

KTÜ Bilgisayar Mühendisliği / Bilgisayar Organizasyonu Lab. Final Sınavı

Öğr. No: Ad Soyad:Grup: ☐ A ☐ B

Soru 1) a) Analog Dijital Çevirici deneyinde rampa ve successive approximation yöntemleri incelenmişti. Bu yöntemleri analog sayısal çevirme hızı ve algoritma karmaşıklığı yönlerinden karşılaştırınız. Hangi durumda rampa yöntemi daha hızlı cevap verir.

[illegible]

b) Girişine -2.56V ile 2.55V arasında değerler uygulanabilen 8 bitlik bir ADC dönüştürücünün analog girişine 1V bir gerilim uygulanmaktadır. Aşağıdaki soruları cevaplayınız. ADC nin V_a , V_{min} , V_{max} , çözünürlük(n) ve kuantum(q) değerlerini yazınız.

c) ADC'nin çalışma prensibi successive approximation olduğuna göre ara işlem adımlarını deneyde yaptığınız gibi aşağıdaki tabloyu doldurarak gösteriniz, elde ettiğiniz sayısal sonucu hexadecimal biçimde veriniz.

[illegible]

Soru 2) Aşağıdaki kodu mikroişlemcide (8086) koşturduğumuzda kaydedicilerdeki değerleri yazınız.

```

ORG 100H
MOV DH, 67H
MOV AL, 34H
MOV SI, 0130H
MOV CL, 00H
DONGU: MOV [SI], AL
        INC CL
        INC SI
        MUL AL
        ADD AL, DH
        MOV BH, [SI-1]
        ADD BH, [SI-1]
        DIV BH
        CMP AL, [SI-1]
        JZ FINISH
        JMP DONGU

```

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| AX | BX | CX | 130 | 131 | 132 |
| — — | — — | — — | — | — | — |

Soru 3) Arduino Uno kullanılarak, bir butona basılması durumunda bir ledin kesmeli G/Ç kullanılarak yakılması istenmektedir. Buton ve ledin sırasıyla 9 ve 12 no'lu dijital pinlere bağlı olduğunu varsayarak, ilgili pinleri uygun şekilde konfigüre edip ilgili görevi yerine getirecek programı yazınız.

[illegible]

Soru 4) Aşağıda verilen Assembly kodunda adım motoru 1 tam tur dönme işlemini turun ilk yarısında motor yavaştan hızlı harekete geçerken turun ikinci yarısında yavaşlamaya başlayıp durur (Yavaş-Hızlı-Yavaş). Kodda değişiklikler (kod ekleme, kod silme ve kod değiştirme) yaparak motorun 1 tam turunu yavaştan hızlıya doğru hızının giderek artacak şekilde tamamlayacağı hale getiriniz.

```
#start=stepper_motor.exe#
name "stepper"
#make_bin#
steps_before_direction_change = 8h ;
32 (decimal)
jmp start
; ===== data =====
; bin data for clock-wise
; half-step rotation:
datcw      db 0000_0110b
            db 0000_0100b
            db 0000_0011b
            db 0000_0010b

start:
mov bx, offset datcw ; start from
clock-wise half-step.
mov si, 0
mov cx, 0 ; step counter
mov dx, 24h

next_step:
; motor sets top bit when it's ready
to accept new command
wait:      in al, 7
            test al, 10000000b
            jz wait
mov di, dx
delay2:
sub di, 2h
nop
jnz delay2

cmp cx, 4h
jb azaltim
jmp arttirim
add dx, 2h

arttirim:
add dx, 2h
jmp devam

azaltim:
sub dx, 2h
jmp devam

devam:
mov al, [bx][si]
out 7, al
inc si
```

```
cmp si, 4
jb next_step
mov si, 0
inc cx
cmp cx,
steps_before_direction_change
jb next_step
int 21h
end
```

Soruların program çıktıları ile ilişkisi: **S1:** PÇ2 → **S2:** PÇ2, PÇ4 → **S3:** PÇ3 → **S4:** PÇ2, PÇ3

Başarılar.