A NOVEL MULTIPLEXER BASED TRUNCATED ARRAY MULTIPLIER

Chip-Hong Chang, Ravi Kumar Satzoda and Swaminathan Sekar

Centre for High Performance Embedded Systems, Nanyang Technological University,
50 Nanyang Drive, Research Techno Plaza, 3rd Storey, Border X Block, Singapore 637553.

ABSTRACT

The design of high-speed, area-efficient and low power multiplier is essential for the VLSI implementation of DSP systems. In many applications, like digital filtering, the inputs are contaminated by noise and precise outputs are often not required. It has been shown that the area and power of multiplier can be significantly reduced by truncation techniques at the expense of truncation errors. This paper presents a novel multiplexer based truncated array multiplier, which has leveraged and improved upon three existing truncation algorithms. An exhaustive error analysis was also performed to evaluate the truncation errors of the new truncated multiplier. The proposed truncated multiplier was compared to one implemented with the standard truncation schemes in latency, silicon area and power dissipation. Simulation results have attested the accuracy and VLSI performance ascendancy of the proposed truncated multiplier.

1. INTRODUCTION

Digital multiplication is a key operation in digital signal processing algorithms like digital filtering, linear transformation, correlation and wavelet compression, etc. [3]. Various efficient array and parallel multipliers have been proposed and many of them boost the speed of multiplication at the cost of large VLSI area and high power dissipation. Several techniques to eliminate spurious computations have been developed to minimize power consumption [5][6]. These techniques have been applied to full-width multiplier in general. As wordlength increases, the hardware complexity of a full-width multiplier exacerbates. However, in most signal processing applications, full-width multiplication is not necessary and the products are usually rounded to avoid the growth of word size [3][7]. Truncated multiplication is a good alternative to reduce power dissipation, delay and more remarkably, the silicon area. In truncated multiplication, a portion of the multiplier matrix involving the less significant partial product bits is eliminated and additional gates are added to compensate for the errors resulting from the removal of a partial multiplier matrix.

Several techniques have been proposed to improve the accuracy of truncated multiplication. Yoshida et. al. [8] proposed a pseudorandom rounding for an n-bit truncated multiplier wherein the least significant n columns of the partial products are simply eliminated, and the least significant bit (LSB) of the result is replaced with a logic sum of the original nth and $(n+1)$th bits. Although this method saves considerable hardware, the magnitude of the truncation error is very large. In general, it is sufficient to compute only the sum of $n+k$ most significant columns of the partial product terms to estimate the full-width output. Two sources of errors are introduced in this process: reduction error and rounding error. Reduction error occurs because the $n-k$ least significant columns of the multiplication matrix are not used to compute the product, while rounding error is a consequence of rounding the product to n-bits. Thus, a correction constant, C is added to the truncated partial product to compensate for these two errors. A systematic method for selecting the value of C was proposed [7] to minimize the average and mean square errors. Since the correction constant is fixed for specific values of n and k regardless of the value of the multiplicand and multiplier, this scheme is called constant correction truncation (CCT). A non-zero DC component of the resulting product is incurred by this fixed correction constant. To adapt the correction to the input values, a variable correction truncation (VCT) scheme was proposed in [4]. The correction term varies depending on the number of logic ones in the $(n-k-1)$th column. Compared to CCT, the maximum error, the mean error and the standard deviation of the error are reduced. The drawback is that it requires more hardware. Recently, a hybrid correction truncation (HCT) technique [9] has been proposed, which uses both constant and variable correction techniques to reduce the overall error. A new parameter is introduced to modulate the percentage of variable correction to be applied. Since a CCT multiplier has a maximum absolute error when the truncated partial product matrix is maximum and a VCT multiplier has a maximum absolute error when the truncated partial product matrix is close to zero, HCT multiplier achieves a lower average and maximum absolute errors compared to the CCT and VCT multipliers [9].

In this paper, a novel multiplexer based truncated multiplier founded on the full-width multiplier architecture of [1], [2] is presented. The fundamentals of multiplexer based multiplier and the truncation schemes [3], [9] are reviewed in Section 2 followed by a detailed description of the proposed truncated multiplier architecture in Section 3. A new truncation scheme to offset the truncation error is also proposed. This truncated multiplier is prototyped and compared with conventional truncated array multiplier under the same truncation scheme in terms of dynamic power dissipation, delay and area. In addition, their truncation errors are evaluated against the schemes proposed in [3], [9].

2. PRELIMINARIES

Two unsigned n bit numbers X and Y, and their product P can be represented by the following equations.

$$X = x_{n-1}x_{n-2}....x_0 = \sum_{i=0}^{n-1} x_i 2^i \quad (1)$$

$$Y = y_{n-1}y_{n-2}....y_0 = \sum_{i=0}^{n-1} y_i 2^i \quad (2)$$

$$P = XY = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} x_i y_j 2^{i+j} \quad (3)$$

(3) can be reorganized as follows:

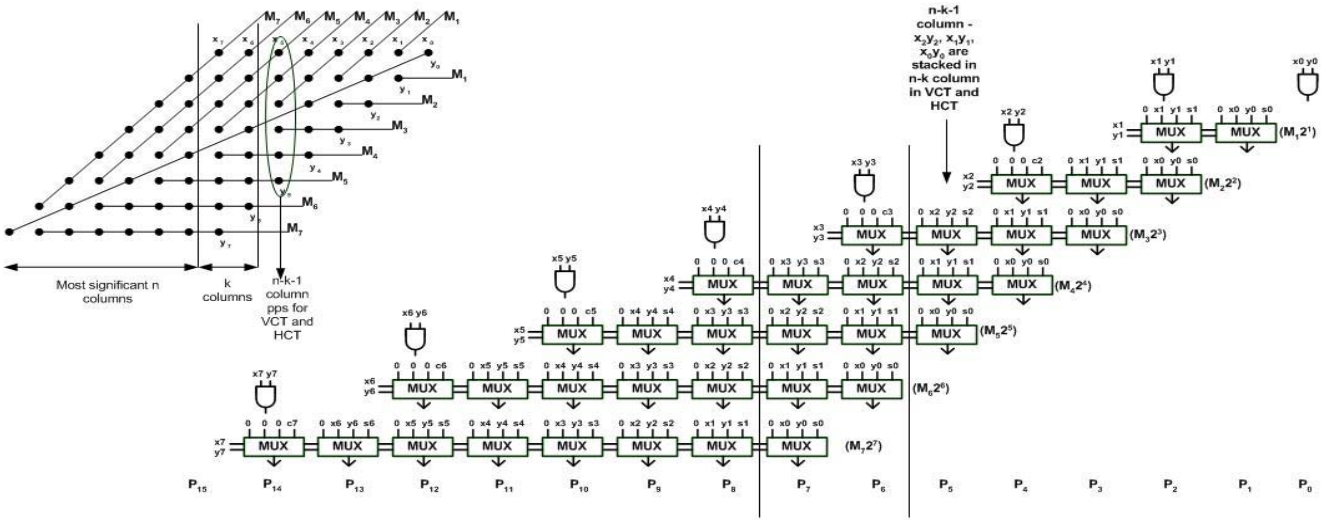


Figure 1. Partial product dot matrix and the corresponding multiplexer based architecture

$$P = \sum_{i=0}^{n-1} x_i y_i 2^{2i} + \sum_{i=1}^{n-1} M_i 2^i \qquad (4)$$

where $M_i = x_i(y_{i-1}y_{i-2}...y_0) + (x_{i-1}x_{i-2}...x_0)y_i$

M_i can be realized as a multiplexer with x_i and y_i as select signals. For n = 8, the partial product (pp) matrix of the above multiplication is shown on the upper left hand corner of Fig. 1. Each slanted line passes through the partial products (pp's) with the same x_i and each horizontal line passes through the pp's with the same y_i. The remaining 8 dots lying on the major diagonal represent the pp's, x_iy_i for i = 0, 1, …, 7. M_i is the multiplexed result (selected by x_i and y_i) of the pp's given by the two lines symmetrical about x_iy_i. This pp matrix can be mapped to the hardware architecture to its right in Fig. 1 [1], [2]. The bits from each multiplexer is added to the next stage using full adders (FAs) and half adders (HAs).

In the implementation of truncated multiplier, the hardware generating the least significant n bits of the 2n-bit product is ignored but some hardware is added to round the product to the nearest value representable in limited precision. Three efficient truncation schemes, namely Correction Constant Truncation (CCT), Variable Constant Truncation (VCT) and Hybrid Constant Truncation (HCT), have been described in [3] and [9]. Analysis of truncation and rounding errors, and their hardware compensation methods can be found in [7]. These schemes are briefly described here with the aid of Fig. 1.

In CCT, (n+k) most significant columns in the pp matrix are used to produce the most significant n bits of the product, where k is the number of extra columns used to calculate the compensation for the truncation error. The error produced by omitting n–k columns is diminished by adding a correction constant to the (n–1)th to (n–k)th columns. The constant is determined by exhaustive evaluation of the errors for different values of inputs for each k. In VCT, the pp's in (n–k–1)th column are used for the compensation. All the pp's in (n–k–1)th column are stacked over the (n–k)th column as shown in Fig. 2. In HCT, selected pp's in the (n–k–1)th column determined by a constant factor p are stacked over the (n–k)th column. Besides the additional pp's in VCT and HCT, correction constants are added to the (n–1)th to (n–k)th columns. These constants are fixed as '1' in [3], as shown in Fig. 2. In what follows, we will improve the truncation schemes to reduce the hardware overhead and we show that this can in fact be achieved at a reduced truncation error.

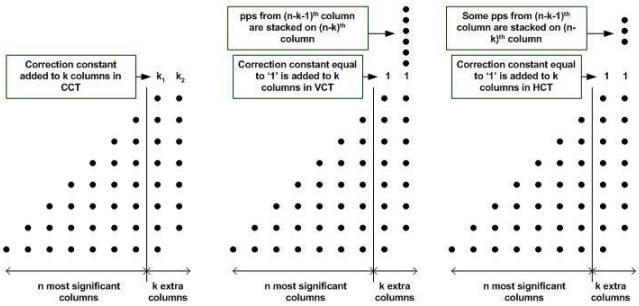


Figure 2. Dot matrix representations of CCT, VCT and HCT

3. NEW TRUNCATED MULTIPLIER

An ingenious truncated multiplication scheme that suits the aforementioned multiplexer based array multiplier architecture is proposed in this section. For ease of exposition, it is exemplified with n = 8 and k = 2 and unsigned mulitplication is assumed. However, the architecture can be easily extended to deal with signed truncated multiplication.

From Fig. 1, besides the n most significant columns, the pp's in the (n–1)th and (n–k)th columns are also retained to account for the n–k columns that have been discarded. Thus, M_7, M_6, M_5 and M_4 in Fig. 1 will be kept and all other multiplexers can be discarded. Moreover, only the pp's in k columns of M_5 and M_4 need to be preserved in the truncated product matrix. Without incorporating any truncation scheme, almost halve the hardware of full-width multiplier can be saved.

The implementation of CCT scheme is straightforward. It involves the addition of a correction constant in the (n–1)th to (n–k)th columns. For n = 8 and k = 2, this constant is '11' as determined by exhaustive iteration. In the original VCT proposed in [3], the pp's in the (n–k–1)th column are stacked over the pp's of the (n–k)th column. This causes the hardware overhead to

exacerbate as n increases. Our proposed new truncation scheme mitigates this overhead by using only the carry signals generated from the $(n–k–1)$th column instead of all the pp's of the column.

There are 6 pp's, two in each M (M_5, M_4 and M_3), in the $(n–k–1)$th column of Fig. 1. Consider the two pp's, x_5y_0 and y_5x_0 of M_5. In [3], these two partial products are stacked in the $(n–k)$th column. With the multiplexer based array multiplier, the select signals x_5 and y_5 select either x_0 or y_0 or 0 (when both select signals are 0) or the sum $x_0 + y_0$ (when both select signals are 1). In the worst case when both select signals are '1', instead of both pp's, x_5y_0 and y_5x_0 being stacked in the $(n–k)$th column, the ANDed output of x_0 and y_0, which is the carry from $(x_0 + y_0)$, is stacked. In HCT, only selected pp's in the $(n–k–1)$th column are considered for stacking according to a modulating factor p.

Besides the correction hardware to compensate for the elimination of the least significant $n–k$ columns, a constant to approximate the rounding error in VCT and HCT schemes is also added. In [3], this is done by adding '1's in the $(n–1)$th to $(n–k)$th columns. Our proposed truncated multiplier adds 1 to only the nth column. It should be noted that the overall error performance will not be jeopardized by this hardware simplification.

The multiplexers shown in Fig. 1 can be rearranged to generate the parallel architecture of the proposed truncated multiplier based on the VCT scheme. The implementation for $n = 8$, $k = 2$ is shown in Fig. 3. The structure resembles a carry save adder (CSA) tree. The sum signal is added to the multiplexed signal from the next row of the same column while the carry signal is input to the adder in the next column. Only two types of cells, FMUX and SA are defined in the full width array multiplier [1][2]. FMUX comprises a multiplexer driving a full adder. SA consists of two full adders (FAs) and an AND gate. SA adds the signals from the AND gates (see Fig. 1) to the signal from the multiplexers below it in the same column. For truncated multiplication, a number of such cells have at least one constant "0" input, making some of their resources redundant. Those FMUXs with a constant "0" input are substituted by HMUXs where a FA is replaced by HA. Besides, some of the outputs in [1] are superfluous. For instance, the sum outputs of the SA cells towards the left boundary are known to be zero and ignoring them has no effect on the final result. Therefore, SA cells at the left boundary are substituted by MA cells. An extra OR gate is added to the left of the last FA to take care of the possible carry-out signal generated from the MA cell. It is impossible to have a carry out from the lower FA in SA at the left boundary of the full-width multiplier, but in the proposed truncated multiplier, adding "1" to the HMUX cell at the first row may generate a carry out from MA for certain input combinations. TA is similar to SA except that a FA has been replaced by HA. The AND (A) gates generate the carry signals from the pp's of the $(n–k–1)$th column. At each stage, two CSA columns will complete their computations. Therefore, a 2-bit carry lookahead adder (CLA) is used to synchronize the summation for one stage of operation, while lowering the logic complexity of the entire circuit. CGEN is a 2-bit CLA without the sum output circuit. In the k extra columns, the product bits are not required and CGEN is used to compute only the carry signal required for P_8. The FOA cell is driven by MA. It consists of a FA and an OR gate. The FA generates P_{14} and the OR gate generates P_{15} from the carries of FA and MA.

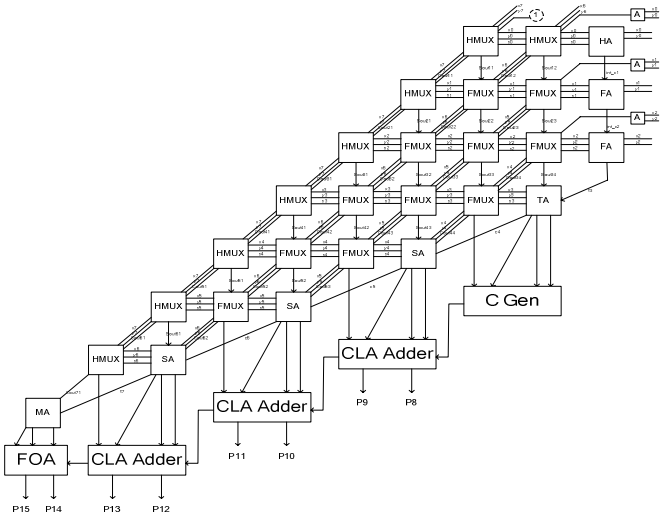


Figure 3. Truncated multiplier for $n = 8$, $k = 2$ with VCT.

3. EXPERIMENTAL RESULTS

3.1 Truncation Error Analysis

An exhaustive MATLAB simulation was performed by considering all input combinations of an 8×8 bit multiplier. Error statistics like average error (E_{ave}), variance (σ^2), absolute maximum error ($|E_{max}|$), maximum positive error (E_{max_p}) and maximum negative error (E_{max_n}) of conventional and proposed multiplexer based truncated array multipliers were measured relative to the results of full width multiplication. Fig. 4 shows the absolute maximum error $|E_{max}|$ generated from conventional and proposed truncated multipliers for different truncation schemes. Four different truncation lengths, $k = 1$, 2, 3 and 4 were simulated. From Fig. 4, $|E_{max}|$ of the proposed truncated multiplier are smaller than those of the conventional multiplier by around 55% in CCT and HCT and 40% in VCT. Fig. 5 shows their error histograms for three truncation schemes with $n = 8$ and $k = 2$ and $p = 0.3$ for HCT. Considerably narrow spreads of well defined normal distributions were obtained for the proposed truncated multiplier. Conversely, twin-peak and dispersive error spreads were observed in the conventional truncated array multiplier, leading to greater computational uncertainty.

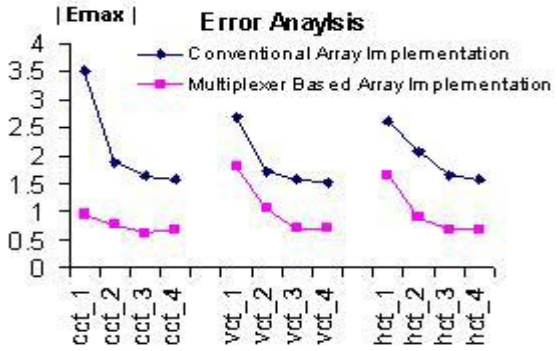


Figure 4. $|E_{max}|$ for different truncation schemes

3.2 Hardware Simulation Results

The proposed and conventional 8×8 truncated array multipliers were modeled using structural HDL and synthesized by

Synopsys Design Compiler in TSMC 0.18μm CMOS standard cell technology to determine their VLSI area, dynamic power dissipation and latency. The optimization was delay driven with the input and output loads set to 0.8pF and 0.9pF, respectively. The power simulations for all the designs were performed at a clock frequency of 20MHz at 1.8V. Table 1 shows the simulation results obtained for different schemes and truncation lengths. It is found that the proposed multiplexer based multiplier outperforms the conventional array multiplier in all performance metrics for any value of *k*. The improvements are more prominent in VCT and HCT schemes attributable to the stacking of only the carry signals in the new truncation scheme.

Table 1. Synthesis results (a) proposed (b) conventional truncated array multipliers

	Delay (ns)		Area (μm^2)		Power(μW)	
	(a)	(b)	(a)	(b)	(a)	(b)
VCT_1	4.34	4.67	5980.86	6719.32	450.93	463.07
VCT_2	4.61	4.94	6902.28	6962.15	466.77	471.88
VCT_3	4.44	5.27	7081.90	8083.15	468.87	480.77
VCT_4	4.52	5.2	7584.19	7630.76	476.72	482.75
CCT_1	4.35	4.58	5997.49	5891.05	450.46	451.62
CCT_2	4.61	4.78	6729.31	6559.66	462.43	464.69
CCT_3	4.44	4.64	8106.43	8153.00	468.12	474.20
CCT_4	4.45	4.66	7491.05	8096.45	474.99	481.24
HCT_1	4.42	4.66	6093.96	6110.59	453.23	456.26
HCT_2	4.61	5.05	6729.30	6829.09	462.43	469.52
HCT_3	4.47	5.22	7035.33	7367.97	468.06	477.42
HCT_4	4.52	5.34	7497.70	7733.88	474.69	482.56

4. CONCLUSION

Truncated multipliers have been widely employed to reduce hardware complexity and power consumption with the trade off in result accuracy. In this paper, a multiplexer based truncated array multiplier architecture is devised with a new truncation scheme. The circuit carries out synchronous computation of the partial sums of two operands in a regular iterative array structure. Compared to the original circuit presented in [1] for full-width multiplication, superfluous circuitry has been eliminated and optimized to cut down the spurious computations. Exhaustive error analysis and VLSI area-time-power evaluation have been performed on the proposed and conventional truncated multipliers for three different truncation-and-rounding schemes, CCT, VCT and HCT. The results show that proposed truncated multiplier surpasses the conventional design in terms of accuracy, area, power and delay.

5. ACKNOWLEDGEMENT

This research is supported in part by the Panasonic Singapore Laboratory-Nanyang Technological University Joint R&D Collaboration (No. M48900001).

6. REFERENCES

[1] K. Z. Pekmestzi, "Multiplexer-based array multipliers," *IEEE Trans. on Computers*, vol. 48, no. 1, pp. 15-23, Jan. 1999.

[2] Y. Wang, Y. Jiang and E.Sha, "On area-efficient low power array multipliers," in *Proc. 8th IEEE Int'l Conf. on Electronics, Circuits and Systems*, vol. 3, pp 1429-1432, Sep. 2001.

[3] M. J. Schulte, J. E. Stine and J. G. Jansen, "Reduced power dissipation through truncated multiplication," in *Proc. IEEE Alessandro Volta Memorial Workshop on Low-Power Design,* pp. 61-69, Mar. 1999.

[4] E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers", in *Proc. 35th Asilomar Conf. on Signals, Systems & Computers,* vol. 2, pp. 1178-1182, Nov. 1998.

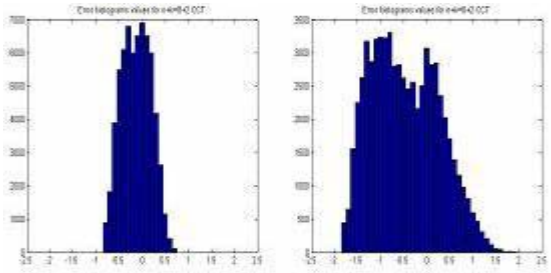
[5] C. Lemonds and S. S. Shetti, "A low power 16 by 16 multiplier using transistion reduction circuitry," in *Proc. Int'l Workshop on Low Power Design,* pp. 139-142, 1994.

[6] G. E. Sobelman and D. L. Raatz, "low-power multiplier using delayed evalutaion," in *Proc. Int'l Symp. on Circuits and Systems,* vol. 3, pp. 1564-1567, 1995.

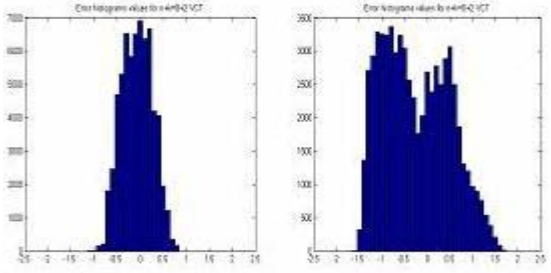
[7] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," in *Proc. IEEE Workshop on VLSI Signal Processing VI,* pp. 388-396, Oct. 1993.

[8] N. Yoshida, E. Goto and S. Ichikawa, "Pseudorandom rounding for truncated multipliers," *IEEE Transactions on Computers*, vol. 40, no. 9, pp. 1065-1067, Sep. 1991.

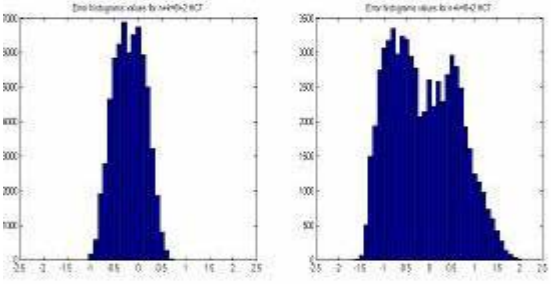
[9] J. E. Stine and O. M. Duverne, "Variations on truncated multiplication," in *Proc. Euromicro Symp. on Digital System Design,* pp. 112-119, Sep. 2003.



(a) CCT, $n = 8$, $k = 2$



(b) VCT, $n = 8$, $k = 2$



(c) HCT, $n = 8$, $k = 2$, $p = 0.3$

Figure 5. Error Histograms of proposed multiplexer based (left) and conventional (right) truncated array multipliers