ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

High Speed 128-bit BCD Adder Architecture Using CLA

J.S.V.Sai Prasanthi [1], Y.Yamini Devi[2]

PG Student [VLSI&ES], Dept. of ECE, Swamy Vivekananda Engineering College, Kalavarai, Andhrapradesh, India[1]

Assistant Professor, Dept. of ECE, Swamy Vivekanada Engineering College, Kalavarai, Andhrapradesh, India[2]

ABSTRACT: Arithmetic and memory address computation are performed using adder operations. Hence, design of adders forms an important subset of electronic chip design functionality. BCD numbers play a prominent role in number system. To perform arithmetic operations on BCD numbers respective circuit has to be designed. To perform BCD addition, BCD adders are used. But the drawback with the BCD adder is low speed in operation due to delay in propagating carry output. This low speed operation will affect the operation of entire system in which it is used. As the technology is advancing day by day there is demand for chips with high speed. So to overcome this drawback, BCD adder using CLA is proposed in this paper. The proposed design is attempted here to reduce the delay and thereby increasing the speed of response. In existing BCD architecture, RCA is used to add numbers. The delay of RCA is high so it is effecting the speed of adder. So in the proposed design, CLA is used instead of RCA and also a parallel prefix network is to be used to produce the carry outputs for all stages. In this paper, a BCD adder using CLA is to be designed for 8, 16, 32, 64 and 128-bit size using VHDL with the help of ISE Xilinx design suite 14.1. The designed adder will be functionally verified by using ISIM simulator. Later, it will be synthesized using XST synthesizer to get the area (in terms of LUTS) and delay(ns). Finally, the designed BCD adders will be compared with conventional BCD adder in terms of delay(ns).

KEYWORDS: BCD Adder, RCA, CLA, parallel prefix network, ISE Xilinx design suite14.1, ISIM simulator, XST synthesizer

I.INTRODUCTION

Human beings have preferred decimal as their number base for all calculations done by hand, since the time when the man learned to count on his ten fingers. This fact has never changed, although binary has been selected as the default base for almost all computers due to the storage and the speed efficiency of binary hardware. The success of binary numbers was introduced in 1946, by the report of John von Neumann and his colleagues at the Institute for Advanced Study [1]. Afterwards, the designers have preferred binary computers due to the speed and the simplicity of binary arithmetic, but nowadays, there is an increasing demand for the decimal arithmetic hardware support in financial and commercial applications.

In all arithmetic units, whether binary or decimal, an adder is used [2]. Therefore, it is not surprising that various addition techniques have been invented up to now, even for the decimal addition, which is much less popular than the binary addition BCD is very common in electronic system where a numeric value is to be displayed, especially in systems consisting solely of digital logic, and not containing a microprocessor.

BCD adder is used to perform addition of two BCD numbers. Addition is the important arithmetic operation which is to be performed by the processor. The speed of the addition operation influences the speed of the processor. So adder performance must be high in order to increase the performance of processor and also of entire system. While performing addition operation using BCD adder, there is a delay in propagation of carry output. This delay will affect the speed of the adder which in turn affects the speed of the entire system in which it is used [3], [4], [5]. So there is a need to design the BCD adder with less delay in order to increase the speed of the operation. In this paper, a BCD adder is designed with less delay by modifying the present architecture. And also the proposed BCD adder will be compared with the existing BCD adder in terms of delay (ns).

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

This paper is structured in brief as Follows-Section II explains about the BCD addition algorithm and Section III describes the structure and working of conventional BCD adder (nothing but present existing BCD adder). Section IV discusses about the proposed BCD adder structure by using CLA. The results are explained in section V Finally, the paper ends with conclusion and future scope.

II.BCD ADDITION ALGORITHM

BCD or Binary Coded Decimal is the number system which has the binary numbers or digits to represent decimal number. A decimal number contains 10 digits (0-9). Now the equivalent binary numbers can be found out of these 10 decimal numbers. Like other number systems BCD operation may be required. BCD is a numerical code which has several rules for addition [6]. The rules are given below in three steps.

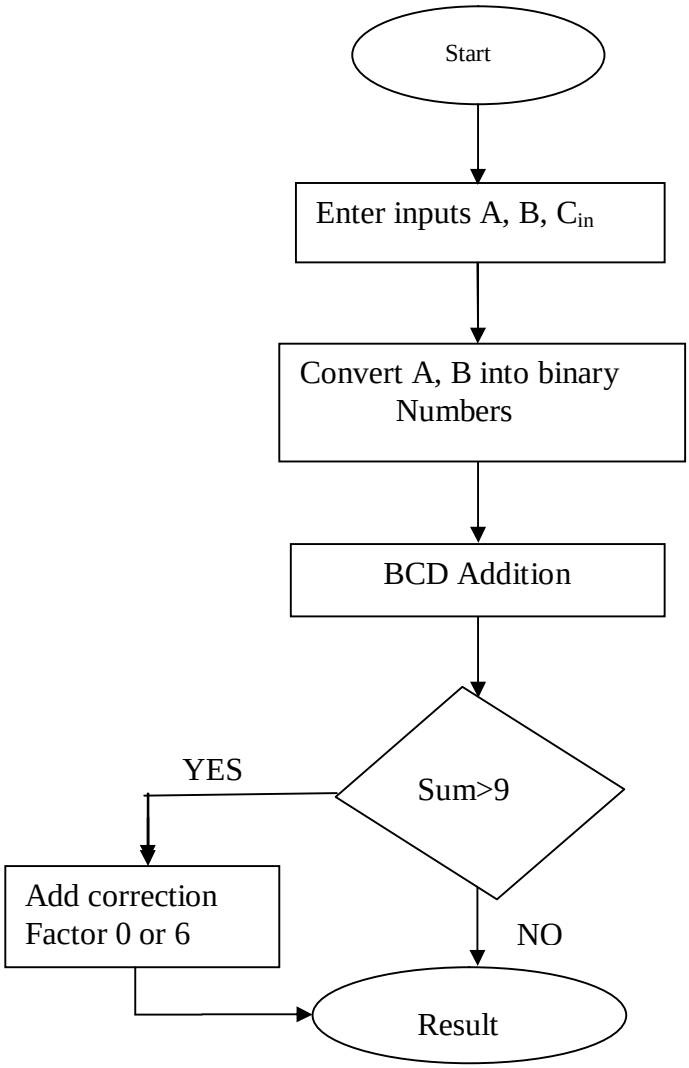

Fig. 1 Flowchart for BCD addition algorithm

- If the sum of the two digits is smaller than 9 there is no carry even though there is a carry
- If the sum of the two digits is greater than 9 a correction is required

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

- If the sum of the two digits is exactly 9 the input carry determines whether a correction is required and a carry Output is produced. Figure 1 represents the algorithm for BCD addition.

III.CONVENTIONAL BCD ARCHITECTURE

Figure 3 shows a conventional BCD adder consists of 4-bit binary adder and carry out circuit. The top module of the 4-bit binary adder act as the ripple carry adder .it consists of series of full adder. The middle module of the carryout circuit consists of OR gate and 2 AND gates. The bottom module of the 4-bit binary adder act as ripple carry adder. Ripple Carry adder consist of series of one half adder and number of full adders. The carryout of one stage is fed directly to the carry-in of the next stage. Cout is used for the correction factor. The first four level BCD adders produce the binary addition. If the result is greater than '9' a carry output is produced. The result of first level 4-bit adder is corrected by adding '6' and also this carry output is used as carry input for the next digit. The 1-bit full adder is the building block of all the modules. The output of full adder is sum and carry output. The binary adder is made up from standard AND and OR gates and allow us to add together single bit binary numbers A and B produce the outputs the sum of the addition and a carry called the carry out (c out bit). The full adder gate level circuit is as shown in fig .3. The first four level BCD adders produce the binary addition. If the result is greater than '9' a carry output is produced. The result of first level 4-bit adder is corrected by adding '6' and also this carry output is used as carry input for the next digit.

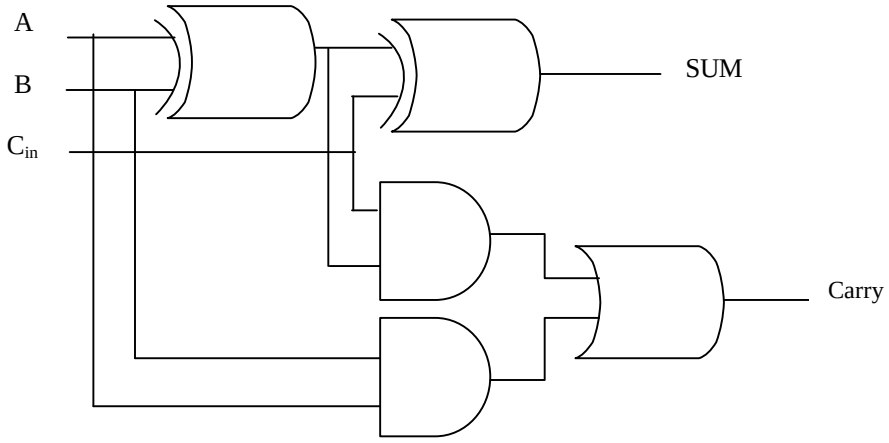

Fig. 2 Full adder

The Boolean expression for S and C_o is given As

$$S = A \oplus B \oplus C_i$$

$$= \overline{A}\overline{B}\overline{C_i} + \overline{A}B\overline{C_i} + \overline{A}\overline{B}C_i + ABC_i$$

$C_o = AB + BC_i + AC_i$

The 1-bit full adder is the building block of all the modules. The output of full adder is sum and carry output. The binary adder is made up from standard AND and OR gates and allow us to add together single bit binary numbers A and B produce the outputs the sum of the addition and a carry called the carry out (c_{out} bit). Figure 2 shows the full adder gate level circuit.The third level of CBCD architecture consists of a RCA. This RCA is used to correct the sum output by adding correction factor to sum output of first level RCA. This RCA doesn't need any carry input, so the first block of this RCA is a half adder. The half adder can add two numbers and produce sum and carry output.
The carry out circuit consists of two AND gates and one OR gate. The inputs to carry AND gates are sum output of first level adder. The outputs of AND gates are acting as input to OR gate. Also the carry output of first level adder is applied as input to OR gate.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

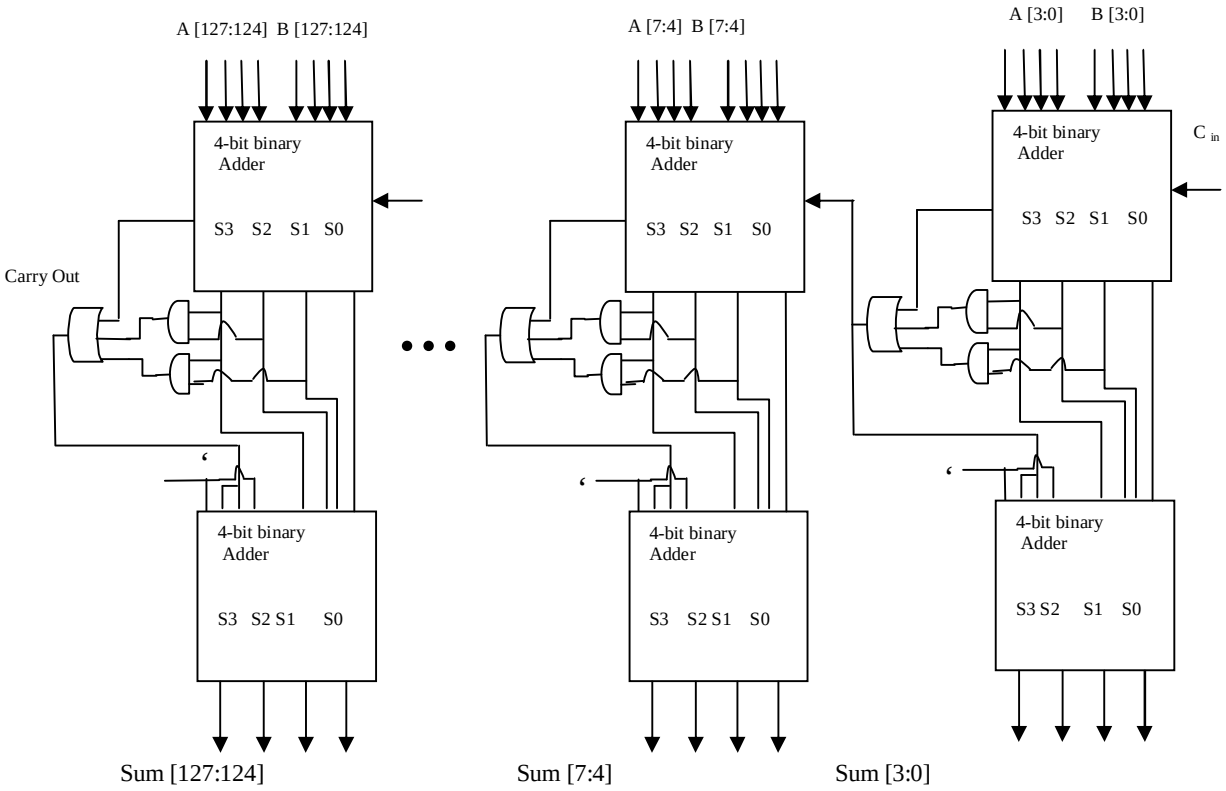

Fig. 3 Conventional BCD adders

The output of carry out circuit is used carry input for next BCD digit addition. Also the carry out circuit output used to generate 1st and 2nd bit of correction factor. The main drawback of in this architecture is very slow because of carry propagation which affecting the speed of the adder.

IV.BCD ADDER USING CLA ARCHITECTURE

In order to speed up the BCD adder operation, a new architecture is proposed in this paper. This architecture consists of CLA, analyzer unit to produce digit generate and propagate signals to form carry network. The carry look ahead adder determines whether that bit pair will generate a carry or propagate a carry. The carry network is a parallel prefix network [7], [8]. The carry valve for each digit is computed inside the carry network by using below equation. The network may be parallel prefix or two levels carry look ahead logic and is used for correction.

$$\text{Output carry} = DG + DP.\text{INPUT CARRY} \quad \ldots\ldots\ldots\ldots (1)$$

Figure4 shows the complete BCD adder including 4-bit adders for correction. Correction is done by adding 0 or 6 to the binary sum coming from the first level adder. The correction step must be fulfilling two requirements.

First the carry, coming from the previous digit should increase the binary sum of the digit by 1. Second the output carry of the related digit should determine whether the binary sum will be corrected by adding 6 or not. The correction is required by adding 6 when there is a decimal carry out which is coming from the carry network. The combination of Carry look ahead adder is cascading up to 128 bit. When we give 4-bit inputs to the circuit it produces DP, DG, sum which are used as inputs for the carry network.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

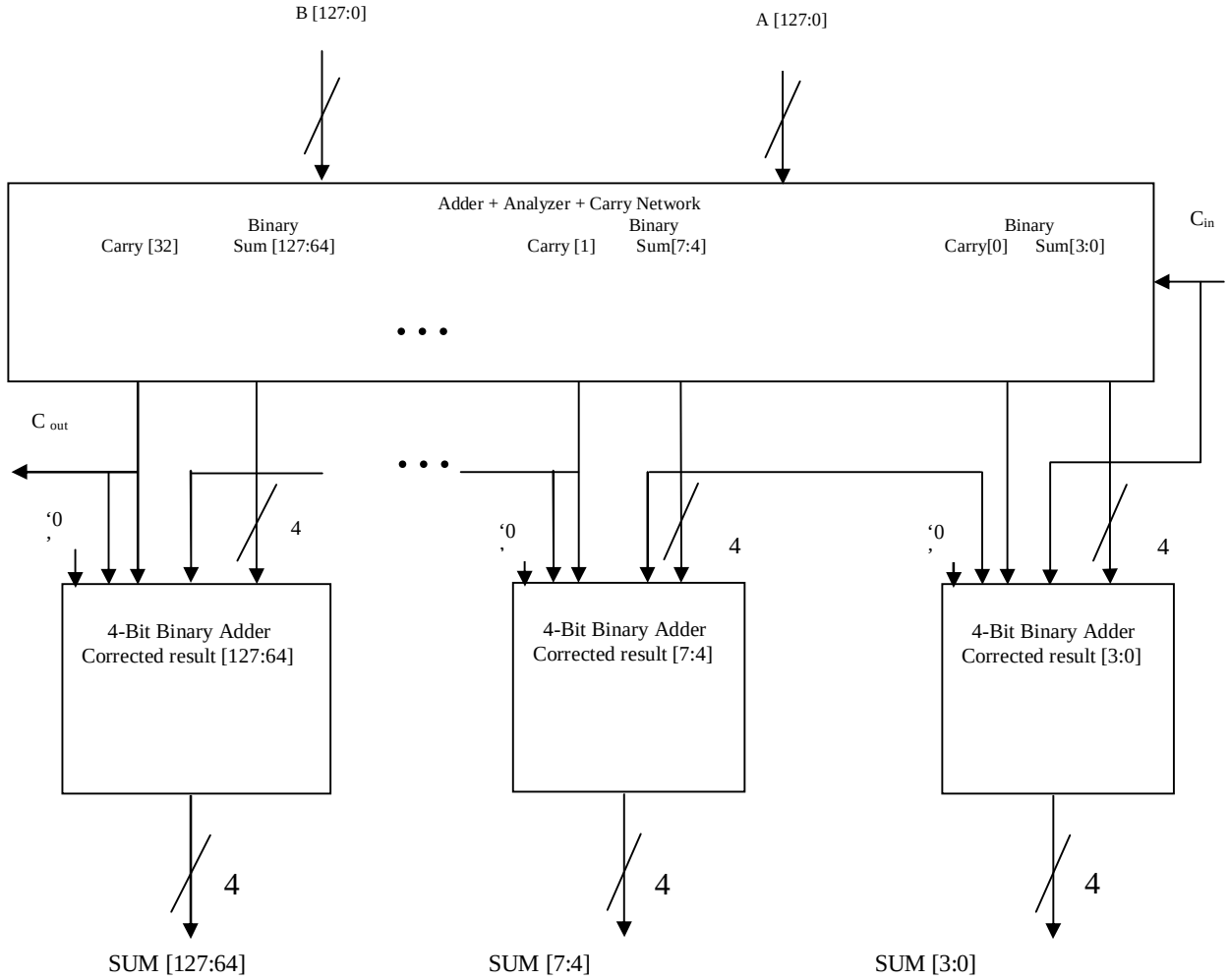

Fig. 4 128-bit BCD using CLA

When correction is required the carry input from the carry network and the input values given to the adder and the default value 0 is given as inputs to the first stage of the BCD adder using CLA and if carry produces then the carry from binary adder again acts as an input to the next stage along with usual inputs.

V. RESULT ANALYSIS

Fig.1 represents the algorithm for BCD addition, which is the important operation and performed by the processor. Fig.2 represents Full Adder which is the important building block of all the modules. Fig.3 represents conventional BCD Adder by which we are forming a carry network. And Fig 4. Represents the complete BCD adder.These two BCD adder architectures i.e., conventional BCD adder and BCD adder using CLA are designed for 8,16,32,64 and 128 bit input sizes using VHDL [9]. Functional simulations are carried out by using Xilinx ISIM simulator. The adder designs are synthesized using a Xilinx 14.2[10]. It is a synthesis tool from Xilinx. It provides area report and delay report of the design in terms of number of LUTs, slices occupied and in Nano seconds respectively.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

Table 1 Comparison of two BCD adders in terms of delay (ns)

Input size	Adder Type	Delay(ns)
8 bit	Conventional BCD adder	20.160
	BCD adder using CLA	18.294
16 bit	Conventional BCD adder	30.370
	BCD adder using CLA	29.462
32 bit	Conventional BCD adder	55.742
	BCD adder using CLA	50.446
64 bit	Conventional BCD adder	106.478
	BCD adder using CLA	93.910
128 bit	Conventional BCD adder	207.950
	BCD adder using CLA	172.236

From table 1, it is clear that the BCD adder using CLA is faster when compared with the CBCD Adder. As shown in table 1, BCD adder using CLA for 8, 16, 32, 64 and 128 bit has 1.866 ns,0.908 ns,5.296 ns, 12.568 ns and 35.714 ns delay less respectively when compared with CBCD Adder for respective bit size. So it can be said that the designed BCD adder using CLA is faster.

VI.CONCLUSION

In this paper, BCD adder using CLA is proposed. The BCD adder using CLA is designed for 8, 16, 32, 64 and 128 bit using VHDL and functionally simulated using Isim simulator. The designed adder is synthesized using Xilinx 14.2 ISE EDA tool. In order to compare the performance of designed adder with the existing adder, the existing adder i.e., CBCD adder also designed for the same bit sizes using VHDL, functionally simulated and synthesized using same tool mentioned above. Then the designed adders are compared with each other in terms of delay (ns). As shown in table 7.1, BCD adder using CLA for all bit sizes has minimum delay compared to conventional BCD adder. The BCD adder using CLA has a delay of 18.294 ns, 29.462 ns, 50.446 ns, 93.910 ns and 172.236 ns for 8, 16, 32, 64 and 128-bit size respectively where as CBCD has a delay of 20.160 ns, 30.370 ns, 55.742 ns, 106.478 ns and 207.950 ns for 8, 16, 32, 64 and 128-bit size respectively. So BCD adder using CLA for 8, 16, 32, 64 and 128 bit has 1.866 ns,0.908 ns,5.296 ns, 12.568 ns and 35.714 ns delay less respectively when compared with CBCD Adder for respective bit size. So it can be concluded that the designed adder i.e., BCD using CLA is having less delay when compared to CBCD adder. The BCD adder using CLA can be used for high speed applications.

VII.FUTURE SCOPE

In this paper a high speed BCD adder using CLA is designed. So this adder can be used in the design of high speed ALUs, computational units and processors. But there is a drawback with this adder. The area of adder is increasing with increase in the speed of the adder which in turn increases the power consumption of the adder. So, one can work to decrease the adder area without affecting the speed of the adder.

REFERENCES

[1] A. H. Burks, H. H. Goldstein, and J. von Neumann. Preliminary Discussion of theLogical Design of an Electronic Computing Instrument. Technical report, Institute forAdvanced Study, June 1946.
[2] M. M. Mano. Digital Design, pages 129–131. Prentice Hall, third edition, 2002.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 10, October 2016

[3] B. Shirazi, D. Y. Y. Young, and C. N. Zhang. RBCD: RedundantBinary Coded Decimal Adder. In IEE Proceedings, Part E, No. 2, volume 136, pages 156–160, March 1989.
[4] "Reduced delay BCD adder", Alp Arslan Bayrakc¸i and Ahmet Akkas¸ ComputerEngineering Department, Koc University, 2007 IEEE
[5] M. F. Cowlishaw. Decimal Floating-Point: Algorism for Computers. In Proceedings ofIEEE Symposium on Computer Arithmetic, pages 104–111, June 2003.
[6] M. F. Cowlishaw. Decimal Floating-Point: Algorism for Computers. In Proceedings ofIEEE Symposium on Computer Arithmetic, pages 104–111, June 2003.
[7] J. D. Thompson, N. Karra, and M. J. SchulteB. A 64-Bit Decimal Floating-PointAdder. In Proceedings of the IEEE Computer Society Annual Symposium on VLSI, pages297–298, February 2004.
[8] P. M. Kogge and H. S. Stone. A Parallel Algorithm for The Efficient Solution of aGeneral Class of Recurrence Equations. IEEE Trans. on Computers, C-22(8), Aug. 1973.
[9] VHDL primer by J. Bhaskar
[10] Xilinx 14.2 user manual
[11] Digital design Principles & practices by John F. Wakerly
[12] I. S. Hwang. High Speed Binary and Decimal Arithmetic Unit. United States Patent, (4,866,656), September 1989.