



***BLM 101- Bilgisayar Mühendisliğine Giriş
2019-2020 Güz Dönemi***

FB-CPU V1

***Proje Teslim Raporu
13 Ocak 2020***

Berk TUNÇ, Arda ALHAN, Ogün GÜRSES, Arda KALAFAT

1	GİRİŞ	1
1.1	Projenin Amacı.....	1
1.2	Proje Ekibi.....	1
2	SİSTEM MİMARİSİ	2
2.1	Kullanılan Araçlar	2
2.2	Tasarım	2
3	GELİŞTİRİLEN YAZILIM	3
4	SONUÇLAR.....	4

1 Giriş

1.1 Projenin Amacı

Bu proje kapsamında çıkarma, çarpma, bölme, toplama işlemlerini yapan FB-CPU isiminde bir işlemcinin tasarımı ve tasarlanan işlemci üzerinde makine dili ile yazılan çeşitli kod parçacıkları çalıştırılacaktır.

1.2 Proje Ekibi

Adı ve Soyadı Berk Tunç (190301003). 04.02.2000 yılında doğdu. 2018 yılında Şehit Osman Altinkuyu Anadolu Lisesinden mezun oldu. Şuan da Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde Lisans eğitimi almakta. C, C#, HTML dilleriyle ilgileniyor.

Arda Alhan 18.05.2001 (190301020) yılında doğdu. 2019 yılında Eyüp Anadolu Lisesi'nden mezun oldu. Şuan da Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde Lisans eğitimi almakta. C, C#, Python dilleriyle ve Unity oyun motoruyla ilgileniyor.

Ogün Berat Gürses (190301005) 10.11.2000 yılında Sakarya ili Adapazarı ilçesinde doğdu. 2018 yılında Sakarya Anadolu Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. C ve benzeri dillerle ilgilenmektedir.

Evrin Arda KALAFAT 25.09.2001 yılında istanbulda doğdu. 2019 yılında Final Kadıköy Temel Lisesinden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. C diline hakim, Başlangıç seviyesinde Python biliyor.

2 Sistem Mimarisi

2.1 Kullanılan Araçlar

- 1) Von Neumann Simulatörü
- 2) Logisim-Evolution

2.2 Tasarım

İşlemcinin nasıl çalıştığı, tasarım sürecinde neler ile karşılaşıldığı, tasarımın nasıl yapıldığı ve varsa tasarımdaki eksiklikler anlatılacaktır. İşlemci tasarımı için başlangıç tasarımı önceden verildi. Saklayıcılar, bellek, işlem ünitesi, kontrol ünitesi olarak 4 temel elemana sahiptir. FB-CPU Von Neumann Mimarisinde tasarlanmış, 10 adet komutu çalıştırmaktadır. Tasarımda 8 adet saklayıcı bulunmaktadır.

Durum: Durumun sinyaline göre MUX'un girişi belirlendiği için saklayıcının D girişi sonraki durum olmaktadır.

PC: 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: Bu saklayıcı RAM'in adres girişine bağlanmıştır. RAM'in 2^6 lokasyonu olduğu için MAR 6 bitlidir. Saklayıcı RAM'in içerisindedir

MDRIn: RAM'e bir veri yazılacağı zaman kullanılan saklayıcıdır. RAM'in bir lokasyonu 10 bitlik olmasından ötürü, saklayıcı 10 bittir. Saklayıcı RAM'in içerisindedir.

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

MDROut: RAM'den veri okunacağı zaman kullanılan saklayıcıdır. RAM'in bir lokasyonu 10 bit olmasından dolayı, saklayıcı 10 bittir. Saklayıcı RAM'in içerisindedir.

IR: RAM'den okunan kodun (instruction) saklandığı saklayıcıdır.

ACC: Aritmetik işlem sonuçlarının tutulduğu saklayıcıdır.

FB-CPU'nun komutları okuyup, hesaplanan değerleri geri yazacağı belleğe bağlı 4 saklayıcı ve bir clock sinyali bulunmaktadır. FB-CPU'da 4 adet aritmetik işlem vardır. Bunlar toplama, çıkartma, çarpma ve bölmedir, gelen operasyon koduna göre işlemleri gerçekleştirip ACC saklayıcısına yazmaktadır. Saklayıcılar, Aritmetik İşlem Ünitesi ve RAM'e verilerin birbirleri arasında transferinden sorumludurlar. İşlemci içi veri akışını yönetir.

3 Geliştirilen Yazılım

```
,0: 0000_110011 // LOD 51
1: 0011_110001 // SUB 49
2: 0111_001010 // JMZ 10
3: 0000_110000 // LOD 48
4: 0010_110010 // ADD 50
5: 0001_110000 // STO 48
6: 0000_110001 // LOD 49
7: 0010_101110 // ADD 46
8: 0001_110001 // STO 49
9: 0110_000000 // JMP 0
10: 0000_110000 // LOD 48
11: 0001_110100 // STO 52
12: 1001_000000// HLT
```

```
46: 1
48: 0
49: 0
50: 0000000101
51: 0000001010
```

0: 51. adresi acc'ye yüklüyoruz.
1: acc'dan 49. Adresi çıkarıyoruz.
2: eğer sonuç 0 ise 10. adrese atlıyoruz.
3: 48. adresi acc'ye yüklüyoruz.
4: elimizdeki acc'ye 50. adresteki değeri ekliyoruz.
5: Elimizdeki acc'yi 48. Adrese yüklüyoruz.
6: 49. Adresin değeri acc'ye yüklüyoruz.
7: elimizdeki acc'ye 46. adresteki değeri ekliyoruz.
8: acc'yi 49. Adrese yüklüyoruz.
9: 0. Adrese atlıyoruz.
10: acc'ye 48. Adresi yüklüyoruz.
11: acc'yi 52. Adrese yüklüyoruz.
12. HLT yazılımı bitiriyor.

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

4 Sonular

Bu proje iin ncelikle takım olarak anlayabildiğimiz kadarıyla ilerlettik. Durum durum CPU'muzu oluřturmaya bařladık. Projeyi ilk bitiriřimizde kk aksaklıklar ıktı durum 2'de ancak hocanın yardımıyla o sorunu hallettik. O sorunu hallettikten sonra her řey gayet iyi alıřıyordu. CPU'nun tasarımından ok video ve slayt hazırlamada sorun ektik. O bizim iin daha zorlu bir sreti. Ama takımca bu sorunun da stesinden geldik. Bu alıřma bize ekipe alıřmayı, ekipe alıřmanın zorluklarını ve saėladıėı yararları ğretti. Proje sayesinde bu alanda ok kk de olsa deneyim sahibi olmuř olduk.

Hazırlanan sunum video'su adresi:

<https://www.youtube.com/watch?v=nyh9Nurv1mQ>

Dosyaların github adresi: <https://github.com/brktnrc/FB-CPU-TasarimiTT>