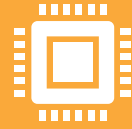


FB CPU YAPIMI

- MEHMET RAUF FÜZÜN
- MERT MERİÇ KARADENİZ
- AHMET BATUHAN YILMAZ
- HAYAT ZEHRA DEMİR

PROJENİN TANIMI



Bu proje kapsamında FB-CPU isminde bir işlemcinin tasarımı yapılacak 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 ile geliştirilmiştir



VON NEUMANN MİMARİSİ

Temel olarak 4 elemanı vardır.

Saklayıcılar

Bellek (RAM)

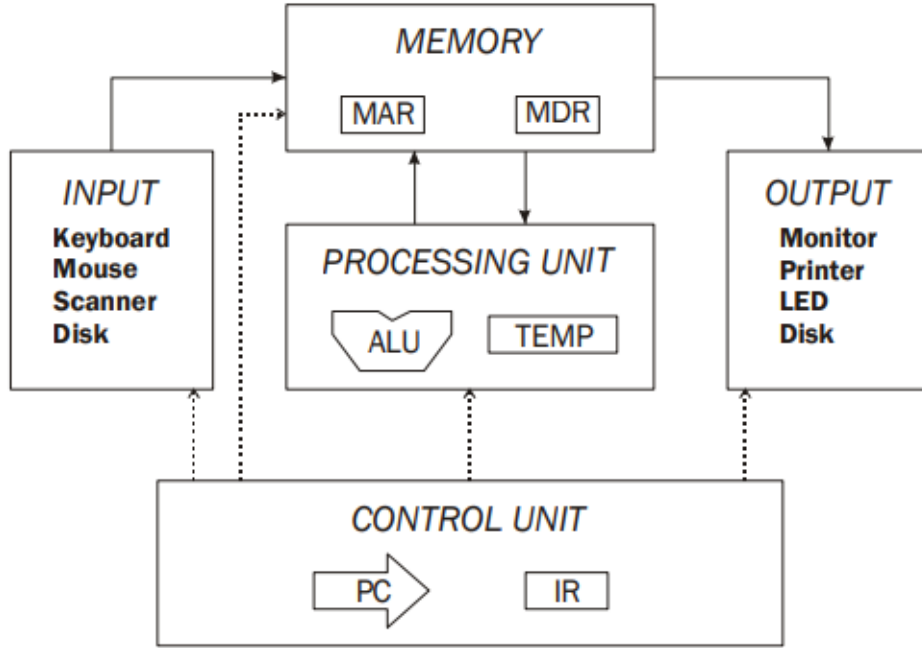
İşlem Ünitesi (ALU)

Kontrol Ünitesi

Mimarinin Elemanları;

SAKLAYICILAR

- Her bir saklayıcı aslında birden çok bir araya gelmiş d tipi saklayıcılardan oluşmuştur
- PC (6 Bit): 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 (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.
- MDRIn (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.
- IR (10 Bit): Instruction Register, RAM'den okunan kodun (instruction) saklandığı saklayıcıdır.
- ACC (10 Bit): Accumulator, aritmetik işlem sonuçlarının tutulduğu saklayıcıdır.



Şekil 2. Von Neumann Mimarisi

Von Neumann mimarisinde, kullanılan ünitelerin görevleri:

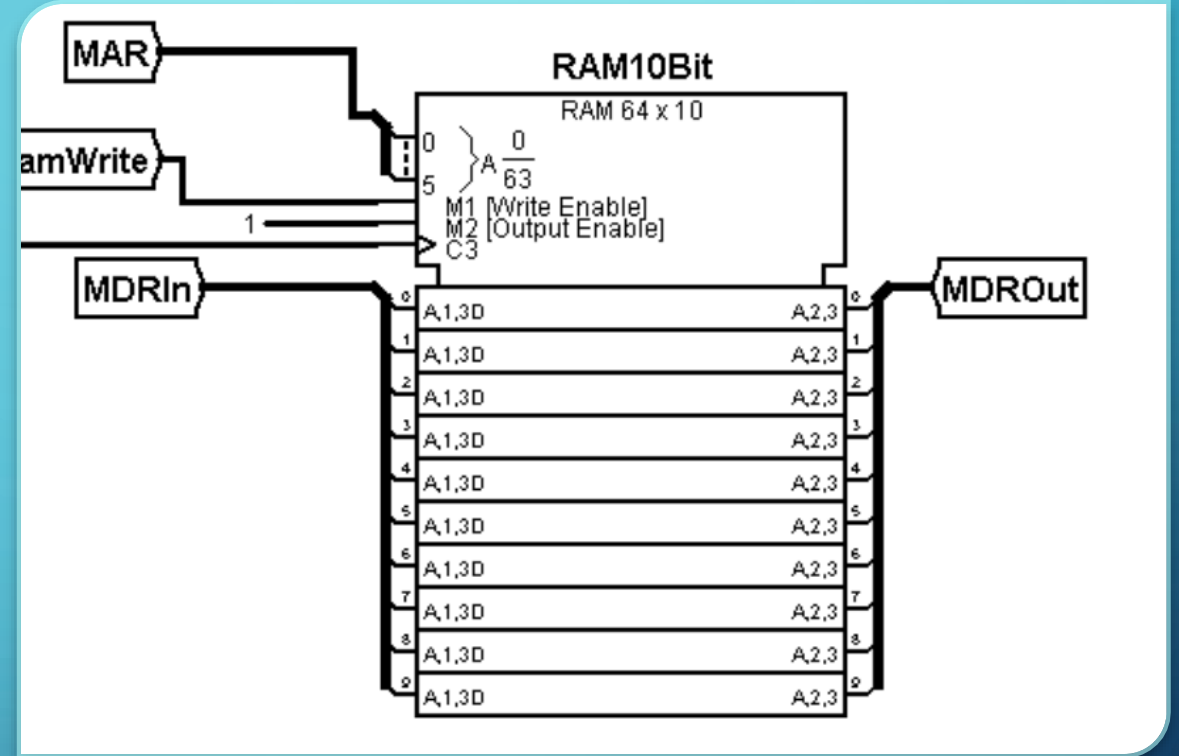
Bellek, operasyon komutlarını ve değişkenleri tutmaktadır.

İşlemci Ünitesi, aritmetik ve mantık işlemlerini yapmaktadır.

Kontrol Ünitesi, komutların çözülmesi için gereklidir.

BELLEK

- $2^k \times m$ saklama alanı vardır
- Adres ;
 - 2^k adet farklı saklama alanı vardır.
- İçerik ;
 - Her bir saklama alanı, m bitlidir.
- Basit operasyonlar:
 - Yükleme ;
 - Bellekteki bir adresten verinin okunup, bir saklayıcıya yazılmasıdır.
 - Kaydetme ;
 - Bir saklayıcıdaki içeriğin, bellekteki bir adrese yazılmasıdır.



• İşlemci 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

- ALU = Aritmetik ve Lojik Ünitesi
- Saklayıcılarda işlem sonuçları tutulabilir.
- ACC : Accumulator, aritmetik işlem sonuçlarının tutulduğu saklayıcıdır.
- İçerisindeki saklayıcılarda geçici değişkenler saklanmaktadır.

KONTROL ÜNİTESİ

PROGRAMIN AKIŞINI YÖNETİR

Kontrol Ünitesi: Saklayıcılar, Aritmetik İşlem Ünitesi ve RAM'e verilerin birbirleri arasında transferinden sorumludurlar. İşlemci içi veri akışını yönetir. Bellekten program counter'ın gösterdiği komutu okur.



Instruction Register (IR) şu anki koşturulan komutun adresini tutmaktadır.

Program Counter (PC) bir sonraki koşturulacak olan komutun adresini tutar.

.

İŞLEMÇİ 10 ADET KOMUTU DESTEKLEMEDİR;

FB-CPU ISA (INSTRUCTION SET ARCHITECTURE).

Komut Adı	Görevi	Operasyon Kodu
LOD ADDR	Yükleme (Load), Bellekteki verilen adresin içerisinde 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

FB-CPU Örnek Komut Binary Gösterimi

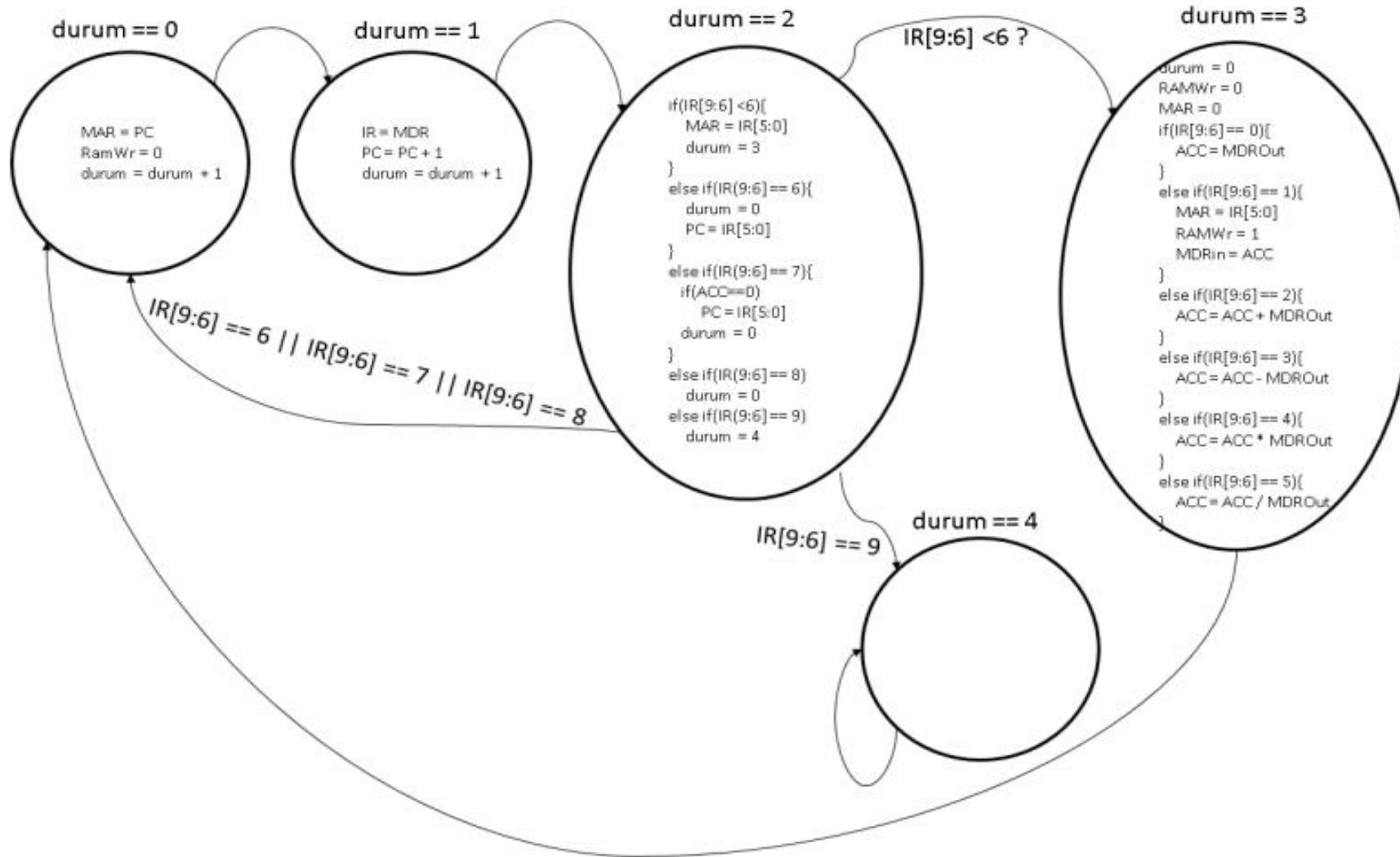
Komut (Instruction) (10 Bit)

1010001111

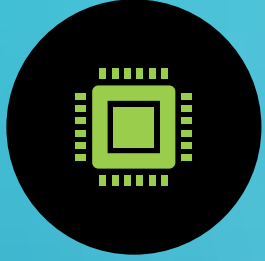
Operasyon
Kodu

Adres veya
Sayı

FB-CPU DURUM MAKİNESİ GÖSTERİMİ



İşlemcinin adım adım yapması gereken işler bir arada göstermektedir.

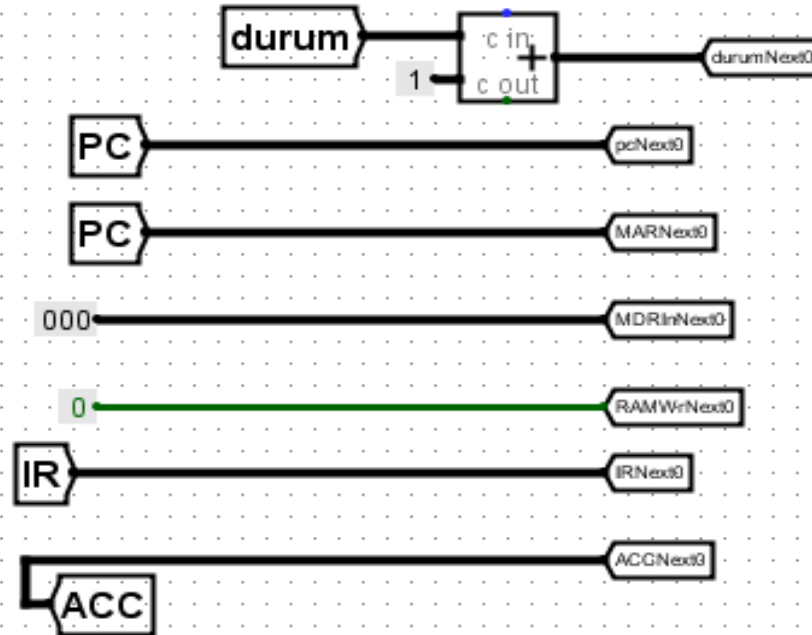


CPU durum makinesinin mimarisi 4 Farklı şekilde oluşmaktadır

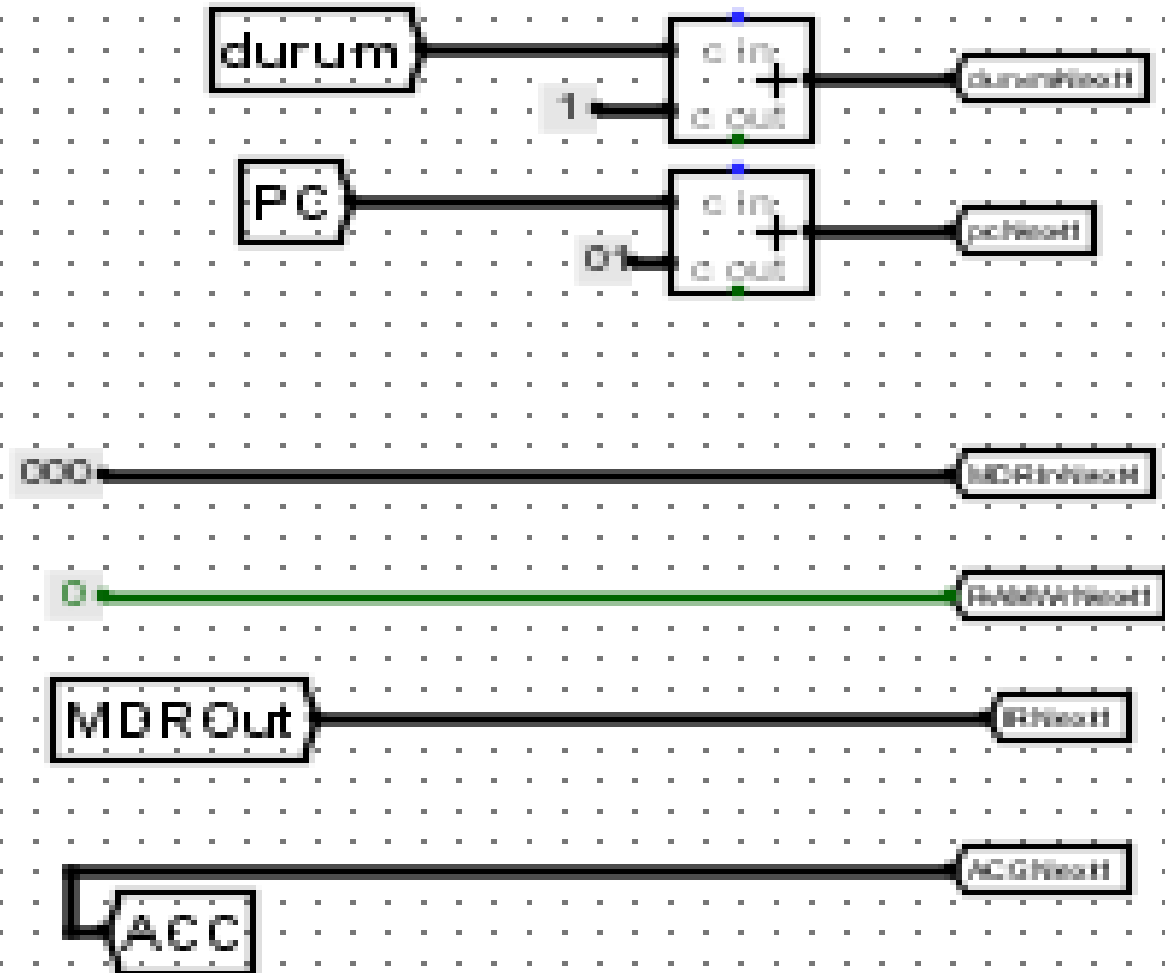


Durum makinasında kurduğumuz mantık yapıları çeşitli devre elemanlarıyla sağlanmaktadır.

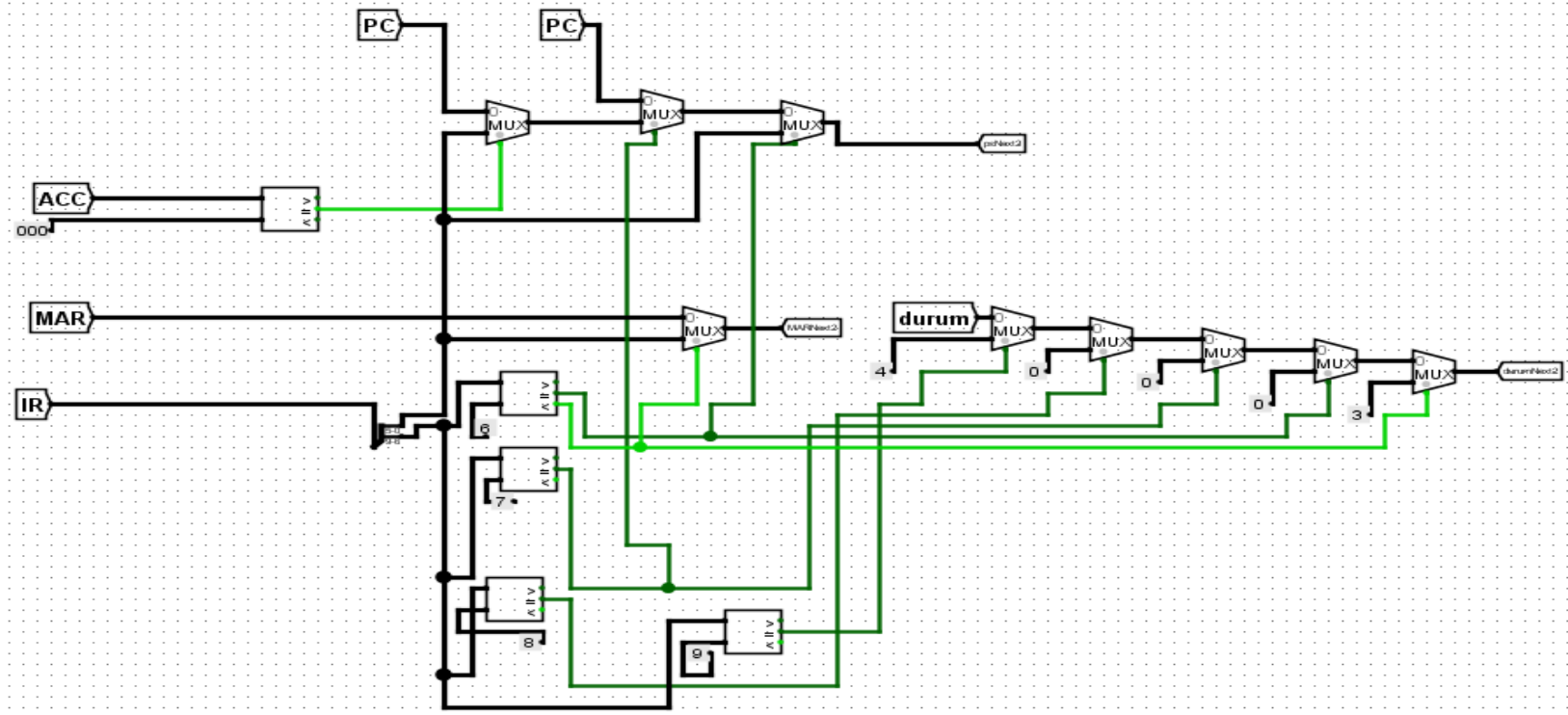
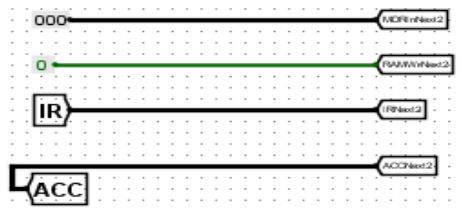
durum0



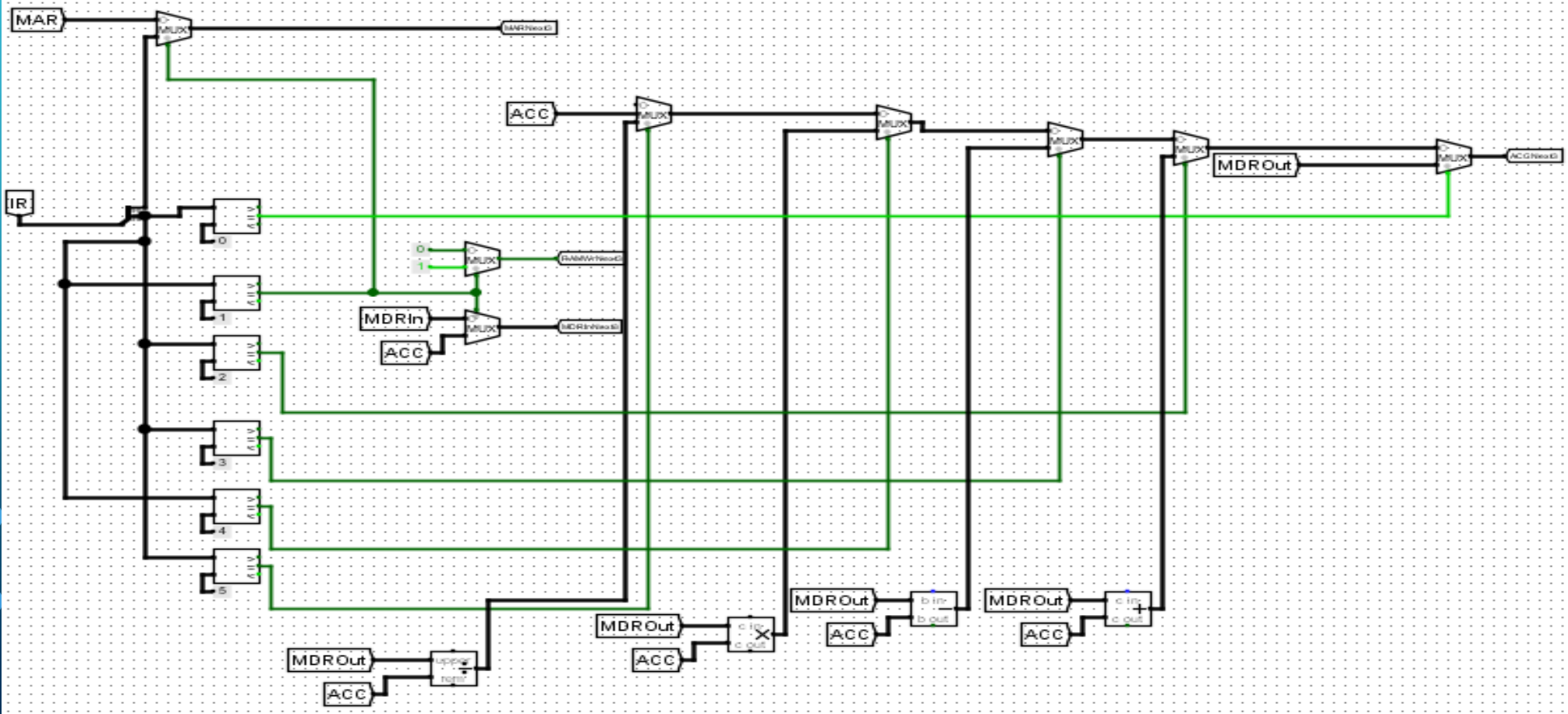
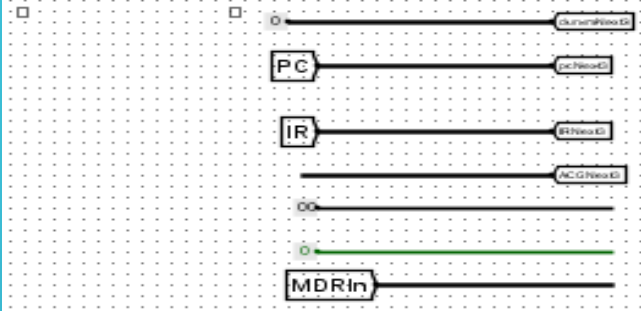
durum1



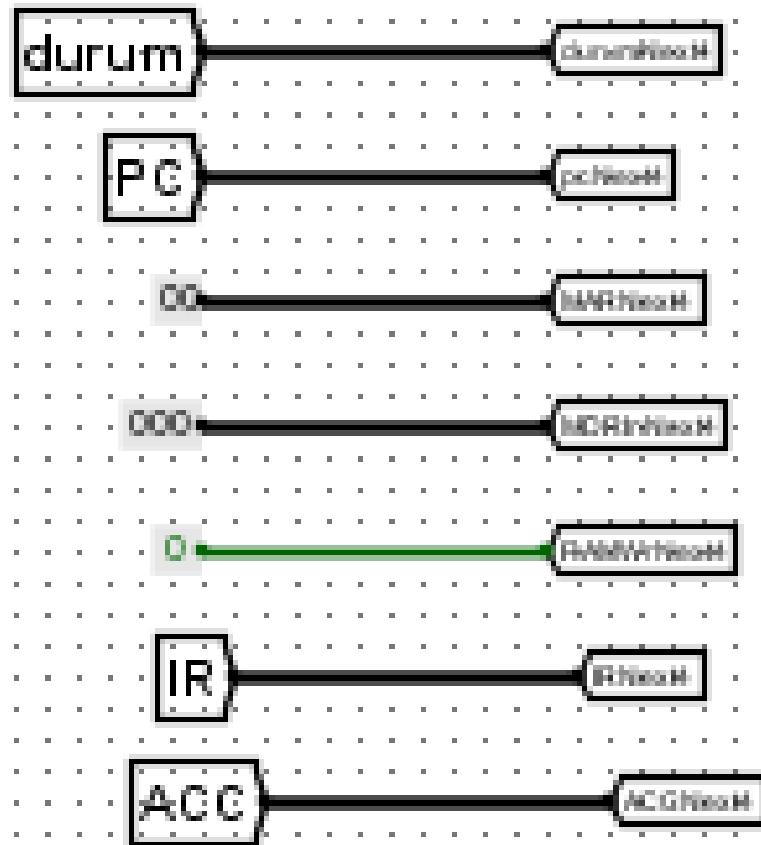
durum2



durum3



durum4



Tasarladığımız işlemcide 10 adet komut çalıştırılabilmekteydi

Bu sebepten dolayı makine dilinde çeşitli yazılımlar işlemci üzerinde yürütülebilmektedir.

Örneğin ;

FB-CPU için bellekte 50 ve 51 adresteki iki sayının çarpımını 52 no'lu adrese kaydeden uygulamaya bakalım

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



SON