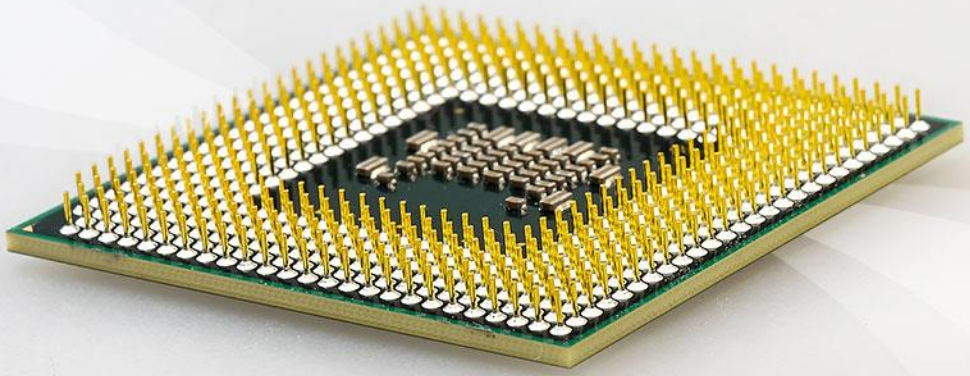
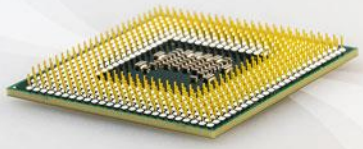


FB-CPU

Berk TUNÇ
Arda ALHAN
Evrim Arda Kalafat
Ogün Berat Gürses

TANIM

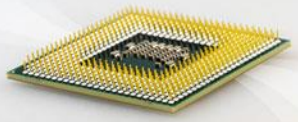


- FB-CPU işlemcilerin temel çalışma prensiplerini anlatmak için, eğitim amaçlı bir işlemcidir. Bu proje kapsamında FB-CPU isminde bir işlemcinin tasarımı ve tasarlanan işlemci üzerinde makine dili ile yazılan çeşitli kod parçacıkları yazılacaktır. Proje sonunda basit bir işlemcideki RAM, Kontrol Ünitesi ve Saklayıcıların bir arada çalışıp, makine dilindeki kod parçacıklarını nasıl yürütebildiği gözlemlenecektir.

- Von Neumann mimarisi ile geliştirilmiştir.

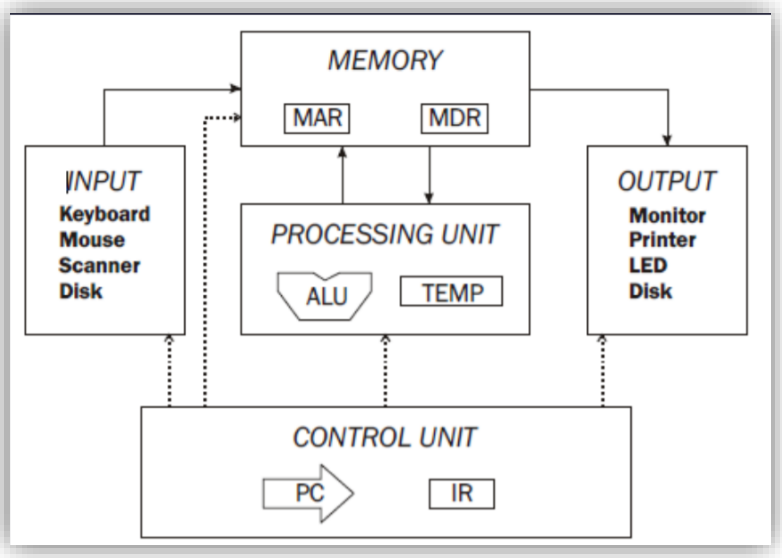
FB-CPU Von Neumann Mimarisinde tasarlanmış, bazı değişiklikler içermektedir. FB-CPU'nun desteklediği işlemler aşağıda verilmektedir.

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



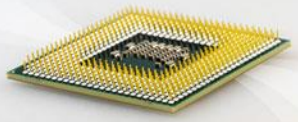
İşlemci 10 adet komutu desteklemektedir.

İşlemci;



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

yapılarını içermektedir.

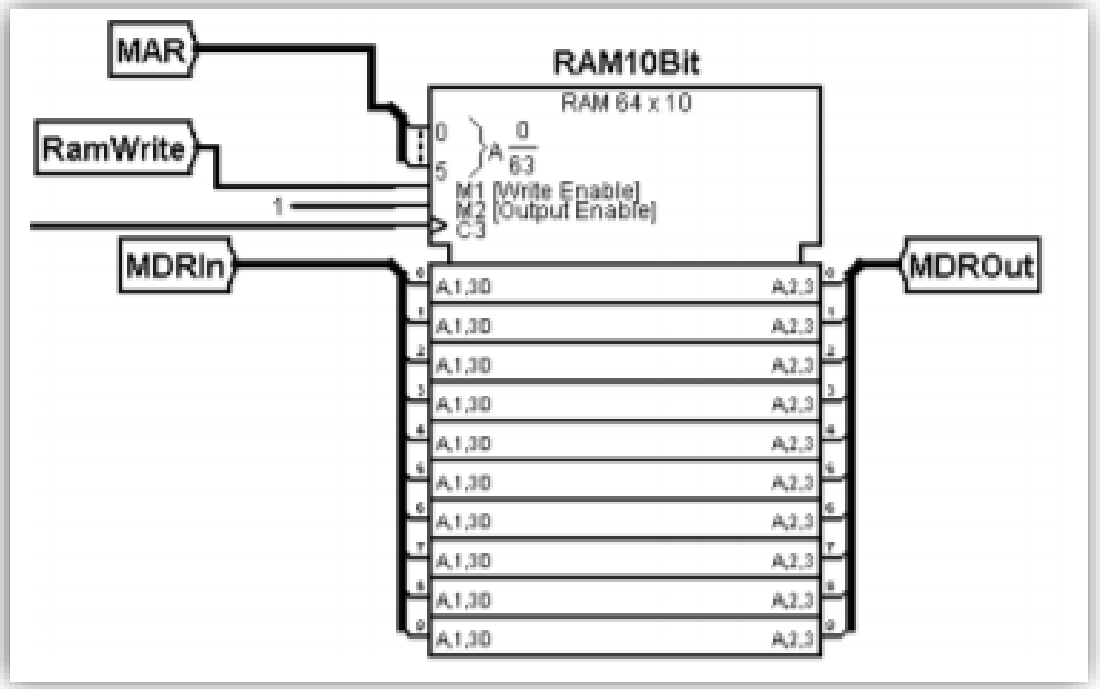


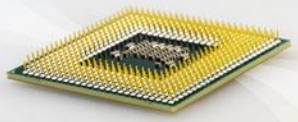
Bellek

FB-CPU'nun komutları okuyup, hesaplanan değerleri geri yazacağı bellek aşağıdaki şekilde verilmektedir.

RAM'e bağlı 4 saklayıcı ve bir clock sinyali bulunmaktadır.

RAM'e bağlı saklayıcıların görevleri saklayıcılar bölümünde açıklanmıştır.





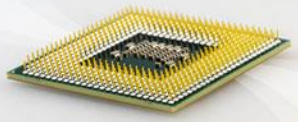
Saklayıcılar

Başlangıç tasarımında 8 adet saklayıcı bulunmaktadır.

4 tane d tipi saklayıcı ve RAM'in içerisinde 4 saklayıcı bulunmakta dır.

Her bir saklayıcı aslında birden çok bir araya gelmiş d tipi saklayıcılardan oluşmuştur.

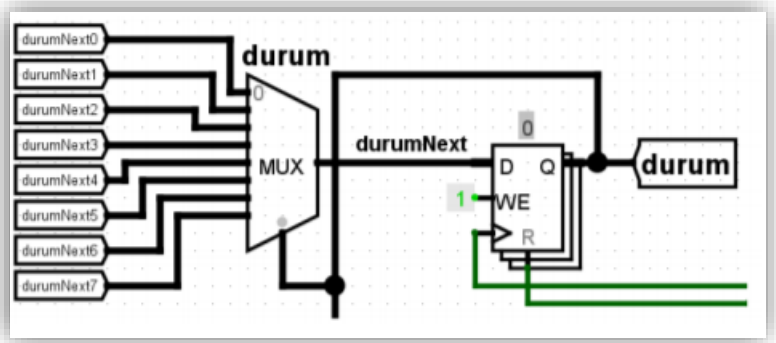
Bu saklayıcılar sonraki slaytlarda görevleri ile birlikte verilecektir.



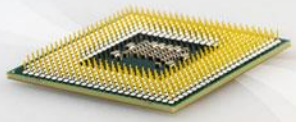
Durum Saklayıcısı

FB-CPU durum makinaları yöntemi ile gerçeklenecektir. Yani bu işlemci durum ismindeki saklayıcının değerine göre 2^3 = 8 farklı durumda çalışan bir tasarımı olacaktır.

Aşağıda durum saklayıcısını ve kendisine bağlı olan MUX yapısı görül mektedir.

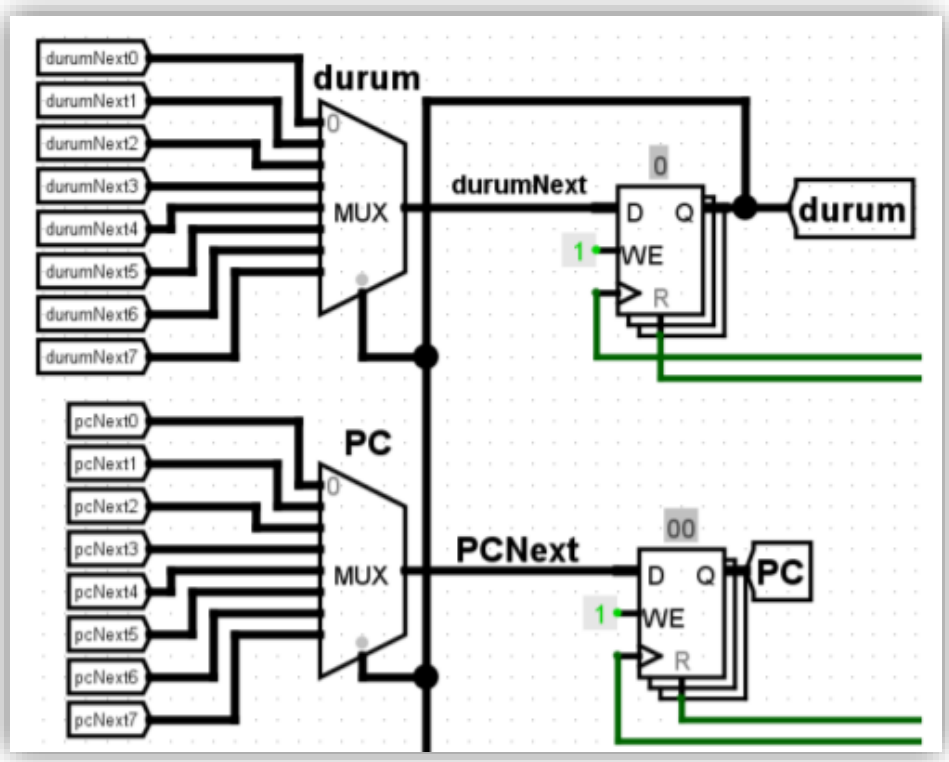


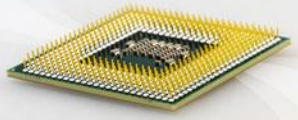
Durum saklayıcısının bir sonraki değeri, clock'un yükselen kenarında ke ndisine D girişinden gelen değer olacaktır. D girişinden gelecek olan de ğer ise MUX'un çıkışı olduğu görülmektedir. MUX yapısının select bitler ine yine durum saklayıcısının çıktısı bağlanmıştır.



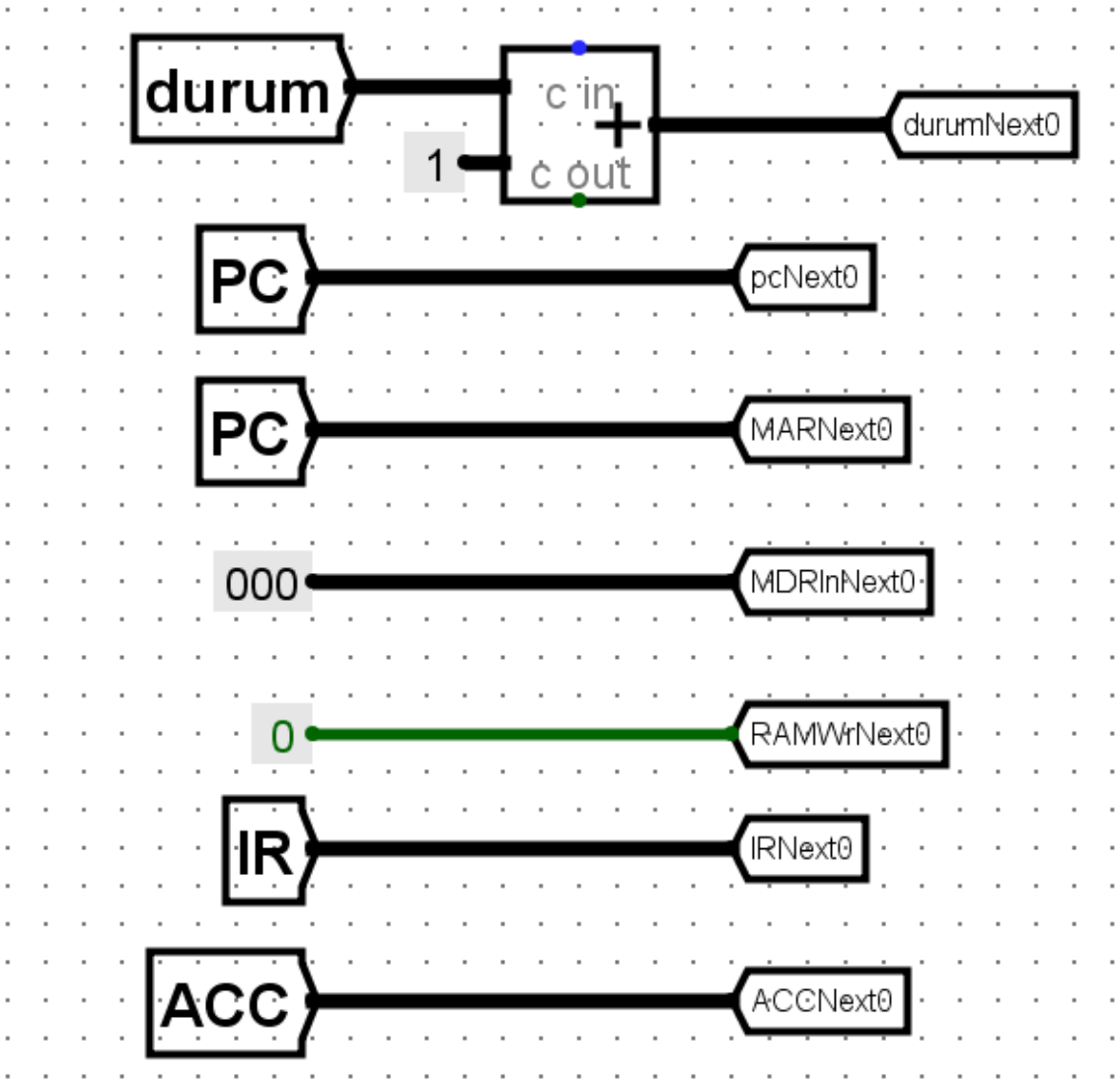
Durum Saklayıcısı

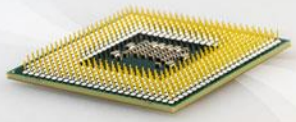
Diğer tüm saklayıcıların da MUX select bitlerine durum saklayıcısının çıktısı bağlanmıştır. Yani durum'un değerine göre tüm saklayıcıların giriş sinyalleri değişmektedir. Sistemin ilerlemesi durum sinyaline bağlıdır.



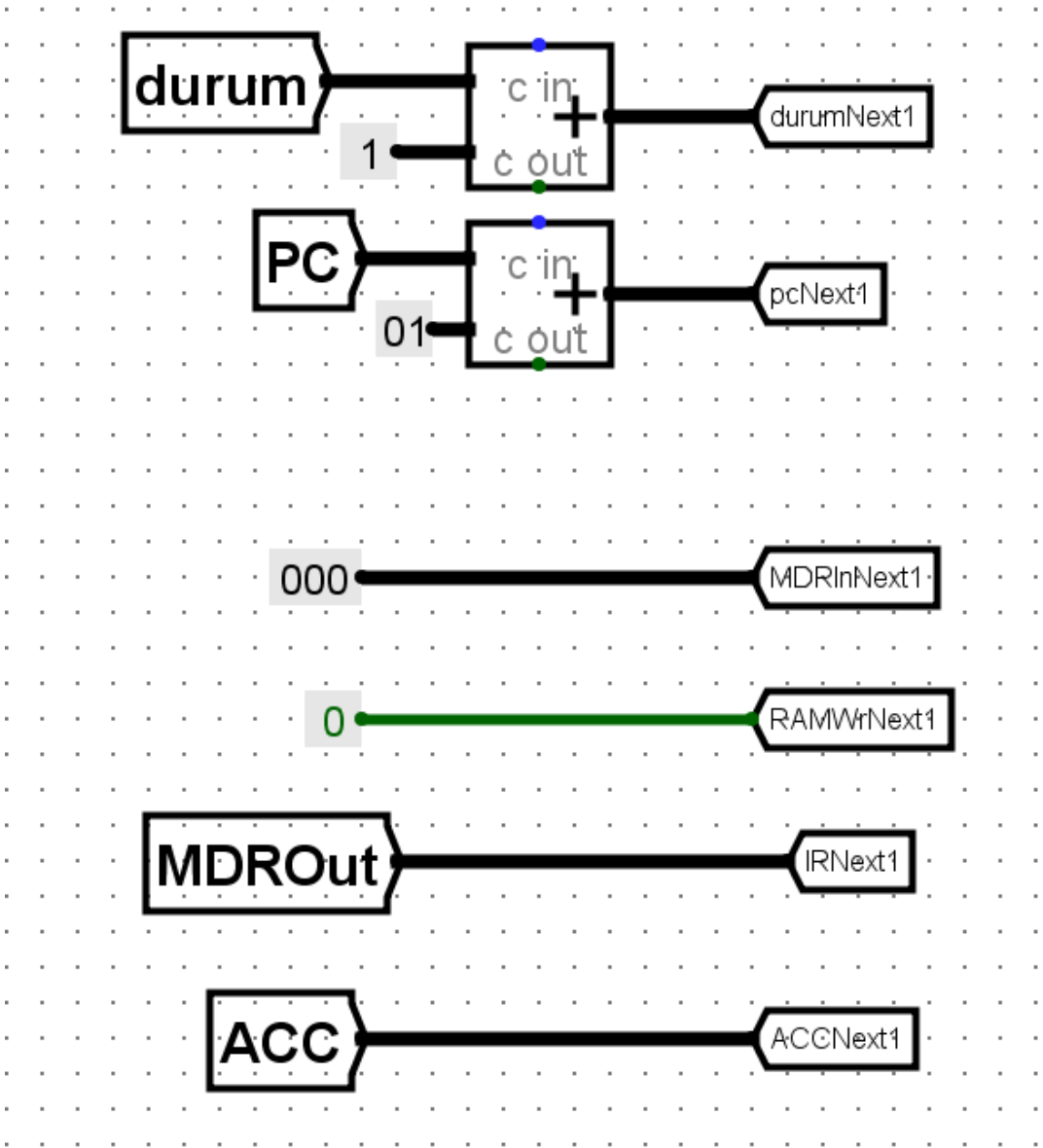


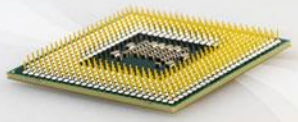
Durum == 0


durum
c in
+
1
c out
durumNext0
PC
pcNext0
PC
MARNext0
000
MDRInNext0
0
RAMWrNext0
IR
IRNext0
ACC
ACCNext0

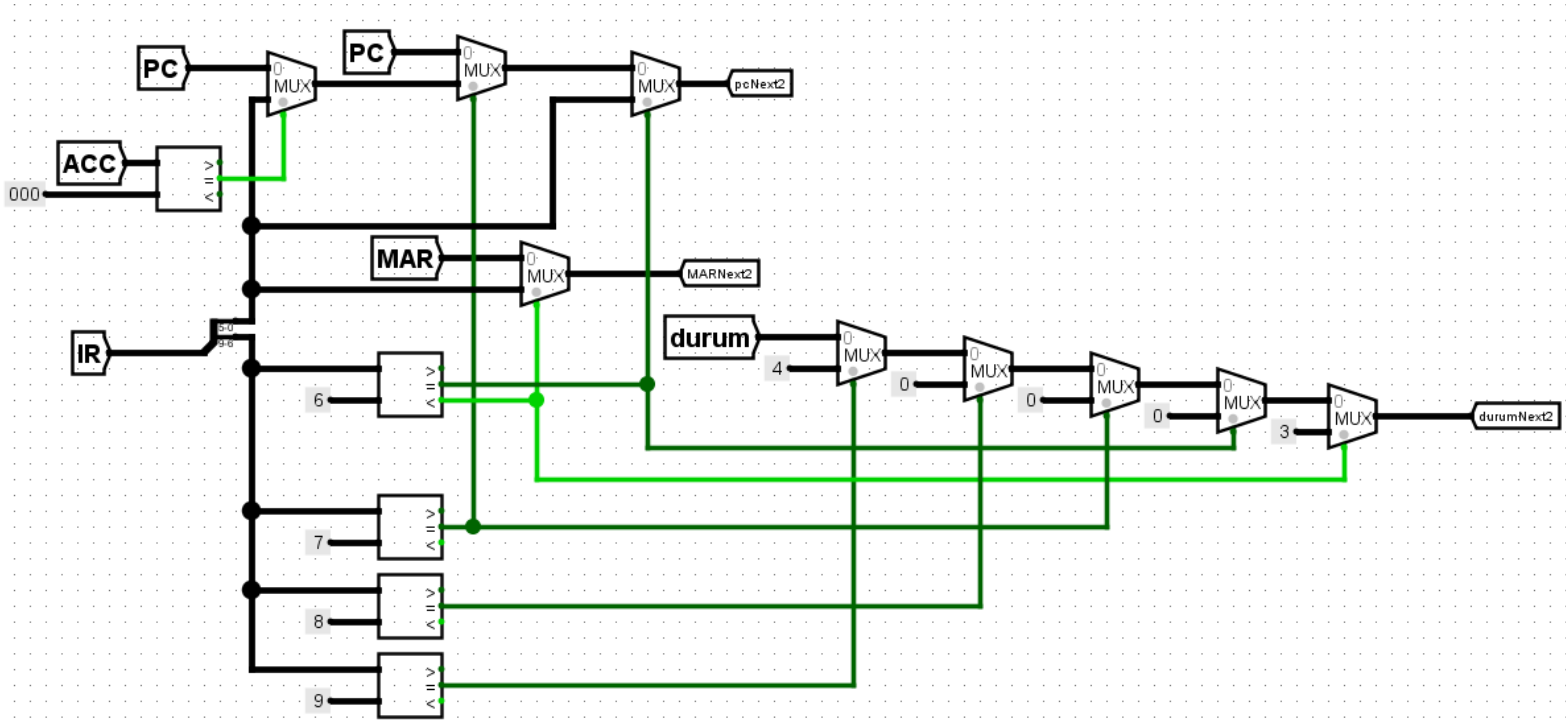


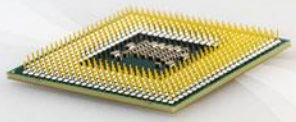
Durum == 1


durum
1
c in
c out
durumNext1
PC
01
c in
c out
pcNext1
000
MDRInNext1
0
RAMWrNext1
MDROut
IRNext1
ACC
ACCNext1

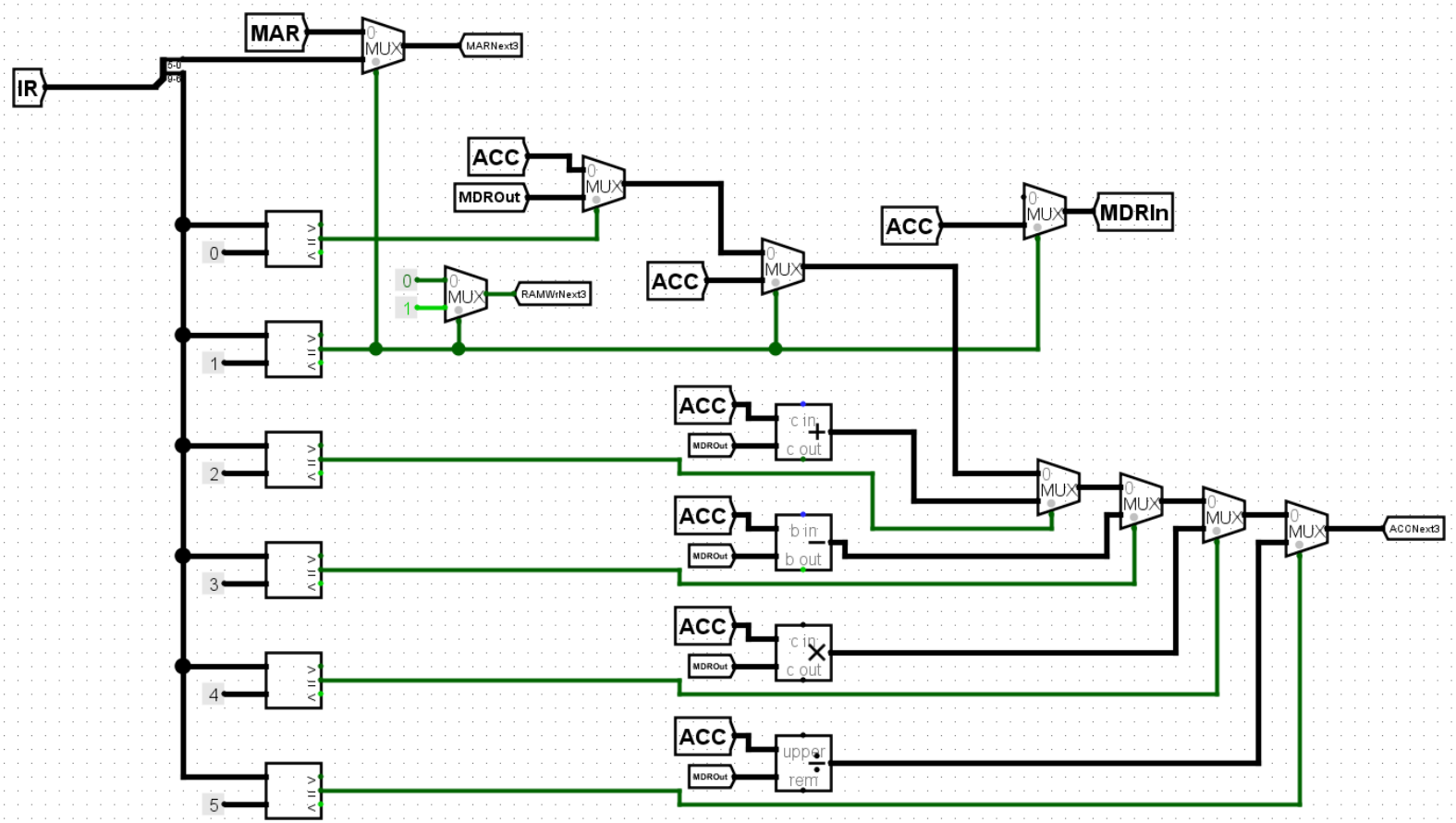


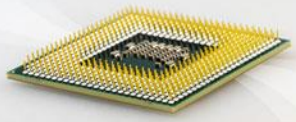
Durum == 2


PC
PC
MUX
MUX
MUX
pcNext2
ACC
000
MAR
MUX
MARNext2
IR
durum
4
MUX
0
MUX
0
MUX
0
MUX
3
MUX
durumNext2
6
7
8
9

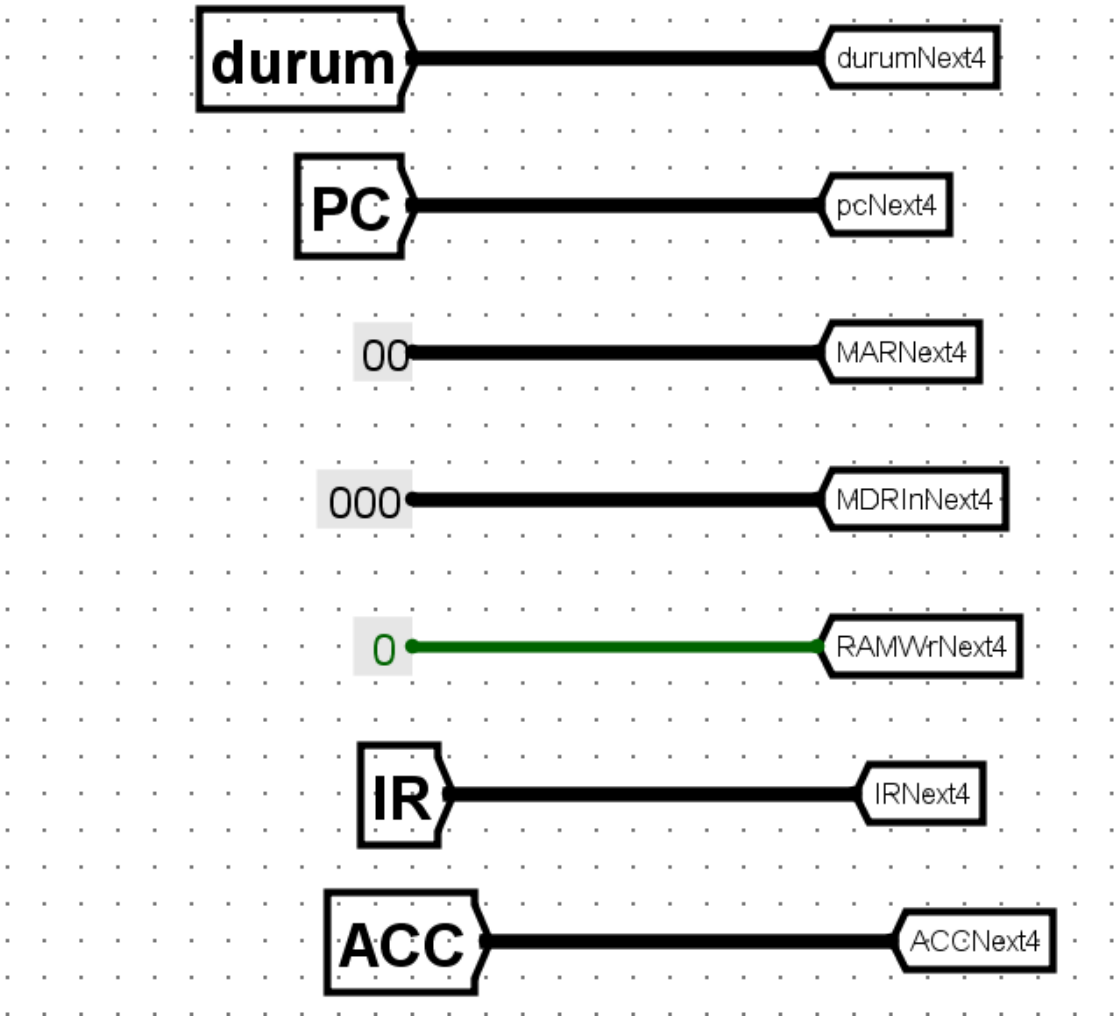


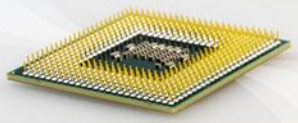
Durum == 3


MAR
IR
MUX
MARNext3
ACC
MDROut
MDRIn
RAMWrNext3
0
1
2
3
4
5
c in
c out
b in
b out
upper
rem
ACCNext3



Durum == 4

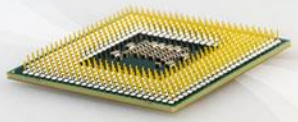

durum
durumNext4
PC
pcNext4
00
MARNext4
000
MDRInNext4
0
RAMWrNext4
IR
IRNext4
ACC
ACCNext4



PC ve MAR Saklayıcıları

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

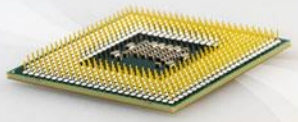
MAR (6 Bit): Memory Address Register isminde bir saklayıcıdır. Bu saklayıcı RAM'in adres girişine bağlanmıştır. RAM'in 2^6 lokasyonu olduğu için MAR 6 bitliktir. Saklayıcı RAM'in içerisindedir.



MDRIn ve RAMWr Saklayıcıları

MDRIn (10 Bit): Memory Data Register In, RAM'e bir veri yazılacağı zaman kullanılan saklayıcıdır. RAM'in bir lokasyonu 10 bitlik olmasın dan 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 içerisin dedir.

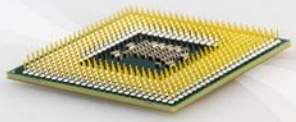


MDROut IR ACC Saklayıcıları

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.

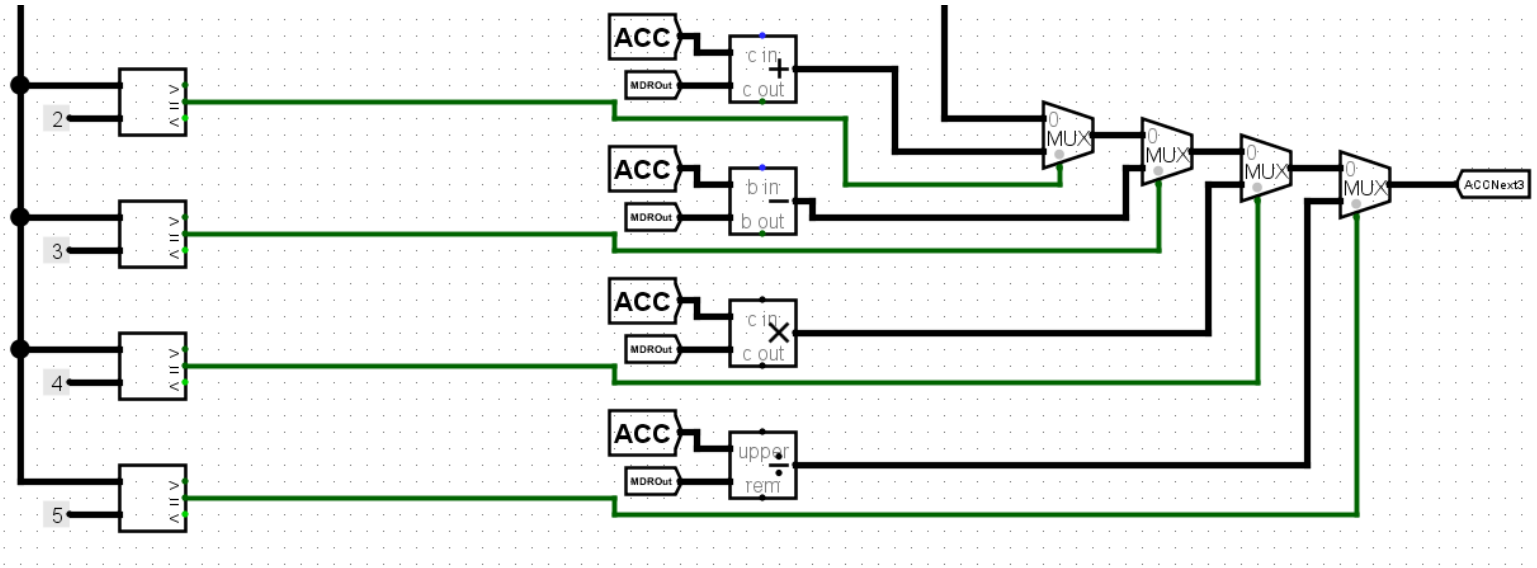
IR (10 Bit): Instruction Register, RAM'den okunan kodun (instruction) saklandığı saklayıcıdır.

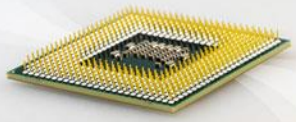
ACC (10 Bit): Accumulator, aritmetik işlem sonuçlarının tutulduğu saklayıcıdır.



İşlem Ünitesi

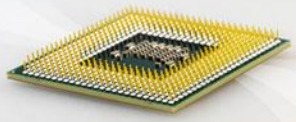
Aritmetik işlemlerin gerçekleştirildiği bölümdür. FB-CPU'da 4 adet aritmetik işlem vardır. Bunlar toplama, çıkartma, çarpma ve bölmedir, gelen operasyon koduna göre işlemleri gerçekleştirip ACC saklayıcısına yazmaktadır.



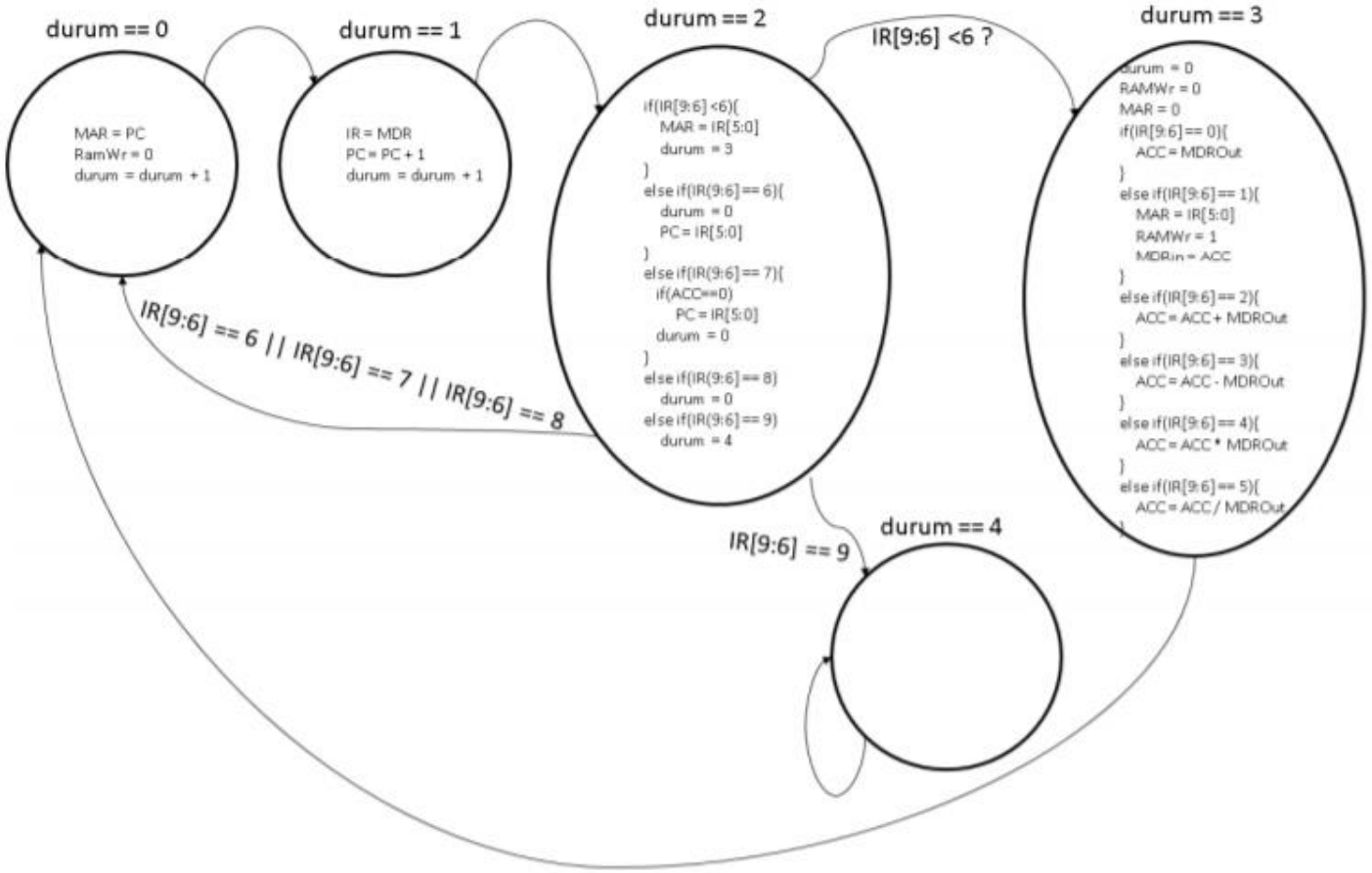


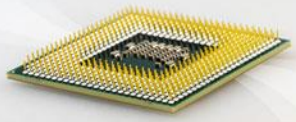
Kontrol Ünitesi

Saklayıcılar, Aritmetik İşlem Ünitesi ve RAM’e verilerin birbirleri arasında transferinden sorumludurlar. İşlemci içi veri akışını yönetir.



FB-CPU’nun durum diyagramı olarak ifade edilmiş hali;





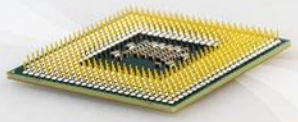
İşlemcinin adım adım yapması gereken işler bir arada gösterilmek tedir.

Örnek Test Yazılımı

FB-CPU için bellekte 50 ve 51 adresteki iki sayının çarpımını 5 2 no'lu adrese kaydeden uygulamayı geliştiriniz. Ancak çarpm a operasyonunu kullanmayınız.

Çarpma işlemi için 50'deki sayıyı 51'deki sayı defa toplayıp 52 no'lu adrese yazınız.

Gerekli değişkenler için istediğiniz adresleri kullanabilirsiniz



Örnek Test Yazılımı

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