International Conference on Advances In Computing ,Electrical and Communication Engineering(ICACECE-2017)

International Journal of Advanced Scientific Technologies, Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) *Volume.3,Special Issue.1,May.2017*

A Novel Approach to Redundant BCD Codes for High Speed Radix-10 Multiplication

1 N.Prakash Babu, 2 B.Siva Prasad, 3 S.Chandra Kantha Rao
1,2,3 Associate Professor, Department of Electronics and Communication Engineering, PACE Institute of Technology and Sciences, Ongole, Andhra Pradesh.

Abstract: ***This paper recommend a formula and structure of a BCD similar multiplier that uses some qualities of two different repetitive BCD requirements to accelerate its calculations. In this document, proposed a new techniques to lower the latency and region of past associate top rated implementations. The Limited items are produced in similar using a signed-digit radix-10 recoding of BCD multiplier with the number set [-5, 5], and set of beneficial multiplicand many (0X, 1X, 2X, 3X, 4X, 5X) written in excess-3 code(XS-3). The partial items can be recoded to the bombarded BCD reflection (ODDS) by just including a continuous aspect into the partial product decrease shrub. To demonstrate the key benefits of our suggested structure, we have produced a RTL design for 16 x 16-digit multiplications and conducted a relative study of the current associate styles.***

Keywords: Parallel multiplication, decimal hardware, overloaded BCD representation, redundant excess-3 code, redundant arithmetic.

I.INTRODUCTION

Decimal fixed-point and floating-point types are important in financial, professional, and user-oriented processing, where transformation and rounding mistakes that are natural to floating-point binary representations cannot be accepted. The new IEEE 754-2008 Conventional for Floating- Aspect Mathematics, which contains a structure and requirements for decimal floating-point (DFP) arithmetic has motivated a lot of research in decimal components. BCD encodes a wide range X in decimal (non-redundant radix-10) structure, with each decimal wide range Xi E [0,9] showed in a 4-bit binary wide range program. (1)it is a repetitive decimal reflection so that it allows carry-free creation of both simple and sophisticated decimal many (2X, 3X, 4X, 5X, 6X,. . .) In this work, we concentrate on the enhancement of similar decimal multiplication by taking benefit of the redundancy of two decimal representations: We recommend the use of a common repetitive BCD arithmetic (that contains the ODDS, XS-3 and BCD representations) to speed up similar BCD multiplication in two ways: Partial item creation (PPG).

II. REDUNDANT BCD REPRESENTA-TIONS

The proposed decimal multiplier uses internally a redundant BCD arithmetic to speed up and simplify the implementation. This arithmetic deals with radix-10 ten's complement integers of the form:

$$Z = -s_z \times 10^d + \sum_{i=0}^{d-1} Z_i \times 10^i,$$

where d is the number of digits, sz is the sign bit, and Zi E [l – e,m- e] is the ith digit, with

$$0 \le l \le e, \quad 9 + e \le m \le 2^4 - 1 (= 15).$$

Parameter e is the excess of the representation and usually takes values 0 (non excess), 3 or 6. The redundancy index p is defined as p=m-l+1-r, being r=10. On the other hand, the binary value of the 4-bit vector representation of Zi is given by

zi;j being the jth bit of the ith digit.

$$Z_i = [Z_i] - e.$$

Note that bit-weighted codes such as BCD and ODDS use the 4-bit binary encoding (or BCD encoding) defined in Expression (2). Thus, Zi =[Zi] for operands Z represented in BCD or ODDS. In our work we use a SD radix-10 recoding of the BCD multiplier [30], which requires to compute a set of decimal multiples ({-5X, . . . , 0X, . . . , 5X}) of the BCD multiplicand. The main issue is to perform the x3.The nine's complement of a positive decimal operand is given by

$$-10^d + \sum_{i=0}^{d-1} (9 - Z_i) \times 10^i.$$

The implementation of (9- Zi) leads to a complex implementation, since the Zi digits of the multiples generated may take values higher than 9.

TABLE 1
Nine's Complement for the XS-3 Representation

Digit			Nine's Complement		
4-bit Encoding	Z_i	$[Z_i]$	4-bit Encoding	$9 - Z_i$	$[9 - Z_i]$ $(=15 - [Z_i])$
0000	-3	0	1111	12	15
0001	-2	1	1110	11	14
0010	-1	2	1101	10	13
0011	0	3	1100	9	12
0100	1	4	1011	8	11
0101	2	5	1010	7	10
0110	3	6	1001	6	9
0111	4	7	1000	5	8
1000	5	8	0111	4	7
1001	6	9	0110	3	6
1010	7	10	0101	2	5
1011	8	11	0100	1	4
1100	9	12	0011	0	3
1101	10	13	0010	-1	2
1110	11	14	0001	-2	1
1111	12	15	0000	-3	0

In Table 1 we show how the nine's complement can be performed by simply inverting the bits of a digit Zi coded in XS-3. At the decimal digit level, this is due to the fact that:

$$(9 - Z_i) + 3 = 15 - (Z_i + 3)$$

for the ranges Zi E[-3,12] ([Zi] E [0, 15]). In summary, the main reasons for using the redundant XS-3 code are: (1) to avoid long carry-propagations in the generation of decimal positive multiplicand multiples, (2) to obtain the negative multiples from the corresponding positive ones easily.

III.HIGH-LEVEL ARCHITECTURE

The high-level prevent plan of the suggested similar structure for dx d-digit BCD decimal integer and fixed-point multiplication is caved Fig. 1. It comprises of the following three stages1: (1) similar creation of limited items written in XS-3, such as creation of multiplicand many and recoding of the multiplier operand, (2) recoding of limited items from XS-3 to the ODDS reflection

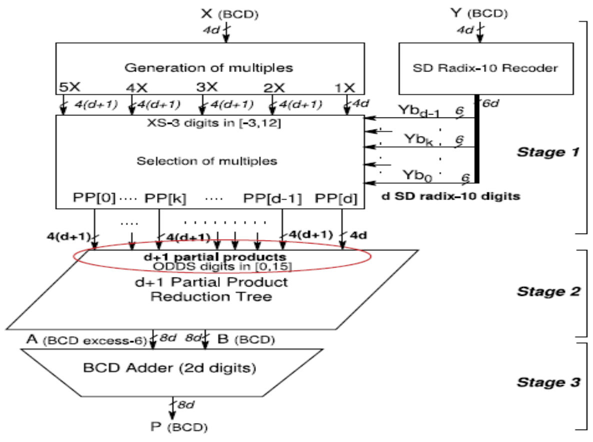


Fig. 1. Combinational SD radix-10 architecture

Stage 1) Decimal partial product generation. Stage 2) Conversion to (non-redundant) BCD. The proposed architecture is a 2d-digit hybrid parallel prefix/carry-select adder, the BCD Quaternary Tree adder. We opt for representing operand A in BCD excess-6 (Ai e [0, 9], [Ai] = Ai + e, e =6), and B coded in BCD (Bi E [0, 9], e = 0).

IV.DECIMAL PARTIAL PRODUCT GENERATION

The partial product generation stage comprises the recoding of the multiplier to a SD radix-10 representation, the calculation of the multiplicand multiples in XS-3 code and the generation of the ODDS partial products.

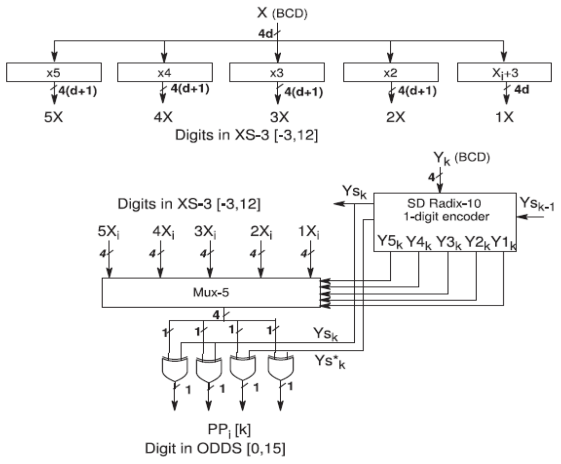


Fig. 2. SD radix-10 generation of a partial product digit.

A. Generation of the Multiplicand Multiples

Fig. 3 shows the high-level block diagram of the multiples generation with just one carry propagation. This is performed in two steps:

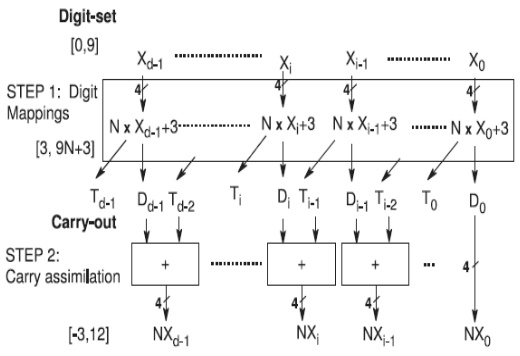


Fig. 3. Generation of a decimal multiples NX.

1)digit recoding of the BCD multiplicand digits Xi into a decimal carry 0< Ti<Tmax and a digit -3< Di <12- Tmax, such as

$$D_i + 10 \times T_i = (N \times X_i) + 3,$$

being T max the maximum possible value for the decimal carry.

2) The decimal carries transferred between adjacent digits are assimilated obtaining the correct 4-bit representation of XS-3 digits NXi, that is

$$[NX_i] = D_i + T_{i-1}, [NX_i] \in [0,15] (NX_i \in [-3,12]).$$

The constraint for NXi still allows different implementations for NX. Table 2 shows the preferred digit recoding for the multiples NX.

$$\overline{NX_i} = 15 - [NX_i],$$

Replacing the relation between NXi and [NXi] in the previous expression, it follows that

$$\overline{NX_i} = 15 - (NX_i + 3) = (9 - NX_i) + 3.$$

B. Most-Significant Digit Encoding

The MSD of each PP[k], PP[dk], is directly obtained in the ODDS representation. For positive partial products we have

$$PP_d[k] = T_{d-1}$$

with Td-1 E {0, 1, 2, 3, 4}. Therefore the two cases can be expressed as

$$PP_d[k] = -10 + (9 - T_{d-1}) = -1 - T_{d-1}$$

$$PP_d[k] = -8 + [PP_d[k]],$$

With

$$[PP_d[k]] = 8 - Ys_k + (-1)^{Ys_k} T_{d-1.}$$

TABLE 2
Preferred Digit Recoding Mappings for NX Multiples

C. Correction Term

The pre-computed correction term is given by

$$f_c(d) = -8 \times \sum_{k=0}^{d-1} 10^{k+d} - 3 \times \left(\sum_{i=0}^{d-1} (i+1)10^i + \sum_{i=0}^{d-2} (d-1-i)10^{i+d} \right).$$

D. Product Array

Fig. 4 illustrates the shape of the partial product array, particularizing for d =16. Note that the maximum digit column height is d+1.

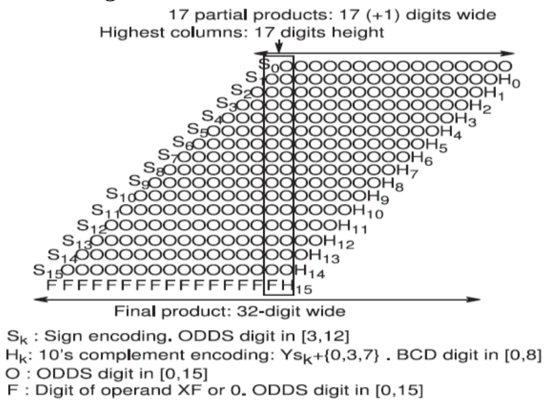
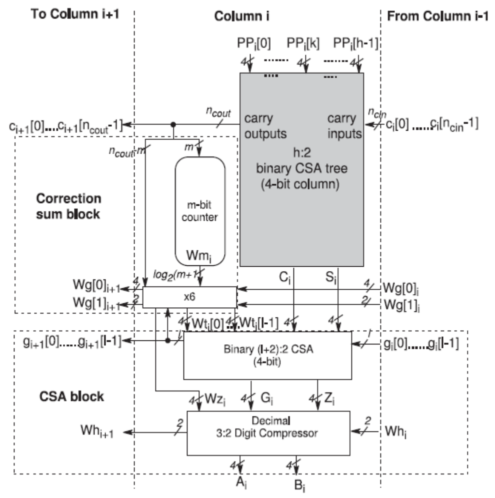

17 partial products: 17 (+1) digits wide
Highest columns: 17 digits height
Final product: 32-digit wide
S_k : Sign encoding. ODDS digit in [3,12]
H_k: 10's complement encoding: Ys_k+{0,3,7} . BCD digit in [0,8]
O : ODDS digit in [0,15]
F : Digit of operand XF or 0. ODDS digit in [0,15]

Fig. 4. Decimal partial product array generated for d =16

V. DECIMAL PARTIAL PRODUCT REDUCTION

The PPR tree consists of three parts: (1) a regular binary CSA tree to compute an estimation of the decimal partial product sum in a binary carry-save form (S, C), (2) a sum correction block to count the carries generated between the digit columns, Fig. 5 shows the high-level architecture of a column of the PPR tree (the ith column) with h



ODDS digits in [0, 15] (4 bits per digit).

Fig. 5. High-level architecture of the proposed decimal PPR tree (h inputs, 1-digit column). This difference, T, is computed in the sum correction block of every digit column and added to the partial product sum (S, C) in the decimal CSA.

$$W_i = \sum_{k=0}^{n_{cout}-1} c_{i+1}[k],$$

the contribution of the column i to the sum correction term T is given by

$$W_i \times 16 - W_i \times 10 = W_i \times 6.$$

Therefore, the sum correction is given by

$$T = \sum_{i=0}^{2d-1} (W_i \times 6 \times 10^i) = 6 \times \sum_{i=0}^{2d-1} W_i \times 10^i.$$

Consequently, the sum correction block evaluates Wix6. This module is composed of a m-bit binary counter and a x6 operator.

VI.FINAL CONVERSION TO BCD

The chosen structure is a 2d-digit multiple identical prefix/ carry-select adder, the BCD Quaternary Shrub adder. To style the bring prefix tree we examined the indication appearance information from the PPRT tree.

VII. SYNTHESIS RESULTS

Finally, we existing a more in depth evaluation of the quickest BCD 16x16-digit combinational multipliers in regards to latency and place. The corresponding place wait synthesis principles are proven in Fig.6.

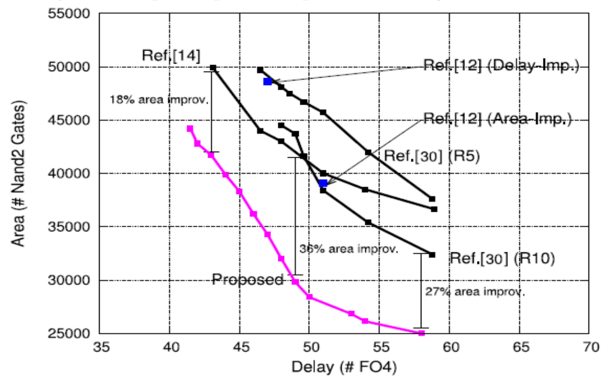


Fig. 10. Area-delay space for the fastest 16x16-digit mults.

VIII. CONCLUSION

In this document we have provided the criteria and structure of a new BCD similar multiplier. The developments of the suggested structure depend on the use of certain repetitive BCD requirements, the XS-3 and ODDS representations. Limited items can be produced very quick in the XS-3 reflection using the SD radix-10 PPG scheme: beneficial multiplicand many (0X, 1X, 2X, 3X, 4X, 5X) are pre calculated in a carry-free way, while adverse many are acquired by bit inversion of the beneficial ones.

REFERENCES

[1] A. Aswal, M. G. Perumal, and G. N. S. Prasanna, "On basic financialdecimal 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. Arithmetic, Jun. 2001, pp. 147–154.
[3] M. F. Cowlishaw, "Decimal floating-point: Algorism for computers," in Proc. 16th IEEE Symp. Comput. Arithmetic,
[4] S. Carlough and E. Schwarz, "Power6 decimal divide," in Proc. 18th IEEE Symp. Appl.-Specific Syst., Arch., Process., Jul. 2007, pp. 128–133. [5]

S. Carlough, S. Mueller, A. Collura, and M. Kroener, "The IBM zEnterprise-196 decimal floating point accelerator," in Proc. 20th IEEE Symp. Comput. Arithmetic, Jul. 2011, pp. 139–146.
[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] L. Eisen, J. W. Ward, H.-W. Tast, N. Mading, J. Leenstra, S. M. Mueller, C. Jacobi, J. Preiss, E. M. Schwarz, and S. R. Carlough,
[9] M. A. Erle and M. J. Schulte, "Decimal multiplication via carrysave addition," in Proc. IEEE Int. Conf Appl.-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," 2005, pp. 21–28.
[11] Faraday Tech. Corp. (2004). 90nm UMC L90 standard performance low-K library(RVT).[Online].Available:http://freelibrary.faraday-tech.com/