BLM 101- Bilgisayar Mühendisliğine Giriş
2019-2020 Güz Dönemi

FB-CPU V1

Proje Teslim Raporu
13 Ocak 2020

Alp Yılmaz , Hüseyin Berk Işıldak Erdem Şentürk , Serhat Erdoğan

1 Giriş

1.1 Projenin Amacı

FB-CPU adında işlemci tasarlamak ve tasarlanan işlemci üzerinde çeşitli kod parçacıkları yazmak. Proje sonunda bir işlemcideki RAM, Kontrol Ünitesi ve Saklayıcıların çalıştırıp, makine dilindeki kodları nasıl çalıştırdığını gözlemlemek.

1.2 Proje Ekibi

Serhat Erdoğan 190301026

03.06.2000 tarihinde Doğdu 2018 yılında Yalova Uğur Okulları'ndan mezun oldu. Fenerbahçe Üniversitesi'nde Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta.İngilizce biliyor. Orta seviye C dili ve giriş seviyesi HTML ve CSS biliyor.
.

Alp Yılmaz 190301017

18.10.2000 tarihinde doğdu.2018 yılında Celal Aras Anadolu Lisesi'nden mezun oldu.Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta.Giriş seviyesinde almanca biliyor. Orta seviye C dili biliyor.

Hüseyin Berk Işıldak 190301006

25.04.2000 tarihinde doğdu.2018 yılında Yusuf Kemalettin Perin Fen Lisesi'nden mezun oldu. Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. Orta seviye C dili biliyor.

Erdem Şentürk 190301009

05.09.2000 tarihinde doğdu. 2018 yılında İstek Bilge Kağan Lisesi'nden mezun oldu. Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta.İngilizce biliyor.Orta seviye C dili biliyor.

2 Sistem Mimarisi

2.1 Kullanılan Araçlar

Von Neumann Simulatörü , Logisim-Evolution

2.2 Tasarım

Durum saklayıcısının gösterdiği değerlere göre komutlar çalıştırılıyor.

FB-CPU Von Neumann Mimarisinde tasarlanmış,komutları çalıştırmaktadır.8 adet saklayıcı bulundurmaktadır.

SAKLAYICILARIN GÖREVLERİ

• PC (6 Bit): RAM üzerinde hangi satırdaki komutun alınacağını belirler. 6 bit olmasının nedeni RAM'in 2^6 lokasyonu olmasındandır. Dolayısıyla PC değeri RAM'deki her yeri gösterebilmektedir.

• MAR (6 Bit): Memory Address Register isminde bir saklayıcıdır. Bu saklayıcı RAM'in adres girişine bağlanmıştır. RAM'in 2^6 lokasyonu olduğu için MAR 6 bitliktir. Saklayıcı RAM'in içerisindedir.

• MDRIn (10 Bit): Memory Data Register In, RAM'e bir veri yazılacağı zaman kullanılan saklayıcıdır. RAM'in bir lokasyonu 10 bitlik olmasından ötürü, saklayıcı 10 bittir. Saklayıcı RAM'in içerisindedir.

• RAMWr (1 Bit): RAM'e veri yazılacağı durumlarda aktif edilmektedir. 1 olmadığı durumlarda RAM'e veri yazılmaz. Saklayıcı RAM'in içerisindedir.

• MDROut (10 Bit): Memory Data Register, RAM'den veri okunacağı zaman kullanılan saklayıcıdır. RAM'in bir lokasyonu 10 bit olmasından dolayı, saklayıcı 10 bittir. Saklayıcı RAM'in içerisindedir.

• IR (10 Bit): Instruction Register, RAM'den okunan kodun (instruction) saklandığı saklayıcıdır.

• ACC (10 Bit): Accumulator, aritmetik işlem sonuçlarının tutulduğu saklayıcıdır.

3 Geliştirilen Yazılım

```
0: 0000_110011 // LOD 51, ACC = *51, Hex = 33
1: 0011_110001 // SUB 49, ACC = ACC - *49, Hex = F1
2: 0111_001010 // JMZ 10, döngü bittiyse, döngüden çıkartacaktır (ACC-49 == 0), 10. Satır, Hex = 1CA
3: 0000_110000 // LOD 48, temp değerini yükle, başlangıçta 0, Hex = 30
4: 0010_110010 // ADD 50, ikinci sayıyı ACC'nin üstüne ekle, Hex = B2
5: 0001_110000 // STO 48, ACC'nin değerini temp'e ata, Hex = 70
6: 0000_110001 // LOD 49, ACC = i, Hex = 31
7: 0010_101110 // ADD 46, ACC = i + 1, Hex = AE
8: 0001_110001 // STO 49, i = i + 1, Hex = 71
9: 0110_000000 // JMP 0, döngünün başına dön 0. satır, Hex = 180
10: 0000_110000 // LOD 48, ACC = temp, Hex = 30
11: 0001_110100 // STO 52, *52 = ACC, Hex = 74
10: 1001_000000// HLT, bitirme, Hex = 240

46: 1 // 1 sayısı
48: 0 // Hex = 0, temp
49: 0 // Hex = 0, i index'i için
50: 0000000101 // Hex = 5
51: 0000001010 // Hex = A
```

0: 51. adresi acc'ye yüklenir.
1: acc'dan 49. Adres çıkarılır.
2: sonuç 0 ise 10. adrese atlanır.
3: 48. adres acc'ye yüklenir.
4: acc'ye 50. adresteki değer eklenir.
5: acc 48. Adrese yüklenir.
6: 49. Adresin değeri acc'ye yüklenir.
7: elimizdeki acc'ye 46. adresteki değer ekleniyor.
8: acc 49. Adrese yüklenir.
9: 0. Adrese atlanıyor.
10: acc'ye 48. Adres yüklenir.
11: acc'yi 52. Adrese yüklüyoruz.
12. HLT programı bitiriyor.

46: 1
48: 0
49: 0 5
0: Hex = 5
51: Hex = A

Bu proje vasıtasıyla ekipçe çalışmayı, bir işlemcinin temel bileşenleri , proje ile birlikte bu konu hakkında bilgi ve deneyim edindik ve çalışma mantığını öğrenmiş bulunduk.

Hazırlanan sunum video’su adresi:

https://youtu.be/yDo6V1HGTjM

Dosyaların github adresi:

https://github.com/erdem106/BLM-101-FBCPU-Erdem-Alp-Serhat-Berk