Electrical and Computer Engineering Department
University of Maryland
College Park, MD 20742-3285

Glenn L. Martin Institute of Technology ♦ A. James Clark School of Engineering

Dr. Charles B. Silio, Jr.
Telephone 301-405-3668
Fax 301-314-9281
silio@umd.edu

ENEE 350 Homework Set No. 6

(Due: Class 15, Wed., June 25, 2014)

1. Read textbook section 5.4. (Also, reread sections 5.5 and 5.6, especially 5.5.7.)
2. Work text problem 5-6.
3. Work text problem 5-25.
4. Print out and read the notes on PDP-11 addressing on the course webpage under the subdirectory Notes. Then work the following problems:

   a. Suppose a DEC PDP-11 instruction word fetched and placed in the IR contains the bit pattern:

   0001100100010001

   Just before this instruction is executed, relevant registers and memory locations contain the following:

$[\text{R1}] = 002000_8$	$[2000_8] = 002002_8$
$[\text{R2}] = 002002_8$	$[2002_8] = 002004_8$
$[\text{R3}] = 002004_8$	$[2004_8] = 002006_8$
$[\text{R4}] = 002006_8$	$[2006_8] = 002010_8$
$[\text{R5}] = 002010_8$	$[2010_8] = 002012_8$
	$[2012_8] = 002000_8$

   Give the (octal) contents, immediately after execution of the instruction in the IR, of registers and any memory locations whose contents could have changed by execution of this instruction.

   b. Suppose instead that the PDP-11 instruction fetched into the IR is given by the bit pattern:

   0110100100010001

   and suppose the contents of registers and memory locations are the same as is given in part a. Now what are the octal contents of registers and memory locations changed by execution of this instruction?

5. Another PDP-11 problem: Suppose the IR contains the following bit pattern:

   0001001001001010

   and suppose that registers and memory locations contain the following:

$[\text{R0}] = 001000_8$	$[1000_8] = 001000_8$
$[\text{R1}] = 002000_8$	$[2000_8] = 004000_8$
$[\text{R2}] = 003000_8$	$[3000_8] = 005000_8$
$[\text{R3}] = 004000_8$	$[4000_8] = 002000_8$
$[\text{R4}] = 005000_8$	$[5000_8] = 000000_8$

   Give the (octal) contents, immediately after execution of the instruction in the IR, of registers and any memory locations whose contents could have changed by execution of this instruction.

6. The DEC PDP-11 instruction word in memory location i is fetched and placed in the IR for decoding and execution; this word contains the bit pattern (shown in binary); also shown is the word that follows it in memory:

   1110100001110011 = [ir] = m[word i]
   0000000000000110 = m[word i+2]

   Just before this instruction is executed, relevant registers and memory locations contain the following (in octal shorthand):

$[\text{R0}] = 002014_8$	$[2000_8] = 002002_8$
$[\text{R1}] = 002002_8$	$[2002_8] = 002004_8$
$[\text{R2}] = 002004_8$	$[2004_8] = 002006_8$
$[\text{R3}] = 002006_8$	$[2006_8] = 002010_8$
$[\text{R4}] = 002010_8$	$[2010_8] = 002012_8$
$[\text{R5}] = 002012_8$	$[2012_8] = 002000_8$
$[\text{R6}] = 002000_8$	$[2014_8] = 164163_8$
$[\text{R7}] = 002016_8$	$[2016_8] = 000006_8$

   Give the (octal) contents, immediately after execution of the above instruction and immediately before start of the fetch cycle for the next instruction, of only those registers and memory locations whose contents have been changed during execution of this instruction.

Continued on page 2.

7. Read Chapter 7 in A. S. Tannenbaum, *Structured Computer Organization*, 5th. ed., Prentice Hall, 2006, and learn how assemblers and linkers work.

8. Programming Assignment 1: Consider the following main program and subroutine for the MAC-1 machine:

```
          /Main
           EXTRN   sum
ans1       RES      2
ans2       RES      1
start      loco    4020
           swap
           stod    ans1
           loco    x1
           push
           lodd    x2
           push
           loco    ans1
           push
           call    sum
           stod    ans2
           halt
x1         -40
x2          19
           END     start
```

```
       /Subroutine sum
            ENTRY  sum
i           EQU     1
j           EQU     2
k           EQU     3
sum         lodl    j
            addl    k
            push
            lodl    j
            addd   (1)
            popi
            lodl    k
            pshi
            pop
            addd  (-23)
            retn
            END
```

Add comments to each line of code in the main program and subroutine explaining what is taking place (and preferrably why); don't simply restate the obvious. Adjust the comments after you analyze the programs (using the simulator) so that they meaningfully describe the code.

Assemble, link, and simulate execution of the above main program and subroutine. Do two different simulations. Turn in printouts of the separate assemblies, linkages, and the snapshots of memory before and after execution in each case. Write up a summary of the differences in the addresses used in the two different absolute programs, and place comments on the symbolic source statements to describe what the program statements are accomplishing. Explain the final contents of locations ans1, ans1+1, and ans2.

Assuming that the main program is in file "main" and that the subroutine is in the file "sum", be sure to do the following:

assem main
qpr -q "your favorite printer" main.list
assem sum
qpr -q "your favorite printer" sum.list
load main sum
qpr -q "your favorite printer" main.abs
sim main.abs $EE350/halt $EE350/halt.pascal

- produce printout of before and after memory contents and highlight and explain the contents of ans1, ans1+1, and ans2.

Also, do the following: **load sum main**
qpr -q "your favorite printer" sum.abs
sim sum.abs $EE350/halt $EE350/halt.pascal

- produce printout of before and after memory contents and highlight and explain the contents of ans1, ans1+1, and ans2.

Note: Be sure to edit the memory dumps to get rid of excess words with all zeros (however, be sure to keep both the code and stack sections of memory from both sets of dumps). Both sets of dumps should fit on one page with judicious editing and use of enscript.