HACETTEPE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ELE227: FUNDAMENTALS OF DIGITAL SYSTEMS LABORATORY

LABORATORY RULES

Experiments

- There will be **five (5)** experiments with **five (5) preliminary works and no lab reports.**
- The first experiment will be held on **Monday, 11th November** at **Modelling and Simulation Laboratory 1** on the second floor of new building.
- A short Quiz will be held on the preceding Friday (2 days before experiment day) for each experiment in E4 classroom at 12:15, hence the first quiz will be on **Friday, 8th November.**
- **Very High Speed Integrated Circuit Hardware Description Language**, named as **VHDL**, is the programming language, which you will learn, during the term.
- **ISE Design Suite of XILINX** will be used as coding and simulation environment. (You can download it with your **school e-mail** from website of XILINX.)
- A simultaneous question based on preliminary work of related experiment is asked to you during experiment, in order to get full credit from experiment, **you need to make a proper design by your hand** then **implement and test your design as VHDL code.**
- Laboratory day will be announced to you at least two weeks before the experiment is performed.

Preliminary Work Rules

- You have to bring your **'printout'** preliminary works to laboratory when you perform the experiment. Printouts **must be stapled** and **printed using both sides**. Preliminary work printouts must be printed double sided in order not to waste space and paper.
- **If you do NOT have preliminary work, you have NO chance to attend the experiment.**
- To get full credit from preliminary works, it **MUST** contain
  - ✓ **Hand-drawn circuit design**
  - ✓ **RTL schematics** *(change black background to white.)*
  - ✓ **VHDL code of top block** *(it is enough to copy entity part and inside the part of architectural behavior, please take your preliminary works as short as possible.)*
  - ✓ **Test Bench code** *(only the parts with tested values)* and
  - ✓ **Output graph of Test Bench**

  for each question **IN THE GIVEN ORDER!!!**
- All the text should be printed as text; you should **not** get any screenshot of your codes or comments. **No screenshots of text parts!!**
- All the text in your document must be placed properly in a Word processing program (MS Word, OpenOffice Write, LaTeX etc.), you should **not** get printouts directly from Xilinx ISE code editor.
- All the graphics in your report must be with white background. These include RTL schematics and Test bench outputs.
- When preparing the preliminary work, **you should not put any photograph taken with a camera or camera scanner app like CamScanner**. If your design includes a part with a design that can be prepared with pen and paper, you should leave those parts empty in your document, then you should draw them by hand later. This is valid for logic gate drawings, truth tables, k-maps, state diagrams etc.

HACETTEPE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ELE227: FUNDAMENTALS OF DIGITAL SYSTEMS LABORATORY

Attendance:

- You have to attend all of experiments.
- IF YOU DON'T ATTEND AN EXPERIMENT, **YOU MAY MAKE-UP FOR IT IF YOU HAVE A VALID REASON WITH NECESSARY DOCUMENTATION!**
- If you don't attend and don't make up more than one experiment, **YOU WILL AUTOMATICALLY GET GRADE F1.**

Grading:

- Preliminary Works → 10 points
  - ✓ 5 Preliminary Works → **50 points**
- Experiment Performance → 10 points
  - ✓ 5 Experiments → **50 points**
- Quiz Performance → 5 points = 5 Quizzes → **25 points**
- Total → (125 x 60%)+(100 x 40%)(FINAL) = 115 points with grading over 100 points

CONFIRM YOUR E-MAIL ADDRESS at STUDENT DATABASE SYSTEM (OGR). TEST IT WHETHER YOU GET A MAIL OR NOT! UPDATE YOUR PHOTO ON OGR SYSTEM WITH A PROPER ONE!

Xilinx ISE and VHDL Rules

- Create a new directory on Desktop for Xilinx project, always work in this project folder.
- After completing a part of your design, **backup** your project folder in case you broke anything in your code. If you find a correct result, **backup** your work before changing the design.
- Always name your modules/components/variables/signals with meaningful names. Designs that include elements like "asdf" or "qwe1, qwe2" will not be accepted or evaluated.
- Unless otherwise stated, **you should always use logic gates in your design**. We **do not** use statements like if-else, case-when etc. in the scope of this laboratory.
- You **should not use** "inout" type in any of your design. You should only use "in" or "out" as a port type. If you need to set and read a particular signal in several parts of your design, you should create a "signal" and use/connect it to proper ports.
- If you have an input/output/signal with more than 1 bit, it should be an "STD_LOGIC_VECTOR".
  Example: One of the inputs of a 3-bit multiplier should not be 3 inputs as "STD_LOGIC" named A1, A2 and A3. Instead it should be an input A with the type "STD_LOGIC_VECTOR" and with the length 3 bits.

```
entity Multiplier3bits is
    Port ( A1 : in  STD_LOGIC;
           A2 : in  STD_LOGIC;
           A3 : in  STD_LOGIC;
           B1 : in  STD_LOGIC;
           B2 : in  STD_LOGIC;
           B3 : in  STD_LOGIC;
```

WRONG ✗

```
entity Multiplier3bits is
    Port ( A : in  STD_LOGIC_VECTOR (2 downto 0);
           B : in  STD_LOGIC_VECTOR (2 downto 0);
           C : out  STD_LOGIC_VECTOR (5 downto 0));
end Multiplier3bits;
```

RIGHT ✓

HACETTEPE UNIVERSITY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
ELE227: FUNDAMENTALS OF DIGITAL SYSTEMS LABORATORY

General rules & recommendations about lab work:

- You should properly design your work on paper with all necessary steps such as truth tables, state diagrams, k-maps etc. before advancing to the coding step.
- You should try to troubleshoot any problem you may encounter before seeking help from others.
- When designing a sequential circuit that have several states, you should decide the correct and number of states that is required when designing on paper.
- You should be careful about using minimum number of gates, selecting number of inputs/outputs while designing and implementing your logic circuits.
- When forming a k-map you should be careful about using the maxterms or minterms as reference.
- Reset circuit, especially in sequential designs, is very important and it should be properly implemented in design part and correctly used in test bench part.
- Design details such as with/without overlap, type of the state machine (Mealy or Moore), type of the component that must or must not be used (e.g. type of the flip-flops) are important and obeying them are your responsibility. **If you do not conform to the exact design requirements, you design may be considered as incorrect.** If you have a doubt about any detail, you should ask laboratory assistants before assuming and accepting any option as the correct one.
- Text book of the lecture "**Mano M.M., Ciletti M.D., Digital Design, 6/e**" is your best companion for this laboratory. It is in your best interest to solve the related problems before each experiment.
- Every preliminary work for this laboratory is to be prepared individually. Collaborating on a report is not a valid reason the copying someone's work and will be treated as cheating.
- Violation of any condition mentioned in this document will be graded as invalid in your preliminary and experimental works.

TROUBLESHOOTING

Common mistakes about VHDL coding:

- Signal and component definitions should be made in "architecture" part **before** the **"begin"**, not in the entity.

- Instantiations of components (i.e. **component port maps**) should be **after** the **"begin"** in the "architecture".

- If your design has failed to synthesize, you should check the errors starting **from the first line** of the "Errors" tab at the bottom of Xilinx ISE window.

- You should be careful about **undefined** inputs/outputs/signals. You should define these before using them in your design.

- When an entity of a component is changed, component definition should also be changed in any place that component is used.

- You can only assign values to types "out" or "signal". You cannot assign values to "in" type ports.

- You can only read values from types "in" or "signal". You cannot read values from "out" type ports.

- In your entity definition, last definition of ports **do not** end with a semicolon " ; ", if you edit your entity ports later in the code, you should be careful about this.

```
entity tff is
    Port ( clock : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           T : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           nQ : out  STD_LOGIC;
           );
end tff;
```

WRONG ✗

```
entity tff is
    Port ( clock : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           T : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           nQ : out  STD_LOGIC
           );
end tff;
```

RIGHT ✓

- When an entity is used as a component in another entity, ending line should be in the form of "end component;" **not** "end nameOfComponent;".

```
component TFlipFlop is
    Port ( T : in  STD_LOGIC;
           Reset : in STD_LOGIC;
           Clock : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           Qn : out  STD_LOGIC);
end TFlipFlop;
```

WRONG ✗

```
component TFlipFlop is
    Port ( T : in  STD_LOGIC;
           Reset : in STD_LOGIC;
           Clock : in  STD_LOGIC;
           Q : out  STD_LOGIC;
           Qn : out  STD_LOGIC);
end component;
```

RIGHT ✓

- When creating a test bench for sequential circuits, any change of inputs should be made after 1 clock period.

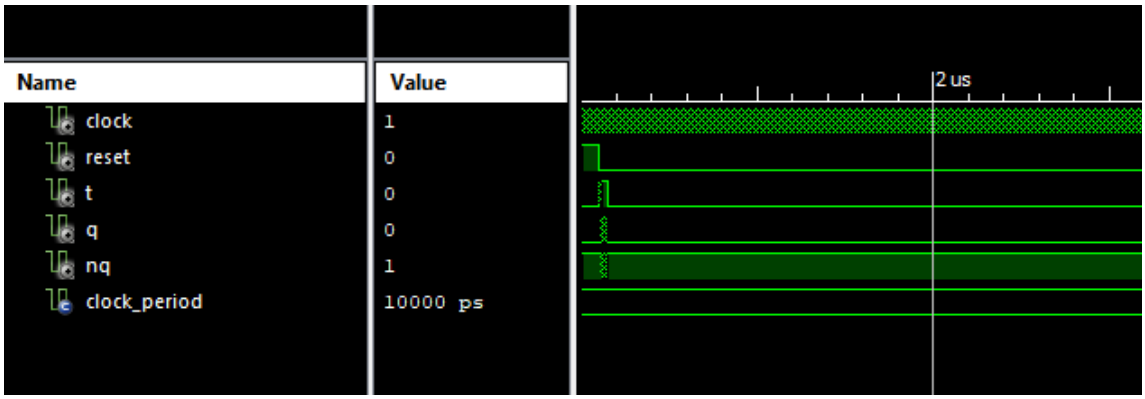
WRONG ✗

```
Reset <= '0';
wait for Clock_period*2;
J <= '1'; K <= '1';
wait for Clock_period*10;
J <= '0'; K <= '0';
wait for Clock_period*10;
J <= '0'; K <= '1';
wait for Clock_period*8;
J <= '1'; K <= '0';
wait for Clock_period*3;
```

RIGHT ✓

```
Reset <= '0';
wait for Clock_period;
J <= '1'; K <= '1';
wait for Clock_period;
J <= '0'; K <= '0';
wait for Clock_period;
J <= '0'; K <= '1';
wait for Clock_period;
J <= '1'; K <= '0';
wait for Clock_period;
J <= '1'; K <= '1';
```

Common mistakes when preparing graphs:



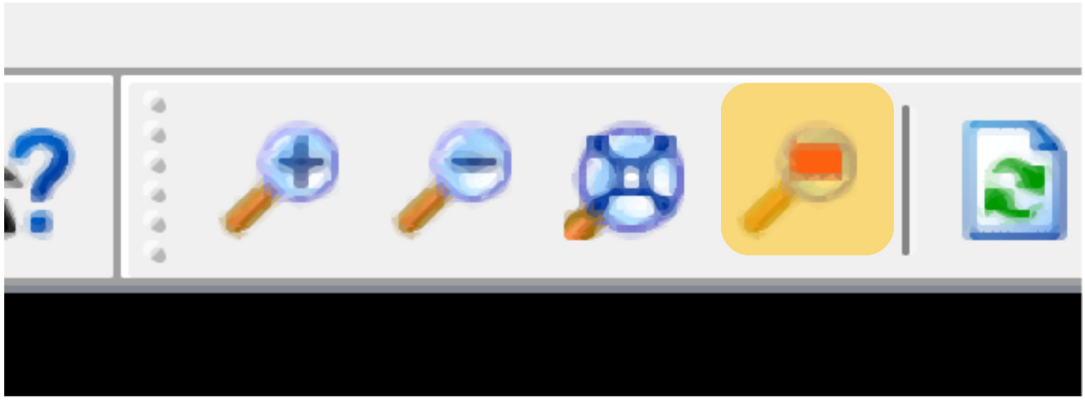
This is a bad example!

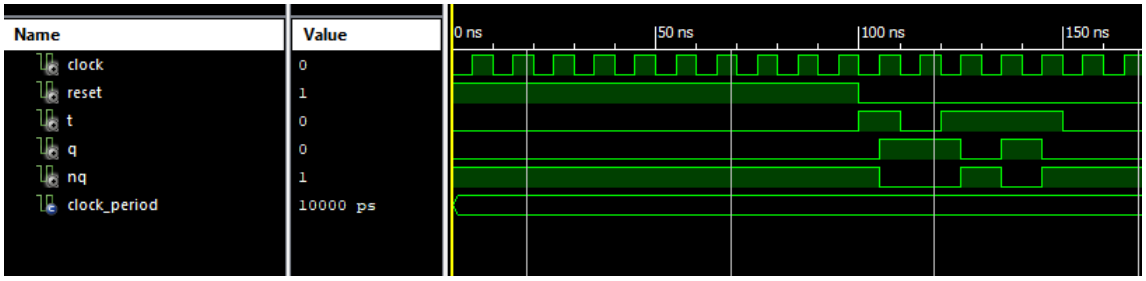
Problem 1: Colored content on black background, background should be white.

Problem 2: Signals are not readable, scaling is wrong, you should zoom to signal region before getting the screenshot

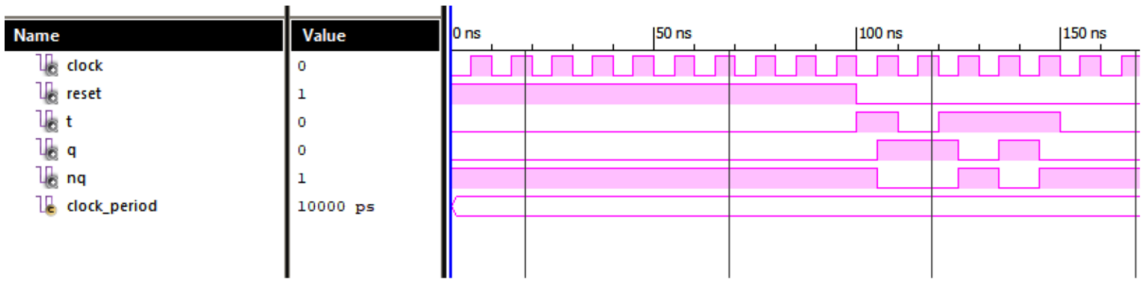
Solution for Problem 1: Use an image processing program to invert the colors.
You can paste your screen shot to MS Paint and use shortcut Ctrl+Shift+I (uppercase letter i) to invert the colors.

Solution for Problem 2: Select the correct time interval with mouse, then click "Zoom to Cursors" button on toolbar.



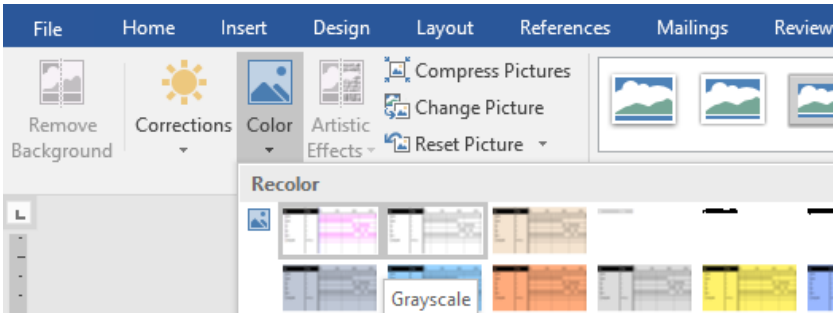


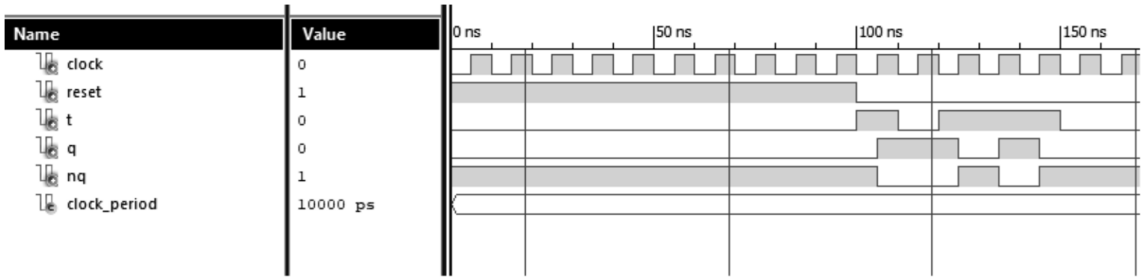
This is a bad example!
Problem: Colored content on black background, background should be white.
Solution: Use an image processing program to invert the colors.
You can paste your screen shot to MS Paint and use shortcut Ctrl+Shift+I (uppercase letter i) to invert the colors.



This is not bad but it may be a problem with an improper printer!
Possible problem: Colored content may not be printed well on black and white, convert image to grayscale for better result.
Solution: You can either convert your image to grayscale in a paint application or you can easily make it grayscale after pasting in the word processing program like Microsoft Word. Steps for making an image grayscale:

- Select the image in MSWord
- From the upper menu (it is called Ribbon menu in new MS Office programs) select "Format" tab
- Select the third option "Color" and select the second color set which is "Grayscale"





This is a very good example!

Signals values are visible, background is white, everything will be printed as seen with a black and white printer.