FB-CPU System Verilog Testbench

Ömer AYDENİZ , Hayat Zehra DEMİR ,Mehmet Rauf FUZÜN , Mert Meriç KARADENİZ ,Ahmet Batuhan YILMAZ, Bedirhan İLERİ

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: {omer.aydeniz,mert.karadeniz,hayat.demir,ahmet.yilmaz,mehmet.fuzun,bedirhan.ileri,[}@fbu.edu.tr](mailto:ahmet.mehmet}@fbu.edu.tr),

***Özetçe*—** Bu proje kapsamında dijital tasarım **dersini takip eden** FBCPU işlemcisinin SystemVerilog dili ile **otomatik** kontrolünü **gerçekleştiren** bir doğrulama ortamı geliştirilecektir.

***Anahtar Kelimeler — FPGA, CPU***

***Abstract*—**Within the scope of this project, a verification environment that performs automatic control of the FBCPU processor with the SystemVerilog language following the digital design course will be developed.

***Keywords — FPGA, CPU.***

# I. Giriş

# Geçen yıl dijital tasarım kursunda ne yaptık FB-CPU, geliştirdiğimiz tasarımı tam olarak doğrulamadı. Öte yandan bu proje, System Verilog dilini kullanarak otonom kontrol ile profesyonel doğrulama gerçekleştirme projesidir. Geçen sene yazdığımız işlemci üzerine Testbench yazılarak deneyler yapılacak.

# II. Sistem Mimarisi

FB-CPU işlemcisinin Testbench’i tasarlanırken Vivado ve simülasyon ortamı kullanılmıştır.

SystemVerilog dili kullanılarak yaptığımız Testbench’te ise her şeyin otonom olması sağlanmıştır.

Dosyadan otomatik olarak test dosyalarını okuyup, daha önceden yazmış olduğumuz CPU’ya besleyip sonuçlarını otomatik olarak karşılaştırıp, hata var mı yok mu bilgisini otonom olarak söyleyen bir yapı yapılmıştır. Bu şekilde geçen yılki kodumuzda hata olup olmadığı, yazdığımız yeni testbench ile ortaya çıkabilecektir. Testbench tasarımında Verilog yerine SystemVerilog dilinin tercih edilmesindeki en önemli faktör, SystemVerilog’un objeye yönelimli bir dil olmasıdır. Objeye yönelimli bir dil olması daha hiyerarşik, daha kullanılabilir, tekrar yazılması mümkün bir dil yapar. Bu sayede esnek doğrulama projelerinde kullanılabilir.

Bu bilgiler ışığı altında “sınıflar” kullanılabilir hale gelmektedir. Sınıflarda veri elemanları ve fonksiyonlar bulunmaktadır. Sınıflardan istendiği zaman dinamik olarak objeler türetilip silinebilir.

r.

# III. Kullanılan Yazılım

- fileDescriptor: Bir dosyayı açarken kullanılır. SystemVerilog’da bir şeyler yazar veya okurken, “oku” dediğimizde geriye bir fileDescriptor döner. Onu tutabilmek için bu sınıfın içerisinde int türünde bir fileDescriptor’e ihtiyaç vardır.

-scanFile: Her bir satırı okudukça doğru okuyup okumadığını gösterir.

-lineCount: Okudunan dosyanın kaç satırdan oluştuğunu sayıp, simülasyonun sonucunda kaç satırı test ettiğini/ kaç satırlık bir okuma yapıldığını ekrana bastırmak için kullanılan değişkendir.

- fileInitialized: Hata çıkmasını engellemek için bu sınıfın içerisinde önceden dosya açmamışken bir şeyleri okumaya çalışırsak, daha dosyayı açıp initalize etmediysek patlamamak için yani korumak için 1 bitlik bir fileInitialized isimli bir logic değişkeni tuttuk. Açılabilecek bir dosya olduğunu gördükten sonra değeri 1’e çekilir; değilse 0’da bırakılır. Bu bize henüz dosyayı okuyamadıysa/açamadıysa, dosyadan okumaya çalışmamasını sağlar.

-fileComplated: Dosyanın sonuna erişip erişmediğini anlamak için kullanılan bir değişkendir. Eğer sonuncu satıra geldiysek artık yeni bir şey okumamayı sağlayan bir saklayıcıdır.

-data ve memAddr: Test giriş dosyalarından okudukça değişkenlerin içerisine doldurmayı sağlar. İlk veri adres, ikinci veri ise data’yı temsil etmektir.

# IV. Sonuçlar

En son ise tcl konsolda simülasyonun tamamlandığının yazısını gördük.

Tüm test dosyaları başaralı bir şekilde okunmasının haricinde waveform ekranını incelediğimizde verilerimizin okunduğunu gözlemlemiş olduk.

Bu proje sonrasında geçen yıl dijital sistem tasarımında yazmış olduğumuz FB-CPU kodunun doğruluğunu otonom şekilde doğrulamış olduk. SystemVerilog diline olan hakimiyetimizin artması haricinde bu dildeki objeye yönelimli programlama yöntemlerini, sınıf kullanımını öğrenmiş ve deneyimlemiş olduk.

##### Proje Ekibi

Bedirhan İleri

Okul Numarası:210301501

Doğum Tarihi:06.12.2001

Mezun Olduğu Lise:Prof. Dr. Nabi Avcı Anadolu Lisesi

Ömer Aydeniz

Okul Numarası:200301502

Doğum Tarihi 03.07.2001

Mezun Olduğu Lise:Açık Öğretim Lisesi

Mert Meriç Karadeniz

Okul Numarası:190301013

Doğum Tarihi : 17.09.2001

Mezun Olduğu Lise:Mustafa Kemal Anadolu Lisesi

Mehmet Rauf Fuzün

Okul Numarası:190301023

Doğum Tarihi 12.11.2001

Mezun Olduğu Lise:Akhisar Anadolu İmam Hatip Lisesi

Hayat Zehra Demir

Okul Numarası:190301014

Doğum Tarihi 20.09.2001

Mezun Olduğu Lise:Antalya İstek Koleji

##### Referans Dosyalar

<https://www.youtube.com/watch?v=vxvIsN6nd7M>

<https://github.com/DeveloperBedirhan/FB-CPU-SystemVerilog-Testbench>

##### Kaynaklar

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

[2] <https://www.chipverify.com/systemverilog/systemverilog-functions>

[3] <http://www.levent.tc/courses/electronic-circuits>