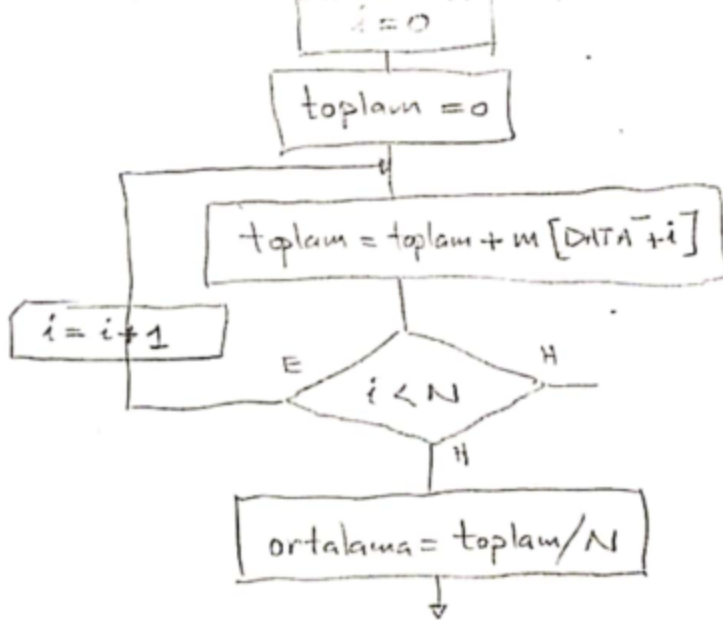


1	2	3	Toplam
1	1	3, 4	PÖÇ

1. a. Which addressing mode should be used to access all the elements of an integer array corresponding to student grades to compute the average grade? $\Rightarrow$ Ortalama not hesaplamak için öğrenci notlarına karşılık gelen bir tamsayı dizisinin tüm elemanlarına erişmek için hangi adresleme modu kullanılmalıdır? *Otomatik artırmalı (Auto-increment) (Register Indirect) Kaydedici üzerinden dolaylı adresleme veya İndisli adresleme (indexed-addressing) modu kullanılmalıdır.*
- b. Explain briefly in a sentence the most important advantage of using this addressing mode. $\Rightarrow$ Bu adresleme modunu kullanmanın en önemli avantajını bir cümle ile kısaca açıklayın. *Adresin her bit emir ile belirtilmesi gerekmeyişi için adres alanı olabilmek en büyük avantajdır. 32 bit ile temsil edildiği için çalışma verimi ve hızı yüksektir.*
2. a. Assume that a 64-bit microprocessor executes a compute intensive program while expecting interrupt signal from I/O devices. The maximum number of clock cycles to execute a complex instructions is 19 and the minimum number of clock cycles to execute a simple instruction is 1. What would be the average latency to respond to an incoming interrupt? $\Rightarrow$ 64 bitlik bir mikroişlemcinin, G/Ç aygıtlarından kesme sinyali beklerken yoğun hesaplamalı bir program yürüttüğünü varsayın. Karmaşık bir emri yürütmek için en fazla saat döngüsü sayısı 19 ve basit bir emri yürütmek için en az saat döngüsü sayısı 1'dir. Gelen bir kesmeye yanıt vermek için ortalama gecikme ne olur?
- $(19+1)/2 = 10$ saat periyodu*
- b. Briefly explain the reason in one sentence why microprocessors have a flip-flop to mask the interrupt input signal under programmers' control. $\Rightarrow$ Mikroişlemcilerin programcıların kontrolünde kesme giriş sinyalini maskeleyerek için neden bir flip-flop'a sahip olduklarını bir cümle ile kısaca açıklayın. *Kesmeye verilen hizmet programına gitmeden önce işlemcinin tüm (kaydedicilerinin) yığına itilmesi ve hizmet programından geri dönerken tüm durumun yığından geri yüklenmesi sayesinde oluşabilecek gecikmenin, kritik işler yapılırken kabul edilemez.*
3. From Wikipedia, the Free Encyclopedia; The Global Positioning System or GPS project was launched in the United States in 1973 to overcome the limitations of previous navigation systems, combining ideas from several predecessors, including classified engineering design studies from the 1960s. Initially, the highest-quality signal was reserved for military use, and the signal available for civilian use was intentionally degraded, in a policy known as **Selective Availability**. Even though differential GPS services by private industry can improve civilian accuracy, it costs more than a standard GPS receiver. One can also improve the precision of the randomly modified GPS coordinates by averaging the coordinates by a simple algorithm for non real-time tasks. $\Rightarrow$ Vikipedi, Özgür Ansiklopedi'den; Küresel Konumlandırma Sistemi veya GPS projesi, 1960'lerden kalma askeri amaçlı mühendislik tasarım çalışmaları da dahil olmak üzere birkaç öncüden gelen fikirleri birleştirerek önceki navigasyon sistemlerinin sınırlamalarının üstesinden gelmek için 1973'te Amerika Birleşik Devletleri'nde başlatıldı. Başlangıçta, en yüksek kaliteli sinyal askeri kullanım için ayrıldı ve sivil kullanım için mevcut sinyal, **Seçici Kullanılabilirlik** olarak bilinen bir politikada kasıtlı olarak düşürüldü. Özel sektör tarafından sağlanan farklı (Differential) GPS hizmetleri sivil doğruluğu iyileştirebilse de, standart bir GPS alıcısından daha maliyetlidir. Gerçek zamanlı olmayan uygulamalar için basit bir algoritma ile koordinatların ortalamasını alarak rastgele değiştirilmiş GPS koordinatlarının kesinliği de iyileştirilebilir.
- a. Draw the flowchart of the averaging algorithm for the last $N=2^k$ samples assuming that integer data to be averaged is stored in an array at address DATA in memory. $\Rightarrow$ Ortalama alınacak tamsayı verilerinin bellekteki DATA adlı adresteki bir dizide saklandığını varsayarak son $N=2^k$ örnek için ortalama alma algoritmasının akış şemasını çizin.



CamScanner ile tarandı

- b. Code averaging algorithm in any high level language. $\Rightarrow$ Ortalama algoritmasını herhangi bir yüksek seviyeli dil ile kodlayınız.

```

unsigned
for(i=0, toplama=0; i<N; i++)
    toplama += M[DATA+i];
ortalama = toplama >> k;
  
```

- c. Code averaging algorithm in Assembly language. $\Rightarrow$ Ortalama algoritmasını Assembly dili ile kodlayınız.

```

MOV R1, #0;      R1 i indisi
MOV R2, #0;      R2 toplama
MOV R3, DATA;   R3 dizi elemanı başlangıç adresi
Döngü: ADD R2, R3;  tüm elemanları topla
      INC R3;      indisi ilerlet
      INC R1;      dizi sonu mu?
      CMP R1, #N;
      BL      değilse döngüye devam
      SHR R2, K;   N'ye böl ve ortalamayı hesapla
  
```

- d. What potential problem does the averaging method have for improved precision by increasing number of samples N and how would you tackle the problem? $\Rightarrow$ N örnek sayısını artırarak gelişmiş hassasiyet için ortalama alma yönteminin hangi potansiyel sorunu vardır ve bu sorunu nasıl çözersiniz?

Örnek sayısı N artırılarak doğruluğa yükseltmeye çalışırken tek bir kaydedici üzerinde taşma olması riski vardır. Bu problemi gidermek için toplama tek yerine 2 veya daha fazla kaydediciyi birlikte kullanarak toplama sırasında taşma riski engellenebilir.

CamScanner ile tarandı