

FBU-CPU System Verilog Testbench

Recep Gemalmaz, Berk Tunç, Ogün Berat Gürses, Evrim Arda Kalafat, Cüneyt Balcı

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: [evrim.kalafat@stu.fbu.edu.tr](mailto:evrim.kalafat@stu.fbu.edu.tr), [cuneyt.balci@stu.fbu.edu.tr](mailto:cuneyt.balci@stu.fbu.edu.tr), berk.tunc@stu.fbu.edu.tr, ogun.gurses@stu.fbu.edu.tr, recep.gemalmaz@stu.fbu.edu.tr,

**Proje özeti:** Bu proje kapsamında dijital tasarım dersinde tamamlanan FB-CPU işlemcisinin SystemVerilog dili ile otonom kontrolünü yapan bir doğrulama ortamı geliştirilecektir.

**Anahtar Kelimeler:** FPGA, CPU,SystemVerilog.

**Abstract:** Within the scope of this project, the SistemVerilog of the FB-CPU system, which was completed in digital education, will be developed in a way that will make it an autonomous controller.

**Keywords:** FPGA, CPU, SystemVerilog.

1. **Giriş**

tb\_fbcpu.sv dosyasında iki adet sınıf tanımı yapılmıştır. Bunlar; dosyaSinifi, testSinifi

Bu sınıflardan objeler türetilerek initial begin bloğu içerisinde sınıfın çeşitli fonksiyon ve değişkenleri kullanılarak FBCPU test edilmektedir.

Proje kapsamında dosyaSinifi ve testSinifi isimli sınıfların boş bırakılan sınıf içerikleri doldurulacaktır. Aşağıda dosyaSinifi ve testSinifi isimli sınıfların içerdiği değişken ve fonksiyonların bulunduğu tablo verilmektedir. testSinifi isimli sınıf dosyaSinifi sınıfından kalıtım yapılmıştır.

Proje kapsamında geliştirilen testbench hatalı bir FBCPU kodu ile denenerek testbench görevinin doğru yerine getirip getirmediği kontrol edilecektir.

1. **Gerekli Fonksiyon Bilgilendirmeleri**

Aşağıda dosyaSinifi isimli sınıfın fonksiyonları açıklanmıştır.

• new():Constructor’dır. Tüm değişkenleri 0’a atamaktadır.

• dosyayiInitializeEt(string girisDosyaAdi):Kendisine verilen string argümandaki dosya adı ile dosyayı açmaya çalışır. Dosya açıldığında geriye dönen file descriptor’u fileDescriptor değişkenine atar. Dosya başarılı olarak açılırsa fileInitialized değişkeni 1 olur, diğer durumda ise 0 olur. Fonksiyon geriye başarılı iken 1, değil iken 0 döndürür.

• dosyadanOku(): fileInitialize değişkeni 1 ise, dosyadan 1 satır okuyup bunları memAddr ve data değişkenlerine yazar. Dosyanın sonuna erişildiğinde ise fileInitialize değişkenini 0 yapar. Dosyadan her başarılı okunan satır için lineCount değişkenini bir arttırır. Başarılı okunmalarda fonksiyon geriye 1, diğer durumda ise 0 döndürür.

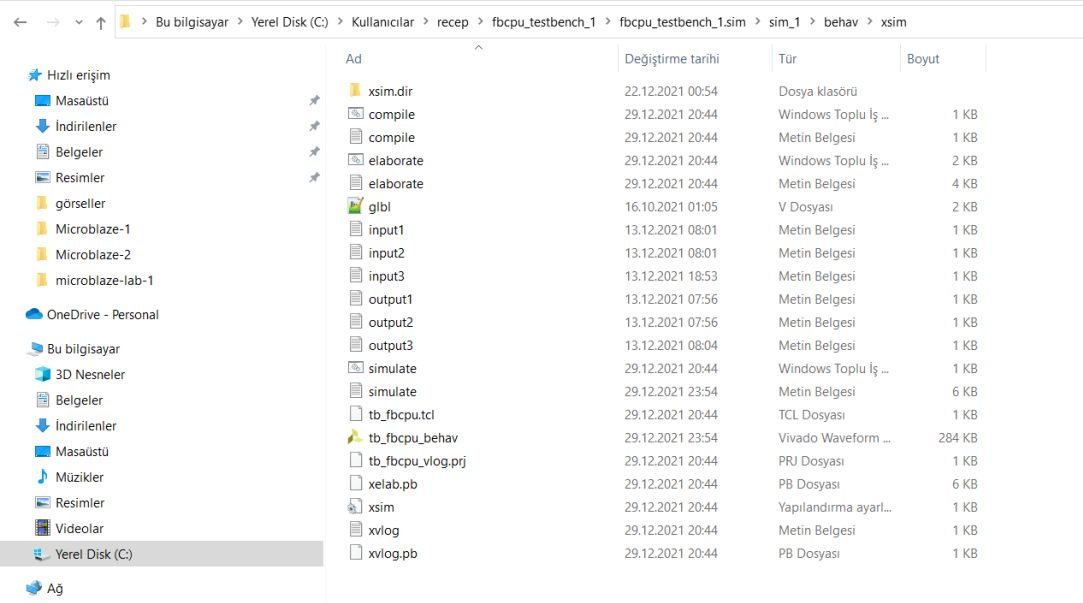
• new():Constructor’dır. testNo değişkenini 0’a atar ve girisDosyasi, cikisDosyasi değişkenlerini new ile initialize eder.

• testNoAyarla(int girisTestNo):testNo değişkenine girisTestNo argümanını yazar.

• testInitializeEt(): Sınıfın içinde bulunan testNo değişkenin değerine göre girisDosyasi.dosyayiInitializeEt fonksiyonu ile input1, input2, input3 txt dosyalarından birini açar (örn girisDosyasi.dosyayiInitializeEt("input1.txt")) ve çıkış dosyalarından output1, output2, output3 dosyalarından birini açar. Dosyaların açılmasında sorun olursa $finish; komutu ile simülasyonu durdurur.

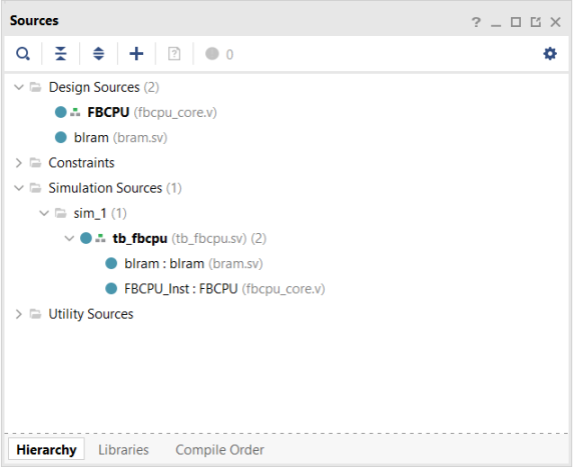
• kontrolEt(reg [7:0] memLocation, reg [21:0] expectedValue ):Kendisine argüman olarak verilen memLocation bilgisini kullanarak, BRAM’deki adres’e bakar. O adresteki içeriğin değeri ile expectedValue değerini karşılaştırır. Aynı ise simülasyon başarılı olarak çıktı verir, değil ise simülasyon hatalı olarak çıktı verir.

**Input ve Outputların Konulduğu Yer**

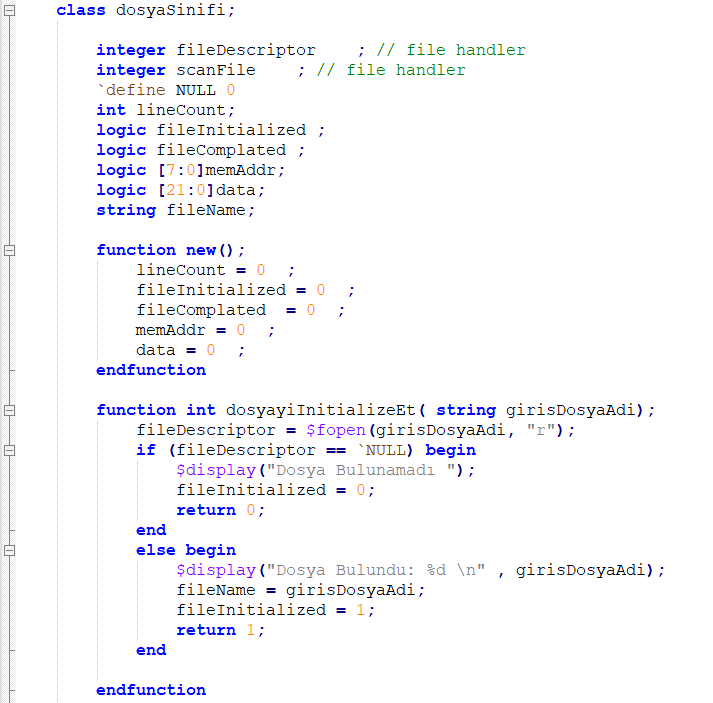


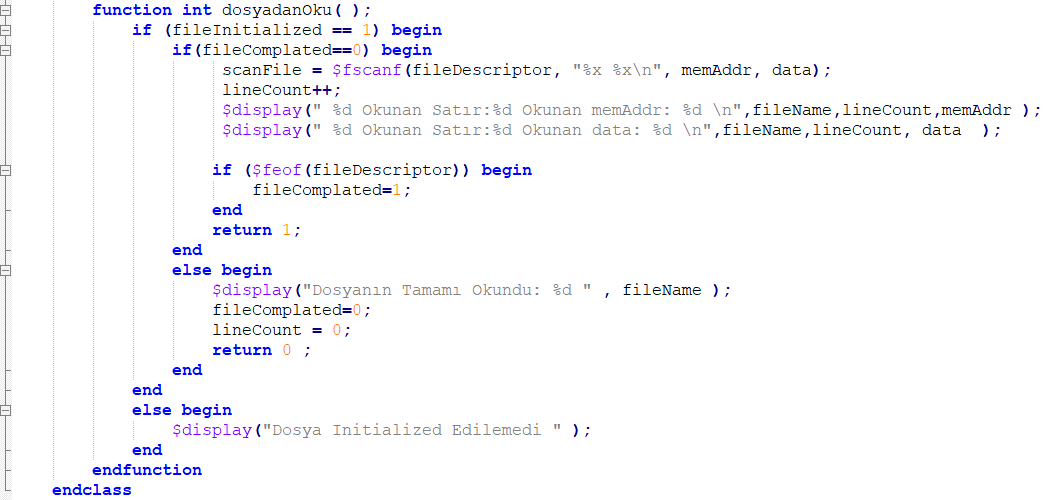
**Sources**

* Resimde de görüldüğü üzere Sources kısmında FBCPU’nun ve blram’in design source olarak, tb\_fbcpu nun da simülasyon olarak atandığı görünmektedir.



**Dosya Sınıfı**



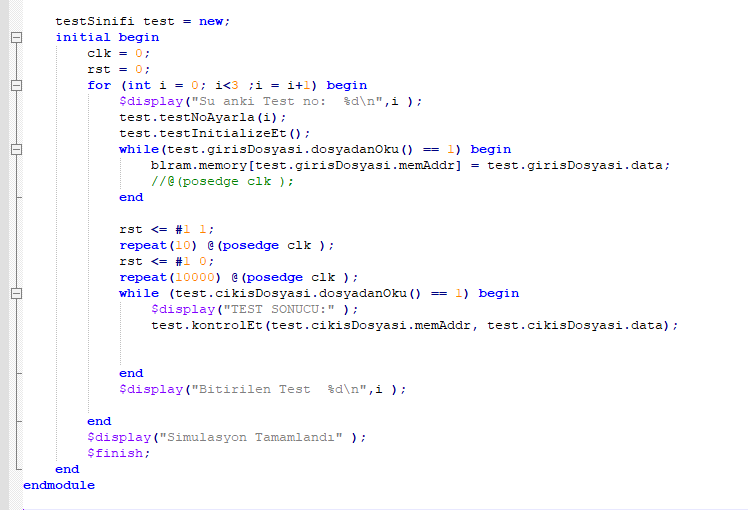


**Test Sınıfı**

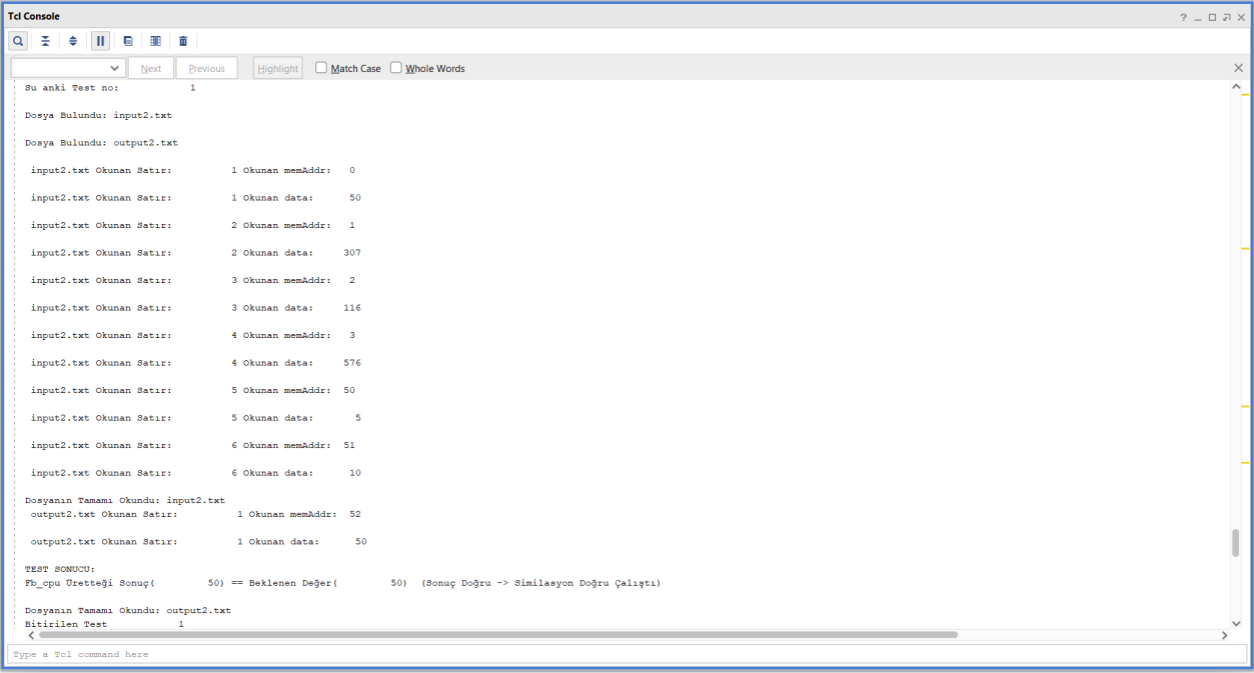


**Sonuçlar**

**Test Kısmı**



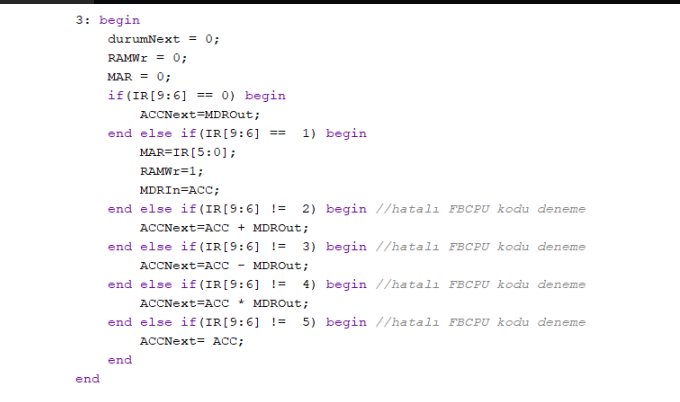
**Elde Edilen Çıktılar**



Elde edilen sonuçlara göre doğru çalışan bir CPU’da alınan sonuçların da doğru olduğu gözükmektedir.

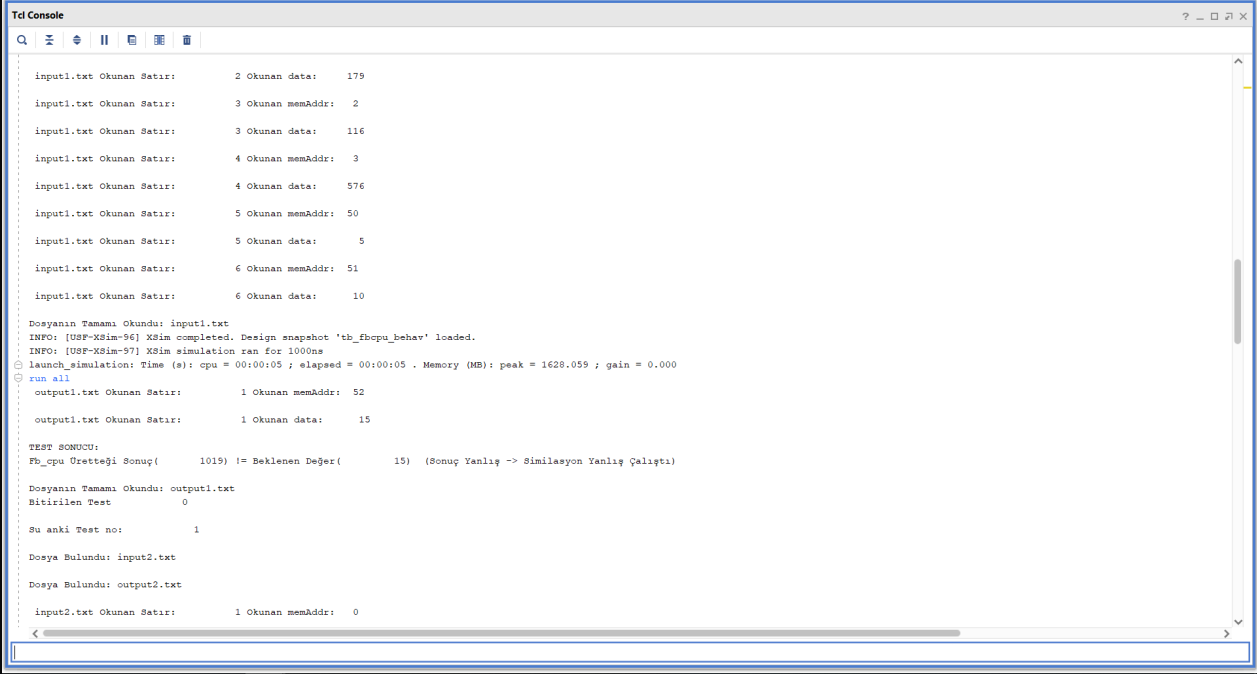
**Test Kısmı 2**

* Eğer aşağıdaki kod parçasında görüldüğü gibi olması gerekenin aksine hatalı çalışan bir CPU koyarsak testbench hatalı olduğunu teyit edecektir.



Bu kısımda olması gereken eşit eşittir (==). eşit değil (!=) olarak değiştirildiğinde kod yanlış çalışacak simülasyon hatayı bulacaktır.

**Elde Edilen Çıktılar**



1. **Kullanılan Yazılım**

Tasarımlarımızı yapmak için Xilinx tarafından geliştirilen Vivado Design Suite yazılımını kullandık. Vivado Design Suite, HDL tasarımlarının sentezi ve analizi için üretilmiş bir yazılım paketidir ve Xilinx ISE'nin yerine çip geliştirme ve üst düzey sentez sistemi için ek özellikler sunar. Biz de SystemVerilog dilini kullanarak Vivado üzerinde tasarımımızı yaptık. IEEE 1800 olarak standartlaştırılmış SystemVerilog ise elektronik sistemleri modellemek, tasarlamak, simüle etmek, test etmek ve uygulamak için kullanılan bir donanım açıklaması ve donanım doğrulama dilidir.

1. **Proje Ekibi**

**Evrim Arda KALAFAT**, 25.09.2001 yılında İstanbul’da doğdu. 2019 yılında Kadıköy Final Temel Lisesi’nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. Bilgisayar Mühendisliği Bölüm Temsilciği yapmaktadır. C, C++ ve Python dillerinde bilgili. Programlama, yapay zeka ve siber güvenlik ile ilgileniyor.

**Ogün Berat GÜRSES**: 10.11.2000 Sakarya ili Adapazarı ilçesinde doğdu. 2018 yılında Sakarya Anadolu Lisesi’nden mezun oldu. Şu anda Fenerbahçe Üniversitesi’nde Bilgisayar Mühendisliği bölümünde eğitim almaktadır. C, C++, mySQL, Verilog, Sys-Verilog ve Python ile ilgilenmektedir.

**Berk TUNÇ:** 04.02.2000 yılında Yalova ili Merkez ilçesinde doğdu. 2018 yılında Şehit Osman Altınkuyu Anadolu Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi’nde Bilgisayar Mühendisliği bölümünde bölümünde eğitim almaktadır. mySQL, HTML, C, C++, Verilog, Sys-Verilog ve Python ile ilgilenmektedir.

**Cüneyt BALCI:** 28.08.2000 yılında İstanbul'da doğdu. 2018 yılında Final Temel Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi’nde Bilgisayar Mühendisliği bölümünde lisans eğitimi almaktadır. C, C++, Python yazılım dilleri ve Blockchain alanıyla ilgilenmektedir.

**RECEP GEMALMAZ:** 16.10.2000 tarihinde Kadıköy’de dünyaya geldi. 2018 yılında Alparslan Anadolu Lisesi’nden Mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. Java, C, C++ ve Pyhton dillerinde bilgili. Android Programlama, Verilog ve Sys-Verilog ile ilgilenmektedir.

1. **Referans Dosyalar**

<https://www.youtube.com/watch?v=llLQw3KqpOg>

[FBU-CPU-System-Verilog-Testbench (github.com)](https://github.com/brktnc/FBU-CPU-System-Verilog-Testbench)

1. **Kaynaklar**

[1] http://www.levent.tc/courses/electronic-circuits