

1. Bir işlemcide kullanılan stack pointer(yığın bellek göstergeci) in iki farklı çalışma şeklini yazınız. 10 Puan
- i. push R4 ile R4 kaydedicinin içeriği stak belleğe adress gerektirmeden saklanırken, pull R4 ile staktaki bilgi R4 kaydedicisine adres belirtmeden okunur.*
- ii. call #subprog ile alt programa gidilirken bir sonraki komutun adresi, dönüşte koşulacak komutun adresi stak-pointere otomatik saklanır. Ne zaman RET komutu koşulacak olsa bu saklanan adress PC ye otomatik yüklenir.*

2. Bir işlemcinin belleğinde kayıtlı 0e3dh sayısının q.11 formatında dönüştürme yapan 12 bitlik ADC den geldiği bilinmektedir. Eğer bu işlemcinin analog referans gerilimleri VREF_=-1.2 volt ve VREF+=+1.2 volt ise 0e3dh sayısının gerilim değeri nedir? 20 Puan

12-bit iki'nin tümleyeni olarak yorumla:

$0xE3D = 3645_{10} > 2047 \Rightarrow \text{negatif.}$

$N = 3645 - 4096 = -451$

Q.11 değeri (normalize):

$x = N/211 = -451/2048 \approx -0.22021484375$

Gerilime çevir: ($\pm 1 \leftrightarrow \pm 1.2 V$)

$V = x \times 1.2 V \approx -0.2642578125$

3. Kayan noktalı sayılar kullanılarak elde edilen 67.3 değerinin IEEE754 standardında 32 bitlik sayı biçimi kullanılarak belleğe kaydedilmiş değeri ne olur? (16 lı sayı sistemindeki gösterimle bellekteki sayıyı bulunuz) 20 Puan

Aşağıdaki IEEE-754 tek duyarlık (32-bit) gösterimdir:

Sayı: 67.3 (decimal)

İşaret biti: 0

Üs (bias=127): gerçek üs =6=6=6 $\Rightarrow$ alan 10000101 (133 = 0x85)

Kesir (mantissa) bitleri: 00001101001100110011010

32-bit ikili: 01000010 10000101 00001101001100110011010

Hex (big-endian kelime): 0x4286999A

4. Aşağıdaki **ghh** alt programının işlevini bir veya iki cümle ile yazınız. 10 P

```
float sd=0.7,ss=0;
void ghh(void){
    sd=ss;
    if (ss > 0.7) sd=0.7
    if(ss < -0.7) sd=-0.7
}
```

Eğer ss deki sayı 0.7 den büyükse sd 0.7, ss deki sayı -0.7 den küçükse sd -0.7 yoksa sd=ss olur.

5. Aşağıdaki **altpr** alt programın işlevini bir iki cümle ile yazınız. 10 P

```

altpr    clr    r5
ss       mov 0200h(r5), r6
         rla    r6
         rla    r6
         rla    r6
         rla    r6
         mov r6,0200h(r5)
         incd   r5
         cmp    #020h,r5
         jnz    ss
         ret

```

Alt program 0200 h adresinden itibaren 16 tane 16 bitlik sayıyı 16 sayısına böler.

6. Bir seq dizisi 40 kelime uzunluğunda ve işaretli tam sayı formatındadır (**int seq[40]**). Bu dizideki negatif elemanları pozitif yapan bir alt programı c dilinde yada bildiğiniz standard bir programlama dilinde (c, fortran, python, go veya R) yazınız. 10 P

```

void make_positive(int seq[], int n) {
    for (int i = 0; i < n; i++) {
        if (seq[i] < 0) {
            seq[i] = -seq[i]; // Negatif olanı pozitif çevir
        }
    }
}

```

7. Bir ADC nin dönüştürdüğü 30 adet sayı bellekte int samples adlı dizide yer almaktadır. Bu sayıların ortalama değerini 30 a yada başka bir sayıya *doğrudan bölme kullanmadan* sağa sola kaydırma ile yapmanız istenmektedir. En iyi sonucu vereceğini düşündüğünüz yöntemi matematiksel olarak veriniz veya bir akış diyagramı ile gösteriniz (Bir alt program da olabilir). 20 P

```

int mean30(int samples[], int n) {
    long sum = 0;
    for (int i = 0; i < n; i++) {
        sum += samples[i];
    }
    // Bölme yerine bit kaydırma ile yaklaşık
    int mean = (sum >> 5) + (sum >> 9);
    return mean;
}

```