Abu Dhabi University

CEN 466 - Advanced Digital Design

Project Report
Digital Clock

Author:
Muhammad Obaidullah 1030313

Supervisor:
Dr. Mohammed Assad Ghazal

Section 1

December 15, 2012

Abstract

In this project we were assigned to build a Digital Clock using VHDL code and the components provided to us.

1 Introduction

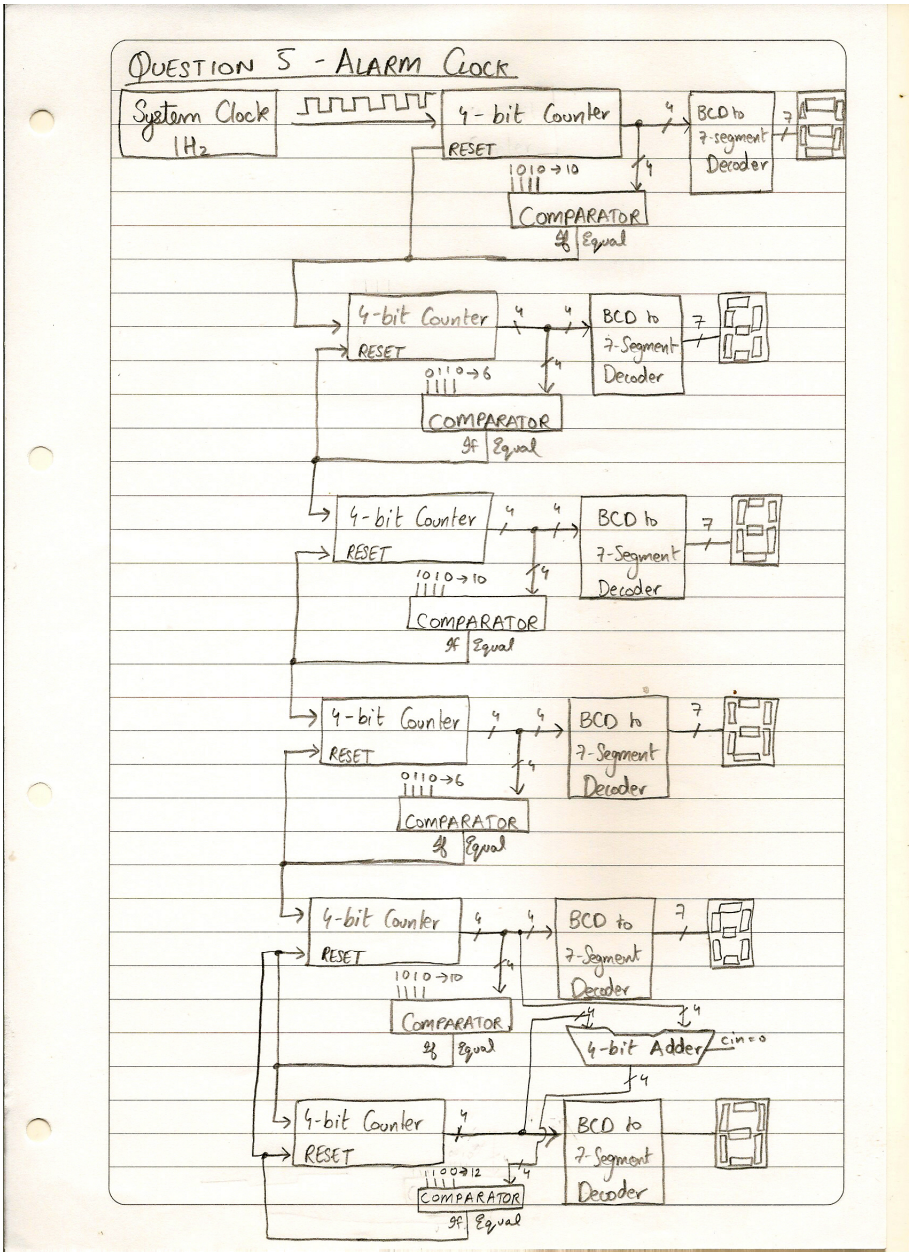
Typically it is possible to write a single VHDL code for the whole design you require, but using components as pre-programmed blocks to perform simple tasks is a intelligent and professional way to solve a design problem. Take for example a BCD to seven segment Decoder. This Decoder take a input of 4 bits which represents a number from 0-15 and converts it into a output which a seven segment display can understand and display. When designing larger systems, you can just take the output from your component and connect it to the input of this decoder and there you go, You have just displayed what the output was from your component in a seven segment display. By combining and embedding many such blocks of components, one can end up making sophisticated systems to solve huge problems.

2 List of Equipment used

- Computer.
- Quartus II Web Edition.
- Altera Cyclone II board.
- Cyclone II Pin Map.

3 Project Design Set-up

The aim of the project was to use the basic components provided to us from blackboard and use them to design a top level entity which contained and used these components. My design layout for the project was as follows:-



4 Procedure

- Open Quartus II and create a new project.
- Name the project to my liking but without spaces in the name.
- Download the VHDL code for Seven Segment Display Decoder, 4-bit comparator, 4-bit counter, and Clock Divider from blackboard and include them in my project directory.
- Click on new and create a blank VHDL file.
- Name the new VHDL file to be the same as the project name. This is essential as this VHDL file will be set as my top level entity.
- now write the following code into the VHDL file.

```
1  library ieee;
   use ieee.std_logic_1164.all;
3
   entity FifteenSecondsTimer is
5  port(
7
       hour0 : out   std_logic_vector(6 downto 0);
9      hour1 : out   std_logic_vector(6 downto 0);
       minute0 : out   std_logic_vector(6 downto 0);
11     minute1 : out   std_logic_vector(6 downto 0);
       second0 : out   std_logic_vector(6 downto 0);
13     second1 : out   std_logic_vector(6 downto 0);
       clk : in    std_logic
15 );
   end FifteenSecondsTimer;
17
   architecture FifteenSecondsTimer_arch of FifteenSecondsTimer is
19
   component SevenSegmentDecoder is
21 port(
       output: out   std_logic_vector(6 downto 0);
23     input : in    std_logic_vector(3 downto 0)

25 );
   end component;
27
   component clk_div IS
29 PORT(
       clock_50Mhz   : IN  STD_LOGIC;
31     clock_1MHz    : OUT STD_LOGIC;
       clock_100KHz  : OUT STD_LOGIC;
33     clock_10KHz   : OUT STD_LOGIC;
       clock_1KHz    : OUT STD_LOGIC;
35     clock_100Hz   : OUT STD_LOGIC;
       clock_10Hz    : OUT STD_LOGIC;
37     clock_1Hz   : OUT STD_LOGIC);
   END component;
39
   component counter is
41 port(
```

```
    clock:  in  std_logic;
    clear:  in  std_logic;
    count:  in  std_logic;
    Q:    out std_logic_vector(3 downto 0)
);
end component;


component Comparator is
port(
    A:      in  std_logic_vector(3 downto 0);
    B:      in  std_logic_vector(3 downto 0);
    less:   out std_logic;
      equal:    out std_logic;
    greater:  out std_logic
);
end component;



signal sec0, sec1, min0, min1, hr0, hr1: std_logic_vector(3 downto 0);
signal reset0,reset1, reset2, reset3, reset4, reset5, myclocksignal : std_logic;



begin


ClockDivider: clk_div   port map(clock_50Mhz=>clk, clock_1Hz=>myclocksignal);

---- FOR 1ST SECONDS DIGIT ----

s1: counter          port map(clock=>myclocksignal, clear=>reset0,count=>'1',Q=>sec0)
    ;
s2: SevenSegmentDecoder   port map(input=>sec0,output=>second0);
s3: Comparator         port map(A=>sec0,B=>"1010",equal=>reset0);

---- FOR 2ND SECONDS DIGIT ----

s4: counter          port map(clock=>reset0,clear=>reset1,count=>'1',Q=>sec1);
s5: SevenSegmentDecoder   port map(input=>sec1,output=>second1);
s6: Comparator         port map(A=>sec1,B=>"0110",equal=>reset1);

---- FOR 1ST MINUTES DIGIT ----

m1: counter          port map(clock=>reset1,clear=>reset2,count=>'1',Q=>min0);
m2: SevenSegmentDecoder   port map(input=>min0,output=>minute0);
m3: Comparator         port map(A=>min0,B=>"1010",equal=>reset2);

---- FOR 2ND MINUTES DIGIT ----

m4: counter          port map(clock=>reset2,clear=>reset3,count=>'1',Q=>min1);
m5: SevenSegmentDecoder   port map(input=>min1,output=>minute1);
m6: Comparator         port map(A=>min1,B=>"0110",equal=>reset3);

---- FOR 1ST HOURS DIGIT ----

h1: counter          port map(clock=>reset3,clear=>reset4,count=>'1',Q=>hr0);
h2: SevenSegmentDecoder   port map(input=>hr0,output=>hour0);
```

```
h3: Comparator          port map(A=>hr0,B=>"1010",equal=>reset4);

---- FOR 2ND HOURS DIGIT ----

h4: counter           port map(clock=>reset4,clear=>reset5,count=>'1',Q=>hr1);
h5: SevenSegmentDecoder    port map(input=>hr1,output=>hour1);
h6: Comparator          port map(A=>hr1,B=>"0010",equal=>reset5);




end FIFteenSecondsTimer_arch;
```

5 Results and Discussions

At the end of the project, I uploaded the code onto the Altera board and got the following results:-

- Build was complete with some warnings but no errors.
- At first the segments were showing inverted outputs but I figured it out that I had planned the pins in the reverse order.
- For testing purposes I used 100Hz or sometimes 10KHz clock output from the clock divider to quickly reach hours so that I can test the hours working.
- The seven segment display of the Altera board uses inverted input. For example for lighting up 1 on the segment display you have to give input to it "1111001"

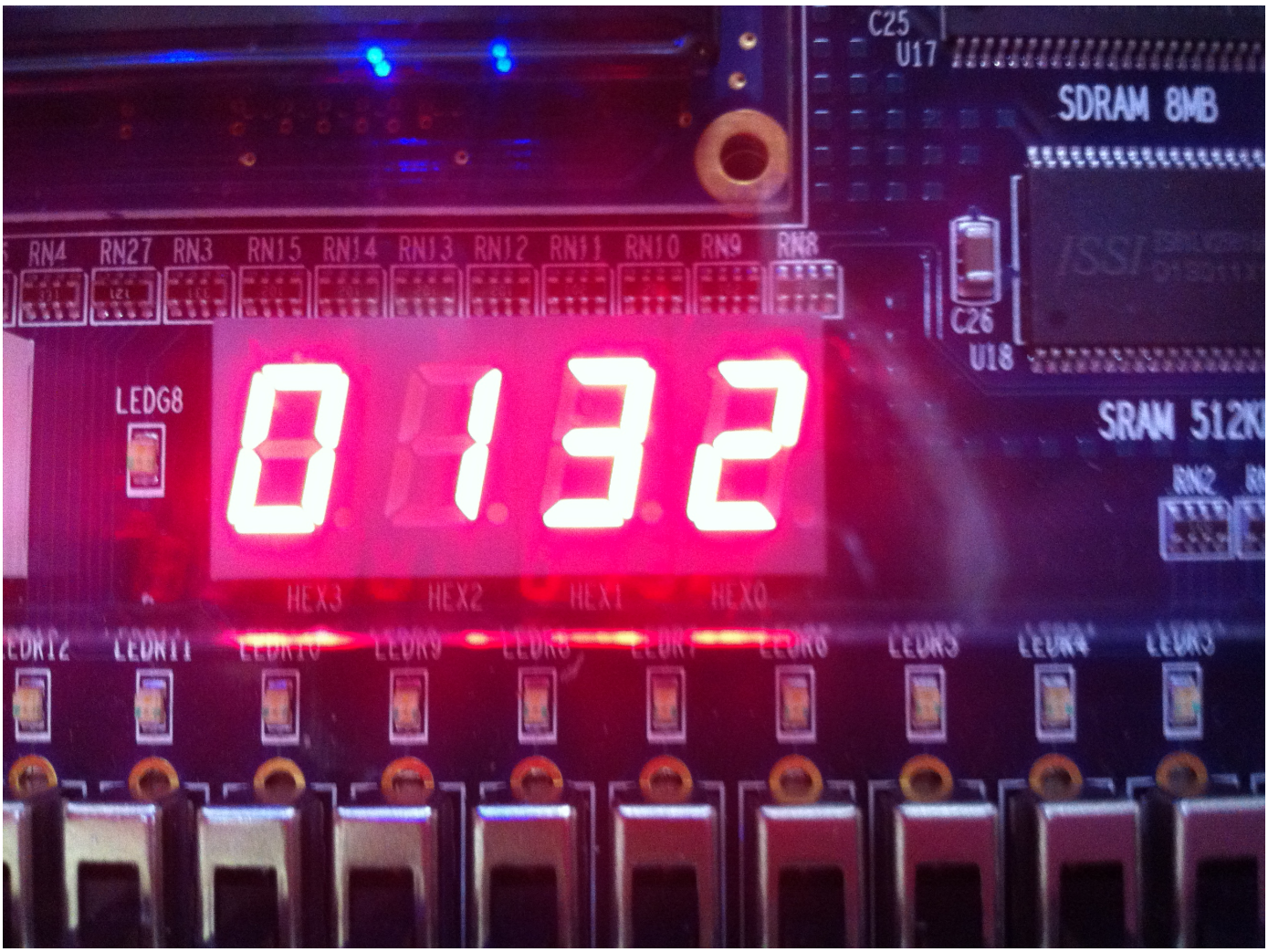


Figure 1: The clock is now showing 32 seconds past one minute

Figure 2: The left two seven segment displays are for seconds and the right ones are for the minutes

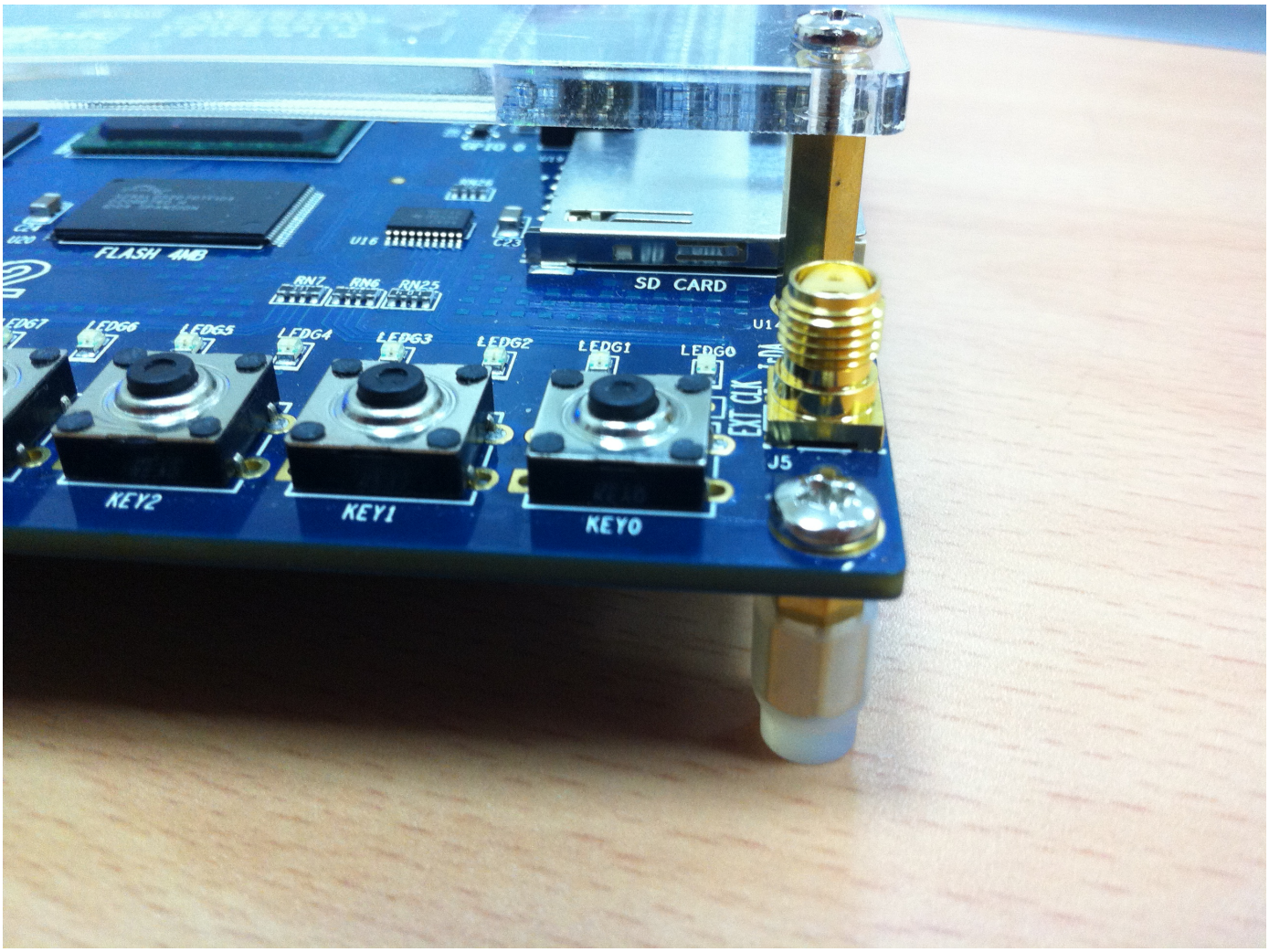


Figure 3: Initially I used the push button as the clock but it got tiring and was slow. So I used a 100Hz clock from the clock divider to just test out all the digits were working fine or not

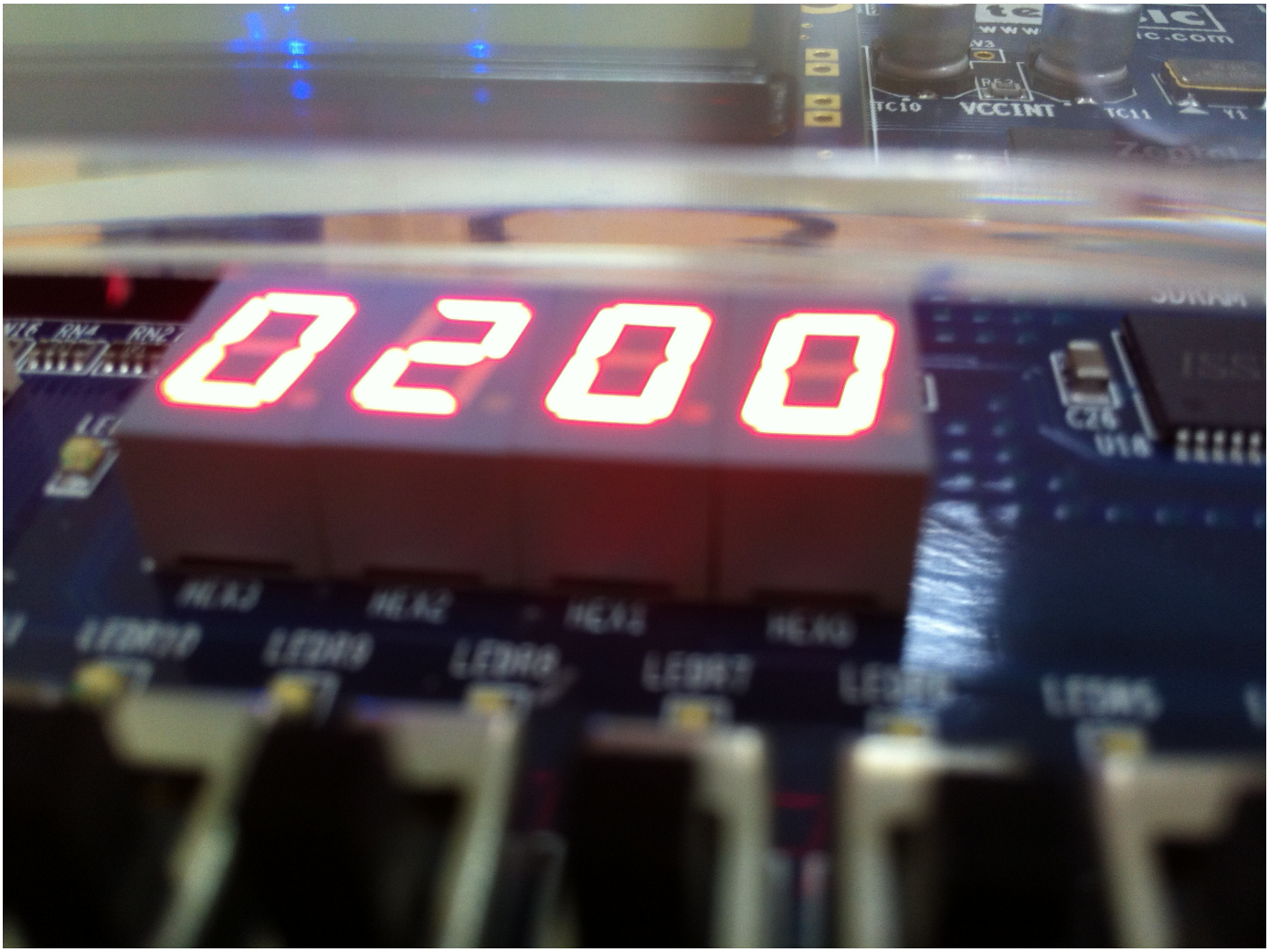


Figure 4: The seven segment display showing the clock now at 02:00 minutes mark

6 Conclusion

- This type of implementation is called the combinational implementation where I have used combinational circuits to make a large entity.
- Statistically, Combinational implementations are more faster than the sequential implementations because sequential implementations use sequence and timing to execute the instructions which is time consuming and depends heavily on the hardware's CPU clock.

7 For more Information:

Please visit this link for a video of the running clock.
`http://youtu.be/36MtfkjD7cE`