Memorial University of Newfoundland

Engineering 4862 MICROPROCESSORS

Assignment 2

Instructor: Cheng Li Due: May 31, 2005

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

0. List all 8086/8088 registers that can be accessed as both words and bytes.

1. Use a memory map to show the contents of memory locations DS: 1000H to DS: 1004H after all of the following instructions have executed:

```
MOV AX, 56H
MOV [1001H], AX
MOV [1003H], 9A5FH
MOV [1000H], AL
```

2. Assume that (CS) = B795H; (DS) = 2000H; (SS) = 0AD4H; (ES) = 30FFH; (SP) = 00FFH; (BP) = 1DF7H; (AX) = 0B24H; (CX) = 1EE4H; (SI) = 3C00H; (DX) = 329FH.

a. Calculate the beginning and ending addresses for all of the segments.

b. Suppose that the offset address for the next instruction to be fetched (that is, the contents of IP) is 902DH. Calculate the physical address from where the next instruction will be fetched.

c. What will be the contents of BL and AX after the following instruction is executed? Give your results in decimal and hexadecimal.

```
MOV BL, AL
```

d. Calculate the physical addresses of the memory locations referred to in the following instructions and the contents of all the location(s):

```
AND CX, [1200H]
OR ES:[0E8C9H], SI
PUSH DX
```

3. What is wrong with, or missing from, each of the following instructions:

a. MOV ES, 249EH

b. MOV [BX+3EH], 2 (Hint: the use of PTR directive)

c. MOV [78H], [79H]

4. In lab 1, we found that the MUN-88 single board computer has a number of mirror images. Without replacing the 3x8 decoder or the current 2KB SRAM chip, propose a simple scheme to eliminate these mirror images for the RAM and draw a sketch to show that. (Hint: The mirror images are caused by the don't care lines of the address bus. You can use those lines connect to chip select, enable pins of the decoder or SRAM chips).

5. Instruction XCHG achieves a swap between the source operand and the destination operand. (Consult the 8086/88 user manual for more detailed information). Assume that the instruction XCHG does not exist in the 8086 instruction set. Write a sequence of instructions to duplicate the instruction XCHG AL, DL. Note that the values in **all other** registers (including AH and DH) should be their original values when your instruction sequence finishes.

6. Can the 8-bit input port at location 911 be accessed using direct port I/O addressing? Give an instruction sequence to copy the data from this port to register CL.

7. When a CALL is executed, how does the CPU know where to return? What is the difference between a FAR call and a NEAR call?

8. Find the contents of the stack and stack pointer after the execution of the CALL instruction shown next: Assume that SS:1296 right before the execution of CALL and SUM is a NEAR procedure.

```
  CS:IP
2450:673A     CALL SUM
2450:673D     DEC AH
```

9. Translate one of the following two quotes to 8-bit ASCII format (ignore the names and dates) – (Text book: Section 3.4):

a. “640K ought to be enough for anybody.” - Bill Gates, 1981
b. “I think there is a world market for about 5 computers.”
– Thomas J. Watson, founder of IBM, 1943

10. Write your MUN student number and convert to its unpacked BCD binary equivalent (Text book: Section 3.4).

11. Find the precise **offset** location in memory of each ASCII character or data in the following use a memory map (Text book: Section 3.4 for ASCII numbers):

```
        ORG 20H
Data1   DB      73H, 2FH
Data2   DB      “737 3527”
        ORG 30H
Data3   DW      2560H, 101100010101B
        ORG 40H
Data4   DD      25684FC4H
Data5   DQ      7F5EC4527271FEH
```

12. It is common practice to save all registers at the beginning of a subroutine. Assume that SP=1288H before a subroutine CALL. Show the contents of the stack pointer and the exact memory contents of the stack after PUSHF, for the following:

```
1132:0450     CALL PROC1
1132:0453     INC BX
  …….         ……

PROC1         PROC
              PUSH AX
              PUSH BX
              PUSH CX
              PUSH DX
              PUSH SI
              PUSH DI
              …
              …
PROC1         ENDP
```

13. The following program adds four words and saves the result. The program contains some errors, fix the errors and make the program run correctly:

```
TITLE         PROBLEM PROGRAM
PAGE          60, 132
STSEG         SEGMENT
              DB 32 DUP(?)
STSEG         END
;--------------------------------------------------
DTSEG         SEGMENT
DATA          DW      1234H, 3344H, 5FE2H, 85FAH
              ORG 10H
SUM           DW      ?
DTSG          ENDS
```

```
;--------------------------------------------------
CDSEG           SEGMENT
START:          PROC FAR
                ASSUME          CS:CDSEG, DS:DTSEG, SS:STSEG
                MOV DS, DTSEG
                MOV CS, 4
                MOV BX, 0
                MOV DI, OFFSET DATA
LOOP1           ADD BX, [DI]
                INC DI
                DEX BX
                JNZ     LOOP1
                MOV SI, OFFSET RESULT
                MOV [SI], BX
                MOV AH, 4CH
                INT 21H
CDSEG:          ENDS
START           ENDP
                END CDSEG
```

14. Write an Assembly Language Program that summarize eight unsigned byte numbers stored in memory and store the result back to the next memory location. The data segment can be defined as following:

```
DTSEG                   SEGMENT
        Data            DB      23H, 34H, 32H, 45H, 1FH, 27H, 7FH, 90H
        Result          DW      ?
DTSEG                   ENDS
```

(Hint: 1. Use loop to write an efficient code. 2. Pay attention to how to handle the carry bit. You can use a 16-bit register to hold the result, and then you don’t need to worry about the carry)