

# Mantıksal Sistem Tasarımı – BLM 201

## Hafta 10: FB CPU RTL Tasarımı



Fenerbahçe Üniversitesi

# 10. Hafta İçeriği

- FB-CPU
  - RTL Tasarım

## FB-CPU

- FB-CPU işlemcilerin temel çalışma prensiplerini anlatmak için, eğitim amaçlı bir işlemcidir.
- Von Neumann mimarisi ile tasarlanmıştır.

## FB-CPU

İşlemci;

- Bellek (RAM)
- Saklayıcılar
- Kontrol Ünitesi
- Aritmetik İşlem Ünitesi

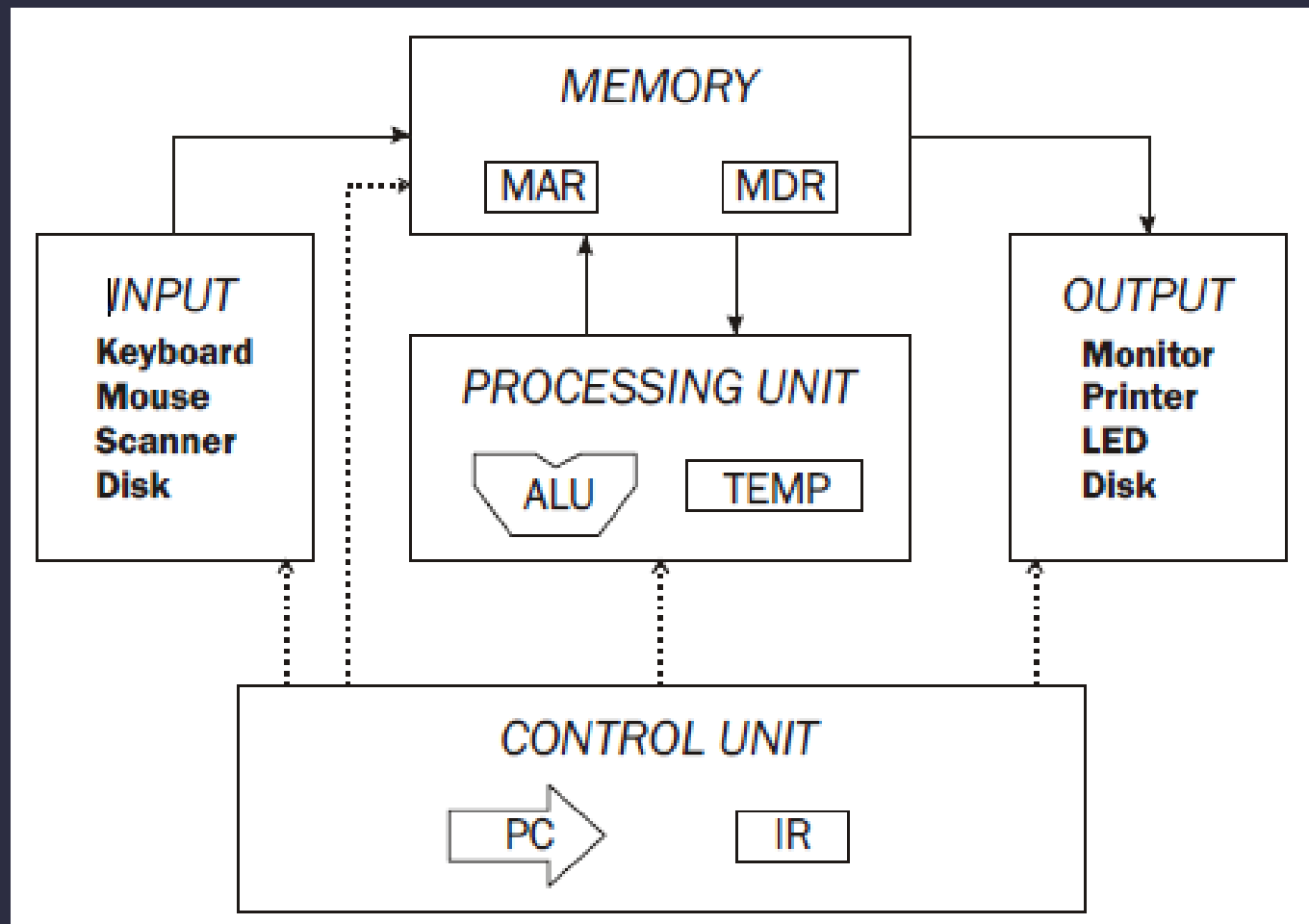
yapılarını içermektedir.

## FB-CPU

İşlemci tasarlanırken kullanılacak araçlar;

- Von Neumann Simulatorü
  - İşlemci için algoritma geliştirmede faydalıdır
- Xilinx Vivado
  - RTL tasarım yapılacaktır

# FB-CPU



# FB-CPU

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

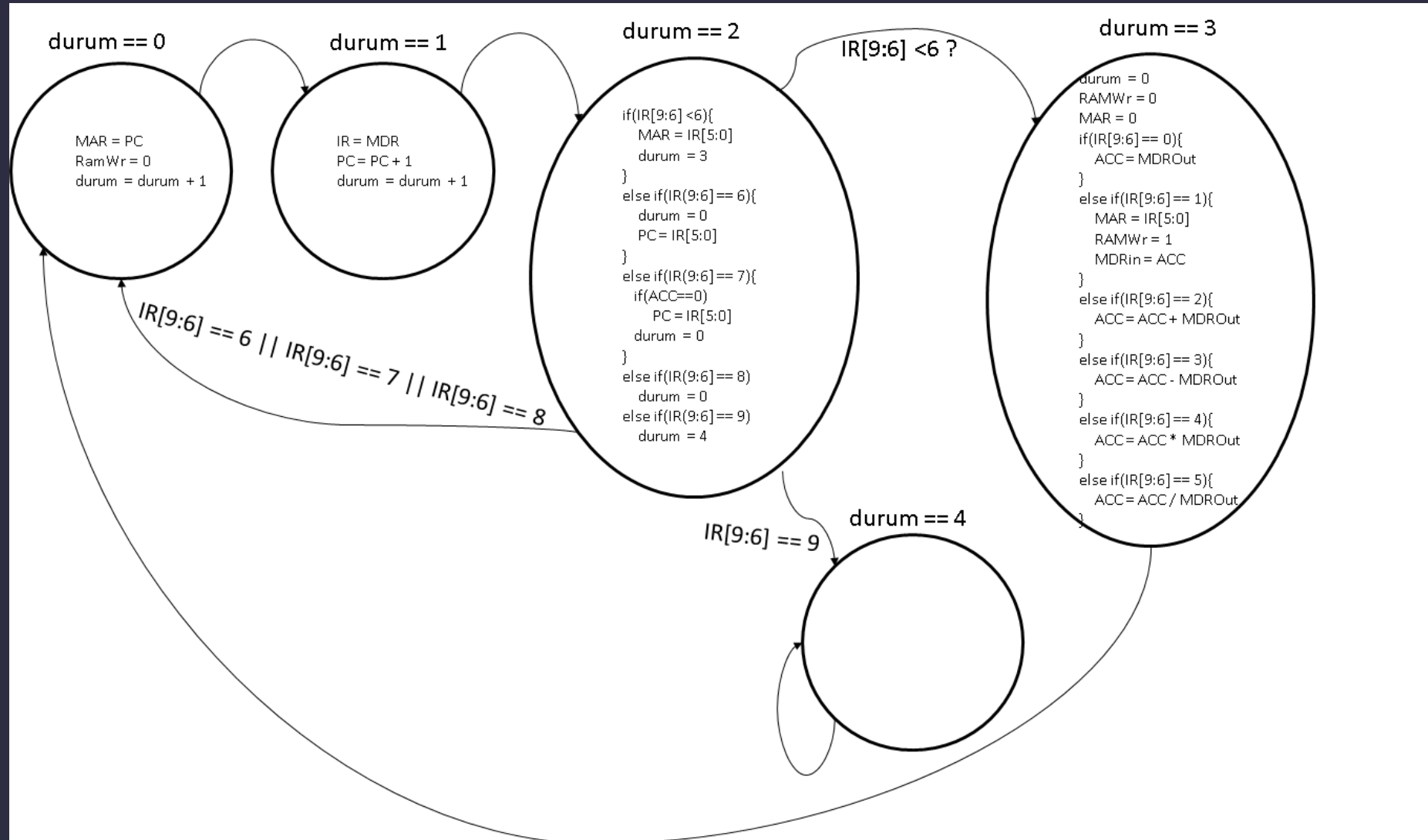
Komut (Instruction) (10 Bit)

1010001111

Operasyon  
Kodu

Adres veya  
Sayı

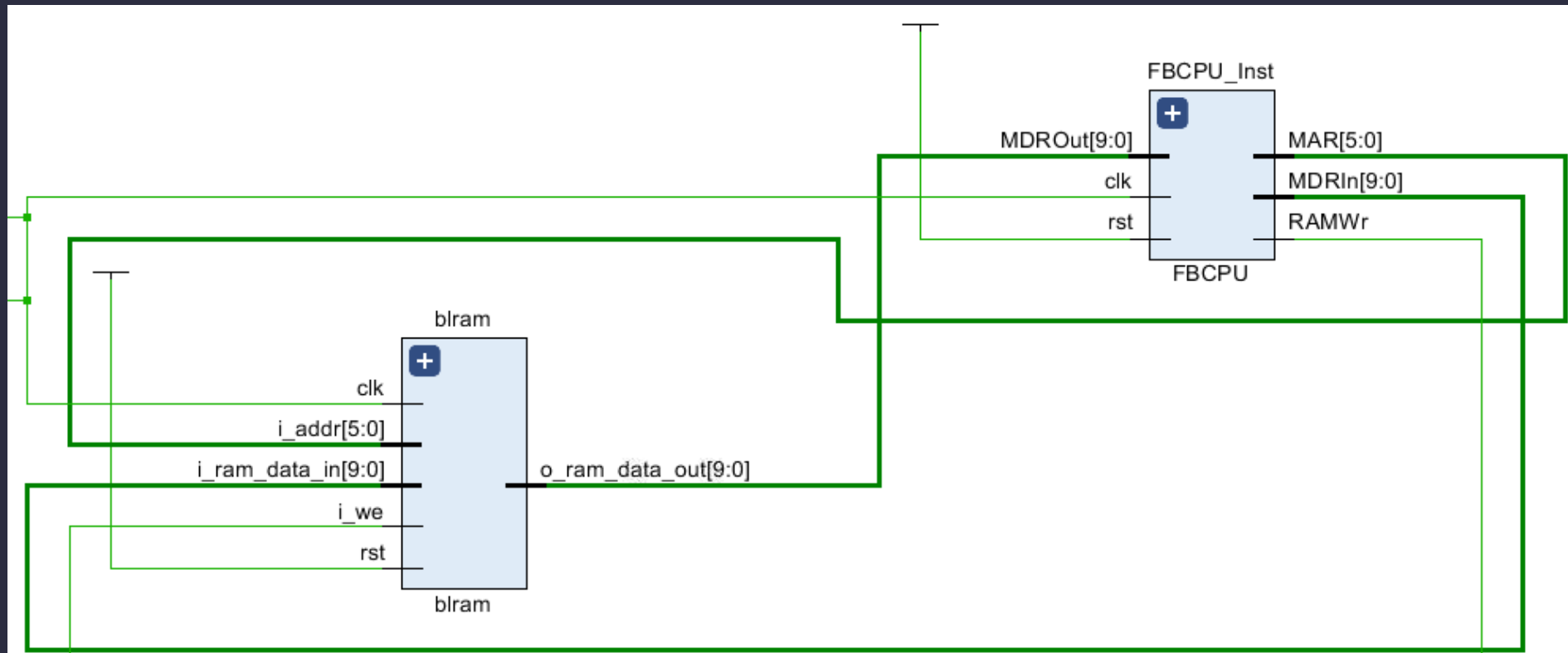
# FB-CPU





# FBCPU

İşlemci tasarlanırken kullanılacak araçlar;



## Tasarımda 4 adet saklayıcı bulunmaktadır.

- durum: Durum makinasında, hangi durumda olduğunu bilgisi tutulur.
- PC: RAM'deki hangi adresteki komutun çalıştığı bilgisi tutulur.
- IR: O anda çalışan komutun kendisi tutulur.
- ACC: Geçici saklama alanı

Tasarımda 4 adet saklayıcı bulunmaktadır.

- Saklayıcı tanımları

```
always@ (posedge clk) begin
    durum      <= #1 durumNext;
    PC         <= #1 PCNext;
    IR         <= #1 IRNext;
    ACC        <= #1 ACCNext;
end
```

## FB-CPU

- Diğer tüm saklayıcılar, durum saklayıcısının değişimine göre çalışacaktı. Yani durum'un değerine göre tüm saklayıcıların giriş sinyalleri değişmektedir.
- Diğer bir değiş ile, durum saklayıcısının değerine göre saklayıcıların üzerine başka başka sinyaller atanmakta, sistemin ilerlemesi durum sinyaline bağlıdır.

## FB-CPU

- Tasarımda giriş çıkış portlarına bağlı olan bellek sinyalleri aşağıda verilmektedir.
- **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 bitlidir. Saklayıcı RAM'in içerisindedir.
- **MDRIIn (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.

## Eksik Ünitelerin Tasarımı

- Verilen başlangıç tasarımında durum 0 ve durum 1 için tasarım verilmiştir.
- LAB'da durum 2'nin tasarımı yapılacaktır.
- İşlemcinin çalışır olması için durum 3'ün tamamlanması gerekmektedir.

# FB-CPU

## Örnek Yazılım 1

FB-CPU için bellekte 50 ve 51 adresteki iki sayının toplamını 52 no'lu adrese kaydeden uygulamayı geliştiriniz.

- 0: 0000\_110010 // LOD 50, (ACC = \*50), Hex = 32
- 1: 0010\_110011 // ADD 51, ACC = ACC + (\*51), Hex = B3
- 2: 0001\_110100 // STO 52, (\*52) = ACC, Hex = 74
- 3: 1001\_000000 // Halt, Hex = 240
- 50: 0000000101 // Hex = 5
- 51: 0000001010 // Hex = A

# FB-CPU

## Örnek Yazılım 2

FB-CPU için bellekte 50 ve 51 adresteki iki sayının çarpımını 52 no'lu adrese kaydeden uygulamayı geliştiriniz.

- 0: 0000\_110010 // LOD 50, (ACC = \*50), Hex = 32
- 1: 0100\_110011 // ADD 51, ACC = ACC \* (\*51), Hex = 133
- 2: 0001\_110100 // STO 52, (\*52) = ACC, Hex = 74
- 3: 1001\_000000 // Halt, Hex = 240
- 50: 0000000101 // Hex = 5
- 51: 0000001010 // Hex = A



# FB-CPU

## Örnek Yazılım 3

FB-CPU için bellekte 50 ve 51 adresteki iki sayının çarpımını 52 no'lu adrese kaydeden uygulamayı geliştiriniz. Ancak çarpma operasyonunu kullanmayınız. Çarpma işlemi için 50'deki sayıyı 51'deki sayı defa toplayıp 52 no'lu adrese yazınız. Gerekli değişkenler için istediğiniz adresleri kullanabilirsiniz.

- 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

# FB-CPU

## Örnek Yazılım 3

- 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,  $*52 = \text{ACC}$ , 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

FB-CPU

## FB-CPU Teslimi

Proje Teslim Dokümanı ve Sunum

## FB-CPU Teslimi

- Proje Teslim Dokümanı.
- Ayrıca LMS'te açılmış olan “Proje Teslim” sayfasına aşağıdaki dosyaların yüklenmesi gerekmektedir.
- Vivado üzerinde yapılan işlemci tasarımı (.v uzantılı dosya)
- Hazırlanan powerpoint sunum dosyası (.ppt uzantılı dosya)
- Proje Teslim Dokümanı (Word formatında yüklenmelidir) o 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 (İşlemci tasarım dosyası (.v), makine dilindeki iki yazılım, ppt uzantılı sunum dosyası ve Proje Teslim Dokümanını (PDF formatında)), github.com sitesine üye olup, yüklenip, dokümanın sonuçlar bölümündeki yere link'i yazılmalıdır.