Q1) *[35 points]*

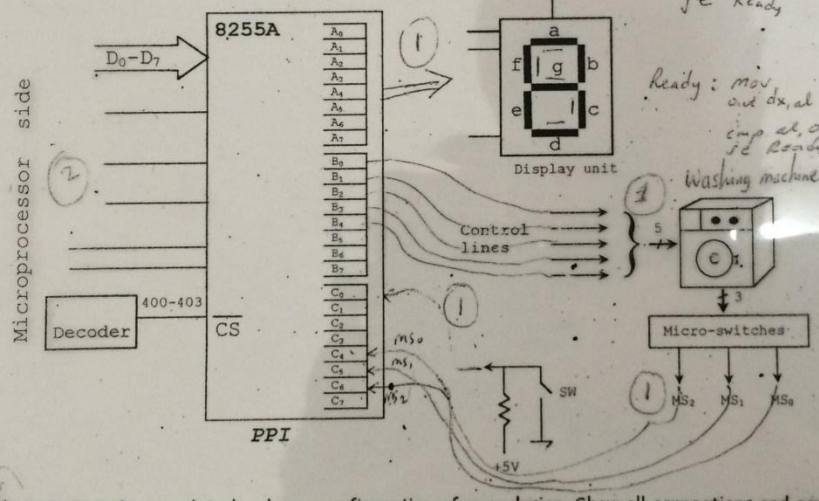
Suppose a microprocessor is to control a washing machine using the components: one common-anode 7-segment display unit, one PPI at I/O addresses 400H-403H, and one external switch. The washing machine contains some devices which have to be controlled: WATER VALVE, HEATER, DRAIN PUMP, MAIN MOTOR. The main motor has two speeds in order to allow it to wash and spin. There are three micro-switches in the washing machine which generate 3-bit binary data for each of the four devices according to the following truth table:

MS_2	MS_1	MS_0		Valve Open	Heat On	Motor Fast	Motor Slow	Pump On
0	0	0	READY	0	0	0	0	0
0	0	1	FILL	1	0	0	0	0
0	1	0	HEAT	0	1	0	0	0
0	1	1	WASH	0	0	0	1	0
1	0	0	DRAIN	0	0	0	0	1
1	0	1	RINSE	1	0	0	1	0
1	1	0	DRAIN	0	0	0	0	1
1	1	1	SPIN	0	0	1	0	0

It is required to display the following message on the 7-segment unit:

When user wants the washing machine to spin (SW is closed) → display 5 ~~on the display unit~~

When user wants the washing machine to spin (SW is open) → nothing is displayed ~~on the display unit~~



a) Give the complete hardware configuration of your design. Show all connections and control signals in details.

b) Write a complete assembly language program to fulfill the above requirements. Write comments where necessary.

Note1: The command register format of the PPI is:

1	0	0	A	Ch	0	B	Cl

1≡input
0≡output

Note2: The display word format is:

x	a	b	c	d	e	f	g

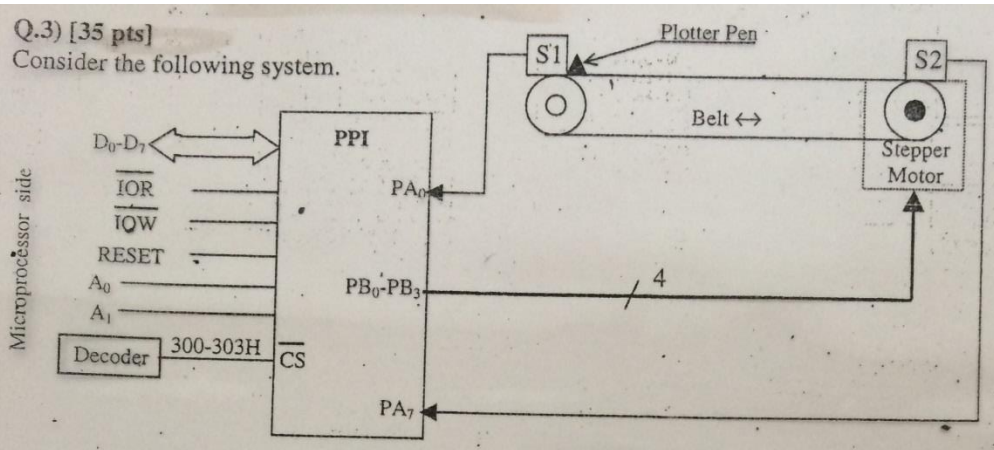
1≡OFF
0≡ON

Q1) b) Completely working = 5

```
        DOSSEG
        .Model Small
        .Data
        mov dx, 403h
        mov al,10001001B
        out dx,al
        dec dx
        in al,dx
        mov dx,400h
        xor bl,bl
        and al,00000001B
        cmp al,01
        je No_spin
        mov al,0A4h     ;display S
        out dx,al
        mov bl,1
        jmp Ready
No_spin: mov al,0FFh ;dark display
        out dx,al
Ready:  inc dx          mov dx, 401h
        mov al,00h
        out dx,al
        inc dx          402
        in al,dx
        and al,01110000B
        cmp al,00000000B
        je Ready
Fill:   dec dx
        mov al,10h
        out dx,al
        inc dx
        in al,dx
        and al,01110000B
        cmp al,00010000B
        je Fill
Heat:   dec dx
        mov al,08h
        out dx,al
        inc dx
        in al,dx
        and al,01110000B
        cmp al,00100000B
        je Heat
Wash:   dec dx
        mov al,02h
        out dx,al
        inc dx
        in al,dx
        and al,01110000B
        cmp al,00110000B
        je Wash
Drain1: dec dx
        mov al,01h
        out dx,al
        inc dx
        in al,dx
        and al,01110000B
        cmp al,01000000B
        je Drain1
Rinse:  dec dx
        mov al,12h
        out dx,al
        inc dx
        in al,dx
        and al,01110000B
        cmp al,01010000B
        je Rinse
Drain2: dec dx
        mov al,01h
        out dx,al
        inc dx
        in al,dx
        and al,01110000B
        cmp al,01100000B
        je Drain2
Spin:   cmp bl,1
        jne Stop
        mov al,04h
        out dx,al
Stop:   mov al,4Ch
        int 21h
        end
```

Q.3) [35 pts]

Consider the following system.



In this system the rotor of the stepper motor is mounted to a belt on which a plotter pen is connected. The plotter pen moves left when the stepper motor rotates counter clockwise, and moves right when the stepper motor rotates clockwise. There are two identical switches (S_1 and S_2) on both sides that are used for end detection.

- A switch normally gives an output of 0 volts. As long as the plotter pen touches a switch, it gives 5 volts.
- A 1 ms delay subroutine called DELAY is available (you don't have to right it).
- The I/O port addresses of the PPI are 300H-303H.
- Initially, assume the plotter pen is touching S1.
- Although not important, assume the motor rotates 1 degree per step.

You are required to write a complete assembly program that will move the plotter pen from left-to-right then right-to-left 50 times, with a delay of 2 ms between successive steps.

[*Your grade will be inversely proportional with the length of your program*]

Q.3 Solution)

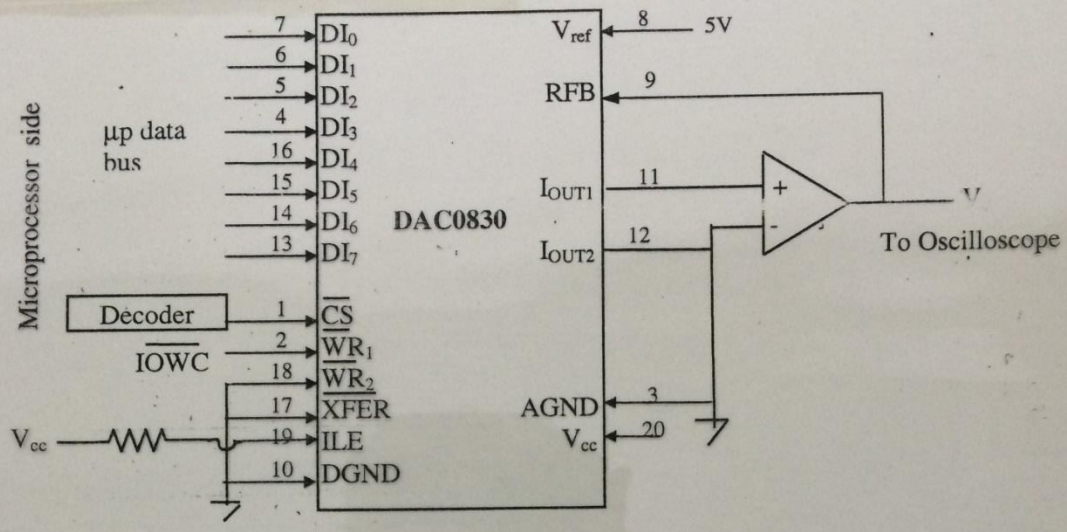
```
            DOSSEG
            .Model      small
            .Code
            mov   dx,303h
            mov   al,10010000B      ;PortA input, PortB output
            out   dx,al
            mov   bl,11001100       ;initial pattern
            mov   cx,50             ;initialize counter

Right:      mov   dx,301
            ror   bl,1              ;move to the right
            mov   al,bl
            out   dx,al
            call  Delay
            call  Delay             ;wait for 2 ms between steps
            dec   dx
            in    al,dx             ;read PortA
            shl   al,1              ;check whether you reached S2
            jnc   Right             ;if not continue moving right

Left:       inc   dx
            rol   bl,1              ;move to the left
            mov   al,bl
            out   dx,al
            call  Delay
            call  Delay             ;wait for 2 ms between steps
            dec   dx
            in    al,dx             ;read PortA
            shr   al,1              ;check whether you reached S1
            jnc   Left              ;if not continue moving left
            loop  Right             ;one turn is completed
            mov   ah,4ch
            int   21h
            END
```

Good Luck in the Final Exam...

Q.4) [20 pts]
The following digital-to-analog converter DAC0830 is interfaced to an 80×86 µp system at I/O port address 700H as shown below.



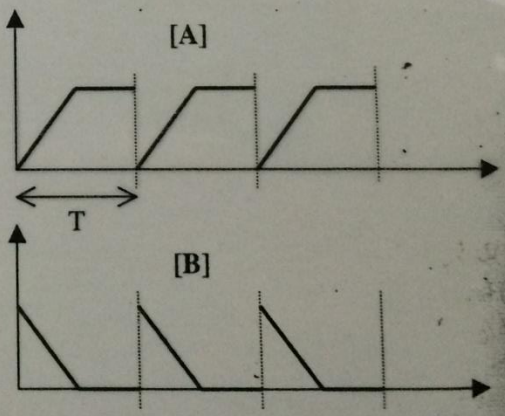
The program given below is run.

```
            DOSSEG
            .Model small
            .Code
            mov   dx,700h
            mov   al,0
Loop1:      out   dx,al
            call  Delay
            inc   al
            cmp   al,0FFh
            jb    Loop1
            mov   bl,0FFh
Loop2:      out   dx,al
            call  Delay
            dec   bl
            jne   Loop2
            inc   al
            jmp   Loop1

Delay:      ...           ;2 µsec delay
            ret

            mov   ah,4ch
            int   21h
            END
```

a) [15 pts]
Draw the analog waveform that will be seen on the Oscilloscope while running this program.



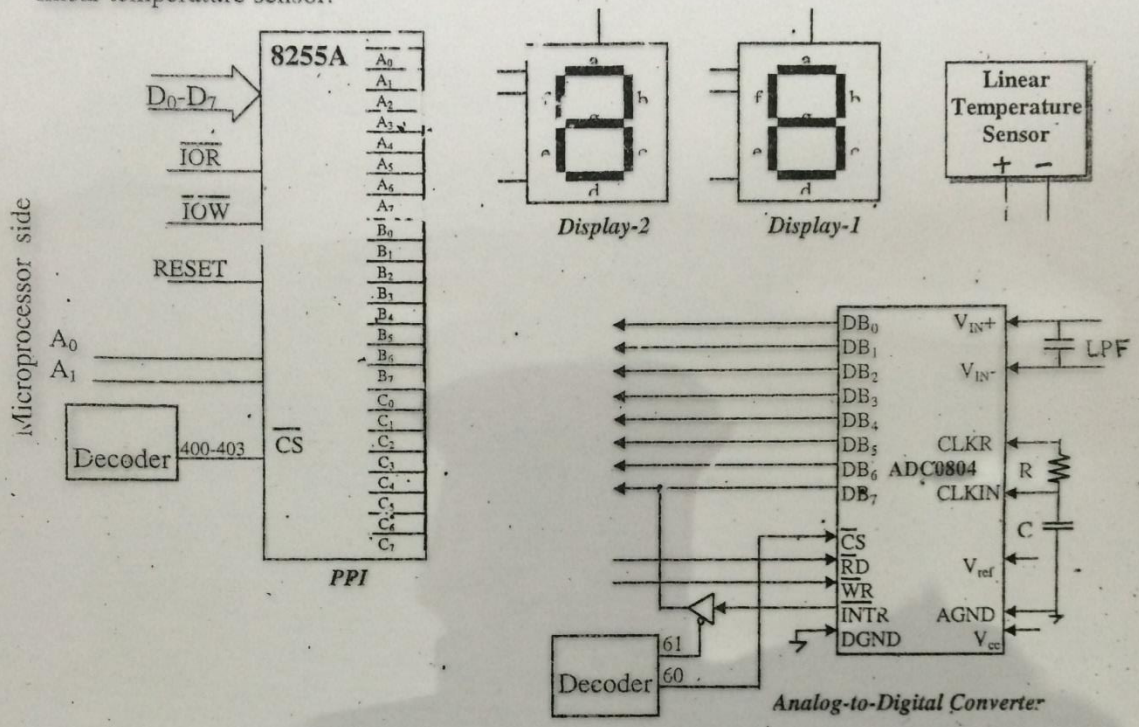
b) [5 pts]
Approximately, what is the frequency of this waveform?

T = 2×256×2 = 1024 µs.

F = 1/T = 1/1024 µs = 976.6 Hz ≈ 1 KHz

Q.3) [23 pts]

You are given the following components: Two common-anode 7-segment displays. One PPI at I/O addresses 400H-403H. One ADC at I/O addresses 60H and 61H, and one linear temperature sensor.



Assume that V_{ref} is set to a value that satisfies the following table:

Temperature To be displayed (^{0}C)	V_{IN} (mV)	Binary output
00	0	0000 0000
01	10	0000 0001
02	20	0000 0010
03	30	0000 0011
.....	.....	.....
99	990	0110 0011

You are required to interface the above components to an 80x86 µp to continuously display the currently measured temperature in BCD form on the 7-segment displays.

a) **[8 pts]** Give the complete hardware configuration of your design..

b) **[15 pts]** Write a complete assembly program to fulfill the above requirements. Write comments where necessary. *(Your grade will be inversely proportional to the length of your program)*

Note1: The command register format of the PPI is:

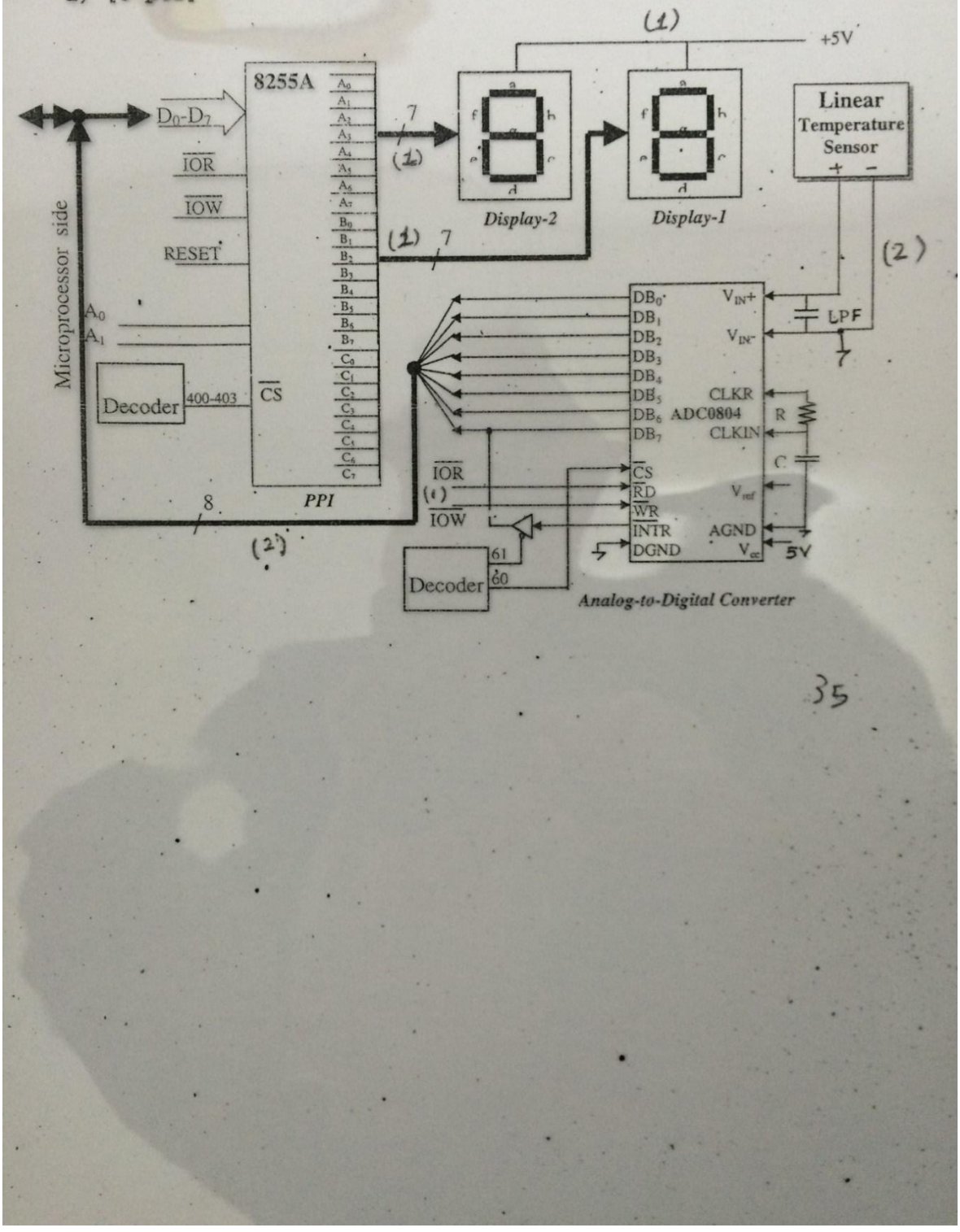
1	0	0	A	Ch	0	B	Cl

1≡input
0≡output

Note2: The display word format is: x a b c d e f g

1≡OFF
0≡ON

Q.3)[23 pts]
a) [8 pts]



35

b) [15 pts]

```
        Dosseg
        .Model small
        .Stack  100
        .386
        .Code
        jmp     Start
LUT     DB      81h,CFh,92h, ..;display words of 0-9
Begin:  mov     dx,403h
        mov     al,10000000B ;PortA and PortB are outputs
        out     dx,al
Start:  out     60h,al          ;start conversion
notyet: in      al,61h          ;read INTR
        test    al,80h          ;conversion finished ?
        jnz     notyet          ;poll until INTR=0
        in      al,60h          ;get temperature from the ADC
        add     al,0            ;convert the read temperature
        daa                     ;to BCD

        push    ax              ;save the two BCD numbers
        xor     bh,bh           ;use bx as index
        mov     bl,al
        and     bl,0Fh          ;prepare the first BCD digit and
        mov     al,byte ptr CS:[offset LUT+bx]      ;get its display word
        mov     dx,401h         ;display the first digit
        out     dx,al           ;on PortB

        pop     ax
        mov     bl,al
        shr     bl,4            ;prepare the second BCD digit and
        mov     al,byte ptr CS:[offset LUT+bx]      ;get its display word
        dec     dx              ;display the second digit
        out     dx,al           ;on PortA
        jmp     Start

        mov     ax,4ch          ;leave control to DOS
        int     21h
        END
```