IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 4, Ver. III (Jul.-Aug. 2017), PP 21-23
www.iosrjournals.org

Fast Radix-10 Multiplication Using Binary Input and Convert into Decimal Codes

*Kajal Bobade[1], Vijay Roy[2], Sunil Kuntawar[3]
[1](Electronics and Communication, BIT College/ Gondwana University, India)
[2](Electronics and Communication, BIT College/ Gondwana University, India)
[3](Electronics and Communication, BIT College/ Gondwana University, India)
Corresponding Author: Kajal Bobade

Abstract: *We present the algorithm and architecture of a BCD parallel multiplier that exploits some properties of two different redundant BCD codes to speedup its computation, the redundant BCD excess-3 code (XS-3) and the overloaded BCD representation (ODDS). The partial products can be recoded to the ODDS representation by just adding a constant factor into the partial product reduction tree. To show the advantages of our architecture, we have synthesized a RTL model for 16×16 digit and 34×34 digit multiplications and performed a comparative survey of the previous most representative 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.*

Keywords: *Decimal hardware, overloaded BCD representation, parallel multiplication, redundant arithmetic, redundant excess-3 code*

Date of Submission: 17-07-2017 Date of acceptance: 28-07-2017

I. 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.

II. Litrature 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.

III. 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.

DOI: 10.9790/2834-1204032123

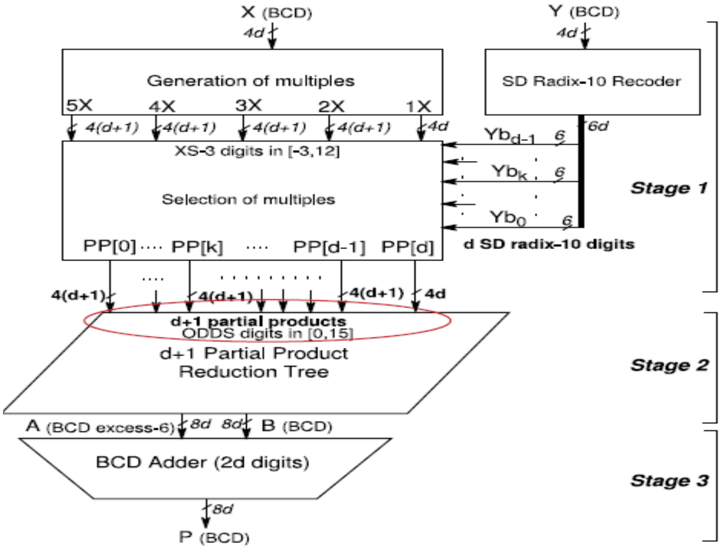


Figure.1: Combinational SD Radix-10 architecture

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.

IV. Applications

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

V. Result

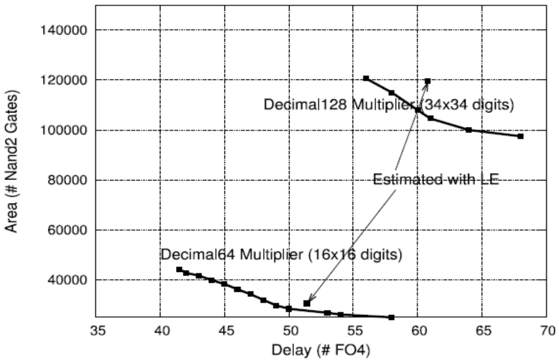
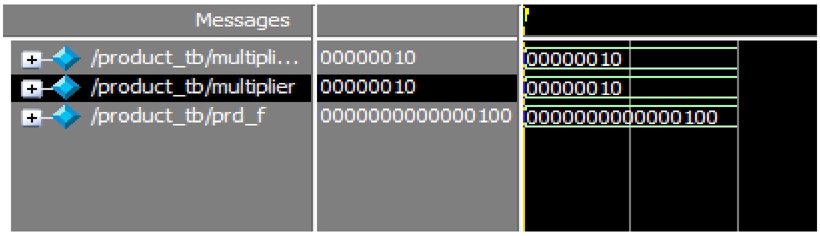


Figure.2: Area-Delay space obtained from synthesis

The designs have been synthesized using synopsis design compiler and a 90 nm CMOS standard cell library. The delay for this library is 49 ps under typical conditions (1V, 25º C). The area delay curves of fig. have been obtained with the constraint Cout = Cin = 4Cinv, where Cinv is the input capacitance of an 1× inverter of the library. We also include in fig. the area and delay points estimated from the LE- based model evaluation. Nevertheless, some specific structures have been optimized internally to reduce the overall delay.
To ensure the correctness of the designs we have simulated the RTL models of the 16×16- digit and 34×34- digit multipliers using the synopsis VCS-MX tool and a comprehensive set of random test vectors.

The proposed combinational architectures for BCD 16×16-digit and 34×34-digit multipliers are evaluated and compared with other representative BCD multipliers. The area and delay figures of our architectures were obtained from an area-delay evaluation model for static CMOS gates, and validated with the synthesis of verified RTL models coded in VHDL.



```
Messages
/product_tb/multipli...   00000010           00000010
/product_tb/multiplier    00000010           00000010
/product_tb/prd_f         0000000000000100   0000000000000100
```

Figure.3: Result

Figure 3 shows the product of two digits, here first term shows the multiplicand and second term shows the multiplier and the last i.e. third term shows the final output of multiplicand and multiplier. Multiplicand is taken as 0010 which is equals to decimal 2 and multiplier is also taken as 0010 which is equals to decimal 2. And this project gives the result of product as 0100 which is equals to decimal 4.

VI. Conclusion

In this paper, redundant BCD codes based high speed radix-10 multiplication is implemented. The result were compared 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 pre computed in a carry free way, while negative multiples are obtained by bit inversion of the positive once. The area and delay figures estimated from both a theoretical model and synthesis shows that our BCD multiplier presents 20-35% less area than other designs for a given target delay.

Acknowledgements

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 system, may 2008, pp. 3370-3373.

[8] L. Eisen, J.W. Ward, H-W. Tast, N. Maoding J. Leenstra, S.M. Mueller, C. Jacobi, J. Preiss, E.M. Schwarz, and S.R. Carlough, "IBM POWER 6 accelarators VMX and DFU," IBM J. Res. Dev., vol.51, no. 6, pp. 663-684, nov. 2007.

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

[10] M.A. Erle, E.M. Schwarz, and M.J. Schulte, "Decimal multiplication with efficient partial product generation," in proc. 17th IEEE symp. Compute. Arithmetic, jun 2005,pp.21-28.

IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) is UGC approved Journal with Sl. No. 5016, Journal no. 49082.

Kajal Bobade. " Fast Radix-10 Multiplication Using Binary Input and Convert into Decimal Codes." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) 12.4 (2017): 21-23.