

Memorial University of Newfoundland  
**Engineering 4862      MICROPROCESSORS**  
**Assignment 4**

Instructor: Cheng Li      Due: July 4, 2005

---

Unless otherwise noted, please show all relevant calculations, and explain your answers where appropriate.

0. Write a sequence of instructions (that is, not a full program with TITLE, ASSUME, etc.) that checks whether the memory in a full 64KB segment is functioning correctly. Assume that register AX contains the segment to be checked. Use the string instruction STOSB to load the byte AAH into all 64K locations, and then use LODSB to check that AAH was correctly stored at each of the locations. If the wrong value is read, then call a fictitious subroutine MemError. Otherwise, call Function 4CH of DOS INT 21H to exit.
  
1. Read Intel 8086/88 User Manual page 6-53 to 6-55: understand how the instruction is encoded to and decoded from machine code. Use the two tables that followed (table 6-22 and 6-23); do the following two questions. Convert the following hexadecimal machine codes to assembly language mnemonics. State what each of the byte fields mean (Table 6-23 from page 6-64 to page 6-69).
  - a. B8 00 20
  - b. 8E D8
  - c. 46
  - d. 90
  - e. 89 7C FE
  - f. 75 F7
  - g. E2 EF
  - h. 26 80 07 78
  
2. Convert the following instructions to machine code – give your answers in hexadecimal. State what each of the bit fields mean.
  - i. PUSH BX
  - j. MOV [SI+490], SP
  - k. OUT DX, AL
  - l. POPF
  - m. AND AX, [BX+DI+2Dh]
  - n. ADD DS:[BP], DX      Note: you will have to 'add' a displacement
  - o. XOR AL, [BX+DI-36H]
  - p. MOV [DI+476], ES
  
3. The following bytes are found in order somewhere in memory. Assuming they are machine codes, decode the values into meaningful assembly language mnemonics.  
B9 00 12 D0 C0 E8 C8 E2 F9
  
4. Use full segment definition, write a DOS compatible program that: a) clears the screen, b) set the cursor to screen position row = 10 and column = 5, c) displays the prompt "Please enter an 8-digit number: ", d) get the keyboard input and save the number to a buffer area in the memory (you define), e) sort the number on its ascending order and save them to another buffer for display. For example, if the input number is 29034765, then after your sort, the result should be 02345679. You can assume that the number for each digit is non-repeat but actually the repeated case is the same, f) after your sort, change to the start of next new line, output "The sorted number is: " and the number, g) exit use DOS function 4CH. Write task a) and b) using subroutines. Test your code on PC by yourself.