

An Improved High-Frequency Sequential Decimal Multipliers Using Hybrid BCD Codes

Mohammed Abdul Naveed, Dr.Srinivasan

Research Scholar, Dept. of Electronics and Communication Engineering, Sri Satya Sai University of Medical Sciences,
Sehore, M.P, India

Associate Professor, Dept. of Electronics and Communication Engineering, Sri Satya Sai University of Technology and
Medical Sciences, Sehore, M.P, India.

ABSTRACT: Hardware implementation of decimal multiplication is considered as one of the most complicated dyadic operations, which requires high-cost hardware implementation. Therefore, the processor industry has opted to use the sequential decimal multipliers to reduce the high cost of parallel architectures. An improved high-frequency sequential decimal multipliers using hybrid BCD codes is proposed in this paper for the parallel architectures to improve performance by mainly exploiting the properties of three different binary coded decimal (BCD) codes, namely the redundant BCD excess-3 code (XS-3), the overloaded decimal digit set (ODDS) code and BCD-4221/5211 code; hence this design is referred to as hybrid. In this paper, a new decimal partial product reduction (PPR) tree is proposed; it consists of a binary PPR tree block, a nonfixed size BCD-4221 counter correction block and a BCD-4221/5211 decimal PPR tree block. A high-frequency sequential decimal multiplier is proposed whose cycle time is reduced to the latency of a binary half-adder plus that of a decimal multiply-by-two operation, which overall is less than that of a decimal carry-save adder.

KEYWORDS: Sequential Decimal Multiplication, Parallel architectures, BCD codes, partial product reduction (PPR) tree.

I. INTRODUCTION

Hardware implementation of decimal floating-point arithmetic has been gaining revived popularity during the last decade. However, the complexity of the decimal number system brings about more challenges to be dealt with during the hardware implementation, compared with the binary counterparts. With the intention to reduce the high cost of these decimal operations the processor industry has opted for sequential realization [1]. However, sequential hardware realization is at a disadvantage because of its high latency.

This paper focuses on the hardware realization of the decimal multiplication and a novel architecture is proposed to ameliorate the problem of slow sequential decimal multipliers via increasing the clock frequency. In this design, the cycle time is reduced to the latency of a binary half-adder (HA) plus that of a decimal multiply-by-two operation, which verall is less than that of a decimal carry-save adder (CSA). The claimed improvement is supported by synthesizing and comparing the proposed design to those of the best previous pertinent works.

A novel selection block can be designed to select appropriate easy multiples of the multiplicand as a partial product. A novel approach is proposed for the partial product accumulation (PPA) with the intention of increasing the speed of this crucial step. The critical path delay of a conventional PPA consists of two decimal CSAs, whereas the latency of the proposed PPA is reduced to less than that of a mere decimal CSA by retiming the constituent parts of a decimal CSA. In this paper, a decimal multiplier based on hybrid BCD codes is proposed; it uses the same coding method as used in [3] for PPG. However, the PP reduction tree is different; the proposed PPR tree consists of a binary PPR tree block, a nonfixed size BCD-4221 counter correction block and a BCD- 4221/5211 decimal PPR tree block. During the PPR stage, ODDS PPs are compressed to two PP rows by a binary CSA tree. The two ODDS PP rows can be converted directly to BCD-4221 (including both 4_BCD-4221 and BCD-4221); the BCD-4221 sum correction block is used to count the carries generated between two decimal digit columns. All BCD-4221 PP rows are then compressed to two

BCD-4221 PP rows by a decimal PPR tree. Finally, two BCD-4221 PP rows are converted into BCD-8421 PP rows and accumulated by a final decimal carry propagate addition. This design is referred to as hybrid, because as outlined above, it utilizes multiple BCD codes.

II. LITERATURE SURVEY

Decimal arithmetic is in demand for many commercial applications; for example in financial computation, binary arithmetic can introduce conversion and rounding errors. Furthermore, the decimal specification has been added to the revised IEEE 754-2008 standard, thus attracting a significant research interest in decimal arithmetic architectures and implementations. Decimal adders and multipliers are one of the most important primitives for decimal arithmetic and recently they have been studied quite extensively. The first parallel decimal multiplier is proposed in [6]. The most representative high-performance BCD multipliers are presented in [6-10]. Non-redundant and redundant decimal formats are used in a decimal multiplier design; two different sign-digit (SD) encodings (radix-5 and radix-10) are proposed for fast partial product (PP) generation. The SD radix-10 recoding is used to recode the signed digits in the digit set $\{5, 5_-\}$. $(d+1)$ PP rows are generated in the radix-10 recoding and $2d$ PP rows are generated in the radix-5 recoding. Double-BCD unsigned partial products are proposed in [7-10]. The number of PPs is equivalent to the SD radix-5 scheme; furthermore, this scheme has the disadvantage of a large area overhead compared with the radix-10 recoding. The new generation of high-performance DFUs is demanding more efficient implementations of decimal multiplication. Although this is an important and frequent operation, current hardware implementations suffer from lack of performance. Parallel binary multipliers are used extensively in most of the binary floating-point units for high performance. However, decimal multiplication is more difficult to implement due to the complexity in the generation of multiplicand multiples and the inefficiency of representing decimal values in systems based on binary signals. These issues complicate the generation and reduction of partial products. Thus, while decimal adders are implemented in a parallel fashion and are almost as efficient as binary ones, commercial implementations of decimal multipliers are sequential. These implementations are based on iterative algorithms for decimal integer multiplication and therefore, present low performance. Recently, several techniques related to the generation of partial products or providing improvements of sequential decimal multipliers have been proposed. Other recent works have proposed new techniques to speed up multi-operand BCD addition using tree-like (parallel) structures.

The first implementation of a parallel decimal multiplier is described in [7]. Several different parallel decimal multiplier architectures are proposed in [6], which use new techniques for partial product generation and reduction. Furthermore, some of these architectures were extended to support binary multiplication. Some concepts of [6] were applied in [5] to design decimal 4:2 compressor trees. All of the previous designs are combinational fixed-point architectures.

III. PROPOSED METHOD

Digital multiplication mainly consists of three stages: the generation of partial products, the fast reduction (addition) of partial products and the final carry propagate addition. Both non-redundant and redundant BCD digits are usually used in high-performance decimal multipliers.

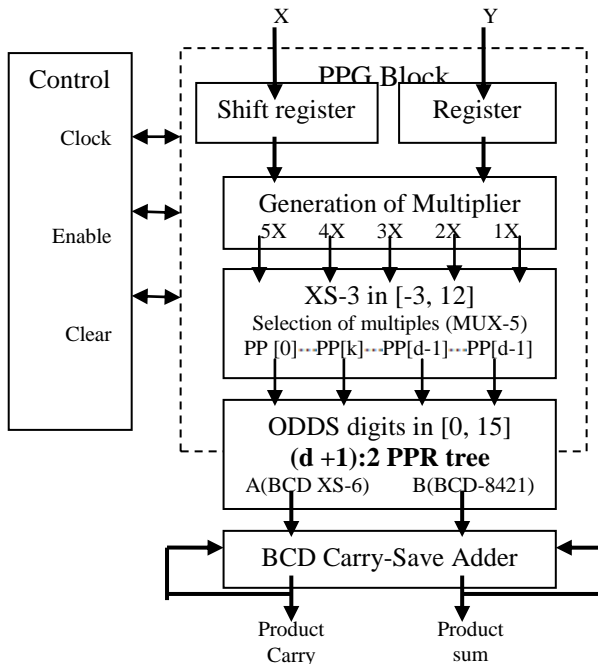


Fig. 1: Proposed Sequential Decimal Multiplier Using Hybrid BCD Codes

A. Sequential Decimal Multiplier

The details of the proposed sequential decimal multiplier are presented in this Section; the first part is devoted to elaborate on the architecture of the partial product generation (PPG) step and then the proposed PPA architecture are described. The representations of the multiplicand X , multiplier Y and the final product P are assumed to be standard BCD.

B. Product Generation and Reduction

The partial products, in the proposed design, are generated with easy multiples of X , $2X$, $4X$, $5X$. It should be noted that although there are other sets of easy multiples (e.g. $-X$, $-2X$ and $10X$), we use the mentioned easy multiples to avoid dealing with negative numbers in the PPA. With the intention of lowering the area cost, we opt for decimal 4-2-2-1 encoding as the representation of the multiples X , $2X$, $4X$, $5X$ in which the PPG mostly consists of wired shifts and various re-coders. However, we decided on generating the $4X$ directly from X , not by cascading two $2X$ blocks, so as to manage to perform PPG in one cycle. Consequently, the area of the proposed PPG is slightly higher than the previous counterpart

C. Partial Product Reduction (PPR) Tree

The proposed BCD multiplier uses a PPG circuit (based on both the ODDS and the XS-3 representations). The PPG circuit uses the SD radix-10 recoding to reduce the number of multiples with $\{0X, 1X, 2X, 3X, 4X, 5X\}$. The PPR tree is however different. $(d-1)$ carries (for the maximum digit column height) are generated by a binary CSA tree; a binary BCD-8421 counter is used in the (counter) correction block and the ODDS partial products are then accumulated. The proposed PPR tree consists of a binary PPR tree block (generating $d-2$ carries in the maximum digit column height), a non-fixed size BCD-4221 counter correction block and a BCD-4221 decimal PPR tree block. The functions of the three blocks are as follows:

- (1) A regular $(d+1):2$ binary CSA tree is used to compress the $d+1$ ODDS PP rows to two PP rows (i.e., sum and carry). The carry and sum signals can be directly represented in BCD-4221 recoding (both $4 \times \text{BCD-4221}$ and BCD-4221);

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

(2) A BCD-4221 sum correction block counts the carries generated between the digit columns in the binary CSA tree and a multiplication by six ($\times 6$) is then performed. (3) The BCD-4221 partial products in steps (1) and (2) are added by a decimal digit 3:2 compressor tree to obtain the final 2d-digit PPs (i.e., H and S). Every decimal bit: H_i is required to be multiplied by 2 before adding H and S.

D. Decimal Multiplier based on BCD-4221/5221

A d-digit decimal integer operand can be expressed as follows:

$$Z = \sum_{i=0}^{d-1} Z_i \times 10^i \text{ ----- (1)}$$

$$Z_i = \sum_{j=0}^3 Z_{i,j} \times r_j \text{ ----- (2)}$$

Where $Z_i \in [0, 9]$ is the i^{th} decimal digit, $Z_{i,j}$ is the j^{th} bit of the i^{th} digit, and $r_j \geq 1$ is the weight of the i^{th} bit. The most common used 4-bit decimal weighted codes include BCD-8421, BCD-4221, and BCD-5211.

The multiplicand $X(X = X_{d-1}X_{d-2} \dots X_1X_0)$ and multiplier $Y(Y = Y_{d-1}Y_{d-2} \dots Y_1Y_0)$ are assumed to be unsigned BCD decimal integer of d digits each. The product $P = X \times Y$ is in a non redundant BCD format with 2d digits. The decimal radix-10 recoding is used for a high performance parallel decimal multiplier; it leads to a simple generation of the partial products. The decimal carry-save algorithm based on BCD-4221/5211 is used for partial product reduction.

The SD radix-10 recoding transforms a BCD digit $Y_i(Y_i \in \{0,1, \dots, 8,9\})$ into an SD radix-10 digit $Yb_i(Yb_i \in \{-4, -5, \dots, 4,5\})$. The value of the recoded digit Yb_i depends on the decimal value of Y_i and the sign signal $ys_{i-1}(ys_{i-1} = 1 \text{ if } y_{i-1} \geq 5)$. Each digit Yb_i is represented by five multiple select signals $\{y1_i, y2_i, y3_i, y4_i, y5_i\}$ and a sign bit ys_i . Multiple select signals are used as a selection control signals for the 5:1 MUXs to select the positive multiplicand multiples $\{X, 1X, 2X, 3X, 4X, 5X\}$.

Due to the redundancy of the BCD-4221 decimal codes, when $ys_i = 1$, a negative multiplicand multiple $\{-X, -1X, -2X, -3X, -4X, -5X\}$ is selected by bit inversion of the positive BCD-4221 coded multiple. The bit inversion operand is accomplished by using the XOR gates controlled by ys_i . Therefore, $p(p = d + 1)PP$ rows are generated in the SD radix-10 decimal multiplier; each partial product $PP[i]$ is at most (d+3)-digits in length.

Each column of & PP rows is compressed to two digits by a decimal digit p:2 CSA tree; decimal carries are passed between adjacent digit columns. The decimal CSA uses a conventional 4-bit binary 3:2 CSA; this can be expressed as:

$$A_i + B_i + C_i = \sum_{j=0}^3 (a_{i,j} + b_{i,j} + c_{i,j})r_j$$

$$= \sum_{j=0}^3 s_{i,j}r_j + 2 \times \sum_{j=0}^3 h_{i,j}r_j = s_i + 2 \times H \text{ ----- (3)}$$

Where, the weight of $r_3r_2r_1r_0$ is 4221 or 5211, $s_{i,j}$ and $h_{i,j}$ are the sum and carry bits of a full adder (also defined as a 3:2 CSA or a 3:2 compressor). $S_i \in [0,9]$ and $H_i \in [0,9]$ in Eq. (3) are the decimal sum and carry digits at position i. Fig. 1 shows an example of a decimal 3:2 CSA for operands coded in BCD- 4221. Every decimal digit $H_i \in [0,9]$ ($h_{i,3}, h_{i,2}, h_{i,1}, h_{i,0}$) is required for multiplication by 2 (i.e., 2). The double of H ($2 \times H_i$) can be performed by converting the BCD-4221 to BCD-5211s (expressed by W_i) and left shifting W_i by 1-bit. The resultant digit is coded as BCD-4221s.

IV. RESULTS

The evaluation results of the proposed multiplier, in terms of latency and area, are presented in this Section where previous works are also examined and compared. The architectures are compared based on the required number of cycles for a single multiplication, the cycle time and the area. Moreover, the simulation results of the proposed

Organized by

Department of ECE, Adhityamaan College of Engineering, Hosur, Tamilnadu, India

multiplier based on delay, latency and area constraints are depicted compared to proposed design with others in Table (2).

Model	Area μm^2	Ratio	Delay nsec	No. of cycles
Proposed Sequential Decimal MUL	22500	1	2.3	21
High-frequency Decimal MUL	32430	1.32	3.2	25
Decimal MUL with efficient PPG	20638	1.35	2.8	19

As Figure (2) show, dynamic power is the main source of power consumption in parallel decimal multipliers and the PPR is a complex part which has the most hardware complexity, delay and it is the dominant part of the power consumption in a parallel decimal multiplier.

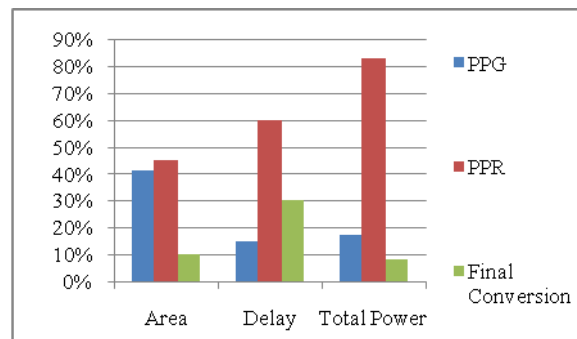


Fig. 2: Performance Analysis of Proposed Sequential Decimal Multiplier

V. CONCLUSION

A sequential decimal multiplier based on a new PPR tree using the hybrid BCD codes for the parallel structures has been proposed in this paper for improved performance. A high-frequency sequential decimal multiplier is proposed where the easy multiples (i.e. X, 2X, 4X, 5X) are used to generate the partial products represented in 4-2-2-1 encoding. The PPA unit is divided into two paths operating in parallel for a high-frequency design. In the proposed PPR tree, a BCD-4221 sum correction block is used to count the carries generated between the digit columns in a binary CSA tree; a decimal PPR tree based on BCD-4221 decimal digit 3:2 compressors is used to add the result of the carries count and the result of the binary CSA PPR tree to obtain the final two PP rows. Analysis and comparison show that the proposed decimal multiplier is faster than previous state-of-the-art designs. The proposed high-speed decimal multiplier can be used for both high performance fixed-point and floating-point decimal multiplications.

Organized by

Department of ECE, Adhityamaan College of Engineering, Hosur, Tamilnadu, India

REFERENCES

- [1] M. C. Wong and Razaidi Hussin, "Speed and Area Analysis on Hierarchy Multiplier," in International Conference on Applied Photonics and Electronics 2017 (InCAPE2017), 2017.
- [2] A. Kumar and S. Kamyra, "Design of a High Speed Multiplier by Using Ancient Vedic Mathematics Approach for Digital Arithmetic," International Journal of Electrical and Electronics Engineers, vol. 8, no. 2, pp. 244-255, July-Dec 2016.
- [3] A. Vazquez, E. Antelo, and J. Bruguera, "Fast Radix-10 Multiplication Using Redundant BCD codes", IEEE Transactions on Computers, vol. 63, no. 8, pp. 1902-1914, Aug. 2014.
- [4] S. Gorgin and G. Jaberipur, "A Fully Redundant Decimal Adder and Its Application in Parallel Decimal Multipliers", Microelectronics Journal, pp. 1471-1481, vol. 40, no. 10, Oct. 2009.
- [5] I.D. Castellanos and J.E. Stine, "Compressor Trees for Decimal Partial Product Reduction," Proc. 18th ACM Great Lakes Symp, VLSI, pp. 107-110, Mar. 2008.
- [6] A. Vazquez, E. Antelo, and P. Montuschi, "A New Family of High-Performance Parallel Decimal Multipliers," Proc. 18th IEEE Symp, Computer Arithmetic, pp. 195-204, June 2007.
- [7] T. Lang and A. Nannarelli, "A Radix-10 Combinational Multiplier," Proc. 40th Asilomar Conf. Signals, Systems, and Computers, pp. 313-317, Oct. 2006.
- [8] R.D. Kenney and M.J. Schulte, "High-Speed Multioperand Decimal Adders," IEEE Trans. Computers, vol. 54, no. 8, pp. 953- 963, Aug. 2005.
- [9] M.A. Erle and M.J. Schulte, "Decimal Multiplication via Carry- Save Addition," Proc. IEEE Int'l Conf. Application-Specific Systems, Architectures, and Processors, pp. 348-358, June 2003.
- [10] M.F. Cowlishaw, E.M. Schwarz, R.M. Smith, and C.F. Webb, "A Decimal Floating-Point Specification," Proc. 15th IEEE Symp. Computer Arithmetic, pp. 147-154, June 2001.