Efficient Design of Fixed Width Multiplier using Truncation and Error Compensation

Aiman Tabassum[1], Prof. M. T. Hasan[2]
ME Student[1], Assistant Professor[2]
Department of Electronics & Communication Engineering
Anjuman College of Engineering & Technology, Sadar, Nagpur, India

Abstract:
We propose a parallel fixed-width multiplier design that receive two n-bit numbers and produce a n bit product. To design the proposed fixed-width multiplier, three multiplication modules are used that can work as independent smaller-precision multiplications. In order to add the outputs of the multiplication modules, carry save adder and Brent-Kung adder is used which can further improve the performance of the design. Implementation results demonstrate that the proposed fixed-width multiplier with parallel multiplication modules achieve significant improvement in delay and power delay product when compared with previous architectures

Keywords: Y Fixed-width multiplier, carry-save adder, Brent-Kung adder , carry Generator

I. INTRODUCTION

We know as, nowadays of growing demands on mobile communications and portable equipment systems, the power-efficient and high speed multiplier plays an important role of very large-scale integration (VLSI) systems. With advances in technology, many researchers have come up with various multipliers such as Baugh-Wooley, Dadda, Wallace tree and Booth multiplier. In many digital signal processing (DSP) algorithms such as digital filters, discrete Fourier transform (DFT), discrete cosine transform (DCT), single precision results are required in order to prevent the result from growing in size after every multiplication.

By directly truncating nbit Least Significant Bit (LSB) output a fixed-width multiplier (single precision) can be achieved, that produces n-bit output product with n-bit multiplier and n-bit multiplicand. However, truncating the LSB part of the multiplication product leads to large truncation errors (sum of the reduction and rounding). In order to mitigate this truncation error, many error compensation circuits are designed with less area. Limit first utilized statistical techniques for compensating the error introduced by truncating the LSB part.

Eliminated LSB part and is substituted, the design is further improved based on truncating the LSB columns and rounding the result to columns. However, these approaches introduces a large error as the width of the n increases rapidly, resulting impractical in most applications. For a fixed width multiplier better accuracy is obtained by using adaptive techniques instead of constant correction. In these techniques, a variable correction is employed that compensate the effect of the eliminated terms with a non-constant compensation function which is used to estimate the weighted sum of LSB part and to reduce the output error. However, in today's applications one of the major challenges for high-performance DSP applications is the power consumption, both static and dynamic. Therefore, instead of targeting them independently, there is a need to find an optimum between speed and power.

This is represented by the average energy dissipated for one switching event which is known as power-delay product.

II. MULTIPLICATION MODULES MP1 AND MP2

The multiplication modules MP1 and MP2 are designed using Baugh-Wooley array algorithm. The detailed diagram of the corresponding modules MP1 and MP2 are exposed, where A, HA, and FA denote an AND gate, a half adder and a full adder, the logic diagram of AOR and AFA are depicted in fig. For MP1 module the inputs are $x[7:3]$ and $y[3:0]$ that generates partial product output as $p1[5:0]$. Similarly, for MP2 module the inputs are $x[3:0]$ and $y[7:4]$ that generates partial product output as $p2[5:0]$. These outputs are generated independently by using MP1 and MP2, which can improve the speed of the design.

III. MULTIPLICATION MODULES MP3

In general, applying the divide and conquer algorithm, we can partition an n-bit operand into two independent n/2-bit operands. Similarly, the input bits $x[7:4]$ and $y[7:4]$ of the module MP3 are partitioned into two independent 2-bit operands as $xh=x[7:6]$ and $xl=x[5:4]$ for x and $yh=y[7:6]$ and $yl=y[5:4]$ for y, where h and l represents higher and lower order bits of x and y. These operands are computed in parallel using four 2×2 multiplier blocks that can generates four partial product outputs $q0[3:0]$, $q1[3:0]$, $q2[3:0]$ and $q3[3:0]$. The internal logic diagram of 2×2 multiplier block contains AND gates and XOR gates as shown in Fig. 4. The first two final product outputs of this module $p3[0]$ and $p3[1]$ are same as that of the partial products $q0[0]$ and $q0[1]$. The remaining product terms $p3[7:2]$ are obtained by using carry save adder and carry propagate adder as shown. Thus, the final product of this module is obtained in a three level computation that can improve the speed of the design.

IV. CONCLUSIONS

This paper presents a parallel fixed-width Baugh-Wooley multiplier that contains three multiplication modules

MP1,MP2 and MP3 which can computes in parallel to generate partial product outputs. These outputs are summed by using carry save adder and Brent-Kung adder, which improves the speed of the design in order to obtain the final product. For n = 8 and h = 2 of a fixed-width multiplier, the implementation results showed that the proposed multiplier can achieve 11.81% and 28.20% improvement over earlier reported fixed-width multipliers in terms of delay. The proposed design also improves power-delay product compared to other fixedwidth multipliers

V. REFERENCES

[1].K. Pahri, "Vlsi digital signal processing systems," John Wiley& Sons, 1999.

[2]. P. Bougas, P. Kalivas, A. Tsirikos, and K. Z. Pekmestzi, "Pipelined array-based fir filter folding," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 52, no. 1, pp. 108–118, 2005.

[3]. H.-Y. Lee and I.-C. Park, "Balanced binary-tree decomposition for area-efficient pipelined fft processing," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 54, no. 4, pp. 889–900, 2007.

[4]. Y. Lim, "Single-precision multiplier with reduced circuit complexity for signal processing applications," Computers, IEEE Transactions on, vol. 41, no. 10, pp. 1333–1336, 1992.

[5]. S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol. 43, no. 2, pp. 90–95, 1996.

[6]. E. Swartzlander, "Truncated multiplication with approximate rounding," in Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on, vol. 2. IEEE, 1999, pp. 1480– 1483.

[7]. J. M. Jou, S. R. Kuang, and R. Der Chen, "Design of low-error fixedwidth multipliers for dsp applications," Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol. 46, no. 6, pp. 836–842, 1999.

[8]. L.-D. Van, S.-S. Wang, and W.-S. Feng, "Design of the lower error fixed-width multiplier and its application," Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol. 47, no. 10, pp. 1112–1118, 2000.

[9]. L.-D. Van and C.-C. Yang, "Generalized low-error area-efficient fixedwidth multipliers," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 52, no. 8, pp. 1608–1619, 2005.

[10]. S.-R. Kuang and J.-P. Wang, "Low-error configurable truncated multipliers for multiply-accumulate applications," Electronics letters, vol. 42, no. 16, pp. 904–905, 2006.

[11]. N. Petra, D. De Caro, and A. G. Strollo, "Design of fixed-width multipliers with minimum mean square error," in Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference on. IEEE, 2007, pp. 464–467.

[12]. N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G. Strollo, "Design of fixed-width multipliers with linear compensation function," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 58, no. 5, pp. 947–960, 2011.

[13]. I.-C. Wey and C.-C. Wang, "Low-error and hardware-efficient fixedwidth multiplier by using the dual-group minor input correction vector to lower input correction vector compensation error," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 20, no. 10,pp.1923–1928,2012