent of input values. The variable correction method [5] is based on the input values hence it improves the accuracy of the truncated multiplier.

Most of the previous work [6-10] to obtain the compensation circuits were based on exhaustive simulation and probabilistic estimation hence it is the time consuming process. The work in [12] provides the compensation function derived analytically and does not provide minimum error. This work provides a quantization method which quantizes the coefficients in the compensation function in such a way to minimize the error. After the formation of partial product matrix, carry save reduction approach is used, that ends up the reduction stage with two rows, which is the final stage. These two rows are then added using carry propagate adder or carry ripple adder. In order to examine the efficiency of this truncated multiplier, it is incorporated in the (MAC) multiplication accumulation part of the FIR filter. The entire work is coded using Hardware Description Language (VHDL) then simulated and synthesized using Xilinx ISE and the corresponding reports are obtained to evaluate the performance of the design.

This paper is organized as follows. The section II describes the formation of partial product matrix of truncated multiplier. The section III explains the reduction approach of the partial product matrix. The section IV describes the incorporation of truncated multiplier in FIR filter. The section V shows the simulation and synthesis results. The section VI concludes this work.

2. PARTIAL PRODUCT MATRIX FORMATION

An efficient truncated multiplier design with hardware representable compensation function is presented. The partial product matrix is divided into MSP (Most Significant Part) and LSP (Least Significant Part). Further the LSP is subdivided into LSP major and LSP minor. The elements of LSP major which consists of ‘k’ columns where ‘k’ indicates design parameter. The remaining "n = n – k” columns forms the elements of LSP minor. It is known that in multiplier design each column is found to have its own weight. For simplicity the operands are considered as unsigned binary numbers. Thus the weight of the Least Significant Bit of the truncated multiplier...
output is given by “\(2^n\)”, where ‘n’ represents the number of significant bits. The column that is left most to LSP minor is represented as Correction Column (CC). The weight of CC is given by “\(2^{n-k-1}\)”, and it is used for the calculation of compensation function. The elements of the Correction column were indicated as ‘b_i’ and these elements get inserted once or twice to improve the accuracy of the multiplier.

The modified partial product matrix of the truncated multiplier thus can be easily obtained by the compensation function which is given by:

\[
f_c = \text{LSB}_{CC} \left\{ a_0 + \sum_{i=1}^{n_p} a_i b_i \right\}
\]

(2.1)

where ‘a_i’ represents coefficients and ‘b_i’ represent elements of the Correction Column. Based on desired values for “n, k”, the compensation value can be obtained using equation (1) and the coefficients which are obtained originally are in real form, hence they need to be quantized for hardware representation. The previous work provides the rough quantization of coefficients. This work involves the quantization of the coefficients using two bits to have minimum error. The reason to have minimum mean error is that it gets related to the signal to noise ratio of the filtered signal. The compensation function provides the sum of linear terms they are inserted at column weight given by LSB_{CC}. The Fig1 shows the modified partial product matrix for the values “n = 8” and “k = 1”, thus the “n_r = 7”.

![Fig.1 Modified Partial product matrix for n=8, k=1](image)

3. PARTIAL PRODUCT REDUCTION

The modified Partial Product Matrix of the truncated multipliers can be summed by using a carry-save reduction approach, and it is followed by a fast carry-propagate adder. The block diagram of overall implementation of parallel truncated multiplier is shown in Fig 2. The input operands multiplier and multiplicand are of ‘n’ bit length and the bit length of output operand Product is also of length ‘n’ bit. The carry save reduction method adopted for the implementation of proposed multiplier is Wallace Reduction and for CPA implementation, Carry ripple adder technique is used.

The Wallace reduction scheme starts by grouping the partial product bit matrix into sets of three rows. Then each set get reduced to two rows by incorporating half adders on columns of two bits and full adders on columns of three bits. The rows that are excess not belonging to the set of three are passed to the next reduction stage without any change. Each reduction step gets processed in a similar manner until only two rows are left.

![Fig.2. Overview of the steps in truncated multiplier design](image)
In Wallace reduction for carry propagate addition, Ripple Carry Adder is incorporated, where it employs full adder cells. The carry out signal of each full adder is connected to the carry in signal of the full adder which is present in the next most significant column. The sum output of the “n” full adders form a bit vector that produces the “n” bit product of the truncated multiplier.

4. IMPLEMENTATION IN FIR FILTER

Filters remove the unwanted components of the signal and they are known as signal conditioners. As compared to analog filters, digital filters have advantages like good stability and reliability. These digital filters are of two types: Finite Impulse Response filter and Infinite Impulse Response filter. Compared with the Infinite Impulse Response, the FIR filter is capable of avoiding the drift and noise, which is produced by the IIR filter. Moreover it has linear phase response and high stability. Multiplication with constant has to be performed in FIR filter. The digital FIR filter consists of multiplier block, delay block and adder block. The FIR structure is of two types; they are direct form and transformed form. This work makes use of direct form structure. In direct form, the multiplication accumulation part of FIR filter multiplies the individually delayed signal with the corresponding coefficients concurrently. The Fig.3 shows the block diagram of direct form FIR filter with truncated multiplier.

In the Fig.3 the block with “z⁻¹” indicates the delayed version of the input signal. The input signal x(n) is the binary vector and the multiplier block is implemented using the truncated multiplier describe above and the bit width of input operand and output operand are same. The input to the multiplier block is coefficients and they are obtained for the desired frequency response. The multiplied outputs get accumulated and added to produce the output of the filter.

5. EXPERIMENT RESULTS AND COMPARISONS

To verify the performance of the FIR filter with truncated multiplier, every block is designed using hardware description language VHDL. The conventional FIR filter implements the multiple constant multiplications without considering the bit width. By employing the truncated multiplier in the filter structure, an efficient structure in terms of area and power can be obtained. The truncated multiplier is designed where the input operands and product output are of width 8 bits, the corresponding compensation function is calculated and forms the modified partial product matrix. After the formation of the partial product matrix, they are reduced to two rows using Wallace reduction and then they are summed using the ripple carry adder.

The filter is designed by replacing the conventional multiplier with the truncated multiplier. The number of computations involved depends on the order of the filter. Based on the desired response and the order the filter coefficients can be generated using the Filter Design Tool. They can also be quantized without causing significant change to the frequency response of the filter. It is obvious that increasing the order of the filter provides better frequency response. The input samples to the filter are stored in the form of array and it get shifted, multiplied with the corresponding coefficients which are known earlier and constant, produced the output of the filter upon each positive edge of the clock signal. The design is simulated and synthesized using the tool Xilinx ISE and the corresponding waveforms are analyzed to verify the correctness of the design. In addition area consumption of the design is obtained and also the power consumption of the filter is obtained using Xilinx Power Analyzer.

The simulation result of the truncated multiplier is shown in Fig.4. The coefficients are obtained for 5 tap Low Pass Filter whose frequency specifications are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Frequency</td>
<td>5Hz</td>
</tr>
<tr>
<td>Pass band</td>
<td>40mHz</td>
</tr>
<tr>
<td>Stop band</td>
<td>2.05Hz</td>
</tr>
<tr>
<td>Pass band ripple</td>
<td>0.09dB</td>
</tr>
<tr>
<td>Stop band attenuation</td>
<td>56 dB</td>
</tr>
</tbody>
</table>

The corresponding coefficients are obtained which are in floating form representation, they are then quantized to 8 bits and converted to binary form and are shown in Table 1.
Table 1. Coefficients of 5 Tap Low Pass Filter

<table>
<thead>
<tr>
<th>Coefficients h(n)</th>
<th>Floating Representation</th>
<th>Binary Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(0)</td>
<td>0.067627287400174957</td>
<td>00010001</td>
</tr>
<tr>
<td>h(1)</td>
<td>0.24945378935619286</td>
<td>01000000</td>
</tr>
<tr>
<td>h(2)</td>
<td>0.36447240180296297</td>
<td>01011101</td>
</tr>
<tr>
<td>h(3)</td>
<td>0.24945378935619286</td>
<td>01000000</td>
</tr>
<tr>
<td>h(4)</td>
<td>0.067627287400174957</td>
<td>00010001</td>
</tr>
</tbody>
</table>

From the simulation results, it is observed that the output of the filter is nearer representation of the fractional value that obtained through theoretical calculation. In Fig. 4 for first five input samples the output of the filter is obtained as (11110110) whose decimal value is 250, and it is nearer representation of fractional value 249.140625, which is obtained as output of the filter for first five input samples.

Table 2. Comparison of Various Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Full width filter</th>
<th>Truncated filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of occupied slices</td>
<td>41%</td>
<td>28%</td>
</tr>
<tr>
<td>No. of 4 input LUTs</td>
<td>36%</td>
<td>23%</td>
</tr>
<tr>
<td>No. of bonded IOBs</td>
<td>77%</td>
<td>65%</td>
</tr>
<tr>
<td>Power consumed</td>
<td>87mW</td>
<td>63mW</td>
</tr>
<tr>
<td>Delay</td>
<td>34.420ns</td>
<td>27.771ns</td>
</tr>
<tr>
<td>Energy</td>
<td>2.99njoule</td>
<td>1.75njoule</td>
</tr>
</tbody>
</table>

6. CONCLUSION

In this work an area and energy efficient truncated FIR filter with finest quantization of the coefficients in the compensation function is introduced and they are efficiently represented in hardware. The performance of the proposed truncated filter is compared with the conventional filter and it is experimentally observed that it provides better results in terms of area and power. The proposed truncated filter is area and power efficient and it could be incorporated in the digital signal processing applications where the error can tolerate to a certain level.

ACKNOWLEDGEMENT

The author would like to thank Chairman, Principal and Head of the Department of Electronics and Communication Engineering of Angel College of Engineering and Technology, Tirupur, Tamilnadu, India for their Valuable support.
REFERENCES