**Wenzhou-Kean University Project Fall 2021**

**CPS 2390 W\_\_ Computer Organization & Architecture   
Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Part-1**

In this project you will design an LED scrolling display board using PennSim Simulator. Your task is to:

1. Complete the code that displays letters ‘A’. (The code for displaying ‘A’ is given to you)
2. Using bitmap and code of ‘A’ as template, write the code for displaying ‘B’. Also, display both ‘A’ and ‘B’ on screen.
3. Make the letters ‘A’ and ‘B’ scroll on the display.

The characters are displayed on the screen. The entire screen is divided into a coordinate system with top-left corner (0,0), top-right corner (32,0), bottom-left corner (0,31) and bottom-right corner (32,31). (as shown below). i.e. there are in total 32 rows and 33 columns.

These characters are stored in the form of ‘bitmaps’ in memory.

Information regarding Bitmaps:

Bitmap is a ‘map’ of the character that you want to display. Each bit-map is 7x7 blocks in size (where each block is one particular (x,y) point on the screen). Consider the bitmap of character ‘A’ which has already been provided to you:

0 1 1 1 1 1 0

1 0 0 0 0 0 1

1 0 0 0 0 0 1

1 1 1 1 1 1 1

1 0 0 0 0 0 1

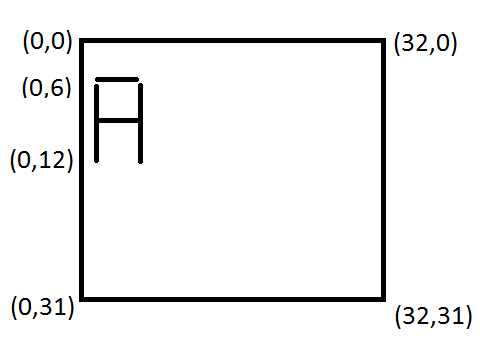
1 0 0 0 0 0 1

1 0 0 0 0 0 1

As you can see ‘1’ and ‘0’s are stored as per the structure of character ‘A’. The bitmap value for each block is placed row-wise and starting address is given in the template code. So, bitmap of A in memory would look like this:

0 , 1 , 1 , 1 , 1 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 1, ……

The following figure shows the layout of coordinate system on the screen (from (0,0) to (32,31)). The character ‘A’ is displayed starting at coordinates (1,6).



To display each block, you will use instruction “TRAP 0x0040” which takes in x and y coordinates as inputs. X-coordinate should be stored in Register R0 and Y-coordinate in Register R1 before calling the TRAP.

Details on the tasks to be completed:

1. Complete code for displaying ‘A’:

The COLUMN\_LOOP and ROW\_LOOP will traverse through the bitmap values. COLUMN\_LOOP increments x-coordinate by 1 and ROW\_LOOP increments y-coordinate by 1. The value for x-coordinate should be reset to its start after COLUMN\_LOOP completes. The ROW\_LOOP has already been provided for displaying char ‘A’. You need to complete the COLUMN\_LOOP. COLUMN\_LOOP would read the bitmap value of that particular coordinate and then call TRAP to draw the block if value read was 1. The bitmap for character ‘A’ is provided in file hw6\_bitmap.asm and template code is provided in hw6.asm. Only rows 7-13 (i.e. y-coordinate from 6 till 12) should be used to display the characters, as shown in the figures above, since that area is only being cleared off before next iteration of next loop.

1. Display ‘A’ and ‘B’:

Using the code for displaying ‘A’ as template, design the bitmap for ‘B’ and then write code for displaying both ‘A’ and ‘B’ (with 1-block space in between) on the screen. You should add the bitmap of B in the same file where bitmap for A is defined and you should use the starting address of the bitmap of B in your main code for displaying it on screen. You can look for how it is done for character ‘A’ as hint.

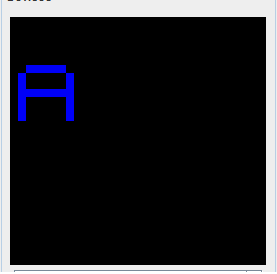
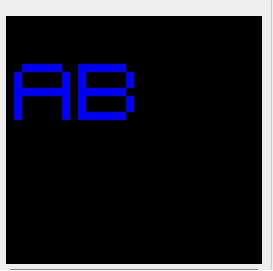
1. Scroll left for 10 blocks:

You need to complete parts of SCROLL\_LOOP for scrolling. You need to scroll 10 blocks to the left (i.e. decrease the starting x coordinate by 1 each time for 10 times – tracked using Global\_X or Register R2 in the code). The key point here is to have Global Start coordinate which has initial value of 1 and can be decremented by 1 each time scroll loop runs. You don’t have to worry about coordinates having negative values since TRAP for drawing block can take in negative arguments as well. Block will be displayed only if the arguments are positive and lie in the coordinate system of the screen as shown in the above figure.

All the places where you need to insert the code have been marked as TODO with a comment on what needs to be inserted. Register usage has been provided in the program itself. You should use lc3.asm as your LC3 OS. It contains the necessary modifications to include TRAP for draw\_block.

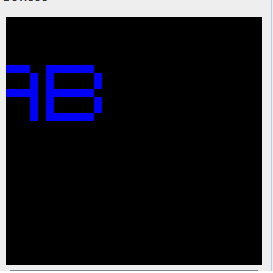
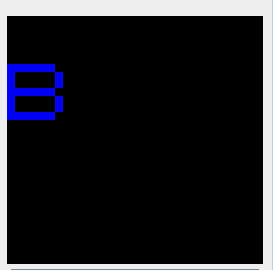
For you reference, the outputs for each of the three tasks should look something like this:

TASK: 1 (Display of letter ‘A’) TASK: 2 (Display of letter ‘A’ and ‘B’)

TASK: 3

(Display of Half-Scrolling) (Display of Full Scrolling)

Turn in your completed source code for this project. Then send it with the (.asm file(s) ) through the Blackboard system. ‘Attach a screen shot of the LED scrolling display you designed along with the codes. Don’t forget to write names of the group members and section number on all the files. Also, only one copy per group should be submitted and you should mention name of that person on the hard copy being submitted.