



**Fenerbahçe Üniversitesi**  
**BLM 205 – Nesneye Yönelimli Programlama**  
**FBCPU Assembly Makine Kodu Çevirici**  
**Proje İçeriği**

**Veriliş Tarihi: 13.12.2020**

**Teslim Tarihi ve Yeri: 15.01.2021, Ders Saatlerinde, Elektronik olarak**

**1. Tanım:**

Fenerbahçe İşlemcisi (FBCPU) isminde bir işlemcinin kendine özel bir komut seti (Instruction Set)'inden, FBCPU'nun makine koduna dönüştürecek ve aynı zamanda makine kodundan da geriye FBCPU Assembly'e dönüşüm yapabilecek bir çevirici aracı tasarlanacaktır.

**2. Proje Ekibi:**

Proje 4 kişiye kadar olan ekiplerden oluşacaktır. Her bir proje ekibinin bir sorumlusu olacaktır. Öğrenciler 1, 2, 3 veya 4 kişilik kendi proje ekiplerini ve proje sorumlusunu belirlemelidirler.

Ekiplerin kurulması ve proje sorumlusunun belirlenmesi en geç **18.12.2020** tarihine kadar tamamlanmalıdır. Ekip sorumluları, Teams'te açılmış olan "Proje Ekip Sorumluların Takımlarını Bildirmesi" başlığının altına, ekip üyelerinin isimlerini göndermelidirler.

**3. Kullanılacak Araçlar:**

Proje kapsamında, ödev ve LAB'larda kullanılan Microsoft'un derleyicisi olan Visual Studio Community kullanılacaktır.

**4. Tasarım Gereksinimleri**

FB-CPU'nun desteklediği işlemler Tablo 1'de verilmektedir.

Tablo 1. FB-CPU ISA (Instruction Set Architecture)

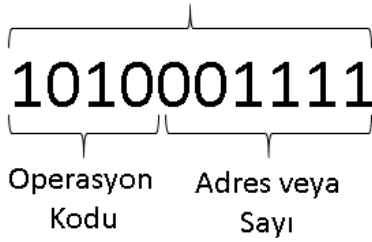
Komut Adı	Görevi	Operasyon Kodu
LOD ADDR	Yükleme (Load), Bellekteki verilen adresin içerisinden değeri alıp, ACC saklayıcısına yerleştirir. $ACC = *(ADDR)$	0000
STO ADDR	Kaydetme (Store), ACC'nin içerisindeki değeri alıp, bellekte verilen adrese yazar. $*(ADDR) = ACC$	0001
ADD ADDR	Bellekteki verilen adresteki değeri alır, ACC ile toplayıp, ACC'nin üzerine yazar. $ACC = ACC + *(ADDR)$	0010

SUB ADDR	Bellekteki verilen adresteki değeri alır, ACC ile çıkartıp, ACC'nin üzerine yazar. ACC = ACC - *(ADDR)	0011
MUL ADDR	Bellekteki verilen adresteki değeri alır, ACC ile çarpıp, ACC'nin üzerine yazar. ACC = ACC * *(ADDR)	0100
DIV ADDR	Bellekteki verilen adresteki değeri alır, ACC ile bölüp, ACC'nin üzerine yazar. ACC = ACC / *(ADDR)	0101
JMP SAYI	PC = Sayı olur.	0110
JMZ SAYI	ACC'ın değeri 0 ise, verilen sayı değerini PC'e atar, değilse işlem yapmaz.	0111
NOP	No Operation, hiçbir işlem yapılmaz.	1000
HLT	Uygulama durur	1001

İşlemci 10 adet komutu desteklemektedir.

Şekil 1'de FB-CPU'nun 10 bitlik komutunun, operasyon ve adres için bitlerinin ayrılması gösterilmiştir.

Komut (Instruction) (10 Bit)



Şekil 1. FB-CPU Örnek Komut Binary Gösterimi

Tablo 2, 3 ve 4'te örnek Assembly ve Makine kodu dönüşümleri verilmektedir.

Tablo 2. FB-CPU ISA (Instruction Set Architecture)

0: LOD 50	↔	0: 0000110010
1: ADD 51		1: 0010110011
2: STO 52		2: 0001110100
3: HLT		3: 1001000000
50: 5		50: 0000000101
51: A		51: 0000001010

Tablo 3. FB-CPU ISA (Instruction Set Architecture)

0: LOD 50	↔	0: 0000110010
1: MUL 51		1: 0100110011
2: STO 52		2: 0001110100
3: HLT		3: 1001000000
50: 5		50: 0000000101
51: A		51: 0000001010

Tablo 4. FB-CPU ISA (Instruction Set Architecture)

0: LOD 51		0: 0000110011
1: SUB 49		1: 0011110001
2: JMZ 10		2: 0111001010
3: LOD 48		3: 0000110000
4: ADD 50		4: 0010110010
5: STO 48		5: 0001110000
6: LOD 49		6: 0000110001
7: ADD 46		7: 0010101110
8: STO 49		8: 0001110001
9: JMP 0		9: 0110000000
10: LOD 48		10: 0000110000
11: STO 52		11: 0001110100
12: HLT		12: 1001000000
46: 1		46: 0000000001
47: 0		47: 0000000000
48: 0		48: 0000000000
50: 5		50: 0000000101
51: A		51: 0000001010

Makine kodundan assembly'e dönüşüm işleminde bir bilinmezlik oluşmaktadır. Bu bilinmezlik, aslında sabit sayı olarak konmuş içeriklerin Instruction olarak yorumlanabileceğidir. Bunu engellemek için 40. Adresten sonraki içeriklerin sabit sayı olacağı varsayımı ile geliştirme yapınız.

## 5. Gerçekleme Yöntemi:

Geliştirilecek uygulama komut satırında 3 argüman alacaktır. İlk argüman dönüşüm türünü (Makine Kodundan Assembly veya Assembly'den Makine koduna), ikinci argüman ise giriş dosya ismi ve üçüncü argüman ise çıkış dosya ismi olacaktır.

Örnek:

donusturucu.py 0 giris.txt cikis.txt

donusturucu.py 1 giris.txt cikis.txt

0, makine kodundan assembly'e; 1 ise assembly'den makine koduna dönüştürüleceğini ifade etmektedir.

Tablo 2,3 ve 4'te verilen giriş örneklerini uygulamanızda test ederek beklenen çıktıları kontrol ediniz.

## 6. Notlandırma ve Proje Teslimi:

Bu başlık "FBCPU Assembly Makine Kodu Çevirici" projesinin teslimi ve notlandırılması hakkında bilgiler içermektedir.

### 6.1. Notlandırma:

Projenin **iki** ana değerlendirme kriteri vardır. Her iki kriter 50 şer puandır.

İlk kriter "FBCPU Assembly Makine Kodu Çevirici" 'nin **doğru çalıştırılması ve objeye dayalı programlama dili yaklaşımlarının kullanılmış olmasıdır.**

İkinci kriter ise **Proje Teslim Dokümanı ve Sunumdur.**

- **Proje Teslim Dokümanı:**

Öğrenciler, proje raporlarını verilen “Proje Teslim Dokümanı” ‘nın içerisini doldurarak yapacaklardır.

Proje Teslim Dokümanı:

[http://levent.tc/files/courses/object\\_oriented\\_design/projects/BLM205\\_proje\\_teslim\\_dokumani.docx](http://levent.tc/files/courses/object_oriented_design/projects/BLM205_proje_teslim_dokumani.docx)

Proje teslim dokümanı tamamlanıp PDF’e dönüştürüldükten sonra, PDF’in sonuna Latex (Genellikle bilimsel çalışmaların yazımında kullanılan, görsel olarak Word ile yazılmış dokümanlara göre çok daha başarılı olan bir doküman oluşturma dilidir) ile tasarlanmış özgeçmişler eklenecektir. Özgeçmişin minimumda kullanılan diller, yabancı diller, projeler (kısa açıklamaları ile), kullanılan kütüphaneler (varsa, örn. OpenCV), not ortalaması, github adresi, youtube adresi bilgileri verilmelidir.

Latex ile özgeçmiş oluşturma:

[http://levent.tc/files/courses/object\\_oriented\\_design/projects/BLM205\\_latex\\_ozgecmis\\_olusturma.pdf](http://levent.tc/files/courses/object_oriented_design/projects/BLM205_latex_ozgecmis_olusturma.pdf)

Latex ile özgeçmiş oluşturulduğunda PDF uzantılı dosyalar elde edilecektir. Proje teslim dokümanının PDF’inin arkasına eklemek için PDF birleştirici araçları ([https://www.ilovepdf.com/merge\\_pdf](https://www.ilovepdf.com/merge_pdf) gibi) kullanılmalıdır. İşlem sonucunda tek bir PDF dosyası elde edilecektir.

- **Proje Sunumu:**

Powerpoint üzerinde ortalama 5 dakika (4-6 dakika arası) sürecek bir sunum hazırlayarak kayıt etmelidirler. Kayıt işlemi, cep telefonu veya bilgisayar ekran kayıt yazılımları (Screen-Recorder, Bandicam vb...) ile yapılabilir.

Sunum, ekip üyeleri içinden biri tarafından, projenin nasıl yapıldığı, nasıl çalıştığı vb.. konularının powerpoint slaytları üzerinden anlatılırken kaydedilmesi ile olmalıdır. Sunum video’sunda powerpoint slaytları okunabilir ve konuşmacının sesinin anlaşılır olması gerekmektedir. Powerpoint slayt görünüm tasarımı istenildiği gibi yapılabilir.

Proje ekibinin tamamı, notlarını bu değerlendirmeye göre alırlar.

## 6.2. Teslim:

Projenin teslimi için aşağıdaki adımların gerçekleştirilmesi gerekmektedir. İstenen dosyaları sadece proje ekip sorumlusunun getirmesi, Teams ve Github (Çok yaygın bir açık kaynak kod paylaşım platformudur)’a yüklemelidir.

Teams’te açılmış olan “Proje Teslim” sayfasına aşağıdaki dosyaların yüklenmesi gerekmektedir.

- Proje kaynak kodları
- Hazırlanan powerpoint sunum dosyası (.ppt uzantılı dosya)
- Proje Teslim Dokümanı (Word formatında yüklenmelidir)
  - Dokümanın alt başlıkları doldurulmalıdır.
  - Kaydedilen powerpoint sunum video’su youtube’a yüklenip, adresi, dokümanın sonuçlar bölümündeki açılmış yere link’i yazılmalıdır (Video’nun herkes’e görünür olmamasını istiyorsanız, youtube’a yükledikten sonra liste dışı seçeneğini seçerek, sadece link’e sahip olan kişilerin görmesini sağlayabilirsiniz).

- Teams'e yüklenen tüm dosyalar (Proje kaynak kodları, ppt uzantılı sunum dosyası ve Proje Teslim Dokümanını (PDF formatında)), github.com sitesine üye olup, yüklenip, dokümanın sonuçlar bölümündeki yere link'i yazılmalıdır.