International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11

http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

FPGA Logic Circuit Implementation and Synthesis with VHDL Programming: A Learning Approach

L. A. Ajao[1], M. A. Adegboye[2], J. Agajo[3], A. O. Ajao[4], A. A. Yunus[5]
[1,3,5]Department of Computer Engineering, Federal University of Technology, Minna, Nigeria
[2]Department of Computer Engineering, Federal University Oye-Ekiti, Nigeria
[4]Department of Computer Engineering, Federal Polytechnic, Offa, Nigeria

ajao.wale@futminna.edu.ng, mutiu.adegboye@fuoye.edu.ng,
james.agajo@futminna.edu.ng, ajaodoc@yahoo.com,abey4luv@gmail.com

Received: 2017-10-27 **Accepted: 2017-11-05** **Published online: 2017-11-10**

Abstract

There are significant numbers of relevant research works available that concerns VHDL programming and Field Programmable Gate Array (FPGA) based hardware design, simulation and implementation. This is because; both the FPGA design and VHDL programming realization are quiet new and useful to accomplish various tasks in the field of research for digital system designed and development of miniaturized embedded system. It is found difficult to understand and put into practice by the learners in the tertiary institution, which still requires rapid training and development. In this paper, we design and demonstrate an FPGAlogic circuit using 4-bit BCD adders and parallel 4-bit comparator with a stepwise development of the vital soft logic design flow, simulation and timing analysis. It also presents an educational concept designed for complementing courses offered like FPGA prototype and ASIC design. This perception of FPGA-based design flow will facilitate learner's understanding, skills and it will provide detailed insights into various aspects of microelectronics, digital logic systems design and VHDL programming.

Keywords: Behavioral synthesis, BCD Adder, Four-bit comparator, FPGA prototype, Hardware design, Microelectronic, VHDL programming.

1. Introduction

Worldwide, it has been noticed that students tend to be less interested in the areas of embedded systems development, using either the custom IC, microcontroller technology or FPGA based system due to the poor technical and practical experience they are able to gather over the years of studies [1]. More so, the limited experts in the fields of embedded systems have also contributed immensely to the inadequacy, and availability of resources such as hardware and software development tools which has accounted for the loss of interest.

In spite of these difficulties, today's students can confidently step into the area of FPGA circuit development and VHDL programming by undertaking certain simple in-house designs, simulation and implementation with frequent practices using related software like Xilinx, Alteral and others [2]. FPGA strategy becomes one of the most successful technologies for developing embedded systems which require a real time operation, less power consumption and are faster in operations [3]. The FPGA is two dimensional arrays of logic blocks and flip-flops with an electrically programmable interconnection between logic blocks as shown

in figures 1 and 2 respectively. The interconnections of logic blocks consist of electrically programmable switches which differentiate Custom ICs from an FPGA type. The Custom IC is programmed by means of integrated circuit fabrication technology to form metal oxide semiconductor interconnections between logic blocks. In an FPGA, logic blocks are implemented using multiple level of low fan-in gates, which offer a more compact design in contrast to an implementation with two-level AND-OR logic [4]. The FPGA provides its user a way to configure large amount of logic in a single IC called logic block. The logic block of an FPGA can be configured in such a way that it can provide functionality as simple as that of the transistor or as complex as that of a microprocessor, which can be used to implement different combinations of combinational and sequential logic functions [5].

It is pragmatic that Binary Coded Decimal (BCD) digit adder is the essential component of the additionalaccurate decimal for the computer arithmetic operational system. Where adders are configured to perform high speed arithmetic operations and are very important components in digital systems as logic circuits [6]. Apart from other fundamental operations like addition, subtraction, multiplication and division, this can be handled by adder without modification of its module. It'salso capable of performing addition/subtraction of signed magnitudes by converting them into 1's complement or 2's complement.

In some instance, the computational decimal based prediction software is inefficient to give a fixed performance required by the applications with an extensive range of decimal calculation. Since the output figures are necessary to be converted from the binary system to decimal form for certain usages like commercial, financial, industrial or economical purposes that required a large figure of input/output conversion [7]. For that reason, the BCD systems enhanced the conversion of binary-decimal form through encoding each decimal digit separately as a structure of 4-bits binary system. In most digital computers and other kinds of microprocessors employed adders to play important role in the arithmetic logic unit and in other parts of the central processing unit, where addresses, index table, and several kinds of operations are calculated.

Furthermore, a number of ideas have been put forward for the purpose of computational in minimizing time during the course of literature reviewing. In the classical algorithms, it is confirmed that the completion time of any program or logic circuit is dependent on the number of digits/bits available in the operands. Therefore, the convenient execution of BCD adders do not only save the coding interfaces, but also saves the time consumption through different approaches engaged to design high speed decimal adders [8]. The structural design and pictorial illustration of the logic block of an FPGA device is clearly shown in the figure 1 and figure 2 respectively.

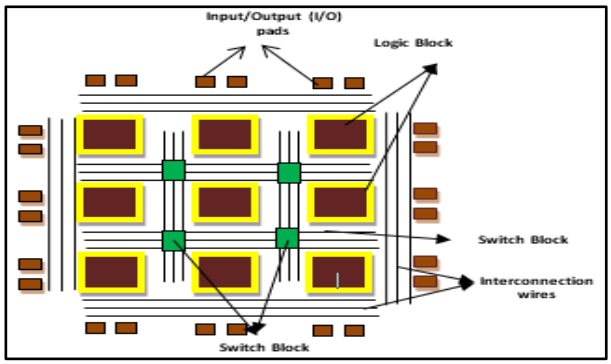


Figure 1: Architecture of an FPGA Device

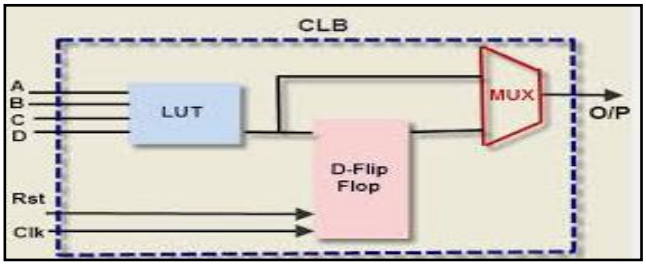


Figure 2: FPGA logic block diagram [14]

II. Related Works

The technicality in FPGA-based logic design flow, behavioral synthesis, simulation and implementation has been put forward and detailed by several authors. In [9], "FPGA Implementation of CRC with Error Correction" discussed on the CRC decoder using VHDL for the implementation which gives advantages of correcting more than one bit error at a goal. [10], the major concern of this author is about "FPGA-based New Hybrid Adder Design with the Optimal Bit-Width Configuration". Their work is centered towards the design of a specific technology and some implementation challenges, providing high level automated methodology for designing hybrid adders with high performance and neglecting the low level circuit issues.

"An FPGA-based Implementation of Digital Logic Design using Altera DE2 Board" was demonstrated by [11]. This work presents an experimental implementation of digital logic design on the Altera DE2 board which is an accessible educational and development board. It helps to check the flexibility implementation with FPGA and to get the better and safe ways to use these specifications during any design implementations. Also, [12] worked on "Design of Optimized Reversible BCD Adder/Subtractor". They proposed a method to realize a Reversible Binary Coded Decimal (BCD) adder/subtractor circuit, simply to optimize the design of nines compliment gate (NCG) and BSCL gates.

Conclusively, this work employs Xilinx ISE design 14.1 for exhibit simple methods of learning the difficulties in FPGA circuit design techniques, digital logic, both simple and complex programmable logic devices with configurations, as well as simulation and implementation with VHDL using 4-bits BCD Adders and parallel comparator. With this illustration of FPGA circuit design and synthesis, hope that students, young researchers, and novices will be assisted to have total understanding over the complexities in the design flow of FPGA-based logic circuit, simulation and hardware descriptive language (HDL).

III. The Design Flow of an FPGA-based Methodology

The FPGA contain thousands of the basic logic gates with advance sequential logic functions all together in a single package. Since the internal digital logic has not been configured to perform any function, the only way to make this device useful is to configure it by using the FPGA computer software like Xilinx ISE design suit 14.1 to outline the schematic logic and finally achieved hardware implementation and programming on the development board.

There are two method that can be adopts for the FPGA design flow which is through a process called *schematic capture*, this reads the graphic drawing of the logic and converts (compile) it to a binary file that accurately describes the logic to be implemented and generate VHDL code. This binary file is used as an input in a programming process that electronically alters the internal part of the PLD connection (synthesizes) to make it function specifically as required [13].

Another approached to define the logic for programmed into the PLD is to adopt a programming method using high level language called *VHDL* means Very High Speed Integrated Circuit Hardware Description Language (HDL). In this case, the inputs, outputs and logic processes are defined using a programming language, this method is somewhat require programming knowledge for the hardware logic design executions as described as follows.

The logic processes are defined using a programming language which comprises of (Library body, Entity and Architecture).

i. "The LIBRARY declaration", This refer to the files source that used for translating and resolving the language within the body of the programs e.g.

LIBRARY ieee;
USE ieee.std_logic_1164.ALL

International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11
http://ijcsc.ielas.org/index.php/ijcsc ISSN: 2545-4153

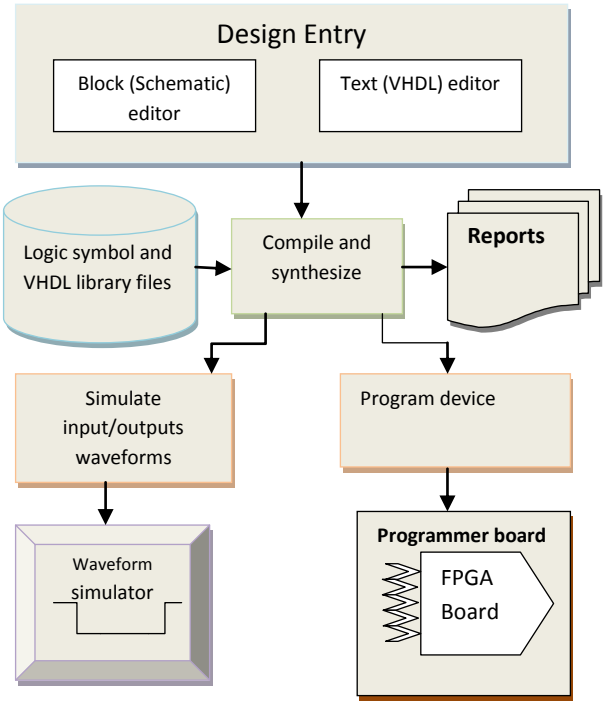
ii. "ENTITY declaration", used to defines the inputs and outputs ports of the CPLD. e.g

```
ENTITY cpe516_1 IS
    PORT (
        a, b: IN std_logic;
        x:    OUT std_logic  );
END cpe516_1;
```

iii. "Architecture Declaration", used to defines the internal logic operations that will be performed on the CPLD ports. e.g

```
ARCHITECTURE arc OF cpe516_1 IS
    BEGIN
        x<=a AND b;
END arc;
```

The figure 3 depicts block diagram of an FPGA design and implementation, while figure 4 gives details of the design flow of an FPGA circuit simulation, behavioral synthesis, and its implementation.



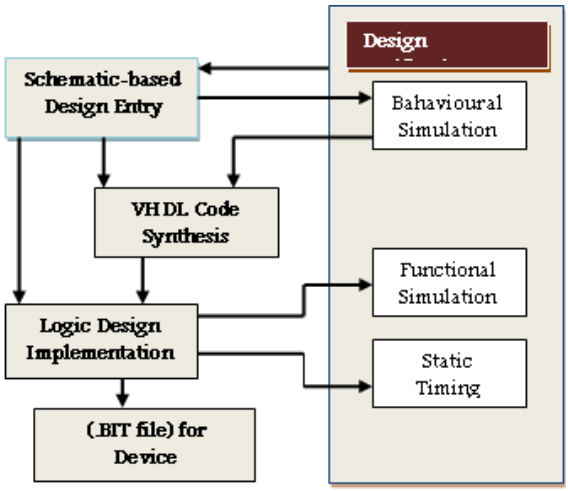


Figure 3: FPGA Design Flow

Figure 4: FPGA design flow chart, behavioral synthesis, simulation and implementation

A. Procedure for the Behavioural Synthesis of 4-bits BCD Adder using Schematic

The method adopted in this paper for the FPGA-based 4-bits BCD Adder synthesis, is a schematic capture called graphical. This method is chosen as a learning approach because of ease and its flexibility for digital logic design, FPGA circuit synthesis with or little knowledge of Verilog/VHDL programming.

The following are the steps taken during the graphical design entry, synthesis, simulation and implementation of the 4-bit BCD adders system.

Stage One:

Description of internal architecture of 4-bit BCD adders

i. First, launch the Xilinx ISE project wizard, click on file to create a ***new project,*** type the project name for the new project, and then click on next button then the ***Finish*** button.
ii. On the project, click on ***new source*** select the ***Schematic*** and type the file name, then click on the next button and click on the ***Finish*** button.

International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11
http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

iii. In the schematic design wizard, click on add symbol to select a logic symbol in order to begin the design. Two XOR-2 inputs, two AND-2 inputs and one OR-2 input were selected to create a single bit full adder, while all the components placed are wired together as shown in the figure 5.That is, 3 inputs(A_0, B_0 and C_{in}), and 2 outputs of a sum and a carry out(S_0 and C_0).

iv. Four of the single bit full adders were created by repeating step 3 above, and the schematic is saved as shown in the figure 5.

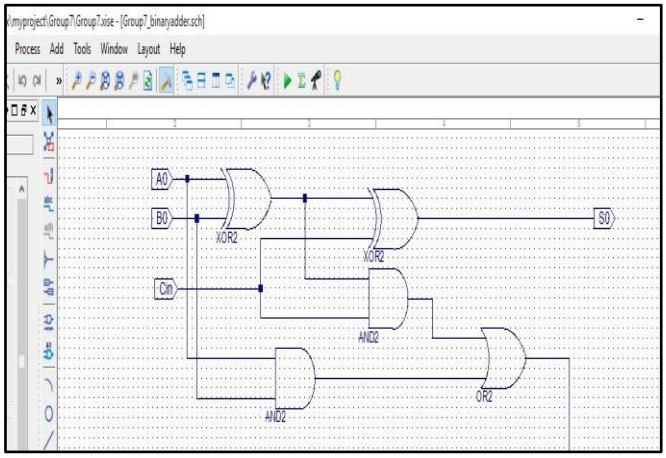


Figure 5: Schematic design of a single bit full adder

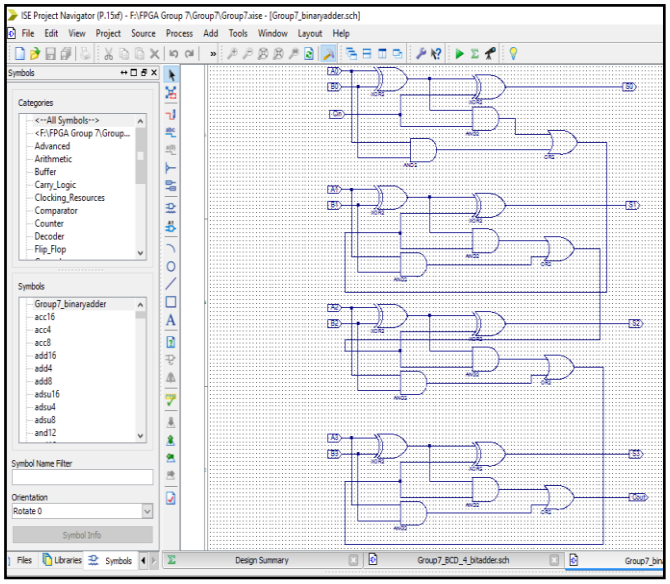


Figure 6: Schematic design of a 4-bit BCD adder

Stage Two:

Test bench of components,design and configuration of 4-bit BCD Adder

i. In the design pane, click on the project, point to a new source, then select schematic and fill the file name, click on the Finish button. New schematic design to simplify the circuit in figure 6 is created in the test bench and saved. This is just for the ease of configuring the system.
ii. In the new schematic feature, click on Tools, point to the symbol wizard, checked the box (using schematics), click next, click on Add pin tab, Click on Next and Finish.

Two of the 4-bit binary adders were used for a circuit design symbol (component), all components are wired together as shown in the figure 6, the internal register transfer logic (RTL) is in figure 7 and the work is saved.

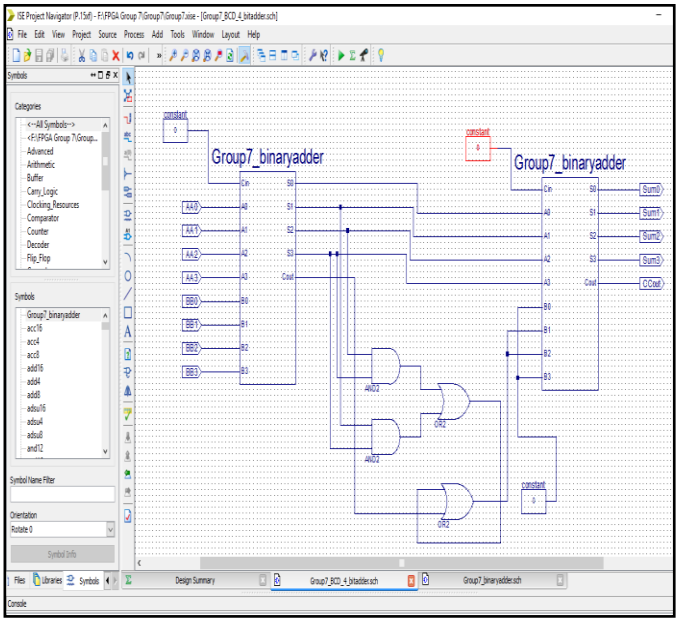


Figure 6: implementation of 4-bit BCD Adder in the test bench environment

International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11
http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

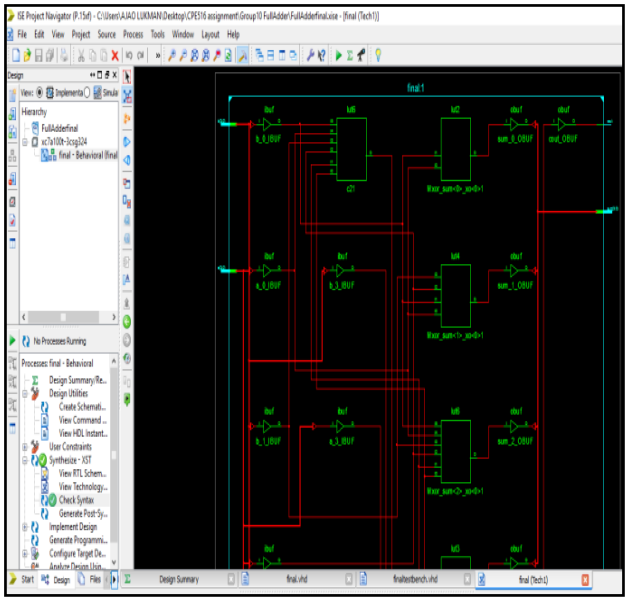


Figure 7: Internal architecture of a simulated 4-bit BCD adder circuit

Stage Three:

VHDL code synthesis and simulation

To generate the VHDL code for the FPGA circuit synthesis (4-bit BCD Adder) in the test bench are as follows:

i. In the design pane of the implementation window, click on the design utility, then point to and select "View HDL function" model. Then the VHDL code for the 4-bit BCD design will be generated as given in the table 1. The code will be studied and it requires little adjustment to be sure that it rightly represents the design architecture.

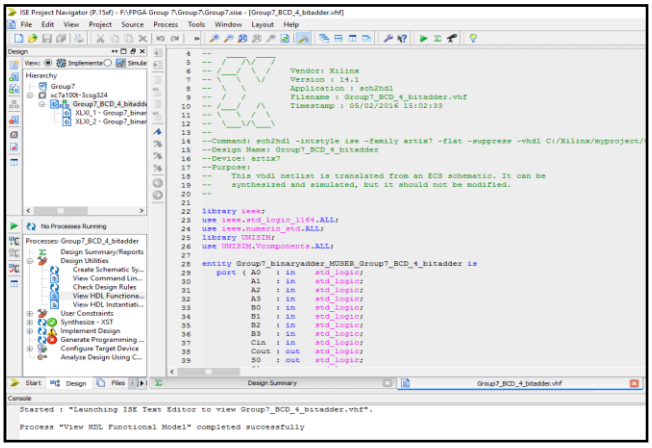


Figure 9: Snapshot of VHDL code generated in Xilinx ISE

ii. Under the design pane, the simulated button is checked. In the *ISim simulator*, click on the simulated behavioral model to generate the waveform for the 4-bit BCD adder designed.

A. Procedure for the Behavioral Synthesis of 4-bits Parallel Comparator using VHDL programming

In this scenario, the hardware descriptive approach using VHDL program was adopted to demonstrate the comparison for the parallel 4-bit data which are (0000-0001) and programmed clock to respond at every 5ns.The flowchart in thefigure 10 depicts an outline procedure for the FPGA development, synthesis and simulation for the parallel 4-bit comparator.

International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11
http://ijcsc.ielas.org/index.php/ijcsc ISSN: 2545-4153

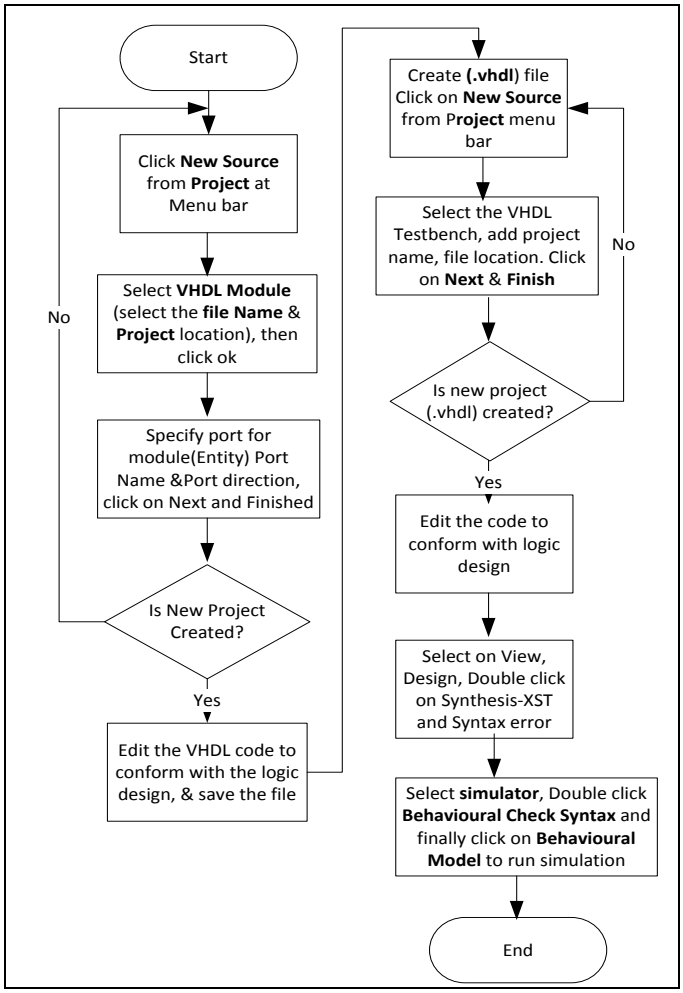


Figure 10: Flowchart procedures for synthesis parallel 4-bit comparator FPGA device

The register transfer logic (RTL) and the internal logic synthesis of the 4-bit comparator are depicted in the figure 11 and 12. The VHDL testbench programme is presented as follows.

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity Group1_Parallel_4Bit_Comparator is

Port (A: in STD_LOGIC_VECTOR (3 downto 0);

B: in STD_LOGIC_VECTOR (3 downto 0);

W: out STD_LOGIC);

end Group1_Parallel_4Bit_Comparator;

architecture group1_arc of Group1_Parallel_4Bit_Comparator is

begin

w<=(A(0) XNOR B(0)) AND(A(1) XNOR B(1)) AND

(A(2) XNOR B(2)) AND (A(3) XNOR B(3));

end group1_arc;

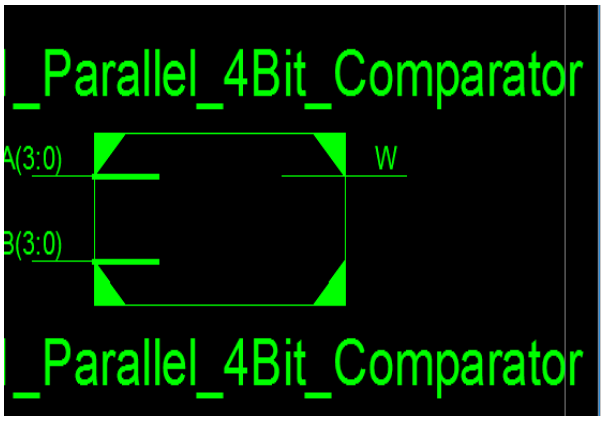


Figure 11: RTL schematic for the parallel-4-bit comparator

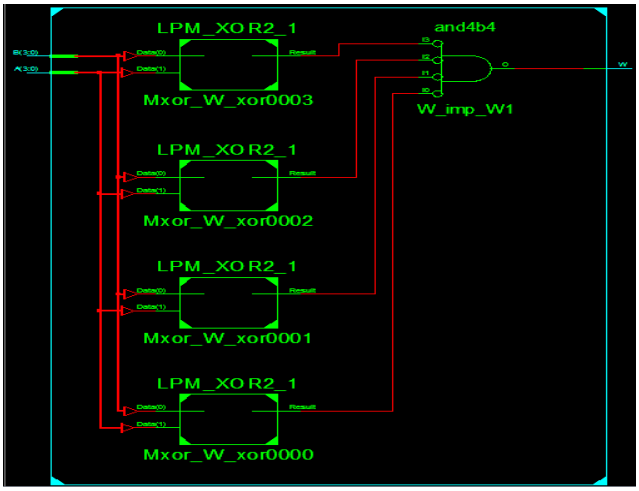


Figure 12: The logic design of the parallel-4-bit comparator

International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11
http://ijcsc.ielas.org/index.php/ijcsc
ISSN: 2545-4153

IV. Discussion

The basic method for the logic design and behavioral synthesis of 4-bit BCD adder and parallel 4-bit comparator has been outlined and clearly demonstrated. The brief on the logic synthesis, simulation and implementation of 4-bit BCD adder on the schematic captured is discussed here. They carry out (C_0) of the first full adder is connected to the carry in (C_{in}) of the second adder, and the carry out (C_0) bit of the second is also connected to the carry in (C_{in}) of the third adder, and finally the carry out (C_0) of the third was connected to the carry in (C_{in}) of the fourth full adder as shown in figure 6.

Therefore, the output of the second OR-2 input pin was connected to the second and the third "B" input bits (i.e. B1, B2) of the second component. Finally, the first and the fourth "B" input bits (i.e. B0, B4) were connected to constant "0", and the two carry-in inputs were also connected to constant "0" respectively.

V. Simulation Waveform Result

The timing analysis for the FPGA circuit design for the 4-bits BCD adder has been demonstrated after functional simulation processes with VHDL code generated; the waveform signal to verify the behavior of the **4-bits BCD adders** at every 50ns has been generated as shown in the figure below. The simulation result gives the desired output for 4-bits BCD adders and 4-bit parallel comparator based on the pins configuration and parameter used to perform high speed on every arithmetic operation, and it's capable of performing addition/subtraction of signed magnitudes by converting them into 1's complement or 2's complement. The table 1 shows the details of VHDL programming for 4-bit BCD adder generated from an ECS Schematic, Synthesized and Simulated.

This is to demonstrate the simplicity in learning and understanding FPGA design, synthesize and programming with hardware descriptive language (HDL) with little or no knowledge about VHDL programming. Also, to encourage more learners, developers and researcher in the field of engineering and embedded system development. The result and details performances of gate delay at each level of the full-adder system are contains in the Table 2. Finally, the simulation result of parallel 4-bit comparator is depicted in the figure 14.

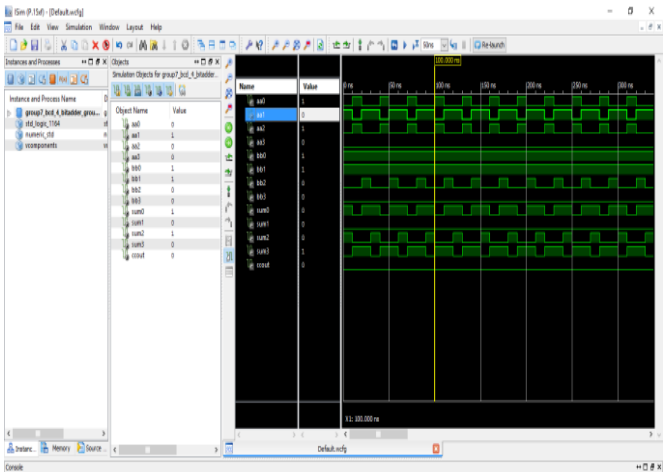


Figure 13a: 4-bit BCD adder waveform at 100ns

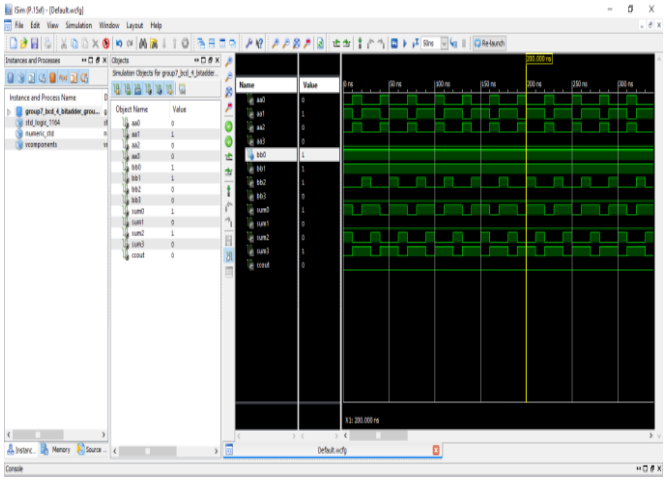


Figure 13b: 4-bit BCD adder waveform at 200ns

International Journal of Computer Science & Communications
Vol. 2, No. 1, 2017, pp. 1-11
http://ijcsc.ielas.org/index.php/ijcsc ISSN: 2545-4153

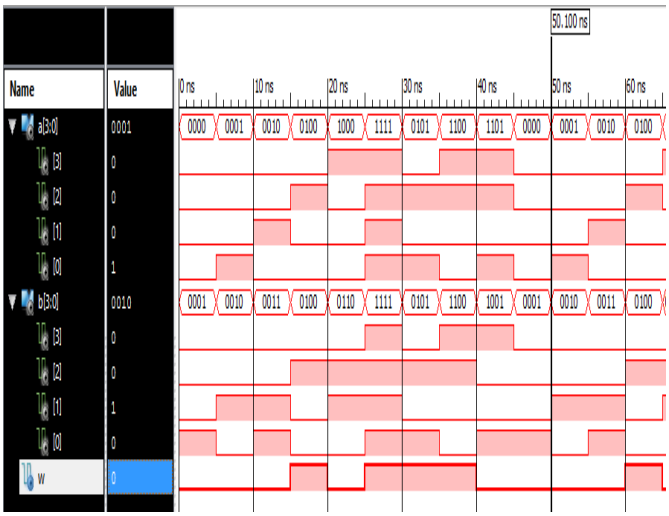


Figure 14: Parallel4-bit comparator waveform at 0 to 50ns

Table 1: VHDL code generated for the 4-bit BCD adderfrom an ECS Schematic, Synthesized and Simulated.

VHDL Netlist is translated from an ECS Schematic, Synthesized and Simulated.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
ENTITY
Group7_BCD_4_bitadder_Group7_BCD_4_bitadd
er_sch_tb IS
END
Group7_BCD_4_bitadder_Group7_BCD_4_bitadd
er_sch_tb;
ARCHITECTURE behavioral OF

Group7_BCD_4_bitadder_Group7_BCD_4_bitadd
er_sch_tb IS

  COMPONENT Group7_BCD_4_bitadder
  PORT( AA0  :       IN     STD_LOGIC;
        AA1  :       IN     STD_LOGIC;
        AA2  :       IN     STD_LOGIC;
        AA3  :       IN     STD_LOGIC;
        BB0  :       IN     STD_LOGIC;
        BB1  :       IN     STD_LOGIC;

        BB2  :       IN     STD_LOGIC;
        BB3  :       IN     STD_LOGIC;
        Sum0  :      OUT    STD_LOGIC;
        Sum1  :      OUT    STD_LOGIC;
        Sum2  :      OUT    STD_LOGIC;
        Sum3  :      OUT    STD_LOGIC;
        CCout  :     OUT    STD_LOGIC);
  END COMPONENT;

  SIGNAL AA0        :       STD_LOGIC;
  SIGNAL AA1        :       STD_LOGIC;
  SIGNAL AA2        :       STD_LOGIC;
  SIGNAL AA3        :       STD_LOGIC;
  SIGNAL BB0        :       STD_LOGIC;
  SIGNAL BB1        :       STD_LOGIC;
  SIGNAL BB2        :       STD_LOGIC;
  SIGNAL BB3        :       STD_LOGIC;
  SIGNAL Sum0       :       STD_LOGIC;
  SIGNAL Sum1       :       STD_LOGIC;
  SIGNAL Sum2       :       STD_LOGIC;
  SIGNAL Sum3       :       STD_LOGIC;
  SIGNAL CCout      :       STD_LOGIC;

BEGIN

  UUT: Group7_BCD_4_bitadder PORT MAP(
            AA0 => AA0,
            AA1 => AA1,
            AA2 => AA2,
            AA3 => AA3,
            BB0 => BB0,
            BB1 => BB1,
            BB2 => BB2,
            BB3 => BB3,
            Sum0 => Sum0,
            Sum1 => Sum1,
            Sum2 => Sum2,
            Sum3 => Sum3,
            CCout => CCout
  );
-- *** Test Bench - User Defined Section ***
  tb: PROCESS
  BEGIN
       AA0 <= '0';
       AA1 <= '1';
       AA2 <= '0';
       AA3 <= '0';
       BB0 <= '1';
       BB1 <= '1';
       BB2 <= '0';
       BB3 <= '0';    WAIT for 10ns;
       AA0 <= '1';
       AA1 <= '0';
       AA2 <= '1';
```

```
        AA3 <= '0';
        BB0 <= '1';
        BB1 <= '1';
        BB2 <= '0';
        BB3 <= '0';    WAIT for 10ns;
        AA0 <= '0';
        AA1 <= '1';
        AA2 <= '0';
        AA3 <= '0';
        BB0 <= '1';
        BB1 <= '1';
        BB2 <= '1';
        BB3 <= '0';    WAIT for 10ns;
    END PROCESS;
-- *** End Test Bench - User Defined Section ***
END;
```

Table 2: Result for the 4-bit BCD adder synthesis and configuration

Data Sheet report:

```
----------------
All values displayed in nanoseconds (ns)
--------------+--------------+---------+
Source Pad    |Destination Pad|  Delay  |
--------------+--------------+---------+
AA0           |CCout         |   9.525|
AA0           |Sum0          |   6.037|
AA0           |Sum1          |   8.474|
AA0           |Sum2          |   9.513|
AA0           |Sum3          |   9.845|
AA1           |CCout         |   9.918|
AA1           |Sum1          |   8.867|
AA1           |Sum2          |   9.906|
AA1           |Sum3          |  10.238|
AA2           |CCout         |   8.984|
AA2           |Sum1          |   7.933|
AA2           |Sum2          |   8.972|
AA2           |Sum3          |   9.304|

AA3           |CCout         |   8.694|
AA3           |Sum1          |   7.643|
AA3           |Sum2          |   8.682|
AA3           |Sum3          |   9.014|
BB0           |CCout         |   9.778|
BB0           |Sum0          |   6.179|
BB0           |Sum1          |   8.727|
BB0           |Sum2          |   9.766|
BB0           |Sum3          |  10.098|
BB1           |CCout         |   9.738|
BB1           |Sum1          |   8.687|
BB1           |Sum2          |   9.726|
BB1           |Sum3          |  10.058|
BB2           |CCout         |   9.115|
BB2           |Sum1          |   8.064|
BB2           |Sum2          |   9.103|
BB2           |Sum3          |   9.435|
BB3           |CCout         |   8.881|
BB3           |Sum1          |   7.830|
BB3           |Sum2          |   8.869|
BB3           |Sum3          |   9.201|
--------------+--------------+---------+
Peak Memory Usage: 616 MB
```

VI. Conclusion

This paper presents an approach of an FPGA logic circuit simulation with VHDL programming using 4-bits BCD Adders and parallel 4-bit comparator as an experimental project by using Xilinx ISE 14.1. It also demonstrates the simplicity and a stepwise idea to involve in study and undertaken an in-house project of digital logic system, PLDs configuration and FPGAs circuit design with VHDL programming for educational and research purposes.
Also, it offers students and professionals a wide range of knowledge, and expert in the field of embedded system design, ASIC programming and implementation. The Xilinx software offers a rich set of features that make it suitable for researchers, developers and learners to use in the laboratory environment or workshop for their practices. FPGA based model has a much simpler designing cycle because of the electronic design automated (EDA) software that provides much of routing, placement and timing. Future task may involve in: design of the single-on chip (SoC) embedded system and BCD adder with higher numbers of bit like16-bit, 32-bits or 64-bits adder as a reference adder.

References

[1] L. A. Ajao, J. Agajo, J. G. Kolo, M. A. Adegboye, and Y. Yusuf, "Learning of Embedded System Design, Simulation and Implementation: A Technical Approach", American Journal of Embedded Systems and Applications, Vol. 3, No. 3, May, 2016, pp. 35-42.

[2] L. A. Ajao, O. M. Olaniyi, J. G. Kolo, A. O. Ajao, "Project-Based Microcontroller System Laboratory Using Bk300 Development Board With PIC16F887 Chip", International Journal of Embedded Systems and Applications, Vol.5, No.3, September 2015

[3] J. P. Deschamps, G. J. Bioul, and G. D. Sutter, "Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems", 1st ed. Wiley Interscience, 2006.

[4] FPGA tutorial "Over view on FPGA", www.Tutorialreports.com, 2008. http://www.tutorialreports.com/computer-science/fpga/overview.php

[5] M. Y. Hassan and W. F. Sharif, "Design of FPGA based PID-like Fuzzy Controller for Industrial Applications", IAENG International Journal of Computer Science, 34:2, IJCS_34_2_05, November, 2007.

[6] M. I. Ibrahimy, MD. R. Ahsan and I. B Soeroso, "Hardware Modeling of Binary Coded Decimal Adder in FPGA", WSEAS Transactions on Computers, Issue 10, Volume 11, October 2012.

[7] S. Mamataj, B. Das, A. Rahaman, "A More Effective Realization Of BCD Adder By Using A New Reversible Logic BBCDC", International Journal of Computational Engineering Research, Vol, 04 Issue, 2, February, 2014.

[8] M. F. Cowlishaw, "Decimal Floating-Point: Algorism for Computers". Proceedings of the 16th IEEE Symposium on Computer Arithmetic 2003, pp: 104-111.

[9] W. M. El-Medany, "FPGA Implementation of CRC with Error Correction", The Eighth International Conference on Wireless and Mobile Communications, ICWMC, 2012.

[10] M. M. Alshewimy and A. Sertbas, "FPGA-based New Hybrid Adder Design with the OptimalBit-Width Configuration" International Journal of Computer Applications (0975 – 8887), Volume 65 – No.11, March, 2013.

[11] Z. A. Obaid, A. Sulaiman and M. N. Hamidon, "FPGA-based Implementation of Digital Logic Design using Altera DE2 Board" IJCSNS International Journal of Computer Science and Network Security, Vol.9 No.8, July, 2009

[12] S. B. Rashmi, B. Praveen, and B. G Tilak, "Design of Optimized Reversible BCD Adder/Subtractor", IACSIT International Journal of Engineering and Technology, Vol.3, No.3, June 2011.

[13] W. Kleitz, "Digital Electronics with VHDL" Quartus II version, Published by Pearson Prentice Hall, Upper Saddle River, New Jersey Columbus, Ohio, 2006.

[14] RF Wireless World 2012, RF & wireless Vendors and Resources "www.rfwireless-world.com/Articles/FPGA-architecture.html