



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

FB-CPU V1

***Proje Teslim Raporu
6 Ocak 2020***

***AHMET HAZAR HASPOLAT, EKREM BÜYÜKKKYA, MUSTAFA
BERK TAŞKIN, ÖMER SAİT YORULMAZ ...***

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	4
4	SONUÇLAR.....	6

1 Giriş

1.1 Projenin Amacı

Bu proje kapsamında FB_CPU isimli bir işlemci tasarlayıp, bu tasarım üzerinden simülatör aracılığıyla, makine dili ile yazılan kod parçacıkları çalıştırılacaktır. Yapılan tasarım sonucunda ram, kontrol üniteleri ve saklayıcılar gözlemlenebilecek ve aralarındaki ilişkiler bütünü ile birlikte bu makine kodlarının nasıl çalıştığı deneyimlenebilecektir.

1.2 Proje Ekibi

AHMET HAZAR HASPOLAT – 190301012

Çamlıca Doğa kolejinde lise eğitimi tamamlayan Haspolat, daha sonrasında Fenerbahçe Üniversitesi Bilgisayar Mühendisliğine girmeye hak kazanarak, kariyerine bilgisayar bilimlerinde devam etmeyi planlamaktadır.

EKREM BÜYÜKKAYA – 19030101

Deepzen ltd, Fintables firmalarında software developer olarak çalışmaktadır.

MUSTAFA BERK TAŞKIN – 190301021

Bursa Doğa kolejinden mezun olan Taşkın, Fenerbahçe Üniversitesi Bilgisayar Mühendisliğinde eğitim hayatına devam etmektedir.

ÖMER SAİT YORULMAZ – 190301025

Ücret toplama sistemi olan HGS bünyesinde yaklaşık 2 yıldır software developer olarak çalışmaktadır.

2 Sistem Mimarisi

2.1 Kullanılan Araçlar

Proje kapsamında 2 araç kullanılmaktadır.

Von Neumann Simulatörü:

Tasarımı gerçekleştirilen ve veri akışlarının daha rahat bir şekilde gözlemlenebildiği simülatördür.

<http://levant.tc/araclar/vonneumann/> erişim adresi. Açılan adresten sağ tarafta bulunan komutlar ve değişkenler ram içerisinde ki verilerdir. Komutları makine dili olarak değil, assembly olarak ifade edebiliyoruz.

Logisim-Evolution:

Logisim Evolution, digital mantık devrelerini tasarlamak ve bunları simüle etmek için kullanılan java tabanlı bir eğitim aracıdır. Bu devrelerle alakalı kavramları öğrenmeyi kolaylaştırmakla birlikte, asıl tasarım bu araç kullanılarak yapılacaktır.

2.2 Tasarım

Tasarlanan cpu, dört temel eleman içermektedir.

Saklayıcılar

RAM

İşlem Ünitesi

Kontrol Ünitesi

Kabataslak olarak işlemci, RAM'e kullanıcı tarafından yazılan değerleri okur ve tasarlanan üniteler tarafından gerekli işlemleri yaparak sonuçları tekrardan RAM'e yazar.

Daha Detaylı olarak, aslında sistem bir durum makinesi (state-machine) olarak çalışmaktadır. Durumun o an ki değerine göre PC, IR ve ACC değerleriyle birlikte, RAM'den okuduğu değerlerin son 4 [9-6] bitlerine bakarak operasyon kodunu ayırıştırıp muxlar ve comperatorlar yardımıyla gerekli olan ünitlere giderek işlemleri tamamlayıp gereken durumlarda durum, program counter, IR ve ACC gibi değerleri değiştirip süreçlere devam ederek işlemleri tamamlamakta ve nihai sonuçları ram'e geri yazmaktadır.

3 Geliştirilen Yazılım

Test yazılım 1 :

Bellekte 50. ve 51. adresteki iki sayının toplamını 52 nolu adrese kaydeder. Adım adım aşağıdaki sırayla işlem yapılmaktadır.

- 50. Adresteki değeri LOD komutuyla (0000) ACC'nin içerisine alır.
- Daha sonra ADD komutuyla (0010) 51. Adresteki değeri ACC'nin içerisindeki değer ile toplar ve tekrardan ACC'nin içerisine yazar.
- STO komutuyla (0001) ACC'nin içerisinde ki değeri 52 adrese yazar.
- Yazdıktan sonra halt komutuyla (1001) işlemi sonlandırır.

Test Yazılım 2 :

Bellekte 50. ve 51. adresteki iki sayının çarpımını 52. Adrese yazar. Adım adım aşağıda ki şekilde takip edilebilmektedir.

- LOD komutuyla (0000) 50. Adresteki değeri ACC'nin içerisine alır.
- ADD komutuyla (0100) 51. Adresteki değeri ACC'nin içerisinde ki değer ile çarpıp tekrardan ACC'nin içerisine yazar.
- STO komutuyla (0001) ACC'nin içerisinde ki değeri 52. Adrese yazar.
- Halt komutuyla (1001) programı bitirir.

Test Yazılım 3 :

Bellekte ki 50. ve 51. adresteki iki sayının çarpımını, çarpma işlemi kullanmadan 52. Adrese hesaplayıp yazar.

- LOD komutuyla 51 adresteki değeri ACC'nin içerisine alır.
- SUB komutuyla 49 adresteki değeri alıp ACC'den çıkartıp tekrardan ACC'ye geri yazar.
- JMZ 10 komutuyla ACC' değeri 0 olduğunda 10 satıra atlar değilse döngüye devam eder.
- LOD komutuyla 48. Adresteki ACC'nin içerisine temp değer olarak alır.
- ADD komutuyla 50. Adresteki değeri ACC'nin üzerine ekler.
- STO komutuyla ise 48. Adrese temp değer olan ACC'nin değerini yazar
- LOD komutuyla 49. Adresteki değeri ACC'nin içerisine alır.

-
- ADD komutuyla ACC içerisine aldığı değere 46. Adresteki değeri ekler.
 - STO komutuyla ACC'nin içerisinde ki değeri ise 49 adrese geri yazar.
 - JMP komutuyla ACC'nin değeri pozitif ise 0. Satıra geri gider.
 - 2. Adımda JMZ komutuyla ACC'nin değeri 0 ise 10 adıma gidecek olan program şöyle devam eder ;
 - o LOD değeriyle 48. Adresteki değeri ACC'nin içerisine alır.
 - o STO komutuyla ACC'nin içerisinde ki değeri 52 nin içerisine yazar
 - o HALT komutuyla işlemi sonlandırır.

Kurulmuş döngü ile iki değerden bir tanesi baz değer diğerini ise tekrar etme sayısı olarak belirledik. Örnek vermek gerekirse 3 ve 5 sayıları. 5 Sayısını baz değer olarak alıp , 3 sayısını ise 3 kere tekrar etme olarak kullanıp toplama işlemlerini yaptık. Sürekli azaltılan tekrar etmek değeri (3) 0 olduğunda çarpma işlemi bitmiş oluyor.

4 Sonular

Yapılan bu sistem ile, iřlem srelerinin grselleřtirilmesi sonucunda gerekli iř kırımları daha net gzlemlenebilmiř ve etkili ğrenme ıktıları alınmıřtır.

Hazırlanan sunum video'su adresi: <https://www.youtube.com/watch?v=9JtBQp-6RMg>

Dosyaların github adresi: <https://github.com/fbuni/BLM101>