University of Tehran
College of Engineering
School of Electrical & Computer Engineering

Experiment 1
Sessions 1, 2, 3

Clock and Periodic Signal Generation

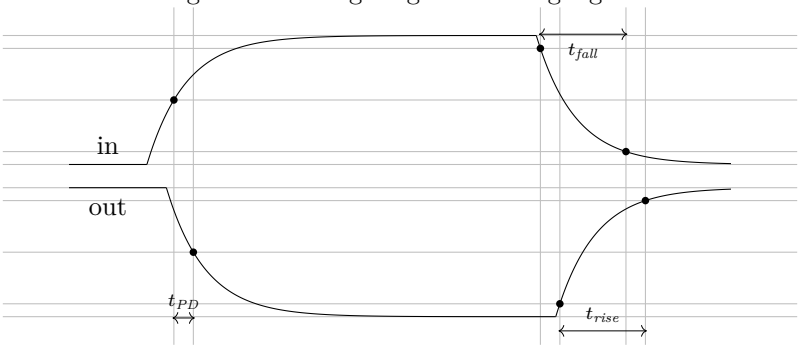
Digital Logic Laboratory
ECE 045
Laboratory Manual

Spring 1399

Contents

Prepared and developed by Katayoon Basharkhah under supervision of Professor Z. Navabi

Figure 1: Timing diagram of a logic gate



Introduction

The goal of this experiment is to introduce the concepts of static characteristics of digital logic gates, delay times, clock frequency generation and digital system using schematic diagram and *Verilog* HDL. This lab is organized in two separate segments. The first segment emphasizes on the analog implementation of different oscillator circuits using LTspice simulator.In the second part you will implement the same circuits in Verilog HDL using Quartus. Finally you will get into FPGA implementation using Altera-Modelsim simulation tool.

By the end of this experiment, you should have learned:

- Power supply, Function Generator, and Oscilloscope
- CD 4000 Series Basic Logic Gates
- Different oscillator circuits (a LM555 timer IC, Schmitt trigger Oscillator)
- Cyclone IV FPGA Devices

1 Clock Generation using ICs and Analog components

In this part, you will understand different methods of clock generation in digital systems. You will use LTspice simulation tool for implementing the circuits of this section. Also a separate demo on running examples in LTspice will be provided for you.

1.1 Ring Oscillator

One of the most important parameters in digital logic gates is the propagation delay, that is defined as the time from the 50% point of input to the 50% point of output, as shown in figure 1. There are also two more parameters named t_{fall} and t_{rise} that are measured as the time between 10% and 90% point of the signal and vice versa.

The delay of logic gates is very small and this imposes a large bandwidth. So, measuring this delay directly using a relatively low-cost oscilloscope may be difficult. An alternative method for measuring this parameter is using a *ring oscillator* as shown in figure 2. A ring oscillator is

Prepared and developed by Katayoon Basharkhah under supervision of Professor Z. Navabi

Figure 2: Ring oscillator

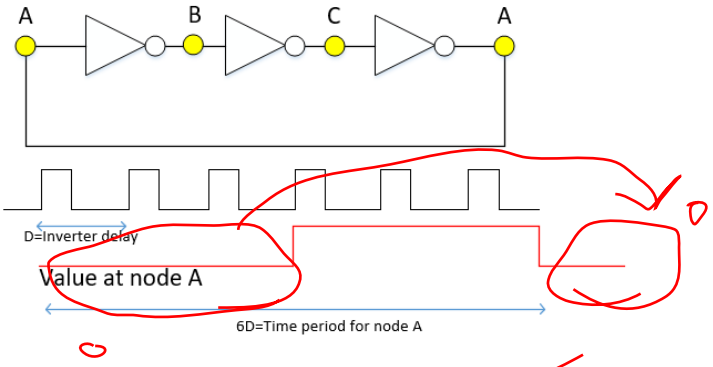
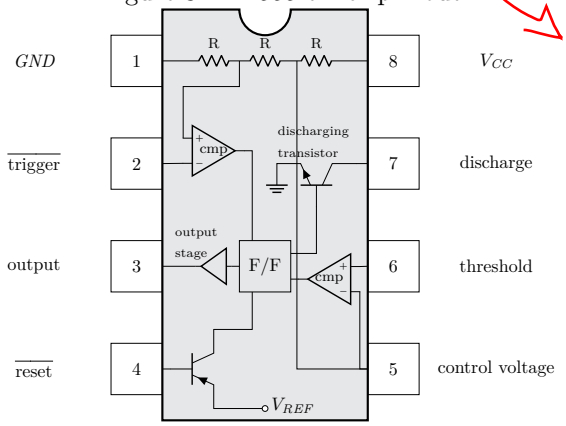


Figure 3: LM555 timer pin-out



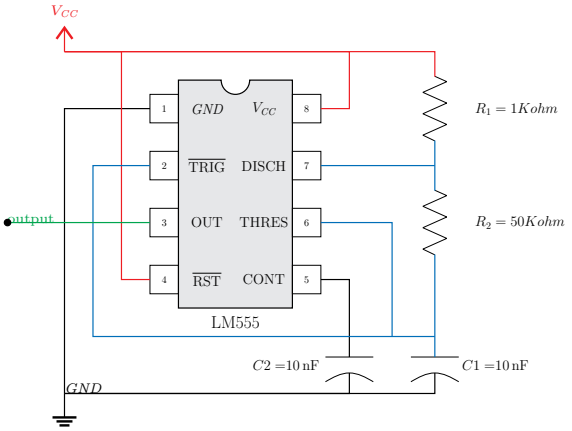
composed of a chain of odd number of inverters, in which output of last inverter is connected to the input of first one. The time at which a value feeds back to the same node is the time period of the ring oscillator which equals $2N * Delay_{inv}$, where N is the odd number and $Delay_{inv}$ is the delay of each inverter gate. The delay of each single inverter can be determined by measuring the total delay. Simulate the circuit shown in figure 2 using inverter gate in CD4000 series (CD4049B).

1. Measure the propagation delay of the chain by measuring the period time of the output.
2. Measure the delay of a single inverter and compare it with the delay in 7404 TTL specification (use from 7404 TTL datasheet).

1.2 LM555 timer

LM555 is among the devices can be used for generating clock signal or time delays. The pin layout of this IC can be seen in figure 3.

Figure 4: LM555 in astable mode



This IC operates in three modes: Monostable, Bistable and Astable. The astable mode that we use in this experiment allows the timer to operate as an oscillator that outputs a continuous rectangular pulse of a required frequency. For astable operation, we need two resistors and one capacitor to design a circuit that operates at the frequency required. The timing during which the output is either high or low is determined by these externally connected resistors and capacitors. Note that the durations of the low and high states may be different. Figure 4 illustrates an LM555 configuration for astable mode operation.

The external capacitor $C1$ charges through $R_1 + R_2$ and discharges through R_2. Thus, the duty cycle and frequency may be precisely set by selecting the right combination of resistances and capacitance. According to figure 3 and figure 4, the charge time (output high) is given by $T_1 = 0.693 * (R_1 + R_2) * C$ and the discharge time (output low) by $T_2 = 0.693 * R_2 * C$. Thus, the total time period of square wave is $T = T_1 + T_2 = 0.693 * (R_1 + 2R_2) * C$. Consequently, the frequency of oscillation is $\frac{1}{T}$. The duty cycle also can be computed by $\frac{R_1+R_2}{R_1+2R_2}$.

These equations describe how we can choose these three values to decide on the frequency and the high and low duration of our signal. With the LM555 timer, the default value of R_1 in this configuration is $1\,k\Omega$ and this means that we cannot get a perfect 50% duty cycle. (If we make $R_2 \gg R_1$ then we can get close.)

Do the following work. Your report must include the procedure you followed, as well as any observation and results.

1. Implement the LM555 in astable mode using the wiring diagram from figure 4 and observe the output. Report the clock frequency and the duty cycle and include the waveform of the output in your report.

2. Change the value of R_2 resistors to produce different clock frequencies. To do so, R_2 should be $1\,k\Omega$, $10\,k\Omega$ and $100\,k\Omega$. For this resistor, use a $50\,k\Omega$ variable resistor. Calculate the frequency and duty cycle using above equations and compare them to the clock signal you see on the output.

1.3 Schmitt Trigger Oscillator

You can see the realization of the Schmitt inverter oscillator in figure 5. In the Schmitt inverter oscillator, $f = \frac{\alpha}{RC}$, where α is a constant. Study the principle of this circuit and be prepared for it. For schmitt inverter oscillator circuit use CD40106B.

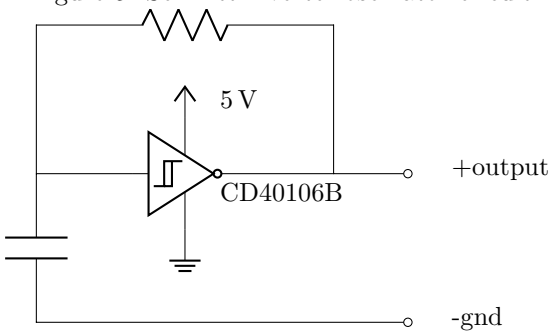
1. Considering the given equation, try the circuit with different values for the resistor and the capacitor and observe the changes. Use 470 Ω, 1 kΩ and 2.2 kΩ for the resistor and 10 nF for the capacitor.
2. Find α parameter.

2 Clock Generation using Verilog HDL

In the previous section, you have learned the functionality of different clock generation systems by simulating them in LTspice. In this section, you will see how you can implement the same circuits in Verilog. There are two ways for implementing such clock circuits in Verilog. You can simply use Verilog delay expressions, for setting the ON and OFF duration of the clock pulse. This kind of modeling cannot be synthesized in the synthesis tool but can be simulated in Modelsim which is proper enough for debugging your design. In the second way, you can model the circuit behaviorally using RTL hardware components. This would be a synthesizable model but with lower precision compared to the real hardware. To observe these facts a Verilog code, `LM555_timer.v` for LM555 timer circuit is provided to you. Follow the instructions below:

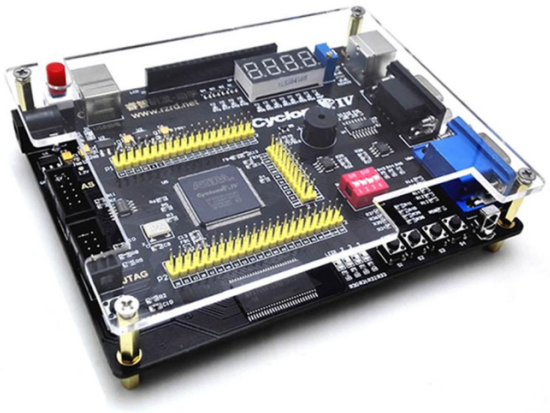
1. Before simulating the design, first read the code completely and understand whats going on.
2. Draw a block diagram of the design in your report and explain the hardware and the functionality.
3. To verify this design, write a simple testbench. Make an instance of the design in the testbench and test the design by providing the parameters.

Figure 5: Schmitt inverter oscillator circuit



Prepared and developed by Katayoon Basharkhah under supervision of Professor Z. Navabi

Figure 6: Altera cyclone IV board



4. Measure the duty cycle of this circuit by setting resistor values of section 1.2 and include the waveforms of the inputs and outputs in your report.Are the results of this two sections completely matched? Explain the reason if not.

5. Write the design with the alternative method, using delay statements and run the results in Modelsim. Show the waveforms and duty cycles for this method.

3 FPGA Design

In this part, you will be familiar with FPGA design. First you will start with simulating different design using 74series IC. In this part you will use Intel Quatus Synthesis tool for mapping your designs on Altera CycloneIV FPGA devices and Modelsim simulation tool for verifying your designs. The FPGA device that you will use in this and all other experiments until the end of the semester is *Cyclone IV* device. Figure 6 shows the layout of the board. In this section you use 74 series ICs. These ICs are placed in Quartus tool inside the Altera MAXPLUS2 libraries. In order to access these libraries and for simplicity, you will design the circuits in a block diagram format. Appendix A shows a guideline for building a schematic design project in Quartus software.

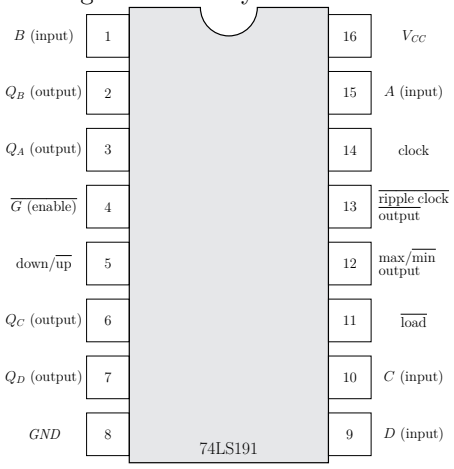
In this part you will use the clock circuits of the first section along with FPGA to simulate a clock frequency divider. There are many parts which should be considered in designing the desired system. Below is a short description of these parts and what you should follow for this experiment.

3.1 Ring Oscillator

You have seen the functionality of a ring oscillator by simulating the circuit in LTspice. Now you will redo the simulation using Verilog HDL code.

1. Write a Verilog description of the ring oscillator of section 1.1.For this purpose you need to use an odd number of inverters. Your design must be parameterized so that it receives the

Figure 7: Pin layout of 74191



delay value of the inverters and the number of inverters as parameters and generates the corresponding oscillator as the output.

2. To verify your design, you need a simple testbench. Make an instance of the ring oscillator in your testbench and test the design by providing the parameters. For real modeling of the ring oscillator, in your testbench use the delay values that you calculated in section one.

3. Measure the ring oscillator frequency and include the waveforms of the inputs and outputs in your report.

3.2 Synchronous Counter as a Frequency Divider

Different clock signals can be produced by the aforementioned methods, but not all of them are suitable for all applications. Consider a 1 Hz clock signal which can be easily produced by a LM555 timer. The timing error of this signal can be 1-2%, which is too much for a low frequency like this, while this error range is acceptable for higher frequencies. So, a higher frequency can be chosen and then a frequency divider can reduce the frequency to the desired one.

Counters can be used as a frequency divider. 74LS191 is a synchronous 4-bit up/down counter. As figure 7 shows, it has 4 inputs D-A(most to least), 4 outputs Q_D-Q_A (most to least)and a parallel load, which allows for presetting an initial value for the counter. With this pin layout two counters can be cascaded when the modulus is more than 4 bits.

The timing diagram of 74191 is shown in figure 8. When up counting is desired the initial value is obtained by:

$$\textit{Initial value} = \textit{Maximum value} - \textit{Modulus}$$

Construct a divide by 113 synchronous up-counter as shown in figure 9. You should:

1. Use the MAXPLUS2 74 series (74191) IC in your block diagram and connect them based on figure 9 and layout offigure 7.

Prepared and developed by Katayoon Basharkhah under supervision of Professor Z. Navabi

Figure 8: Timing diagram of 74191

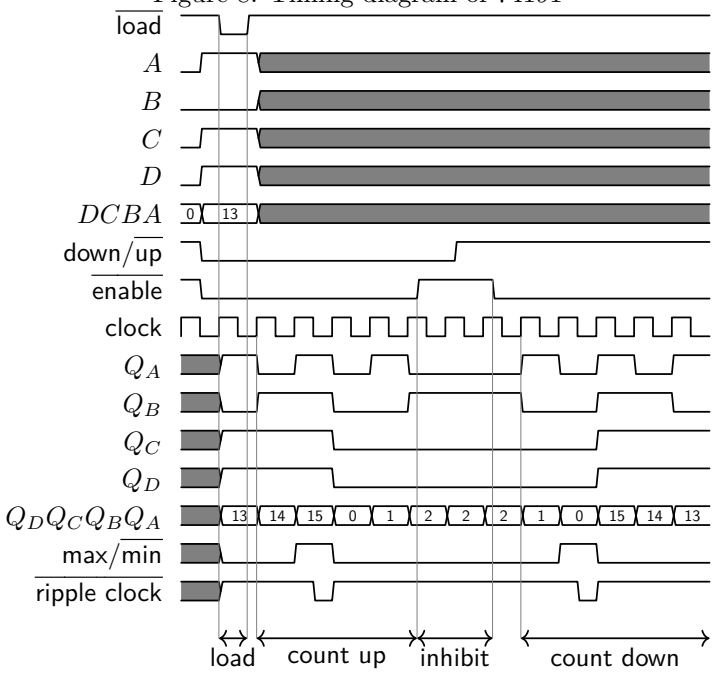


Figure 9: Frequency divider using 74191

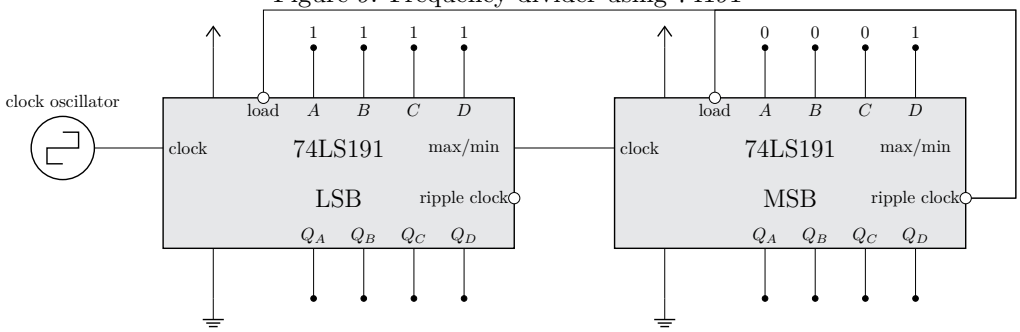
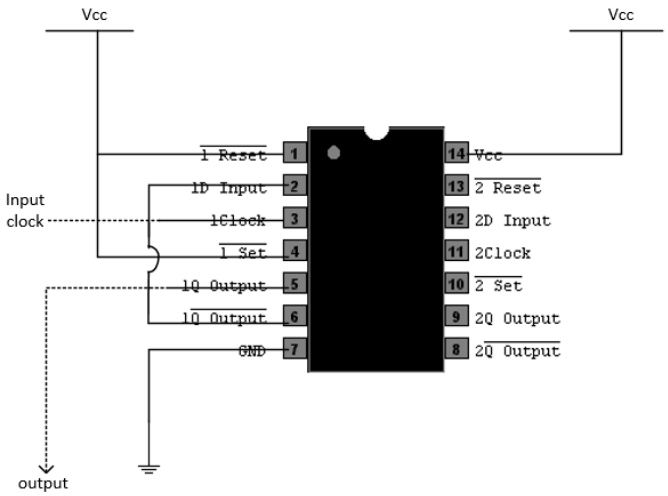


Figure 10: 7474 as a T Flip-Flop



2. A Presetting mechanism is necessary for initial loading of the counters of figure 9. The mechanism as you have learned in the logic design course can include an AND gate for anding a preset input signal with the load inputs of the counter. Perform this presetting by using 7408 AND gate of MAXPLUS2 74 series.

3. Write a testbench for the design and set the required input values like the initial load value, enable and up/count signal. for counting up the DNU signal must be set to zero.

4. Use the ring oscillator of the previous part as the clock input of the LSB counter. Since the ring oscillator model is not a synthesizable design, connect the ring oscillator inside the testebench and not in the Quartus block diagram.

5. After simulating the design in Modelsim software, record the results of carry out of the max/min signal of the MSB counter and measure its frequency, compare the results with the frequency of the input clock.

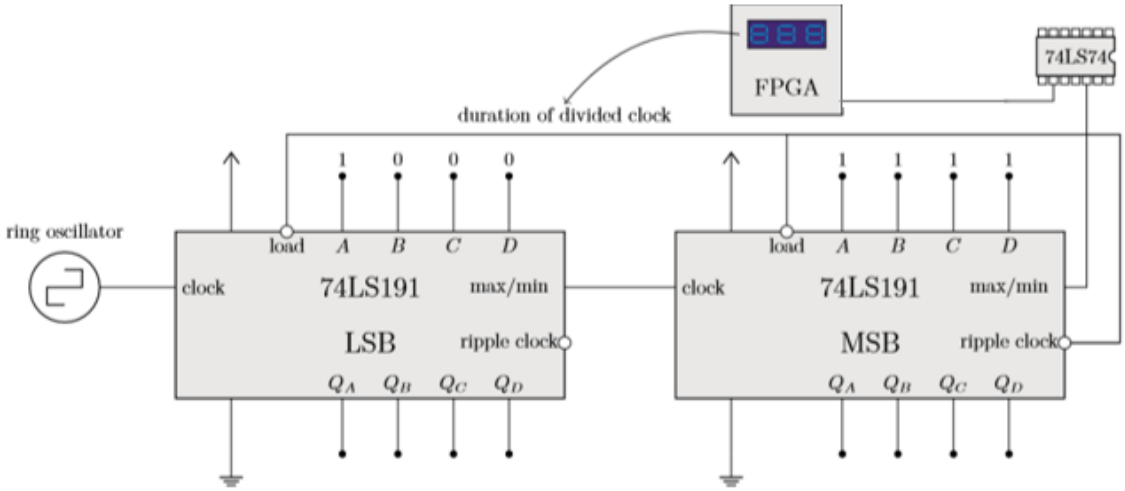
3.3 T Flip-Flop

You should use a T Flip-Flop after counter to produce a 50 percent duty cycle signal. Use 7474 IC from the MAXPLUS2 library for this purpose. 7474 is a dual D Flip-Flop that can be converted to a T Flip-Flop if it is used as in figure 10.

3.4 Display module in FPGA

As the output of ring oscillator is a high frequency signal, it may be difficult to observe it on an analog oscilloscope. Alternatively, you can use a FPGA and calculate the frequency of ring oscillator. To do this, the output of the counter, after passing a T flip-flop, will be connected as the

Figure 11: Clock adjusting system



input of the FPGA. The verilog code for the display module is provided to you (display.v). This module counts the number of clock cycles that the input signal is on which is called *duration*. The duration indicates the divider divisor value.

1. Add the Verilog code `display.v` to your project.
2. Set this module as the top-level entity and synthesize it.
3. Generate a schematic symbol for this module.
4. Set back the Clock adjusting block diagram as the top-level entity.
5. Now connect these circuits together according to figure 11. Connect the display module block diagram to the proper pins of the counter.
6. Write a testbench for this design and set the required inputs and outputs and test your design in *ModelSim* and check if it is working correctly.
7. Calculate the ring oscillator frequency based on the results you have recorded. You should calculate the frequency of the divided signal based on the number you observed on the duration signal and then calculate the frequency of ring oscillator.

Appendices

Appendix A Using Quartus II

A.1 Create the Project

1. Click on *File* ▷ *New Project Wizard*
2. Create an appropriate directory for your project and complete the form
3. Select the FPGA device as *Cyclone EP2C20F484C7* and then click *Finish*
4. From *File* ▷ *New* select the *Veriog HDL File*
5. Write your Verilog code in this window

A.2 Compilation

- Select *Processing* ▷ *Start Compilation*
- Click on quick shortcut ▶

There may be lots of warnings and some errors after compilation. The warnings are not so important while the errors should be completely removed.

A.3 Pin Assignment

After you successfully compile your design, you should set your design with physical pins of the Cyclone II FPGA on DE1 board. You can find the position and the index of each pin from the DE1 user manual.

Figure 12: Creating new project

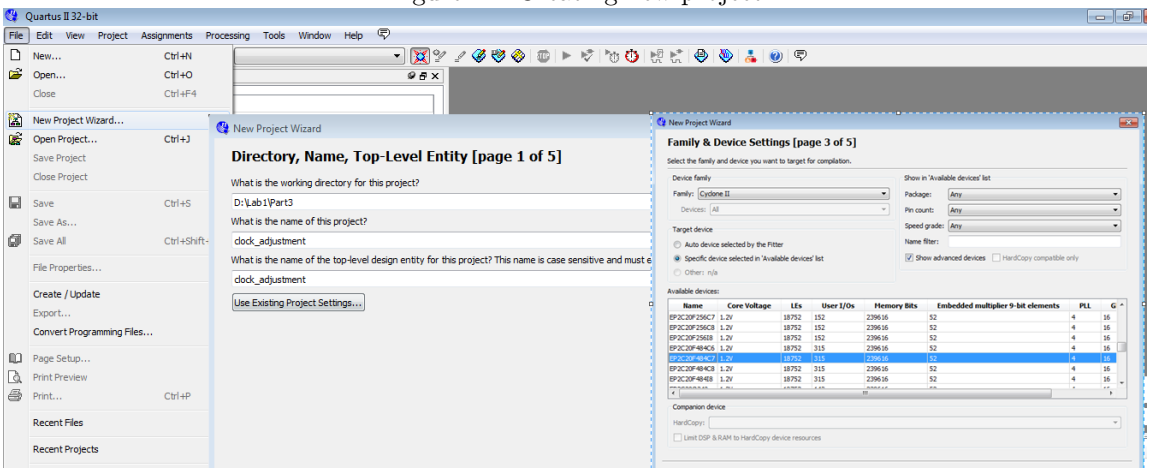
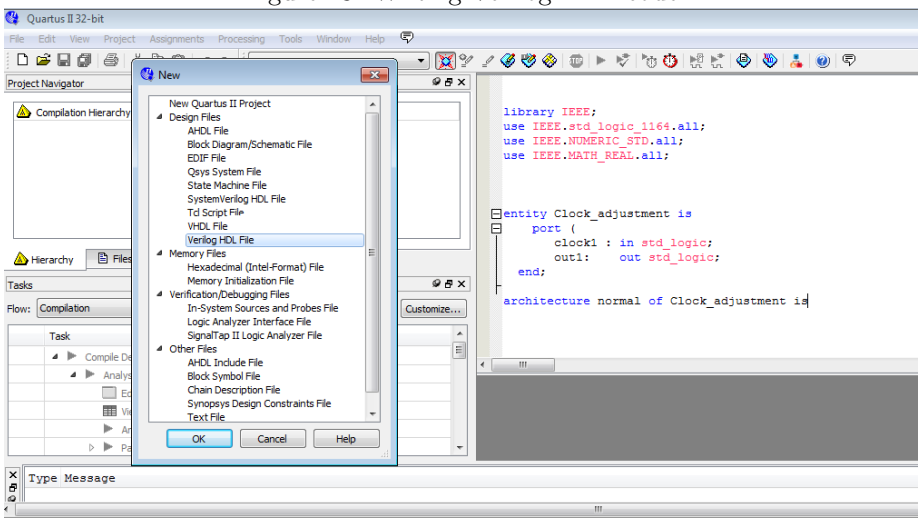


Figure 13: Writing Verilog HDL code



From *Assignments* ▷ *Pin Planner*, Pin Planner window will appear and you can select the corresponding locations from the list. When you are done with this, recompile your project.

A.4 Program the Design

1. Click on the icon on the toolbar to open *Programmer*
2. Choose *USB Blaster* from the *Hardware setup* in the new window
3. Click on the start button

Your design will be programmed on the board when it is finished.

A.5 Examine the Timing and Resources

Now you can examine the resource usage of your design from *Processing* ▷ *Compilation Report* or by clicking on quick shortcut , and the timing from the *Tools* ▷ *TimeQuest Timing Analyzer*.

Acknowledgment

This lab manual was prepared and developed by Katayoon Basharkhah, PHD student of Digital Systems at University of Tehran, under the supervision of professor Zain Navabi.

This manual has been revised and edited by Hadi Safari, undergraduate student of Computer Engineering at University of Tehran and Pedram Anbaz, undergraduate student of Digital Systems at University of Tehran .

Figure 14: Pin Planner window