



**TOBB Ekonomi ve Teknoloji Üniversitesi**  
**Bilgisayar Mühendisliği Bölümü**  
**Elektrik ve Elektronik Mühendisliği Bölümü**

13 Aralık 2023  
BİL 265/264/264L – Mantıksal Devre  
Tasarımı ve Laboratuvarı  
2023 – 2024 Öğretim Yılı  
Güz Dönemi  
Final Sınavı

**AÇIKLAMALAR:**

1. Sınavı çözmeye başlamadan önce tüm açıklamaları ve soruları okuyun. Sınavda toplam 4 sayfa (2 adet A4 kağıdı), 2 soru var ve soruların toplam değeri 100 puandır. Bütün soruların değeri köşeli ayraç ile belirtilmiştir. Sınav süresi 120 dakikadır.
2. Sınav sırasında soru kabul edilmeyecektir.
3. Sınav esnasında internet ve tarayıcı kullanımı yasaktır. Bilgisayarda Xilinx Vivado programı dışında hiçbir program **KESİNLİKLE** açık olamaz.
4. İnternete bağlı olduğu veya herhangi bir tarayıcısı açık olduğu görülen kişilerin sınavları geçersiz sayılacak ve kopya olarak değerlendirilip gerekli işlemler yapılacaktır.
5. Sınav boyunca **sadece** bir yüzü dolu A4 sayfadan faydalanabilirsiniz. Bunun dışında her türlü araç/gereç ve kaynak kullanımı yasaktır.(hesap makinesi, akıllı saat, telefon, pdf dosyaları vb.)
6. Sonucu yanlış olan yanıtlar puan alamayabilir. Gidiş yolunun ayrıntılı gösterilmesi sorudan puan alınması için gereklidir ancak yeterli değildir. Açıklamasız kod yazmamaya özen göstermeniz alacağınız puanı artıracaktır.
7. <Projenizin bulunduğu dizin>\<Proje ismi>\<Proje ismi>.srcs\sources\_1\new → dizininde yazdığınız “.v” uzantılı dosyaları bulabilirsiniz. Simülasyon dosyalarını ise aynı uzantıda .srcs’den sonra \sim\_1 klasöründe bulabilirsiniz.
8. 9. talimata uyulmaması ve dosya isimlerinin yanlış yazılması durumlarında toplam puanınız üzerinden 20 puan kırılacaktır.
9. Dosya gönderimi için sorularda belirtilen “.v” uzantılı dosyalarınızı “isim\_soyisim\_numara\_final” isimli bir klasöre attıktan sonra klasörü sıkıştırınız ve sınav sırasında gözetmenin getireceği USB veya Uzak’a yüklemeye hazır olacak şekilde bekleyiniz.

Sınavda Göndermeniz Gereken .v dosyaları: (Gönderim yapmak istemediğiniz soruları eklemek zorunda değilsiniz.)

- matriscarpici.v
- hamur.v
- malzeme.v
- pisir.v
- pizza.v

**1. [40 Puan] Matris Çarpıcı (Verilog Davranışsal Modelleme)**

Oluşturacağınız modüle “matriscarpici”( oluşacak dosya “matriscarpici.v”) ismini verin.

İstenen modülü Verilog dilinde **davranışsal modelleme** ile gerçekleştirin.

**Devrenin parametre girişi:**

**M:** A ve B matrislerinin her bir elemanının bit genişliğini gösterir ( $4 \leq M \leq 16$ )

**Devrenin girişleri:**

**clk:** (1 bit) saat girişi

**rst:** (1 bit) reset girişi (sinyal 1 olunca saat ile senkron şekilde resetler)

**matris\_veri:** (M bit) Matris eleman girişi

**matris\_gecerli:** (1 bit) **matris\_veri**’nin geçerli sinyal girişi

**Devrenin çıkışları:**

**carpim\_veri:** (? bit) Matris çarpım çıkışı

**carpim\_gecerli:** (1 bit) **carpim\_veri**’nin geçerli sinyal çıkışı

**Not:** ? bit **carpim\_veri** çıkışının boyutunu sizin bulmanız beklenmektedir. C matrisi elemanları olası tüm çarpım sonuçlarını gösterebilmelidir.

**Not:** Eğer devrenizi parametrik tasarlamıyorsanız kısmi puan alabilmek için soruyu  $M = 8$  olarak çözün.

Tasarlayacağınız modül A ve B matrisleri çarpıp sonucunda oluşan C matrisini çıktı olarak verecektir. A matrisi 2 x 4 (satır x sütun), B matrisi 4 x 2 (satır x sütun) boyutlarında olacaktır (matris boyutları hiçbir zaman değişmeyecek). Devrenize **rst** uygulandıktan sonra A ve B matrisinin elemanları **satır sıralı** olarak (önce 1. satır, sonra 2. satır, her satır içerisinde elemanlar soldan sağa sıralı gelecek şekilde) **matris\_veri** girişinden verilecektir. Sizden A ve B matrislerinin tüm elemanları alındıktan sonra oluşan C matrisini yine **satır sıralı** şekilde çıkışta vermeniz gerekmektedir.

A ile B matrislerinin elemanları okunurken geçerli eleman girişte bulunduğu çevrimlerde **matris\_gecerli** sinyali 1 olacaktır. Devreniz bu sinyalin aktif olmadığı çevrimlerdeki elemanları değerlendirmemelidir. Benzer şekilde sizden çarpımınızı bitirip C matrisinin elemanlarını çıkışınıza verirken **carpim\_gecerli** sinyalini de 1 olarak vermeniz beklenmektedir. Elemanlar okunurken ve verilirken ardışık çevrimde alınması/verilmesi zorunluluğu **yoktur**. Çarpımınızı istediğiniz şekilde ve çevrim sayısında yapabilirsiniz.

### Örnek:

$$\begin{bmatrix} 2 & 3 & 4 & 0 \\ 1 & 1 & 9 & 8 \end{bmatrix} \times \begin{bmatrix} 2 & 0 \\ 9 & 3 \\ 1 & 4 \\ 6 & 1 \end{bmatrix} = \begin{bmatrix} 35 & 25 \\ 68 & 47 \end{bmatrix}$$

Devrenize elemanlar aşağıdaki sırada gelmektedir:

2, 3, 4, 0, 1, 1, 9, 8, 2, 0, 9, 3, 1, 4, 6, 1

Görseldeki A (2x4) ve B (4x2) matrisin çarpımı aşağıdaki gibi yapılmaktadır:

$$C[1,1] = 2 \times 2 + 3 \times 9 + 4 \times 1 + 0 \times 6 = 35$$

$$C[1,2] = 2 \times 0 + 3 \times 3 + 4 \times 4 + 0 \times 1 = 25$$

$$C[2,1] = 1 \times 2 + 1 \times 9 + 9 \times 1 + 8 \times 6 = 68$$

$$C[2,2] = 1 \times 0 + 1 \times 3 + 9 \times 4 + 8 \times 1 = 47$$

Devrenizden elemanlar aşağıdaki sırada çıkmalıdır:

35, 25, 68, 47

## 2. [60 puan] Pizza (Verilog Davranışsal Modelleme)

Bu kısımda 4 adet modül oluşturacaksınız. Modüllerinize “hamur”, “malzeme”, “pisir” ve “pizza” isimlerini verin (oluşacak dosyalar: hamur.v, malzeme.v, pisir.v, pizza.v). Modüllerinizi Verilog dilinde **davranışsal modelleme** ile gerçekleştirin. Modüllerin birbiriyle ilişkisini ve giriş çıkışlarını verilen şemada görebilirsiniz.

### a-) [15 puan] “hamur” modülü:

Pizza hamurunun hazırlandığı modüldür. Modül, her saat darbesinde giriş olarak un\_miktari, su\_miktari, tuz\_miktari ve maya girişlerini alır. un\_miktari ve su\_miktari çarpıp, üzerine tuz\_miktari ekleyerek hamurun ağırlığını elde eder. Eğer hamur ağırlığı; mayalıysa 10000 (mayasızsa 8000) ve üzeri ise hamur kalın(2), mayalıysa 5000 (mayasızsa 4000) ve üzeri ise hamur orta(1), altında ise hamur ince(0) olur ve kalinlik çıkışına verir. Eğer hamurda maya kullanıldıysa mayalı çıkışına 1 verir. Eğer tuz\_miktari 5 ve 5’ten fazlaysa tuzlu çıkışına 1 verir. Modül, basla sinyali geldiğinde işleme başlar ve bir saat darbesinde kendi işlemini bitirir. (Birimler önemsizdir.)

#### Devrenin girişleri:

**saat:** 1 bitlik saat sinyali

**reset:** 1 bitlik reset sinyali (senkron)

**basla:** 1 bitlik devrede işlemi başlatan giriş sinyali

**un\_miktari:** 6 bitlik hamurda kullanılacak un miktarını belirten giriş sinyali  
**su\_miktari:** 8 bitlik hamurda kullanılacak su miktarını belirten giriş sinyali  
**tuz\_miktari:** 3 bitlik hamurda kullanılacak tuz miktarını belirten giriş sinyali  
**maya:** 1 bitlik hamurun mayalı(1) mı mayasız(0) mı olacağını belirten giriş sinyali

**Devrenin çıkışları:**

**kalinlik:** 2 bitlik hamurun kalınlığını (ince(0), orta(1), kalın(2)) belirten çıkış sinyali  
**mayali:** 1 bitlik hamurun mayalı(1) mı mayasız(0) mı olduğunu belirten çıkış sinyali  
**tuzlu:** 1 bitlik hamurun tuzlu olup olmadığını belirten çıkış sinyali  
**bitti:** 1 bitlik devrenin çalışmasını bitirdiğini belirten çıkış sinyali

**b-) [15 puan] “malzeme” modülü:**

Pizzaya malzemelerin koyulduğu modüldür. “**malzemeler.mem**” dosyasında bulunan 2 adet 10 bitlik sayı dosyadan okunarak bir hafızada tutulur. İlk sayı 1. malzemeleri, ikinci sayı 2. malzemeleri temsil eder. Bu iki sayıdaki 1’ler pizzada olması istenen malzemeleri, 0’lar olmaması istenen malzemeleri temsil eder ve sayılardaki 1’ler sayılır. 1 sayısı çok olan sayı seçilir ve secilen\_malzeme çıkışına; ilki seçildiyse 0, ikincisi seçildiyse 1 verilir. (1 sayısı eşit olan sayılar için ilk malzemeyi(0) seçin.) Sayılan 1 sayısı ise malzeme miktarını temsil ettiği için malzeme\_miktari çıkışına verilecektir ancak eğer giriş olarak gelen sos da eklenecekse(1 ise) 1 fazlası malzeme\_miktari çıkışına verilir. Hamur tuzlu ise; pizzaya malzeme koyulmaz ve malzeme\_miktari çıkışına 0 verilir, tuzlu çıkışına yine 1 verilir. (Bu durumda secilen\_malzeme çıkışına ne verildiği önemli değil.) Modül, basla sinyali geldiğinde işleme başlar ve bir saat darbesinde kendi işlemini bitirir. (malzemeler.mem dosyasındaki veriler binary şekilde ve her satırda bir veri olacak şekilde tutulmaktadır.) (İpucu: \$readmemb(“dosya.mem”, hafiza\_reg); komutu dosya.mem binary bellek dosyasını hafiza\_reg değişkenine okumayı sağlar.)

**Devrenin girişleri:**

**saat:** 1 bitlik saat sinyali  
**reset:** 1 bitlik reset sinyali (senkron)  
**basla:** 1 bitlik devrede işlemi başlatan giriş sinyali  
**sos:** 1 bitlik pizza sosu koyulup(1) koyulmayacağını(0) belirten giriş sinyali  
**tuzlu:** 1 bitlik malzeme koyulacak hamurun tuzlu olup olmadığını belirten giriş sinyali

**Devrenin çıkışları:**

**secilen\_malzeme:** 1 bitlik seçilen malzemeyi belirten çıkış sinyali  
**malzeme\_miktari:** 4 bitlik malzeme miktarını belirten çıkış sinyali  
**tuzlu:** 1 bitlik hamurun tuzlu olup olmadığını belirten çıkış sinyali  
**bitti:** 1 bitlik devrenin çalışmasını bitirdiğini belirten çıkış sinyali

**c-) [15 puan] “pisir” modülü:**

Pizzanın pişirildiği modüldür. Eğer hamur mayalı ise kabarır ve kabarık çıkışına 1 verilir. Tuzlu ise pizza pişirilmez, tuzlu çıkışına 1 verilir, bu durumda kabarık çıkışı ise zaten pizza pişirilmeyeceği için 0 olur. Eğer pizza pişirildiyse pizza sayısına eklenir, pizza\_sayisi çıkışından toplam pizza sayısı verilir. Toplam 100 pizzaya ulaştıktan sonra pizzacı için gün bitmiş demektir ve yeni pizza pişirilmez.(100 pizzaya ulaştıktan sonra pizza\_sayisi olarak 100 vermeye devam eder, pizza\_sayisi başlangıçta 0’dan başlar.) Modül, basla sinyali geldiğinde işleme başlar ve bir saat darbesinde kendi işlemini bitirir. Her bir saat darbesinde sadece bir pizza pişirebilir.

**Devrenin girişleri:**

**saat:** 1 bitlik saat sinyali  
**reset:** 1 bitlik reset sinyali (senkron)  
**basla:** 1 bitlik devrede işlemi başlatan giriş sinyali  
**mayali:** 1 bitlik hamurun mayalı(1) mı mayasız(0) mı olduğunu belirten giriş sinyali  
**tuzlu:** 1 bitlik pişirilecek hamurun tuzlu olup olmadığını belirten giriş sinyali

**Devrenin çıkışları:**

**kabarık:** 1 bitlik pişirilen hamurun kabarık olup(1) olmadığını(0) belirten çıkış sinyali  
**tuzlu:** 1 bitlik hamurun tuzlu olup olmadığını belirten çıkış sinyali  
**pizza\_sayisi:** 7 bitlik toplam pizza sayısını belirten çıkış sinyali  
**bitti:** 1 bitlik devrenin çalışmasını bitirdiğini belirten çıkış sinyali

**d-) [15 puan] “pizza” modülü:**

Aşağıda basitleştirilmiş şekilde görüleceği üzere, pizza üst seviye bir modül olacak ve yazdığınız ilk 3 modülü kullanacaktır. Modüller; **hamur**→**malzeme**→**pisir** sırasıyla çalışacaktır ve her saat vuruşunda bu farklı modüller **aynı anda** çalışabilmelidir (gelen girişlerin farklı aşamalarındaki işlemlerini aynı anda yapabilmeliler). **İlk 3 çevrimden sonra** sürekli olarak girişlere uygun kalınlık, secilen\_malzeme, malzeme\_miktari, kabarık, tuzlu ve pizza\_sayisi çıkışları üretilecektir (0. çevrimde gelen girişlerin sonucuna ait çıkışlar 3. çevrimde, 1. çevrimde gelen girişlerin sonucuna ait çıkışlar 4. çevrimde, 2. çevrimde gelen girişlerin sonucuna ait çıkışlar 5. çevrimde çıkmalı, ...). Ayrıca, şekilde de göreceğiniz üzere, bu modülde aynı çevrimde gelen sinyallerin senkronizasyonu için; sos girişini malzeme modülüne vermeden önce 1 çevrim, hamur modülünün mayalı çıkışını pisir modülüne vermeden önce 1 çevrim, kalınlık çıkışını pizza modülünün kalınlık çıkışına vermeden önce 2 çevrim, malzeme modülünün secilen\_malzeme ve malzeme\_miktari çıkışlarını pizza modülünün çıkışlarına vermeden önce 1 çevrim bekletmeniz gerekmektedir. reset sinyali geldiğinde, saat ile senkron şekilde, alt modüller de dahil bütün devrenin durumunu sıfırlamanız gerekmektedir.

**Devrenin girişleri:**

**saat:** 1 bitlik saat sinyali

**reset:** 1 bitlik reset sinyali (senkron)

**un\_miktari:** 6 bitlik hamurda kullanılacak un miktarını belirten giriş sinyali

**su\_miktari:** 8 bitlik hamurda kullanılacak su miktarını belirten giriş sinyali

**tuz\_miktari:** 3 bitlik hamurda kullanılacak tuz miktarını belirten giriş sinyali

**maya:** 1 bitlik hamurun mayalı(1) mı mayasız(0) mı olacağını belirten giriş sinyali

**sos:** 1 bitlik pizza sosu koyulup(1) koyulmayacağını(0) belirten giriş sinyali

**Devrenin çıkışları:**

**kalınlık:** 2 bitlik hamurun kalınlığını (ince(0), orta(1), kalın(2)) belirten çıkış sinyali

**secilen\_malzeme:** 1 bitlik seçilen malzemeyi belirten çıkış sinyali

**malzeme\_miktari:** 4 bitlik malzeme miktarını belirten çıkış sinyali

**kabarık:** 1 bitlik pişirilen hamurun kabarık olup(1) olmadığını(0) belirten çıkış sinyali

**tuzlu:** 1 bitlik hamurun tuzlu olup olmadığını belirten çıkış sinyali

**pizza\_sayisi:** 7 bitlik toplam pizza sayısını belirten çıkış sinyali

