LABORATORY MANUAL

Experiment 1
Clock and periodic signal generation

UNIVERSITY OF TEHRAN

School of Electrical and Computer Engineering

Digital Logic Laboratory
ECE 045

Fall 1396

EXPERIMENT 1 (Sessions 1,2,3)

Introduction to Clock generation methods
FPGA Educational Boards and Altera Quartus II

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.

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

- Power supply, Function Generator, and Oscilloscope
- TTL 74 Series Basic Logic Gates
- Different oscillator circuits (a LM555 timer IC, Schmitt trigger Oscillator)
- DE1 FPGA Educational Board

PART I

In this part, you will understand and measure the static characteristic of an inverter using 74LS04 TTL gate. The pin diagram of 74LS04 is shown in Fig1.
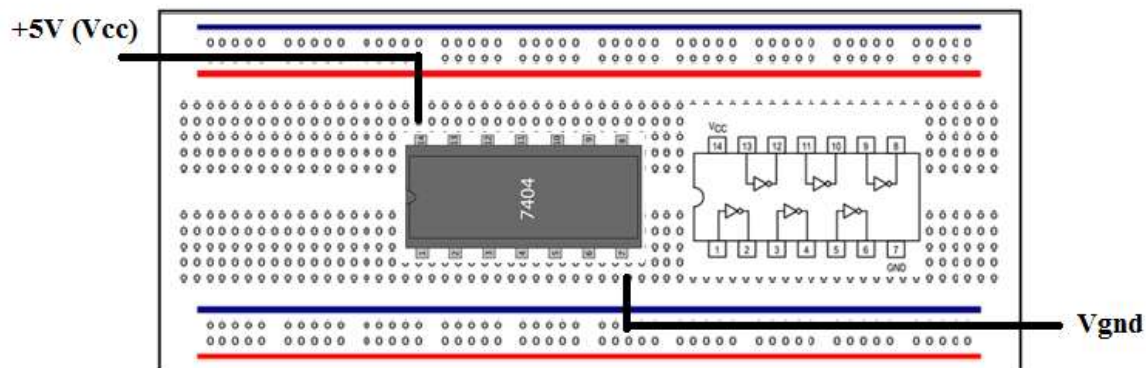

Fig 1: 74LS04 IC

1- Voltage transfer characteristic

The relationship between input and output of a logic gate is an important issue since the voltage level of a logic gate in a real world depends on the logic family resulting in values other than "1" and "0". The voltage transfer characteristic is a representation of output voltage variation when the input voltage changes continuously as in Fig 2.
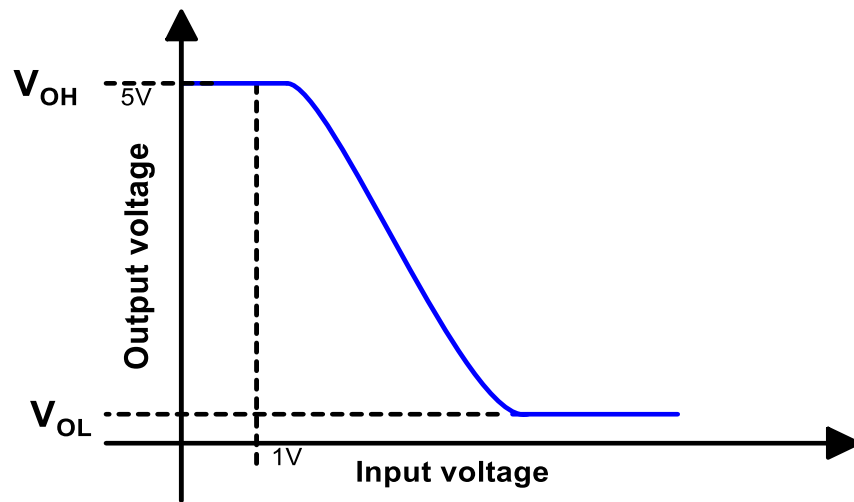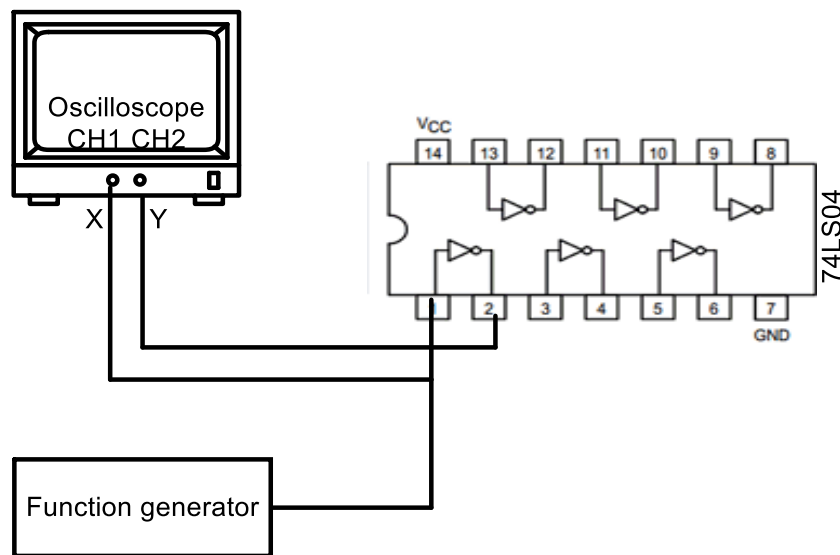
Fig 2: Voltage transfer characteristic

Fig 3: VTC determination set up

Construct the circuit shown in Fig3. The oscilloscope must be set in X-Y mode in order to observe output variation based on the input. Hence PIN1 and PIN2 are connected to X and Y channels, respectively. Adjust the function generator for a triangular input waveform which varies from 0 to 5V when the input goes from 0 to +5V, the output changes from V_H to V_L and then back. Report the V_{OH}, V_{OL}, V_{IL} and V_{IH} values in a table and answer these question in your report:

1- Why are the values of V_{OH}, V_{OL}, V_{IL} and V_{IH} important?
2- Find the nominal value of 'V_{OH}' and 'V_{OL}' by referring to datasheet.
3- Explain the hysteresis observed in VTC diagram

PART II

In this part, you will understand different methods of clock generation in digital systems.

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 Fig 4. 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.
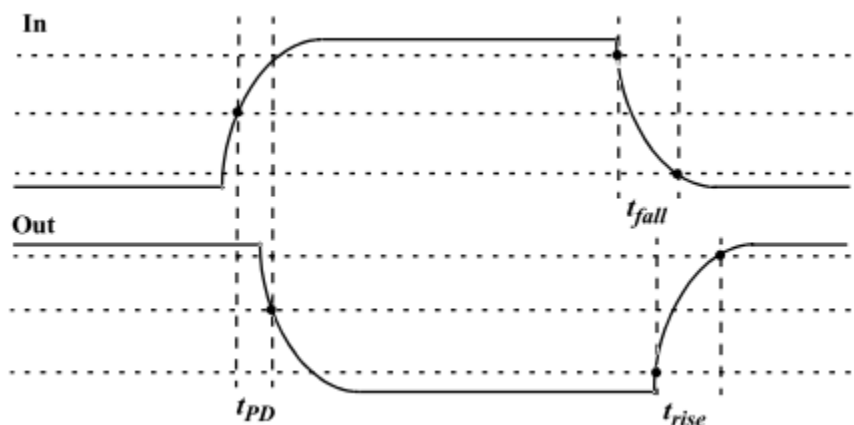
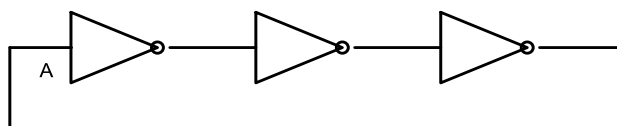**Fig 4:** Timing diagram of a logic gate

Fig 5: Ring oscillator

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 Fig 5. A ring oscillator is composed of an odd number of inverters. The 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 for each inverter gate. The delay of each single inverter can be determined by measuring the total delay.

Construct the circuit shown in Fig 6 with as little wire as possible. You should wire up three of six inverters to a chain.
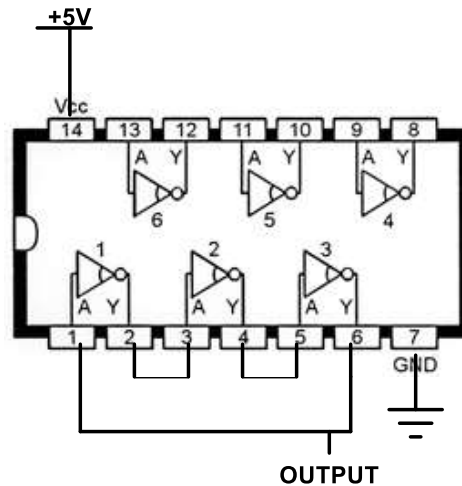
Fig 6: Ring oscillator using 74LS04

- Measure the propagation delay of the chain by measuring the period time of the output.
- Measure the delay of a single inverter and compare it with the delay in 7404 TTL specification. You can also use a 74HC14 IC for this part.

2- LM555 timer

LM555 is among the devices for generating clock signal or time delays. The pin layout of this IC can be seen in Fig 7.
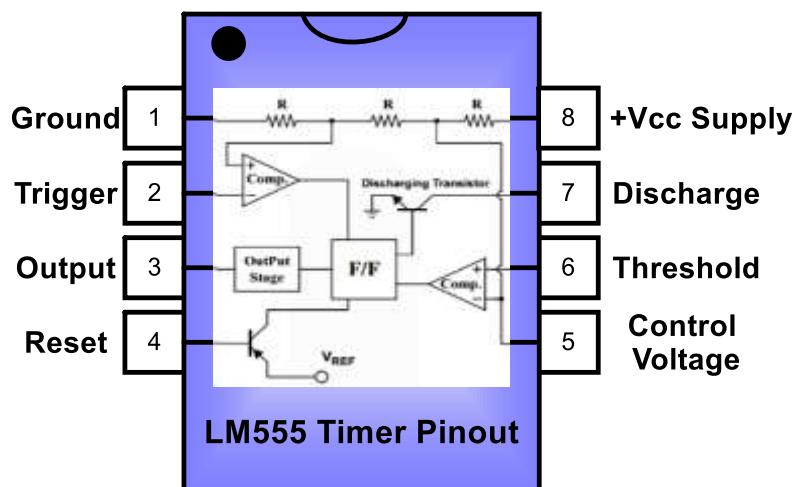

Fig 7: LM555 timer pinout

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 capacitor. Note that the durations of the low and high states may be different. Fig. 8 illustrates an LM555 configuration for astable mode operation.
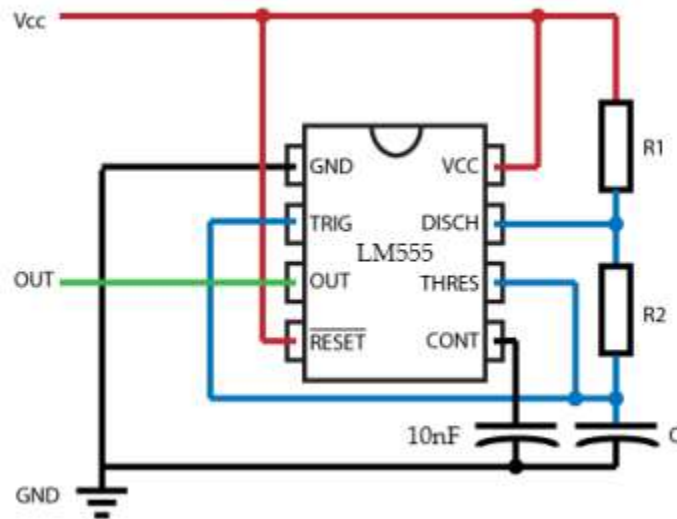

Fig 8: LM555 in astable mode

The external capacitor (C) charges through R1 + R2 and discharges through R2. Thus, the duty cycle and frequency may be precisely set by selecting the right combination of resistances and capacitance. According to Fig 7 and Fig 8, the charge time (output high) is given by T1 =0.693 × (R1+ R2) × C. The discharge time (output low) by: T2 = 0.693 × R2 × C. Thus, the total time period of square wave is T = T1 + T2 = 0.693 × (R1 +2R2) × C. Consequently, the frequency of oscillation is = 1/T. The duty cycle also can be computed by $D = \dfrac{R1+R2}{R1+2R2}$.

These equations describe how we can choose these three values to decide on the frequency and the high and low durations of our signal. With the LM555 timer, the default value of R1 in this configuration is 1kΩ and this means that we cannot get a perfect 50% duty cycle. (If we make R2 >> R1 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 Fig 8 and observe the output on oscilloscope. Measure the clock frequency and the duty cycle.
2. Change the value of R2 resistors to produce different clock frequencies. To do so, R_2 should be 1kΩ, 10KΩ, and 100KΩ. Calculate the frequency and duty cycle using above equations and compare them to the clock signal you see on the oscilloscope.

3- Schmitt inverter oscillator

The "Schmitt inverter oscillator". Below you can see the realization of the Schmitt inverter oscillator.
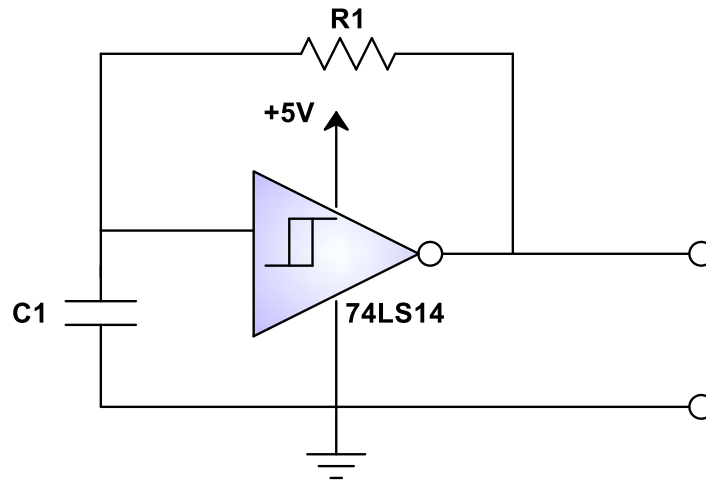

Fig 9-Schmitt inverter oscillator circuit

$$f = \frac{\alpha}{RC}$$

where α is a constant.

• Considering the given equation, try the circuit with different values for the resistor and the capacitor and observe the changes. Use R1 = 470Ω, 1k Ω, 2k Ω, and C = 10nF.

• Find α parameter.

4- 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 y 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 Fig10 shows, it has 4 input A-D and outputs Q_A-Q_D and a parallel load which allow for presetting an initial value for the counter. With this pin layout two counter can be cascaded when the modulus is more than 4 bits. The timing diagram of 74LS191 is shown in Fig11. When up counting is desired the initial value is obtained by:

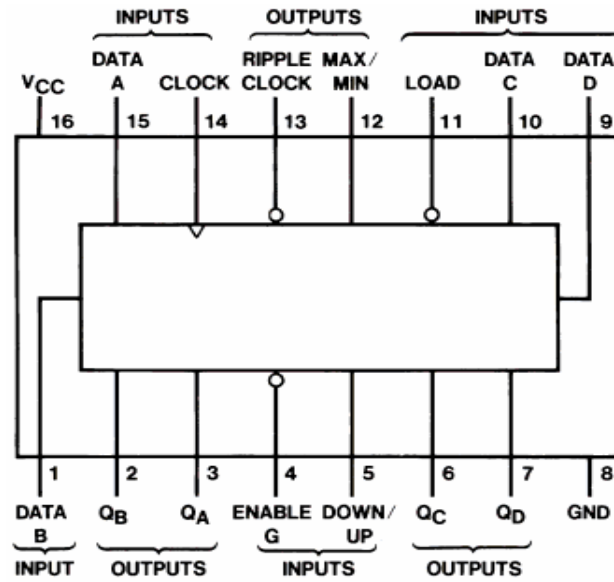Initial value= Maximum value - Modulus
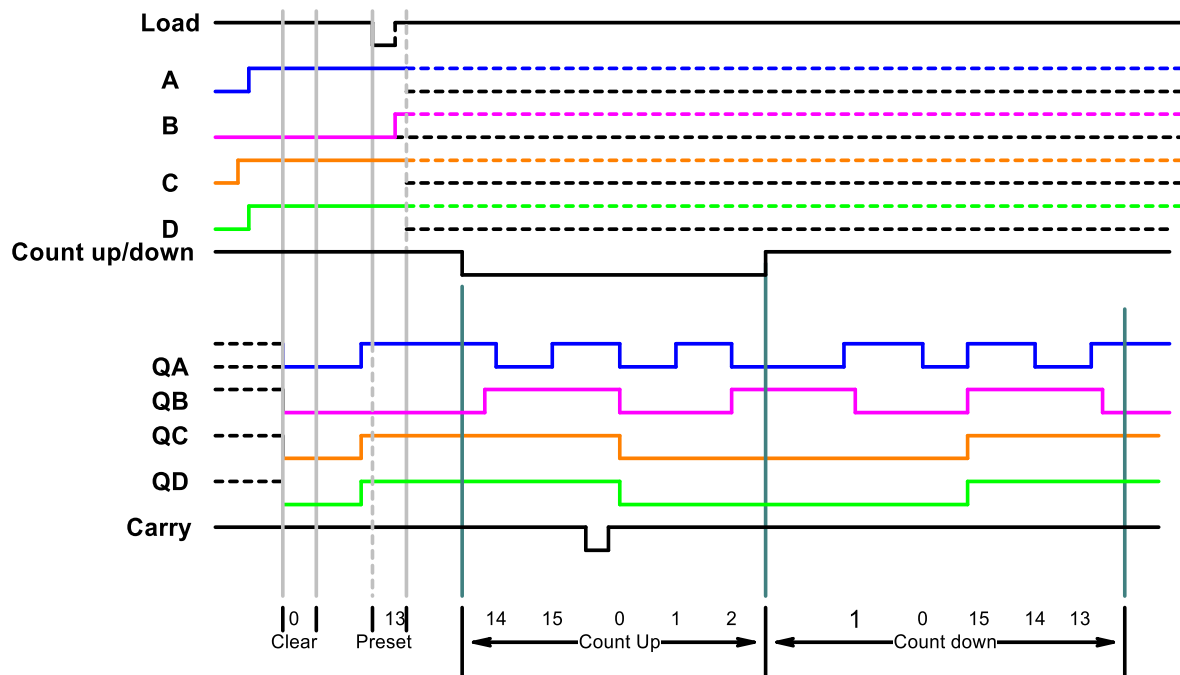
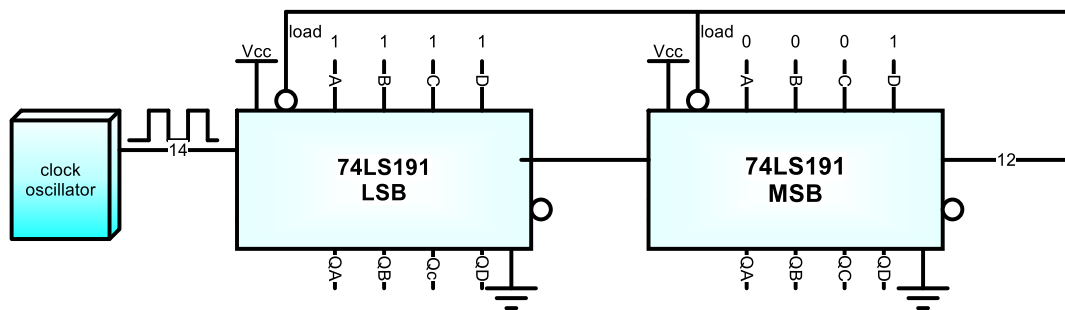**Fig 10:** Pin layout of 74LS191

Fig 11: Timing diagram of 74LS191

Fig

Fig 12. Frequency divider using 74LS193

Construct a divide by 113 synchronous counter as shown in Fig12. You should use the

- Use the ring oscillator to generate a 20 MHz clock signal.
- Connect the generated clock to the count up pin of counter.
- Preset the counter at the initial value.
- Record the results of carry out of the MSB counter and measure its frequency, compare the results with the input.
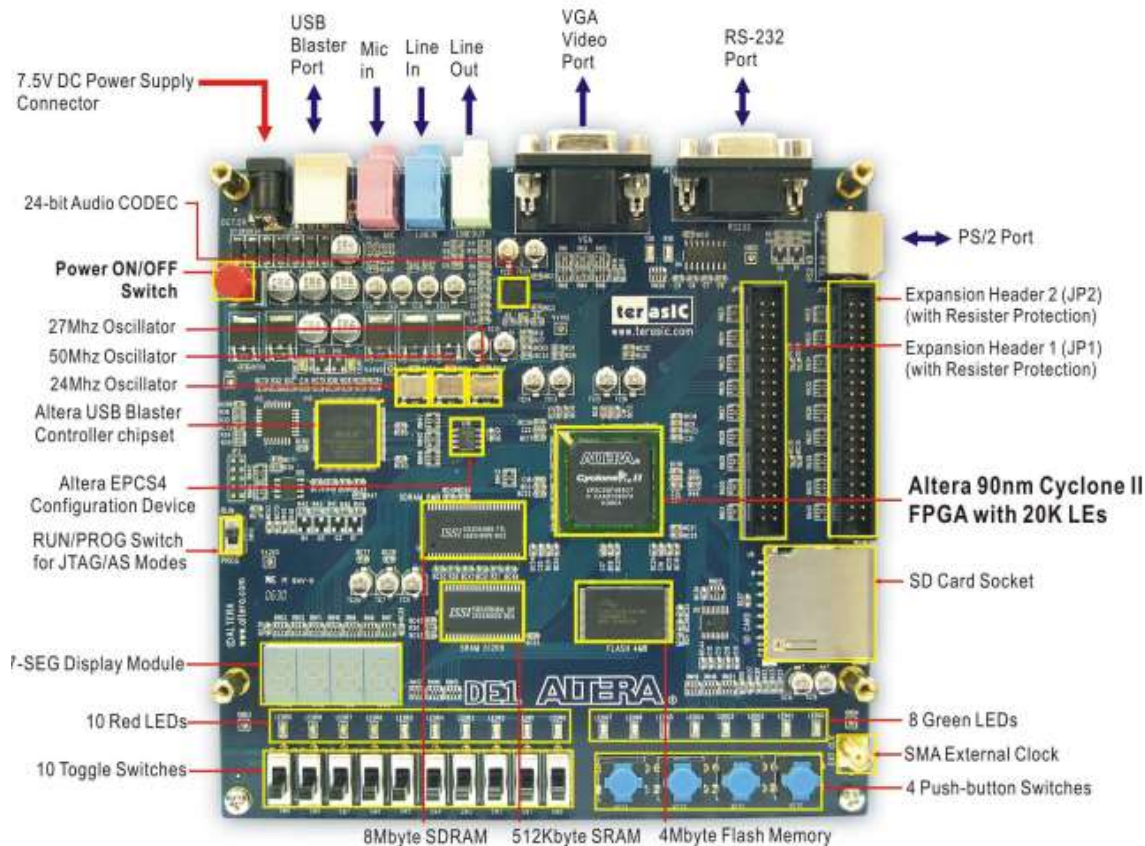
PART III

Fig 13. DE1 board

In this part, you will be familiar with FPGA design. You will learn how to run and program a Verilog code on a FPGA board. The FPGA board that you will use in this and all other experiments until the end of the semester is Altera DE1 board. Fig 13 shows the layout of the board and indicates the location of the connectors and key components.

In this part you should use a FPGA to observe and calculate the frequency of a ring oscillator. You will use the circuits of previous sections along with FPGA as shown in Fig15. There are many parts which should be considered in designing such system. Below is a short description of these parts and what you should follow for this experiment.

1- Ring oscillator
Use the ring oscillator of part II for this part. Use 3 of 6 inverters of 74LS14 IC to generate a 20 MHz clock frequency. You can also use other types of inverters, like 74LS04 to produce a high frequency clock. But pay attention that the inverter delays are different, refer to the datasheet of each inverter.

2- Counter

Use the counter of Part II. Design a divide by 50 counter. Connect the output of ring oscillator to the input of counter and observe the results. Check the output frequency of the counter if it is correctly divided by 50. It should be 400 kHz.

3- Voltage level converter

When the term logical 0 (GND) is used, it means almost 0 voltage but logical 1 can mean different values for the VCC voltage (5v, 3.3v, 2.8v, etc.). If all devices used in the circuit have the same voltage as logical 1 it would work correctly, but sometimes there are devices in the same circuit which expect different VCCs. In such situations level shifters are used, this device has two sides, one dedicated to high voltage VCC (HV) and one side to low voltage VCC (LV). In "level converter" both sides have GND and VCC pins, GNDs are connected to the shared ground, HV to the higher VCC, and LV to the lower VCC. There are also four I/O pins on the HV side, and four on the LV side. Any of this pins can be driven by logical 1s and 0s, and the corresponding pin would change. For example, in this circuit that FPGA uses 3.3v as VCC and ICs have 5v VCCs, if HV1 is driven by 5v, LV1 would have the value 3.3v, and if LV2 is driven by 3.3v you can see 5v on HV2 pin, and 0 volts on any pin would result 0 volt on the corresponding pin.

CAUTION: It is obvious that connected pins cannot be driven together, for example if HV4 is driven, FPGA cannot drive LV4, it should use this pin as its input.
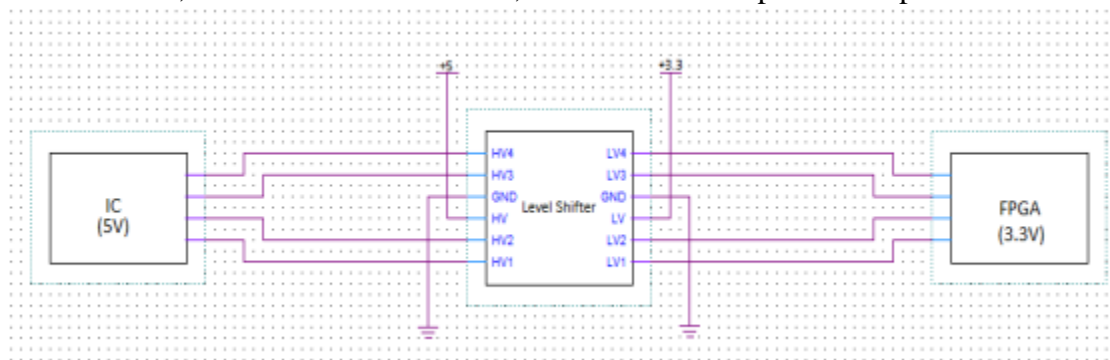

Fig 14: Level converter layout

4- 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 counter after passing through a voltage level converter, will be connected as the input of the FPGA. It uses the 50 MHz Clock_50 of the DE1 board and count the number of the input signal cycles which is called duration. The duration will be displayed on three 7 segments on the DE1 board as a BCD number. You should use this number to calculate the original frequency of ring oscillator.

- Make a project in Quartus II using the Verilog code **display.v**.
- Set the pin assignment of your design using KEY, HEX and 50 MHz Clock.

- Simulate the test bench of your design in ModelSim and check if it is working correctly.
- Program the design on DE1 board and record the result shown on 7 segments.
- 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 7 segments and then calculate the frequency of ring oscillator).

Metastability

When a signal is transferred between circuits in asynchronous clock domains, metastability can cause system failure. In order to prevent this phenomenon, asynchronous inputs to other parts of the design must be stable for a short time before clock edge and for a short time after that. Our answer to this problem in this experiment is using synchronization registers. Use a register for each input and output of the FPGA.

5- Inverter and T Flip Flop

Use an inverter to invert the output for parallel load of counters. You should also use a T-Flip Flop after counter to produce a 50 duty cycle signal.

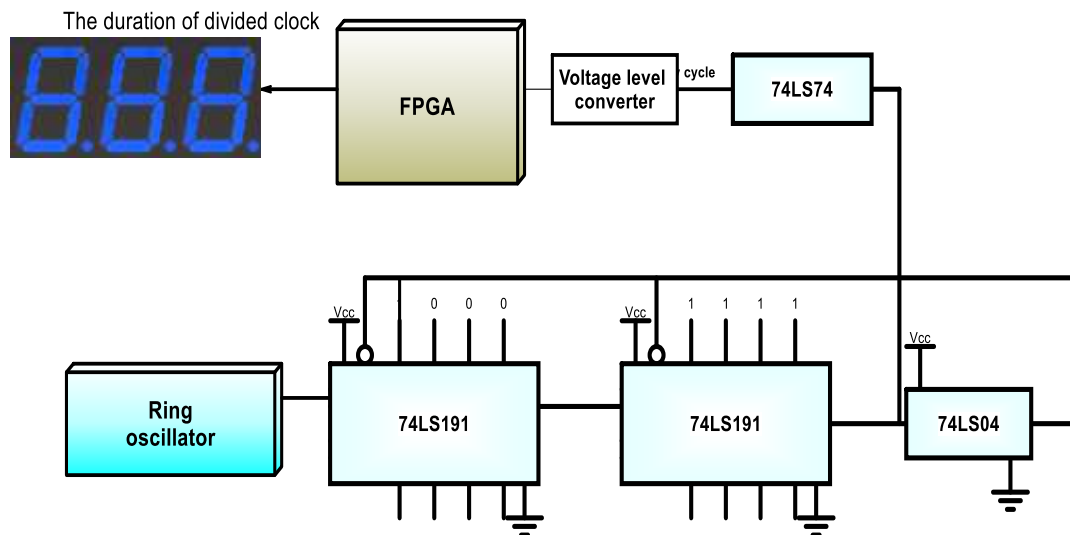Now connect these circuits together using I/O pins of FPGA. (According to Fig. 15).

Fig 15- clock adjusting system

APPENDIX
Using Quartus II

1-Create the project
- Click on **File> New project wizard**
- Create a good directory for your project and complete the form
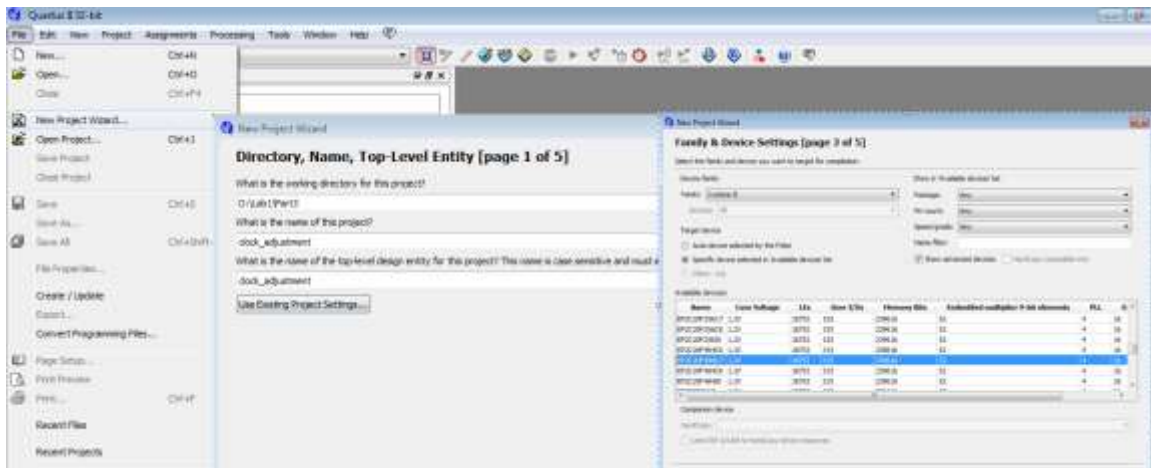- Select the FPGA device as **Cyclone EP2C20F484C7**. Then click Finish.
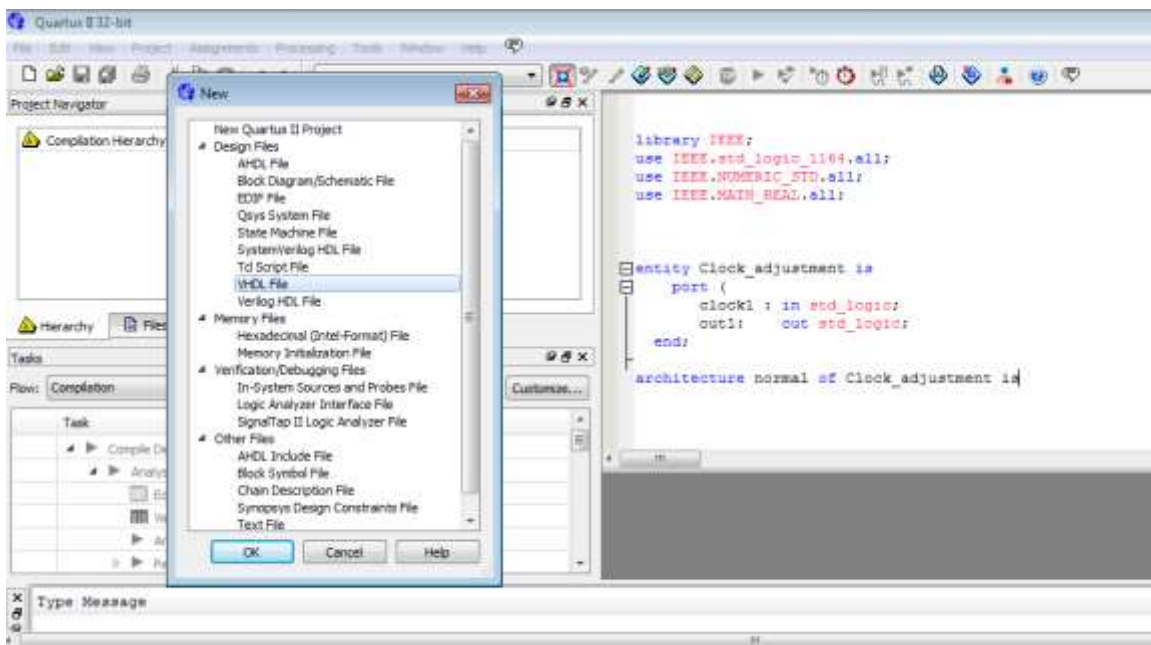

Fig 16: creating new project

Fig 17: Writing Verilog HDL code

- From **File> New** select the **Veriog HDL File**
- Write your Verilog code in this paper

2-compilation

- From **Processing> start compilation** or the quick shortcut ▶ compile your project. There may be lots of warnings and some errors after compilation. The warnings are not so important while the errors should be completely removed).

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.

Fig 18: Pin planner

From **Assignments> Pin planner**, a window like the Fig 18 will appear and you should select the corresponding location from the list. When you are done with this, recompile your project.

4-Program your design

Click on the **programmer** 🖑 icon on the Tools bar and from the Hardware setup in the new window, choose the USB Blaster. Click on the start button and your design will be programmed on the board when it is finished.

5-Examine the timing and resources

Now you can examine the resource usage of your design from compilation report 🔷 and the timing from the **Tools>TimeQuest Timing Analyzer**