CS270 Recitation 7
LC3 Microarchitecture Simulation

Goals

To visualize the execution of LC3 instructions using a microarchitecture-level Visualizer LC3uArch. You will load assembled LC3 object code into the LC3uArch tool that demonstrated simulation at the microarchitecture level.

Note that when a computing process is viewed at a specific level, the details of the lower level are hidden from the view. In this recitation, you will actually visualize how the information flows. If an assembly language programmer (perhaps a compiler writer) want to optimize the code, they would need to track how many clock cycle the execution would take.

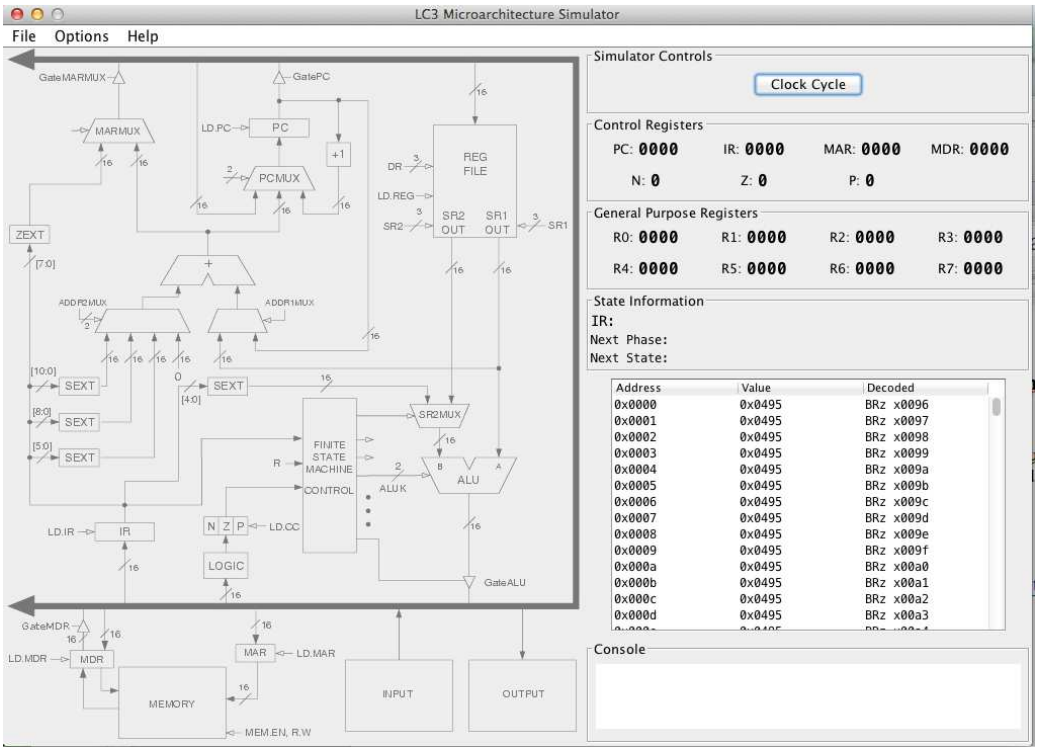
Recall that the for the six phases of instruction execution,

1. fetch instruction
2. decode instruction
3. evaluate address (address generation)
4. fetch operands (read memory data)
5. execute (ALU access)
6. store result (write-back memory data)

some of them are not needed for some instruction. LC3uArch displays the current phase (or the next phase at the end of the clock period) in the **State Information box**.

Tools needed:

1. LC3 Tools assembler to generate the object code.
2. LC3uArch tool, which is an executable JAR file. The toll is similar to the simulator in the LC3 Tools, however it will provide animation that shows the flow of information.



The Assignment

Make a subdirectory called R7 for the recitation assignment, all files should reside in this subdirectory. Download the LC3uArch software (jar file) from http://sourceforge.net/projects/lc3uarch, and put it in your R5 directory.

1. Download or create the file sumlist.asm (below)

```
.ORIG 0x3000
      LEA R1,DATA       ;R1 is pointer
      AND R3,R3,#0      ;R3 is accumulator
      LD R0, Num        ;R0 has count
LOOP  BRz  DONE
      LDR R4,R1,#0      ;load into R4
      ADD R3,R3,R4
      ADD R1,R1,#1
      ADD R0,R0,#-1
      BRnzp LOOP
DONE  ST R3, SUM        ;save sum
      HALT
Num   .FILL 3           ;number of data items
DATA  .FILL 4           ;beginning of list
      .FILL -4
      .FILL 7
SUM   .BLKW 1           ;result goes here
      .END
```

2. Read and try to mentally comprehend the program. The rest of the exercise will show hos whte program will actually run. Then assemble the file to generate sumlist.obj.

3. Start LC3uArch. Load sumlist.obj.

4. Observe the program execution for the first six instruction. Note that the Fetch phase and Decode phase are same for all instructions, information flowing from PC-MAR-Memory-MDR-IR and then to the Control FSM. Also note that PC is incremented.

For each of the six instructions, fill the corresponding row of the table below before executing an instruction, then verify that your expectation was right. First few rows are filled as an example.

Instruction	Source 1		Source 2		Destination		Next PC value	Comment
	Address	Value	Address	Value	Address	Value		
LEA		x300C			R1	x300C	x3001	
AND	R3	undefined	immed	0	R3	0	x3002	
LD	x300B	3	-		R0	3	x3003	
BRz	-		-		-		x3004	
LDR	x300C	4	-		R4	4	X3005	
ADD								
ADD								

Convince yourself that the microarchitecture correctly implements the ISA as expected. If not, ask the TA. Show the TA the filled table.