Instructor: Andy Phelps
TAs: Peter Ohmann, Newsha Ardalani and Jai Menon

Midterm Examination 4
In Class (50 minutes)
Friday, May 6
Weight: 15%

This exam has 12 pages, including a blank page.
You must turn in all pages.
For your convenience, a listing of the ASCII code is given at the end of this booklet, as well as a listing of the LC-3 instruction set.

NO: BOOK(S), NOTE(S), CALCULATORS OF ANY SORT.

LAST NAME: ________________________________________________
FIRST NAME:_________________________________________________
SECTION: ___________________________________________________
ID# ________________________________________________________
<table>
<thead>
<tr>
<th>Question</th>
<th>Maximum Points</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>
1. Assembly Errors (4 Points)

In the following template for an assembly program, circle any labels where the first pass of the assembler would report an error (if any).

.ORIG x3000

Main ...

Loop ...

End ...

Add ...

Sum ...

Or ...

And ...

Loop ...

X10 ...

.END
2. Two-Pass Assembly Process (7 points)
An assembly language LC-3 program is given below:

```assembly
.ORIG x3000
AND R2, R2, #0
AND R3, R3, #0
AND R5, R5, #0
LD R0, M0
LD R4, M1
LOOP BRz DONE
JSR INCR3
ADD R2, R2, R0
JSR DECR4
BR LOOP
DONE ST R2, ANSWER
HALT
INCR3 ADD R3, R3, #1
RET
DECR4 ADD R4, R4, #-1
RET
```

: Storage area for variables below:
ANSWER .BLKW #4
M0 .FILL x0007
M1 .FILL x0010
.END

a) Fill in the symbol table created by the assembler on the first pass of the above program. (4 points)

<table>
<thead>
<tr>
<th>Symbol Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Once the symbol table is created, the assembler then creates a binary version (.obj) of the program. A portion is listed below. Three lines are missing the binary.

```
0101 0100 1010 0000 ; AND R2, R2, #0
0101 0110 1110 0000 ; AND R3, R3, #0
0101 0110 1110 0000 ; AND R5, R5, #0
0010 0000 0001 0000 ; LD R0, M0
0010 0010 0001 0000 ; LD R4, M1
0000 0100 0000 0100 ; BRz DONE

____ ____ ____ ____ ; JSR INCR3
0001 0100 1000 0000 ; ADD R2, R2, R0

____ ____ ____ ____ ; JSR DECR4
0000 1111 1111 1011 ; BR LOOP
0011 0100 0000 0001 ; ST R2, RESULT

____ ____ ____ ____ ; HALT (TRAP x25)
0001 0110 1110 0001 ; ADD R3, R3, #1
1100 0001 1100 0000 ; RET
0001 1001 0011 1111; ADD R4, R4, #-1
1100 0001 1100 0000 ; RET
```

b) For each of the above missing lines, circle the correct binary value: (3 points)

Missing Line #1

1) 0100 1000 0000 1011
2) 0100 1000 0000 0100
3) 0100 1000 0000 0101

Missing Line #2

1) 0100 1000 0000 1101
2) 0100 1000 0000 0100
3) 0100 1000 0000 0101

Missing Line #3

1) 1111 0000 0000 1101
2) 1111 0000 0010 0101
3) 1111 0000 0001 1001
3. I/O in LC-3 (7 Points)

An LC-3 program is provided below:

```
.ORIG x3000
LD R0, ASCII
LD R1, NEG
AGAIN
LDI R2, DSR
BRzp AGAIN
STI R0, DDR
ADD R0, R0, #1
ADD R3, R0, R1
BRn AGAIN
HALT

ASCII .FILL x0043
NEG .FILL xFFB6
DSR .FILL xFE04 ; Address of DSR
DDR .FILL xFE06 ; Address of DDR
.END
```

a) What is the purpose of reading the Display Status Register (DSR)? (2 points)

b) What is the purpose of "STI R0, DDR"? (1 point)

c) What does this program do? (3 points)

d) What is in DSR bit 14, but we are not using in this program? (1 point)
4. Subroutines (7 Points)

In the code below, the Subroutine ONECHAR takes 1 character from the user (keyboard) and saves it into memory. The assembly code uses ONECHAR in a loop 6 times to input 6 characters and save them to memory. Finally, it prints the string to the screen.

```assembly
; CODE TO INPUT AND PRINT 6 CHARACTERS

.ORIG x3000
AND R0, R0, #0 ; Initialize R0, our counter

LOOP
LEA R1, INPSTRING ; R1 now has base of INPSTRING
ADD R1, R1, R0 ; R1 now has base + offset = R0
ST R0, SAVEREG1 ; SAVE R0
JSR ONECHAR ; Call Subroutine
LD R0, SAVEREG1 ; Restore R0
ADD R0, R0, #1 ; Increment R0
LD R1, LENGTH ; Load R1 with minus length
ADD R1, R1, R0 ;
BRp LOOP ; loop till 6 characters are reached
LEA R0, INPSTRING ; Get ready to print
PUS ; TRAP x22: Print string
HALT ; We're done

ONECHAR
ST _____, SAVEREG2 ; SAVE ?? upon entering the subroutine
GETC ; TRAP x20: Get a character from keyboard.
STR R0, R1, #0 ; Save keyboard input (R0 contains input)
LD _____, SAVEREG2 ; Restore ?? before leaving
RET

LENGTH .FILL 0xFFFA ; minus Length (-6)
KBSR .FILL 0xFE00
KBDR .FILL 0xFE02
SAVEREG1 .FILL 0x0
SAVEREG2 .FILL 0x0
INPSTRING .BLKW 7
.END
```

(a) Line 8 saves R0 before calling the subroutine ONECHAR. Briefly explain why this is necessary. (2 points)

(b) What other register needs to be stored and restored inside the subroutine? Fill in lines 19 and 22 above. (2 points)

(c) In the program we save and restore registers two places, in lines 8 and 10 and again in lines 19 and 22. For lines 8 and 10, indicate whether it illustrates Caller-save or Callee-save. (1 point)

(d) How many times will each character appear on the screen? Why? (2 points)
5. General Questions (10 points)

Circle the best answer.

I. A new service routine is defined starting in memory location x3700. After loading a program that calls this subroutine, the user sets memory location x0067 to x3700. Which of the following can be used to call this subroutine?
   a. TRAP x3767
   b. TRAP x0037
   c. TRAP x67
   d. TRAP x3700

II. JSR <label> is equivalent to
   a. LEA R7, <label>  
      JMP R7
   b. LEA R7, #0 [that is, 1110 111 000000000]  
      BRnzp <label>
   c. LEA R7, #1  
      BRnzp <label>
   d. All of the above are equivalent

III. JSRR R3 is equivalent to
   a. LEA R7, #1  
      JMP R3
   b. LEA R3, #0  
      JMP R3
   c. LEA R3, #1  
      JMP R7
   d. LEA R3, #0  
      JMP R7

IV. Into what phase of the control state machine is the logic to test for an interrupt signal usually added?
   a. just after Evaluate Address
   b. just before Execute
   c. just after Store Result
   d. just before Decode
   e. just after Decode

V. Which of the following can be used only once in the program?
   a. .HALT
   b. .END
   c. .FILL
   d. .BLKW
VI. Assembling the instruction ADD R1, R1, #45 causes which of the following errors?
   a. ADD instruction takes only 3 register sources (2 sources + 1 destination)
   b. Immediate value is out of range
   c. R1 is not initialized
   d. The instruction does not cause an error.

VII. How many memory locations are used by the following assembly directive?:
    MYFAVORITE .STRINGZ "Exam 4"
   a. 3
   b. 4
   c. 6
   d. 7
   e. 8

VIII. Which one of the following is correct about "Caller-save"? (circle the correct answer)
   a) It's particularly useful when entering interrupt handlers, to save the user's registers.
   b) It is better than Callee-save, since the caller knows exactly which registers
      the subroutine will overwrite.
   c) It is used primarily in ISAs that have very wide registers.
   d) It might end up saving registers that didn't really need to be saved.

IX. Comments are:
   a) Useful, but only if someone else takes over working on your program.
   b) So important that you should have a comment on every line, e.g.
      ADD R0,R0,R1 ; Adds R1 into R0
   c) Important for the author, as well as for reviewers, maintainers, and users of the software.
   d) Unimportant because they are only for people, not the assembler.

X. If you execute a RET instruction in memory location x4231, what is the possible range of values
   for the PC afterwards? [PC_{inc} indicates PC after it is incremented.]
   a. (PC_{inc} - 2048) -- (PC_{inc} + 2047)
   b. (PC_{inc} - 256) -- (PC_{inc} + 255)
   c. (PC_{inc} - 16) -- (PC_{inc} + 15)
   d. x0000 -- xFFFF
<table>
<thead>
<tr>
<th>Char</th>
<th>Dec</th>
<th>Oct</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Oct</th>
<th>Hex</th>
<th>Char</th>
<th>Dec</th>
<th>Oct</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>(nul)</td>
<td>0 0000 0x00</td>
<td>(sp)</td>
<td>32 0040 0x20</td>
<td>@</td>
<td>64 0100 0x40</td>
<td></td>
<td></td>
<td>96 0160 0x60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(soh)</td>
<td>1 0001 0x01</td>
<td>!</td>
<td>33 0041 0x21</td>
<td>A</td>
<td>65 0101 0x41</td>
<td>a</td>
<td>97 0161 0x61</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(stx)</td>
<td>2 0002 0x02</td>
<td>&quot;</td>
<td>34 0042 0x22</td>
<td>B</td>
<td>66 0102 0x42</td>
<td>b</td>
<td>98 0162 0x62</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(etx)</td>
<td>3 0003 0x03</td>
<td>#</td>
<td>35 0043 0x23</td>
<td>C</td>
<td>67 0103 0x43</td>
<td>c</td>
<td>99 0163 0x63</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(eot)</td>
<td>4 0004 0x04</td>
<td>$</td>
<td>36 0044 0x24</td>
<td>D</td>
<td>68 0104 0x44</td>
<td>d</td>
<td>100 0164 0x64</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(enq)</td>
<td>5 0005 0x05</td>
<td>%</td>
<td>37 0045 0x25</td>
<td>E</td>
<td>69 0105 0x45</td>
<td>e</td>
<td>101 0165 0x65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ack)</td>
<td>6 0006 0x06</td>
<td>&amp;</td>
<td>38 0046 0x26</td>
<td>F</td>
<td>70 0106 0x46</td>
<td>f</td>
<td>102 0166 0x66</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(bel)</td>
<td>7 0007 0x07</td>
<td>'</td>
<td>39 0047 0x27</td>
<td>G</td>
<td>71 0107 0x47</td>
<td>g</td>
<td>103 0167 0x67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(bs)</td>
<td>8 0110 0x08</td>
<td>(</td>
<td>40 0050 0x30</td>
<td>H</td>
<td>72 0110 0x48</td>
<td>h</td>
<td>104 0168 0x68</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ht)</td>
<td>9 0111 0x09</td>
<td>)</td>
<td>41 0051 0x31</td>
<td>I</td>
<td>73 0111 0x49</td>
<td>i</td>
<td>105 0169 0x69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(nl)</td>
<td>10 0112 0x0a</td>
<td>*</td>
<td>42 0052 0x32</td>
<td>J</td>
<td>74 0112 0x4a</td>
<td>j</td>
<td>106 0170 0x6a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(vt)</td>
<td>11 0113 0x0b</td>
<td>+</td>
<td>43 0053 0x33</td>
<td>K</td>
<td>75 0113 0x4b</td>
<td>k</td>
<td>107 0171 0x6b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(np)</td>
<td>12 0114 0x0c</td>
<td>,</td>
<td>44 0054 0x34</td>
<td>L</td>
<td>76 0114 0x4c</td>
<td>l</td>
<td>108 0172 0x6c</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(cr)</td>
<td>13 0115 0x0d</td>
<td>-</td>
<td>45 0055 0x35</td>
<td>M</td>
<td>77 0115 0x4d</td>
<td>m</td>
<td>109 0173 0x6d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(so)</td>
<td>14 0116 0x0e</td>
<td>.</td>
<td>46 0056 0x36</td>
<td>N</td>
<td>78 0116 0x4e</td>
<td>n</td>
<td>110 0174 0x6e</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(si)</td>
<td>15 0117 0x0f</td>
<td>/</td>
<td>47 0057 0x37</td>
<td>O</td>
<td>79 0117 0x4f</td>
<td>o</td>
<td>111 0175 0x6f</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dle)</td>
<td>16 0120 0x10</td>
<td>0</td>
<td>48 0060 0x38</td>
<td>P</td>
<td>80 0120 0x50</td>
<td>p</td>
<td>112 0176 0x70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dc1)</td>
<td>17 0121 0x11</td>
<td>1</td>
<td>49 0061 0x39</td>
<td>Q</td>
<td>81 0121 0x51</td>
<td>q</td>
<td>113 0177 0x71</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dc2)</td>
<td>18 0122 0x12</td>
<td>2</td>
<td>50 0062 0x3a</td>
<td>R</td>
<td>82 0122 0x52</td>
<td>r</td>
<td>114 0178 0x72</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dc3)</td>
<td>19 0123 0x13</td>
<td>3</td>
<td>51 0063 0x3b</td>
<td>S</td>
<td>83 0123 0x53</td>
<td>s</td>
<td>115 0179 0x73</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(dc4)</td>
<td>20 0124 0x14</td>
<td>4</td>
<td>52 0064 0x3c</td>
<td>T</td>
<td>84 0124 0x54</td>
<td>t</td>
<td>116 017a 0x74</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(nak)</td>
<td>21 0125 0x15</td>
<td>5</td>
<td>53 0065 0x3d</td>
<td>U</td>
<td>85 0125 0x55</td>
<td>u</td>
<td>117 017b 0x75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(syn)</td>
<td>22 0126 0x16</td>
<td>6</td>
<td>54 0066 0x3e</td>
<td>V</td>
<td>86 0126 0x56</td>
<td>v</td>
<td>118 017c 0x76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(etb)</td>
<td>23 0127 0x17</td>
<td>7</td>
<td>55 0067 0x3f</td>
<td>W</td>
<td>87 0127 0x57</td>
<td>w</td>
<td>119 017d 0x77</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(can)</td>
<td>24 0130 0x18</td>
<td>8</td>
<td>56 0070 0x38</td>
<td>X</td>
<td>88 0130 0x58</td>
<td>x</td>
<td>120 017e 0x78</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(em)</td>
<td>25 0131 0x19</td>
<td>9</td>
<td>57 0071 0x39</td>
<td>Y</td>
<td>89 0131 0x59</td>
<td>y</td>
<td>121 017f 0x79</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(sub)</td>
<td>26 0132 0x1a</td>
<td>:</td>
<td>58 0072 0x3a</td>
<td>Z</td>
<td>90 0132 0x5a</td>
<td>z</td>
<td>122 0180 0x7a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(asc)</td>
<td>27 0133 0x1b</td>
<td>;</td>
<td>59 0073 0x3b</td>
<td>[</td>
<td>91 0133 0x5b</td>
<td>[</td>
<td>123 0181 0x7b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(fis)</td>
<td>28 0134 0x1c</td>
<td>&lt;</td>
<td>60 0074 0x3c</td>
<td>\</td>
<td>92 0134 0x5c</td>
<td>\</td>
<td>124 0182 0x7c</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(gs)</td>
<td>29 0135 0x1d</td>
<td>=</td>
<td>61 0075 0x3d</td>
<td></td>
<td></td>
<td>93 0135 0x5d</td>
<td></td>
<td></td>
<td>125 0183 0x7d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(rs)</td>
<td>30 0136 0x1e</td>
<td>&gt;</td>
<td>62 0076 0x3e</td>
<td>^</td>
<td>94 0136 0x5e</td>
<td>^</td>
<td>126 0184 0x7e</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(us)</td>
<td>31 0137 0x1f</td>
<td>?</td>
<td>63 0077 0x3f</td>
<td></td>
<td></td>
<td>95 0137 0x5f</td>
<td></td>
<td></td>
<td>(del) 127 0177 0x7f</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>