# CIS11 Final Review Part 2: Chapter 6 – 19

1. What is an instruction in computing?

The fundamental unit of work

1. Identify the 3 types of instructions.

Computational instructions (ADD, AND,…)

Data movement instructions (LD, ST, ….)

Control instructions (JMP, BRnz, … )

1. Create an assembly program that contains the following criteria:

X is set at location x3200

Y is set at location x3201

X = 4

Y = 3X

Display the value of Y.

.ORIG x3000

LDI R1, X

ADD R1, R1, #4

LDI R2, Y

AND R2, R2, x0

ADD R2, R1, R1 ;R2 = R1 + R1, OR 2X

ADD R2, R1, R1 ; R2 = 3X

STI R2, Y

ADD R0, R2, x0

PUTS

HALT

X .FILL x3200

Y .FILL x3201

.END

1. Create an assembly program using LC-3 that contains the following criteria:

A + 4 > B, given A is at location x3100, B is at location x3101

.ORIG x3000

LDI R1, A

ADD R1, R1, #4

LDI R2, B

NOT R2, R2

ADD R2, R2, #1

COMP

STI R1, A

STI R2, B

HALT

BRp COMP

A .FILL x3100

B .FILL x3101

COMP .FILL x3102

1. Provide the comment for various LC-3 code:
2. ADD R3, R0, x0 ; copying contents from R0 to R3
3. ADD R3, R3, #-16 ;convert values from ASCII to hex by subtracting 48

ADD R3, R3, #-16

ADD R3, R3, #-16

1. BEGIN LEA R0, PROMPT

PUTS

GETC

ADD R1, R0, x0

PROMPT .STRINGZ “Enter first number: “

;LOAD EFFECTIVE ADRESS OF STRING PROMPT “Enter first number”

;Display prompt on console

;obtain character input

;copy input value from R0 to R1

1. ADD R1, R3, x0 ;if R1 = 0, branch to SHOW

BRz SHOW

1. In LC-3, what are opcodes?

Opcodes: reserved symbols that correspond to LC-3 instructions

1. In LC-3, what can be considered operands?

Registers – specified by Rn, where n is the register number

Numbers – indicated by # (decimal) or x (hex)

Label – symbolic name of memory location

1. Specify the purposes of the following opcodes.
2. .BLKW 2

Allocate 2 words of storage (2x16bits)

1. .FILL x3101

Allocate one word, initialize with value x3101

1. .STRINGZ “This is a string “

Allocate 13+1 locations, initialize w/ characters and null terminator

1. Identify the appropriate TRAP vector for the following directives:
2. HALT

TRAP x25

1. OUT

TRAP x21

1. PUTS

TRAP x22

1. IN

TRAP x23

1. GETC

TRAP x20

1. What are the basic constructs of LC-3 programming?

**Sequential:** Instructions naturally flow from one to the next, so no special instruction needed to go from one sequential subtask to the next

**Conditional and Iterative**: create code that converts condition into N, Z, or P

1. What are the steps in debugging?
2. Examine the sequence of instructions being executed
3. Keep track of the results being produced
4. Compare result from each instruction to the expected result
5. Using LC-3 Assembler, what type of files are generated on Windows systems?

.bin (binary), .hex (hex to decimal), .obj (object), .sym (system), .lst (List)

1. In LC-3 programming, what does an object file contain?

Starting addresses, machine instructions.

1. Describe loading and Linking in LC-3 programming.

**Loading** is the process of copying an executable image into memory

**Linking** is the process of resolving symbols between independent object files

1. What is the purpose of Control/Status Registers?

CPU tells device what to do – write to control register

CPU checks whether task is done – read status register

1. Describe “Polling” in ISA I/O management.

CPU keeps checking status register until new data arrives OR device ready for next data “Are we there yet?”

1. Describe “Interrupts” in ISA I/O management.

Device sends a special signal to CPU when new data arrives or device ready for next data CPU can be performing other tasks instead of polling device

1. What are the memory address assignments for Keyboard data register (KBDR) and keyboard status register (KBSR)?

KBDR is assigned to xFE02; KBSR is assigned to xFE00

1. Describe a “stack.”

An abstract data type is a storage mechanism that is defined by the operations performed on it and not at all by the specific manner in which it is implemented

1. Identify the operation to add or remove an item onto a stack.

PUSH: add an item to the stack

POP : remove an item from the stack

1. Describe the Supervisor Stack.

A special region of memory used as the stack for interrupt service routines. LC-3 saves this state information on a special stack, called the Supervisor Stack that is used only by programs that execute in privileged mode.

1. What does the Activation Record Bookkeeping contain?

Return value: space for value returned by function allocated even if function doesn’t return a value

Return address: save pointer to next instruction in calling function, convenient location to store R7 in case another function (JSR) is called

Dynamic link: caller’s frame pointer used to pop this activation record from the stack

1. When should you save and restore registers?

Its value will be destroyed by service routine, and

We will need to use the value after that action

1. What is a stack pointer?

The stack consists of a sequence of memory locations along with a mechanism, called the stack pointer that keeps track of the top of the stack, that is, the location containing the most recent element pushed

1. What occurs when there are too many items being pushed onto the stack?

Overflow condition occurs

1. What causes an interrupt-driven I/O?

Interrupt is an unscripted subroutine call, triggered by an external event

* External device signals need to be serviced
* Processor saves state and starts service routine
* When done, processor restores state and resumes program

1. Why should you use a stack?

- Limited registers

- convenient calling convention for subroutines

- algorithm naturally expressed using FIFO data structures

1. If you use base address x31A0 as a base address to allocate a word for each element, what is the address of the 10th element of the array?

X31AA

1. If you use base address x31F0 to allocate a double-word for each element, what is the address of the 7th element of the array?

X31FE

1. How can you use pointers to obtain multiple results throughout the program?

Use arguments in results: Pass address of variable where you want result to be stored, which is useful for multiple results

1. What is a recursive function?

A recursive function is one that solves its tasks by calling itself on smaller pieces of data

1. Provide example of recursive function in programming.

Running sum, Fibonacci numbers, binary search, …

1. In C programming, what is the purpose of using a data structure?

Group related items together

Organize these data bundles in a way that is convenient to program and efficient to execute

An array is one kind of data structure

1. In C programming, what is a struct?

A mechanism for grouping together related data items of different types

1. What is a linked list in C programming?

A linked list is an ordered collection of nodes, each of which contains some data, connected using pointers

1. What are the advantages of using a linked list in C programming?

Dynamic size

Easy to add additional nodes as needed

Easy to add or remove nodes from the middle of the list