ECE 3544: Digital Design I

Project 4: Clock and Countdown Timer Alarm (Design project)

Objectives:

In this project, you are going to design and implement a timeclock with countdown timer alarm. The project should give you more experience working with Finite State Machines (FSMs) and sequential logic.

Requirements:

You must have ModelSim ALTERA STARTER EDITION and Quartus II Web Edition 16.1 installed on your computer. The instructions are done using these versions of ModelSim and Quartus. While the programs may be consistent in other versions, it is recommended that you use these versions.

Note: The DE1-SoC Board IS REQUIRED for this project.

Project Description:

In this project, you will design and implement a simple timepiece with two functions: time-of-day clock and countdown alarm. You will have to create the state machine for this timepiece and implement it in Verilog. The inputs and outputs for the timepiece and its expected behavior are specified in this document.

Submit a detailed block diagram representing your implementation of the system as part of your report.

In Quartus, choose **Tools > Netlist Viewers > RTL Viewers** *to view the RTL schematic that represents your top-level Verilog module. You may submit a screenshot of this schematic if you believe that it is sufficiently detailed, simple to follow, and descriptive of the structure of your implementation. It might interest you to view the structure of the instances in your top-level module; clicking the boxed plus sign on any instance allows you to see the structure of that instance.*

For this project, the basic module that you will receive only represents the inputs and outputs of the DE1-SoC board. The system will not require all of the inputs and outputs that the top-level module declaration includes on its port list or that are listed in the port declaration. You may use the unused inputs and outputs for any testing or debugging purpose that you decide is appropriate. **You must perform the pin assignment for the top level module before downloading your design to the DE1-SoC board.**

You will be responsible for writing all of the Verilog that implements the design elements that comprise your stopwatch system. *You may re-use the* keypressed *modules from Project 3B. You may also re-use any of your own original Verilog code.*

Countdown Alarm Functionality:

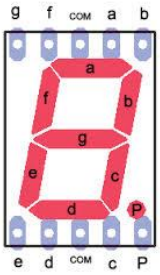
The countdown alarm will allow the user to set an arbitrary time in minutes from 1 minute up to 9:59 (9 hours and 59 minutes) until an alarm occurs. The time is kept in hours, minutes and seconds until the alarm, H:MM:SS. The alarm should count down from the start value at 1 second intervals based upon pushbutton input and can be paused, restarted from the current value, or reset to the start value. When the alarm reaches 00:00, the seven segment displays should begin to blink at approximately 1 second intervals (on for 0.5 seconds and off for 0.5 seconds, showing 00:00 while on) and continue

blinking until any pushbutton KEY[1]-KEY[3] is pressed. The alarm should then remain at 00:00 until reset to its start value.

For the countdown alarm, the DE1-SoC board should display time in binary coded decimal (BCD) on the seven segment displays. For the countdown alarm, time will be displayed as five BCD digits, one digit for the hours (0 to 9), two digits for the minutes (0 to 59 minutes) and two digits for the seconds (0 to 59 seconds). Since there are five digits in the time, the board should display the remaining time on the rightmost five digits. The leftmost digit should show an upper case A when in countdown alarm mode.

Operation in countdown alarm mode is as follows:

Seven segment displays: The time remaining until the alarm flashes is displayed on the seven segment displays, with capital A as the leftmost digit (HEX5) to indicate the countdown alarm mode and the time remaining on the rightmost five digits (HEX4:HEX0). The capital A should be formed by turning on all segments except the bottom horizontal segment (segment d in the diagram below).



To set the start value: Press KEY[1] when at the starting value for the timer. The current start value in hours and minutes is displayed on the displays, with hours flashing and minutes not flashing. Press KEY[3] to switch to setting the minutes, with minutes flashing and hours not flashing. Pressing KEY[3] repeatedly will cycle between whether the hours or minutes are selected to be changed (flashing). It should not be possible to set the seconds. Press KEY[2] to increment the selected value by one. The count should turn over between 9 and 0 for hours and 59 and 0 for minutes. When the desired value is reached, press KEY[1] to save it as the start value.

To start/stop the countdown alarm: Press KEY[2] to start the countdown alarm counting down from its current value. Press KEY[2] again to stop the countdown. It should be possible to start/stop the countdown as many times as desired.

To reset the countdown alarm to the start value: Press KEY[1] to reset the countdown alarm to the start value. If the countdown alarm is counting when KEY[1] is pressed, the alarm should NOT continue counting after being reset by KEY[1].

To shut off the flashing alarm: Press any of KEY[1], KEY[2], or KEY[3] while in countdown alarm mode. If the current mode is the time-of-day clock, then the mode must be changed back to countdown alarm before shutting off the alarm.

Once the alarm has reached 0:00:00, it should stop there until reset to the start value with KEY[1], i.e., it is not possible to start the countdown again with KEY[2] once the alarm has gone off.

To switch to time-of-day mode: Press KEY[3] except for when shutting off the alarm or when setting the start value. When switching to time-of-day clock mode, the countdown alarm should continue counting if it is counting, and it should maintain its current value if not counting.

Time-of-day clock Functionality:

The time-of-day clock shows the current time, ignoring A.M. and P.M., in 12-hour format (i.e., the digits for the hours range from 1 to 12). The time will be kept as hours, minutes, and seconds, XX.YY.ZZ, where X are the hours digits, Y are the minutes digits and Z are the seconds digits. The clock should be always running except for when reset is active (i.e., KEY[0] is held low). The time will also be displayed as five or six BCD digits, with one or two digits for the hour (i.e., one digit for 1-9, and two digits for 10-12, two digits for the minutes (00 to 59 minutes), and two digits for the seconds (0 to 59 seconds). At 12:59:59, the clock should turn over to 1:00:00. The rightmost five seven segment displays should be used for the time, with the leftmost digit being used when the hour is in the range of 10-12. For hours from 1-9, the leftmost digit should be blank (off).

Operation in time-of-day mode is as follows:

Seven segment displays: The time should be displayed on the seven segment displays, using the rightmost five digits for times from 1:00:00 to 9:59:59 and all six digits for 10:00:00 to 12:59:59. The left digit should be blank when not in use.

Setting the time of day: Press KEY[1] at any time when the time of day is displayed to set the current time. While setting the time, the clock should continue running, i.e., it should not halt while setting the time. When KEY[1] is first pressed, the current time should continue to be displayed on the digits, with the seconds flashing, and hours and minutes not flashing. Press KEY[3] to switch to setting the minutes, with minutes flashing and seconds and hours not flashing. Press KEY[3] to switch to setting the hours, with hours flashing and minutes and seconds. Pressing KEY[3] repeatedly will cycle between whether seconds, minutes, or hours are selected to be changed (flashing). When setting the seconds, pressing KEY[2] will reset the seconds to 00 without affecting the hours and minutes. When setting the minutes and hours, pressing KEY[2] will increment the selected value by one. When setting the hours or minutes, the value should turn over from 12 to 1 for hours and 59 to 00 for minutes. When the desired time is reached, press KEY[1] to set it as the current time.

To switch to countdown alarm mode: Press KEY[3]. When switching to countdown alarm mode, the time-of-day clock should continue running.

Summary of Inputs and Timepiece Behavior

The functions of the timepiece are controlled using the pushbuttons as summarized in Table 1.

Table 1: Summary of pushbutton control

Input	Behavior
KEY[0] Reset	Global reset for the system, asynchronous and active low. The countdown alarm start value should be set to 1:00 (1 minute), and the current time to 12:00:00. The countdown alarm should be stopped (not counting) and the time-of-day should start running as soon as the reset button is released. During reset (KEY[0] = 0), the rightmost four seven segment displays should display the last four digits of your student ID in BCD. When exiting reset, the seven segment displays should display the time-of-day clock.
KEY[1] Adjust	Time-of-day clock: Set and save the time of day. Countdown alarm: Reset to the start value of the timer. If the timer is at the start value and not running, then KEY[1] is used to set and save the starting value. If

	the alarm is flashing, KEY[1] will turn off the alarm.
KEY[2] Start/stop	Time-of-day clock: When setting the time of day, KEY[2] is used to clear the seconds, or increment the minutes and hours, depending on which is currently selected to be set. Countdown alarm: Start/stop the counting of the countdown alarm. When setting the starting value, KEY[2] is used to increment the hours and minutes of the start value, depending on which is selected. When the alarm is flashing, KEY[2] will turn off the alarm.
KEY[3] Mode	Time-of-day clock: When setting the time of day, KEY[3] is used to switch between which digits are being set (seconds, minutes, and hours). When not setting the time of day, KEY[3] switches to countdown timer Countdown alarm: In setting a time, KEY[3] is used to switch between which digits are being set (hour or minutes). When alarm is flashing, KEY[3] will turn off the alarm. Otherwise, KEY[3] will switch to time-of-day mode and the countdown alarm will continue with its current behavior (counting if counting, or at its current value if stopped).

The LEDS and DIP switches are included the module declaration in the Quartus project. You can use them for debugging your design, but debugging modes are optional.

In addition to the behavior shown in Table 1, your design must meet the following requirements:

1. All sequential elements of your design must use the on-board 50 MHz clock as the clock. For sequential elements that change values at lower frequencies, use an enable signal at the desired frequency as a logic input rather than as a clock.

2. The timepiece must keep time to within 0.01%, assuming that the 50MHz clock is correct. For example, if the time-of-day clock were to be run for 12 hours, the elapsed time should be within 4.3 seconds of 12:00:00.

3. When either of the countdown alarm or clock is counting, they should continue counting when the mode is switched to some other mode, including debug mode. For example, if you have started the countdown alarm, then switch to the time-of-day clock, the alarm should continue counting down. When displaying the current value of the countdown alarm, the clock should continue running. Furthermore, the buttons should only affect whichever timepiece mode is currently displayed on the seven segment displays. For example, KEY[2] should only start/stop the countdown alarm when it is the currently displayed timepiece mode.

4. If the countdown alarm reaches 0:00:00, the seven segment displays should flash no matter what mode is currently active. To stop them from flashing, the user must go back to countdown alarm mode and press a pushbutton.

5. Your design must use the templates for finite state machines described in class. For example, your module must not create any warnings about latches or signals not appearing on the sensitivity list.

Design

As always, you should design smaller parts of this project individually and implement and test them before moving on to the next part of the design, rather than trying to implement the whole system

before testing any of it. In particular, you should consider the counter(s) required to scale the 50 MHz clock down to frequencies that are appropriate for the time-of-day clock and countdown alarm, the types of counters required for the time-of-day clock and the countdown alarm, and the FSM for implementing the input/output behavior for the pushbuttons, switches, and seven segment displays.

You should test each of your components in simulation using a testbench before trying them on the DE1 SoC. Your report should include waveforms showing correct behavior of your design for all of the required operations.

Design Tips

- Design smaller aspects of this system individually. Implement and test one before moving on the next one. *Don't try to implement the whole system before testing any of it*.
- Make a state diagram to represent the manner in which button presses change the mode and level of operation. *Include the state diagram for your "system controller" in your report.*
- Decompose the design into communicating finite state machines, e.g., a set of counters that receive direction from your system controller and might also provide information to the system controller, or counters that supply outputs that control the inputs of other counters.
- Make a block diagram of the system before you write any of the module. This will help you settle the manner in which the various state machines should interact. *Include a detailed block diagram representing your implementation of the system as part of your report. You need not break-out the structure of every block in your diagram, but a good guideline is to include a block for every Verilog module you write and instantiate in some other module.*
- Use the keypressed module from Project 3B to generate a one-clock-cycle enable signal when a pushbutton is pressed and released.

Submission:

Write a report describing your design and implementation process. As stated above, your report should include waveforms showing the correct behavior of your design. Your report should include a discussion of the decisions you made about how to implement each aspect of the design. Your report should specifically address the accuracy of the time kept by your design, assuming that the 50 MHz on-board clock is accurate. You should devise a simple test (presumably involving repeated trials) to test the accuracy of your timer, and include the results of your test in your report. Your report should include block diagrams for your design, state diagrams for any FSMs in the design, waveforms showing correct operation, and any information necessary to operate your design that is not included in this project specification. If you had any debugging modes, you should document how the debug mode operations can be used to test your design on the hardware and describe the motivation for each debug operation.

I have included the validation sheet that the GTAs will use to test your design after the submission deadline. You do not have to go to the CEL to have your project validated. Instead, you should use the information in the validation sheet as one basis for testing your design before you submit your project.

Your project submission on Canvas should include the following items:

1. Project report in Word or PDF

2. A Quartus Archive containing the source files for your top-level module, any modules that the top-level module requires to function, and your test benches for the top-level module.

To create the Quartus Archive, choose **Project > Archive Project** after you complete your implementation. When prompted for a name for your archive, the default archive name will be the

same as the original archive. *Append your Virginia Tech PID to the end of the filename*. Make certain that you submit the archive that you create – the one containing your solution to the project – and not the one that I provided to you – the one that only contains the initial files.

Your timepiece module will be tested with our own secret testbench, so it is important that you use the module declaration in the files provided with this specification. You must also include the source files for every module required for your top-level module. Failure to do so will result in a grade of 0 for those portions of the project. Your module may be submitted to the MOSS service (theory.stanford.edu/~aiken/moss/) to check for plagiarism. Any copying flagged by MOSS will be treated as Honor Code violations.

Your report and Quartus archive files should be submitted as a single zip file on the project assignment page.

Grading for your submission will be as described on the cover sheet included with this description.