Project #1 Design Log

ECE372
Ryan Nand

01/07/2021

- I decided to review material from ECE371 by rereading chapters 1-5. My plan was to finish this goal over the weekend.
- Ordered materials (B3, emulator, kit)
- Installed IDE CCS

01/14/2021

- I read the project specifications and requirements.
- I created this document for logging purposes.
- I am now looking into the LCD module and how I would wire it to the B3.
    - Inputs: RAW, Rx, GND
        - RAW - 3.3V to 9V power input
        - Rx - UART signal input
    - Default Baud rate - 9600bps
    - Change to line to 1
        - Send ascii ‘|’ set command
        - Followed by “7” byte
    - Change background to green
        - Send ascii ‘|’ set command
        - Followed by “187” byte
        - To turn off same procedure as above but r=“128”, g=“158” and b=“188”
    - To write my name
        - simply use “.ascii “Ryan”"
    - To shift entire display
        - Send “254” command character
        - Followed by “0x1C” byte
    - To home the cursor
        - Send “254” command character
        - Followed by “128” byte

01/15/2021

- I start to write the high level algorithm following the guidance from the book
    - For primary guidance I am using the UART section pages 256-261
    - For secondary guidance I am using the information from creating the timer program pages 241-247

01/18/2021

- I am now writing the low level algorithm after finishing the high level algorithm
    - My strategy here is to go slow and make sure each instruction is translated correctly
    - For technical details I am using the book, B3 Reference Manual, and the AM335x datasheet.
        - Everything helps make sense of the register mapping, the pinout of the B3, and the instructions I need.
        - The datasheet is useful in not only finding the registers I need but also understanding why the book uses particular registers and their offsets.

        - Mapping the UART was not too difficult if I followed the examples from the book and the tips from the assignment page. The datasheet made things simpler as well.

01/19/2021

- Today I will be setting up my hardware station
    - Emulator
    - B3
    - Push button circuitry and LCD on a breadboard

01/20/2021

- I am now writing the code using the low level algorithm and the timer program from the book as guidance pages 241-247
- I also run through the tutorial to familiarize myself with the IDE again
- I did a quick debug session of the program
    - Using hardware breakpoints I noticed the program would not pass the block of that clears bit 7 in the CPSR

01/21/2021

- I realized that I forgot to edit the startup_ARMCA8.s file for the INT_DIRECT
    - The program was entering a dead loop
    - After adding “b INT_DIRECT” the program still didn’t leave the dead loop
    - After some thinking/reading/comparing I found I missed “.extern INT_DIRECT”
- Another issue that came up was about my .data section
    - I didn’t realize that I needed two “.align 2” statements
        - One for the message
        - One for the stacks
- From here on out the program worked as intended

01/26/2021

- Now I am taking the high level algorithm written for the first program and updating it for a timer
    - I will be using timer7 since that is the timer I used in ECE371

01/26/2021

- I am now building the low level algorithm using the timer program as reference
    - This wasn’t too hard since I had done it before
    - Much of the register mapping was already done
    - The main difference would be that the interrupt director has more than two sources of interrupts to check for and count registers have a different value
    - I had to think of a way to organize the interrupts so that they happen in the correct sequence (i.e button, UART2, timer/UART2). UART should still be checked first since it will happen multiple times within a second.
    - I had to do a lot of restructuring. I could have simply just shifted the name and reprinted it once it was off screen, however, that did not look like a loop to me. So that is why I wanted to restructure major parts of the program
    - Initially, I thought that all I had to do was turn off UART2 and then simply utilize

the timer to update the THR register. However, because the LCD needs multiple writes to shift the screen that idea failed. Cannot add a delay, that would defeat the purpose of interrupts.

- For the two reasons above, I believe that I made part 2 harder than it needed to be. However, I believe the result/programming was better structured than the first part.

01/28/2021

- I am now updating the code to implement a timer for rotating the name.
- Debugging starts now as well.
    - This had lots of problems
        - Debugging issue from ECE371 using timers, where I had to restart the debugging for it to work properly
        - The home cursor for the LCD did not work as intended. This lead to writing a new program…
        - Due to the reason above, I had to think of several ideas for the project to work. However, I had to go back to my original idea of just shifting the name as it was. This still doesn't look the best, but it will have to do for now.

02/04/21

- I am finalizing all documentation and making sure everything is complete
    - Logging in this document
    - Commenting code
    - Organizing high and low level algorithms