# Programming Studio #5

ECE 190

# Programming Studio #5

- Topics this week:
  - LC-3 Assembly
  - I/O
  - Strings
- In Studio Assignment
  - Lookup tables in LC3-b assembler

# Announcements

- Exam 1 – Thurs 2/25 – 7pm-9pm
  - Location: ASL 131 & 150, WOHLR 141
  - Review Session: Mon Feb 22 6-8pm, EL 151
  - HKN Review: Tue Feb 23 6-8pm, MSEB 100
- Conflict Exam - Mon 2/22 6-8pm EL 165

# Exam 1 Topics

- Binary
- Logic
- Memory
- Von Neumann Model
- LC3 ISA
- Systematic Decomposition
- Assembly
- MP1/MP2
- Debugging
- Code Optimization

# LC-3 Assembly

- Write out code in plain language
  - No need to encode into binary
- Symbols make programming easier
  - Use a label to specify an address
  - Specify an immediate in decimal or hex
- What is loaded into R0 in each example?

```
LD R0, #3
VAL0    .FILL x25F3
VAL1    .FILL xF5D6
VAL3    .FILL x0000
VAL4    .FILL x1234
```

```
LD R0, x3
VAL0    .FILL x25F3
VAL1    .FILL xF5D6
VAL3    .FILL x0000
VAL4    .FILL x1234
```

```
LD R0, VAL4
VAL0    .FILL x25F3
VAL1    .FILL xF5D6
VAL3    .FILL x0000
VAL4    .FILL x1234
```

# Example: Multiplication

- The example from last week in Assembly

| Binary |
|---|
| 0011 0000 0000 0000 |
| 0101 010 010 1 00000 |
| 0010 100 0 1111 1110 |
| 0010 101 0 1111 1110 |
| 0001 010 010 0 00 100 |
| 0001 101 101 1 11111 |
| 0000 001 1 1111 1101 |
| 0011 010 0 1111 1011 |
| 1111 0000 0010 0101 |

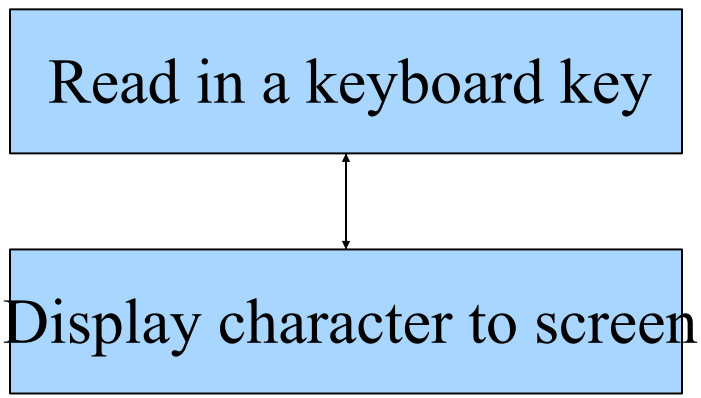
```
Assembly                                Comment

.ORIG x3000
                AND R2, R2, #0          ; Initialize
                LD R4, INPUT0           ; Load INPUT0
                LD R5, INPUT1           ; Load INPUT1
LOOP            ADD R2, R2, R4          ; Add to running sum
                ADD R5, R5, #-1         ; Decrement counter
                BRp LOOP                ; Check if counter == 0
                ST R2, RESULT           ; Store result in RESULT
                TRAP x25                ; Halt
INPUT0          x0003
INPUT1          x000A
RESULT          x0000
.END
```

# I/O

- I/O is used for sending and receiving messages from the user
- Use memory locations to represent display and keyboard
  - Both also contain status registers to indicate when they are ready to write/read

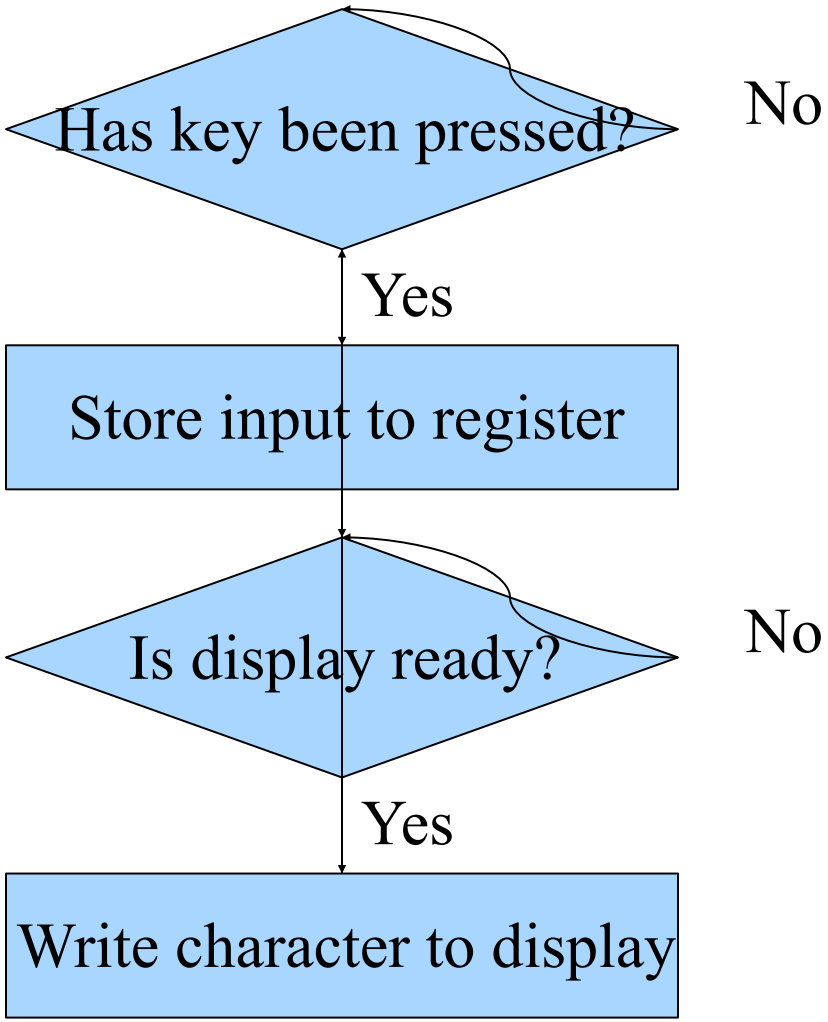
# Example: Echo

- We want to create a program that will display the keyboard key being pressed




- What is missing?

# Echo Decomposition



# Echo Assembly

```
START   LDI R1, KBSR    ; Test for character input
        BRzp START
        LDI R0, KBDR    ; R0 = Keyboard character
ECHO    LDI R1, DSR     ; Test output register ready
        BRzp ECHO
        STI R0, DDR     ; Send R0 to display
        BRnzp START
KBSR    .FILL xFE00     ; Address of KBSR
KBDR    .FILL xFE02     ; Address of KBDR
DSR     .FILL xFE04     ; Address of DSR
DDR     .FILL xFE06     ; Address of DDR
```

- Download file from ECE 190 website
- lc3as echo.asm
- lc3sim-tk echo.obj

# Strings

- A string is a sequence of ASCII characters represented by hex numbers
  - Digits 0-9: x30 – x39
  - Lowercase Letters a-z: x61 – x7A
  - Uppercase Letters A-Z: x41 - x5A
- Each string terminates with a null character (x0000)
- .STRINGZ is used to specify a string

# In-studio Assignment

- Assignment: Write a program to compute f(x), where f is stored in a lookup table
  - Assume 1:1 mapping between lookup table and function entries
  - Input: 16 bit integer x at x4000
  - Output: 16 bit integer f(x) at x4001
  - Fill the lookup table with:

    .STRINGZ "this is function data"

  after your halt instruction: f(x) is defined in range [0, 21)
- Write a system decomposition before coding