Department of Electronics & Communication Engineering

केन्द्रीय प्रौद्योगिकी संस्थान, कोकराझार
Central Institute of Technology, Kokrajhar
(A Centrally Funded Institute Under Ministry of HRD, Govt. of India)
Kokrajhar, BTAD, Assam, INDIA, PIN-783370

Implementation of Electronic Voting Machine with Multiple Preferences and Priority using VHDL

Under the supervision of

Mr. Kaushik Chandra Deva Sarma

(Asst.professor, Dept. of ECE)

A project word submitted by

Mr. Monoj Dutta	**Gau-c-10/235**
Mr. Manash K Hazarika	**Gau-c-10/L-317**
Mr. Anupam Rajbongshi	**Gau-c-10/260**
Mr. Suryakanta Ghose	**Gau-c-10/217**

Department of Electronics & Communication Engineering

केन्द्रीय प्रौद्योगिकी संस्थान, कोकराझार
Central Institute of Technology, Kokrajhar
(A Centrally Funded Institute Under Ministry of HRD, Govt. of India)
Kokrajhar, BTAD, Assam, INDIA, PIN-783370

Certificate of Approval

This is to certify that the work embodied in this project entitled **Implementation of Electronic Voting Machine with Multiple Preferences and Priority using VHDL programming** submitted by **Mr. Monoj Dutta**, **Mr. Manash Hazarika, Mr. Anupam Rajbongshi, Mr. Suryakanta Ghose** to the department of **Electronics and Communication Engineering** of Central Institute of Technology, Kokrajhar is carried out under our direct supervisions and guidance.

The project work hasbeen prepared as per the rules and regulation of Central Institute of Technology and I strongly recommend that this project work be accepted in partial fulfillment of the requirement for the degree of B.Tech.

Supervisor

21/11/13

(Mr. Kaushik Chandra Deva Sarma)

Assistant Professor,Dept. of ECE

Countersigned by

(Mr. Arindam Mukherjee)

HoD i/c ,Dept of ECE

Contents:

Abstract

Traditional paper based voting procedure was very long and time-consuming process and very much prone to errors. Polling by Electronic Voting Machine (EVM) is a simple, safe and secure method that takes minimum of time. Current Electronic Voting Machine (EVMs) used in LOK SABHA and ASSEMBLY elections accepts only one vote from each voter. But in elections such as GRAMA PANCHAYATH and COOPERATIVE SOCIETIES, where each voter casts their votes to more than one candidate, available voting machines will not work. The paper presents a PROGRAMMABLE ELECTRONIC VOTING MACHINE that accepts one or more votes depending on requirement. Mode control is included in EVM, through which it is possible to set the EVM to accept more than one vote from each voter depending on the type of elections. The main advantage of this type of EVM is to avoid the invalid votes especially in co-operative society elections where each voter has to cast vote for nine candidates.

ACKNOWLEDGEMENT

On the very outset of this report, we would like to extend our sincere and heartiest obligation towards all the personage who have helped us in this Endeavour. Without their active guidance, help, co-operation & encouragement, we would not have made headway in the project.

First and formost, we would like to express our sincere gratitude to our project guide, **Mr. Kaushik Chandra Deva Sarma (Asst. professor,Dept. of ECE,CIT,Kokrajhar)**. We are privileged to experience a sutained enthusiastic and involved interest from his side.He always fuelled our thoughts to think broad and out of box.

We would also like to thank **Mr. Haradhan Chel(Asst. professor,Dept. of ECE,CIT,Kokrajhar)** and **Mr. Arindam Mukherjee (HoD i/c,Dept. of ECE, CIT, Kokrajhar)**,for giving us the chance to do this project.

Date: 21 Nov, 2013

Monoj Dutta (Gau-c-10/235)

Manash K Hazarika (Gau-c-10/L-317)

Anupam Rajbongshi (Gau-c-10/260)

Suryakanta Ghose (Gau-c-10/217)

Introduction

Elections in India are conducted almost exclusively by using electronic voting machines developed over the past two decades by a pair of government-owned companies. These devices, known in India as EVMs, have been praised for their simple design, ease of use, and reliability. The Electronic Voting Machine (EVM) consists of two interconnected units, the Ballot Unit where the voter casts his vote by pressing a button alongside the name of the candidate and symbol of the party for whom the person chooses to vote for and the Control Unit by which the polling official enables the Ballot Unit for the voter to cast his vote and where all related data like number of votes polled for each candidate, total number of votes cast etc. resides. EVMs reduce the time in both casting a vote and declaring the results compared to the old ballot paper system. The control unit can store the result in its memory for more than 10 years. Invalid votes can be greatly reduced by use of EVMs. Blank votes can be counted. Despite many advantages of EVMs, there are certain issues in terms of software and hardware.

In the existing voting machines, number of candidates supported is limited to around 20 only, if number of candidates exceeds 20 we are supposed to use 2 voting machines. At present, EVMs are used only in LOK SABHA and ASSEMBLY elections (it accepts only 1 vote from each voter), but in elections such as GRAMA PANCHAYATH and COOPERATIVE SOCIETIES where each voter cast their votes to more than one candidate; available voting machines can not beused.

This paper presents a PROGRAMMABLE ELECTRONIC VOTING MACHINE that accepts one or more votes depending on our requirement. Here, we have included the mode control, through this we can set the EVM to accept more than one vote from each voter (it can accept 1 or 2 or 3 .... or n votes) depending on the type of elections [3].

. Our Contributions are mainly in designing priority and multiple weighted based voting, where a vote for multiple persons for a positions. Weights of preferences will be used to get final vote counting

The paper is organized as follows. First, working of present EVM is discussed, with its limitations in present Voting system Second, proposed work to overcome present EVM is discussed along with block diagram (modules) and working(in terms of state diagram.). Finally, simulation results, conclusion, and future work are presented.

Working of EVM

EVM consists of mainly two interconnected units, ballot (implemented with FPGA/CPLD) and control unit (implemented with FPGA), and display module (LEDs) as given in figure 1. In the ballot unit, voter casts his/her vote by pressing a button alongside the name of the candidate and symbol of the party for whom the person chooses to vote. Control unit is given to the polling official, who enables the ballot unit for the voter to cast his vote, and all related data, like number of votes polled for each candidate, total number of votes cast, etc., resides in it. The controls such as ballot, count, result, clear, total, Nvotes, clk, and cd are used to control operations of control unit. Display module is used to display alphanumeric characters. MSB 2 digits are used to display candidate serial number and rest 4 digits are used to display the total number of votes casted for corresponding candidate.

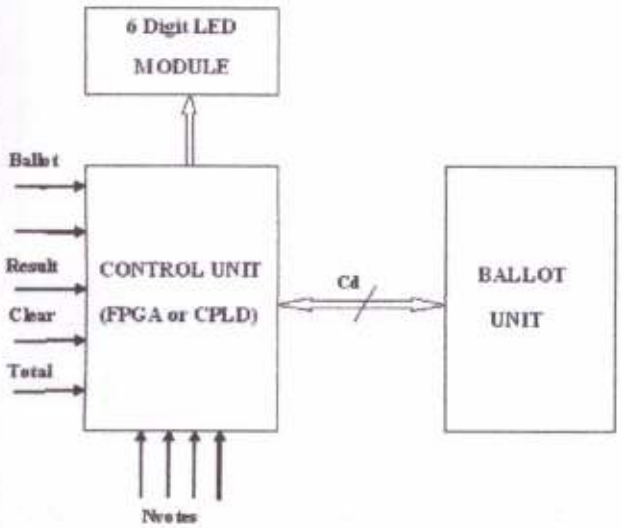


Figure 1. Block Diagram of Programmable EVM

The process of voting is as follows. Once the validity of the voter has been ensured, the polling official enables the ballot unit and the voter is asked to go to the area where the ballot unit is placed. Then, the voter scrutinizes the names of the candidates or parties displayed on the ballot unit and cast his vote by pressing the blue button beside the chosen candidate's name or symbol. The corresponding LED is lit and an audible beep is heard confirming the registration of vote in the system. This process is repeated for the next voter. On operation of the "Result" button of the Control Unit, the display indicates the results of the poll including total number of votes cast and the number of votes polled for each candidate.

(i) **Ballot unit**:

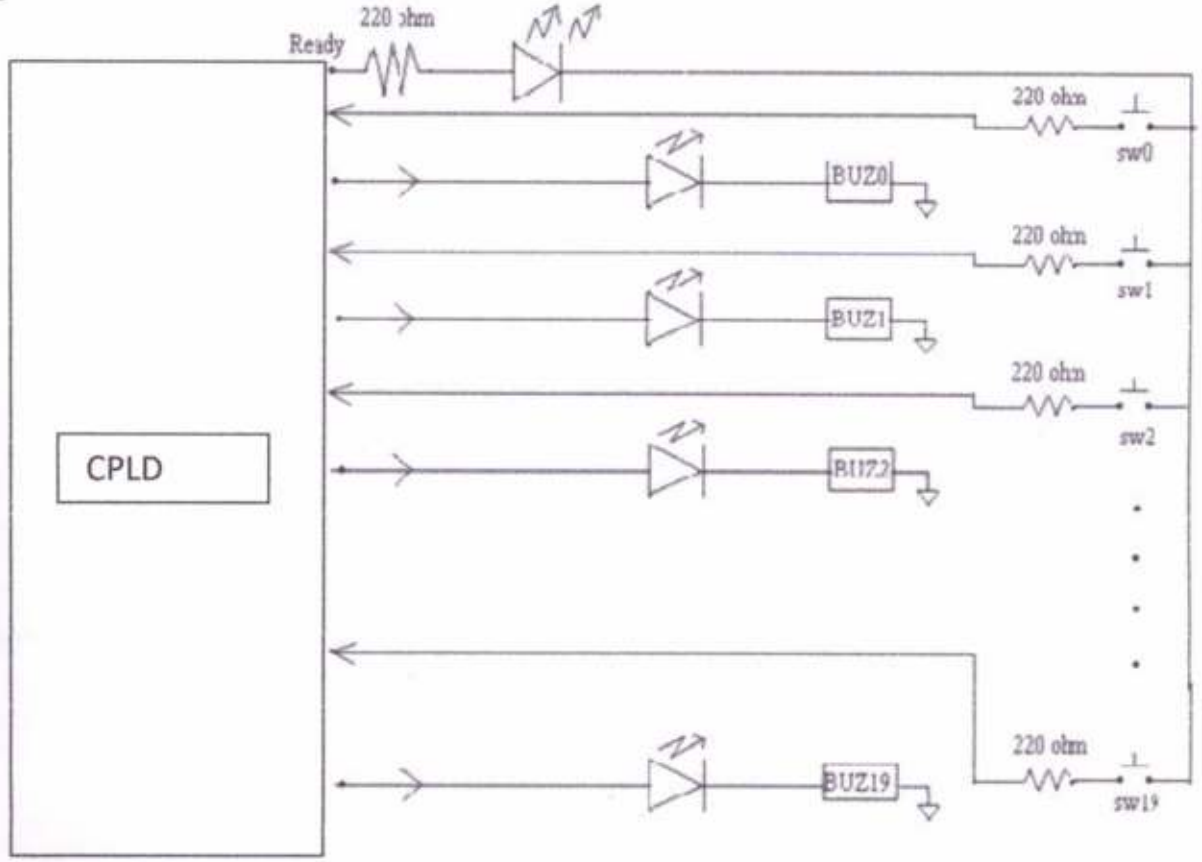


Figure 2. Circuit diagram of Ballot unit

The figure 2 shows the internal block diagram of the ballot unit of the programmable EVM. It consists of LEDs and push to on switches corresponding to the number of candidates on the ballot unit along with 220 ohms resistors in series used to drive the push to on switches, with the power supply of +5v .

The operation is as follows when the voter presses the blue button (one of sw0 to sw19) to cast the vote for his chosen candidate, the corresponding push to on switch gets closed, and a high signal is sent to the control unit, if this signal is enabled, then the counter gets incremented indicating the registration of the vote and the confirmation is done when the red LED glows (corresponding to pressed switch) and a beep sound is heard.

(ii) **Control Unit** :The 'Nvotes' is set by the presiding officer, which indicates the number of votes to be accepted from individual voters. 'Ballot' is also under the control of presiding officer, which is set to accept votes from voter. 'Clear' is set in order to store the votes registered for individual candidate in to FPGA ROM. When 'Result' is set high, number of votes accepted for each candidate, which is stored in ROM is displayed on multiplexed seven segment LED display.

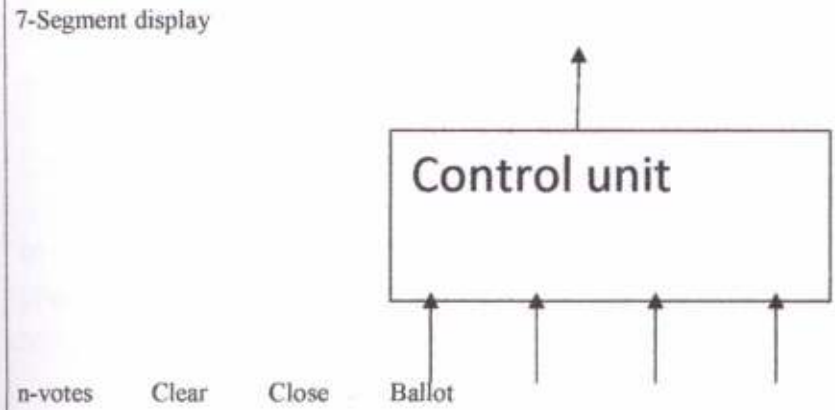


Figure3: Circuit diagram of control unit

Introduction to VHDL

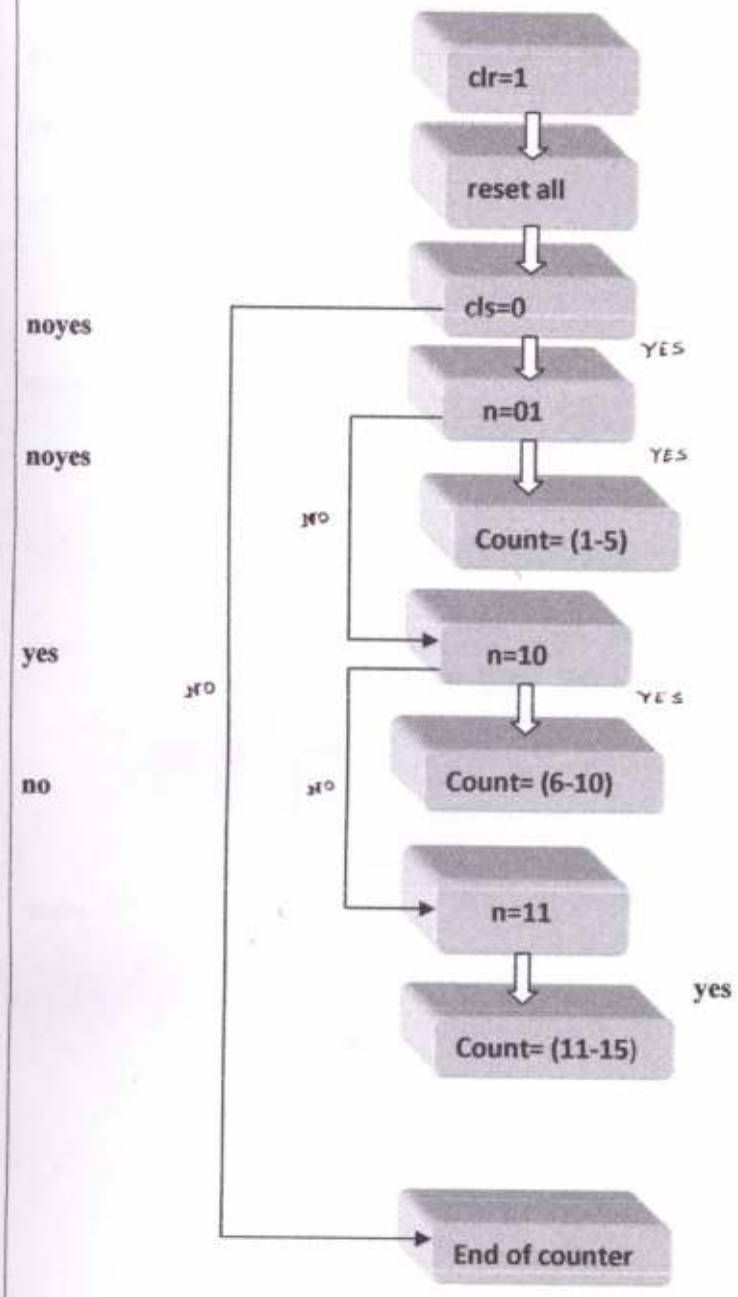
The VHSIC Hardware Description Language is an industry standard language used to describe hardware from the abstract to the concrete level. VHDL resulted from work done in the '70s and early '80s by the U.S. Department of Defence.VHDL usage has risen rapidly since its inception and is used by literally tens of thousands of engineers around the globe to create sophisticated electronic products. VHDL is a powerful language with numerous language constructs that are capable ofdescribing very complex behaviour.

Learning all the features of VHDL is not a simple task. Complex features will be introduced in a simple form and then more complex usage will be described. In 1986, VHDL was proposed as an IEEE standard. It went through anumber of revisions and changes until it was adopted as the IEEE 1076 standard in December 1987.

Entity: - All designs are expressed in terms of entities. An entity is the most basic building block in a design. The uppermost level of the design is the top-level entity. If the design is hierarchical, then the top-level description will have lower-level descriptions containedin it. These lower-level descriptions will be lower-level entities contained in the top-level entity description.

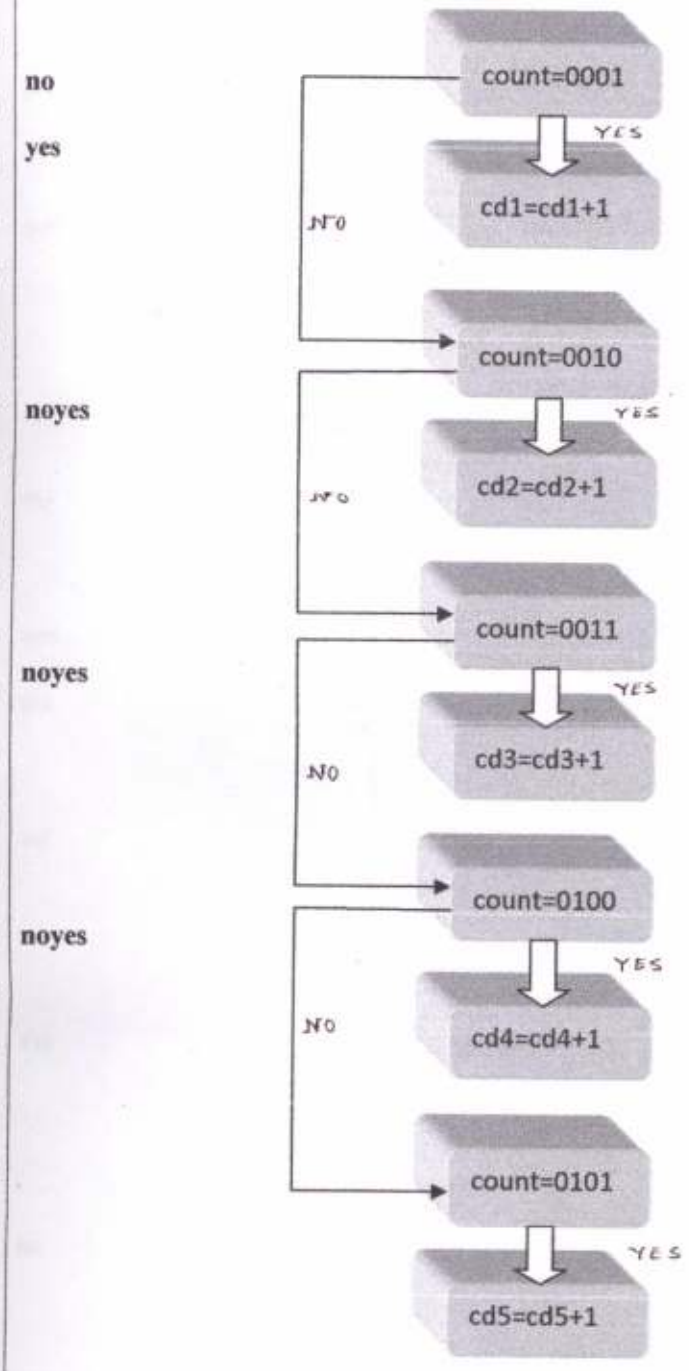
Architecture: - All entities that can be simulated have an architecture description. The architecture describes the behavior of the entity. A single entity can have multiple architectures. One architecture might be behavioural while another might be a structural description of the design.

Flow Diagram of EVM

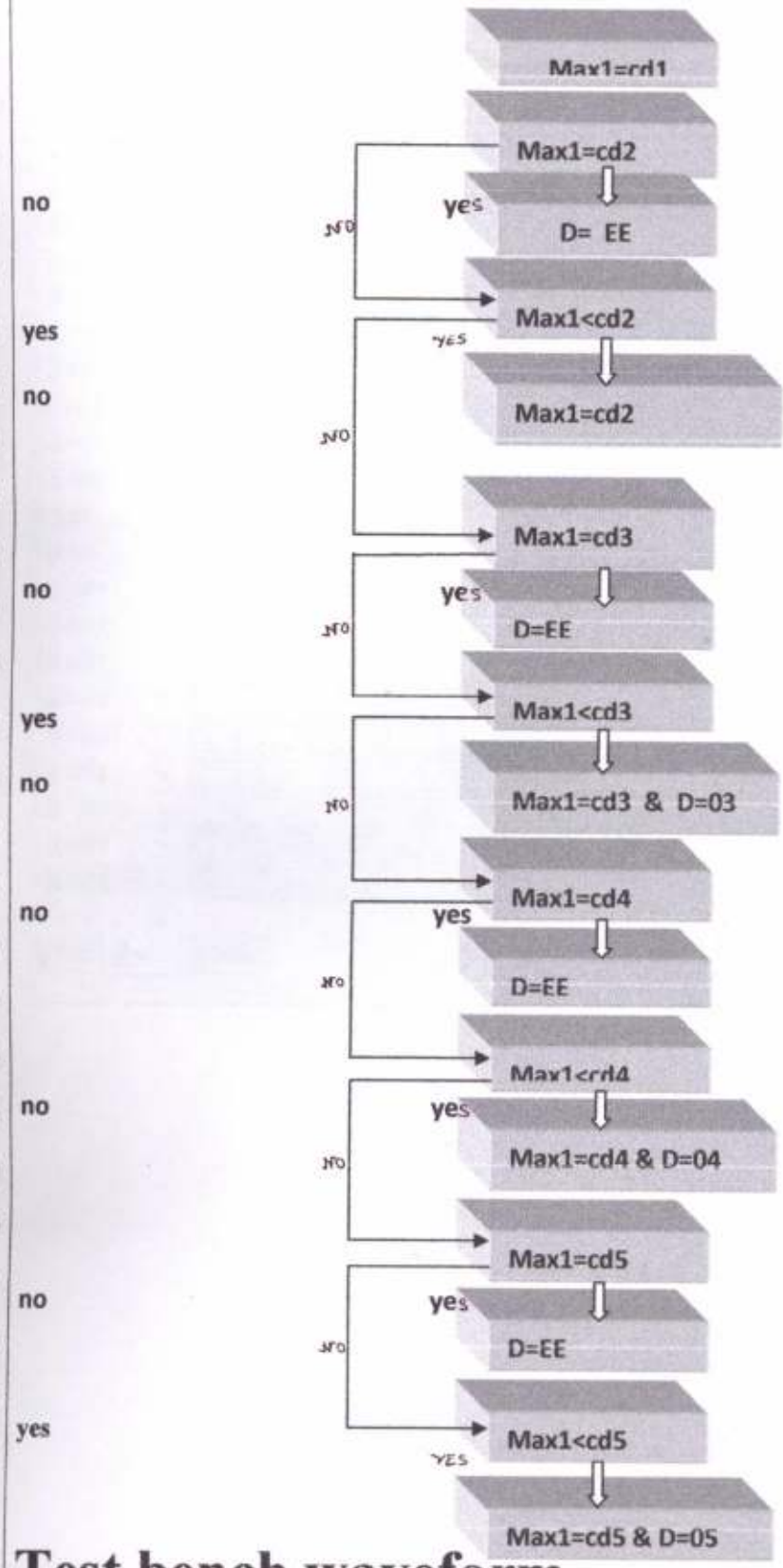


Design of counter

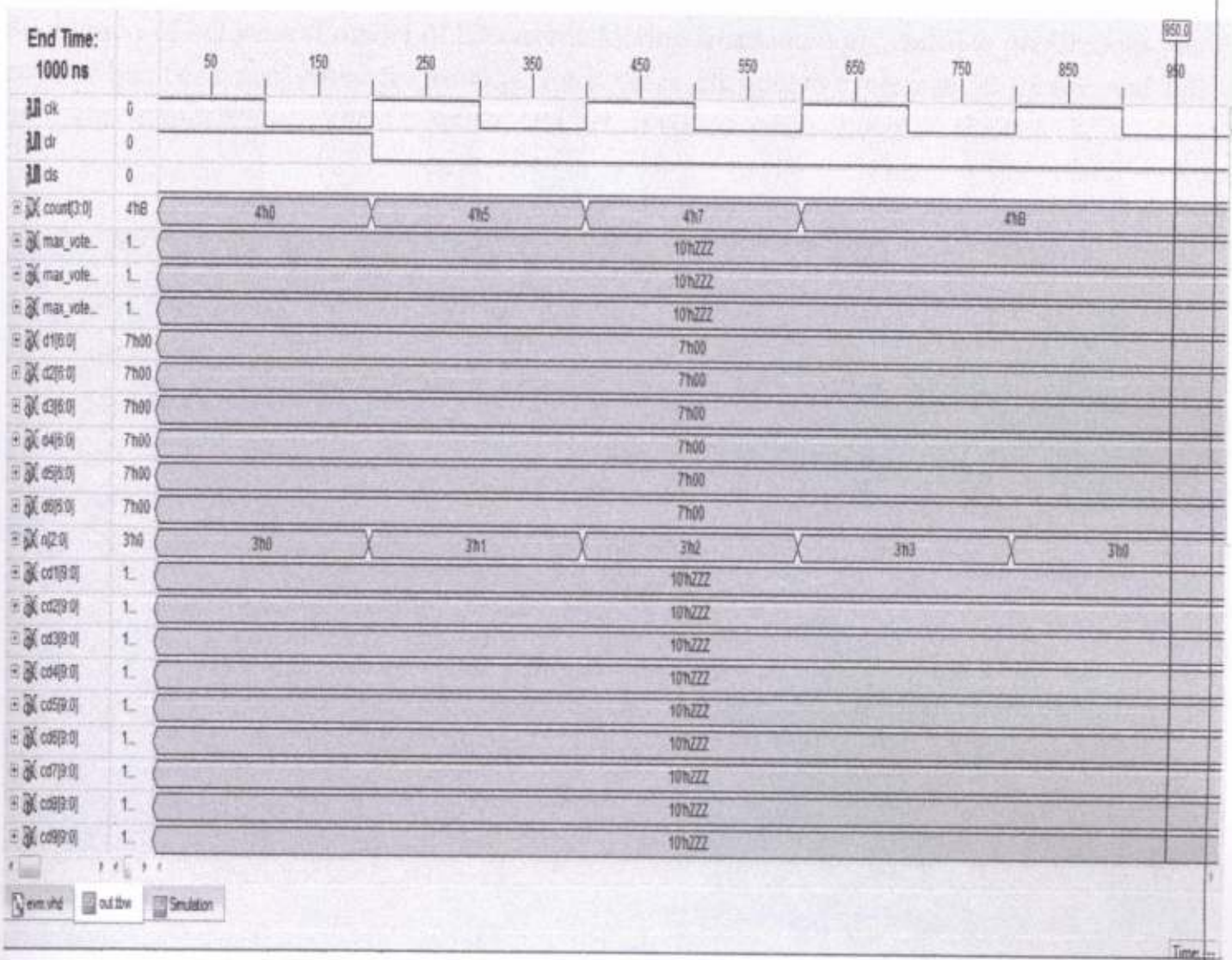
Counter is designed for incrementing the number of votes of each candidate after a voter cast his/her vote to a particular candidate



Design of comparator: The control unit compares the total votes of each candidate and displays the candidate no. having maximum vote in the seven segment display.

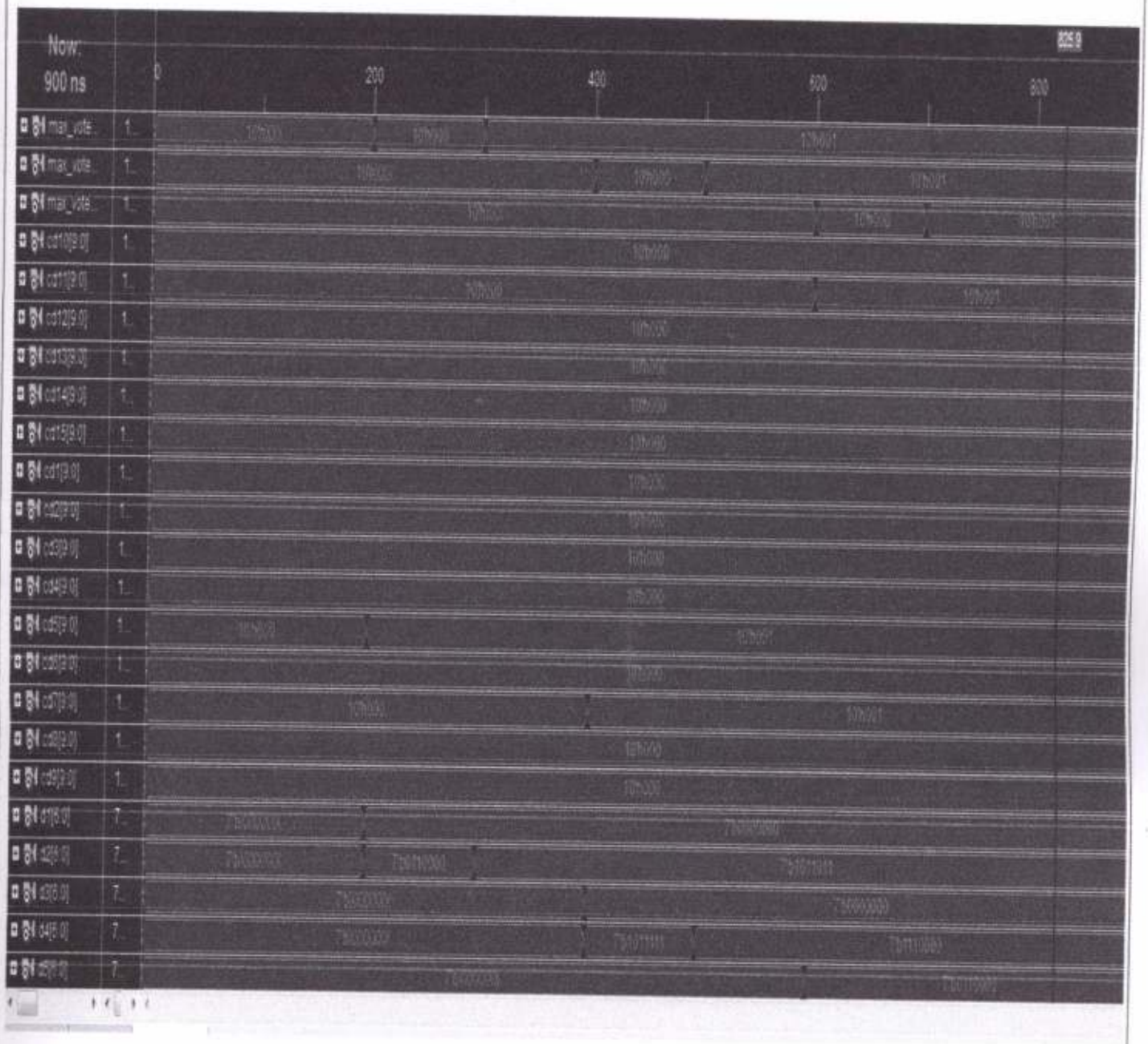


Test bench waveform


End Time:
1000 ns
50
150
250
350
450
550
650
750
850
950
950.0
clk
clr
cls
count[3:0]
max_vote...
d1[6:0]
d2[6:0]
d3[6:0]
d4[6:0]
d5[6:0]
d6[6:0]
n[2:0]
cd1[9:0]
cd2[9:0]
cd3[9:0]
cd4[9:0]
cd5[9:0]
cd6[9:0]
cd7[9:0]
cd8[9:0]
cd9[9:0]
4'h0
4'h5
4'h7
4'h8
3'h0
3'h1
3'h2
3'h3
10'hZZZ
7'h00
evm.vhd
out.tbw
Simulation

Result

Simulation of behavioral model of Electronic Voting Machine with Multiple preferences and priority has been performed for 1000ns. Each clock(clk) pulse cycle has 50 ns rise and fall time.The simulation generated from the test bench waveform is shown in figure4.



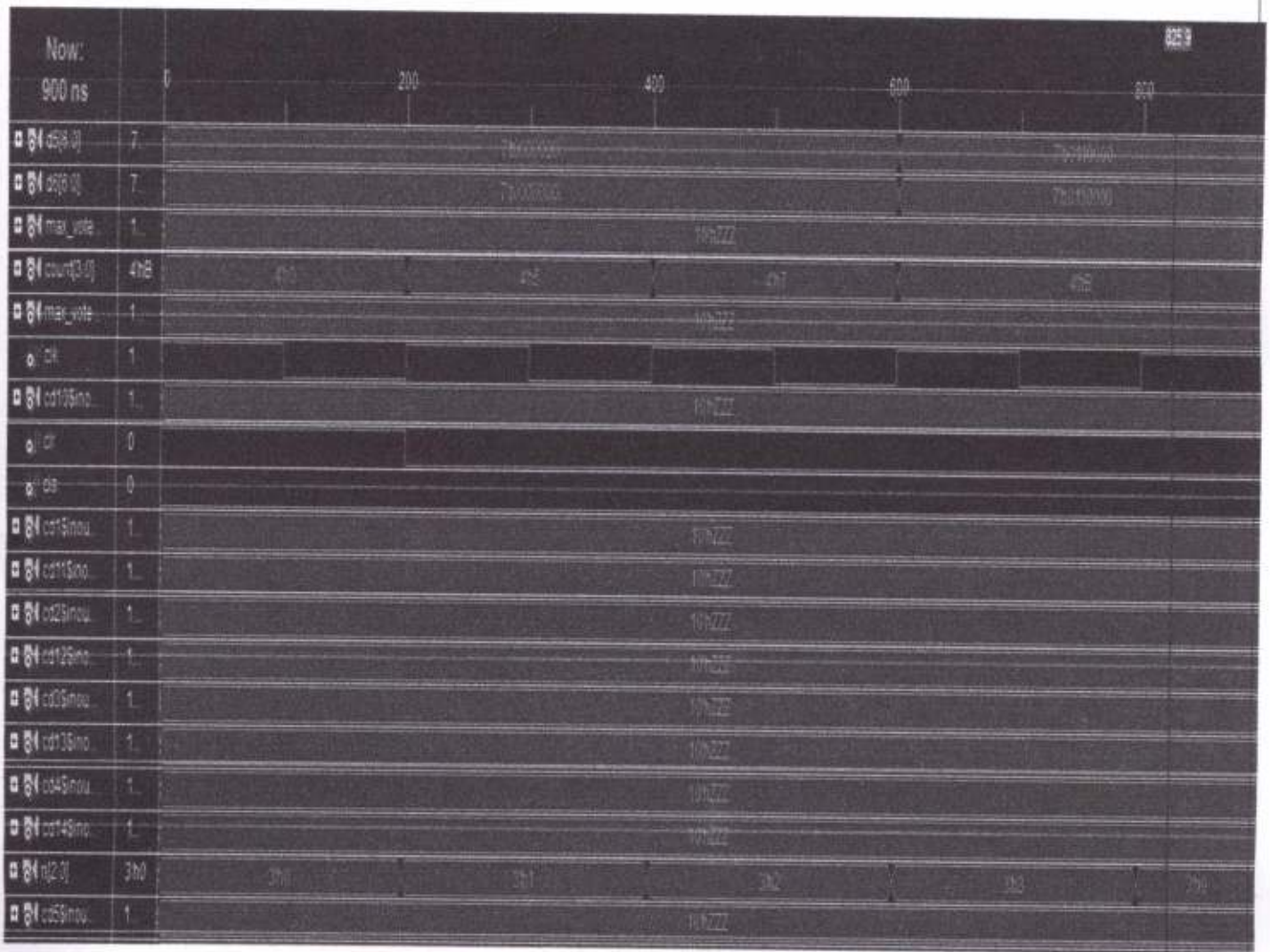
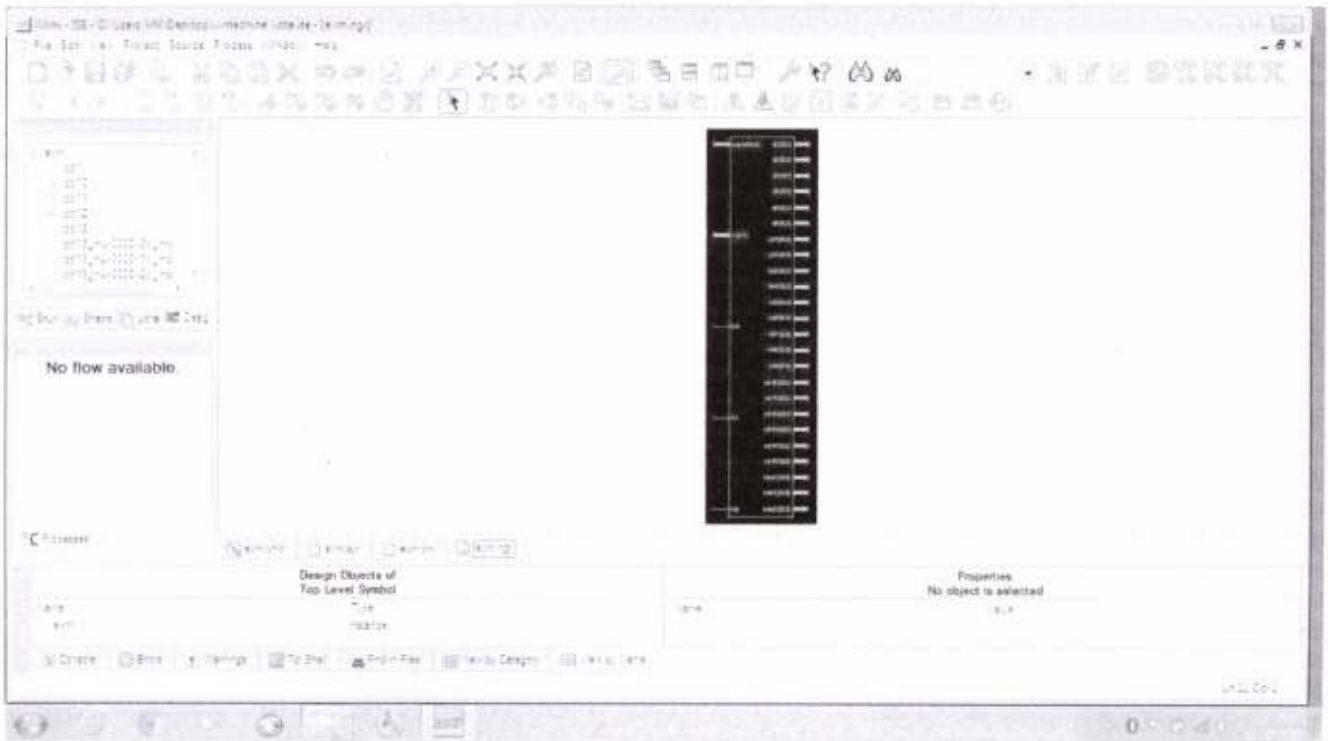
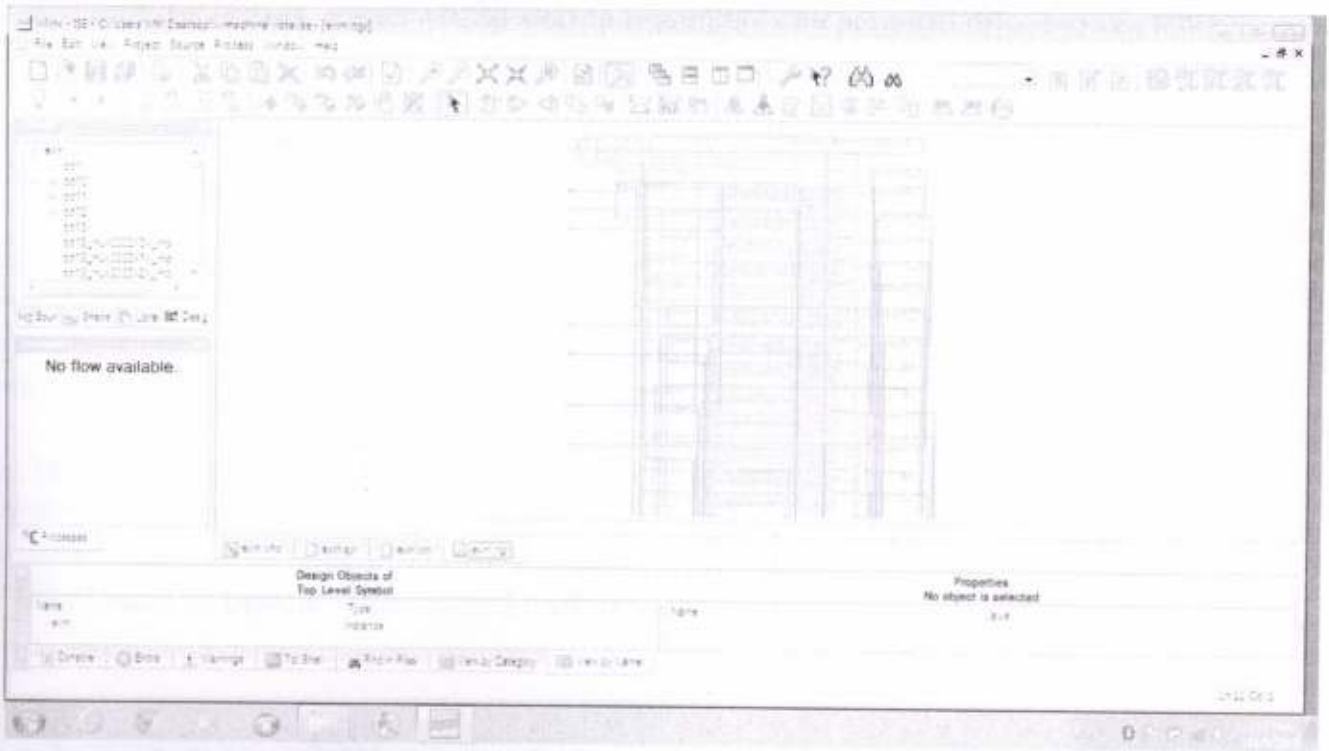


Figure4: Simulation from Test bench waveform

RTL Schematic



Schematic:


No flow available.
Design Objects of
Top Level Symbol
Properties
No object is selected

Advantages of Electronic Voting:

One of the advantages of electronic voting is that, in most cases, most ballots will be tabulated into the results. Paper-based voting machines can actually miss ballots because of human error in placing the paper-based ballot in the machine.

Another advantage is obviously the ease of tabulating the results. All counting and ordering is done by a machine, quickly and efficiently, and without human error.

DRE voting machines also have the advantage of never running out of paper ballots at a polling center, since the computer can count an unlimited number of ballots. They also can provide multiple languages to users who may not have English as a first language. This cannot be achieved through a paper-based system.

There are also advantages when dealing with people with disabilities, such as blindness. Electronic voting machines can provide headphones to read off instructions to the blind user. Also other tools can be added to these electronic voting machines to help with other disabilities such as people with limited mobility or the elderly.

Disadvantages of Electronic Voting:

Also, besides tampering with the machines electronically, machines could be tampered physically, with foreign software being uploaded into the machine by someone trying to corrupt the election results.

Another disadvantage of electronic voting systems could be the overall costs. Software, machines, installations, proper software protection, and validation of results could be expensive; even more expensive than paper-based machines.

There could also be just a general error in the system, without any outsider tampering. Computer software and systems can have problems that may delay or even halt voting, or may cause errors in calculations.

Future Enhancement

Future Enhancement of this project involves the security section of the EVM. The security section of the EVM is one of the main parts of the project where each voter will hold an indivisual voting card and his/her finger print will be the password. The computer will scan the identity of the voter and then the polling officer will allow the voter to cast his/her vote if found illegible.

Conclusion

In earlier elections, we as a voter, casted vote to our favourite candidate by putting the stamp against his/her name and then folding the ballot paper as per a prescribed method before putting it in the Ballot box. This is a long, time-consuming process and very much prone to errors. Polling by Electronic Voting Machine (EVM) is a simple, safe and secure method that takes minimum of time. In this article, Programmable EVM that accepts more than one vote from individual voter, depending on type of elections, is presented. This voting machine has an advantage that it can be used in all kinds of elections like grampanchyat, Co-operative societies, assembly elections etc. It can also consider priority as weightage when voter has capability to provide multiple preferences.

References

[1].Basic VLSI Design by Douglas A. Puncknell & Kamran Eshraghian

[2].http://www.xilinx.com/itp/xilinx10/books/docs/qst/qst.pdf

[3]. www.xilinx.com ISE Tutorial in Depth Chapter2, Chapter 3

[4]. www.xilinx.com ISE Tutorial in Depth Chapter4,Chapter 5,Chapter 6

[5].http://www.ece.wpi.edu/spartan3_Tutorial.pdf , accessed date- Mar 5 2103

[6].http://www.belIndia.com/BELWebsite/images/EVM_Features.pdf

Appendix

i. program for Electronic Voting Machine

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entityevm is
   Port ( clk : in  STD_LOGIC;
clr : in  STD_LOGIC;
                        cls : in  STD_LOGIC;
count : in  STD_LOGIC_VECTOR(3 downto 0);
                         max1,max2,max3: inout  STD_LOGIC_VECTOR(9 downto 0);
                         d1,d2,d3,d4,d5,d6: out  STD_LOGIC_VECTOR(6 downto 0);
                        n:in STD_LOGIC_VECTOR(2 downto 0);
      cd1,cd2,cd3,cd4,cd5,cd6,cd7,cd8,cd9,cd10,cd11,cd12,cd13,cd14,cd15: inout  STD_LOGIC_VECTOR (9
downto 0));
endevm;
```

```
architecture Behavioral of evm is

begin
process(clk,clr,count,n,cls, cd1,cd2,cd3,cd4,cd5,cd6,cd7,cd8,cd9,cd10,cd11,cd12,cd13,cd14,cd15,max1,max2,max3)
begin
if(clk='1' and clk'event) then
        if (clr='1') then
                cd1<="0000000000";cd2<="0000000000";cd3<="0000000000";
                cd4<="0000000000";cd5<="0000000000";cd6<="0000000000";
                cd7<="0000000000";cd8<="0000000000";cd9<="0000000000";
                cd10<="0000000000";cd11<="0000000000";cd12<="0000000000";

                cd13<="0000000000";cd14<="0000000000";cd15<="0000000000";
                max1<="0000000000";max2<="0000000000";max3<="0000000000";
                d1<="0000000";d2<="0000000";d3<="0000000";
                d4<="0000000";d5<="0000000";d6<="0000000";
                end if;
if(cls='0')then

        if ( n="01") then
                if (count="0001") then
                        cd1<=cd1+1;
                elsif (count="0010") then
                        cd2<=cd2+1;
                elsif (count="0011") then
                        cd3<=cd3+1;
```

```
        elsif (count="0100") then
                cd4<=cd4+1;

        elsif (count="0101") then
                cd5<=cd5+1;
        end if;

elsif (n="10") then
if (count="0110") then
                cd6<=cd6+1;
        elsif (count="0111") then
                cd7<=cd7+1;
        elsif (count="1000") then
                cd8<=cd8+1;
        elsif (count="1001") then
                cd9<=cd9+1;
        elsif (count="1010") then
                cd10<=cd10+1;
        end if;

elsif (n="11") then
        if (count="1011") then
                cd11<=cd11+1;
        elsif (count="1100") then
                cd12<=cd12+1;
        elsif (count="1101") then
                cd13<=cd13+1;
        elsif (count="1110") then
```

```
                cd14<=cd14+1;
            elsif (count="1111") then
                cd15<=cd15+1;
            end if;
        end if;
end if;
if (n="01") then

            max1<=cd1;
                d1<="0000000";
                d2<="0110000";

            if (max1=cd2)then
        d1<="1001111";
                d2<="1001111";
            elsif(max1<cd2)then
                max1<=cd2;
                d1<="0000000";
                d2<="1101101";

            elsif(max1=cd3)then
        d1<="1001111";
                d2<="1001111";
            elsif(max1<cd3)then
                max1<=cd3;
                d1<="0000000";
                d2<="1111001";
```

```
            elsif (max1=cd4)then
                    d1<="1001111";
                    d2<="1001111";

            elsif(max1<cd4)then
                    max1<=cd4;
                    d1<="0000000";
                    d2<="0110011";

            elsif (max1=cd5)then
        d1<="1001111";
                    d2<="1001111";
            elsif(max1<cd5)then
                    max1<=cd5;
                    d1<="0000000";
                    d2<="1011011";

    end if;
end if;

if (n="10") then
                max2<=cd6;
                    d3<="0000000";
                    d4<="1011111";

                if(max2=cd7)then
                    d3<="1001111";
```

```
            d4<="1001111";
elsif(max2<cd7)then
            max2<=cd7;
            d3<="0000000";
            d4<="1110000";

elsif(max2=cd8)then
            d3<="1001111";
            d4<="1001111";
elsif(max2<cd8)then
            max2<=cd8;
            d3<="0000000";
            d4<="1111111";

elsif(max2=cd9)then
            d3<="1001111";
            d4<="1001111";
elsif(max2<cd9)then
            max2<=cd9;
            d3<="0000000";
            d4<="1111011";

elsif(max2=cd10)then
            d3<="1001111";
            d4<="1001111";
elsif(max2<cd10)then
            max2<=cd10;
            d3<="0110000";
```

```
                d4<="0000000";
        end if;
end if;
if ( n="11") then
                max3<=cd11;
                        d5<="0110000";
                        d6<="0110000";

                if(max3=cd12)then
                        d5<="1001111";
                        d6<="1001111";
                elsif(max3<cd12)then
                        max3<=cd12;
                        d5<="0110000";
                        d6<="1101101";

                elsif(max3=cd13)then
                        d5<="1001111";
                        d6<="1001111";
                elsif(max3<cd13)then
                        max3<=cd13;
                        d5<="0110000";
                        d6<="1111001";

                elsif(max3=cd14)then
                        d5<="1001111";
                        d6<="1001111";
                elsif(max3<cd14)then
```

```
                max3<=cd14;
                d5<="0110000",
                d6<="0110011";

            elsif(max3=cd15)then
                d5<="1001111";
                d6<="1001111";
            elsif(max3<cd15)then
                max3<=cd15;
                d5<="0110000";
                d6<="1011011";
end if;
end if;
        end if;
end process;

end Behavioral;
```

ii. Xilinx 9.1i

Xilinx ISE(**I**ntegrated **S**oftware **E**nvironment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

Synthesis report:

Release 9.1i - xst J.30

Copyright (c) 1995-2007 Xilinx, Inc. All rights reserved.

--> Parameter TMPDIR set to ./xst/projnav.tmp

CPU : 0.00 / 0.10 s | Elapsed : 0.00 / 0.00 s

--> Parameter xsthdpdir set to ./xst

CPU : 0.00 / 0.10 s | Elapsed : 0.00 / 0.00 s

--> Reading design: evm.prj

TABLE OF CONTENTS


```
=========================================================================
*                     Synthesis Options Summary                         *
=========================================================================
---- Source Parameters
Input File Name                    : "evm.prj"
Input Format                       : mixed
Ignore Synthesis Constraint File   : NO
```

---- Target Parameters

Output File Name : "evm"

Output Format : NGC

Target Device : CoolRunner XPLA3 CPLDs

---- Source Options

Top Module Name : evm

Automatic FSM Extraction : YES

FSM Encoding Algorithm : Auto

Safe Implementation : No

CASE Implementation Style : Full-Parallel

Mux Extraction : YES

Resource Sharing : YES

---- Target Options

Add IO Buffers : YES

MACRO Preserve : YES

XOR Preserve : YES

Equivalent register Removal : YES

---- General Options

Optimization Goal : Speed

Optimization Effort : 1

Library Search Order : evm.lso

Keep Hierarchy : YES

RTL Output : Yes

Hierarchy Separator : /

Bus Delimiter : <>

Case Specifier : maintain

Verilog 2001 : YES

---- Other Options

Clock Enable : YES

wysiwyg

==========

* HDL Compilation *

==========

Compiling vhdl file "C:/Users/MK/Desktop/v.machine/evm.vhd" in Library work.

Entity <evm> compiled.

Entity <evm> (Architecture <behavioral>) compiled.

==========

* Design Hierarchy Analysis *

==========

Analyzing hierarchy for entity <evm> in library <work> (architecture <behavioral>).

==========

* HDL Analysis *

==========

Analyzing Entity <evm> in library <work> (Architecture <behavioral>).

Entity <evm> analyzed. Unit <evm> generated.

==========

* HDL Synthesis *

==========

Performing bidirectional port resolution...

Summary:

inferred 3 Counter(s).

inferred 12 Adder/Subtractor(s).

inferred 24 Comparator(s).

Unit <evm> synthesized.

HDL Synthesis Report

Macro Statistics

Adders/Subtractors : 12

10-bit adder : 12

Counters : 3

10-bit up counter : 3

Registers : 21

10-bit register : 15

7-bit register : 6

Comparators : 24

10-bit comparator equal : 12

10-bit comparator less : 12

Advanced HDL Synthesis Report

Macro Statistics

Adders/Subtractors : 12

10-bit adder : 12

Counters : 3

10-bit up counter : 3

Registers : 192

Flip-Flops : 192

Comparators : 12

10-bit comparator equal : 12

==========

* Final Report *

==========

Final Results

RTL Top Level Output File Name : evm.ngr

Top Level Output File Name : evm

Output Format : NGC

Optimization Goal : Speed

Keep Hierarchy : YES

Target Technology : CoolRunner XPLA3 CPLDs

Macro Preserve : YES

XOR Preserve : YES

Clock Enable : YES

wysiwyg : NO

Design Statistics

IOs : 232

Cell Usage :

BELS : 4466

AND2 : 1397

AND3 : 253

AND4 : 53

AND8 : 12

GND : 1

INV : 1619

OR2 : 732

OR3 : 90

OR4 : 52

OR8 : 2

XOR2 : 255

FlipFlops/Latches : 220

FDCE : 220

IO Buffers : 232

IBUF : 10

OBUF : 222

CPU : 7.25 / 7.36 s | Elapsed : 8.00 / 8.00 s

Total memory usage is 162036 kilobytes