A Review On Fast Radix-10 Multiplication Using Binary Input And Convert Into Decimal Codes

Kajal B. Bobade, Prof. V. G. Roy, Prof. S. Kuntawar

***A*bstract–This paper presents a new architecture for a 7-bit binary to BCD (BD) converter which forms the core of our proposed high speed decimal multi-operand adder. Our proposed design contained various improvements over existing architectures. Simulation results indicate that with a marginal increase in area, the proposed BD converter exhibits an improvement of 55% in delay and up to 27% reduction of power delay product over earlier designs. New techniques are developed to reduce significantly the latency and area of previous representative high performance implementations. Partial products are generated in parallel using a signed digit radix-10 recoding of the BCD multiplier with the digit set [-5,5] and a set of positive multiplicand multiples (0X, 1X, 2X, 3X, 4X, 5X) coded in XS-3. This encoding has several advantages, first is self-complementing code. So that a negative multiplicand multiples in a free way.**

Index Terms **- Fast Multiplication, Redundancy, Partial Products, BCD Conversion**

Manuscript received May, 2017

First Author Name*, Dept. Electronics and Communication Engg., Gondwana University, Ballarpur, India, PH-8275246240.*

Second Author Name*, Dept. Electronics and Communication Engg., Gondwana university, Ballarpur, PH-9372425983.*

Third Author Name, *Dept. Electronics and Communication Engg., Gondwana University, Ballarpur, India PH-8983200835.*

INTRODUCTION

Radix-10 means base-10 i.e. decimal number is internally converted to BCD to perform fast multiplication. Decimal fixed point and floating point formats are important in financial, commercial, and user oriented computing, where conversion and rounding errors that are inherent to floating point binary representations cannot be tolerated. The area and power dissipation are critical design factors in state of the art decimal floating point units (DFPUs), multiplication and division are performed iteratively by means of digit by digit algorithms and therefore they present low performance.

The use of decimal arithmetic has been increasing over binary due to increase in the applications of internet banking and there are many others places where precision is very important. Binary digits have a disadvantage of not being able to represent digits like 0.1 or 0.7, requires an infinitely recurring binary number. The simultaneous addition of several decimal numbers is the common operation in multiplication and division algorithms. In case of decimal multiplication multi operand decimal addition comes in handy for swiftly summing large amounts of decimal data.

A novel design for 7-bit binary to BCD converter circuit is proposed. Further, analysis is done with respect to the existing binary to BCD converter architectures. As the decimal corrections are achieved separately from the computation of the binary sum, such that the layout of the binary carry save adder does not require any further rearrangement, the design can perform as unified binary/BCD multi-operand adder.

LITERATURE SURVEY

This section describes previously proposed studies.

As in (1) paper revealed by A. Aswal, M.G. Perumal , and G.N.S. Prasanna has mentioned regarding the many early computers used decimal arithmetic at the hardware level but binary computing in hardware soon took over after Von Neumann pointed out the advantages of simplified hardware.

As in (2) paper revealed by M.F. Cowlishaw, E.M. Schwarz, R.M. Smith and C.F. Webb has mentioned regarding the people habitually perform arithmetic in base 10. When calculations are moved to computers there is usually a loss in translating a decimal fraction to binary representation.

As in (8) paper revealed by M.A. Erle and M.J. Schulte has mentioned regarding the current floating point units are typically binary based for largely two reasons. Binary data can be stored efficiently and manipulated very quickly on two-state computers.

HIGH-LEVEL ARCHITECTURE

The high-level block diagram of the proposed parallel architecture for d×d digit BCD decimal integer and fixed point multiplication is shown in fig. 1. This architecture accepts conventional BCD inputs X, Y, generates redundant BCD partial products PP, and computes the BCD product P=X×Y. It consists of the following 3 stages. (1) parallel generation of partial products coded in XS-3, including generation of multiplicand multiples and recoding of the multiplier operand, (2) recoding of the partial products from XS-3 to the ODDS representation and subsequent reduction, and (3) final conversion to a non-redundant 2d-digit BCD product.

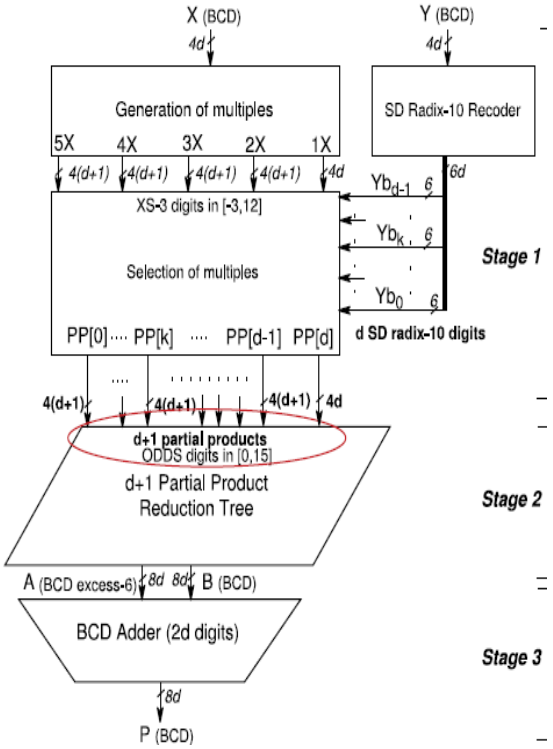

Fig.1 Combinational SD Radix-10 architecture

Stage 1) Decimal partial product generation.

SD radix-10 recording of the BCD multiplier has been used. This recoding produces a reduced number of partial products that leads to a significant reduction in the overall multiplier area. The recoding of the d digit multiplier Y into SD radix-10 digits Ybd-1...Yb0, produces d partial products PP[d-1]...PP[0], one per digit. An additional partial product PP[d] is produced by the most significant multiplier digit after the recoding, in case the total number of partial products generated is d+1.

Stage 2) Decimal partial product reduction.

The array of d+1 ODDS partial products are reduced to two 2d-digit words (A, B). The array of d+1 ODDS partial products can be viewed as adjacent digit columns of height h≤d+1. This augment significantly the latency of the partial product reduction tree. Moreover, the proposed architecture accepts an arbitrary number of ODDS or BCD operand inputs. Our proposal aims to

provide an optimal reuse of any binary CSA tree for multioperand decimal addition, as it was done for the 4221 and 5211 decimal codings.

Stage 3) Conversion to BCD

We consider the use of the BCD carry-propogate adder to perform the final conversion to a non redundant BCD product P=A+B. The proposed architecture is a 2d-digit hybrid parallel prefix/carry-select adder, the BCD quaternary tree adder. Furthermore, to generate the correct decimal carry, the BCD addition algorithm implemented depend upon Ai+Bi to be obtained in excess-6.

APPLICATION

1. Generation of the multiplicand multiples
2. Data processing
3. Encoder and decoder devices
4. Correction term
5. Redundant excess-3 code
6. Redundant arithmetic

CONCLUSION

In this paper, redundant BCD codes based high speed radix-10 multiplication will be implemented. The result will compare for both existing and proposed methods. From the received results, it is clear that the proposed decimal multiplier gives reduced delay and area because of redundant BCD representation. Partial products can be generated very fast in the XS-3 representation using the SD radix-10 PPG scheme: positive multiplicand multiples (0X, 1X, 2X, 3X, 4X, 5X) are precomputed in a carry free way, while negative multiples are obtain by bit inversion of the positive once. The area and delay figures estimated from both a theoretical model and synthesis show that our BCD multiplier presents 20-35% less area than other designs for a given target delays.

ACKNOWLEDGMENT

This research was supported by publisher of this paper. We thank our colleagues from Ballarpur Institute of Technology who provided expertise that greatly assisted the research.

REFERENCES

[1] A. Aswal, M.G. Perumal, and G.N.S. Prasanna, “On basis financial decimal operations on binary machines,” IEEE trans. Comput., vol. 61, no. 8, pp. 1084-1096, Aug. 2012.

[2] M.F. Cowlishaw, E.M. Schwarz, R.M. Smith, and C.F. Webb, “A decimal floating-point specification,” in Proc. 15th IEEE symp. Comput. Arithmatic, June 2001, pp. 147-154.

[3] M. F. Cowlishaw, “Decimal floating- point: Algorithm for computers,” in Proc. 16th IEEE symp. Comput. Arithmatic, Jully 2003.

[4] S. Carlough and E. Schwarz, “Power6 decimal divide,” in Proc. 18th IEEE Symp.

[5] S. Carlough, S. Mueller, A. Collura, and M. Kroener, “The IBM zEnterprise-196 decimal floating point accelerator”.

[6] L. Dadda, “Multioperand parallel decimal adder: A mixed binary and BCD approach,” IEEE Trans. Comput., vol. 56, no.10,pp. 1320-1328, oct. 2007.

[7] L. Dadda and A. Nannarelli, “A variant of a Radix-10 combinational multiplier,” in proc. IEEE Int. Symp. Circuits syst, May 2008, pp. 3370-3373.

[8] M.A. Erle and M.J. Schulte. “Decimal multiplication via carrysane adding,” in proc. IEEE Int. Conf Apple. – specific syst., Arch., Process, Jun 2003,pp. 348-358.