Homework 10

See lecture notes for parallel port discussion - see next page for figure 8-10

8.5 Refer to the printer control program in Figure 8–10. Assume bytes 0–5 of the Print_Data segment store 00 30 00 02 10 E0, and answer the following questions about this program:

(a) How many bytes will the program output to the printer?

(b) Which instruction is used to fetch these bytes from memory?

(c) Which instruction outputs these bytes to the printer?

(d) What is the *segment* address of the memory location where these bytes are stored?

(e) What is the *offset* address of the memory location where these bytes are stored?

8.6 The bit assignments for an industrial control system are shown in the table below. Write a polling routine that tests each assigned bit and, if active, transfers control to a routine with the same bit name. For example, if $\overline{AH0}$ is low, transfer control to a routine name AH0. Write your routine so that the highest numbered bits are given priority (that is, tested first). *x* indicates that that bit is unassigned. Assume the I/O mapped port address is E000H. Be sure to add meaningful comments to each program line.

Status Port Bit Assignments

Bit number	7	6	5	4	3	2	1	0
Bit function	*x*	$\overline{SPL}$	*x*	NPL	*x*	AH2	$\overline{AH1}$	$\overline{AH0}$

8.7 Using the bit and port assignments in Problem 8.6, write a polling program that transfers control to location READY if NPL is inactive *and* AH2 *or* $\overline{AH0}$ is active. If the condition is not met, continue to poll. Be sure to add meaningful comments to each program line.

Figure 8-10 for problem 8.5

```
                        ;Function:  Polled printer driver for LPT1.
                        ;           Written as a far procedure.

                        ;Inputs:    PRINT_DATA segment holds number of bytes
                        ;           to be printed and the address of the buffer.
                        ;Outputs:   Characters in the buffer are output to LPT1.
                        ;Calls:     None
                        ;Destroys:  AX, CX, SI, DS, flags


0000                    PRINT_DATA      SEGMENT WORD
0000 0000                       NUMB    DW  ?       ;Number of bytes to print
0002 00000000                   ADR     DD  ?       ;Address of first byte

0006                    PRINT_DATA ENDS

= 0378                  LPT1    EQU     378H        ;Printer data port
= 0379                  STATUS  EQU     379H        ;Printer status port
= 037A                  Control EQU     37AH        ;Printer control port
= 000A                  INIT    EQU     0AH         ;Unidirectional, no IRQ,
                                                    ;select printer, init, auto
                                                    ;STROBE=1
= 000E                  S_HIGH  EQU     0EH         ;STROBE=1 and no init
= 000F                  S_LOW    EQU    0FH         ;STROBE=0 and no init
```

(a)

(Continued on next page)

Figure 8–10.
Control program for the parallel printer interface in Figure 8–7.

```
0000                    CODE    SEGMENT 'CODE'
0000                    FIG8_10 PROC    FAR
                                ASSUME  CS:CODE, DS:PRINT_DATA


                        ;Initialize pointers:
                        ;       DS:SI to start of data
                        ;       CX with number of bytes to be printed

0000  B8 ---- R                 MOV     AX,PRINT_DATA   ;Load DS with
0003  8E D8                     MOV     DS,AX           ;address of PRINT_DATA.
0005  8B 0E 0000 R              MOV     CX,NUMB         ;Get number of bytes
0009  C5 36 0002 R              LDS     SI,ADR          ;Get address of data
                                                        ;to DS:SI
000D  FC                        CLD                     ;Auto increment


                        ;Initialize and select printer, auto line feed, STROBE=1
000E  B0 0A                     MOV     AL,INIT         ;Initialization code
0010  BA 037A                   MOV     DX,CONTROL      ;Control port access
0013  EE                        OUT     DX,AL           ;Write the code


                        ;Poll the printer waiting for BUSY to be low

0014  B0 0E                 NEXT:   MOV     AL,S_HIGH       ;Be sure STROBE is high
0016  EE                            OUT     DX,AL           ;Write to control port
0017  BA 0379               POLL:   MOV     DX,STATUS       ;Status port access
001A  EC                            IN      AL,DX           ;Get BUSY status
001B  A8 80                         TEST    AL,10000000B    ;Test BUSY bit
001D  74 F8                         JZ      POLL            ;Wait until READY
```

(b)

Figure 8–10. *(continued)*

```
                        ;Printer is ready, fetch and output a byte
001F  AC                          LODSB                 ;Get byte
                                                        ;and advance pointer.
0020  BA 0378                     MOV     DX,LPT1       ;Data port access
0023  EE                          OUT     DX,AL         ;Output to printer


                        ;Strobe the printer
0024  B0 0F                       MOV     AL,S_LOW      ;STROBE=0
0026  BA 037A                     MOV     DX,CONTROL    ;Control port access
0029  EE                          OUT     DX,AL         ;Write to LPT1 control port


                        ;Repeat the polling loop until all data has been printed
002A  E2 E8                       LOOP    NEXT          ;Do CX times

002C  CB                                  RET           ;Then return
002D                              FIG8_10 ENDP
002D                              CODE    ENDS
                                          END
```

Figure 8–10. *(continued)*

Homework 10 - answers

8.5 (a) 3000H (12 KB); (b) LODSB in 001F; (c) OUT DX,AL in 0023; (d) E010; (e) 0200

8.6
```
        mov     dx,0e00h    ;Point at status port
poll:   in      al,dx       ;Read status
        test    al,40h      ;Check
        jz      spl         ;If low, then ready
        test    al,10h      ;Else check NPL
        jnz     npl         ;If high, then ready
        test    al,04       ;Else check AH2
        jnz     ah2         ;If high, then ready
        test    al,02       ;Else check
        jz      ah1         ;If low, then ready
        test    al,01       ;Else check
        jz      ah0         ;If low, then ready
        jmp     poll        ;Else repeat loop
```

8.7
```
        mov     dx,0e00h    ;Point at status port
poll:   in      al,dx       ;Read status
        test    al,10h      ;Check NPL
        jnz     poll        ;Wait for NPL to be inactive
        and     al,05h      ;Test AH0 and AH2
```

```
cmp     al,01h      ;Are both inactive?
je      poll        ;If yes then continue to poll
jmp     ready       ;Else goto READY
```