**FENERBAHÇE ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ 2. SINIF**

**COMP 201**

**FBU-CPU RTL TASARIMI**



Barış Subaşi

Ece Aydınkaptan

Tenay Dilara Özdemir

Tuğba Karadeniz

Aybüke Şen

**e-mail:**

baris.subasi@stu.fbu.edu.tr

ece.aydinkaptan@stu.fbu.edu.tr

tenay.ozdemir@stu.fbu.edu.tr

tugba.karadeniz@stu.fbu.edu.tr

aybuke.sen@stu.fbu.edu.tr

**Özetçe:**

FB-CPU, işlemcilerin temel çalışma prensiplerini anlatmak için, eğitim amaçlı bir işlemcidir. Bu proje kapsamında FB-CPU isminde bir işlemcinin Verilog dili ile RTL tasarımı ve tasarlanan işlemci üzerinde makine dili ile yazılan çeşitli kod parçacıkları yazılmıştır. Proje sonunda basit bir işlemcideki RAM, Kontrol Ünitesi ve Saklayıcıların bir arada çalışıp, makine dilindeki kod parçacıklarının nasıl yürütebildiği gözlenmiştir. Kullanılacak Basys3 FPGA geliştirme kartı üzerinde FB-CPU demosu yapılmıştır.

**Anahtar Kelimeler:**

FPGA, CPU, RAM, İşlemci, RTL

**Abstract:**

FB-CPU is an educational processor to explain the basic working principles of processors. Within the scope of this project, RTL design of a processor named FB-CPU with Verilog language and various code snippets written in machine language on the designed processor were written. At the end of the project, it was observed how RAM, Control Unit and Stores in a simple processor can work together and execute code snippets in machine language. FB-CPU demo was made on Basys3 FPGA development board to be used.

**Keywords:**

FPGA, CPU, RAM, Processor, RTL

**I. GİRİŞ**

FB-CPU isminde işlemcinin Verilog dili ile RTL tasarımı ve tasarım üzerinde makine dili ile çeşitli kod parçacıkları yazımı yapılmıştır. Sonrasında bu elemanlar tümü beraber çalıştırılıp simüle edilip gözlemlenmiştir.

**II. SİSTEM MİMARİSİ**

2 Farklı araç kullanılmıştır:

1)Von Neumann Simulatörü: Tasarımı gerçekleştiren ve veri akışını daha rahat bir şekilde gözlemleyebildiğimiz bir simülatördür.

2)Xilinx Vivado Design Suite: Xilinx tarafından donanım tanımlama dili tasarımlarının sentezi ve analizi için üretilmiş bir yazılım paketidir. Tasarımımızı oluşturmak ve FPGA kartlarında çalışma yapmak için kullanılmıştır.

**III. TASARIM MİMARİSİ**

Durum saklayıcısı 0’a eşit ise; MAR(Memory Adress Register) saklayıcısındaki veri PC(Program Counter)’a besleniyor. Sonra Ram Write(RamWR) sinyali 0 besleniyor ve durum saklayıcısı 1 artırılıyor.

Durum saklayıcısı 1’e eşit ise: IR(Instruction Register) saklayıcısındaki veri MDR(Memory Data Register)’a besleniyor. PC(Program Counter) saklayıcısındaki değer 1 artırılıyor ve durum saklayıcısındaki değer de 1 artırılıyor.

Eğer operasyon kodu 6 dan küçükse yani datanın 0 dan 5 e kadarki değerleri ,memory adress register ın içine yazılır ve durum 3 e geçilir.

Eğer operasyon kodu 6 ya eşitse jump işlemi yapılır. Data , program counter a yazılır ve durum 0 a getirilir.

Eğer operasyon kodu 7 ise ve accumulator değeri 0 sa data yine program counter içine yazılır ve durum 0 a gidilir.

Eğer operasyon kodu 8 e eşitse hiç işlem yapılmaz ve direk durum 0 a geçilir.

Eğer operasyon kodu 9 ise halt olur ve durum4 e gidilir ve durum4 te sürekli kendi içinde döndürülür.

Durum3 te operasyon koduna göre işlemler yapılır.

Eğer operasyon kodu 0 ise memory data out değeri accumulator a yazılır.

Eğer operasyon kodu 1 ise store işlemi yapılır , data memory adress registerın içine yazılır.

RAMWR(ramviar) değeri 1 olur ve accumulator içindeki değer memory data register in‘in içine yazılır.

Operasyon kodu 2 ise accumulator ile data toplanır.

Eğer operasyon kodu 3 ise accumulator dan memory data register out çıkartılır.

Eğer operasyon kodu 4 ise accumulator ile memory data register out çarpılır

Bölme işlemi yapılmış ama şemada durum 3 içerisinde bulunan 5 e eşit olma durumu verilog dilinde yapılmamıştır.

**IV. KULLANILAN YAZILIM**

<http://www.levent.tc/files/courses/digital_design/project/fbcpu_baslangic.rar>

Verilmiş olan fbcpu\_baslangic.rar dosyasında bulunan:

\* fbcpu\_core.v

\* memory.v

\* tb\_fbcpu.v

\* testcase1.v

\* testcase2.v

\* testcase3.v

dosyaları referans alınmıştır.

**V. SONUÇLAR**

Bu proje ile bir işlemcinin temel bileşenleri, çalışma prensipleri ve çalışma mantığını anlayarak elde edilen RTL tasarımındaki iş, akış süreçlerindeki gereksinimleri edinmiş olduk. Geliştirilen işlemci; yükleme, kaydetme, toplama, çıkarma, çarpma ve bölme işlemlerini desteklemektedir.

**PROJE EKİBİ**

**Barış Subaşi:** 06.10.2001 Kars doğumlu. İstanbul Köy Hizmetleri Anadolu Lisesi mezunu. Fenerbahçe Üniversitesi bilgisayar mühendisliği 2. sınıf öğrencisi.

**Ece Aydınkaptan:** 02.10.2000 Kocaeli doğumlu. Kadıköy Fen Bilimleri Anadolu Lisesi mezunu. Fenerbahçe Üniversitesi bilgisayar mühendisliği 2. sınıf öğrencisi.

**Tuğba Karadeniz:** 03.07.2000 Bursa doğumlu. Bursa Anadolu Lisesi mezunu. Fenerbahçe Üniversitesi bilgisayar mühendisliği 2. sınıf öğrencisi.

**Tenay Dilara Özdemir:**11.09.2001Ankara İstek Acıbadem Fen Lisesi mezunu. Fenerbahçe Üniversitesi bilgisayar mühendisliği 2. sınıf öğrencisi

**Aybüke Şen:** 08.05.2002 İstanbulAtaşehir Okyanus Koleji Fen Lisesi mezunu. Fenerbahçe Üniversitesi bilgisayar mühendisliği 2. sınıf öğrencisi

**REFERANS DOSYALAR**

**Github:** <https://github.com/barisubasi/FB-CPU-RTL-DESIGN>

**YouTube:** <https://youtu.be/WTsN4qjRFPs>

**KAYNAKLAR**

<http://www.levent.tc/>