



Ad	Soyad	Öğrenci Numarası	Bölüm

## Açıklamalar

1. Sınavı çözmeye başlamadan önce tüm açıklamaları ve soruları okuyun. Sınavda toplam 4 soru vardır ve soruların toplam değeri 105 puandır. Bütün soruların değeri köşeli ayraç ile belirtilmiştir.
2. Adınız ve öğrenci numaranızı ilk sayfaya yazmayı unutmayın.
3. Her türlü dijital ve yazılı kaynak serbesttir fakat başkasından yardım almak kesinlikle yasaktır. Kopya kontrolü yapılacaktır.
4. Sonucu yanlış olan yanıtlar puan alamayabilir. Gidiş yolunun ayrıntılı gösterilmesi sorudan puan alınması için gereklidir ancak yeterli değildir. Açıklamasız işlem yapmamaya özen göstermeniz alacağınız puanı artıracaktır.
5. Sınav sırasında soru alınmayacaktır.
6. Herhangi bir soruda hata ya da eksik olduğunu düşünüyorsanız varsayarak yapın, varsayımınızı açıklayın.
7. Teslim tarih ve saatinden önce sınavınızı yüklediğinizden emin olun. 1 dakika bile geç göndermeniz halinde sınavınız geçersiz sayılacaktır.
8. Aşağıda verilen namus sözünü imzalayın (isteğe bağlı).

<b>Bu sınavda kimseye yardım etmedim, kimseden yardım almadım ve kimseyi yardım alırken ya da başkasına yardım ederken görmedim.</b>	<b>İmza:</b>
--	--------------

Soru	a (45)	b (30)	c (10)	d (20)	Toplam (105)
Not					

## [105 Puan] Uzay İstilasası Oyunu

Şekil 1'de Uzay İstilasası oyunundan bir kare gösterilmektedir.



Şekil 1: Uzay İstilasası Oyunundan Bir Kare

Uzay İstilasası oyununun özellikleri şunlardır:

- 1-) Ana oyuncunun hareketi için aritmetik işlemler kullanılmaktadır.
- 2-) Düşman hareketleri mantıksal işlemler ile sağlanmaktadır.
- 3-) Düşman vurma işlemleri için yüksek miktarda matris çarpım işlemi vardır. Düşman vuruldu durumu için matris karşılaştırma işlemi yapılmaktadır.
- 4-) Oyuncu bir düşmanı vurduğunda puan kazanmakta, düşman tarafından vurulduğunda puan kaybetmektedir.
- 5-) Silah imleç hareketinde sabit değerler ile toplama ve çıkarma işlemleri yapılmaktadır.
- 6-) Oyunda gece görüşü moduna geçildiğinde, oyun ekranında gösterilen renklere ait sayılar bitsel olarak ters çevrilmektedir. (Örneğin 1101 sayısı 1011 sayısına çevrilmektedir.)
- 7-) Oyunun başlangıç dosyası başlangıçta bellekten okunmaktadır.
- 8-) Oyunu kaydettiğinizde belleğe yazılmaktadır ve tekrar başlamak isterseniz kaldığı yerden devam etmektedir.
- 9-) Bir bölüm bittiğinde başka bir bölüme geçilmektedir ve bölüm değiştirdiğinizde bellekte birbirinden uzak, ardışık olmayan adresler arasında otomatik olarak geçiş yapılmaktadır.
- 10-) Oyun başlangıç, oynanış ve bitiş kontrolleri yapılmakta ve duruma göre bellekte birbirinden uzak, ardışık olmayan adresler arasında geçiş yapılmaktadır.

11-) Oyundaki düşman vurma ve karşılaştırma kod parçası aşağıdaki gibi C dilinde yazılı olarak verilmiştir (A, B, C değişkenleri *int* matrisleridir(dizileridir).):

```
for (int i = 0; i < N; i++) {  
    for (int j = 0; j < N; j++) {  
        // matris carpma  
        for (int k = 0; k < N; k++) {  
            C[i][j] = A[i][k] * B[k][j];  
        }  
  
        // matris karsilastirma  
        if (C[i][j] > 0) {  
            C[i][j] = 1;  
        }  
        else {  
            C[i][j] = 0;  
        }  
    }  
}
```

**Not:** C kodu temsildir, gerçeği yansıtmamaktadır, saçma olabilir, mantıklı olmasına gerek yoktur ve doğruluğu önemli değildir. C kodunda gereksiz olsa bile bu kod parçaları için optimizasyon yapılmadığını varsayın.

Uzay İstilasası oyununun çalışması için bir buyruk kümesi mimarisi ve buna uygun işlemci tasarlamamız gerekmektedir. İşlemci tasarımı için gereken minimum kriterler şunlardır:

Tasarlayacağımız mimari/mimaride/işlemci/işlemcide;

- 1-) Tek vuruşlu 16-bit ya da 32-bit mimariye sahip olabilir.
- 2-) Oyun kriterlerini sağlayacak minimum sayıda buyruk kullanılmalıdır. (Minimum sayıda buyruk derken kastedilen şey gereksiz buyrukların olmamasıdır, belli bir sayısı yoktur.)
- 3-) Buyruk isimleri anlamlı ve Türkçe olmalıdır.
- 4-) 2-bit adresleme kullanılmalıdır.
- 5-) Program sayacı 0x0000 adresinden, veri belleği adresi 0x0200 adresinden başlamalıdır.
- 6-) Yazmaç sayısı tasarlanacak mimariye göre belirlenmelidir.
- 7-) Sayılar ikiye tümleyen olarak yorumlanmaktadır.
- 8-) *int* değişken boyutu tasarlanan mimariye bağlı olarak değişmektedir ve mimari bit sayısı kadar olmaktadır.

**Not:** Mimariyle ilgili belirlenmeyen bir ister varsa varsayarak yapın ve neden öyle varsaydığınızı açıklayın.

Bu oyuna ve oyunu oynatmak için gerekli olan mimari istelere uygun olarak aşağıdaki soruları cevaplayın.

## a. [45 Puan] Buyruk Kümesi Mimarisi ve İşlemci Tasarımı

[10 Puan] a.1-) Oyunu oynatmak için mimari tasarımınıza eklemeniz gereken tüm buyrukları sıralayın ve her bir buyruk için neden gerektiğini, oyunun hangi maddedeki hangi kriterini karşıladığını açıklayın.

Buyrukların iş kodu (opcode), işlev kodu (funct), yazmaç adresi, anlık değer gibi kısımlarını belirleyin, hangi kısmı kaç bit ile gösterdiğinizi ve kaç buyruk tipi kullandığınızı söyleyin. Aşağıda gösterilen formattaki gibi bit alanlarını, binary gösterimini yazın (örnek 32 bittir, siz kendi mimarinize göre yazacaksınız):

31	25	24	20	19	15	14	12	11	7	6	0	
0000000		rs2		rs1		000		rd		0000000		ÖRNEK.BUYRUK

[4 Puan] a.2-) Buyrukların assembly formatını yazın. Örneğin; *ornek.buyruk rd, rs1, rs2*

[1 Puan] a.3-) Tasarladığınız mimaride kaç tane yazmaç kullandığınızı söyleyin, neden?

[1 Puan] a.4-) Tasarladığınız işlemcinin hangi mimariye (Von-Neumann ya da Harvard) sahip olduğunu söyleyin, neden?

[2 Puan] a.5-) Tasarladığınız mimariye göre her yeni buyruk geldiğinde program sayacını kaç arttırmanız gerektiğini ve neden olduğunu söyleyin.

[2 Puan] a.6-) Tasarladığınız işlemcide buyruk belleğine kaç adet statik buyruk sığdırabilirsiniz, sizce oyun için yeterli olur mu, neden?

[10 Puan] a.7-) Tasarladığınız bu mimariye uygun olarak tek vuruşlu işlemcinizi çizin ve denetim tablosunu oluşturun.

**Not:** İşlemciyi çizerken derste anlatıldığı gibi çizmелisiniz, örneğin "tüm işlemleri yapıcı" gibi çok üst düzey bir kapalı kutu kullanamazsınız, aritmetik mantık birimi gibi daha alt birimler kullanmalısınız.

[15 Puan] a.8-) Tasarladığınız işlemciyi Verilog donanım tanımlama dilinde davranışsal modelleme kullanarak yazın. Dosya ismi *islemci.v* olsun. "islemci" modülünün giriş çıkışları Tablo 1'deki gibi olsun:

TABLO 1: islemci Modülünün Giriş ve Çıkış Sinyalleri

Sinyal	Yön	Genişlik	Açıklama
saat	Giriş	1 bit	Sıralı mantık için gerekli saat girişi.
reset	Giriş	1 bit	Devreyi başlangıç durumuna döndüren sinyal. <i>program_sayaci</i> ve <i>yazmaclar</i> sıfırlanmalı.
buyruk	Giriş	X bit	İşlemcinin şu anki saat vuruşunda yürüteceği buyruk.
program_sayaci	Çıkış	X bit	Bir sonraki çevrim getirilecek buyruğun adresi.
yazmaclar	Çıkış	X adet X bit	X bit genişliğe sahip X adet değerlerin saklandığı yazmaçlar.

## b. [30 Puan] Başarım ve Boru Hattı

Tek vuruşluk işlemcinizin 1 GHz’de çalıştığı ve 5 aşamalı boru hatlı olsaydı 2 GHz’de çalışacağı biliniyor.

**[5 Puan] b.1-)** Düşman vurma kod parçasını tasarladığınız mimariye uygun olarak a şıkında belirlediğiniz assembly formatına göre assembly kodu olarak yazın. (Örneğin for döngülerinden biri için *0x0110* gibi bir adres ya da *döngü1* gibi bir etiket olduğunu varsayabilirsiniz.) Assembly kodunu yazarken yanlarına yine a şıkında belirlediğiniz binary formatında binary kodlarını yazın. Örnek bir assembly kodu ve binary kodu aşağıdaki gibi yazılabilir:

```
dongu1:
    ornek.buyruk1 r1, r5, r9          0100_1001_1010_0111
    ornek.buyruk2 r4, r13, dongu2     1010_0111_0100_1001
    ...
```

**[15 Puan] b.2-)** Yazdığımız düşman vurma assembly koduna bakarak  $N=2$  için ve 5 aşamalı boru hattına sahip işlemciniz için boru hattı diyagramını oluşturun. (Aşamalar Getir, Çöz, Yürüt, Bellek, Yazmaç Yaz şeklinde olabilir.) (Yazmaç öbeğine yazılan değerlerin aynı vuruşta okunabildiğini varsayın.)

Bu boru hatlı hali için toplam çevrim sayısını bulun, yürütme zamanını hesaplayın ve buyruk başına düşen çevrimi bulun.

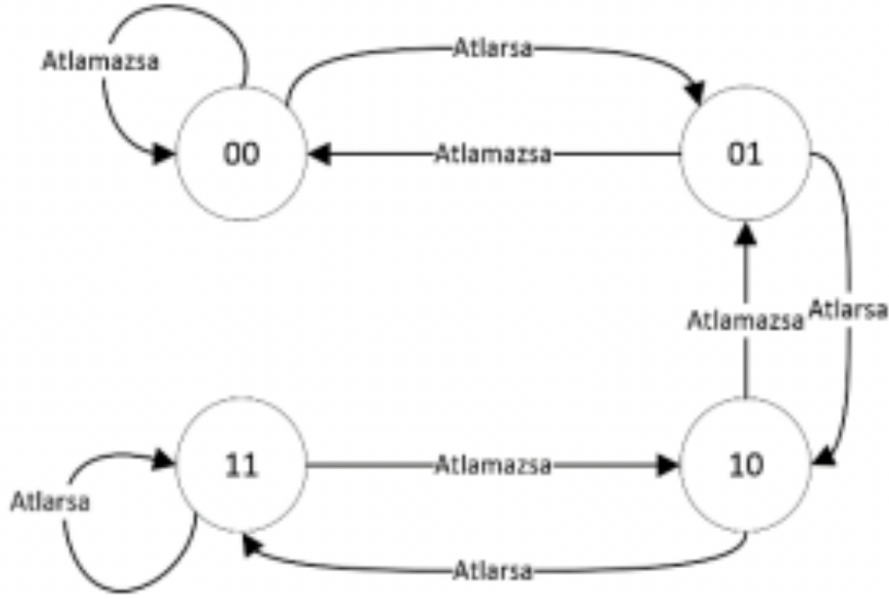
**[5 Puan] b.3-)** Aynı assembly koduna bakarak tek vuruşluk işlemcinin çevrim sayısı ve yürütme zamanını bulun, boru hatlı işlemci ile karşılaştırın ve hangi işlemcinin başarımının neden daha iyi olduğunu açıklayın.

**[Bonus][5 Puan] b.4-)** Veri bağımlılığı olan aşama ve buyrukları belirtin ve bu sorunların nasıl çözülebileceğini anlatın. (Eğer varsa; yoksa neden olmadığını söyleyin.)

### c. [10 Puan] Dallanma Öngörücü

Düşman vurma C koduna ya da b şıkında yazdığımız assembly koda bakarak  $N=3$  için Şekil 2’de durum makinesi verilen çift doruklu dallanma öngörücünün her bir atlama ya da dallanma durumu için ayrı olacak şekilde başarımını hesaplayın. (Her durağan buyruk için birbirinden bağımsız olarak ayrı başarım oranı vermeniz gerekmektedir.) Başlangıçta "11" durumundan başladığını varsayın. (00: Güçlü Atlamaz, 01: Zayıf Atlamaz, 10: Zayıf Atlar, 11: Güçlü Atlar)(İşlemleri açık şekilde gösterin.)

**İpucu:** for döngüsü için atlama durumu "büyükse atlar", if için ise "şartı sağlamıyorsa atlar" şeklindedir.



Şekil 2: Çift Doruklu Dallanma Öngörücü Durum Makinesi

#### d. [20 Puan] Önbellek

- İki yöllü kümeli ilişkili
- En uzun zamandır kullanılmayan verinin çıkarıldığı
- Verilerin satır bazlı (row-major) olarak tutulduğu ve önbellekteki satır başlarından başlanarak sırasıyla yerleştirildiği
- 2 KB boyuta sahip
- Veri öbeklerinin boyutu 32 bayt
- Başlangıçta boş

olan bir önbelleğe sahip olduğunuz biliniyorsa, düşman vurma C koduna bakarak  $N=8$  için;

**Not:** Koddaki matris karşılaştırma işlemi yapan if-else bloğunu yoksayabilirsiniz.

**Not:** Koddaki matris indislerinin veri öbeklerinin başına geldiğini varsayın.

[10 Puan] d.1-) Kod parçası çalıştırıldığında verilerin önbellekte bulunma yüzdesini bulun.

[3 Puan] d.2-) Bu önbellek için yaz ve yerini ayır (write-allocate), yaz ve yerini ayırma (write-no-allocate) yöntemlerinden hangisini kullanmanın başarımı neden artıracığını açıklayın.

[7 Puan] d.3-) Veri öbeklerinin boyutu 8 bayt olsaydı başarım yüzde kaç değişirdi? Oyun performansına etkisi nasıl olurdu, yorumlayın.

#### Sınav Teslimi (Son Teslim Tarihi: 02.04.2023 21.00)

Cevaplarınızı tarayın ya da fotoğrafını çekin ve sayfalar sıralı olacak biçimde <Soyad\_Öğrenci Numarası>.pdf olarak kaydedin. (İnternette bulunan *JPG to PDF* gibi araçlarla resim dosyalarını pdf olarak birleştirebilirsiniz.)

1-) <Soyad\_Öğrenci Numarası>.pdf

2-) işlemci.v

dosyalarını sıkıştırın ve <Soyad\_Öğrenci Numarası>.zip olarak kaydederek <https://uzak.etu.edu.tr>'ye yükleyin.