



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

24 Ekim 2023
BİL 265/264L - Mantıksal Devre
Tasarımı ve Laboratuvarı
2023 - 2024 Öğretim Yılı
Güz Dönemi
Lab 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 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 kırılabacaktır.
9. Dosya gönderimi için sorularda belirtilen “.v” uzantılı dosyalarınızı “isim_soyisim_numara_labsinavi” 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.)

- tamtoplayici.v
- toplayici4bit.v
- hammingbenzer4bit.v
- hammingbenzer16bit.v
- rotadogrula.v
- seyahatdogrula.v

1. [50 Puan] Hamming Benzerliği (Verilog Kapı Seviyesi)

a) [5 Puan] Oluşturacağınız modüle “tamtoplayici” (oluşacak dosya “tamtoplayici.v”) ismini verin. İstenen modülü Verilog dilinde **kapı seviyesinde** tasarım kullanarak gerçekleştirin.

Yazacağınız modülün giriş ve çıkışları aşağıdaki gibidir:

A: (1 bitlik) A girişi

B: (1 bitlik) B girişi

Cin: (1 bitlik) Elde girişi

S: (1 bitlik) Toplam sonucu çıkışı

Cout: (1 bitlik) Elde çıkışı

Tasarlayacağınız modül $A + B + Cin$ (ikilik tabanda 1 bitlik sayı ve elde toplama) işlemini gerçekleştirip toplam sonucunu S çıkışına, elde sonucunu ise Cout çıkışına vermelidir.

Örnek: $A = 1$, $B = 1$ ve $Cin = 1$ girişleri için $S = 1$ ve $Cout = 1$ olmalıdır.

b) [10 Puan] Oluşturacağınız modüle “toplayici4bit” (oluşacak dosya “toplayici4bit.v”) ismini verin. İstenen modülü Verilog dilinde **kapı seviyesi** kullanarak gerçekleştirin.

Yazacağınız modülün giriş ve çıkışları aşağıdaki gibidir:

A: (4 bitlik) İlk sayı girişi

B: (4 bitlik) İkinci sayı girişi

Cin: (1 bitlik) Elde girişi

S: (4 bitlik) Toplam sonucu çıkışı

Cout: (1 bitlik) Elde çıkışı

Tasarlayacağınız modül $A + B + Cin$ (ikilik tabanda 4 bitlik sayı ve elde toplama) işlemini gerçekleştirip toplam sonucunu S çıkışına, elde sonucunu ise Cout çıkışına vermelidir.

Örnek: $A = 0100$, $B = 0100$, $Cin = 1$ girişleri için $S = 1001$, $Cout = 0$ olmalıdır.

c) [10 Puan] Oluşturacağınız modüle “hammingbenzer4bit” (oluşacak dosya “hammingbenzer4bit.v”) ismini verin.

İstenen modülü Verilog dilinde **kapı seviyesi** kullanarak gerçekleştirin.

Yazacağınız modülün giriş ve çıkışları aşağıdaki gibidir:

A: (4 bitlik) İlk sayı girişi

B: (4 bitlik) İkinci sayı girişi

HB: (3 bitlik) Hamming benzerliği çıkışı

Tasarlayacağınız modül 4 bitlik A ve B girişlerindeki sayılar arasındaki **hamming benzerliğini** bulup çıkışa vermelidir. **Hamming benzerliği** iki sayının karşılıklı indislerinde eşleşen bit sayısı olarak tanımlanmıştır.

Örnek: $A = 1001$ sayısı ile $B = 1011$ sayılarının en anlamlı 2 biti ile en anlamsız 1 bit eşleştiği için HB (hamming benzerliği) 3 bulunur.

d) [25 Puan] Oluşturacağınız modüle “hammingbenzer16bit” (oluşacak dosya “hammingbenzer16bit.v”) ismini verin.

İstenen modülü Verilog dilinde **kapı seviyesi** kullanarak gerçekleştirin.

Yazacağınız modülün giriş ve çıkışları aşağıdaki gibidir:

A: (16 bitlik) İlk sayı girişi

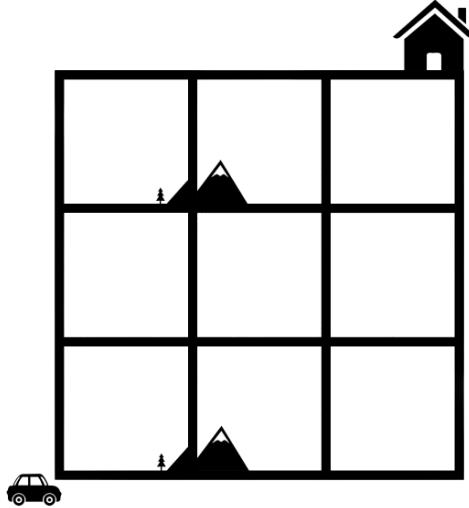
B: (16 bitlik) İkinci sayı girişi

HB: (5 bitlik) Hamming benzerliği çıkışı

Tasarlayacağınız modül 16 bitlik A ve B girişlerindeki sayılar arasındaki **hamming benzerliğini** bulup çıkışa vermelidir. **Hamming benzerliği** iki sayının karşılıklı indislerinde eşleşen bit sayısı olarak tanımlanmıştır.

2. [50 Puan] Manhattan Yolculuğu (Verilog Kapı Seviyesi)

2050 yılında Mantıksal Devre Tasarımı sınavını başarıyla tamamlayan bir grup öğrenci kutlama yapmaya gitmek istiyor. Ancak arabalarına bindiklerinde navigasyonlarının bozulduğunu ve hesapladıkları rotaların doğruluğunu öğrenemediklerini görüyorlar. Bu dersi yıllar önce başarıyla tamamlamış olan büyükleri olarak onlara yardım etmelisiniz.



Verilen haritada araba (sol alt köşe) başlangıç noktası, dağların bulunduğu köşeler engeller ve ev (sağ üst köşe) hedef noktasıdır.

Haritanın boyutu ile engeller, başlangıç noktası ve hedef nokta **her zaman sabit olacaktır**.

a) [35 Puan] Oluşturacağınız modüle “rotadogrula” (oluşacak dosya “rotadogrula.v”) ismini verin.

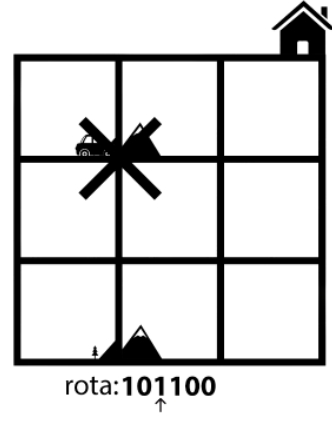
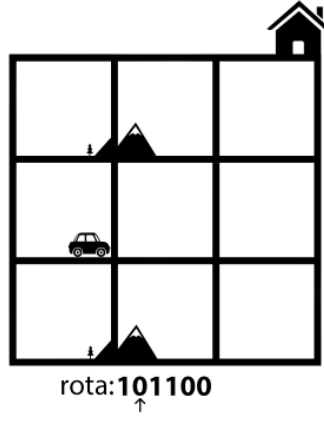
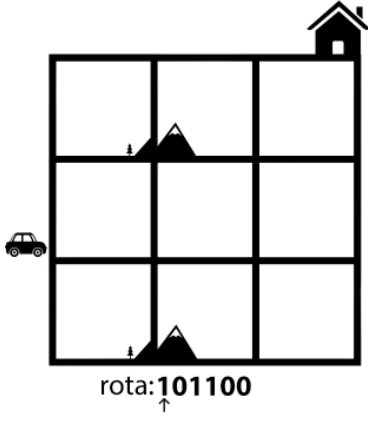
İstenen modülü Verilog dilinde **kapı seviyesinde** tasarım kullanarak gerçekleştirin.

Yazacağınız modülün giriş ve çıkışları aşağıdaki gibidir:

rota: (6 bitlik) Rota girişi

rota_dogru: (1 bitlik) Verilen rotanın hedefe ulaştığını belirten çıkış

6 bitlik **rota** girişi arabanın önerilen rotasını belirtmektedir. Araba bu girişin **en anlamlı bitinden** başlayarak sırayla her bit için bir hareket yapacaktır. İşlem sırasında **0** biti görülürse araba bir birim **sağa** gidecek, **1** biti görülürse araba bir birim **yukarı** gidecek.



Tasarladığınız devre araba ilerlerken **herhangi bir engele çarpıyorsa** (dağ olan bir köşeye uğramıyorsa) ve son hamlesi (en anlamsız biti) işlendikten sonra **hedef noktada duruyorsa** (rota ev olan köşede bitiyorsa) rota_dogru çıkışına 1 vermeli. Diğer tüm durumlarda devrenizin rota_dogru çıkışı 0 olmalıdır.

b) [15 Puan] Oluşturacağınız modüle “seyahatdogrula” (oluşacak dosya “seyahatdogrula.v”) ismini verin. İstenen modülü Verilog dilinde **kapı seviyesinde** tasarım kullanarak gerçekleştirin.

Genç arkadaşlarınızın rotalarını başarılı bir şekilde doğruladınız ancak sizlere bu seyahat için yeterli yakıtları olup olmadığından emin olamadıklarını bildirdiler. Bu nedenle arabanın yakıtına ve rotaya göre seyahatin gerçekleştirilip gerçekleştirilemeyeceğini hesaplayan devreyi tasarlamalısınız. Rotayı doğrulamak için bir önceki şıkta tasarladığınız devreyi kullanın.

Yazacağınız modülün giriş ve çıkışları aşağıdaki gibidir:

yakit: (4 bitlik) Yakıt girişi

rota: (6 bitlik) Rota girişi

seyahat_dogru: (1 bitlik) Verilen rotanın hedefe ulaştığını belirten çıkış

Bir önceki çözdüğünüz soruya (a şıkkı) ek olarak 4 bitlik **yakit** girişiniz bulunmakta. Ek olarak, arabanın her birim hareketi için 2 birim yakıt bulundurması gerektiğini biliyorsunuz.

Tasarladığınız devre **verilen rota doğru olduğunda** (önceki soru ile aynı kurallar) ve **hedefe ulaşmak için yeterli yakıt bulunduğunda**, seyahat_dogru çıkışına 1 vermeli. Diğer tüm durumlarda devrenizin seyahat_dogru çıkışı 0 olmalıdır.

Örnek: yakıt = 0010 (yani 2) olduğunda araba sadece 1 birim hareket edebileceği için seyahat_dogru = 0 olacaktır.