# A VLSI Implementation of Fast Binary to BCD Convertor using Complement Based logic design (CBLD)

[1]S.Farhana, [2]K.Sireesha, [3]P.Shanthamma
[1,2,3]Assistant Professor, Department of Electronics & Communication Engineering,
Annamacharya Institute of Technology & Sciences, Kadapa, Andhra Pradesh.

**ABSTRACT:**

Financial and commercial applications use decimal data and spend most of their time in decimal arithmetic. Software implementation of decimal arithmetic is typically at least 100 times slower than binary arithmetic implemented in hardware. High performance low power arithmetic circuit with least area, is essential for advanced processes. This paper proposes a new view of logic design with the name of Complement Based Logic Design (CBLD) for modeling and optimization of some building blocks at gate level. All these optimization states with the use of CBLD leads to decreasing the no of logic gates, and improving the speed. The efficiency of this technique is demonstrated by applying CBLD on some of the arithmetic building blocks which have a potential of the optimization. Then a fast Binary to BCD converter is designed by using this CBLD logic. This CBLD design, when compared with conventional approach shows significant area, and speed improvement.

**Keywords:** Binary to BCD Convertor, Decimal arithmetic, Fast and area efficient Logic Design, CBLD.

---

## I. INTRODUCTION

Decimal Arithmetic is receiving significant attention in commercial business and internet based applications, hence providing hardware support in this direction is necessary. Recognizing the need for decimal arithmetic has led to specifications for decimal floating point arithmetic to be added in the draft revision of the IEEE-P754 standard [1]. Decimal arithmetic operations are generally slow and complex, its hardware occupies more area. They are typically implemented using iterative approaches or lookup table based reduction schemes. This has led to the motivation behind improving BCD architectures, to enable faster and compact arithmetic.

BCD is a decimal representation of a number directly coded in binary, digit by digit. For example, the number (9321)10 = (1001 0011 0010 0001) BCD. It can be seen that each digit of the decimal number is coded in binary and then concatenated to form the BCD representation of the decimal number.

There are many methods to realize the decimal multiplication in hardware. For example, one method for decimal multiplication is to multiply in decimal directly. Another method suggests converting the operands to binary then multiply and finally convert the result back to decimal. The third method for decimal multiplication is doing the decimal digit by digit multiplication in binary and then turning the binary partial products into decimal and adding them appropriately to form the final result.For this technique, the Binary to BCD converter module performance is critical. The binary to BCD conversion is generally inefficient if the binary number is very large.

In this paper we introduce a new logic which is called Complement based logic design which is fast and area efficient for designing Binary to BCD converters, by modifying the conditional adder blocks architecture of shift Add by constant algorithm which makes it faster, area-efficient.

Topics in the paper are structured as follows: Section II discusses an overview of prior work in binary to BCD conversion done by using the Shift Add by const algorithm. Section III explains the Proposed Complementary Based Logic Design. Section IV gives modified design for shit add by const algorithm for the Binary to BCD convertor reported, this is further extended to 8-bit binary to BCD conversion. Section V gives the simulation results and finally Section VI concludes the paper.

## II. PRIOR WORK

Extensive work has already been done in the field of binary to BCD conversion. The conversion circuit for 4-bit is shown in fig.1. For the given 4- bit input, if the 3 most significant value is greater than 4 then three

is added based on this logic we get a 4 bit bcd output as illustrated in fig.1. The 3-bit adder block adds 3 or 0 based on the other input it gets only if the value is greater than 4.

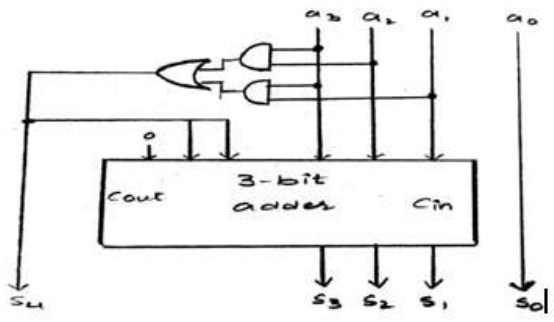


*Fig.1: Traditional 4 bit binary to bcd convertor*

## III. PROPOSED COMPLEMENTARY BASED LOGIC DESIGN:

This section gives briefing of CBL design methodology. Considering a conditional operation which performs addition of a constant number, or passing the input without change to the output. Control functions can control each input for propagation to the output like buffer or complement (Fig. 2).

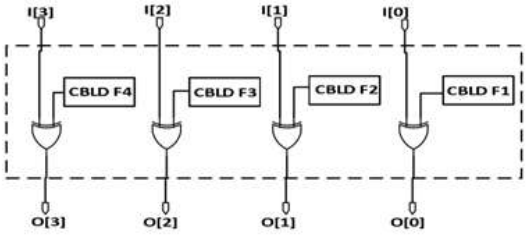


*Fig.2: CBLD Methodology block diagram*

In CBLD methodology, property of XOR is used.

If f[i] = 1 then o[0] = i[O]'

IF f[i] = 0 then o[0] = i[0]

Let's consider 2bit number is added by binary "1" or by "0" depending on control signal X. Consider "1" is added, if X=1 and "0" is added, if X=0. Table 1 represent 2-bit binary add by "1". In this table outputs are defined as complementary based on their inputs.

Table 1: 2-Bit add by one.

| S. no | Input $I_1$ $I_0$ | Output $O_1$ $O_0$ | O1 $O_1 = F(I_1)$ | O0 $O_0 = F(I_0)$ |
|---|---|---|---|---|
| 01. | 0 0 | 0 1 | $I_1$ | $\bar{I}0$ |
| 02. | 0 1 | 1 0 | $\bar{I}_1$ | $\bar{I}0$ |
| 03. | 1 0 | 1 1 | $I_1$ | $\bar{I}0$ |
| 04 | 1 1 | 0 0 | $\bar{I}_1$ | $\bar{I}0$ |

Fig.3.A shows the implementation of the 2-bit

conditional add by 1, and Fig.3.B shows traditional implementation of this circuit with MUXs and two bit binary adder block.

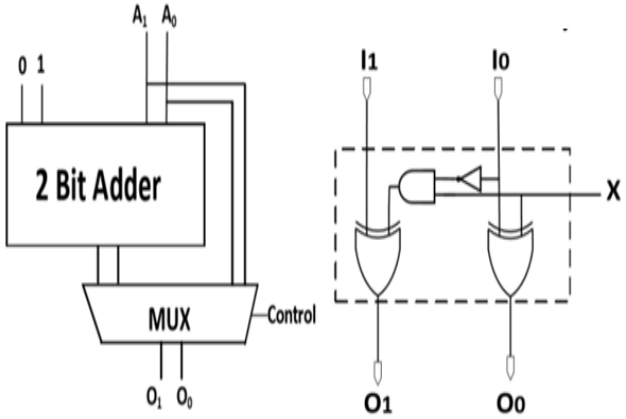


*Fig 3. 2 bit Add by one (a) conditional CBLD module (b) Traditional approach*

**Proposed CBLD Algorithm For Design Of Binary To BCD Convertor:**

Complement Based Logic Design has been introduced in [11] as a new methodology for optimization or redesigning of the arithmetic building blocks in order of decreasing number of logic gates as the resulting improvement of the circuit characteristics. The implementation of the 4bit binary to BCD convertor in fig.1 using CBLD is shown in fig.3.

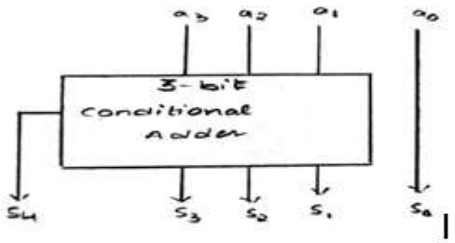


*Fig. 3 4bit binary to BCD convertor using CBLD*

Though the shift add by 3 algorithm is not novel, the architectural implementation using add by constant which makes it area efficient

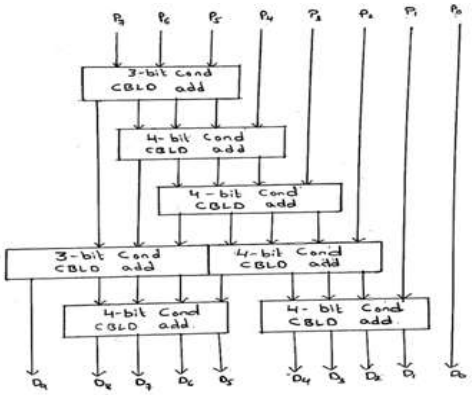


*Fig. 4: 8-bit Binary to BCD convertor using shift add by const.*

As shown in Fig. 4, this architecture has two different modules. (i) 3-bit conditional Adder block.(ii) 4-bit conditional Adder block.

The internal blocks of fig.4 are shown in fig 5,6.

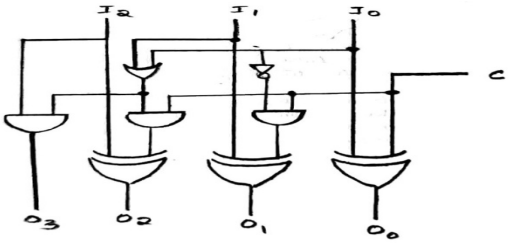


*Fig. 5: 3-bit conditional adder block*

As shown in the figure 5 we verify that if the value of first 3 bits is greater than $(4)_{10}$ if so we will add plus three or else the bits will propagate without any further change.

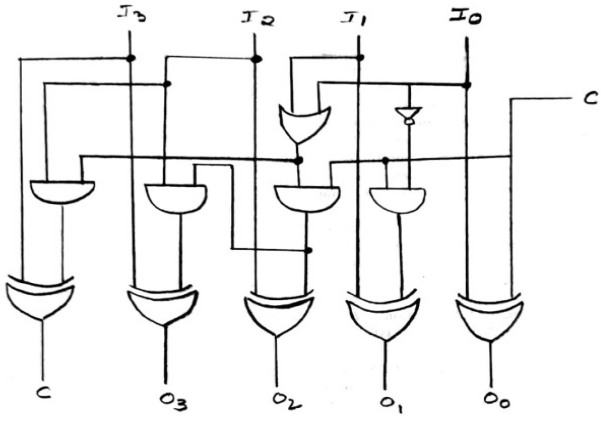


*Fig. 6: 4-bit conditional adder block*

As shown in the Figures 6, we compare that if the value of next 4 bits is greater than $(4)_{10}$ if so, we will add plus three to the bits or else the bits will be propagated without any further change in the output.

**IV.SIMULATION RESULTS:**

All the architectures are described using Verilog HDL data flow and are verified using Xilinx Isim simulator. All designs are synthesized using Synopsis design compiler with standard cell Library.

From the results we can observe that when compared to traditional binary to BCD convertor and CBLD based binary to BCD convertor it is found that the CBLD design works faster and also can be implemented with less no of logic gates which in turn reduces a large amount of area.

From the results we can interpret that: the gate count of 8-bit binary to BCD convertor when implemented traditional using shift add by const is of total 151. But when this is designed with CBLD logic the no of logic gates required is only 71. Therefore we have reduced a significant amount of area and speed is also improved.

**CONCLUSION:**

This paper is proposed a new logic design for optimization of some arithmetic building blocks and design of multi-input multi-output combinational circuits. First, the efficiency of CBLD is proved with the optimization of some arithmetic blocks, then this technique is used for the design of the new architecture of Binary to BCD converter.By comparing the simulation results of the proposed CBLD designs with the conventional ones , we can say that the Proposed Architectural implementation of Shift-add by Constant has an improvement of 22%-138.7% on area and faster.

**REFERENCES**


[1] Schmookler, M.: 'High-speed binary-to-decimal conversion', IEEE Transaction on Computers, C-17, Issue 5, pp. 506–508, 1968.

[2] M.P. Veandstias and H.C. Neto. "Parallel decimal multipliers using binary multipliers", IEEE in Programmable Logic Conference (SPL), VI Southern, pp. 73 –78, 2010.

[3] S. Veeramachaneni and M.B. Srinivas. "Novel high-speed architecture for 32-bit binary coded decimal (bcd) multiplier", IEEE, International Symposium on Communications and Information Technologies, ISCIT pp. 543 – 546, 2008.

[4] Sri Rathan Rangisetti, Ashish Joshi, Tooraj Nikoubin, "Area- Efficient and Power-Efficient Binary to BCD Converters", IEEE, Sixth International Conference on Computing, Communications and Networking Technologies 6th ICCCNT– 35239, Denton, U.S.A, July 13 - 15, 2015.

[5] Arvind Kumar Mehta, Mukesh Gupta, Vipin Jain, Sudhir kumar." High Performance Vedic BCD Multiplier and Modified Binary to BCD Converter". IEEE Annual India Conference (INDICON), pp. 1 – 6 2013.

[6] Osama Al-Khaleel, Zakaria Al-Qudah and Mohammad Al- Khaleel, "Fast and compact binary-to-BCD conversion circuits for decimal multiplication," IEEE 29th International Conf. on Computer Design, pp. 226 – 231, Oct. 2011.

[7] J. Bhattacharya, A. Gupta, and A. Singh. "A high performance binary to BCD converter for decimal multiplication". IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT), pp. 315 – 318, 2010.

[8] Tso-Bing Juang,Yu-Ming Chiu."Fast Binary to BCD Converters for Decimal Communications Using New Recoding Circuits". IEEE International Symposium on Integrated Circuits (ISIC), pp.188 – 191, 2014.