



FB-CPU İŞLEMCİ TASARIM SUNUMU



Proje Tanımı;

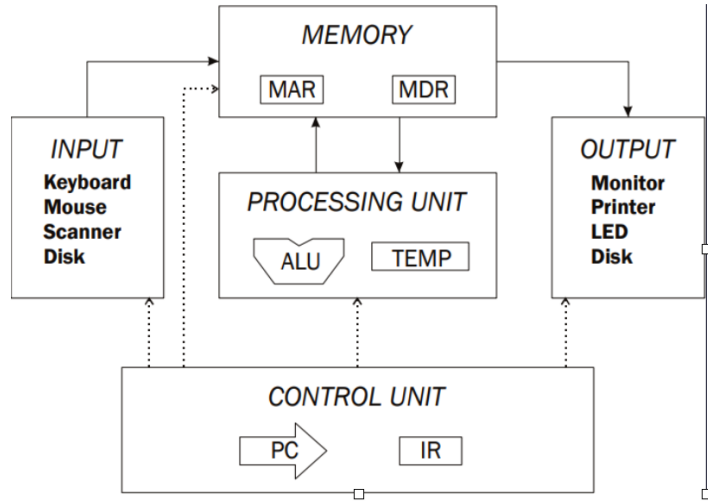
Bu proje kapsamında FB-CPU isminde bir işlemcinin tasarımı ve tasarlanan işlemci üzerinde makine dili ile yazılan çeşitli kod parçacıkları yazılacaktır. Proje sonunda basit bir işlemcideki RAM, Kontrol Ünitesi ve Saklayıcıların bir arada çalışıp, makine dilindeki kod parçacıklarını nasıl yürütebildiği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

KULLANILAN ARAÇLAR;

Proje kapsamında 2 araç kullanıldı.

1) Von Neumann Simulatörü: Von Neumann mimarisi veri ve komutları tek bir birimde bulunduran bilgisayar tasarımıdır. FB-CPU'nun mimarisini görselleştiren ve veri akışının gözlemlenebildiği bir simulatördür.

2) Logisim-Evolution: Dijital mantık devrelerini tasarlamak ve simüle etmek için kullanılan bir eğitim aracıdır. Mantık devreleriyle ilgili temel kavramları öğrenmeyi kolaylaştırmaktadır. İşlemcinin tasarımı bu simulator içerisinde yapılmıştır.

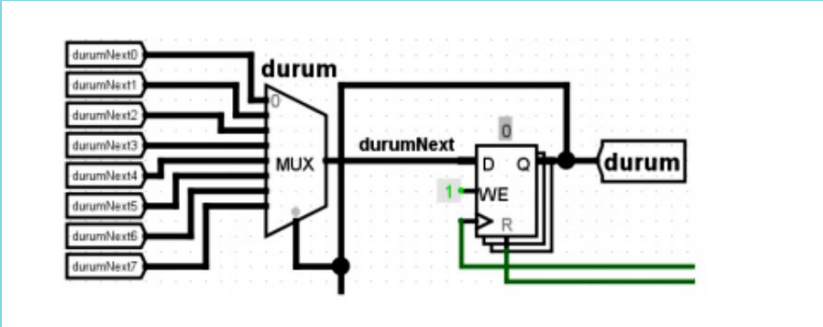


Temel olarak 4 elemanı vardır.

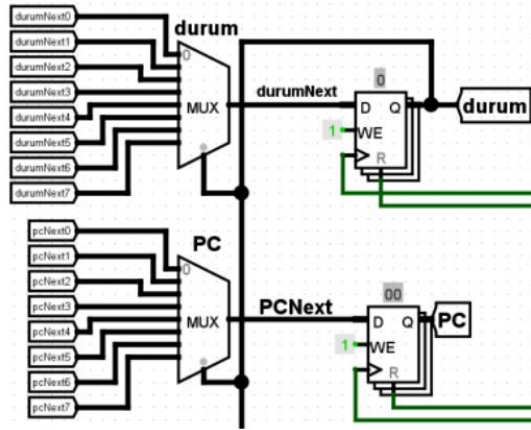
- Saklayıcılar
- Bellek (RAM)
- İşlem Ünitesi (ALU)
- Kontrol Ünitesi

SAKLAYICILAR

Başlangıç tasarımında 8 adet saklayıcı bulunmaktadır. 4 tane d tipi saklayıcı ve RAM'in içerisinde 4 saklayıcı bulunmaktadır.



Durum (3 Bit): FB-CPU durum makinaları yöntemi ile gerçekleştirilecektir. Yani bu işlemci durum ismindeki saklayıcının değerine göre 2^3



- 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.

Bellek (RAM, Random Access Memory):

FB-CPU'nun komutları okuyup, hesaplanan değerleri geri yazacağı bellek şekilde verilmektedir. RAM'e bağlı 4 saklayıcı ve bir clock sinyali bulunmaktadır

Bu yapı komut ve adreslerin tutulduğu bellektir.

İşlem Ünitesi (ALU, Arithmetic Logic Unit): 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.

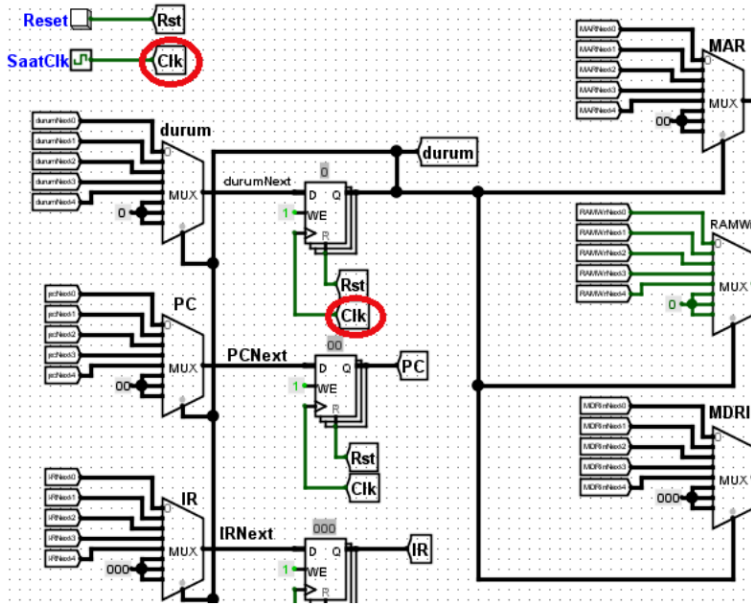
Kontrol Ünitesi: Saklayıcılar, Aritmetik İşlem Ünitesi ve RAM'e verilerin birbirleri arasında transferinden sorumludurlar. İşlemci içi veri akışını yönetir.

- : • Bellekten program counter'ın gösterdiği komutu okur.
- Sistemin geri kalanına, yapılması gereken işlemleri yaptırır.

Bir komut birden çok cycle sürebilir.

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

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



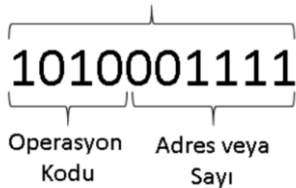
Şekil 3. Tunnel Mekanizması

Tasarım küçük olmaması ve birçok elemanı bir arada bulundurması nedeniyle, bunların birbirlerine arasında gerekli kablo bağlantılarını kurmak oldukça zordur. Bu bağlantılar kablolar ile kurulduğunda ortaya kabloları takip etmesi oldukça güç, çok karmaşık bir yapı ortaya çıkmaktadır. Bunun için Logisim aracının içerisindeki “Tunnel” denen bir mekanizma kullanılmıştır. Tunnel aracı aslında arka planda kablolama işi ile aynı görevi

Tablo 1. 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'in değeri 0 ise, verilen sayı değerini PC'e atar, değilse işlem yapmaz.	0111
NOP	No Operation, hiçbir işlem yapılmaz.	1000
HLT	Uygulama durur	1001

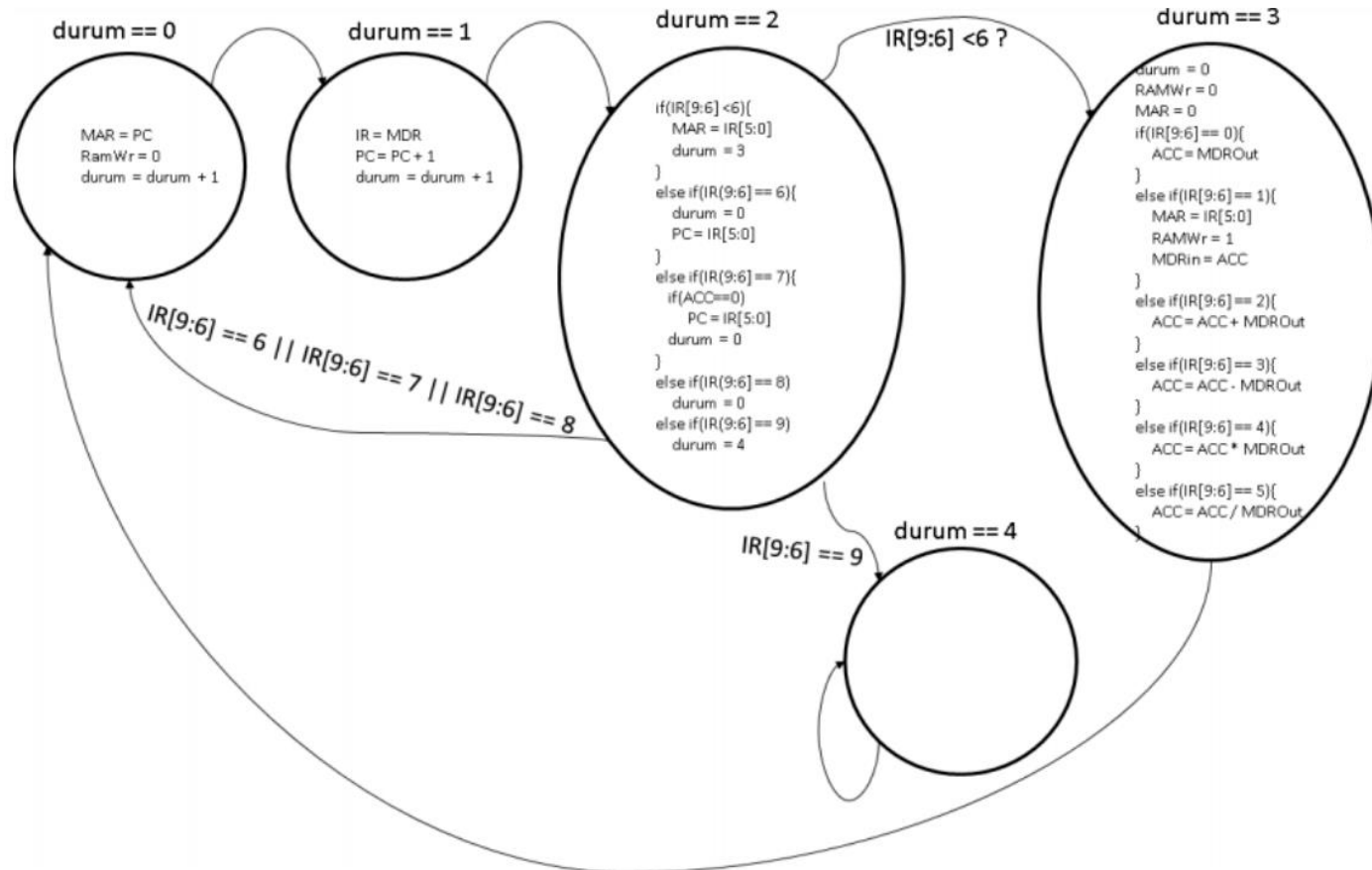
Komut (Instruction) (10 Bit)



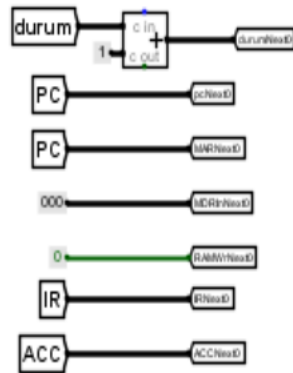
Komutların bellekte okunması ve çözülmesi için durum makinesine 10 bitlik komutunda ilk 4 biti yani [9:6] operasyon kodunu belirtmekte, son 6 biti [5:0] adresi veya sayıyı temsil etmektedir. İşleme sokulacak olan sayı bellekteki bir adresten, komutun içerisinde veya bir saklayıcıdan alınabilir.

FB-CPU tasarımı 10 adet komudu yapabilecek şekilde tasarlanmıştır.

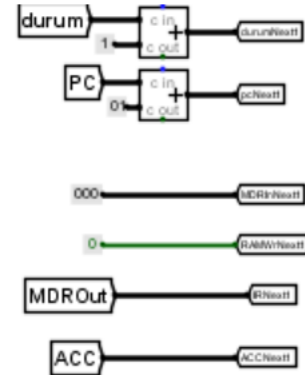
İşlemci belirtilen komutları yerine getirmek için gerekli durum değerleri sağlanmalıdır.



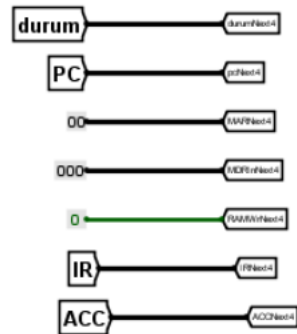
durum0



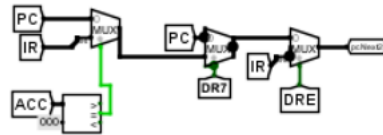
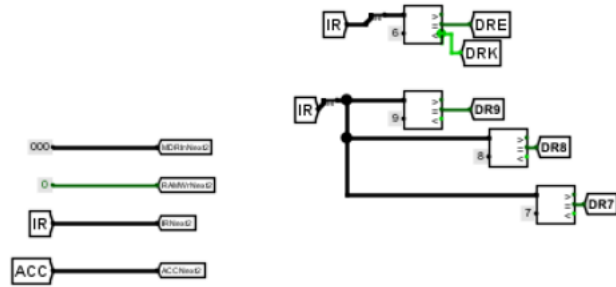
durum1

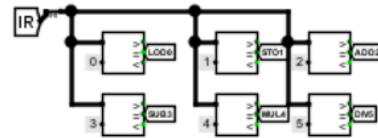
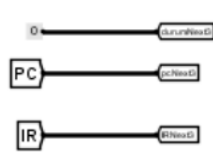


durum4

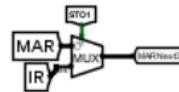
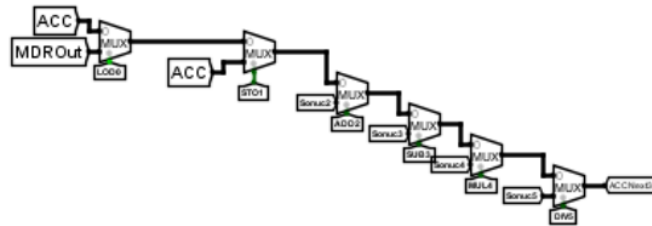


durum2

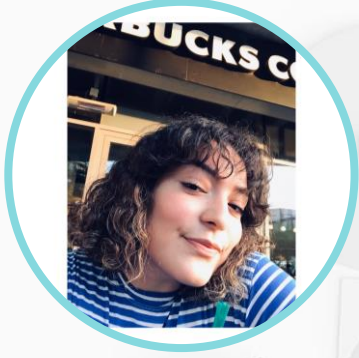




durum3



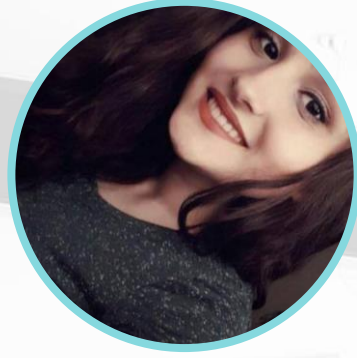
HAZIRLAYANLAR



Deniz Uzun



Özlem Çalı



İrem Kalkanlı



Aysen İpek Çakır



Teşekkür Ederiz