Your Name: ________________________________ netid:______________ Group #:

Name: ____________________________________ netid:______________

Name: ____________________________________ netid:______________

Name: ____________________________________ netid:______________

ECE 120 Worksheet 13: Programming in LC-3

In this discussion, you will develop a systematic decomposition for Conway's "Game of Life" and implement parts of the code in LC-3 assembly language.

Game of Life

The "Life" board that we will use is a 16x16 board (256 total cells) that is stored in row-major order starting at memory address x5000. Row-major order implies that the game board data for first row is stored first, followed by the second row, and so on. Each memory location represents one cell where an x0 represents a dead cell and an x1 represents a live cell. Other states shall not exist (x0002-xFFFF) and you are not required to handle them.

Any two cells are neighbors if they share a border, and each cell has eight neighbors. For example, the live cell that is highlighted with an arrow in the picture below has eight live neighbors. Each cell has an (X,Y) coordinate where the top left corner of the board has coordinate (0,0).

```
(0,0)                          (15,0)
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  #  #  #  .  .  .  .  .  .  .  .  .  .
.  .  .  #  #  #  .  .  .  .  .  .  .  .  .  .
.  .  .  #  #  #  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
(0,15)                        (15,15)
Initial State
```

```
(0,0)                          (15,0)
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  #  .  .  .  .  .  .  .  .  .  .  .
.  .  .  #  .  #  .  .  .  .  .  .  .  .  .  .
.  .  #  .  .  .  #  .  .  .  .  .  .  .  .  .
.  .  .  #  .  #  .  .  .  .  .  .  .  .  .  .
.  .  .  .  #  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
(0,15)                        (15,15)
After One Generation
```

When iterating through the "Life" game, each cell **lives** or **dies** depending on the following rules, all of which are shown in the above example:

1. Any live cell with fewer than two live neighbors dies.
2. Any live cell with two or three live neighbors continues to live on.
3. Any live cell with more than three live neighbors dies.
4. Any dead cell with exactly three live neighbors is resurrected.

1. Algorithm

Assuming that the game board is already loaded in memory starting from address x5000, come up with a high-level description in English language how your game should proceed for just one iteration. Describe in clear English the sequential processes (algorithm) you will need to follow to perform *just one iteration of the game*, that is, check all the cells on the game board and decide which cell will die, live, or be resurrected.

2. Algorithm refinement

Convert your algorithm from Part 1 into a high-level flow chart using sequential, iterative, and conditional constructs. Use English statements and math expressions to describe your steps.

3. Flowchart refinement

Note: The following problem will NOT be graded and you do not need to submit it. It is only provided to you because it is a good practice exercise for the final exam.

Trace through your flow chart, determine how many values you need to use, and assign registers to each of these values. Will you need to use any registers for multiple values?

Redraw your flow chart using RTL statements with your desired registers.