CPEG 324
LAB 1: Calculator ISA
Vinay Vazir & Jason Reynolds

ISA Simulator Implementation and Usage

GETTING STARTED: WRITING AND COMPILING

Using a standard text editor, write your desired code using the instructions and format outlined in the *ISA Documentation*.

Example:

```
lod r0 0111     # Load 7 into r0
lod r1 0110     # Load 6 into r1
add r2 r0 r1    # Add r1 to r0 and store the result in r2
cmp r2 r1 1     # If r2 equals r1, skip the next instruction
dsp r0          # Display r0
dsp r1          # Display r1
```

When your code is completed, save the file with the extension *.txt* in the same directory as the compiler and simulator. Next, open a bash terminal and navigate to the directory that contains the file, compiler, and simulator. To compile the code into binary representation simply enter the following (NOTE: All code was compiled and executed using Ubuntu Subsystem for Windows):

```
user:~$./calcC “yourFile.txt”
```

This will for the above code, this would output the following:

```
Compiling 6 lines
Compiled successfully output is located in yourFile.jv
```

For a more information, you may compile with the verbose option *-v*.

```
user:~$./calcC “yourFile.txt” -v
```

For the example code above the output produced would be the following:

```
Compiling 6 lines
Line 1:
lod r0 0111
Final binary is: 00000111
Line 2:
lod r1 0110
Final binary is: 00010110
Line 3:
add r2 r0 r1
Final binary is: 01100001
Line 4:
```

```
cmp r2 r1 1
Final binary is: 11100101
Line 5:
dsp r0
Final binary is: 11000000
Line 6:
dsp r1
Final binary is: 11010000
```

After, entering either of these commands, you will find that a new file has been created in this directory with the extension “.jv”. The new file will be a single line, binary only, representation of the instructions from your program. You can open the file with any standard text editor.

Example:
0010001000110111000100110100010011000000100101001101000010100101111100000

GETTING STARTED: USING THE SIMULATOR

Now that you have a binary file*, you can use the simulator. To run the simulator, enter the following command in the bash terminal:

```
user:~$./calc “yourFile.jv”
```

For the code above, the output of this command would be as follows:

```
   7 : 00000111
   6 : 00000110
```

For a more information, you may run the simulation with the verbose option *-v*.

```
user:~$./calcC “yourFile.jv” -v
```

For the code above, the output would be as follows:

```
There are 6 instructions in the file yourFile.jv
Reading instructions from file
line#:      |     :Binary:    |     op  r1 r2 r3 imm  extra
---------------+----------------------+--------------------------
ins   0:    |     00000111    |     lod r0 r1 r3 0111
ins   1:    |     00010110    |     lod r1 r1 r2 0110
ins   2:    |     01100001    |     add r2 r0 r1 0001
ins   3:    |     11100101    |     cmp r2 r1 r1 0101 1
ins   4:    |     11000000    |     dsp r0 r0 r0 0000
ins   5:    |     11010000    |     dsp r1 r0 r0 0000
==============Now running instructions==============
PC:  0 ins: lod  r0 r1 r3 0111   | REG: r0=  0 | r1=  0 | r2=  0 | r3=  0
PC:  1 ins: lod  r1 r1 r2 0110   | REG: r0=  7 | r1=  0 | r2=  0 | r3=  0
PC:  2 ins: add  r2 r0 r1 0001   | REG: r0=  7 | r1=  6 | r2=  0 | r3=  0
PC:  3 ins: cmp  r2 r1 r1 0101 1 | REG: r0=  7 | r1=  6 | r2= 13 | r3=  0
PC += 1
PC:  4 ins: dsp  r0 r0 r0 0000   | REG: r0=  7 | r1=  6 | r2= 13 | r3=  0
   7 : 00000111
PC:  5 ins: dsp  r1 r0 r0 0000   | REG: r0=  7 | r1=  6 | r2= 13 | r3=  0
   6 : 00000110
```

---

* If you have read and understood the *ISA Documentation* you may simply write the desired instructions into a simple text editor in their binary format and save the file with the *.jv* extension. When doing this be sure that the binary is all on one line with no characters other than 0 or 1 as the simulator may reject your code.