Design of Addition/Subtraction for BIN/BCD Numbers

Tara Tavakoli, Nastaran Parvin, Safiyeh Nikkhahsani, and Seyed Reza Talebiyan

***Abstract*—This paper aims to find a solution for a new design of binary / BCD adder / subtraction to increase the speed of operations and decrease the delays in signed numbers as well as the unsigned ones. A very important operation in mathematics for digital systems is adding binary/BCD numbers. Many different ways have been presented for addition and subtraction of BCD,this paper has presents no signbit numbers adder and subtractor many papers presented adder and subtractor for binary BCD numbers but all of them have some problems with their final assessment in this paper this problem is solved by adding EOP (end of operation signal).**

Here, a new design for addition or subtraction has been offered, without taking the signbit in BCD format that works perfectly.

***Index Terms*—Adder, subtractor, signbit, condition, Restructuring , EOP (end of operation signal).**

I. Introduction

One of the very important operations in digital systems is adding. A lot of digital systems use binary adding in their operations, but particularly systems with the interest of 10, prefer using BCD. The point is, BCD is more complicated than binary adding and the adders of BCD have a wider scale [1], [2].

Different high-speed adders like CARRY LOOK AHEAD and CARRY SAVE ADDER are used in BCD of [3]-[5] in order to make their circuit work more properly, while in [6] another type of adder is used with a new ability of restructuring and with this, the adder is capable of adding and subtracting BCD practically. In this paper the sign bit of BCD numbers are considered without any importance, and in [6], the restructuring ability is used which causes an output signal called EOP, which only works at the end of the operation.

The signal has not been working properly and may confuse users later on. This paper has tried to clarify the final computation by using the ability of restructuring for BCD/binary and without considering any sign for adding/subtraction. The paper has also discussed the design in [6] and the problems about it below.

II. The Mathematical Background and the Binary Numbers Definition

A. Definition of BCD

The amount of BCD with 10 as the basis of interest is directly used here. Based on their locations, binary bits can take a value or mount. This is the code 1,2,4,8 for BCD. Number 6 decimal is shown by the code 0110, for instance, because $0\times8+1\times4+1\times2+0\times1=6$, and the decimal code can take negative values as well, like: -1, -2, 4, 8. In this situation with the code 0110, number 2 is calculated like next page:

$$0\times8+1\times4+1\times(-2)+0\times(-1)=2$$

B. The Addition of BCD Numbers

In the addition of two decimal numbers, the number before with a less value must be chosen from the pair, considering the possible carry, because the result number will not be more than the amount of 9; and it means the addition is 1+9+9=19 and no more, where the carry before is 1. Operating the same addition in BCD, the number will be from 0000 to 1001, where the carry, the first number, and the continue of the addition in BCD is 1. The operation is the same for the addition of two BCD numbers with the digit numbers of n.

C. Subtracting BCD Numbers

Basically, BCD numbers are decimal and only the illustration of them is different. So, to subtract two BCD numbers one called N and the other M, the second 9's or 10's complement is to be found.

Basically, BCD numbers are decimal and only the illustration of them is different. So, to subtract two BCD numbers one called N and the other M, the second 9's or 10's complement is to be found.

$$\text{EOP} = \left[\text{bin}\left((Xn\odot Yn)\oplus\text{op}\right)\right] + [\text{bin}'.\text{op}']$$

III. Performance of the Presented Circuit and Its Explanation

A. In Case of Binary

In case of $ingned\ magnitude\ n+1\ bit$ X and Y are two numbers, where $X=[X_nX_{n-1}X_{n-2}\dots X_0$ and $Y=[Y_nY_{n-1}Y_{n-2}\dots Y_0$, in which X_n and Y_n , then the signed bit of the numbers are binary and BCD. In this design, some input signals known as bin, EOP and op are considered. op is the identification of the operations of adding and subtraction (op=0 in input data shows the operation of adding, and op=1 show the operation of subtraction). Here, bin is the variable by which type of the operation done on the basis of the binary or BCD is determined. Finally, it is the EOP (END OPERATION) that decides for the type of operation used in the final part, and the sign bit is also considered; where EOP=0 shows the operation of subtraction and EOP=1 shows adding

When the operation of EOP is found through the equation

Manuscript received September 30, 2015; revised December 31, 2015.
The authors are with Imamreza internationaluniversity, Mashhad, Iran (e-mail: tara.tavakkoli@gmail.com).

of EOP, the first operand sing $Xi.e.Xn$ and carry-out are used to calculate the result sign in the binary system. If the effective operation in the final part is adding, the sign is first operand sign, which is $i.eX0$, but the result sign will be dependent on X sign and the carry-out circuit of the adder, if the operation is subtraction. The final result sign for adding will be $Sn = Xn\, if EOP =' 1' \& bin = 1\, i.e.$ and for subtraction will be $Sn = Xn \oplus (cout) if EOP =' 0' \& bin = 1$ if $X > Y$ and $\leq Y$. For addition and subtraction in BCD, the final result sign will be zero; considering no value for sign bit in BCD numbers. (shown in figure next colum)

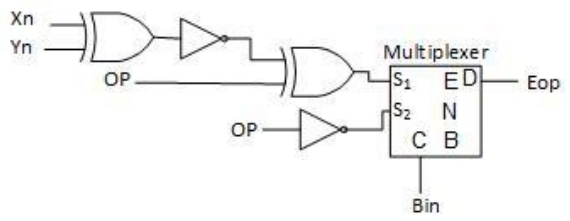


Fig. 1. Eop design.

TABLE I : EXPLANATION OF EOP

X	Y	(	op (add)	op (sub)	Add	Sub
0	0	1	0	1	1	0
0	1	0	0	1	0	1
1	0	0	0	1	0	1
1	1	1	0	1	1	0

B. *The Suggested Circuit Functioning in Decimal Case*

Creating and employing the adder used in [6] and this study is centered on the structure of Sklansky [7] adder.

We will argue the addition and deduction of BCD numbers deprived of any signs, and bring examples for both conditions. We must consider that, in the process of adding BCD numbers, if there exists another level for each carry lever, then the correction block operation will be done; on the opposite if no next level exists, no operation would be needed; that means the carry is transferred via one level and not from one to another.

$X = 547$ In BCD Format 010101000111

$Y = 328$ In BCD Format 001100101000

Afterwards, initially the amount of X NEW is made and then the addition of BCD will be done in the conditions that were said.

X New $= X + 6$

$010101000111 + 011001100110 = 101110101101$

X New 101110101101

$+$ Y 001100101000

111011010101

$+$ 10101010

100001110101

For both cases of $X>Y$ and$\leq Y$, subtraction is tested. The situation here is the transference of the carry from one level to another, if $>y$, the subtraction is binary (Subtraction = X+(-Y) where $-Y$ is Y NEW), and at the start 1's complement determines Y, and as declared in [6], the subtraction is done. View the procedure in the example below.

The carry out is produced in the last level, so the number from the addition is counted up with 1, and next, the number from subtraction is counted up with 6, to change the outcome to BCD.

Let's take a glimpse at the example of subtraction when $X \leq Y$:

Carry out is not built in the last level, so the result number must be reversed (NOT) to determine the subtraction result in binary, and then once more, add it up with 6 to change it to BCD.

$X = 547$ In BCD Format 010101000111

$Y = 328$ In BCD Format 001100101000

Y New = Ones compelemet Y

Y New = 110011010111

1 1 0

$+$ X = 010101000111

Y New = 110011010111

001000011110

1

$+$ 1

001000011111

1

$+$ 1010

001000011001

$X = 328$ In BCD Format 001100101000

$Y = 547$ In BCD Format 010101000111

Y New = Ones compelemet Y

Y New = 101001111000

$+$ X = 001100101000

110111100000

(NOT) 001000011111

$+$ 1010

001000011001

IV. THE RECOMMENDED ARCHITECTURAL DESIGN

The recommended architecture contains different parts such as: pre-correct X and Y "the impression of pre-correct is taken from [6]," post-correct and adder, and restructuring. All design parts are analyzed later.

A. *The Logical Expression of Designing Sign Bit*

Assuming that sign bit for BCD numbers is of meaning (sign bit for BCD numbers is considered zero), the projected

circuit for different values of EOP and bin will function as follows. There exist two probable occasions for sign bit in binary circuit; first is as EOP=0, in which subtraction is done and the sign relies on Cout (which means if Cout= 0, then *X>Y*, and the sign bit is equivalent to the NOT (inversed) of the *X* sign; or else, it would be Cout=1 that demonstrates *X>Y* and then the sign bit is equivalent to the bigger number sign.

In this situation, the bin select line is added to mux lines, that if Bin=0, resembles the subtraction of BCD. In the process of subtraction of these numbers, only smaller numbers are subtracted from the bigger ones and the sign is ignored.

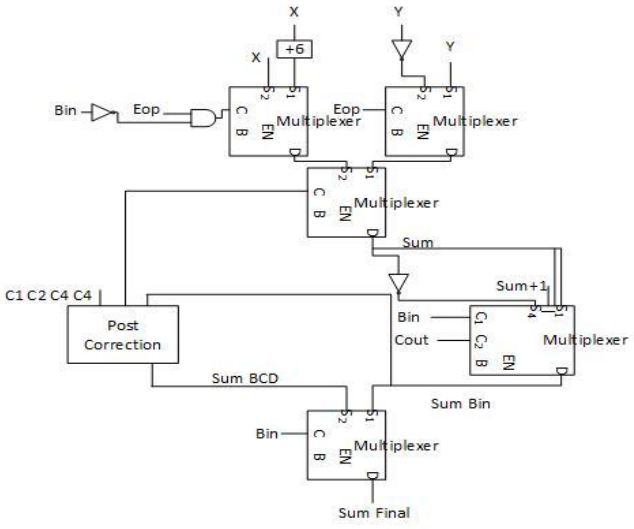


Fig. 2. Final design.

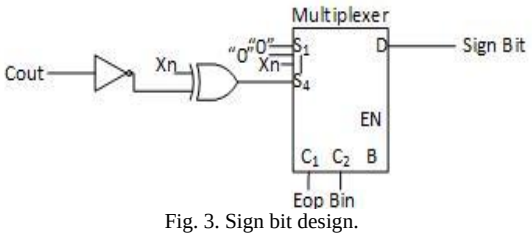


Fig. 3. Sign bit design.

B. The Design of a Mux to Take Addition/Subtraction Is the Idea of Pre-Correction, Since the Binary Addition Relies on the Amount of Cout and Eop, and to Decide Which One to Use Considering to the Situation

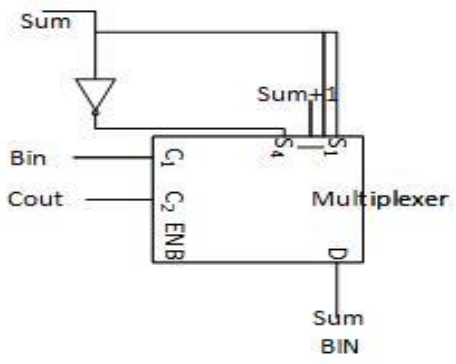


Fig. 4. Pre correction adder/subtractor design.

C. Addition/subtraction Circuit Design after Correction (BCD)

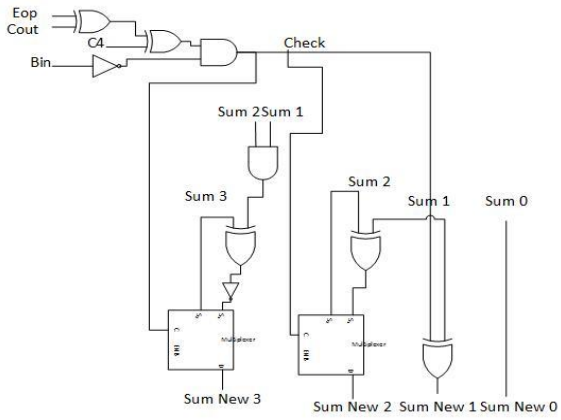


Fig. 5. Post correction BCD design.

TABLE II : EXPLANATION OF SIGN BITEXPLIANTION

EOP	Bin	
0	0	This case displays that the positive number has been considered and the subtraction is BCD and the sign bit is measured zero.
0	1	The subtraction is done on binary numbers and there are two probable instances for sign bit in binary circuit; when EOP=0, that subtraction is done and the sign relies on Cout (which means if Cout= 0, then *X<Y*, and the sign bit is equal to the NOT (inversed) of the *X* sign; else, it would be Cout=1 that shows *X>Y* and then the sign bit is equivalent to the bigger number sign.
1	0	It signifies that the number is BCD, so the sign is pointless in that.
1	1	This connotes that addition should be done and the *Xn* sign will be known as the result.

V . THE SIMULATION

In most of the simulations, usually the XILINX software is used. Here the new design of this circuit is compared with design that presented in [6]. The Delay and also Logic Utilization results are illustrated in Table III.

TABLE III: RESULT FOR DELAY AND LOGIC UTILIZATION

	[6]	MY ARCHITECTURES
DELAY	36.619 ns	35.665ns
Logic Utilization	115	110

VI. CONCLUSION

This paper presented the efficient adder and subtractor unit for binary and BCD numbers, this design can support both sign and no signbit numbers in binary mode and also new ability of restructuring addition and subtraction without considering any signbit in BCD format, unlike in other designs end of operation is clarify with EOP signal but this new structure has better delay and area in camper with [6].

REFERENCES

[1] M. J. Adiletta and V. C. Lamere, "BCD adder circuit, digital equipment corporation," US patent 4805131, pp. 1–18, Jul. 1989.

[2] *Computer Arithmetic Algorithms*, Israel Koren, Pub A K Peters, 2002.

[3] S. Veeramachaneni, K. M. Krishna, G. V. Prateek, S. Subroto, S. Bharat, and M. B. Srinivas, *A Novel Carry-Look Ahead Approach to An UNIFIED BCD and Binary Adder/ Subtractor*, 2008.

[4] O. A. Khaleel, M. A. Khaleel, Z. A. Qudah, C. A. Papachristou, K. Mhaidat, and F. G. Wolff, *Jordan University of Science and Technology, Irbid, Jordan, Fast Binary/Decimal Adder/Subtractor with a Novel Correction-Free BCD Addition*, 2011.

[5] K. J. Lin, J. L. Shih, T. H. Lin, and Y. M. Wang, "A parallel decimal adder with carry," *Correction during Binary Accumulation*, 2012.

[6] S. E. Ahmed, S. Veeramanchaneni, M. N. Muthukrishnan, and M. BSrinivas, *Reconfigurable Adders for Binary*, 2011.

[7] J. Sklansky, "Conditional-sum addition logic," *IRE Trans. Electronic Computers*, vol. EC-9, pp. 226-231, June 1960

[8] M. S. Schmookler and A. W. Weinderger, "High speed, decimal addition," *IEEE Transactions on. Computers*, vol. C-20, pp. 862-867, August 1971.

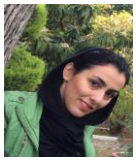
[9] U. Grupe, "Decimal adder," *Vereinigte Flugtechniche Werkefokker gmbH*, US patent 3935438, pp. 1–11, Jan 1976.

[10] M. F. Cowlishaw, "Decimal floating-point: Algorism for computers," in *Proc. 16th IEEE Symp. Computer Arithmetic*, pp. 104-111, 2003.

[11] M. J. Adiletta and V. C. Lamere, "BCD adder circuit, digital equipment corporation," US patent 4805131, pp. 1–18, Jul 1989.

[12] D. R. H. Calderon, G. N. Gaydadjiev, and S. Vassiliadis, "Reconfigurable universal adder," in *Proc. the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, pp. 186-191, July 2007.

[13] W. Haller, U. Krauch, and H. Wetter, "Combined binary/decimal adder unit," *International Business Machines Corporation*, US patent 5928319, pp. 1-9, Ju11999.



Tara Tavakoli was born in Mashhad, Iran on September 14, 1987. She got her associates degree of software engineering in 2008 from Khayyam University of Mashhad, Iran; her bachelors degree of hardware engineering in 2011 from Khavaran University of Mashhad, Iran; and her masters degree of architecture of computer in 2014 from Imam Reza International University of Mashhad, Iran.