DESIGN OF AREA EFFICIENT TRUNCATED MULTIPLIER FOR DIGITAL SIGNAL PROCESSING APPLICATIONS

V.Suruthi[1], Dr.K.N.Vijeyakumar[2]

[1]PG Scholar, [2]Assistant Professor, Dept of EEE,

Dr. Mahalingam College of Engineering & Information Technology , Pollachi, Tamilnadu, (India)

ABSTRACT

Low-cost finite impulse response (FIR) design are presented using the concept of area efficient truncated multiplier using modified booth multiplier for signed bit multiplication. In this proposed area efficient truncated multiplier the area reduction can be achieved by non-generation of initial partial product and deletion of certain LSBs in generation of partial product of the multiplier. In order to reduce the error factor produced due to non-generation and deletion of certain LSBs bit, the compensation bits are added at the appropriate retained bit position. This non-generation of initial partial product and deletion of certain LSBs in multiplier leads to reduction of number of half adder and full adder used in partial product accumulation for final product. During the truncation process fixed bit width multiplier provide us the less error comparing to conventional multiplier in existing. Comparisons with previous FIR design, the proposed design shows the design achieve the best area, delay and power.

Keywords: Digital Signal Processing(DSP), Finite İmpulse Response(FIR), Least Significant Bit(LSB), Modified Booth Multiplier, Vlsi Design.

I. INTRODUCTİON

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 order M can be expressed as

$$y[n] = \sum_{i=0}^{M-1} a_i x[n-i] \qquad (1)$$

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, [7] direct form and transposed form. In the direct form, 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 signal $x[n-i]$ and coefficientsa_i. In the transposed form, the operands of the multipliers in the MCM module are the current input signal $x[n]$ and coefficients. The result of individual constant multiplications goes through carry save adder (CSA) and delay elements. In the design of FIR filter the multiplier play a major roll and thus occupies more area and power.The multiplier is the slowest component in the system which determines the performance of the system. Multiplier has big issue on optimization of area and speed.

Basically multiplication of n x n bit width will produce the result as 2n bit width product. The system which are used for the [4] [1] digital signal processing (DSP) and multimedia application require output n bit width as same as the input n bit width. The fixed bit width [2] [6] result for multiplier is obtained by truncating the n most significant bit of 2n bit product result. They are rounded to n bits to avoid word size growth. The carry from n least significant bit of 2n bit product result is propagated to n most significant bit of 2n bit product result,In order to reduce the truncation error. The extra [7] compensation bits are added at the retained bit position to reduce the error due to non-generation of initial partial product and deletion error.

The hardware efficient truncated multiplier design [1] can be obtained by two methods: variable and constant correction. These methods reduce the rounding error which produced during truncation process. Since each product bit has an equal probability of being one or zero. By calculating average truncation error and then adds the row of constant into partial product bit matrix to compensate the error. The different [8] compensation techniques are used to reduce error in previous paper truncated multiplier design.

In this paper, low-cost FIR filter structure is designed based on area efficient truncated multiplier which produces the truncated error less than 1 unit of last position (ulp) comparing to previous multiplier. The multiplier is designed using [4] [8] [3] modified booth algorithm which is used for signed bit multiplication. The number of partial products generated is reduced by a factor of 2 through multiplier recording and jointly considering the partial product reduction, truncation, rounding of partial product bit matrix. So the final product meets the precision required. This proposed multiplier reduces the number of half adders and full adders used for accumulation of final product. The dynamic power consumption of the multiplier also reduced. Hence it reduces the area and power of multiplier which in turns leads to low-cost FIR filter design.

II. FIR FILTER DESIGN AND IMPLEMENTATION

Specifically there are three stages in designing the FIR filter:

- Finding the filter order and coefficient,
- Coefficient quantization,
- Hardware optimization.

In the [7] first stage the filter order M are determined in order to satisfy the frequency response specification and coefficient are obtained by MATLAB build-in remez() function. In the second stage Coefficient quantization, the coefficient which is generated in first stage is quantized to finite bit width accuracy. Hardware optimization stage is mainly concentrate on multiplier of FIR filter design because it consumes more power and area during partial product generation, accumulation of partial product to final product. Finally hardware optimization is obtain by enhancing various methods like CSE, MCMA and truncated multiplier which leads to the filter design in less area cost.

III. OVERVIEW OF MODIFIED BOOTH ENCODING MULTIPLICATION

In parallel multipliers, high performance, low power dissipation and speed can be achieved by using [4] modified booth algorithm, which decreases the number of partial products by a factor of 2 through multiplier recording.

Multiplier booth algorithm is used for both signed and unsigned bit multiplication. To obtain the fixed bit width size the n x n multiplication are truncated to n most significant product bit by deleting some of the minor least significant partial product bits of 2n bit output product.

Consider multiplication of two sign n bit number A and B represented in equation:

$$A = -a_{n-1} + \sum_{i=1}^{n-1} a_{n-1-i} 2^{-i} \qquad (2)$$

$$B = -b_{n-1} + \sum_{i=1}^{n-1} b_{n-1-i} 2^{-i} \qquad (3)$$

By modified booth encoding, B can be expressed as in equation

$$B = \sum_{i-0}^{((\frac{n}{2})-1)} b'_{\frac{n}{2}-1-i} 2^{-i(2i+1)} \qquad (4)$$

Where

$$b' = -2b_{2i+1} + b_{2i} + b_{2i-1} \qquad (5)$$

Modified booth multiplication consists of three major steps:

- Recoding and generating partial products.
- Reducing the partial product by partial product reduction schemes to two rows.
- Adding the final two rows obtained by partial product reduction using a carry-propagation adder to obtain the final product.

For modified booth recording, at least three signals are needed to represent the digit set {-2, -1, 0, 1, 2}. Many different ways have been developed and Table 1 shows the encoding scheme of multiplier bits. Based on MB algorithm, the logic that defines encoder and decoder outputs [3] are given by equation

$$neg_i = b_{2i+1} AND (b_{2i} NAND b_{2i-1}) \qquad (6)$$

$$one'_i = b_{2i} XOR b_{2i-1} \qquad (7)$$

$$two'_i = (b'_{2i+1} AND b_{2i}) NOR (b'_{2i} AND b_{2i+1} AND b'_{2i-1}) \qquad (8)$$

$$n_{aj} = a_j XOR b_{2i+1} \qquad (9)$$

$$p_{ij} = (n_{aj} OR one'_i) NAND (n_{aj-1} OR two'_i) \qquad (10)$$

Thus, the number of partial product can be reduced by a factor of 2 using MB recoding.

This algorithm is based on the fact that fewer partial products need to be generated for the group of consecutive ones and for a group of consecutive zeros, there is no need to generate any new partial product. In our proposed method the generated partial products are truncated to fixed bit width and the compensation bits are added at the appropriate position to reduce the error due to truncation.

Table. 1 Truth Table of MBE

b_{2i+1}	b_{2i-1}	b_{2i}	operation	neg_i	two_i	one_i	p_{ij}
0	0	0	+0	0	0	0	0
0	0	1	+A	0	0	1	a_j
0	1	0	+A	0	0	1	a_j
0	1	1	+2A	0	1	0	a_{j-1}

1	0	0	-2A	1	1	0	a'_{j-1}
1	0	1	-A	1	0	1	a'_j
1	1	0	-A	1	0	1	a'_j
1	1	1	-0	0	0	0	0

IV. TRUNCATED MULTIPLIER USING MODIFIED BOOTH MULTIPLIER

Modified booth multiplier is the fastest parallel multiplier where it reduces the number of partial product by factor two. Implementation of truncated multiplier design in modified booth algorithm provides more area reduction.

A faithfully rounded truncated multiplier design [7] is presented where the maximum absolute error is guaranteed to be no more than 1 unit of least position (ulp). The Existing work (Shen-Fu Hsiao) jointly considers the deletion, reduction, truncation, and rounding of partial product bits in order to minimize the number of full adders and half adders during accumulation**.** In order to reduce the half adder and full adder some of bits are omitted ($2^2+2^3+2.2^4+2.2^5$) in minor Least Significant area which does not provide significance in the final output. At bit position (2^7) the compensation bit is added where the effect of omitted bit is refined and the carry from the major Least Significant segment is propagated to Most Significant segmentst in order to reduce the error.

The deletion step in the algorithm removes all the unnecessary PP bits that do not need to be generated. Due to the deletion of unnecessary PP bits deletion error may found, this can be reduced by injecting a correction bias constant of 1/4 ulp, as shown by1c-omsin Figure 4.1.

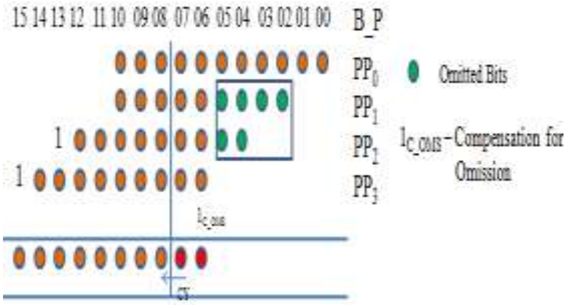

Figure 4.1 Existing Truncated Multiplier

The deletion of PP bits starts from column 1 by skipping the first rows of PP bits because after applying reduction, the resultant one row will be removed in the subsequent truncation and rounding processes. After the deletion of PP bits, we perform the per-column reduction is performed. After reduction, we perform the truncation is performed that further removes the first row of n − 1 bits from column 1 to column n–1. This step of truncation introduces truncation error.The truncation error can be reduced injecting another bias constant at bit position (2^7). After deletion, reduction, and truncation, the PP bits are added using a CSA to generate the final product of P bits, as shown in Figure 4.1. The bit at column after the final CSA is also removed during the rounding process. Thus, the total error for the design of the faithfully rounded truncated multiplier is bounded by $-ulp< E = (E_D+ E_T+ E_R) \leq ulp$ as shown in Equation (14). The truncated multiplier design achieves faithful rounding because the total error is no more than 1 ulp.The error ranges of deletion, truncation, and rounding before and after adding the compensation constants are given in equation (11), (12) ,(13) & (14).

$$-\frac{1}{2}\mathrm{ulp} \le E'_D \le 0 \quad -\frac{1}{4}\mathrm{ulp} \le E_D = E'_D + \frac{1}{4}\mathrm{ulp} \le \frac{1}{4}\mathrm{ulp} \quad (11)$$

$$-\frac{1}{2}\mathrm{ulp} \le E'_T \le 0 \quad -\frac{1}{4}\mathrm{ulp} \le E_T = E'_T + \frac{1}{4}\mathrm{ulp} \le \frac{1}{4}\mathrm{ulp} \quad (12)$$

$$-\frac{1}{2}\mathrm{ulp} \le E'_R \le 0 \quad -\frac{1}{4}\mathrm{ulp} \le E_R = E'_R + \frac{1}{4}\mathrm{ulp} \le \frac{1}{4}\mathrm{ulp} \quad (13)$$

$$-\mathrm{ulp} \le E = (E_D + E_T + E_R) \le \mathrm{ulp} \quad (14)$$

V. PROPOSED AREA EFICIENT TRUNCATED MULTIPLIER

In the proposed Area Efficient Truncated Multiplier design final product is truncated to n bits in nxn signed multiplication using Modified Booth Algorithm and is represented in Figure 5.1. This area efficient truncated multiplier involves two major steps:

- Non-generation of initial partial products since it does not contribute significant for final product.
- Omitting some minor LSB bits since it reduce the number of half adder and full adder during accumulation.

The compensation bits are added in a particular place for the non-generation and omitted bits in order to reduce truncation error. And also addition of carry from the major LSB bit into the MSB bit reduces the truncation error.

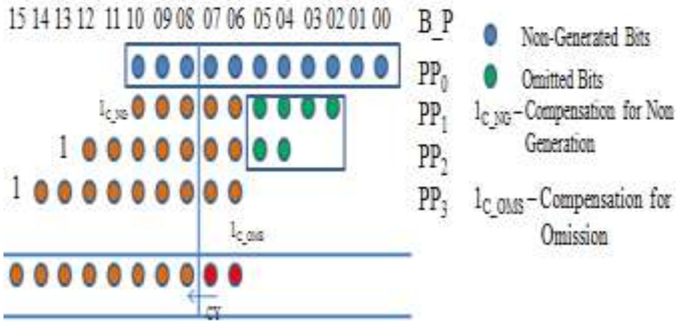

Figure 5.1 Proposed Truncated Multiplier

In the next step, the PP matrix is compressed using HAs and FAs with carry generated in LS being propagated to the MS. In the final step, the sum out of retained bits in LS is truncated and the final product is rounded to n bits. Implementation of area efficient truncated multiplier reduces the hardware required to initial partial product generation which contributes for major area reduction in the proposed multiplier. Also the use of booth algorithm for partial product generation reduces the number of partial product rows by half using two's complement conversion. As two's complement conversion involves sign bits in the partial product rows, the position of placing compensation bit for non-generation pp0 bits reduces the non-generation error. The implementation of the algorithm form n = 8 is shown in Figure 5.1.

VI. REALIZATION OF FIR FILTER USING AREA EFFICIENT TRUNCATED MULTIPLIER

The area efficient truncated multiplier is effectively used in the design of transposed form FIR filter structure. In previous FIR filter structure, normal multiplication of input and coefficient is performed without considering the length. Normal implementation of FIR filter structure will result in large area consumption. This can be overcome by replacing anarea efficient truncated multiplier in FIR filter structure so the area cost can be reduced significantly.

The weighted sum of input sequence in FIR filter structure is called as convolution sum which used to implement various filter structure which satisfy their respective frequency response. The area, power and delay in FIR filter is directly proportional to filter order. If the filter orders can be changed dynamically, the number of multiplier used in FIR filter will reduced to area of multiplier so the filter will occupies less area. The power and delay will reduce due to reduction in area compare to conventional design.

VII. RESULTS AND DISCUSSION

The design is simulated and synthesized using XILINX ISE tool. The coefficients are calculated using remez() function in MATLAB and stored in the ROM as they are fixedin simulation tool. The gate count, delay, and power are calculated using XILINX ISE tool.

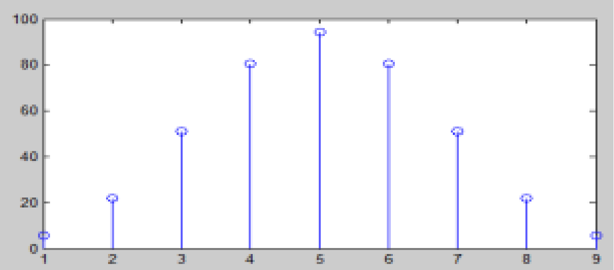

Figure 7.1 Coefficient for 9-tap FIR Filter

To verify the functionality of the proposed truncated MBE multiplier, it is implemented in a 9 tap Finite Impulse Response (FIR) filter. The Filter coefficients are obtained from MATLAB.

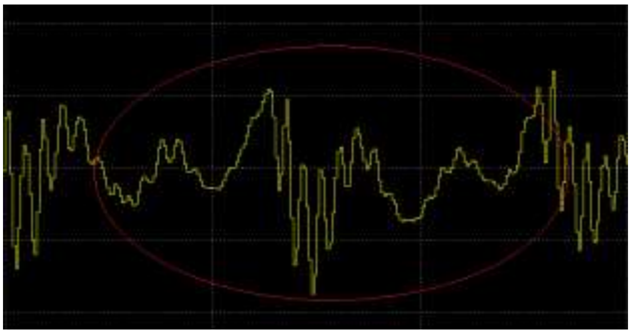

Figure 7.2 Output Waveform of Speech Signal

The input data are sampled (speech signal) as 8 bits with coefficients represented in 8 bits. The multiplication of input sample and filter coefficients are realized with the proposed truncated MBE multiplier with the output maintained as 8 bit samples.

The output of FIR filter implemented with Standard, Shen-Fu Hsiao(2013) et al., multipliers for comparison. The output waveforms of the proposed and systems used for comparison are shown in Figure 7.3.It is seen that the proposed truncated MBE multiplier shows close performance as that of non-quantized standard multiplier and better performance compared to[7] Shen-Fu Hsiao et al (2013)multiplier.

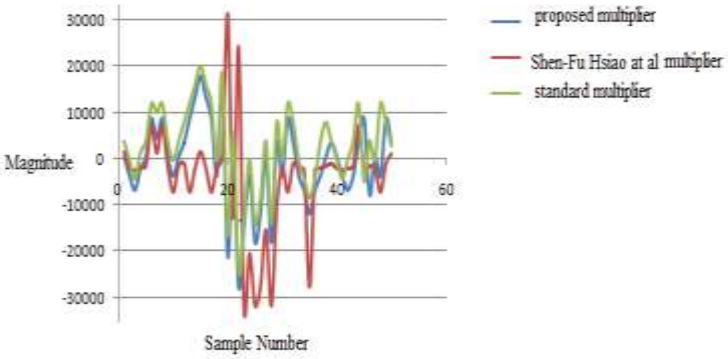

Figure 7.3 Output Waveform of FIR Filter Implementation with Standard, Shen-Fu Hsiao et al and Proposed Multiplier

Table II. Comparison of Conventional Multiplier Parameter

Multipliers	Area (gate count)	Delay (ns)	Power (mW)	Power*delay (mWns)
Standard Multiplier	888	28.67	29.69	851.2
Existing Multiplier	750	24.29	31	752.99
Proposed multiplier	702	20.12	16	332.05

VIII. CONCLUSION

The proposed work developed a generalized truncation multiplier for fixed width multiplication, which is implemented in MBE multiplier for signed bit multiplication. By developing suitable compensation functions for the errors introduced by non-generation and omission of few partial product bits to truncate the final product, the proposed Area Efficient Truncated Multiplier achieves better performance in terms of absolute error comparedto the state-of-the-art designs and maintains maximum error within 1 unit of LSB of truncated product.

REFERENCES

[1] Chip-Hong Chang and Ravi Kumar Satzoda,“Low-error and high performance multiplexer-based truncated multiplier,” IEEE Transaction on very large scale integration systems, vol. 18. No. 12, December 2010.

[2] Jer Min Jou, Shiann Rong Kiang, and Ren Der Chen “ Design of low-error fixed-width multipliers for DSP applications,” IEEE Transaction on circuits and systems-II: analog and digital signal processing, vol. 46. No. 6, June 1999.

[3] Jiun-Ping Wang, Shiann-Rong Kuang and Shish-Chang Liang "High-accuracy fixed-width modified booth multipliers for lossy applications," IEEE Transaction on very large scale integration systems, vol. 19. No. 1, January 2011.

[4] Kyung-Ju Cho, Kwang-Chul Lee, Jin-Gyun Chung, and Keshab K.Parhi, "Design of low-error fixed-width modified booth multiplier," IEEE Transaction on very large scale integration systems, vol. 12. No. 5, May 2004.

[5] Lan-Da Van, and Chih-Chyau Yang "Generalized Low-error area efficient fixed-width multipliers," IEEE Transaction circuits and systems, vol. 52. No. 8, August 2005.

[6] Nicola Petra, Davide De Caro, Valeria Garofalo, Ettore Napoli and Antonio Giuseppe Maria Strollo,"Design of fixed-width multipliers with linear compensation function," in IEEE transaction on circuits and system –I:regular papers, vol. 58, no.5, May 2011.

[7] Shen-Fu Hsiao,Jun-Hong Zhang Jian, and Ming-Chih Chen "Low-cost FIR filter designs based on faithfully rounded truncated multiplier constant multiplication/accmulation," in IEEE transations on circuits and systems-II:express briefs, vol.60, no. 5, May 2013.

[8] Tso-Bing Juang and Shen-Fu Hsiao "Low-error carry-free fixed –width multipliers with low-cost compensation circuits," IEEE Transaction on circuits and systems-II:express briefs, , vol. 52. No. 6, June 2005.