



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

3 Ağustos 2024
BİL 265/264/264L - Mantıksal
Devre Tasarımı ve Laboratuvarı
2023 - 2024 Öğretim Yılı
Yaz Dönemi
Final Sınavı

AÇIKLAMALAR:

1. Sınavı çözmeye başlamadan önce tüm açıklamaları ve soruları okuyun. Sınavda toplam 5 sayfa, 2 soru var ve soruların toplam değeri **110** puandır. Bütün soruların değeri köşeli ayraç ile belirtilmiştir. Sınav süresi 150 dakikadır.
2. Sınav sırasında kodlarınızla alakalı 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 her türlü araç/gereç ve kaynak kullanımı yasaktır. (hesap makinesi, akıllı saat, telefon, pdf dosyaları vb.)
6. Sonucu yanlış olan yanıtlar puan alamayabilir. 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_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.)

- penguen.v
- penguenler.v
- sifreleme.v
- garip_sifreleme.v

1. [50 Puan] Balık Avlayan Penguen

Bu kısımda 2 adet modül oluşturacaksınız. Modüllerinize “penguen” ve “penguenler” isimlerini verin (oluşacak dosyalar: penguen.v, penguenler.v). Modüllerinizi Verilog dilinde davranışsal modelleme ile gerçekleştirin.

a) [30 Puan] Bir penguen avlanmaktadır. Bunu *penguen.v* isminde bir modülle gerçekleyeceksiniz. Midesi yeterince dolunca avlanmasını bitirip yavrusunu besleyecektir. Her çevrim 3 bitlik *avlanılan_balik* girişinden o çevrimde avlanılan balık sayısı gelmektedir. Her çevrim gelen balık sayısı penguenin midesine eklenmektedir. Penguen her üç çevrimde bir avladığı bu balıkların 3’ünü tüketmektedir ve midesindeki balık sayısı 3 azalmaktadır. Penguen midesinde 25 veya daha fazla balık olduğu zaman avını bitirecektir. 25 balık veya daha fazla balık olduğu çevrim 3 balığı tüketmez. Avı bittiği zaman *bitti* isimli tek bitlik bir çıktı 1 olacaktır ve kaç çevrim geçtiğini *bitme_suresi* isimli çıktı ile verecektir. Burada çıktıya 7 bitin yeteceğini varsayabilirsiniz. Bittikten sonra *bitti* ve *bitme_suresi* çıkışları sabit kalacaktır. Modülünüzde saat ile senkron çalışan ve aktif-1 olan bir *reset* sinyali bulunmalıdır. *reset* sinyali geldiğinde devrenin durumunu sıfırlamanız gerekmektedir.

Devrenin girişleri:

saat: 1 bitlik saat sinyali

reset: 1 bitlik reset sinyali (senkron, aktif-1)

avlanan_balik: 3 bitlik kaç balık avlandığını gösteren giriş sinyali

Devrenin çıkışları:

bitti: 1 bitlik avlanmanın bittiğini gösteren çıkış sinyali

bitme_sure: 7 bitlik kaç çevrimde avlanmanın bittiğini gösteren çıkış sinyali

Çevrim:	0	1	2	3	4	5	6	7	8
reset	1	0	0	0	0	0	0	0	0
avlanan_balik	X	111	111	111	000	001	111	X	X
mide	X	0	7	14	18 (21-3)	18	19	26*	X
bitti	0	0	0	0	0	0	0	1	1
bitme_sure	X	X	X	X	X	X	X	6	6

* : Avlanan balık sayısı 25’i geçtiği için 3 balık tüketilmez.

b) [20 Puan] Bu kısmı a şıkında yazdığınız modülü kullanarak yapmanız beklenmektedir. Bu modülün ismi *penguenler.v* olmalıdır. Bu modül avlanmaya çıkan 5 penguen ile ilgili bazı işlemler gerçekleştirir. Devreye 15 bitlik *avlanan_balik* verisi gelecektir. Sırasıyla 1. penguen en anlamsız kısımdan başlayarak 3 bitlik sayı kadar balık avlayacaklardır (5 tane 3 bitlik veri aralarında dağılacak). Yani 1. Penguen için *avlanan_balik*[2:0] yediği balık sayısını gösterirken 2. penguen için *avlanan_balik*[5:3] gösterecektir. Hepsinin avlanması bitince *bitti* isimli çıkış 1 olacaktır. Aynı zamanda en kısa süren, en uzun süren ve 5 penguenin sürelerinin ortalamasını veren *en_kisa*, *en_uzun*, *ortalama* isimli 7 bitlik 3 çıkış olacaktır. Hangi penguenin en uzun ve en kısa sürede bitirdiğini gösteren 3 bitlik *hizli_penguen* ve *yavas_penguen* isimli çıktılar da olacaktır. Burada penguen numaraların 0’dan değil 1’den başlayıp 5’te bittiğine dikkat ediniz. Eğer eşitlik durumu olursa en yüksek numaraya sahip pengueni çıktı olarak veriniz. *bitti* isimli çıktı 1 olduğunda bu çıktılar verilecektir ve öncesinde bu çıkışların değerleri önemli değildir. Saat ile senkron, aktif-1 *reset* sinyali geldiğinde devrenin durumunu sıfırlamanız gerekmektedir.

Not: Ortalama virgüllü bir değer çıkarsa aşağıya yuvarlayın.

Devrenin girişleri:

saat: 1 bitlik saat sinyali

reset: 1 bitlik reset sinyali (senkron)

avlanan_balik: 15 bitlik avlanılan balık sayılarını belirten giriş sinyali

Devrenin çıkışları:

- bitti:** 1 bitlik tüm penguenlerin avının bittiğini gösteren çıkış sinyali
- en_kisa:** 7 bitlik en kısa sürede avı biten penguenin avının bitiş süresini gösteren çıkış sinyali
- en_uzun:** 7 bitlik en uzun sürede avı biten penguenin avının bitiş süresini gösteren çıkış sinyali
- ortalama:** 7 bitlik tüm penguenlerin avlarının bitiş sürelerinin ortalamasını gösteren çıkış sinyali
- hizli_penguen:** 3 bitlik en kısa sürede bitiren penguenin numarasını gösteren (1'den 5'e kadar olduğuna dikkat edin) çıkış sinyali
- yavas_penguen:** 3 bitlik en uzun sürede bitiren penguenin numarasını gösteren çıkış sinyali

2. [60 Puan] Bir Garip Şifreleme

Bu kısımda 2 adet modül oluşturacaksınız. Modüllerinize “sifreleme” ve “garip_sifreleme” isimlerini verin (oluşacak dosyalar: sifreleme.v, garip_sifreleme.v). Modüllerinizi Verilog dilinde davranışsal modelleme ile gerçekleştirin.

Not: Eğer devrelerinizi parametrik olarak tasarlayamazsanız kısmi puan alabilmek için $BIT = 4$ olarak çözün.

a-) [30 Puan] “sifreleme” modülü:

sifreleme modülünün iki adet modu bulunmaktadır. Bu iki moda göre BIT bitlik giriş olarak alınan veri üzerinde gerekli şifrelemeyi aşağıda verilen şifreleme tablosunu kullanarak gerçekleştirir.

Devrenin parametre girişi:

BIT: Veri ve anahtar bit genişliğini gösterir (varsayılan değer: 4) ($2 \leq BIT \leq 64$)

Devrenin girişleri:

- saat:** 1 bitlik saat sinyali
- reset:** 1 bitlik reset sinyali (senkron)
- basla:** 1 bitlik devrede işlemi başlatan giriş sinyali
- mod:** 1 bitlik şifrelemenin nasıl yapılacağı belirten giriş sinyali
- veri:** BIT bitlik şifrelenecek veri giriş sinyali
- secim:** 3 bitlik seçilecek anahtarı belirten giriş sinyali (Örn.: 000 ise şifreleme tablosunun ilk satırının BIT bitlik (BIT kadarlık) en anlamsız kısmı seçilir.)

Devrenin çıkışları:

- bit_cikisi:** 1 bitlik şifrelenen verinin bit bit dışarı verildiği çıkış sinyali
- gecerli:** 1 bitlik **bit_cikisi**'nin dışarı verildiğini belirten çıkış sinyali

İsterler şu şekildedir:

- **mod** 1 ise veri ile şifreleme tablosundan elde edilen değer xor'landıktan sonra 2 bitlik dairesel sola kaydırma işlemi yapılarak **BIT** bitlik şifrelenmiş değer elde edilir.
- **mod** 0 ise veri ile şifreleme tablosundan elde edilen değer xnor'landıktan sonra 2 bitlik dairesel sağa kaydırma işlemi yapılarak **BIT** bitlik şifrelenmiş değer elde edilir.
- **secim** girişinden gelen 3 bitlik değer ile şifreleme tablosundaki satır seçilir ve ilgili satırdaki değer en anlamsız kısmından başlayarak **BIT** bitlik değer şifreleme için kullanılır.
- **basla** sinyalinin 1 olduğu çevrim işleme başlanır. **basla** sinyalinin 1 olduğu çevrim ise şifreleme gerçekleşir.
- Sonraki çevrimler ise şifrelenmiş değer en anlamsız kısmından başlayarak her çevrim 1 bit dışarıya verilecek şekilde **bit_cikisi** sinyali dışarıya verilir.
- Şifrelenmiş değer her bir bitinin dışarıya verildiği çevrimler **gecerli** biti 1 olarak dışarıya verilir.

- **basla** sinyalinin 1 olduğu çevrimde gerçekleşen şifreleme ve **BIT** çevrim boyunca şifrelenmiş verinin dışarıya verilmesi olmak üzere toplamda **BIT+1** çevrim sonunda modülün tekrardan girdi alabilir duruma geçmesi gerekmektedir.
- **reset** sinyali geldiğinde, saat ile senkron şekilde devrenin durumunu sıfırlamanız gerekmektedir.
- Aşağıdaki tabloyu **sifreler** adlı bir değişkende (register) tutun ve şifreleri buradan alın. (**reg [63:0] sifreler [0:7]**)

Seçim	Şifreleme Değeri
0	0x BABA_1453_DEDE_1071
1	0x ACAB_0909_BACA_0707
2	0x ADAB_0606_DADA_0505
3	0x AAAA_0000_FFFF_5555
4	0x CAAA_0101_CAAA_0101
5	0x AACA_0606_AACA_0606
6	0x CAAA_1717_CAAA_1717
7	0x AAAA_0000_FFFF_5555

Örnek:

BIT = 4

mod = 1

veri = 1011

secim = 101 (5 ise şifreleme tablosundaki 6. satırdaki değerin en anlamsız ilk 4 biti gelmeli.)

6. satırdaki **en anlamsız 4 bit** = 0110

Şifreleme (mod = 1):

xor: 1011 ^ 0110 = 1101

2 bit dairesel sola kaydırma: **1101** → **0111**

Sonuç olarak 1011 şifrelenmiş verisi 4 (BIT) çevrim boyunca en anlamsız bitinden başlanarak bit bit dışarı verilmeli, toplamda 5 (BIT+1) çevrim sürmeli (yukarıdaki diğer girişlerin bu sürece sabit olduğunu düşünebilirsiniz):

basla = 0; bit_cikisi = X; gecerli = 0;

basla = 1; bit_cikisi = X; gecerli = 0;

basla = 0; bit_cikisi = **1**; gecerli = 1;

basla = 0; bit_cikisi = **1**; gecerli = 1;

basla = 0; bit_cikisi = **1**; gecerli = 1;

basla = 0; bit_cikisi = **0**; gecerli = 1;

basla = 0; bit_cikisi = X; gecerli = 0;

...

...

basla = 1 → yeni şifreleme işini yapmaya başla

b-) [30 Puan] “garip_sifreleme” modülü:

Bu modül, a şıkında tasarlanmış olduğunuz **sifreleme** modülünden 2 adet kullanarak art arda boru hatlı şekilde şifreleme işlemi yapar.

Devrenin parametre girişi:

BIT: Veri ve anahtar bit genişliğini gösterir (varsayılan değer: 4) ($2 \leq \text{BIT} \leq 64$)

Devrenin girişleri:

saat: 1 bitlik saat sinyali

reset: 1 bitlik reset sinyali (senkron)

basla: 1 bitlik devrede işlemi başlatan giriş sinyali

mod: 1 bitlik şifrelemenin nasıl yapılacağı belirten giriş sinyali (ilk şifreleme birimi için aynı, ikincisi için değil)

veri: **BIT** bitlik şifrelenecek veri giriş sinyali (1. şifreleme modülünün veri girişi)

secim: 3 bitlik 1. şifreleme modülü için seçilecek anahtarı belirten giriş sinyali

Devrenin çıkışları:

bit_cikisi: 1 bitlik (2 kere şifrelenmiş verinin bit bit dışarı verildiği çıkış sinyali)

gecerli: 1 bitlik **bit_cikisi**'nin dışarı verildiğini belirten çıkış sinyali

İsterler şu şekildedir:

- **mod** girişi 1 ise 1. şifreleme modülü mod 1'de, 2. şifreleme modülü mod 0'da çalışır.
- **mod** girişi 0 ise 1. şifreleme modülü mod 0'da, 2. şifreleme modülü mod 1'de çalışır.
- **basla** sinyalinin 1 olduğu çevrim 1. şifreleme modülü işlemi başlatılmalıdır.
- 1. şifreleme modülünün **veri** girişi, **garip_sifreleme** modülüne gelen **veridir**.
- 1. şifreleme modülünün **secim** girişi, **garip_sifreleme** modülüne gelen **secimdir**.
- 2. şifreleme modülünün veri girişi, 1. şifreleme modülünden **BIT** çevrim boyunca dışarıya verilen **BIT** bitlik değerdir.
- 2. şifreleme modülünün **secim** girişi 1. şifreleme modülünden çıkan **BIT** bitlik değer en anlamsız 3 biti seçilerek oluşturulur.
- 2. şifreleme modülünün veri ve secim girişlerini oluşturmak için 1. şifreleme modülünden **BIT** çevrim boyunca gelen her bir biti saklamanız gerekmektedir.
- 1. şifreleme modülünden son **bit_cikisi** değerinin **gecerli** olduğu çevrim 2. şifreleme modülünün **basla** biti 1 yapılarak 2. şifreleme modülü aynı çevrimde işlemi başlatılmalıdır.
- 2. şifreleme modülünden çıkan **bit_cikisi** ve **gecerli** sinyalleri **garip_sifreleme** modülünün çıkışları ile aynı anda aynı şeyi vermelidir.
- **garip_sifreleme** modülü, 1. şifreleme modülü yeni bir işlem almaya hazır olduğu an yeni bir işlem alabilmelidir. Yani **garip_sifreleme** modülü yeni bir işleme başladıktan (**BIT**+1) çevrim sonra yani (**BIT**+2). çevrim yeni bir işlem alabilmelidir.
- **garip_sifreleme** modülüne **basla** sinyalinin geldiği çevrim dahil olmak üzere ve **garip_sifreleme** modülünden son **gecerli** bitin çıktığı çevrim de dahil olmak üzere bir işlemin bitmesi için toplamda ($2 \cdot \text{BIT} + 1$) çevrim gerekmektedir.
- **reset** sinyali geldiğinde, saat ile senkron şekilde, alt modüller de dahil bütün devrenin durumunu sıfırlamanız gerekmektedir.

