Andrew Covarrubias (axc554)
Jonathan Monreal (jem177)

Design Questions

Question 1:
Which modules can you easily reuse from your previous designs? What improvements do you need to make to successfully complete or update the modules?

We can reuse a majority of the modules from lab 3 and lab 5.

From Lab 3 we have mult, shiftreg, slow_down, sync_clk, and values modules

From Lab 5 we have button_clock_generator, char_displayer, char_ram, input_handler, loader, video_controller, and video_position_sync
Most of the modules just needed a few tweaks so that they could function together, but besides that they were left almost intact

Our new modules are char_displayer, second_clock_gen, timer, video_controller, and video_position_sync

Question 2:
Convert the specifications given above and in lecture into your own written description. Draw a simple data-flow block diagram to illustrate the overall design. What sequence of implementation do you plan to use? Give a list of modules you want to implement

mult
shiftreg
slow_down
sync_clk
values

button_clock_generator
char_displayer
char_ram input_handler
loader
video_controller
video_position_sync

char_displayer
second_clock_gen timer
video_controller
video_position_sync

We plan to implement everything to be displayed on the actual LCD first before we begin to work on the alarm functionality of the project which will implement the DAC and sounds functionality.
See attached diagrams for the block diagram.

Question 4:
List the modules your group plans to utilize in the project implementation and sketch a bare-bones block diagram that indicates which modules will have connections.

mult
shiftreg
slow_down
sync_clk
values

button_clock_generator
char_displayer
char_ram input_handler
loader
video_controller
video_position_sync

second_clock_gen timer

See attached block diagram.

Question 5:
What is the planned distribution of implementation and documentation work between the group members for this lab?

We plan to distribute the work based on who did the majority of the work for each lab. So anything related to the DAC and audio would be worked on by Andrew. Anything related to the LCD display would be worked on by Jonathan. The lab report and any of the new code would be worked on in a collaborative effort.

Response Requirments

Requirement 1:
Give a brief summary of the whole labs implementation. This is a sort of high-level preview of the system calling out what technology is used and the resultant functionality. Briefly mention any differences between implementation and specification here.

Our lab will be a 24 hour timer which can go up 23 hours and 59 minutes. It can be set for a specific set of number of hours, minutes, and seconds all of which are displayed on the lcd display screen. There is also an alarm mode which will change the lcd display to red and once the timer reaches zero. There is a reset function attached to KEY0

Requirement 2:
Present the block diagram of your design. Briefly explain the module organization/hierarchy and how data flows through the system.

See attached block diagram.

Requirement 3:
For each module in the project, document its interface, intended behavior, and the functionality it implements. Use a mixture of SignalTap captures, analog/digital oscilloscope captures, and text to develop a complete description of the module. Use captures judiciously; simple functionality needs only a simple description and may be excessively lengthened or made confusing by the inclusion of waveform captures. Both SignalTap and oscilloscope captures must be used

mult- Will multiply output from the DAC
shiftreg- Shift all of the DAC output by 1 bit to the right and replaces the shifted bit with a zero
slow_down - slows the clock down for the DAC to be synced
sync_clk - used to sync the clock with the NCO clock in order for them to match one another
values - gets the values from the NCO and converts them over to values that can be displayed

char_displayer - display the characters drawn out and places them on the LCD
loader- loads the display from the ROM
video_controller - controls the refresh rate and the what is displayed onto the clock
video_position_sync - assigns the characters over to certain positions on the screen

second_clock_gen timer - generates the clock which will be used by the timer on the actual LCD display