Binary Coded Decimal Digit Adders and Multipliers Implementation on FPGA Platform

Prof. R. P. Sarnaik (Assistant Prof.), Dr. S. A. Ladhake (Principal),

Prof. P. D. Gawande (Associate Prof.)

Department of Electronics & Telecommunication
Sipna College of Engineering & Technology

Abstract

In computing and electronic systems, binary-coded decimal (BCD) is a class of binary encodings of decimal numbers where each decimal digit is represented by a fixed number of bits. Decimal arithmetic has been in recent years a subject of several studies but it consumed more power and time. In this paper, we are surveying different methods for BCD digit adders and multipliers which will consist less area, low power and high speed performance. BCD's main virtue is a more accurate representation and rounding of decimal quantities as well as an ease of conversion into human-readable representations. Decimal arithmetic has gained high impact on the overall performance of today's financial and commercial applications. Decimal digit adders and decimal digit multipliers are usually the building blocks for higher order decimal adders and multipliers. FPGAs provide an efficient hardware platform that can be employed for accelerating decimal algorithms.

Introduction

Binary calculations are not suitable for commercial, banking, and business applications due to the unacceptable inexact decimal to binary conversion errors they produce. In [12] a real example shows the extreme effect of these wrong approximations, where it stated that if a communication company approximates a 5% sales tax on an item (such as a $0.70 telephone call), the yearly loss is over than a $5 million. Software solution to the fractional approximation problem for decimal arithmetic is somewhat slower one. Compared to hardware speeds, the performance of existing decimal arithmetic software libraries is very poor. Software solution is slower than a hardware implementation by 100 to 1000 times [4].

Nowadays, decimal arithmetic is implemented using software while binary arithmetic is usually implemented by the hardware. Further, the survey showed that 55% of the numeric data columns, used by 51 major organization's databases, were decimal data types and 43.7% were integer types which could have been stored as decimals. In spite of this, currently, decimal floating-point arithmetic is not supported by any microprocessors. Decimal floating-point coprocessor could be included in the machines that handle these calculations to speed up these applications.

In order to convey the growing evolution of the decimal arithmetic, efficient decimal algorithms have to be investigated. Decimal digit adders and decimal digit multipliers are key components of any decimal hardware to support decimal arithmetic applications. Therefore, efficient BCD digit units to be used in high performance decimal hardware accelerators. Two main contributions of this work can be highlighted: proposing two new BCD digit adders and proposing one new BCD digit multiplier. These designs are described and simulated using VHDL hardware description language and implemented on an FPGA.

Related Work

In general for binary addition the long carry chain of BCD adder slows down the addition operation. The decimal addition is having delay problem if BCD range exceeds. To produce BCD digit product multiplier & multiplicand used a Look Up Table. But for wide range of digits these Look Up Table is inefficient.

To improve the BCD adders and BCD multipliers speed as well as inefficiency, designers have proposed several enhancements as follows

1. Direct decimal addition provides fast operation [1]. In this, the designer provides technique for new high-speed area-optimized correction-free BCD digit adders that can be employed in different decimal applications.

2. Decimal speculative addition gives efficient binary multiplication [3]. In this, the linearity of carry propagation interferes with parallel execution using a tree representation.

3. Conditional speculative decimal addition having low latency and less hardware requirements [6]. Designer preferred the (*Q-T*) carry tree implementation over their parallel prefix carry tree implementation for their proposed adder.

For high performance decimal floating point units are used to improve speed and Optimize area using recoding algorithm by generating partial products. The parallel generation of partial products is performed using signed digit radix-10 or radix-5 recodings of multipliers [6] & [8]. The partial products is implemented in a tree structure based on a decimal multi operand carry-save addition algorithm that uses unconventional (non BCD) decimal-coded number systems.
The proposed a parallel serial decimal multiplier architecture which provides parallel serial proposal to reduce complexity and it exploits overlapping update to speed up pipeline [7]. The values of the Digit Products in the successive columns of the product array are added in binary and converted in decimal. Their decimal alignment generates a set of three or four serial decimal numbers whose sum is the product.
We are using Direct Boolean expressions BCD digit adder's major approach which is the fastest among other proposed adders [1]. The proposed minimal-area BCD digit adder increases the speed by applying correction-free addition techniques [11]. Also two new BCD digit adders and one new BCD digit multiplier are included in paper for the purpose of speeding up decimal arithmetic applications over FPGAs [8] & [9].

A. Configuration 1: Direct Boolean Expressions BCD Digit Adder

Design a direct BCD digit adder using a nine bit input, five bit output combinational logic. The nine bit inputs are the two BCD input digits A and B plus the decimal carry input cin and the five bit outputs are the BCD digit of the decimal sum S plus the decimal carry out cout. For example, to add (6 + 7 = 13), this operation is translated to (0110 + 0111 = 10011)BCD. The output result (10011)BCD is the BCD representation of the decimal number 13. The most significant bit is the decimal carry output generated from the addition operation, while the other bits are the BCD summation digit. So it becomes a correction-free technique. The truth table for all output logic functions is constructed for all possible combinations of the inputs. Since the inputs are nine bits, the number of possible combinations is $2^9 = 512$. Many of these combinations are not valid since a decimal digit is less than (10)10, while 4-bit number can take any value from 0 to (15)10. In the case when the input is not valid, the output is set to don't care. Table I consist the two input BCD digits are assumed to be A = a3a2a1a0 and B = b3b2b1b0. The output consists of the BCD digit sum S = s3s2s1s0 and the decimal carry output.

Table I: Adder Configuration 1

Inputs			Outputs	
C_{in}	$a_3\,a_2\,a_1\,a_0$	$b_3\,b_2\,b_1\,b_0$	C_{out}	$s_3\,s_2\,s_1\,s_0$
0	0110	1000	1	0100
1	1001	1001	1	1001
0	1100	0001	-	----

B. Configuration 2: Minimal-area BCD Digit Adder

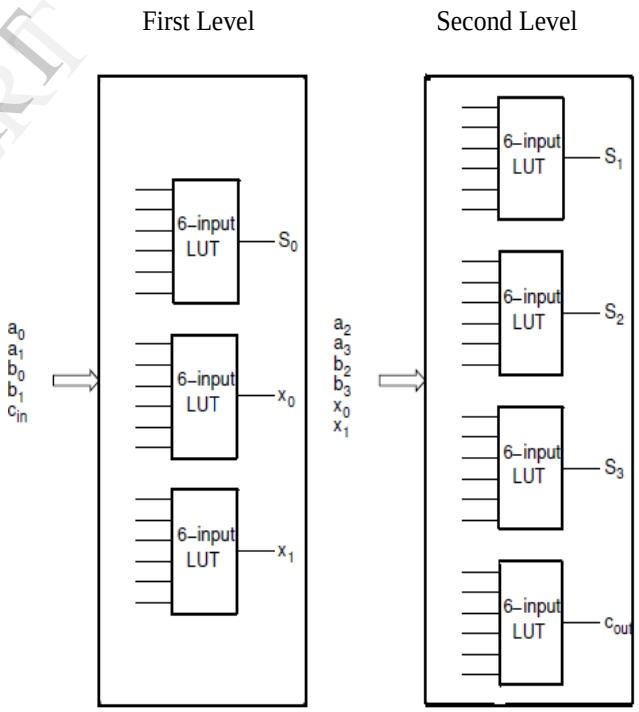


Fig.1 Proposed BCD Digit Adder

The internal architecture of the FPGA device is used in the synthesis/implementation to optimized speed/area. FPGA has 6-input look-up tables and function with six or less inputs, will occupy exactly one look-up table [11].The previously proposed BCD adder have at least nine-input sub-architectures, which complicates the overall design and increases its area and speed. To overcome it, we have proposed a two levels BCD digit adder. The first level has five

inputs and three outputs, and each output requires just one look-up table. The second level has six inputs (two of them from the first level) and four outputs. Therefore, four look-up tables are needed for this stage. The overall occupied area is seven look-up tables, which is same as the traditional BCD digit adder's area.

The first level adds the first two least significant bits of A and B (a1a0, and b1b0), along with the carry input cin, and produces three bits: x1, x0, and the least significant bit of the final BCD summation digit S0. The two most significant temporary bits (x1, x0), are used along with the two most significant bits of two BCD input digits A and B (a3a2, and b3b2) as inputs to the second level. The second level adds the previous inputs and produces the three most significant bits of the final BCD summation digit (S3S2S1), and the final decimal carry output cout. It is notable that the most significant two bits from the first level (x1, x0) are inputs to the the second level, since they are involved in the correction operation. Actually, there is no direct correction in this addition approach.

C. Direct Boolean Expressions BCD Digit Multiplier

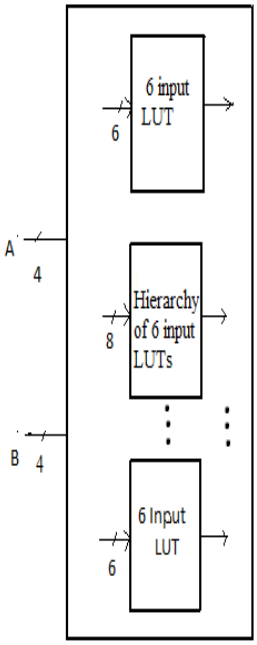


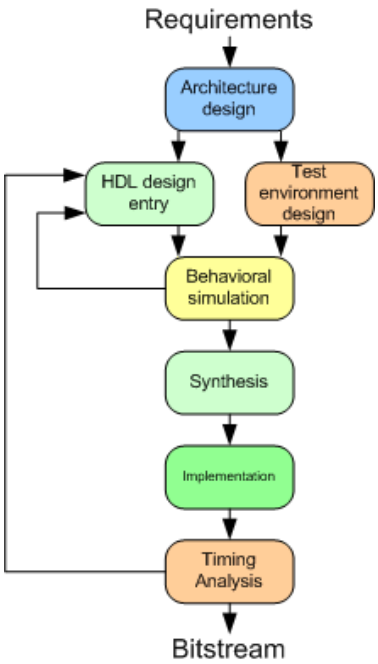
Fig.2 BCD Digit Multiplier

In proposed BCD digit multiplier, word "direct" means no need for neither "first finding the binary multiplication result and then converting the product to a BCD form" [9] nor "any recoding process" [5].Instead, we have used a simplified Boolean expressions to perform the "direct" functionality. In this case, the two operands are two decimal digits A = a3a2a1a0 and B = b3b2b1b0 and the output P = A × B is 8 bit P7P6P5P4P3P2P1P0 (two BCD digits). Since the input is 8 bits wide, the number of combinations in the truth table is 2^8 = 256. Among all these combinations only 100 combinations are valid and the rest are invalid. All outputs for the invalid combinations in the truth table are set to don't care. If the output functions depend on more than six variables from the input variable then it needs hierarchy of LUTs to be implemented. An example on this is P1 which depends on all the eight input variables. On the other hand, some functions depend only on two variables like P0 or four variables like P7. This means that these two outputs consumes a single 6-input LUT each.

Table II : Multiplier Configuration 1

Inputs		Outputs	
$a_3\,a_2\,a_1\,a_0$	$b_3\,b_2\,b_1\,b_0$	$P_7\,P_6\,P_5\,P_4$	$P_3\,P_2\,P_1\,P_0$
0110	1000	0100	1000
1001	1001	1000	0001
1100	0001	----	----

Flow Chart



Conclusion

The different designers have proposed several enhancements for high speed area optimization .But, In this paper, We have studied and implemented Direct Decimal Addition approach

for this. Using this approach the efficient binary multiplication is obtained using Decimal Speculative Addition. The BCD digit adders are used which provides correction-free addition techniques. In this paper, two new BCD digit adders and one new BCD digit multipliers are designed to speed up decimal arithmetic applications over FPGA.

References

[1]. M. S. Schmookler and A. Weinberger, "High speed decimal addition," *IEEE Transactions on Computers*, vol. 20, pp. 862–866, 1971.

[2]. R. H. Larson, "Medium speed multiply," *IBM Technical Disclosure Bulletin*, p. 2055, 1973.

[3]. H. Wetter W. Bultmann, W. Haller and A. Worner, "Binary and decimal adder unit," 2001.

[4]. M.F. Cowlishaw, "Decimal Floating-Point: Algorism for Computers," Proc. 16th IEEE Symp. Computer Arithmetic, pp. 104-111, July 2003.

[5]. E. M. Schwarz, "Decimal multiplication with efficient partial product generation," in *Proceedings of the 17th IEEE Symposium on Computer Arithmetic*, Washington, DC, USA, 2005, ARITH '05, pp. 21–28, IEEE Computer Society.

[6]. A. V´azquez and EAntelo, "Conditional speculative decimal addition," Nancy, France, 2006.

[7]. L. Dadda. Parallel decimal multipliers: a hybrid approach. *AlaRI Internal Report*, 2006.

[8]. A. Va´zquez, 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.

[9]. G. Jaberipur and A. Kaivani, "Binary-coded decimal digit multipliers," *IET Computers and Digital Techniques*, vol. 1, no. 4, pp. 377–381, 2007.

[10]. N. H. Tuli´c, "Fpga implementations of decimal arithmetic cells," M.S. thesis, Jordan University of Science and Technology, May 2009.

[11].Inc.xilinx,"Vertix5datasheet",http://www.xilinx.com/support/documentation/datasheets/ds100.pdf.

[12].IBM Corporation, "Decimal FAQ," http://www2.hursley.ibm.com/ decimal/decifaq1.html.