NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

"DESIGN OF DIGITAL CLOCK"

A MINI PROJECT REPORT

SUBMITTED BY:

KAVYASHREE B (1NH18EC725)

C KEERTHI (1NH18EC708)

SOURABH SRIKUMAR (1NH18EC749)

J AKHIL KUMAR (1NH18EC137)

In partial fulfillment for the reward of the degree

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

NEW HORIZON COLLEGE OF ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the mini project work entitled "DESIGN OF A DIGITAL CLOCK" carried out by **KAVYASHREE B (1NH18EC725), C KEERTHI (1NH18EC708), SOURABH SRIKUMAR (1NH18EC749), J AKHIL KUMAR (1NH18EC137),** bonafide students of Electronics and Communication Department , New Horizon College of Engineering, Bangalore.

The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project Guide	HOD ECE
-------------------------	-------------------------

External Viva

Name of Examiner	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **ASHOK K** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

KAVYASHREE B (1NH18EC725)

C KEERTHI (1NH18EC708)

SOURABH SRIKUMAR (1NH18EC749)

J AKHIL KUMAR (1NH118EC137)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The system expected to be built consists of a 6-digit clock, including hours, minutes, and seconds, and encompasses very favorable space with very cheap equipment. The design contains all the features a digital clock should contain, and the overall delay is negligible as the design integrates.

The module has two inputs - a clock with a frequency of 1 Hz and an active high reset. There are three p outputs to tell time - seconds, minutes and hours.

The time units are incremented in an always block using Behavioral modeling. At every clock cycle we increment 'seconds'.Whenever seconds reaches the value '60' we increment 'minutes' by 1.Similarly whenever minutes reach '60' we increment 'hours' by 1.Once hours reaches the value '23' we reset the digital clock.

CHAPTER 1
INTRODUCTION

TIME is a basic concept that is difficult to define. To measure time, something must be repeated at regular intervals. The number of calculated time periods gives a quantitative measure of the duration. The nearest signs to measure time are the moon and the sun. When the sun and moon were not visible, it was impossible to know the exact time.

Therefore, the watches were developed to measure hours between checks with the sun and moon. The process of measuring time became progressively more accurate, and devices have become more localized ever since. In modern times, time is often measured by mechanical, most recently in electronic watches. All watches measure time, but different hours have status or importance.

Many centuries were spent devising a method for identifying and measuring time. Historically, watches and clocks of all kinds are at an important crossroads between science, technology and society. Changes in timing technology affected the nature of scientific observation, helped in developing other machine techniques and brought important reviews of how people think and act in a timely manner. In this paper, the most accurate watch is presented using FPGA.Dgitial clock is designed using vhdl and verilog. it consists of onew input and three outputs,it converts 50MHz clock to 1Hz.

Hardware Description Language(HDL):

In computer engineering, HDL is a specialized computer language used to describe the structure and behavior of electronic circuits, and the most common digital logic circuits.You can also synthesize HDL descriptions into netlists (specifications of physical electronic components and how they are connected), place and route them, and generate a set of masks used to create integrated circuits. Hardware description languages are very similar to programming languages such as C and ALGOL. This is a text description consisting of expressions, ads, and control structures.An important most programming languages and HDLs is that HDL clearly includes the concept of time. difference between HDL forms an integral part of electronic design automation (EDA) systems, especially for complex circuits such as application-specific integrated circuits, microprocessors, and programmable logic devices.

VHDL:

VHDL (VHSIC-HDL, hardware description language for very high speed integrated circuits) is a hardware description language used in electronic design automation to describe digital and mixed signal systems such as field programmable gate arrays and integrated circuits. VHDL can also be used as a parallel programming language for general purposes.

VERILOG:

Verilog, also known as IEEE 1364, is a hardware description language (HDL) used to model electronic systems. It is commonly used in the design and testing of digital circuits at the log transfer abstract level. It is used in the confirmation of analog circuits and mixed signal circuits and in the design of genetic circuits.

In 2009, Verilog Standard (IEEE 1364-2005) merged with Verilog Standard and created the IEEE 1800-2009 Standard. Subsequently, the Verilog system is practically part of the Verilog language. The current version is the IEEE 1800-2017 Standard.

CHAPTER 2
LITERATURE SURVEY

The digital system takes a complex approach to humanity in all areas of life. In some areas, machines cannot replace machines with exact accuracy. For a long time, humans have used analogue clocks in their daily lives.The first digital pocket watch was the invention of Austrian engineer Joseph Palweber, who created his "jump-hour" system in 1883. Instead of a traditional dial, the jump-hour features two windows on an enamel dial, so that the hours and minutes are visible on rotating disks. The second hand remained traditional.
By 1885, the Polweber system had been marketed in pocket watches by Kartabert and IWC; Controversy Despite the arguments of the toothbrush jump hour movement, the continued growth and commercial success of contributions until the appearance of the 1920 Wristwatches (Chŕnewschwiedegieter) are still in use today.Although the original inventor no longer owns a watch brand, his name has been raised by the newly established watchmaker.

Plato Clocks used a similar idea, but another lay. These spring-wound pieces consist of a column inside a glass cylinder, in which are placed small digital cards, in which the numbers are printed, which flip over time.In 1904, the St. Louis world premiered the Plato Clocks, produced by the Ansonia Clock Company. In 1903 Eugene Fitch of New York patented the design of the clock. 13 years ago Joseph Palweber patented digital cards (unlike the 1885 patent) in Germany for the same invention (DRP No. 54093).The German factory manufactured such digital clocks in 1893 and 1894. The first patent for the digital alarm clock was filed in the United States on October 23, 1956. E. Registered by Protsmanetal.In 1970, Pratzmann and his associates patented another digital clock, which is said to have very few moving parts. There were digital numbers between the two side plates, while an electric motor and cam gear were out of control.

The evolution of digital technology has been astounding over the years, creating digital system designs that in many respects continue to serve as good sources comfort and comfort for humankind. Nowadays, many applications in electronics and other technologies use digital techniques to perform operations that were once performed in analog fashion.
Digital systems are more versatile and versatile than analog systems and are based on the fact that they are immune to voltage spurious variations. High precision and accuracy. You can store billions of bits of information in a relatively small space. Some authors have done extensive research to minimize power requirements and achieve high flexibility and performance in integrated chip manufacturing. FPGA board development is part of the research.
This white paper explores flexible implementations using FPGAs, and digital logic on Altera DE2 boards presented as an education and development board to better and safely use these specifications during design implementation. I will introduce an experimental implementation of the design. A report produced from the FPGA Tutorial "Overview on FPGA" states that FPGAs provide users with a way to configure and that these specifications provide fuzzy logic with a large amount of logic in a single IC.The intersection between logical blocks and the function of each logical block. FPGA logic blocks can be configured to provide simple functions such as transistors or complex functions such as microprocessors. It can be used to implement various combinations of combinatorial and sequential logic functions.

Parth Mehta's research paper reports on the design of a simple digital clock in the Spartan 3 FPGA kit. The digital clock can only display time in minutes and seconds with more delay.Ina 4MHz, it uses only 1hz frequency with a 22 bit register.
Nowadays blocks like current counters, decoders and multipliers are used. Instead of displaying only seconds and minutes from this paper, we can display the total number of hours, minutes and seconds. We can also create a higher frequency clock from the kit's default frequency. That way we can get a better digital clock with the alarm clock from the kit's default frequency. That way we can get a better digital clock with the alarm.

Title of the paper	**Author & Year of Publication**	**Outcome**	**Limitation**
JUMP-HOUR	AUTRAIN ENGINEER JOSEPH PALLWEBER -1883	The jump-hour featured two windows in an enamel dial, through which the hours and minutes are visible on rotating discs.	Conventional Dial
Design of a simple digital clock	ParthMetha	The digital clock displays the time only in minutes and seconds. It uses only 1hz frequency with 22 bit register.	Hour won't be displayed.

Table 2.1 LITERATURE SURVEY

CHAPTER 3

PROPOSED METHODOLOGY

Parth Mehta's research paper reports on the design of a simple digital clock in the Spartan 3 FPGA kit. The digital clock can only display time in minutes and seconds with more delay .In a 4MHz, it uses only 1hz frequency with a 22 bit register. It uses blocks like counter, decoder and multiplier that already exist.From this model we can improve some things like instead of displaying only secs and mins we can display total hours,mins,secs. It is also possible to generate multiple frequency clocks from the default frequencies of the kit. This will further improve your digital clock.

3.1 BLOCK DIAGRAM

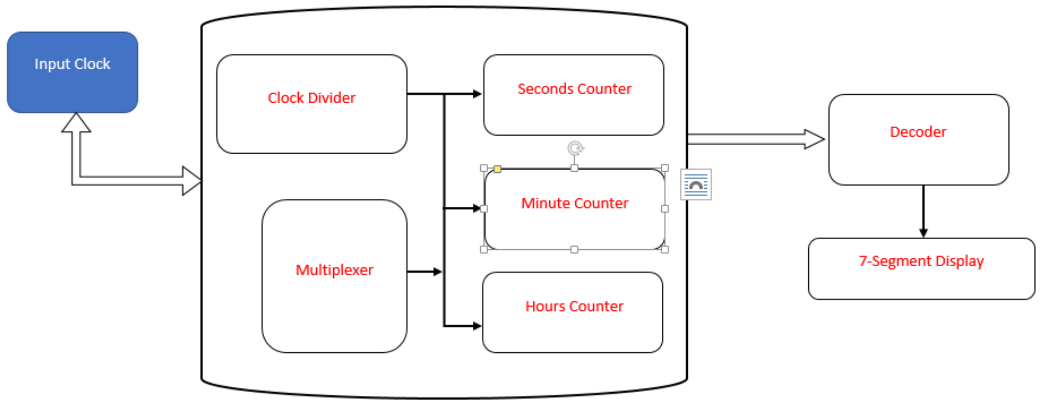


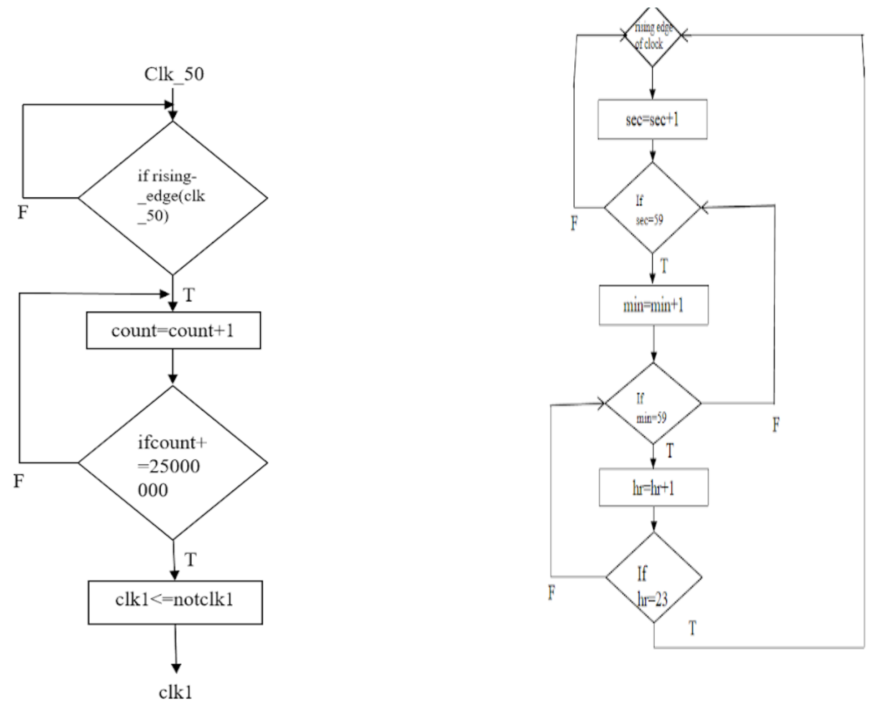
Figure 3.1 Block Diagram of VHDL Code

The digital clock module is the main control for the digital clock. it accepts one input as 50MHz.

- clock divider- it converts 50MHz to 1Hz
- multiplexer - it checks the condition which has to excecuted
- seconds counter - increments seconds upto 59 and then goes to 0
- minutes counter - increments minutes upto 59 and the goes to 0

hours counter - increments hours upto 23 and then goes to 0

3.2 FLOWCHART



clock division counters

Figure 3.2 flowchart of VHDL module

When the input clock enters the second value will increase but up to 59 and then zero again.Likewise the minute value will also increase after the second value reaches 59, but up to 59. The hour value will increase when the minute value reaches 59 and rises to 23 and again goes to zero.

CHAPTER 4

PROJECT DESCRIPTION

4.1 SOFTWARE DESCRIPTION:

Software used is Xilinx ISE 14.7

XILINX ISE (Integrated Control Environment) is a software tool produced by XILINX to collect and analyze HDL designs, enabling the developer to adjust (`` aggregate '') their designs, Performing timing analysis, checking RTL maps, simulating design interaction with different stimuli, and configuring the target device with the programmer.

VHDL CODE EXPLANATION:

It accepts an input as a 50 MHz clock and provides three outputs such as Hour, Minute and Second. This code internally converts 50 MHz into 1 Clock frequency. In this code the first process converts the frequency from 50 MHz to 1 Hz. In the second process at each clock event the second value will increase but up to 59 and then zero again.Likewise the minute value will also increase after the second value reaches 59, but up to 59. The hour value will increase when the minute value reaches 59 and rises to 23 and again goes to zero. In the last integer of ss, mm and h are converted into a standard logic vector and assigned respectively to Second, Minute and Hour.

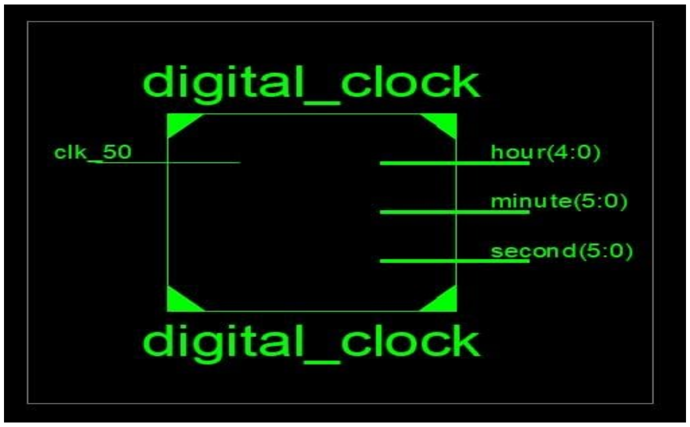


Figure 4.1 RTL schematic diagram

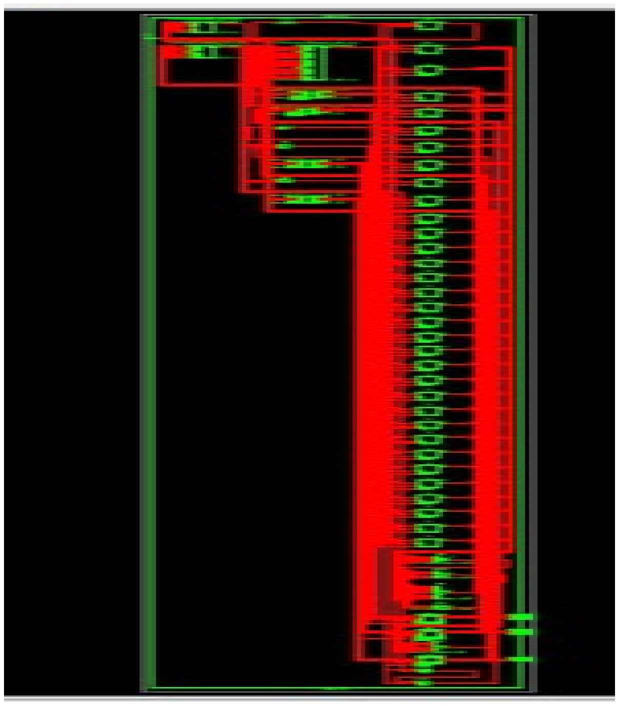


Figure 4.2 Technology schematic

VERILOG CODE EXPLANATION:

The unit has two inputs: 1hz frequency clock and high active reset. There are three PP outputs for hours: seconds, minutes and hours.Using Behavioral Modeling will always increase the time units in the block. We increase 'seconds' with each clock cycle. Whenever the second reaches the value of '60', we increase the 'minute' by 1. Whenever the minutes reach '60' we increase the 'hour' 1. When an hour reaches the '23' value we set the digital clock reset. The Verilog code for the digital clock is given below:

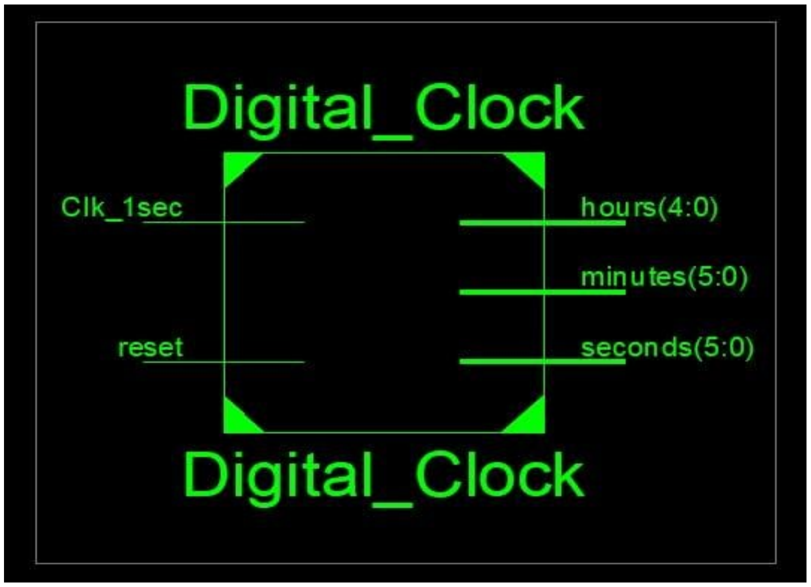


Figure 4.3 RTL SCHEMATIC

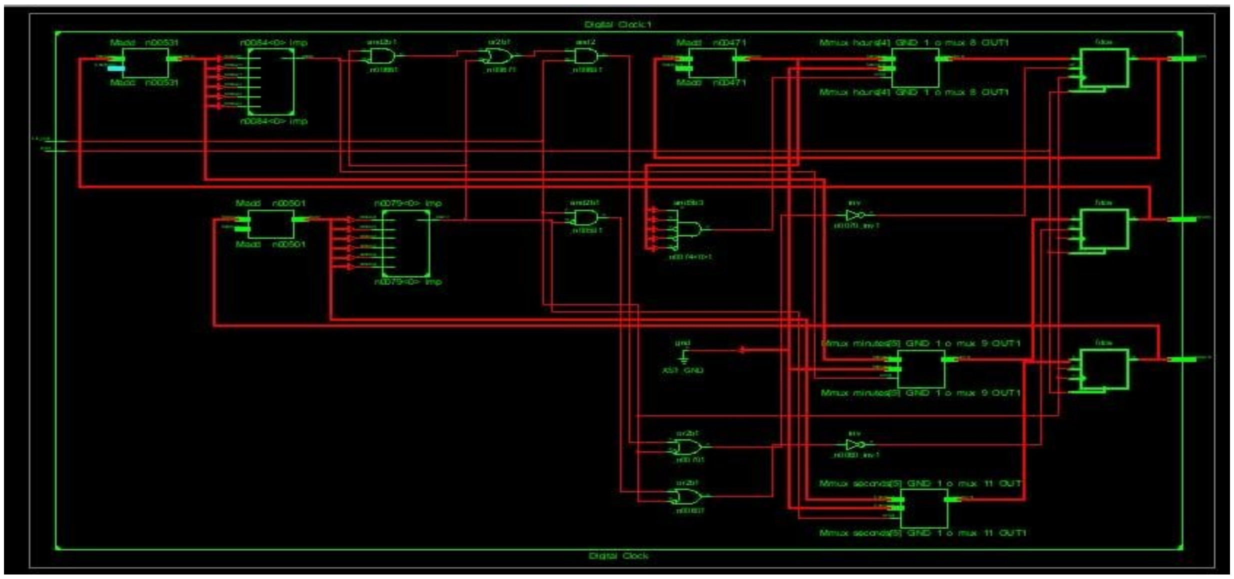


Figure 4.4 TECHNOLOGY SCHEMATIC

4.2 HARDWARE DESCRIPTION:

The hardware component we use here is a FPGA

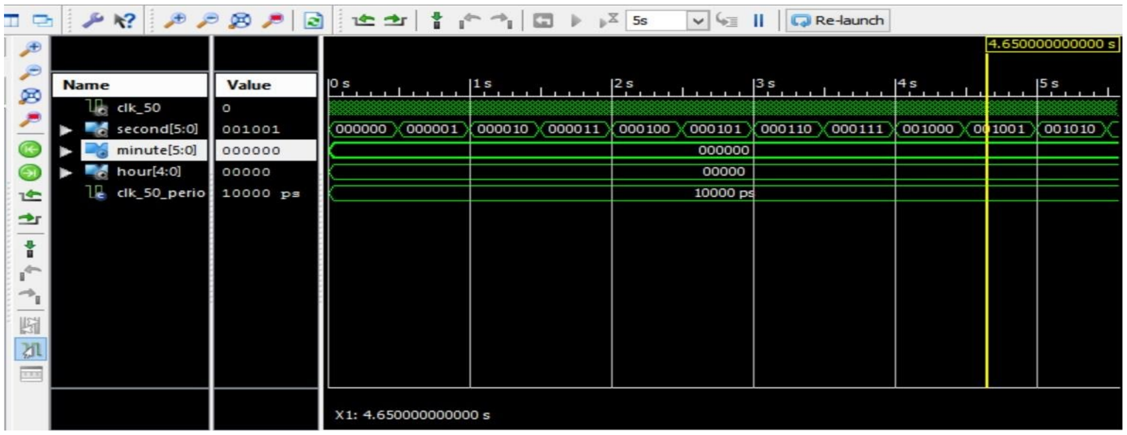
FPGA - Field Programmable Gate Array

FPGA Stands for Field Programmable Gate Array consists of a set of logic gates that store the specified digital logic from the hardware description language. FPGAs can be reproduced to the desired application or to the functional requirements after production. This feature separates FPGAs from custom-built application-specific integrated circuits (ASICs) for specific design tasks.Although single-programmable (OTP) FPGAs are available, the main types are SRAM-based, which can be reproduced as the design develops.

This is an overview of the field programmable gate array. This is a semiconductor IC, where most electrical activity within the device can be changed; The design engineer was replaced, the PCB changed during the assembly process, or even after the equipment was exported to the users field. It consists 144pins.ASIC and FPGA have different value propositions and should be carefully weighed before choosing one over the other.There is ample information to compare the two technologies. While FPGAs were once chosen for designs of lower speed / complexity / size, today FPGAs easily drive the 500 MHz performance barrier. With an unprecedented increase in Boolean density and a host of other features, such as integrated processors, DSP blocks, watches, and a high-speed serial at lower price points than ever before, FPGA is a compelling offering for almost any type of design.

Chapter :5

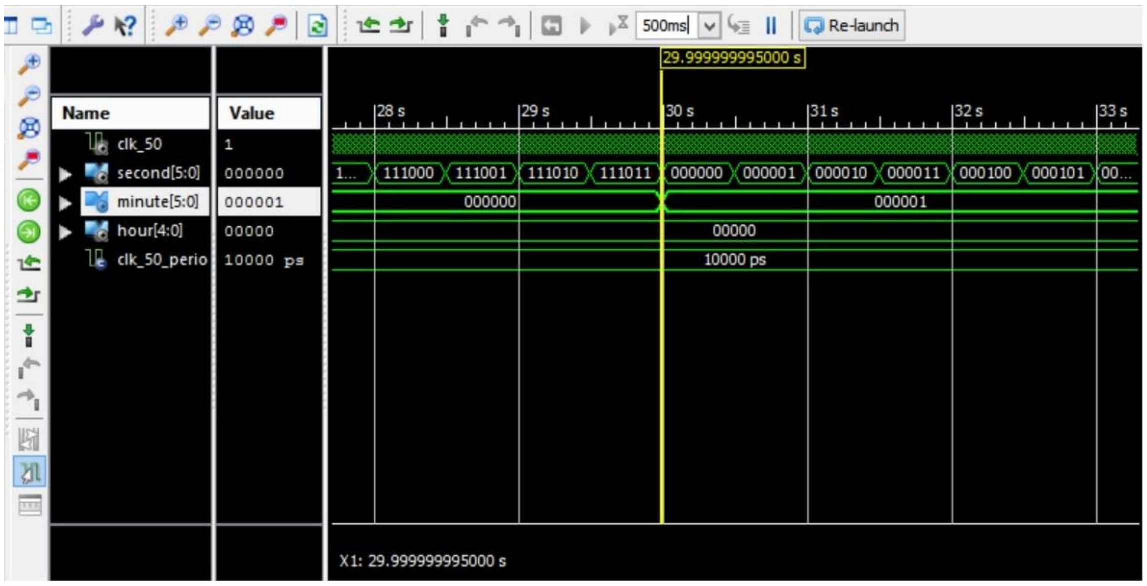
Results and Discussion



Name	Value
clk_50	0
second[5:0]	001001
minute[5:0]	000000
hour[4:0]	00000
clk_50_perio	10000 ps

Figure 5.1 output of VHDL module(seconds counter)

For the positive edge of the clock it is observed that seconds counter is incremented by 1



Name	Value
clk_50	1
second[5:0]	000000
minute[5:0]	000001
hour[4:0]	00000
clk_50_perio	10000 ps

Figure 5.2 output of VHDL module(minutes counter)

It is observed that when seconds counter reaches 59,it comes back to 0 and minutes counter increments by 1.

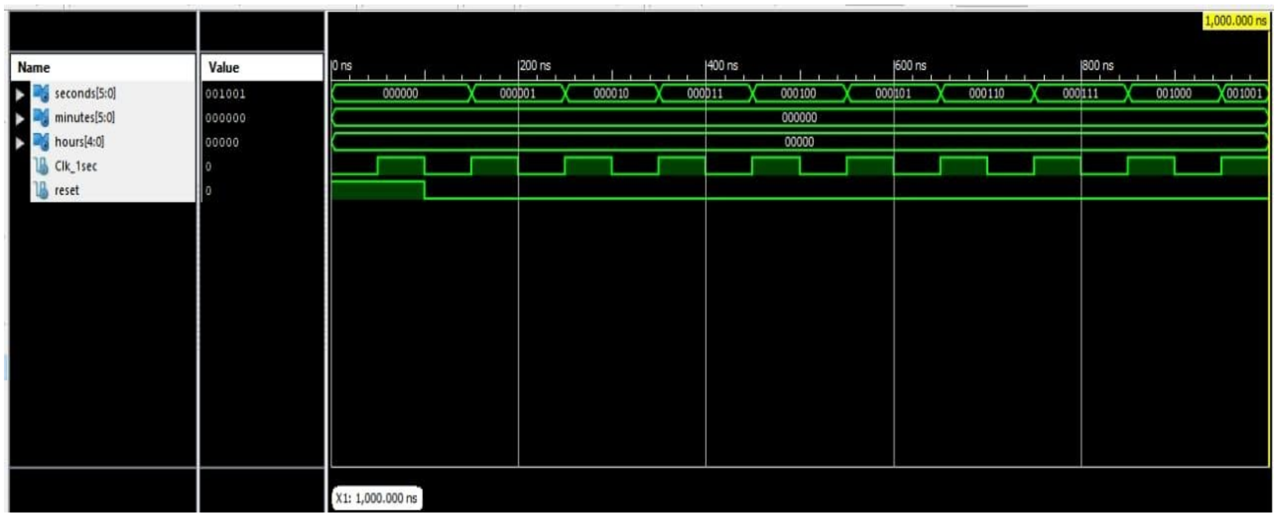


Figure 5.3 output of Verilog module(seconds counter)

For a positive clock pulse seconds counter increments by 1 upto 59.

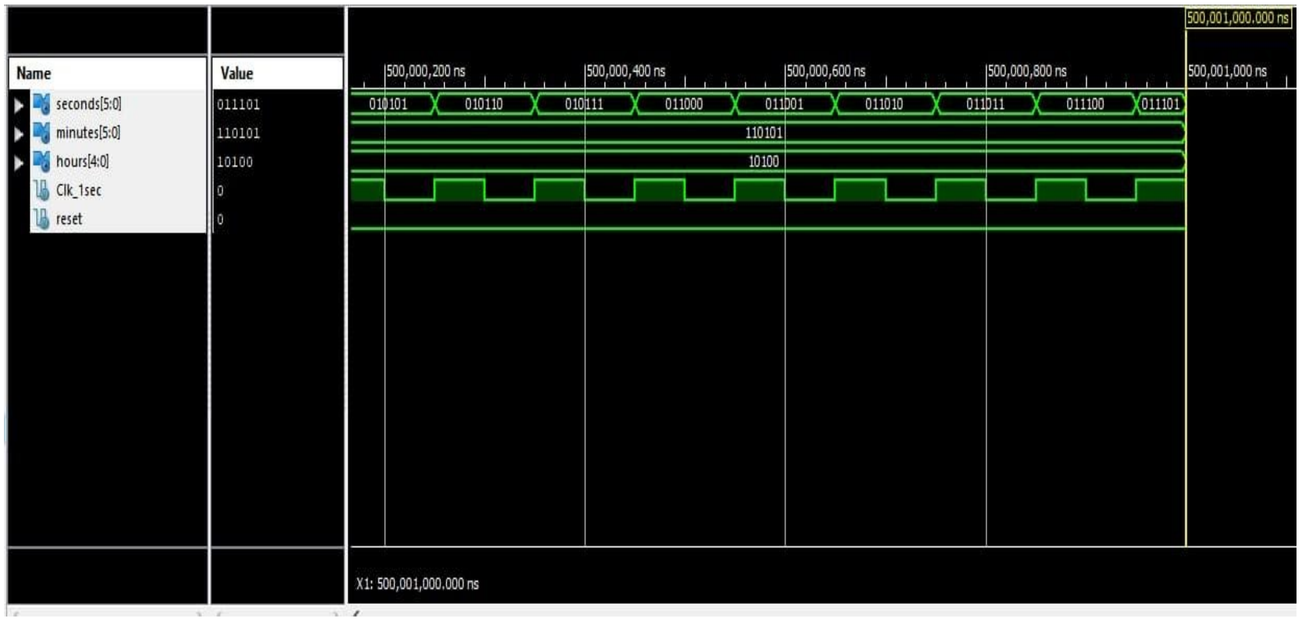


Figure 5.4 output of Verilog code(minutes and hour counter)

It is observed that when seconds counter reaches 59,minutes counter increments by 1 upto 59 and then hour counter increments by upto 23.

SYNTHESIS REPORT

Release 14.7 - xst P.20131013 (nt64)

Copyright (c) 1995-2013 Xilinx, Inc. All rights reserved.

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

Total REAL time to Xst completion: 3.00 secs

Total CPU time to Xst completion: 3.13 secs

--> Parameter xsthdpdir set to xst

Total REAL time to Xst completion: 3.00 secs

Total CPU time to Xst completion: 3.13 secs

--> Reading design: digital_clock.prj

TABLE OF CONTENTS

```
     4.1) HDL Synthesis Report

  5) Advanced HDL Synthesis

     5.1) Advanced HDL Synthesis Report

  6) Low Level Synthesis

  7) Partition Report

  8) Design Summary

     8.1) Primitive and Black Box Usage

     8.2) Device utilization summary

     8.3) Partition Resource Summary

     8.4) Timing Report

        8.4.1) Clock Information

        8.4.2) Asynchronous Control Signals Information

        8.4.3) Timing Summary

        8.4.4) Timing Details

        8.4.5) Cross Clock Domains Report



==============================================================================

*                    Synthesis Options Summary                    *

==============================================================================

---- Source Parameters

Input File Name                    : "digital_clock.prj"

Ignore Synthesis Constraint File   : NO


---- Target Parameters
```

```
Output File Name                  : "digital_clock"

Output Format                    : NGC

Target Device                   : xc7a100t-3-csg324


---- Source Options

Top Module Name                  :digital_clock

Automatic FSM Extraction          : YES

FSM Encoding Algorithm            : Auto

Safe Implementation              : No

FSM Style                         : LUT

RAM Extraction                   : Yes

RAM Style                        : Auto

ROM Extraction                   : Yes

Shift Register Extraction         : YES

ROM Style                         : Auto

Resource Sharing                 : YES

Asynchronous To Synchronous       : NO

Shift Register Minimum Size       : 2

Use DSP Block                        : Auto

Automatic Register Balancing      : No


---- Target Options

LUT Combining                    : Auto

Reduce Control Sets              : Auto

Add IO Buffers                     : YES
```

```
Global Maximum Fanout            : 100000

Add Generic Clock Buffer(BUFG)     : 32

Register Duplication             : YES

Optimize Instantiated Primitives   : NO

Use Clock Enable                 : Auto

Use Synchronous Set              : Auto

Use Synchronous Reset            : Auto

Pack IO Registers into IOBs       : Auto

Equivalent register Removal       : YES


---- General Options

Optimization Goal                : Speed

Optimization Effort              : 1

Power Reduction                  : NO

Keep Hierarchy                   : No

Netlist Hierarchy                :As_Optimized

RTL Output                       : Yes

Global Optimization              :AllClockNets

Read Cores                       : YES

Write Timing Constraints         : NO

Cross Clock Analysis             : NO

Hierarchy Separator              : /

Bus Delimiter                    :<>

Case Specifier                   : Maintain

Slice Utilization Ratio          : 100
```

```
BRAM Utilization Ratio            : 100

DSP48 Utilization Ratio           : 100

Auto BRAM Packing                 : NO

Slice Utilization Ratio Delta      : 5


=========================================================================



=========================================================================

*                          HDL Parsing                              *

=========================================================================

Parsing VHDL file "E:\mini\finalvh\finalvh\finalvh.vhd" into library work

Parsing entity <digital_clock>.

Parsing architecture <beh> of entity <digital_clock>.


=========================================================================

*                          HDL Elaboration                          *

=========================================================================


Elaborating entity <digital_clock> (architecture <beh>) from library <work>.


=========================================================================

*                          HDL Synthesis                            *

=========================================================================
Synthesizing Unit <digital_clock>.
    Related source file is "E:\mini\finalvh\finalvh\finalvh.vhd".
    Found 1-bit register for signal <clk1>.
    Found 6-bit register for signal <second>.
Found 6-bit register for signal <minute>.
Found 5-bit register for signal <hour>.
    Found 32-bit register for signal <count>.
    Found 32-bit adder for signal <count[31]_GND_5_o_add_0_OUT> created at line 24.
    Found 5-bit adder for signal <hr[4]_GND_5_o_add_7_OUT> created at line 43.
    Found 6-bit adder for signal <mm[5]_GND_5_o_add_9_OUT> created at line 46.
    Found 6-bit adder for signal <ss[5]_GND_5_o_add_12_OUT> created at line 49.
    Summary:
            inferred   4 Adder/Subtractor(s).
            inferred  50 D-type flip-flop(s).
Unit <digital_clock> synthesized.


=========================================================================
HDL Synthesis Report


Macro Statistics
# Adders/Subtractors                    : 4
 32-bit adder                           : 1
 5-bit adder                            : 1
 6-bit adder                            : 2
# Registers                             : 5
```

```
 1-bit register                        : 1
 32-bit register                        : 1
 5-bit register                        : 1
 6-bit register                        : 2

=========================================================================

=========================================================================
*                    Advanced HDL Synthesis                      *
=========================================================================


Synthesizing (advanced) Unit <digital_clock>.
The following registers are absorbed into counter <count>: 1 register on signal <count>.
The following registers are absorbed into counter <ss>: 1 register on signal <ss>.
The following registers are absorbed into counter <mm>: 1 register on signal <mm>.
The following registers are absorbed into counter <hr>: 1 register on signal <hr>.
Unit <digital_clock> synthesized (advanced).

=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Counters                             : 4
 32-bit up counter                          : 1
 5-bit up counter                           : 1

 6-bit up counter                           : 2

# Registers                                : 1

 Flip-Flops                                : 1


=========================================================================


=========================================================================

*                       Low Level Synthesis                       *

=========================================================================


Optimizing unit <digital_clock> ...


Mapping all equations...

Building and optimizing final netlist ...

Found area constraint ratio of 100 (+ 5) on block digital_clock, actual ratio is 0.


Final Macro Processing ...


=========================================================================

Final Register Report


Macro Statistics

# Registers                                : 50

 Flip-Flops                                : 50
```

```
=========================================================================

=========================================================================
*                        Partition Report                        *
=========================================================================

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.

-------------------------------

=========================================================================
*                         Design Summary                          *
=========================================================================

Top Level Output File Name         :digital_clock.ngc

Primitive and Black Box Usage:
------------------------------
# BELS                             : 142
#      GND                         : 1
#      INV                         : 4
#     LUT1                     : 31
#     LUT2                     : 22
#     LUT3                     : 3
#     LUT4                     : 3
#     LUT5                     : 3
#     LUT6                     : 11
#     MUXCY                      : 31
#     VCC                     : 1
#     XORCY                     : 32
# FlipFlops/Latches               : 50
#     FD                     : 16
#     FDR                     : 23
#     FDRE                     : 11
# Clock Buffers                  : 1
#     BUFGP                     : 1
# IO Buffers                     : 17
#     OBUF                     : 17

Device utilization summary:
---------------------------

Selected Device : 7a100tcsg324-3

Slice Logic Utilization:
 Number of Slice Registers:             50  out of  126800     0%
 Number of Slice LUTs:                  77  out of  63400     0%
    Number used as Logic:               77  out of  63400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     77
   Number with an unused Flip Flop:      27  out of     77    35%
   Number with an unused LUT:            0  out of     77     0%
   Number of fully used LUT-FF pairs:    50  out of     77    64%
   Number of unique control sets:        5

IO Utilization:
 Number of IOs:                        18
 Number of bonded IOBs:                18  out of    210     8%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:              1  out of     32     3%

---------------------------
Partition Resource Summary:
---------------------------

  No Partitions were found in this design.

---------------------------
```

```
=========================================================================

Timing Report


NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.

      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT

GENERATED AFTER PLACE-and-ROUTE.


Clock Information:

------------------

-----------------------------------+------------------------+-------+

Clock Signal                       | Clock buffer(FF name)  | Load  |

-----------------------------------+------------------------+-------+

clk_50                             | BUFGP                  | 33    |

clk1                               | NONE(ss_0)             | 17    |

-----------------------------------+------------------------+-------+

INFO:Xst:2169 - HDL ADVISOR - Some clock signals were not automatically buffered by XST with BUFG/BUFR resources. Please use the buffer_type constraint in order to insert these buffers to the clock signals to help prevent skew problems.


Asynchronous Control Signals Information:

----------------------------------------

No asynchronous control signals found in this design


Timing Summary:
```

```
---------------

Speed Grade: -3


   Minimum period: 2.681ns (Maximum Frequency: 372.992MHz)

   Minimum input arrival time before clock: No path found

   Maximum output required time after clock: 0.673ns

   Maximum combinational path delay: No path found


Timing Details:

---------------

All values displayed in nanoseconds (ns)


=========================================================================

Timing constraint: Default period analysis for Clock 'clk_50'

  Clock period: 2.681ns (frequency: 372.992MHz)

  Total number of paths / destination ports: 1585 / 50

-------------------------------------------------------------------------

Delay:               2.681ns (Levels of Logic = 2)

  Source:            count_19 (FF)

  Destination:       count_16 (FF)

  Source Clock:      clk_50 rising

  Destination Clock: clk_50 rising


  Data Path: count_19 to count_16

                           Gate     Net
```

```
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)

    ----------------------------------------  ------------

     FDR:C->Q            2   0.361   0.697  count_19 (count_19)

             LUT6:I0->O                            1        0.097       0.693     GND_5_o_count[31]_equal_2_o<31>1
(GND_5_o_count[31]_equal_2_o<31>)

             LUT6:I0->O                          33        0.097       0.386     GND_5_o_count[31]_equal_2_o<31>7
(GND_5_o_count[31]_equal_2_o)

FDR:R                  0.349        count_16

    ----------------------------------------

    Total                   2.681ns (0.904ns logic, 1.777ns route)

                                (33.7% logic, 66.3% route)


=========================================================================

Timing constraint: Default period analysis for Clock 'clk1'

  Clock period: 2.316ns (frequency: 431.760MHz)

  Total number of paths / destination ports: 346 / 45

-------------------------------------------------------------------------

Delay:              2.316ns (Levels of Logic = 2)

  Source:           mm_5 (FF)

  Destination:      mm_0 (FF)

  Source Clock:     clk1 rising

  Destination Clock: clk1 rising


  Data Path: mm_5 to mm_0

                            Gate     Net

Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
```

```
    ---------------------------------------- ------------
     FDRE:C->Q          3   0.361   0.703  mm_5 (mm_5)
     LUT6:I0->O         2   0.097   0.383  PWR_5_o_mm[5]_equal_6_o<5>1 (PWR_5_o_mm[5]_equal_6_o)
     LUT2:I0->O        11   0.097   0.325  _n00511 (_n0051)
FDRE:R               0.349          mm_0
    ----------------------------------------
    Total                  2.316ns (0.904ns logic, 1.412ns route)
                             (39.0% logic, 61.0% route)


=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk1'
  Total number of paths / destination ports: 17 / 17
-------------------------------------------------------------------------
Offset:            0.673ns (Levels of Logic = 1)
  Source:          ss_0 (FF)
  Destination:     second<0> (PAD)
  Source Clock:    clk1 rising


  Data Path: ss_0 to second<0>
                          Gate     Net
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ---------------------------------------- ------------
     FDR:C->Q           8   0.361   0.311  ss_0 (ss_0)
OBUF:I->O             0.000          second_0_OBUF (second<0>)
    ----------------------------------------
  Total                  0.673ns (0.361ns logic, 0.311ns route)
                           (53.7% logic, 46.3% route)

=========================================================================

Cross Clock Domains Report:
--------------------------

Clock to Setup on destination clock clk1
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
clk1           |    2.316|         |         |         |
---------------+---------+---------+---------+---------+

Clock to Setup on destination clock clk_50
---------------+---------+---------+---------+---------+
               | Src:Rise| Src:Fall| Src:Rise| Src:Fall|
Source Clock   |Dest:Rise|Dest:Rise|Dest:Fall|Dest:Fall|
---------------+---------+---------+---------+---------+
clk_50         |    2.681|         |         |         |
---------------+---------+---------+---------+---------+

=========================================================================
Total REAL time to Xst completion: 58.00 secs

Total CPU time to Xst completion: 57.44 secs


-->


Total memory usage is 4655708 kilobytes


Number of errors   :    0 (   0 filtered)

Number of warnings :    0 (   0 filtered)

Number of infos  :    1 (   0 filtered)
```

CHAPTER : 6
APPLICATIONS AND ADVANTAGES:

6.1 APPLICATIONS:

Digital clocks are used in different kinds of devices such as

- cars
- radios
- televisions
- microwave ovens
- computers
- cell phones

6.2 ADVANTAGES:

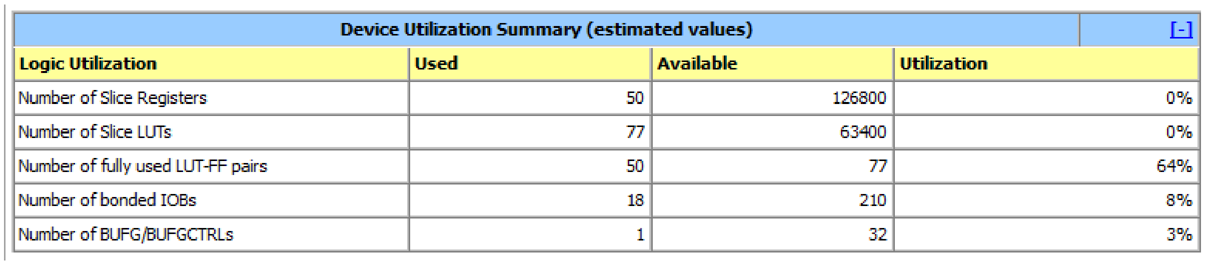
- It is less expensive
- Easy to understand time concept.
- No signal losses due to DA and AD conversion geometry,clock and phase settings is unnecessary therefore simple to use where as in anolog time lags.
- Compared to traditional mechanical watches, electronic watches have many advantages. With the development of digital integrated circuits and the use of advanced quartz technology, electronic watches have the advantages of accurate operating time, stable performance and comfortable carrying. Electronic clocks are used for automatic alarms and calls on time. Areas such as automatic time program control, timing transmission, and automatic control.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONSLUSION:

The high performance digital clock design is proposed with reduces core area.The block digram of clock is designed with the combination of mutltipier,counter,clock divider and decoder.The blocks are implemented using HDL and verified using simulator.we used vhdl and verilog HDL to implement the code part and has really helped,since it combines hardware and software part,it also provides infpormative graphs and waveforms which are helpul in understanding the real concepts of the project .Xilinx also has proved to be the most learnable tool for simulation and a great integrated development environment.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	50	126800	0%
Number of Slice LUTs	77	63400	0%
Number of fully used LUT-FF pairs	50	77	64%
Number of bonded IOBs	18	210	8%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 7.1 Design utilization summary

Total real time to Xst completion:58.00 secs

Total CPU time to Xst completion:57.44 secs

Total delay:1.66 secs

7.2 FUTURE ENHANCEMENT:

This program can be improvised by adding alarm module ,stopwatch module . The result can be verified using MATLAB linking and uploading the VHDL design through Modelsim on the simulink. The VHDL optimized based design can be generated by creating the test bench for the generated Simulink model and using the in-built optimization techniques of Matlab.

REFERENCE :

[1] D. W. Bliss, P. A. Parker, A. R. Margetts, "Simultaneous transmission and reception for improved wireless network performance", Statistical Signal Processing 2007 IEEE/SP 14th Workshop on, pp. 478-482, 2007.

[2] DIGITAL SYSTEM DESIGN using VHDL -Charles H. Roth, Jr. &Lizy Kurian John.

[3] https://www.academia.edu/35206724/Implementation_of_Digital_Clock_on_FPGA

[4] https://ukdiss.com/examples/digital-clock-implementation-with-fpga.php

[5] http://eceprojectsbtechstds.blogspot.com/2018/11/digital-clock-using-verilog.html

[6] https://www.pantechsolutions.net/implementation-of-digital-clock-using-spartan3an-fpga-evaluation-kit

[7] http://electronicsinourhands.blogspot.com/2012/10/digital-clock-using-vhdl.html

APPENDIX :

VHDL CODE

```
libraryieee;
use ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entitydigital_clock is
port(clk_50 : in std_logic;
second : out std_logic_vector(5 downto 0);
minute : out std_logic_vector(5 downto 0);
hour : out std_logic_vector(4 downto 0));
end entity;

architecturebeh of digital_clock is
signalss, mm : integer range 0 to 59;
signalhr : integer range 0 to 23;
signal count : integer := 1;
signal clk1 :std_logic := '1';
begin
-- 1 Hz clock Generation from 50 MHz clock.
```

```
clock_generation : process (clk_50)
begin
if (rising_edge(clk_50)) then
count<= count + 1;
if (count = 25000000) then
        clk1 <= not clk1;
        count<= 1;
end if;
end if;
end process;

-- Functionality of Digital Clock.
digital : process (clk1)
begin
if (rising_edge(clk1)) then
if (ss = 59) then
        ss<= 0;
        if (mm = 59) then
mm<= 0;
if (hr = 23) then
hr<= 0;
else
hr<= hr + 1;
end if;
        else
mm<= mm + 1;
        end if;
```

```
else
        ss<= ss + 1;
end if;
end if;
end process;
--Converts Integer Values into Standard Logic Vector;
second<= conv_std_logic_vector(ss,6);
minute<= conv_std_logic_vector(mm,6);
hour<= conv_std_logic_vector(hr,5);
endbeh;
```

Test bench :

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;
ENTITY finalvh_tb IS
END finalvh_tb;
ARCHITECTURE behavior OF finalvh_tb IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT digital_clock
PORT(
        clk_50 : IN  std_logic;
second : OUT  std_logic_vector(5 downto 0);
minute : OUT  std_logic_vector(5 downto 0);

hour : OUT  std_logic_vector(4 downto 0)

        );
    END COMPONENT;


   --Inputs

signal clk_50 :std_logic := '0';

--Outputs
```

VERILOG CODE

```
module Digital_Clock(
Clk_1sec, //Clock with 1 Hz frequency

reset,   //active high reset

    seconds,

    minutes,

    hours);


//What are the Inputs?

    input Clk_1sec;

    input reset;

//What are the Outputs?

    output [5:0] seconds;

    output [5:0] minutes;

    output [4:0] hours;
```

```
//Internal variables.
   reg [5:0] seconds;
   reg [5:0] minutes;
   reg [4:0] hours;


  //Execute the always blocks when the Clock or reset inputs are
   //changing from 0 to 1(positive edge of the signal)
   always @(posedge(Clk_1sec) or posedge(reset))
   begin
if(reset == 1'b1) begin  //check for active high reset.
         //reset the time.
         seconds = 0;
         minutes = 0;
         hours = 0;  end
       else if(Clk_1sec == 1'b1) begin  //at the beginning of each second
         seconds = seconds + 1; //increment sec
if(seconds == 60) begin //check for max value of sec
            seconds = 0;  //reset seconds
            minutes = minutes + 1; //increment minutes
if(minutes == 60) begin //check for max value of min
               minutes = 0;  //reset minutes
               hours = hours + 1;  //increment hours
if(hours ==  24) begin  //check for max value of hours
                  hours = 0; //reset hours
```

```
                end
              end
            end
          end
        end

endmodule
```

Testbench

```
module tb_clock;

    // Inputs
    reg Clk_1sec;
    reg reset;

    // Outputs
    wire [5:0] seconds;
    wire [5:0] minutes;
    wire [4:0] hours;

    // Instantiate the Unit Under Test (UUT)
Digital_Clockuut (
.Clk_1sec(Clk_1sec),
.reset(reset),
.seconds(seconds),

.minutes(minutes),

.hours(hours)

    );


    //Generating the Clock with `1 Hz frequency

    initial Clk_1sec = 0;

    always #50000000 Clk_1sec = ~Clk_1sec;  //Every 0.5 sec toggle the clock.


    initial begin

        reset = 1;

        // Wait 100 ns for global reset to finish

        #100;

        reset = 0;

    end

endmodule
```