	TOBB Ekonomi ve Teknoloji Üniversitesi **Bilgisayar Mühendisliği Bölümü** **Elektrik ve Elektronik Mühendisliği Bölümü**	13 Nisan 2025 BİL 265/264/264L – Mantıksal Devre Tasarımı ve Laboratuvarı 2024 – 2025 Öğretim Yılı Bahar Dönemi Final Sınavı

AÇIKLAMALAR:

1. Sınavı çözmeye başlamadan önce tüm açıklamaları ve soruları okuyun. Sınavda toplam 6 sayfa (3 adet A4 kağıdı), 2 soru var ve soruların toplam değeri **100** puandır. Bütün soruların değeri köşeli ayraç ile belirtilmiştir. Sınav süresi 150 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.)

   - alarm.v
   - sayac.v
   - sayac_boru.v

1. [40 Puan] Alarm (Verilog Davranışsal Modelleme)

Oluşturacağınız modüle "alarm"(oluşacak dosya "alarm.v") ismini verin. İstenen modülü Verilog dilinde **davranışsal modelleme** kullanarak gerçekleştirin.

Bir sensör ağı üzerinde veri toplama ve işleme işlemleri gerçekleştiren bir sistem tasarlıyorsunuz. Bu sistemde, her saat darbesinde, giriş olarak 7 bitlik Celcius cinsinden pozitif bir sıcaklık değeri alınmaktadır. Modüldeki hesaplamalar ise Celcius veya Kelvin cinsinden yapılır. Eğer K parametresi 1 ise hesaplamalar Kelvin cinsinden, 0 ise hesaplamalar Celcius cinsinden yapılır. Celcius değerinin 273 fazlasının Kelvin değeri olduğu bilinmektedir. 4. saat darbesinden (4. ölçümden) itibaren başlayarak her saat darbesinde, geçmiş 4 ölçümün ortalama sıcaklığını hesaplayan ve bu ortalama sıcaklığın alarm eşiğini geçip geçmediğini belirleyen bir modül tasarlamanız gerekmektedir. Alarm eşiği, tek satırlık 7 bit değer içeren "esik.mem" dosyasından başlangıçta okunarak belirlenmektedir. Bu eşik değeri Celcius cinsindendir ve örneğin Celcius cinsinden 50 (0110010) ise ve son 4 ölçümün ortalaması 50'ye eşit veya 50'den büyükse alarm_cal çıkışına mantık-1, küçükse mantık-0 sürülmelidir. Bir reset sinyali geldiğinde, saat ile senkron şekilde sistemin durumunu sıfırlamanız gerekmektedir. (İpucu: $readmemb("dosya.mem", bellek_reg); komutu binary bellek dosyasını okumayı sağlar.)

Devrenin parametre girişi:

K: 0 ise Celcius, 1 ise Kelvin cinsinden hesaplama yapılıp dışarı verilmesi gerektiğini gösteren parametre girişi (varsayılan değer K=0)

Devrenin girişleri:

saat: 1 bitlik saat sinyali

reset: 1 bitlik reset sinyali (senkron)

sicaklik: 7 bitlik Celcius cinsinden verilen sıcaklık değeri

Devrenin çıkışları:

ortalama_sicaklik: (? bitlik) son 4 sıcaklık değerinin ortalamasını Celcius veya Kelvin cinsinden veren çıkış sinyali

alarm_cal: 1 bitlik son 4 sıcaklık ortalamasına göre alarm eşiğini geçip geçmediğini gösteren çıkış sinyali

Not: ortalama_sicaklik çıkışının kaç bit olduğunu sizin hesaplamanız gerekmektedir.

Örnek:

K=0, esik.mem dosyasından okunan **alarm eşiği 50** iken;

çevrim	1	2	3	4	5	6	7	8
sicaklik	32	45	39	48	70	55	0	…
ortalama_sicaklik	X	X	X	(32+45+39+48) /4 = 41	(45+39+48+70) /4 = 50	(39+48+70+55) /4 = 53	(48+70+55+0) /4 = 43	…
alarm_cal	0	0	0	0	1	1	0	…

Not: Görüleceği üzere küsüratlar önemsizdir çünkü tamsayı bölmesi yapılmaktadır.

2. [60 Puan] Sayaçlı Boru Hattı (Verilog Davranışsal Modelleme)

Bu kısımda 2 adet modül oluşturacaksınız. Modüllerinize "sayac", "sayac_boru" isimlerini verin (oluşacak dosyalar: sayac.v, sayac_boru.v). Modüllerinizi Verilog dilinde **davranışsal modelleme** ile gerçekleştirin. Modüllerin birbiriyle ilişkisini ve giriş çıkışlarını verilen şemada görebilirsiniz.

a-) [40 puan] "sayac" modülü:

sayac modülünde sürekli ileri ve geri bir örüntüde sayabilen ve belli şartlar altında saymayı bitiren bir sayaç devresi tasarlanacaktır. Modülün isterleri aşağıda verilmiştir.

- Eş zamanlı atamalar saatin yükselen kenarında yapılacaktır.
- Saat ile senkron şekilde "reset" sinyali geldiğinde modül başlangıç durumuna dönmelidir.
- Sayaç "basla" sinyali mantık-1 olunca başlar ve saymayı bitirdiği çevrim "hazir" sinyalini mantık-1 (bitirene kadar mantık-0) olarak sürer.
- Sayaç "hazir" sinyalinin "1" olduğu çevrimden sonra "basla" sinyali ile birlikte yeni girişler alabilmelidir.
- Sayaç sayma aşamasındayken "basla" sinyalinin ne olduğu önemli değildir, sadece her işlem başlangıcında "1" olması yeterlidir.
- Sayaç "basla" sinyalinin mantık-1 olduğu ilk çevrim "baslangic degeri"ni yükler ve verilen bu sayıdan itibaren saymaya başlar. Saymaya başlanan ilk çevrim başlangıç değeri "sonuc" çıkışından dışarı verilirken diğer çevrimlerde sayılan sayılar "sonuc" çıkışından dışarı verilmeye devam eder.
- **"miktar" sinyali "0" ise** sayma işlemi gerçekleşmez ve "basla" sinyalinin geldiği ilk çevrim "baslangic_degeri" "sonuc" çıkışından verilirken "hazir" sinyali "1" yapılır. **"miktar" sinyali "1" ise sadece verilen "yon"de** "1" kadar sayılır. miktar sinyali "2" veya daha büyükse ileri-geri ya da geri-ileri bir örüntüde sayma işlemi gerçekleştirilir.
- "yon" sinyali sayacın başlangıç yönünü ve hangi yönde "miktar" kadar sayılacağını belirler ve sayaç bir ileri bir geri (miktar>=2 için) sayar. Örneğin "yon" mantık-1 ise ileri-geri-ileri-geri-... şeklinde bir örüntüde sayılacak iken, mantık-0 ise geri-ileri-geri-ileri-... şeklinde bir örüntüde sayılır.
- Sayacın verilen yönde ne kadar sayacağı "miktar" sinyali tarafından belirlenirken **tersi yönde 1 kadar** saymalıdır. Örneğin 0'dan başlanan, yönün "1" (ileri) ve "miktar" değerinin "2" olduğu durumda sayaç 0-2-1-3-2-4-3-5-... şeklinde yani 2 ileri 1 geri şekilde sayar. Yönün "0", başlangıç değerinin "255" ve "miktar" değerinin yine "2" olduğu durumda ise 255-253-254-252-253-251-252-250-... şeklinde yani 2 geri 1 ileri şekilde sayar. Sayaç "miktar" değerinden bağımsız olarak tersi yönde 1 kadar (miktar>=2 için) sayar.
- Sayaç 8 bitlik olduğundan 0'a ya da 255'e kadar sayabilir. Dolayısıyla sayaç ileriye doğru (yön 1 iken) sayarken sayabildiği 255'e en yakın değerde, geriye doğru (yön 0 iken) sayarken sayabildiği 0'a en yakın değerde (0 ve 255 dahil) sayma işlemini bitirir, bu değerleri aşmaz ve aynı çevrimde "sonuc" cikisindan bu değeri dışarı verirken "hazir" sinyalini de mantık-1 yapar.
- Sayaç sayma işlemine başladığından (basla=1) saymayı bitirene kadar geçen sürede "mesgul" çıkışına mantık-1 verir, diğer durumlarda ve başlangıçta sayaç meşgul değildir. Sayma işlemi bittikten sonraki çevrim -"hazir" sinyalinin "1" olduğu çevrimden sonraki çevrim- "mesgul" sinyali mantık-0 yapılır. "mesgul" sinyalinin 0 olduğu çevrimden sonraki çevrim "basla" sinyali ile birlikte yeni bir işlem alınabilmelidir.
- "cikis_miktar" direkt olarak giriş "miktar" sinyaline bağlı olmalıdır.
- "cikis_yon" direkt olarak giriş "yon" sinyaline bağlı olmalıdır.
- "baslangic_degeri", "yon" ve "miktar" girişlerinin sayma işlemi bitene kadar aynı değerde kalacağını varsayın.

Devrenin girişleri:

saat: 1 bitlik saat sinyali

reset: 1 bitlik reset sinyali (senkron)

basla: 1 bitlik devrede işlemi başlatan giriş sinyali

baslangic_degeri: 8 bitlik sayacın saymaya başlayacağı sayıyı belirten giriş sinyali

yon: 1 bitlik sayacın ileri mi geri mi sayacağını (mantık-1 ileri, mantık-0 geri) belirten giriş sinyali

miktar: 3 bitlik sayma miktarının -yon girişi yönünde sayarken- kaç olacağını belirten giriş sinyali

Devrenin çıkışları:

sonuc: 8 bitlik sayacın sonucunu (saydığı sayıyı) gösteren çıkış sinyali

hazir: 1 bitlik sayacın saymayı bitirip sonucun hazır olduğunu gösteren çıkış sinyali

mesgul: 1 bitlik sayacın meşgul olup olmadığını (sayma işleminin devam edip etmediğini) belirten çıkış sinyali

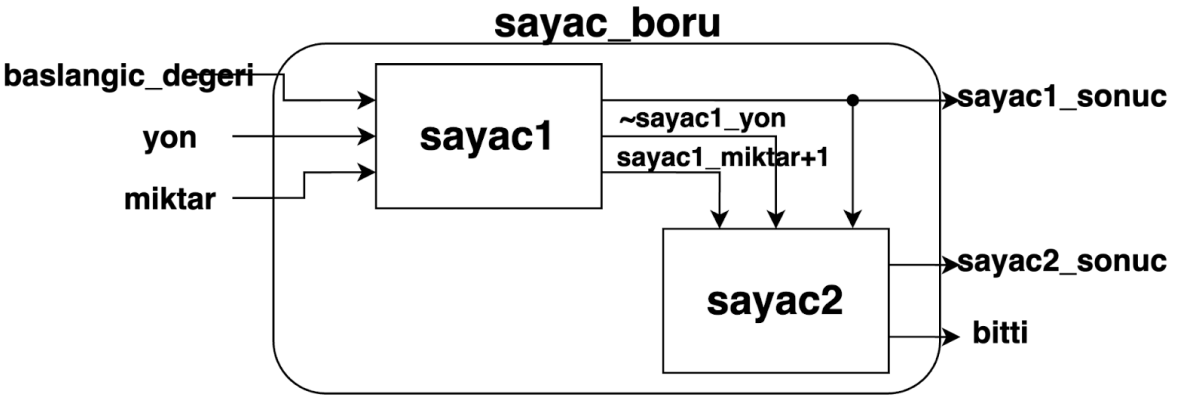
cikis_miktar: 3 bitlik sayma miktarını belirten çıkış sinyali

cikis_yon: 1 bitlik sayacın sayma yönünü belirten çıkış sinyali

Örnekler:

baslangic_degeri = 0 yon = 1 miktar = 3 basla = 1 → **sonuc = 0; hazir = 0; mesgul = 1;** basla = 0 → sonuc = 3; hazir = 0; mesgul = 1; basla = 0 → sonuc = 2; hazir = 0; mesgul = 1; basla = 0 → sonuc = 5; hazir = 0; mesgul = 1; basla = 0 → sonuc = 4; hazir = 0; mesgul = 1; basla = 0 → sonuc = 7; hazir = 0; mesgul = 1; basla = 0 → sonuc = 6; hazir = 0; mesgul = 1; basla = 0 → sonuc = 9; hazir = 0; mesgul = 1; basla = 0 → sonuc = 8; hazir = 0; mesgul = 1; basla = 0 → sonuc = 11; hazir = 0; mesgul = 1; … basla = 0 → sonuc = 253; hazir = 0; mesgul = 1; basla = 0 → sonuc = 252; hazir = 0; mesgul = 1; basla = 0 → **sonuc = 255; hazir = 1; mesgul = 1;** **→ bir sonraki çevrim → mesgul = 0;** … basla = 1 → yeni sayma işlemleri başlar.	baslangic_degeri = 250 yon = 0 miktar = 7 basla = 1 → **sonuc = 250; hazir = 0; mesgul = 1;** basla = 0 → sonuc = 243; hazir = 0; mesgul = 1; basla = 0 → sonuc = 244; hazir = 0; mesgul = 1; basla = 0 → sonuc = 237; hazir = 0; mesgul = 1; basla = 0 → sonuc = 238; hazir = 0; mesgul = 1; basla = 0 → sonuc = 231; hazir = 0; mesgul = 1; … basla = 0 → sonuc = 9; hazir = 0; mesgul = 1; basla = 0 → sonuc = 10; hazir = 0; mesgul = 1; basla = 0 → **sonuc = 3; hazir = 1; mesgul = 1;** **→ bir sonraki çevrim → mesgul = 0;** … basla = 1 → yeni sayma işlemleri başlar.
baslangic_degeri = 105 yon = 1 miktar = 4 basla = 1 → **sonuc = 105; hazir = 0; mesgul = 1;** basla = 0 → sonuc = 109; hazir = 0; mesgul = 1; basla = 0 → sonuc = 108; hazir = 0; mesgul = 1; basla = 0 → sonuc = 112; hazir = 0; mesgul = 1; basla = 0 → sonuc = 111; hazir = 0; mesgul = 1; basla = 0 → sonuc = 115; hazir = 0; mesgul = 1; … basla = 0 → sonuc = 250; hazir = 0; mesgul = 1; basla = 0 → sonuc = 249; hazir = 0; mesgul = 1; basla = 0 → **sonuc = 253; hazir = 1; mesgul = 1;** **→ bir sonraki çevrim → mesgul = 0;** … basla = 1 → yeni sayma işlemleri başlar.	baslangic_degeri = 50 yon = 1 miktar = 1 basla = 1 → **sonuc = 50; hazir = 0; mesgul = 1;** basla = 0 → sonuc = 51; hazir = 0; mesgul = 1; basla = 0 → sonuc = 52; hazir = 0; mesgul = 1; … basla = 0 → sonuc = 254; hazir = 0; mesgul = 1; basla = 0 → **sonuc = 255; hazir = 1; mesgul = 1;** **→ bir sonraki çevrim → mesgul = 0;** → bir sonraki çevrim baslangic_degeri = 50 yon = 0 miktar = 0 basla = 1 → **sonuc = 50; hazir = 1; mesgul = 1;** **→ bir sonraki çevrim → mesgul = 0;**

b-) [20 puan] "sayac_boru" modülü:
Bu modül, a şıkkında tasarlamış olduğunuz sayac modülünden 2 adet kullanarak art arda boru hatlı şekilde sayma işlemlerini gerçekleştirir.
Aşağıdaki şekilde boru hattı diyagramı verilmiştir.



Modülün isterleri aşağıdaki gibidir.

- Eş zamanlı atamalar saatin yükselen kenarında yapılacaktır.
- Saat ile senkron şekilde "reset" sinyali geldiğinde tüm modüller -alt modüller de dahil- başlangıç durumuna dönmelidir.
- 2 adet sayaç modülü birbirine boru hatlı şekilde bağlı çalışacaktır.
- Gelen girişler için 1. sayaç saymayı bitirdiğinde 2. sayaç saymaya başlayacaktır.
- 2. sayacın başlangıç değeri 1. sayacın "hazir" olduğu durumdaki "sonuc" değeridir.
- 2. sayaç modülünde "miktar" girişi 1 artacaktır. Örneğin "sayac1" için miktar 2 iken "sayac2" için miktar 3 olacaktır. 2. sayaç için bu miktar değerini 1. sayacın "hazir" olduğu çevrimdeki "cikis_miktar" değerine göre ayarlamalısınız. Eğer sayac1 için "miktar" değeri "7" ise daha fazla artamayacağı için sayac2 için de "7" olacaktır.
- 2. sayaç modülü, 1. sayaç modülünün tersi yönde çalışacaktır. Örneğin "sayac_boru" modülüne yon=1 geldiyse sayac1 modülü ileri, sayac2 modülü geri; tersi durumda ise sırasıyla geri, ileri yönlerinde çalışacaklardır. 2. sayaç için yön değerini, 1. sayacın "hazir" olduğu çevrimdeki "cikis_yon" değerine göre (tersini alıp) almalısınız.
- Boru hattı sayac1→sayac2 şeklinde çalışacaktır. Başlangıçta sayac1 modülü başlayacak ve son saydığı değer "hazir" olduğunda (saymayı bitirdiğinde), eğer sayac2 modülü de meşgul değilse "~sayac1_yon" ve "sayac1_miktar+1" değerleriyle sayac2 modülünü başlatacaktır. sayac2 modülü saymaya devam ederken sayac1 modülü yeni girişleri alıp kendi içinde farklı bir sayma işlemini başlatabilmelidir. Boru hattı bu şekilde devam edecek ve farklı modüller farklı girişlerin değerlerini aynı anda işleyebilecektir.
- Eğer sayac2 modülü meşgulse bir önceki modülden aldığı değerleri meşgul olmayana kadar boru hattı içinde tutmanız (değerleri kaybetmemek için) gerekmektedir. Meşgul olmadığı andan itibaren ise bu tuttuğunuz değerlerle sayac2 modülünü başlatmalısınız.
- Boru hattı her sayma işlemini bitirdiğinde "bitti" çıkışı mantık-1, diğer durumlarda mantık-0 olur. Yani, sayac2 modülü her işlemini bitirdiğinde -"hazir" olduğunda- sayac_boru modülünün "bitti" çıkışı 1 olmalıdır.
- sayac_boru modülünün "sayac1_sonuc" ve "sayac2_sonuc" çıkışları direkt olarak ilgili sayaçların "sonuc" çıkışlarına bağlı olmalıdır.

Devrenin girişleri:

saat: 1 bitlik saat sinyali

reset: 1 bitlik reset sinyali (senkron)

baslangic_degeri: 8 bitlik 1. sayacın saymaya başlayacağı sayıyı belirten giriş sinyali

yon: 1 bitlik 1. sayacın ileri mi geri mi sayacağını (mantık-1 ileri, mantık-0 geri) belirten giriş sinyali

miktar: 3 bitlik 1. sayacın sayma miktarının -yon girişi yönünde sayarken- kaç olacağını belirten giriş sinyali

Devrenin çıkışları:

sayac1_sonuc: 8 bitlik 1. sayacın sonucunu (saydığı sayıyı) gösteren çıkış sinyali

sayac2_sonuc: 8 bitlik 2. sayacın sonucunu (saydığı sayıyı) gösteren çıkış sinyali

bitti: 1 bitlik boru hattındaki sayma işleminin bittiğini belirten çıkış sinyali

Örnek:

baslangic_degeri = 0, yon = 1, miktar = 2

→ sayac1 0'dan başlayıp 0-2-1-3-2-4-3-5-...-255 örüntüsünde 2 ileri 1 geri şeklinde sayacak ve son sayma çevriminde 255 sonucunu "hazir" sinyali ile birlikte dışarı verecek. Bir sonraki çevrim (sayac2 çalışırken) gelen diğer girişlere göre farklı bir sayma işlemine başlayabilecek.

→ sayac2 "mesgul" olmadığı zaman sayac1'den gelen 255 başlangıç değeri ile başlayacak ve 3 geri 1 ileri şeklinde sayacak, yani 255-252-253-250-251-248-...-0 ve son sayma çevrimi 0 sonucunu "hazir" sinyali ile birlikte dışarı verecek, bu durumda sayac_boru modülünün bitti çıkışı mantık-1 olacak. Bu çevrimde sayac2_sonuc değeri bu örnekteki son sayılan sayı olan 0 olurken sayac1_sonuc değeri o an hangi sayıyı sayıyorsa o olacak.