HO CHI MINH UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Digital Systems – CO1009

Assignment

HCM City, 2018

1. Brief Introduction

- ❖ HDL Language: Verilog
- ❖ Platform: Board De2i-150.
- ❖ The maximum members of a group is 10. And each group will be assigned for 1 topic.
- ❖ Extra function will be counted as a bonus (with maximum is 1 point). Student should discuss with TA first.

2. Warning

- • Any copy or minic action will lead to zero point for both groups.
- • The report document is a requirement.

3. Timeline

- ❖ **Week 1**: Topics for groups.
- ❖ **Week 2:** Progress Report. A short slide presentation:
  - Design of the group (block diagrams).
  - The functions of the blocks in the block diagrams.
  - Difficulties and solutions (if any).
- ❖ **Week 3:** Final Report. Make a report and submit directly when demo with Instructors. Reporting requirements should include the following:
  - Part 1: Introduction. The contents of the topic introduction, tools, equipments used, functions of the product.
  - Part 2: Design. Including block diagrams, descriptions of function blocks.
  - Part 3: Implementation. Including the ways of implementing the design (using a behavioral model, schematic, or structual model ....), it is possible to draw flowcharts that show the flow of the system.
  - Part 4: Conclusion. Including summary, future directions, difficulties encountered, and work breakdown.

Assignment Requirements

Student is encouraged in taking initiative in presenting the layout, content and results.
Howerver, the report has to include these contents:

- ✞ Introducing about the design, the purpose of design.
- ✞ Implementation of the system which includes the state machine, the module structure, the pin diagram and the explaination.
- ✞ Result of the system like how it use, how it functions, its resource and frequency.
- ✞ Each modules or submodules must be accompanied with a specific description.
- ✞ Work Assignment to handle the tasks.

Note: Do NOT add source code to the report.

Topic 1:

A security monitoring system is built by a set of intrusion detection sensors. Each sensor is used for a certain range of areas. There are all 8 areas where sensors must be attached, when detecting unauthorized access, the corresponding sensor of each zone will generate a warning signal. Use your knowledge of digital systems to build this security monitoring system on the Altera DE2-150i.

- Requirements:
  - Simulate the operation of 8 sensors corresponding to 8 zones with 8 switches on board DE2-150i
  - Simulate the operation of 8 sensors on 8 LEDs. When the sensor is in active mode, the LED of the corresponding sensor will be on. In contrast, the LED will turn off.
  - Simulates unauthorized intrusion by switches or buttons
  - Activate the system using three 7-segment led HEX0, HEX1, HEX2:
    - Use Button or Switch to support the function to turn on / off the system.
    - The operating state of the system must be displayed on the 7 segment Led HEX0 (eg system is operating - E, the system is down - 0)
  - Security statuses:
    - When the alarm system is activated and there is intrusion detection, the system will display the alarm on the 7 segment HEX1 LED being penetrated.
    - When the alarm system is activated and there is no intrusion, the system will display on LED 7 segment HEX1 "-"
    - If multiple sensors are found to have intrusion or faults occurring during operation, the system will display on the HEX1 7 segment LED "-" and HEX2 displays the number of intrusion detection sensors

Topic 2: Design an ALU 4-bit on board Altera DE2-150i. (Ref:IC 74LS382)

- Requirements:
  - The ALU has two 4-bit operands, the operand value is entered using a switch, and each input operand must be displayed on a 7-segment LED. Only 8 switches and 2 buttons are allowed. SW [3: 0] is used to enter the number input, SW [4] is used to select which operands to enter, SW [7: 5] is used to select one of the six operations below. Button KEY0 is used to load the number of SW [3: 0] into and display on the 7-segment LED, button KEY1 used to perform mathematical operations and display the result of the 7 segment LED.
  - ALU is capable of performing six operations: addition, subtraction, AND, OR, XOR, translation of the bitwise over the current result of the two input operands.
  - ALU provides the ability to detect digital overflows as a result of operations beyond the allowable range.
  - Results of mathematical operations are displayed on 7 segment LED and single LED.

Topic 3: Design a traffic light.

- Requirements:
  - Build a display module using 3 red leds on DE2i to represent the signal in order Red-> Yellow -> Green.
  - An enable / disable signal. When the signal is enabled, the circuit will operate normally, otherwise the light will turn off.
  - Build a countdown timer for each signal type, when count time is 0, turn signal light in order Red-> Yellow -> Green ...
  - Build a module using 7-segment LEDs to display the countdown time of each type of light signal using 7-segment HEX0 and HEX1 LEDs (assuming a maximum count of 99 seconds).
  - When the system starts, the default red is 35 seconds, the yellow light is 4 seconds, and the green light is 25 seconds.
  - [Bonus] Build a module using switch to set the time for each type of light signal from 0-> 99 using the binary code. Only use 10 switches, in which SW [6: 0] to enter the time value, SW [8: 7] to select the set time of one of the three lights, SW [9] to enable, when enable = 1, the light is on, when enable = 0, enter the time for the light. Use KEY0 button to validate the time assigned to the light.
  - [Bonus] Build a module using 7-segment LEDs to display the time configuration parameters for each type of light signal.

Topic 4: A running LEDs system.

- Requirements:
  - System with enable / disable support. When the signal is enabled, the circuit will operate normally, otherwise the leds will turn off.
  - Build a frequency division module with 3 outputs 1Hz, 3.5Hz, 5.5Hz.
  - Build a selector for frequency selection using SW [1: 0].
  - Build a control module for all red LEDs with rules:
    - Leds start with a length of 3 at the right edge.
    - Leds run from right to left, when the 3-led line to the left edge, leds navigation from left to right.
    - When the 3-led line to the right edge, leds navigation from right to left.
    - The process is repeated.
    - [Bonus] The group can add other types of Led running unlike other groups.
  - Build a display module of the 7 segments with current frequency.

Topic 5: A running LEDs system 2.

- Requirements:
  - System with enable / disable support. When the signal is enabled, the circuit will operate normally, otherwise the leds will turn off.
  - Build a frequency division module with 3 outputs 1Hz, 3.5Hz, 5.5Hz.
  - Build a selector for frequency selection using SW [1: 0].
  - Build a control module for all red LEDs, use SW [2] to enable the circuit, SW [4: 3] to select one of the following 4 modes:
    - Leds are initially off, then leds are on from two edges to the middle according to the clock, when all LEDs are on, turn off the leds, and the process is repeated.
    - Leds are initially off, then leds are on from the middle to two edges according to the clock, when all LEDs are on, turn off the leds, and the process is repeated.
    - Leds are initially on, Leds are off from right to left according to the clock, when all leds are off, turn all leds on, and the process is repeated.
    - [Bonus] The group can add other types of Led running unlike other groups.
  - Build a display module of the 7 segments with current frequency.

Topic 6: A running LEDs system 3.

- Requirements:
  - System with enable / disable support. When the signal is enabled, the circuit will operate normally, otherwise the leds will turn off.
  - Build a frequency division module with 4 outputs 0.75Hz, 2Hz, 3Hz, 5Hz.
  - Build a selector for frequency selection using SW [1: 0].
  - Build a control module for all red LEDs with rules:
    - Leds start with a length of 1 at the left edge.
    - Leds are on from left edge -> right edge -> left edge -> right edge -> ..., until all red leds are on.
    - When all LEDs are on, all LEDs turn on and off 3 times. After flashing, all LEDs are on. Then each Led is off according to the rule: right edge -> left edge -> right edge -> ..., until all red leds are off.
    - The process is repeated.
    - [Bonus] The group can add other types of Led running unlike other groups.
  - Build a display module of the 7 segments with current frequency.

Topic 7: Design a digital clock.

- Requirements:
  - Build an 1 counting module with hours, minutes, seconds
  - Build a module showing hours, minutes seconds to 6 7-segment LEDs.
  - The system has an enable / disable signal. When enabled, the system will work normally, otherwise all LEDs will turn off.
  - The system supports reset function. When positive the counter will reset to 00:00:00. LEDs.
  - [Bonus] The system allows you to set the minutes (or hours and minutes).