Faculty of Science, Engineering and Technology

Computer Systems

Assignment 1

Overview

This assignment requires a knowledge of logic gates, flip-flops, registers, counters, controlled gates and input / output.

Purpose: Building an alarm clock

Task: Develop and submit an original logic circuit for a basic alarm clock. The interface must display the time in 12 hour format, indicate AM/PM, and provide the facilities to set the time and an alarm. When the alarm is enabled, an alarm LED is turned on when the time is reached.

Time: This assignment is due by Sunday Sep 18th midnight (start of Week 7)

Assessment: This assignment is worth 20% of your assessment for this unit.

Resources:

- Lecture slides and recordings
- Swin video tutorials:
  - Logisim introduction
  - Half-adder tutorial
  - Full-adder tutorial
  - Intro to Flip Flops
  - Registers with D Flip Flops
  - Ripple Counters (and HEX Display) with J-K Flip Flops

SWINBURNE UNIVERSITY OF TECHNOLOGY

Introduction

You will be building a collection of circuits to implement a basic alarm clock. The project is divided into stages, each holding some proportion of the total marks. Successful completion of all stages will result in an interface that has:

1. A single 4 digit decimal display, using 4 x Logisim Hex Digit displays, that indicates the hours and minutes.
2. An LED to indicate AM and PM, that toggles as expected when midday and midnight is reached.
3. A button/pin to switch into "Set Time" mode.
4. A button/pin to switch into "Set Alarm" mode
5. A button/pin to enable the alarm
6. An LED to indicate when the alarm has been triggered.
7. A button to increment the minutes of the clock in order to set the time or alarm (when in these modes)
8. A button to increment the hours of the clock in order to set the time or alarm (when in these modes)

Logisim Components

Your implementation of the following stages *must only use components we have used in labs*. That is, gates, wires, LEDs, hex displays, pins/buttons, splitters, and flip flops.

Pre-defined integrated circuits like shift registers, counters, multiplexers etc are *not to be used* in this assignment.

Development Stages (and assessment weighting)

The assignment is divided into stages, with each stage assigned a percentage weighting of total marks available. It is recommended you implement each stage in this order, and back-up each stage separately when complete.

Stage 1: worth 30% of total marks available

Implement the minutes counter and display. For this you will need to implement

- a counter for the "units" column display (value range: 0-9), which increments every clock pulse
- a counter for the "tens" column display (value range 0-5), which:
  - increments every time the units column reaches "9", and
  - wraps back to "0" after reaching "5"

- a two digit display that represents minutes as two decimal digits, and wraps back to "00" on the clock pulse immediately after displaying "59" (i.e., as described in the "units" and "tens" counter behaviour described above).

Stage 2: worth 20% of total marks available

Start implementing the Set Time modality. For this you will need to:

- implement a button/pin that switches the clock into "Set Time" mode. When this mode is enabled the time display should pause.
- implement a single "m+" button that manually increments the minute counter each time the button is clicked. This should allow the user to manually set the minutes of the clock by simply adding minutes.
- When the "Set Time" mode is disabled, normal clock ticking should resume from the set minutes.

Stage 3: worth 20% of total marks available

Implement the hours display and complete the Set Time functionality. For this you will need to implement:

- a counter for the "units" column of the hours display, but unlike Stage 1, this will wrap back to 1 after the "tens" column reaches 2 (i.e, 12 after o'clock, we wrap back to 1 o'clock)**.**
- a counter/toggle for the "tens" column that displays values "0" or "1": "0" for the first ten hours, and "1" for hours eleven and twelve
- a "h+" button that manually increments the hour display each time the button is clicked. This should allow the user to manually set the hour of the clock.
- an AM/PM LED that toggles between "off" and "on" (off for AM, on for PM) when the hour reaches 12.

*****implementing "wrap back to 1" in the units counter will require some extra thought. It is worth 5 of the total 20 points for this stage.

Stage 4: worth 15% of total marks available

Now integrate the hours and minutes of the clock display so that hours and minutes tick with the system clock (when not in Set Time mode):

- Connect up your "minutes" counter from Stage 1 to your "hours" counter from Stage 3 so that the hours increment each time the minutes display wraps back to "00")
- ensure your AM/PM LED toggles as expected
- avoid illegal values appearing on the display

Stage 5: Set Alarm (worth 15% of total marks available)

Now implement the Set Alarm functionality. For this you will need to:

- implement a button/pin that switches the clock into "Set Alarm" mode. When this mode is enabled the time display should pause.
- implement the manual counter setting using the same functionality implemented for "Set Time" in Stage 3. You should allow the user to set an alarm time using the same 4 digit time display. This will need you to think about how you are going to remember the time that was being displayed prior to entering the Set Alarm mode.
- implement a button that enables the alarm, such that when the clock time reaches the set alarm time, an Alarm LED is turned on (indicating the alarm is triggered) until it is turned off again (using the same button).
- When the "Set Alarm" mode is disabled, normal clock ticking should resume from the time that was on the display prior to entering the Set Alarm mode

Submission Instructions

Your completed submission must be made through Canvas - (Go to Assignment 1 under "Assignments" before the due date/time.

Everyday day late will incur a 10*% deduction*.

Each submission should be zip file containing:

1. the actual Logisim file (.circ source file)
2. a report (Word doc or PDF) containing:
    1. Your name, student number, unit code and lab session
    2. A description of the circuit
    3. An outline of your design (in terms of functional blocks of gates, devices)
    4. Any assumptions you have made
    5. Any unresolved problems with your design
    6. Pasted screenshots of your working circuit.

Assessment

Marks will be allocated as per the weighting of each stage. Note that earlier stages attract more marks than later stages, and so obtaining a Pass should be achievable with a solid grasp of the basics.

Your circuit is expected to work! That is, we will be testing your submission in logisim, and most marks will be based on the system's correctness.

Where bonus marks are awarded, they will be awarded only up to the total marks available.

MAKE IT READABLE!

If we cannot follow your logic, we will deduct marks. Keep your layout neat, and use clear labels. You are also encouraged to use sub circuits to make your design more modular (and your circuits more re-usable).

Keep your display and user interface components in one place so we can easily operate your submission.