*International Journal of Environment* 4(**1**): 18-24 (2014)

ORIGINAL ARTICLE

# Proficient FPGA Execution of Secured and Apparent Electronic Voting Machine Using Verilog HDL

*Tabia Hossain, Syed Syed Shihab Uddin, Iqbalur Rahman Rokon, K.M.A Salam, M. Abdul Awal*

ISSN: 2186- 6872 (print)
ISSN: 2186-0009 (online)
http://www.BENJapan.org/IJE

ARTICLE HISTORY

Received: 10 November 2013
Revised: 3 May 2014
Accepted: 5 June 2014

Published online: 14 June 2014

***Abstract***

Electronic Voting Machine is an electronic voting device used for conducting the parliamentary elections electronically. It consists of two units that can be inter-linked; a ballot unit which a voter uses to exercise his vote and a control unit which used by the polling officials. As there is no available design of Electronic Voting Machine using Verilog FPGA, in this paper, we introduce an efficient, transparent and secured FPGA implementation of EVM using Verilog HDL. The design is coded in Verilog hardware description language at Register Transfer Level (RTL), simulated in ModelSim, synthesized in Quartus II and implemented in Cyclone II FPGA using the AlteraDE1 board.

**Key words:** *EVM, RTL, Verilog HDL, FPGA, VLSI, FSM*



## I. INTRODUCTION

Bangladesh has always been able to keep pace with the technology advancements in all sectors worldwide. Compared to the other developed, even to some of the developing countries, Bangladesh lags far behind in the system of national parliamentary voting in technological perspective. For a democratic country, public opinion is the most important to establish a government. Voting is the process through which people display their opinion and help to setup a democratic government. So the voting system should be reliable, accurate and it must be transparent. The voting system of Bangladesh is still paper ballot based, a very outdated process. The whole world is looking forward to E-voting rather than using paper ballots in voting. Today Electronic Voting Machines has become a major concern in electoral mechanisms in general elections. Considering these points, we made the design for an Electronic Voting Machine using Verilog FPGA, which absolutely is an original design of our own where with the advantages of VLSI design methodology and Verilog codes; we focused on implementing proficient, secured and apparent satisfactorily error free electronic voting device.

The rest of this paper is organized as follows. Section II present design specification. Section III shows design hierarchy and section IV shows top view of our design. Section V describes entire block design. Section VI describes methodology and hardware part and section VII describe the result of our design. Advantages and conclusions are given in Section VII and IX respectively.

AUTHORS INFO

*Tabia Hossain**
e-mail: tabi_friend@yahoo.com
Address: Department of Electrical Engineering and Computer Science, North South University Plot 15, Block B, Bashundhara Dhaka 1229, Bangladesh

*Syed Shihab Uddin*
e-mail: rishadetc@gmail.com
Address: North South University, Bangladesh

*Iqbalur Rahman Rokon*
e-mail: irahman@northsouth.edu
Address: North South University,Bangladesh

*K.M.A. Salam*
e-mail: asalam@northsouth.edu
Address: North South University, Bangladesh

*M.Abdul Awal*
e-mail: abdulawal@northsouth.edu
Address: North South University,Bangladesh

*Tabia Hossain
e-mail: tabi_friend@yahoo.com
Tel:+8801777682314

## II. DESIGN DESCRIPTION

This section should provide the reader with all the information necessary to repeat the work. Our voting machine works as same as electronic voting device. But there is a little bit change we made on our voting machine. Our voting machine does not totally depend on electronic system. We also include a punch system which performs quite similar operation like paper ballot system does. Voting contains

the instantiations of the other 5 modules, which are 5 separate source files of the 5 different operations. Inside Ballot module and Control module, there are another two modules separately instantiated.We use five different steps for our design. Firstly we have to prepare our design specification. From our design specification we write RTL Description. Then we convert our RTL description to Gate level design. From gate level design we go to physical layout of our design. Finally we implement our design. These five steps work individually but finally full work depends on each of the state. If we fail to fulfil one step, next step does not work. Though they are five different steps and work individually, they fully depend on each other. Without any of these we cannot complete our design as we want. Design flow diagram is given below:

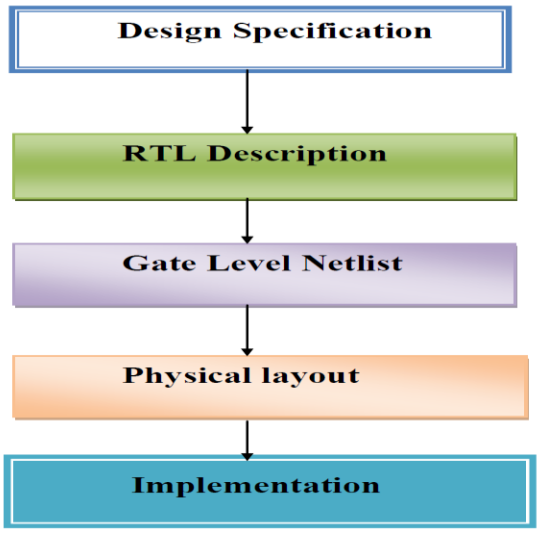


Fig.1. Design Flow Diagram

Our Electronic Voting Machine (EVM) works in two major units: *Control Unit* and *Ballot Unit.* The control unit counts the individual party results as well as the total vote counts. The ballot unit makes a beep sound and gives a green signal when a cast vote is accepted. It will also give out a punched paper for the appropriate party. If any error occurs, the ballot unit will give a red signal indication an error and hence, it will display an error message. Validity of a voter for a certain vote centre will be checked at fist. Only a valid voter with a valid NID can access the machine to cast a vote. Error can occur in three ways: firstly wrong ID encounter; secondly repetition of the same voter and thirdly, if more than one party is pressed simultaneously. When a vote is successfully accepted, along with the sound and the green signal, a punched paper will come out putting a seal on the appropriate party's logo. Individual party counts and the total counts will be monitored and recorded successfully after each successive voting process is done. Our design of the EVM is a Finite State Machine (FSM) approach. A finite-state machine (FSM) or finite-state automaton is a mathematical model of computation used to sequential logic circuits and computer programs [9]. It is conceived as an abstract machine that can be in one of a finite number of states. The state diagram of our design is shown in Fig.2.

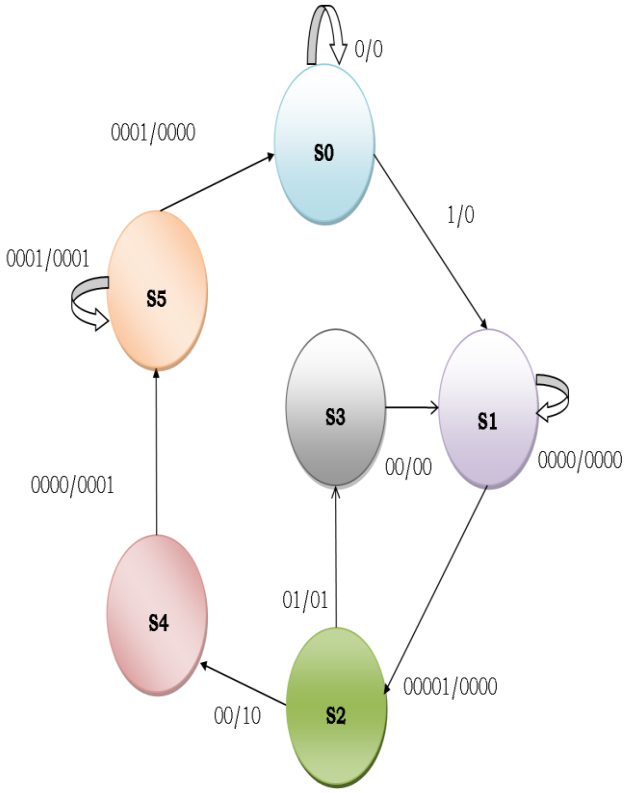


Fig.2. State Diagram of EVM

## III. Design Hierarchy

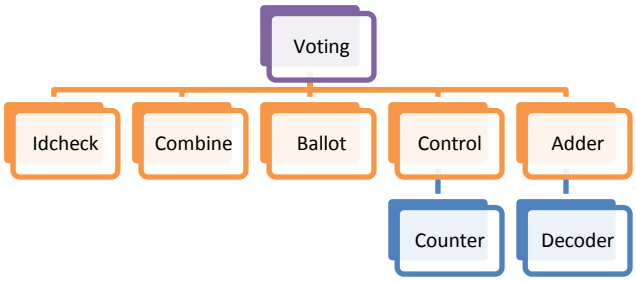


Fig.3. Design Hierarchy of EVM

## IV. Top View of EVM

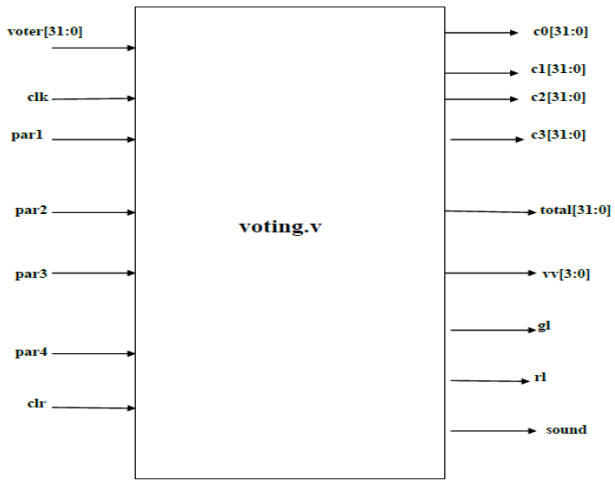


Fig.4. Top View of EVM

## V. Block Description

### A. Idcheck

It consists of clk, sz, voter and xx. This module checks whether the voter id valid or not. Whenever voter id matches with the stored data it makes xx high and allows the voter for the next state.

### B. Combine

It consists of con and party. Whenever id check unit makes xx high this unit allows voter to access his/her vote to any of the party (par1, par2, par3, par4) specified by the machine. Whenever con becomes binary 0001, vote is accepted for the party1, binary 0010 indicates vote for party2, binary 0100 for party3 and binary 1000 for party4.

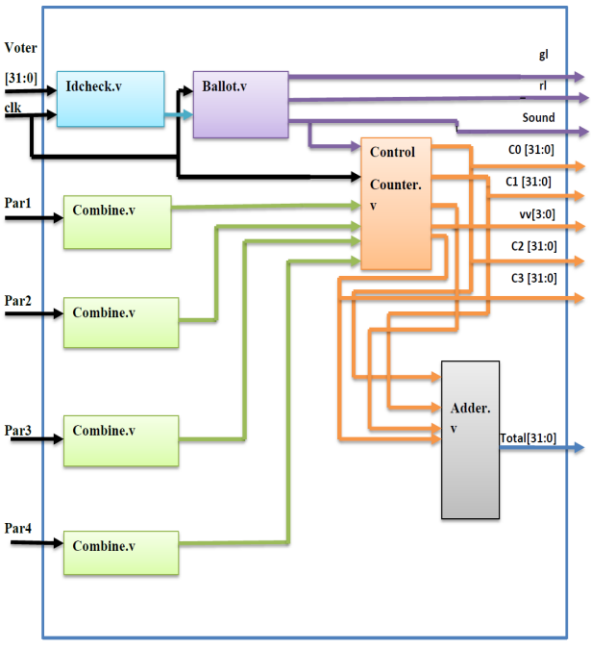


Fig.5.Block Diagram of EVM

### C. Ballot

It consists of clk, gl, rl, sound, and state. This module also consists of a decoder module which is instantiated inside the ballot module. Whenever a vote is properly accepted this unit gives beep sound and green signal which indicate the vote is accepted. If the vote was not accepted or any error occurs during the voting process, this module gives a red signal which indicates the process is wrong and turns the voter back to the second state.

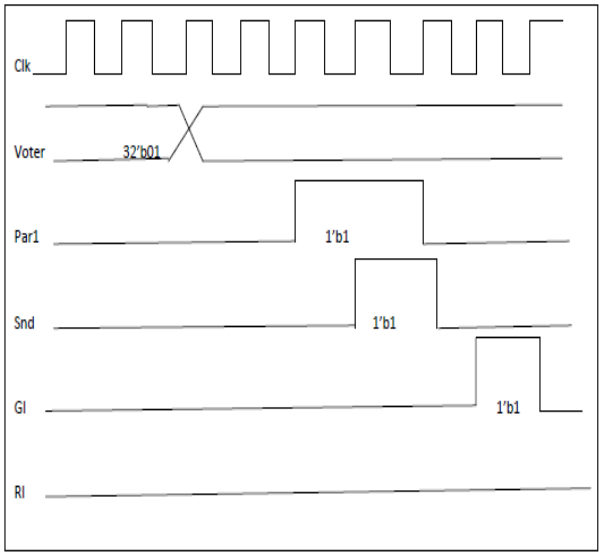


Fig.6. Timing Diagram of Ballot

### D. Control

It consists of the result of four individual party (c0, c1, c2, c3), punch (vv), clk, clr, con and sound. It also consists of a counter module which is instantiated inside it. In this unit machine gets the beep sound from the ballot unit and the counter starts to count the particular party's vote and punches the specific party's logo on a ballot paper. This module gives individual result of the individual party and punches out the sealed hard paper which can be stored outside the machine.

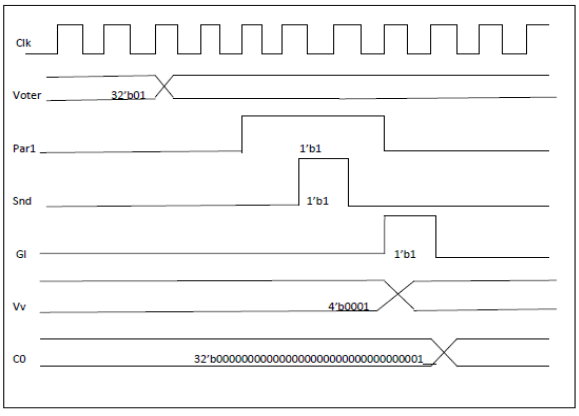


Fig.7. Timing Diagram of Control

*E. Adder*

It consists of individual results of the parties (co, c1, c2, c3) and total result. This unit adds individual results of the parties and gives the total number of vote accepted by the machine.

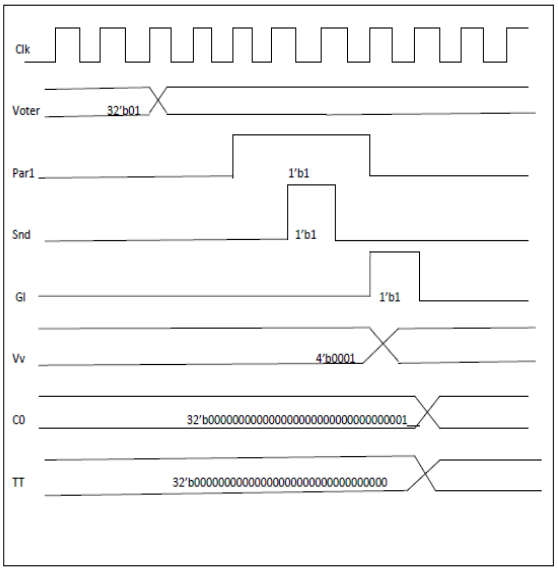


Fig.8. Timing Diagram of Adder

## VI. METHODOLOGY AND HARDWARE ARCHITECTURE

In this section, the design procedure and the architecture of EVM has been described. Fig.9. shows the different stages of the design. [2] The Verilog Codes were at first simulated with ModelSim and then synthesized with Xillinx and Quartus II Software targeted for Cyclone II (EP2C20F484C7) device [1]. FPGA technology is chosen because it provides some important advantages over general purpose processors and application specific integrated circuits (ASICs). To simulate the design, the design under test (DUT) and the stimulus provided by the test bench were required. The test bench was an HDL code that allowed providing a documented, repeatable set of stimuli that was portable across different simulators. The test bench used for this design was a more complicated file that included error checking, file input and output and conditional testing. After simulating the design, synthesis was performed. The Quartus II 11.1 Synthesis tool and the Xilinx Synthesis Technology (XST) of Xilinx ISE 9.2i tool software both synthesized the Verilog codes to create Xilinx / Quartus-specific net-list files.

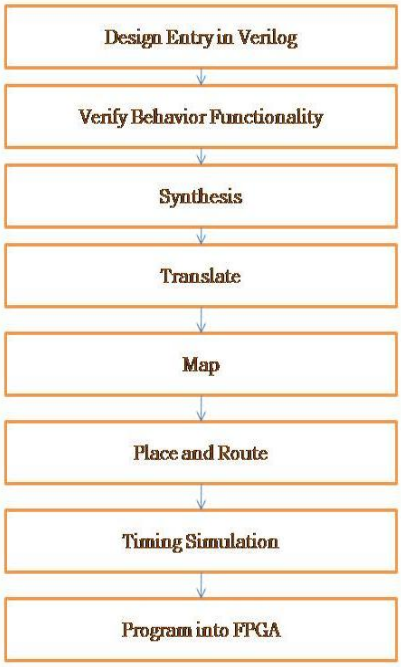


Fig.9. Implementation Flow Diagram

## VII. RESULT

We simulate our design by using Modelsim 6.5b software from which we get output web form of our design.

Fig.10. Shows the Simulation result of EVM:

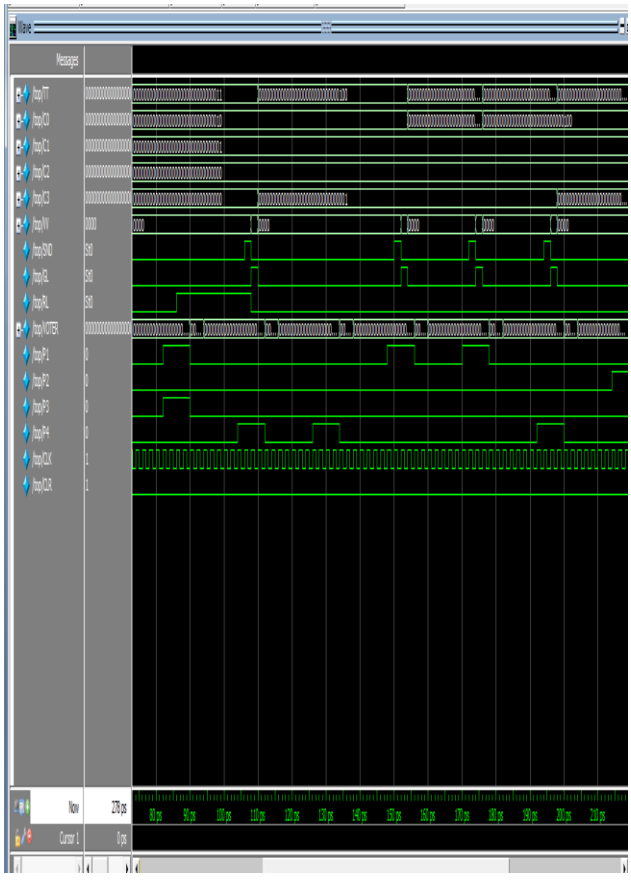

Fig.10. Simulation result of EVM

After simulating the design, synthesis was performed. The Quartus II 11.1 Synthesis tool and the Xilinx Synthesis Technology (XST) of Xilinx ISE 9.2i tool software both synthesized the Verilog codes to create Xilinx / Quartus-specific net-list files. Xilinx and Quartus II generated the following files as output.

RTL Schematic: This representation is in terms of generic symbols and was generated after the HDL synthesis phase of the synthesis process.

Fig.11. Shows the RTL Schematic diagram of EVM:

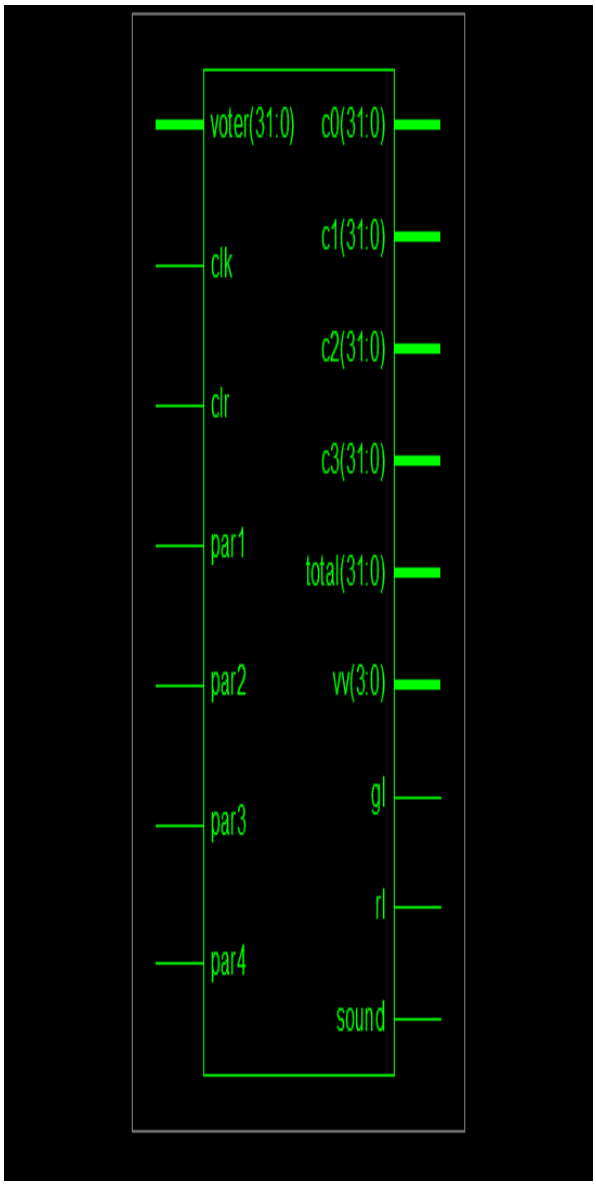


Fig.11. RTL Schematic Diagram of EVM

Technology schematic: This is a schematic representation shown in terms of logic elements which is optimized to the target architecture or "technology". It was generated after the optimization and technology targeting phase of the synthesis process.

Fig.12. Shows the internal block diagram of EVM:

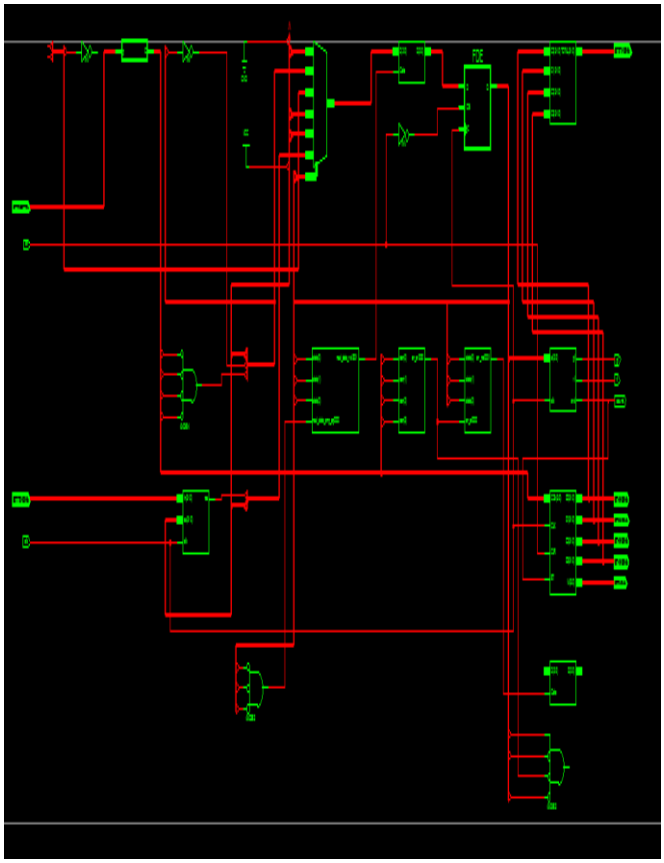

Fig.12. Internal RTL Schematic Diagram of EVM

Fig.13. Shows the Synthesis Summary of EVM

| Flow Summary | |
|---|---|
| Flow Status | Successful - Sat May 25 16:48:01 2013 |
| Quartus II Version | 11.0 Build 208 07/03/2011 SP 1 SJ Web Edition |
| Revision Name | voting |
| Top-level Entity Name | voting |
| Family | Cyclone II |
| Total logic elements | 250 / 14,448 ( 2 % ) |
| Total combinational functions | 250 / 14,448 ( 2 % ) |
| Dedicated logic registers | 137 / 14,448 ( < 1 % ) |
| Total registers | 137 |
| Total pins | 205 / 315 ( 65 % ) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 239,616 ( 0 % ) |
| Embedded Multiplier 9-bit elements | 0 / 52 ( 0 % ) |
| Total PLLs | 0 / 4 ( 0 % ) |
| Device | EP2C15AF484C7 |
| Timing Models | Final |

Fig.13. Device Summary from Quartus II Tool

Fig.14. Shows the data sheet report from Quartus II synthesis tool:

Setup Times

| | Data Port | Clock Port | Rise | Fall | Clock Edge | Clock Reference |
|---|---|---|---|---|---|---|
| 1 | dr_n | clk | 4.367 | 4.367 | Rise | clk |
| 2 | par1 | clk | 3.269 | 3.269 | Rise | clk |
| 3 | par2 | clk | 3.269 | 3.269 | Rise | clk |
| 4 | par3 | clk | 2.962 | 2.962 | Rise | clk |
| 5 | par4 | clk | 3.126 | 3.126 | Rise | clk |
| 6 | voter[*] | clk | 3.655 | 3.655 | Rise | clk |
| 7 | voter[0] | clk | 2.902 | 2.902 | Rise | clk |
| 8 | voter[1] | clk | 3.010 | 3.010 | Rise | clk |
| 9 | voter[2] | clk | 3.655 | 3.655 | Rise | clk |
| 10 | voter[3] | clk | 3.634 | 3.634 | Rise | clk |
| 11 | par1 | state[0] | 4.018 | 4.018 | Fall | state[0] |
| 12 | par2 | state[0] | 4.062 | 4.062 | Fall | state[0] |
| 13 | par3 | state[0] | 4.028 | 4.028 | Fall | state[0] |
| 14 | par4 | state[0] | 4.242 | 4.242 | Fall | state[0] |

Fig.14. Data Sheet Report from Quartus II Tool

Fig.15. shows the timing report from Quartus II synthesis tool:

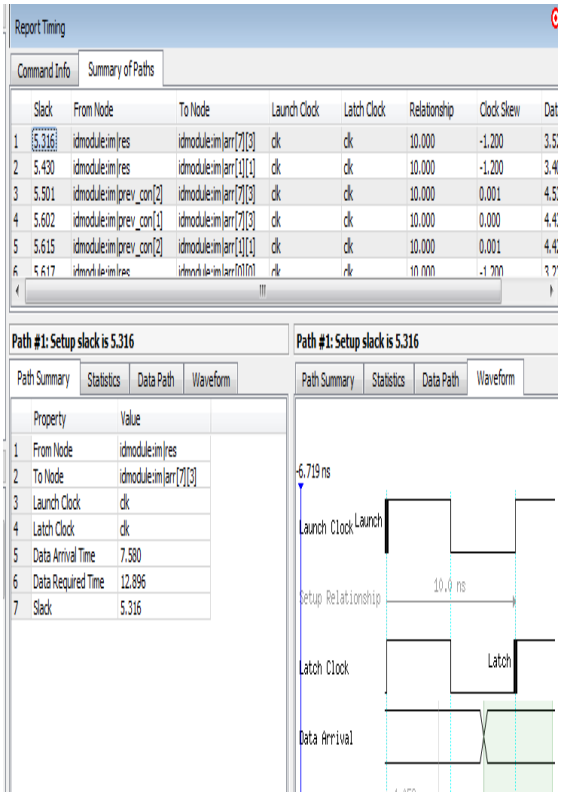


Fig.15. Timing Report from Quartus II Tool

For the implementation of the design, the Altera DE1 Development and Education Board is used. The purpose of the Altera DE1 Development and Education Board is to provide the ideal vehicle for advanced design prototyping in the multimedia, storage and networking. This board uses the state-of-the-art technology in both the hardware and software CAD tools to expose designers to a wide range of topics. The board offers a rich set of features that makes it suitable for use in a laboratory environment for the university and college courses, for a variety of design projects, as well as for the development of the sophisticated digital systems. Altera provides a suite of supporting materials for the DE1 board.

Fig.16. Shows the FPGA implementation on Altera DE1 board:

Fig.16. FPGA Implementation on Altera DE1 Board

## VIII. ADVANTAGES

By using simple Verilog simulations, we can easily hardware implement the design in our real life environment satisfying the issues of efficiency, cost and the transparency. So by using FPGA based EVM, we can save around 1100 corers of taka in the 5 years of electro cycle [4].

We also have FPGA based design advantages on our design.

We can also save huge amount of power by using this voting system as the battery is required only to activate the EVMs at the time of polling and counting. As soon as the polling is over, the battery can be switched off and this will be required to be switched on only at the time of counting.

For each national election alone it is estimated that about 10,000 tons of ballot paper (roughly 200,000 trees) would be saved by using Electronic Voting Machine. There is of course many more state and city/village level elections and the cost of printing those ballot papers would be also enormous [8].

The vote-counting is very fast and the result can be declared within 2 to 3 hours as compared to 30–40 hours, on an average, under the ballot-paper system.

## IX. Conclusion

Considering the fact of the uncompromising advancement of VLSI technology, we have successfully implemented an efficient Electronic Voting Machine on FPGA by satisfactorily meeting the related issues of security and transparency for an EVM. Our EVM deals with the sensitive cases in voting processes like restriction of an invalid voter, prohibition of same voter and simultaneous vote cast for more than one candidate. Along with that, if the punched paper results matches with the software results then we can say the machine result is also transparent. We can use this efficient, secured and transparent voting machine successfully for electoral voting process in Bangladesh and worldwide.

## Acknowledgement

At first the authors would like to thank Allah forgiving them the opportunity to do this work. Then they would want to mention their parents who supported them with mental and financial support.

## References

[1] Field Programmable Gate Array. Wikipedia, the free encyclopaedia. Retrieved on February16, 2012 from http://en.wikipedia.org/wiki/Fpga.

[2]Palnitkar,S.(2008).*VerilogHdL*(2nd).Pearson Education.

[3] *K.C.C.* Wai and S. J. Yang, "*Field Programmable Gate Array Implementation of Reed- Solomon Code*, *RS (255, 239*)", New York (2006).

[4] *fpga4fun.* (2010, September 07). Retrieved on September 07, 2010, from fpga4fun: http://www.fpga4fun.com/PCI1.html

[5]VerilogHardware Description Language Reference, Manual, Version 2.09, Los Gatos, CA, Open Verilog International, March 1993.

[6] *Altera DE1 board.* (n.d.). Retrieved December 10, 2010, from terasic: http://www.terasic.com.tw/cgi-bin/page/archive.p1?Language=English&No=83

[7] Barlow L.(2003, November), *An Introduction to Electronic Voting*, Retrieved from http://brahms.emu.edu.tr/rza/Electronic%20Voting.pdf

[8] Indian Voting Machine, Wikipedia the free encyclopedia. Retrieved from March 28, 2010, from http://en.wikipedia.org/wiki/Indianvoting machine.

[9] Finite State Machine, Wikipedia the free encyclopedia.https://en.wikipedia.org/wiki/Finite-state_machine.

[10] Verilog. Wikipedia, the free encyclopedia. Retrieved March 23, 2012 from http://en.wikipedia.org/wiki/Verilog_HDL

[11] Elections in Bangladesh 2006-2009: Transforming Failure into Success. Retrieved from http://sembec.org.bd/content/elections-bangladesh-2006-2009-transforming-failure-success

[12] Electronic Voting Machines. Retrieved from http://www.gktoday.in/quiz-597-current-general-knowledge