

ECE 252 Fall 2024

HW11B

Before Doing Anything Else...

1. Create a new directory called HW11 under your PennSim directory.
2. Download **HW11B.zip** from canvas. It will be the starting point for your programming assignment.
3. Unzip the .zip file to extract the files inside. If you need help with this, click on the appropriate link: [Windows 10](#), [Mac](#) and select the version of your Mac OS).

Please read the entire remainder of this assignment before starting any of the work described below! You will need to assemble and test (and probably debug) your code in PennSim, so be sure to budget enough time for this!

This is a big assignment, it will probably take you a lot longer than you think! There will be typos in your code that are hard to find. You will likely make mistakes in logic that are hard to track down. Start ASAP!

All code submitted for this assignment is expected to assemble, function correctly, and be well-commented. Please refer to the course's Programming Style Guide.

Be sure to follow the directions below! Failing to follow directions will result in a significant penalty.

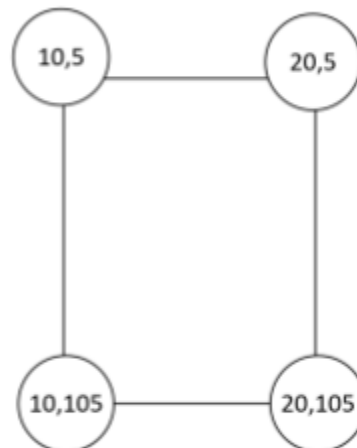
Assignment Purpose

The purpose of this assignment is to give you practice with and assess your ability to write subroutines that not only function correctly but also are written using good design practices and style.

Assignment Description

You will write a subroutine ***draw_rectangle*** that will draw a rectangle on the PennSim LCD display. The shape and color of the rectangle will be defined by values read in from 5 consecutive memory locations. The example below would be used to draw a red rectangle that has the coordinates (10,5), (10,105), (20,5), and (20,105).

```
RT1      .FILL x7C00 ; Color
          .FILL 5     ; Row
          .FILL 10    ; Col
          .FILL 10    ; Width
          .FILL 100   ; Height
```



Writing and Testing Your Code

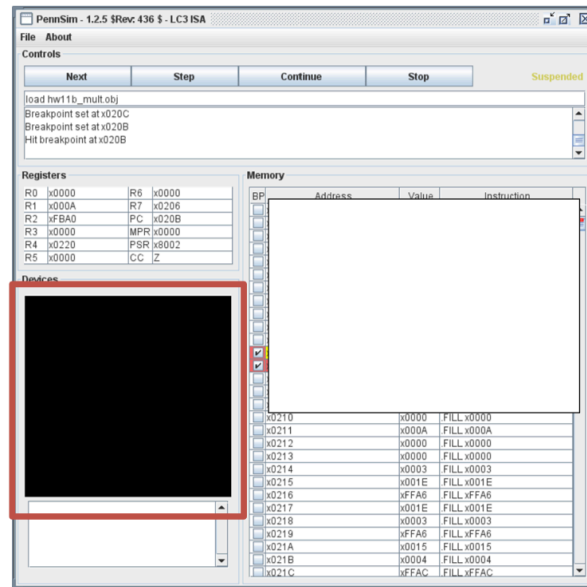
Subroutine *MULT*

Open the file **hw11b_mult.asm** that you extracted from the **HW11B.zip** file. Carry out the following steps:

1. Near the bottom of the file, read the description of the **MULT** subroutine.
2. Starting at the top of the file, *read* the provided code and understand what the test bench is going to do.
3. Add your name and your partner's name in the comments before the subroutine. If you do not have a partner, leave the second name blank.
4. Write your subroutine in the indicated place near the bottom of the file.
5. Assemble the **.asm** file. If assembly results in errors, fix your code and try again. **If you need help, consult with an instructor.**
6. Load the object file into PennSim. Verify that the **.sym** file is also loaded (which it should be automatically).
7. ***In the memory area of PennSim, scroll until the SUCCESS label is at the top.***
8. **Set two breakpoints:** One at the **SUCCESS** label and one at the **ERROR** label.
9. Click "Continue".
10. If the program stops at **SUCCESS** *and is highlighted in yellow*, take a screenshot of the PennSim window when it is side by side with your **hw11b_mult.asm** file. Your **.asm** file should display the top of the **MULT** subroutine. Save it to upload later. You can see an example of what the screen capture should look like at the end of this document.
11. If the program stops at **ERROR** (or doesn't reach either breakpoint), you will have to debug your code. To do this, set the PC to **x0200**, and set a breakpoint at the start of your subroutine. Then "Step" through each instruction to see where the code doesn't do what you expect. **If you need help, consult with an instructor.**

PennSim LCD

PennSim supports a virtual 128x124 pixel LCD shown in the red box below.



Each of the pixels in the LCD are arranged by row and column. You can access the address of an individual pixel using the following formula: **$addr = xC000 + row * x0080 + col$**

For example, if you are writing a value to pixel (3, 10), you would access memory location xC18A

$$xC000 + 3 * x80 + 10 = xC18A$$

The 16-bit value that you write to a memory location will determine the color of the pixel. The color of a given pixel is determined using the following bits:

[15]	[14:10]	[9:5]	[4:0]
0	RED	GREEN	BLUE

Here are a few common colors:

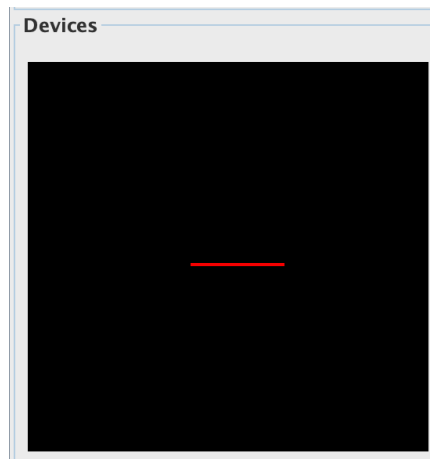
Pixel Code	Color
x7FFF	White
x0000	Black
x7C00	Red
x03E0	Green
x001F	Blue

Subroutine ***DRAW_ROW***

Open the file **hw11b_draw_row.asm** that you extracted from the **HW11B.zip** file. Carry out the following steps:

12. Near the bottom of the file, replace the empty MULT subroutine with the subroutine you completed for MULT above.
13. Starting at the top of the file, *read* the provided code and understand what the test bench is going to do.
14. Add your name and your partner's name in the comments before the subroutine. If you do not have a partner, leave the second name blank.
15. Write DRAW_ROW subroutine in the indicated place.
16. Assemble the .asm file. If assembly results in errors, fix your code and try again. If you need help, consult with an instructor.
17. Load the object file into PennSim. Verify that the .sym file is also loaded (which it should be automatically).
18. In the memory area of PennSim, scroll until the HW11B_END label is at the top.
19. Set a breakpoint at HW11B_END
20. Click "Continue".

21. When the program stops, you should see a red line near the center of the LCD screen. If you see the red line as shown below, take a screenshot of the PennSim window when it is side by side with your hw11b_draw_row.asm file. Your .asm file should display the top of the DRAW_ROW subroutine. Save it to upload later. You can see an example of what the screen capture should look like at the end of this document.



22. If you do not see the red line, you will have to debug your code. To do this, set the PC to x0200, and set a breakpoint at the start of your subroutine. Then “Step” through each instruction to see where the code doesn’t do what you expect. If you need help, consult with an instructor.

Main Program and Subroutine *DRAW_RECTANGLE*

Open the file **hw11b.asm** that you extracted from the **HW11B.zip** file. The main program will test the last subroutine you need to write, **DRAW_RECTANGLE**. Now that your subroutines **MULT** and **DRAW_ROW** are both working correctly, paste them in the indicated locations in **hw11b.asm**.

DRAW_RECTANGLE will accept a single parameter. That parameter is a memory address that contains information needed to draw the rectangle. You will load 5 words of data starting at this address.

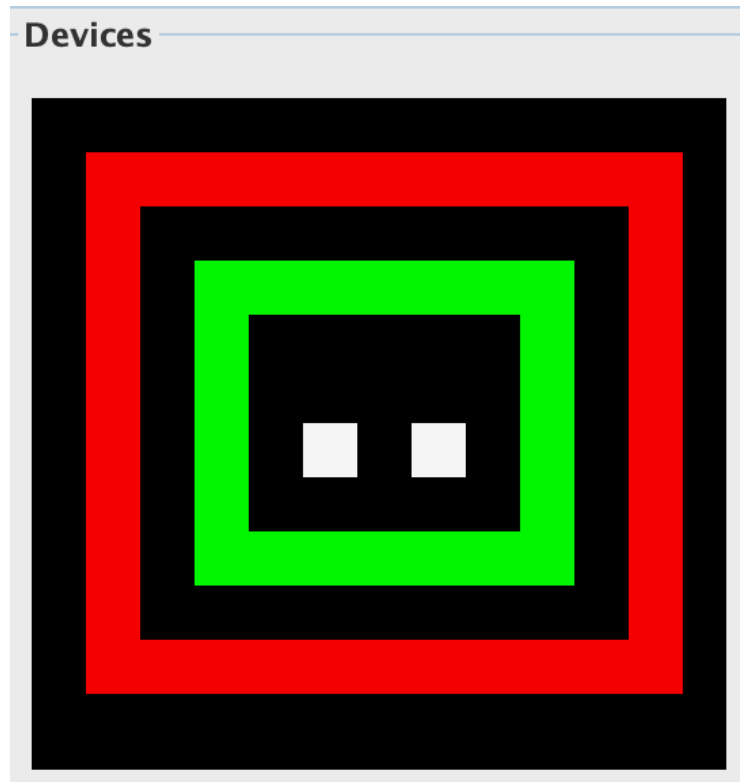
```
RT1          .FILL x7C00 ; Color
              .FILL 5     ; Row
              .FILL 10    ; Col
              .FILL 10    ; Width
              .FILL 100   ; Height
```

If any of the rectangle parameters are invalid, do not draw the rectangle. You can assume that all of the parameters are unsigned numbers. The table below summarizes the invalid parameters or combinations of parameters

Invalid Parameters to Check For
Width == 0
Height == 0
(Row + Height) > 127
(Col + Width) > 123

You will use the functions MULT and DRAW_ROW to draw the rectangle.

23. Near the bottom of the file, replace the empty MULT subroutine with the subroutine you completed for MULT.
24. Near the bottom of the file, replace the empty DRAW_ROW subroutine with the subroutine you completed for DRAW_ROW.
25. Starting at the top of the file, *read* the provided code and understand what the test bench is going to do.
26. Add your name and your partner's name in the comments before the subroutine. If you do not have a partner, leave the second name blank.
27. Write DRAW_RECTANGLE subroutine in the indicated place.
28. Assemble the .asm file. If assembly results in errors, fix your code and try again. If you need help, consult with an instructor.
29. Load the object file into PennSim. Verify that the .sym file is also loaded (which it should be automatically).
30. In the memory area of PennSim, scroll until the HW11B_END label is at the top.
31. Set a breakpoint at HW11B_END
32. Click "Continue".
33. When the program stops, you should see the following image. If you see the image shown below, take a screenshot of the PennSim window when it is side by side with your hw11b.asm file. Your .asm file should display the top of the DRAW_RECTANGLE subroutine. Save it to upload later. You can see an example of what the screen capture should look like at the end of this document.



34. If you do not see this image, you will have to debug your code. To do this, set the PC to x0200, and set a breakpoint at the start of your subroutine. Then "Step" through each instruction to see where the code doesn't do what you expect. If you need help, consult with an instructor.

Your subroutines should not **appear** to modify any of the caller's registers except R7 of course, and any registers used to return a result to the caller.

- Do not add any other subroutines to the code.
- **Be sure to follow the Programming Style Guide (especially the rules about subroutines). For example:**
 - Your subroutine must only contain a single **RET** instruction.
 - Any labels you use must conform to the style guide.
 - Any register saves must be at the very start of the subroutine, and any restores must be just before the **RET**.
 - Register save/restore space and any other data for a subroutine must be located just after that subroutine's **RET**.
 - Your code must be commented appropriately. **In fact, you should be writing comments based on your flowchart before you write the code!**

Submitting Your Assignment

Please follow these directions carefully.

Upload Instructions:

- Upload your PennSim **screenshot (.jpg or .png only)** of hw11.asm after it runs correctly.
- Upload your PennSim **screenshot (.jpg or .png only)** of hw11_mult.asm after it runs correctly.
- Upload your PennSim **screenshot (.jpg or .png only)** of hw11_draw_row.asm after it runs correctly.
- Make a copy of your hw11.asm and name it hw11.txt
- Upload your **text file** hw11.txt
- **VERY IMPORTANT:** After you submit the homework, verify that your code AND screenshots are still attached.

To receive any credit for this programming assignment:

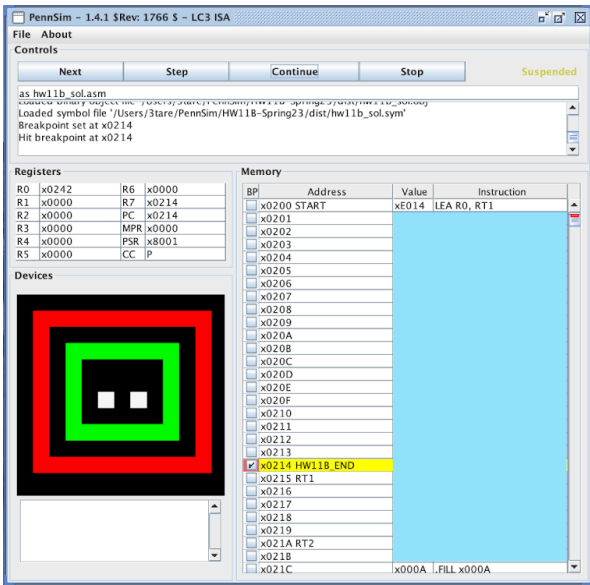
- Your code must assemble in PennSim without errors.
- Your code must be commented according to the course Programming Style Guide.
- You must attach all of the required files.

If you do not meet all of these requirements, you will receive a 0 for your program.

If you are unable to get the program to work correctly by the deadline, but would like it evaluated for partial credit:

- **You must still include screenshots** at the point of execution described above that shows what your code does. This is to demonstrate that you did, in fact, attempt to run it in PennSim.
- **Add a comment** to the top of your code explaining what parts do and do not work correctly, and how you know that to be the case.

Example Screen Capture



```
105 ; Author(s):  NAME
106 ;           NAME
107 ;*****
108 DRAW_RECTANGLE
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
```