



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

BİL 265/BİL 264/BİL 264L  
2021 – 2022 Öğretim Yılı  
Güz Dönemi Ödev 2  
25 /10 /2021

**Not:** Verilen giriş, çıkış ve modül isimlerinin farklı olması durumunda(büyük/küçük harf dahil) puan kırılacaktır.

**IEEE 754 Standardında Floating Point Çarpıcı [100]**

Bu soruda sizden istenilen, giriş olarak gelen ve IEEE 754 standardında olan iki adet 32 bitlik sayıyı çarpmanız ve yine aynı standarda uygun bir şekilde sonuç olarak vermeniz. Gösterim gereği girişler ve çıkış 32 bitlik olmalıdır. Çarpma işlemi sonucuna göre bazı durumlarda değer kaybı yaşanacaktır (32 bit ile göstermekten kaynaklı). Bu soru için “**carpici**” isimli aşağıda ayrıntıları açıklanan modülü verilmiş davranışsal modelleme ile tasarlamanız gerekmektedir. Yapacağınız işlemi kağıt üzerinde daha rahat tasarlayabilmek adına [bu videodan](#) faydalanabilirsiniz.

**Modülünüzün giriş çıkış bilgileri şu şekildedir:**

**clk:** 1 bitlik saat giriş sinyali

**reset:** 1 bitlik giriş sinyali

**sayi1:** 32 bitlik giriş sinyali

**sayi2:** 32 bitlik giriş sinyali

**sonuc:** 32 bitlik çıkış sinyali

Kodunuzu yazarken sıralı mantık kullanmalısınız ve kodlamadan önce tasarımınızı, yapacağınız işlemleri küçük parçalara bölerek bir durum makinesi oluşturacak şekilde tasarlamalısınız. Kodunuz en kötü ihtimalde 35 çevrimde sonuç üretmelidir. Modülünüz ‘reset’ girişi 1 olduğunda kullanacağınız registerların değerlerini senkron olarak sıfırlamalıdır.

**Örnek:**

sayi1: 01000001011100000000000000000000 → 15 (10’luk tabanda)

sayi2: 01000001001000000000000000000000 → 10 (10’luk tabanda)

sonuc: 01000110001011000000000000000000 → 150 (10’luk tabanda)

Modülünüz sayi1 ve sayi2 olarak IEEE 754 standardında gelen iki adet sayıyı çarparak gerektiği kadar çevrimden sonra sonucu çıkış olarak verir. Burada sizden istenilen temel şey kodu olabildiği kadar iş parçacıklarına işlemleri parçalayabilmenizdir.

Sayılar negatif veya pozitif olabilir. Sayıların exponent ve mantissa kısımlarını iyi ayarlamalı ve işlemler sonucunda değişimlerin nasıl olduğunu dikkat ederek kodunuzu yazmalısınız.

**NOT:** Sizden direk sentez veya zaman analizi yapmanızı istemiyoruz. Fakat kodunuzun başarımını anlayabilmemiz için, kodunuzun hatasız sentezlenebilmesine de dikkat ediniz. Ayrıca çarpma işlemini ne kadar iyi parçalara ayırırsanız kodunuzun çalışabileceği frekans da o kadar artacaktır. İş parçacıklarına ayırmadan yaptığınız çarpma işlemi ile parçalara ayırdığınız çarpma işleminin zamansal analizlerini karşılaştırabilirsiniz. Bu konuda

[https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2019\\_2/ug901-vivado-synthesis.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_2/ug901-vivado-synthesis.pdf) ve

[https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2019\\_1/ug906-vivado-design-analysis.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug906-vivado-design-analysis.pdf) kaynaklarından yararlanabilirsiniz.

Dönem sonunda sorun yaşamamanız adına tüm tasarımın özgün ve size ait olmasına dikkat ediniz. Gerekli görülen kişiler için ödevdeki tasarımınızı sunmanız istenecektir.