Gorripati Sreenivasulu* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.4, Issue No.6, October – November 2016, 4690-4692.

Reducing Latency And Receiving High-Recital In Analogous Multipliers

GORRIPATI SREENIVASULU
M.Tech Student, Dept of ECE
SKR College of Engineering & Technology
Nellore, Andhra Pradesh, India

B DORASWAMY
Associate Professor, Dept of ECE
SKR College of Engineering & Technology
Nellore, Andhra Pradesh, India

Abstract: **Partial goods are generated in parallel utilizing a signed-digit radix-10 recoding from the BCD multiplier using the digit set [-5, 5], and some positive multiplicand multiples (0X, 1X, 2X, 3X, 4X, 5X) created in XS-3. This encoding has lots of advantages. We present the formula and architecture of the BCD parallel multiplier that exploits some qualities of two different redundant BCD codes to hurry up its computation: the redundant BCD excess-3 code (XS-3), and also the overloaded BCD representation (ODDS). Additionally, new techniques are designed to reduce considerably the latency and section of previous representative high end implementations. First, it's a self-complementing code, to ensure that an adverse multiplicand multiple could be acquired just by inverting the items of the related positive one. Also, the accessible redundancy enables a easy and quick generation of multiplicand multiples inside a carry-free way. Finally, the partial products could be recoded towards the ODDS representation just by adding a continuing factor in to the partial product reduction tree. To exhibit the benefits of our architecture, we've synthesized a RTL model for 16 x 16-digit and 34 x 34-digit multiplications and performed a comparative survey from the previous most representative designs. Because the ODDS utilize a similar 4-bit binary encoding as non-redundant BCD, conventional binary VLSI circuit techniques, for example binary carry-save adders and compressor trees, could be adapted efficiently to do decimal operations.**

Keywords: **Parallel Multiplication; Decimal Hardware; Overloaded BCD Representation; Redundant Excess-3 Code;**

I. INTRODUCTION

Since area and power dissipation are critical design factors in condition-of-the-art DFPUs, multiplication and division are carried out iteratively by way of digit-by-digit algorithms, and for that reason they present low performance. Furthermore, the aggressive cycle duration of these processors puts yet another constraint on using parallel approaches for lowering the latency of DFP multiplication in high-performance DFPUs. The brand new IEEE 754-2008 Standard for Floating-Point Arithmetic, containing a format and specs for decimal floating-point (DFP) arithmetic has encouraged a lot of research in decimal hardware [1]. Hardware implementations normally use BCD rather of binary to control decimal fixed-point operands and integer significant of DFP figures for simple conversion between machine and user representations. Furthermore, the implementation of BCD arithmetic has more complications than binary, which result in area and delay penalties within the resulting arithmetic units. A number of redundant decimal formats and arithmetic's happen to be suggested to enhance the performance of BCD multiplication. BCD carry-save and signed-digit radix-10 arithmetic's offer enhancements in performance regarding no redundant BCD. However, the resultant VLSI implementations in current technologies of multioperand adder trees may lead to more irregular layouts than binary carry-save adders (CSA) and compressor trees. The BCD carry-save format represents a radix-10 operand utilizing a BCD digit along with a carry bit each and every decimal position. It's meant for carry-free accumulation of BCD partial products using rows of BCD digit adders arranged in straight line or tree-like configurations. The extra redundancy obtainable in some-bit encoding can be used to hurry-up BCD operations while retaining exactly the same data path width. In addition, these codes are self-complementing, so the 9's complement of the digit, needed for negation, is definitely acquired by bit-inversion of their 4-bit representation. The overloaded BCD representation was suggested to enhance decimal multioperand addition, and consecutive and parallel decimal multiplications. Within this work, we concentrate on the improvement of parallel decimal multiplication by exploiting the redundancy of two decimal representations: the chances and also the redundant BCD excess-3 (XS-3) representation, a self-complementing code using the digit set [-3, 12]. We make use of a minimally redundant digit looking for the recoding from the BCD multiplier digits, the signed-digit radix-10 recoding, that's, the recoded signed digits. We advise using a general redundant BCD arithmetic to accelerate parallel BCD multiplication in 2 ways: Partial product generation (PPG). By generating positive multiplicand multiples created in XS-3 inside a carry-free-form. By performing the decrease in partial products created in ODDS via binary carry-

save arithmetic. Partial products could be recorded in the XS-3 representation towards the ODDS representation just by adding a continuing factor in to the partial product reduction tree. The resultant partial product reduction tree is implemented using regular structures of binary carry-save adders or compressors [2]. Some-bit binary encoding of ODDS operands enables a far more efficient mapping of decimal algorithms into binary techniques.

II. SYSTEM DESIGN

The suggested decimal multiplier uses internally a redundant BCD arithmetic to hurry up and simplify the implementation. This binary encoding simplifies the hardware implementation of decimal arithmetic units, because we can use condition-of-the-art binary logic and binary arithmetic strategies to implement digit operations. Particularly, the chances representation presents interesting qualities for a quick and efficient implementation of multioperand addition. Furthermore, conversions from BCD towards the ODDS representation are straightforward, because the digit group of BCD is really a subset from the ODDS representation. Within our work we make use of a SD radix-10 recoding from the BCD multiplier, which requires to compute some decimal multiples from the BCD multiplicand [3]. And so the redundant BCD representations can host the resultant digits with only one decimal carry propagation. An essential problem for this representation may be the ten's complement operation. Since following the recoding from the multiplier digits, negative multiplication digits may end up, it's important to negate the multiplicand to get the negative partial products. This operation is generally made by computing the nine's complement from the multiplicand and adding a 1 within the proper put on the digit array. An easy implementation is acquired by observing the excess-three of the nine's complement of the operand is equivalent to the part-complement from the operand created in excess-3. Therefore to possess a simple negation for partial product generation we make the decimal multiples within an excess-3 code. The negation is conducted by simple bit inversion, that matches the surplus-three of the nine's complement from the multiple.

III. METHODOLOGY

This architecture accepts conventional BCD inputs X, Y, generates redundant BCD partial products PP, and computes the BCD product P = X x Y. It includes the next three stages: parallel generation of partial products created in XS-3, including generation of multiplicand multiples and recoding from the multiplier operand, recoding of partial products from XS-3 towards the ODDS representation and subsequent reduction, and final conversion to some non-redundant 2d-digit BCD product. A SD radix-10 recoding from the BCD multiplier has been utilized. This recoding creates a reduced quantity of partial items that results in a significant decrease in the general multiplier area. Our proposal uses binary carry save adder tree to do carry-free additions from the decimal partial products. Since ODDS digits are encoded in binary, the guidelines for binary arithmetic apply inside the digit bounds, and just carries generated between radix-10 digits lead towards the decimal correction from the binary sum. We consider using a BCD carry-propagate adder to do the ultimate conversion to some non-redundant BCD product P = A B. The suggested architecture is really a 2d-digit hybrid parallel prefix/carry-select adder, the BCD Quaternary Tree adder [4]. The sum of the input digits Ai, Bi each and every position i needs to be within the range [18] to ensure that for the most part one decimal carry is propagated to another position i 1. The partial product generation stage comprises the recoding from the multiplier to some SD radix-10 representation, the calculation from the multiplicand multiples in XS-3 code and also the generation from the ODDS partial products. The negative multiples are acquired by ten's complementing the positive ones. This is the same as using the nine's complement from the positive multiple after which adding 1. Observe that these digits keep carries generated within the computation from the multiplicand multiples and also the sign little bit of the partial product. The decimal carries transferred between adjacent digits are assimilated acquiring the right 4-bit representation of XS-3 digits NX. The constraint for NXi still enables different implementations for NX. For any specific implementation, the mappings for Ti and Di need to be selected. The resultant partial product sum needs to be remedied from the-critical-path with the addition of a pre-computed term. Really, adding these -3 constants is the same as convert the XS-3 digits from the partial products towards the ODDS representation. The PPR tree includes three parts: (1) a normal binary CSA tree to compute an estimation from the decimal partial product sum inside a binary carry-save form (S, C), (2) an amount correction block to count the carries generated between your digit posts, and (3) a decimal digit 3:2 compressor which increments the carry-save sum based on the carries count to get the final double-word product (AB), A being symbolized with excess-6 BCD digits and B being symbolized with BCD digits. The digit posts from the binary CSA tree are implemented efficiently using 4-bit 3:2, 4:2 and greater order compressors made from full adders. These compressors make use of the delay difference from the inputs as well as the sum and carry outputs from the full adders, allowing significant delay reductions. However, to balance pathways and

lower the critical path delay we considered some optimizations. Particularly, the enhanced implementation of the block heavily depends upon the truth from the decimal representation therefore its implementation is just outlined here, without entering details. The reduced-level implementation information on the x6 module relies on the amount of carry-outs. The partial product array generated within the suggested 16 x 16-digit BCD multiplier is proven. The utmost height from the partial product array through the 34 x 34-digit BCD multiplier is h = 35. The suggested implementation for that maximum height posts from the PPR tree is proven. The chosen architecture is really a 2d-digit hybrid parallel prefix/carry-select adder, the BCD Quaternary Tree adder. The delay of the adder is slightly greater towards the delay of the binary adder of 8d bits having a similar topology [5]. The decimal carries are computed utilizing a carry prefix tree, while two conditional BCD digit sums are computed from the critical path using 4-bit digit adders which implements. To create the carry prefix tree we examined the signal arrival profile in the PPRT tree, and regarded using different prefix tree topologies to optimize the region for that minimum delay adder.

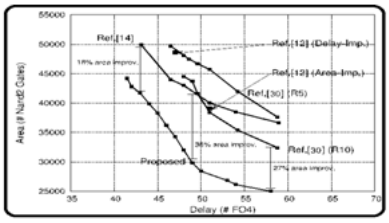

Fig.1.Area-delay space

IV. CONCLUSION

Partial products could be generated extremely fast within the XS-3 representation while using SD radix-10 PPG plan: positive multiplicand multiples (0X, 1X, 2X, 3X, 4X, 5X) are pre-computed inside a carry-free way, while negative multiples are acquired by bit inversion from the positive ones. Within this paper we've presented the formula and architecture of the new BCD parallel multiplier. The enhancements from the suggested architecture depend on prescribed medication redundant BCD codes, the XS-3 and ODDS representations. However, recoding of XS-3 partial products towards the ODDS representation is easy. The Chances representation uses the redundant digit-set [, 15] along with a 4-bit binary encoding (BCD encoding), which enables using a binary carry-save adder tree to do partial product reduction in an exceedingly efficient way. The region and delay figures believed from both a theoretical model and synthesis reveal that our BCD multiplier presents 20-35 % less area than other kinds for any given target delays. We've presented architectures for IEEE-754 formats, Decimal64 and Decimal128.

V. REFERENCES

[1] 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.

[2] R. D. Kenney, M. J. Schulte, and M. A. Erle, "High-frequency decimal multiplier," in Proc. IEEE Int. Conf. Comput. Des.: VLSI Comput. Process., Oct. 2004, pp. 26–29.

[3] L. Han and S. Ko, "High speed parallel decimal multiplication with redundant internal encodings," IEEE Trans. Comput., vol. 62, no. 5, pp. 956–968, May 2013.

[4] 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.

[5] E. M. Schwarz, J. S. Kapernick, and M. F. Cowlishaw, "Decimal floating-point support on the IBM System z10 processor," IBM J. Res. Develop., vol. 51, no. 1, pp. 4:1–4:10, Jan./Feb. 2009.

AUTHOR's PROFILE

Gorripati sreenivasulu completed his Btech in Audisankara Engineering And Technology ,Gudur, Nellore dt, Ap in 2013. Now pursuing Mtech in Electronics & Communication Engineering in SKR College of Engineering & Technology, Manubolu

B Doraswamy , received his M.Tech degree, currently He is working as an Associate Professor in SKR College of Engineering & Technology, Manubolu

