

EASTERN MEDITERRANEAN UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
CMPE224 DIGITAL LOGIC SYSTEMS

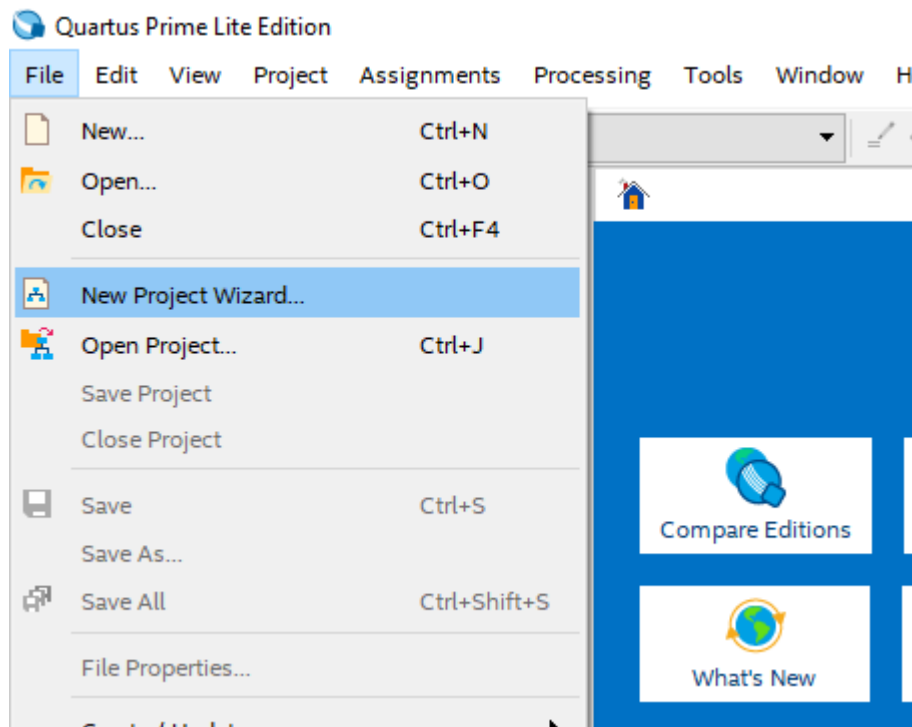
OBJECTIVES:

This laboratory work aims to teach the use of Quartus design environment for the simulation of synchronous digital logic circuits and systems. Basic steps required to open a project, drawing the circuit schematics, entering Verilog code for implementation and steps for simulation will be covered over examples.

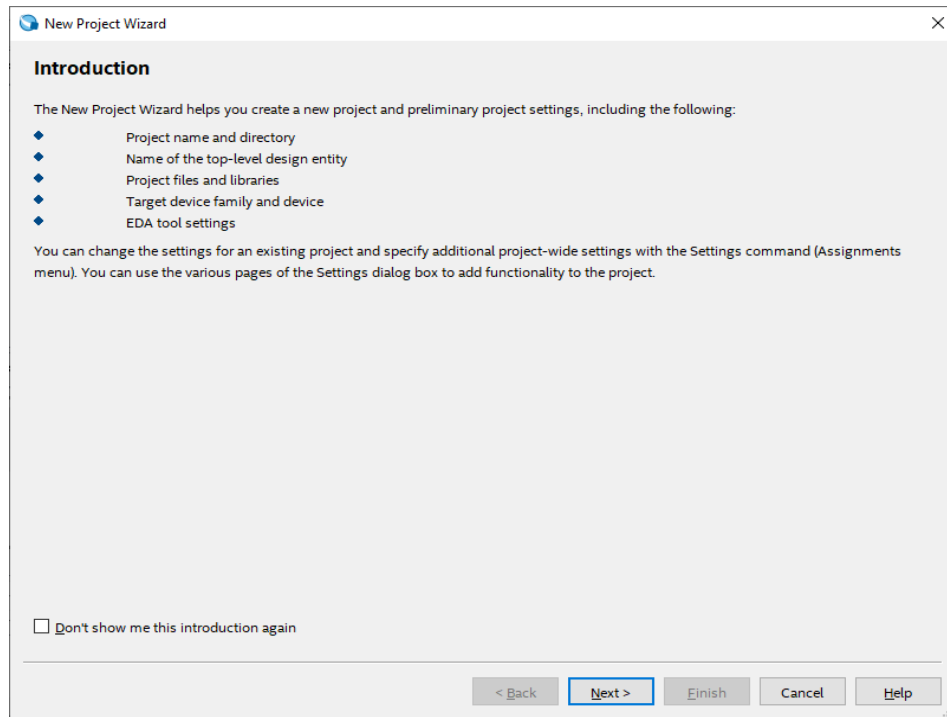
Phase 1:

Introduction to Quartus Lite Environment.

Step 1. Opening a Project: For all new designs, you are required to open a new project. For this purpose, follow : [File] -> [New Project Wizard]

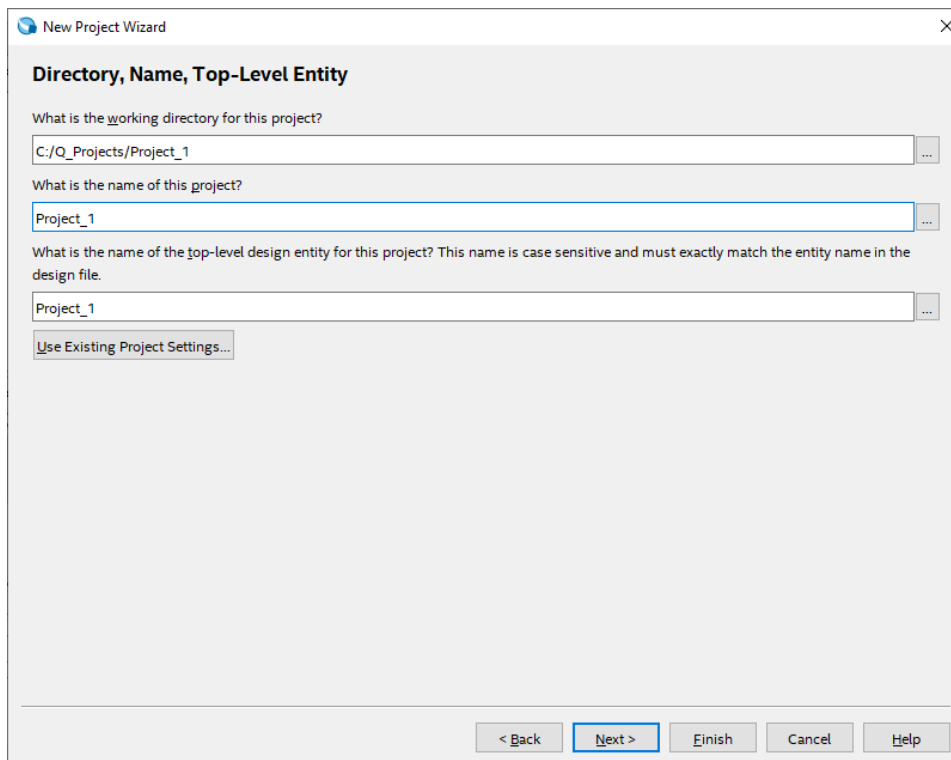


Next, you will see the following screen informing you on the steps you will follow to form your new project.

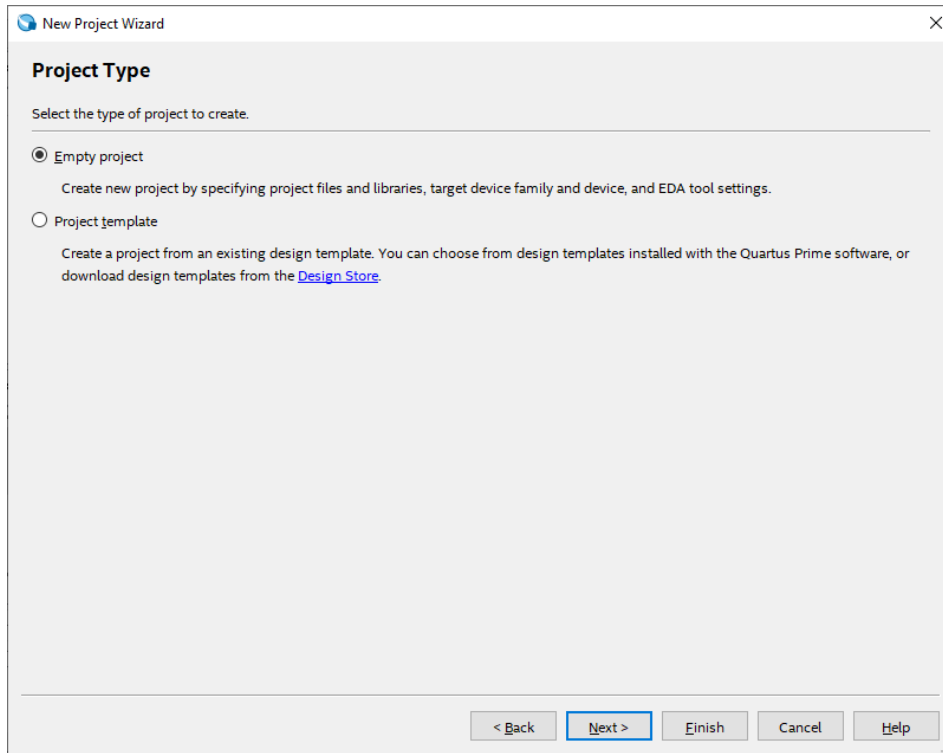


Click on [Next >] to proceed to the next screen.

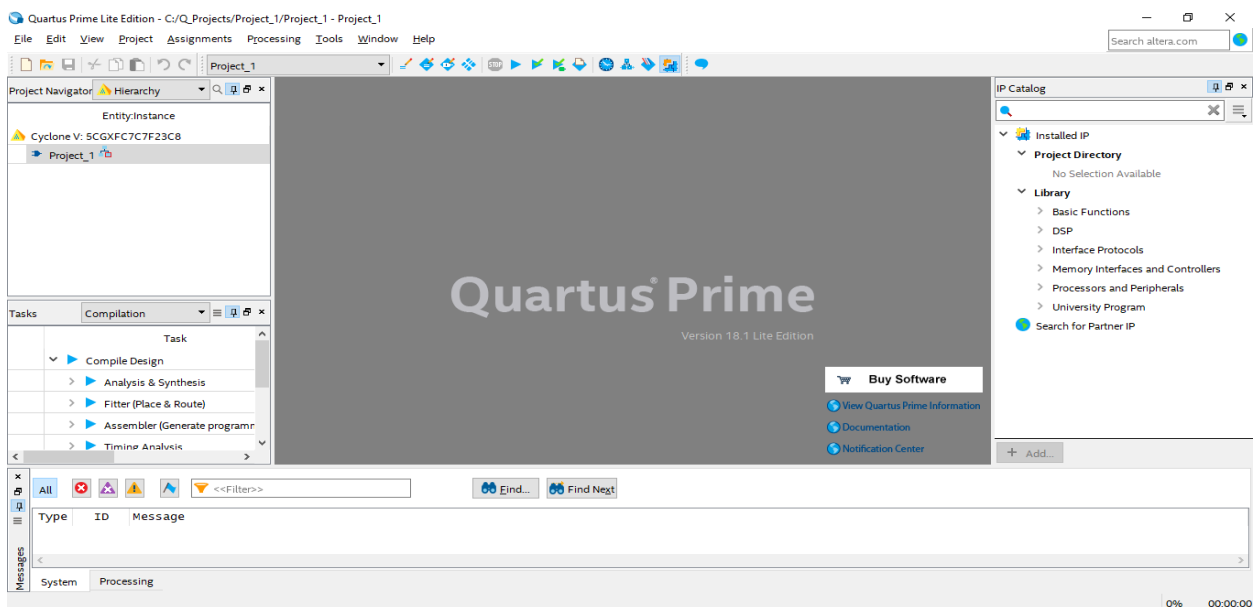
Now, select the working directory, type the name of your project and type the name of your top-level design entity (At this step, type the project name. We will return back to this issue when we start designing systems at multiple levels). Project names are case sensitive and give meaningful names to your projects so that you will recall them later easily.



In this step, you should select the type of your project. If you have just started the project, then you have an empty project. Or, if this project is a continuation of an existing project, then you need to select the second option. For the time being, select the first option and click on the Finish button.

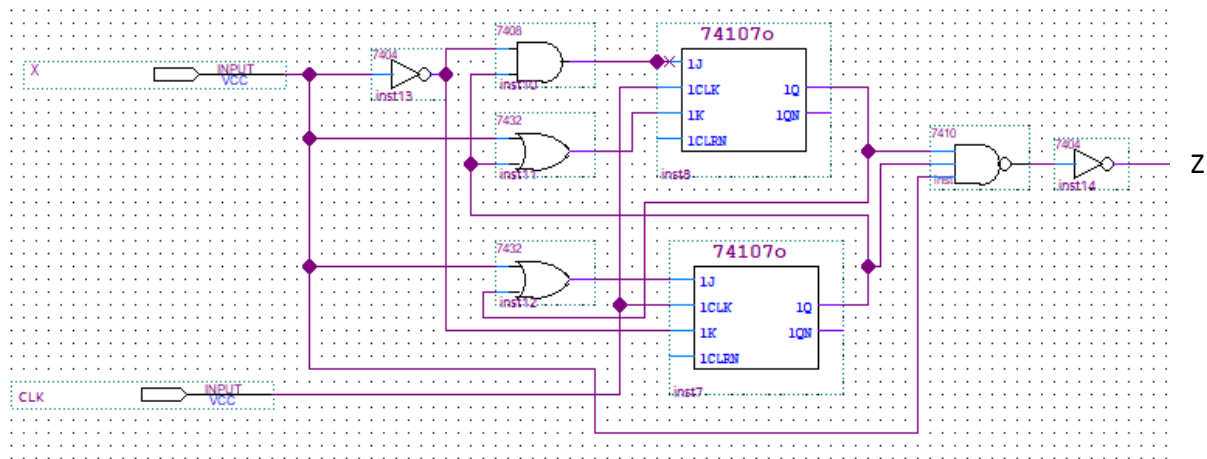


After instantiating your project, you will be faced with the following screen:



Step 2: A simple schematic design entry:

After initializing your project successfully, now you can start a simple project implementation and its simulation. You will implement the following synchronous logic circuit for this purpose. This circuit is designed to detect the sequence 1001 over a single input line X. Output Z is set to 1 when this sequence is detected, Z=0 otherwise. Overlapping of 4-bit codes is allowed.



- i. Using [File] -> [New] selections, follow the option **Block Diagram / Schematic File** from the pop-up menu and click on the [OK] button. You will get an empty design window on your screen.
- ii. On this design window, when you double-click on an empty area, the available design libraries within the environment will be presented to you on a new screen. Under the **primitives library (or from maxplus2 library)**, you will see the **Logic** (or module IDs as above) gates section. Select this library to see all the logic gates available for your design entry.
- iii. As illustrated in the example above, we need one **and gate**, two **or gates**, one **nand gate**, two JK-FFs and, one **input** and one **output** entities for the circuit implementation..

Connecting circuit elements together:

After selecting and placing the circuit elements from the library, we arrange them on the design screen using mouse or arrow buttons on the keyboard (simply select a component and move it to position you want).

When you bring your mouse cursor to an input or output pin of a component, the cursor changes to a “+” sign to start drawing a connection line between pins of two components. Click on the left mouse button and start drawing a connection line as described in the example circuit above. After seeing a small square between your connection line and the connection port, you can release mouse button to complete the connection. In case that your connection line is not completely connected to a port, there will be a “x” symbol at its end. You need to reconnect this line to the appropriate port.

After making all connectins among the circuit components, name the input and output ports by clicking the “PIN_NAME” labels on the ports. Name the ports as shown on the example circuit.


Save your project through [File] -> [Save] selections.

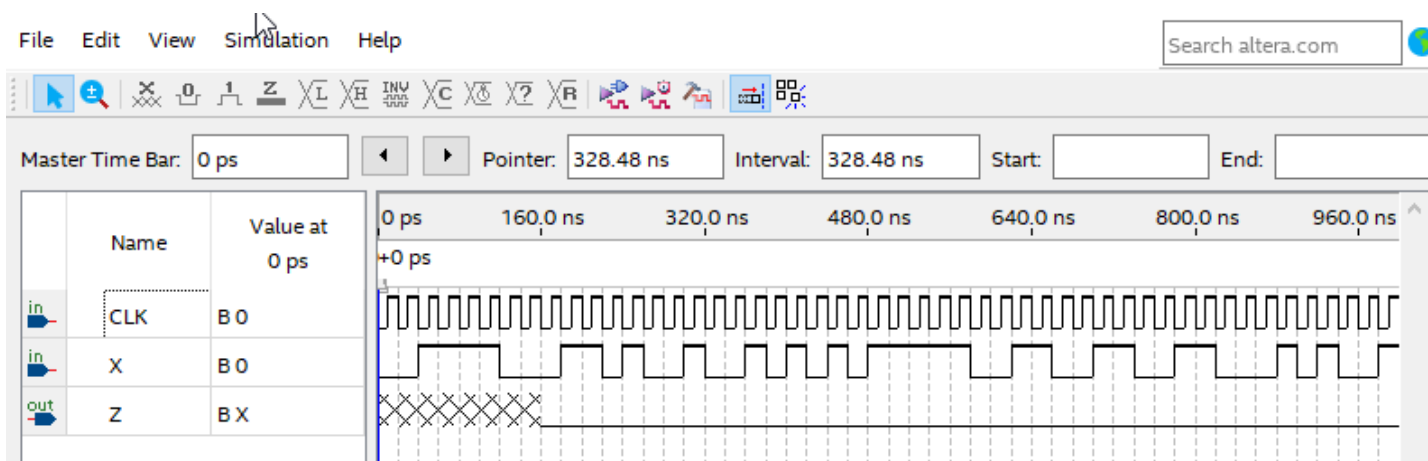
Step 3. Compile your project:

1. Follow [Processign -> [Start Compilation] menu items to compile your project..
2. If there are no errors in your project, you will get “Full compilation was successful” message on a pop-up menu.
3. Close the compiler pop-up menu.

Step 4. Simulation of your project:

You can simulate your project in two ways: In **functional simulayon** we assume that all circuit components snd connection lines are ideal (they work without any signal delay). The other option is **timing simulation** that takes into account the component delays and wiring delays in simulation. This is a more complicated procedure. Functional simulation takes less time and simpler to carry out because it can be implemented over logical behavior of the underlying system. We will use functional simulation for the time being. For this purpose, follow the steps described below:

1. Select [File] -> [New] -> [Verification / Debugging Files] -> [University Program VWF], and click on “OK” button.
2. On the new window, select [Edit] -> [Insert Node or Bus].
3. Click on “Node Finder” button.
4. In the list of your circuit pins, select move-all “>>” and click on the “Start” button.
5. You can change the order of pins.
6. Enter [Edit] -> [End Time] menu and set “set end time” value as 1 microsecond (This is the default value)
7. Click on the “Overwrite clock”  button, arrange time periods of CLK, X as follows:
8. Set the pulses on X randomly.



9. From [Assignments] -> [Settings] menu, follow “Simulation Settings” and change simulation “Mode” as “Functional”.
10. Follow [File] -> [Save] to save your simulation model.
11. Follow [Processing] -> [Start Simulation] to start your simulation.

Examine simulation outputs and verify that your system works as designed.

Phase 2: Implementing the above clocked sequential circuit in Verilog HDL

For the circuit you draw and simulated in phase 1, its Verilog HDL implementation is given below:

```
/* This circuit has one input X and one output Z such tha
Z is equal to 1 if the last four bits on X is equal to 1001,
overlapping of consecutive codes is allowed.
*/
```

```
module SeqDetect1001(CLK, X, Z);
input CLK;
input X;
output Z;
reg Q0, Q1;
wire J0, K0, J1, K1;
wire Z;

// Combinational logic
assign J0 = X | Q1;
assign K0 = ~X;
assign J1 = ~X & Q0;
assign K1 = X | Q0;

assign Z = X & Q1 & Q0;

always @(posedge CLK)
begin
    Q0 <= (J0 & ~Q0) | (~K0 & Q0);
    Q1 <= (J1 & ~Q1) | (~K1 & Q1);
end

endmodule
```

- i. Open a new project in VeriLog HDL and type the above given code in its editor.
- ii. Compile the code.
- iii. Simulate it by following the same procedure as explained in Phase 1.

HOMEWORK: (To be submitted at the beginning of the next laboratory work)

Design a clocked sequential circuit with one input X and one output Z for the detection of the 4-bit sequence 0110 on input line X. Output Z=1 when this sequence is detected, Z=0 otherwise. Overlapping of 4-bit codes are allowed. Assume that MSB arrives first.

- i. Implement and simulate your design in VeriLog HDL environment using its schematics.
- ii. Implement and simulate your design in VeriLog HDL code.
- iii. Provide circuit schematics, HDL code and simulation results in your homework report.