İçindekiler:

Bölüm 4 Intel ailesinin 32 bitlik mikroişlemcileri

Bölüm 5 Pentium mikroişlemcileri

BÖLÜM 1

BİLGİSAYARIN YAPISI

1.1 GİRİŞ

Bilgisayar, fiziki yapısı değişmeden, kendisine verilmiş programa göre çalışan bir araçtır. Daha açık bir tanımlama yapmak istersek, bilgisayar şu dört özelliği yerine getiren araç olarak tanımlanır. Bu dört özellik aşağıda açıklanmıştır:

Saklama yeteneği: Veri ve programlar, bir hafızada saklanabilmelidir. Saklanan veri ve programlar istendiğinde geri alınabilmeli veya yeniden saklanabilmelidir.

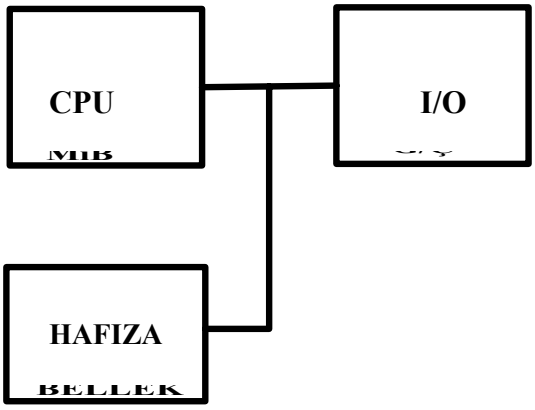
Hesaplama yeteneği: Bilgisayar kendisine verilen veriler üzerinde, yine kendisine verilen programa uygun olarak, aritmetik ve mantıksal işlemler yapabilmelidir.

Karar Verme Yeteneği: İşlemler sonunda veya kendisine verilen verilere bakarak programın akışı ve/veya verilerin değerlendirilmesi konusunda karar verebilmelidir.

Giriş Çıkış Yeteneği: Bilgisayara, dışarıdan veri ve program verilebilmeli ve sonuçlar bilgisayardan dışarı alınabilmelidir. Kısaca, bilgisayar ile çevresi arasında veri iletişimi olmalıdır.

Yukarıdaki özellikleri sağlayan araca veya makineye, bilgisayar demekteyiz. Mikrobilgisayarlarda bilgisayar olmaları nedeniyle yukarıdaki özelliklerin tümünü sağlarlar. Bilgisayar kendisinden beklenen bu dört görevi yapabilmek üzere, üç temel birimden oluşur: Merkezi İşlem Birimi (Central Processing Unit-CPU), Hafıza (Memory) ve Giriş-Çıkış (Input/Output) arabirimi.

Bu bölümde, bilgisayarın yapısı genel boyutları ile ele alınacak ve bilgisayarın çalışma ilkesi sunulacaktır. Bilgisayarı oluşturan CPU, hafıza ve Giriş-Çıkış arabirimi ayrıntılı biçimde daha sonraki bölümlerde anlatılacaktır. Bir bilgisayarın temel elemanları Şekil 1.1'de gösterilmiştir.



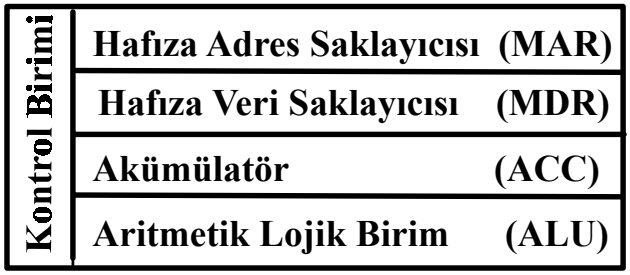
Şekil 1.1:Bilgisayarın üç temel birimi

CPU, bilgisayardan beklenen hesap yapma ve karar verme işlemlerinin gerçekleştiği birimdir. Bu birim aynı zamanda, tüm bilgisayarın çalışmasını da yönetir. Bu nedenle çoğu kez CPU' ya bilgisayarın beyni olarak ta bakılmaktadır. Hafıza, veri ve programların saklanması amacıyla kullanılmaktadır. Bazı bilgisayarlarda, veri ve program aynı hafıza içinde saklanırken, bazılarında veriler için bir hafıza, program için başka bir hafıza bulunur.

Giriş-Çıkış arabirimi, bilgisayarın, çevre birimleriyle bağlantısını sağlar. Bir başka deyişle bilgisayar ile çevre birimler arasında veri iletişimi için port görevini yürütür.

Bilgisayarın, bu üç birimi arasındaki veri ve program akışını sağlamak üzere, çok sayıda bağlantıdan oluşan yollar bulunmaktadır. Bu yollar sırasıyla Veri Yolu, Adres yolu ve Kontrol Yoludur.

Ayrıntıya inmeden CPU içine bakıldığında beş alt birim görülür. Bu alt birimler Şekil 1.2 den görüldüğü gibi, sırasıyla, Hafıza Adres Saklayıcısı (MAR), Hafıza Veri Saklayıcısı (MDR), Akümülatör (ACC), Aritmetik Lojik Birim (ALU) ve Kontrol birimidir (CON).



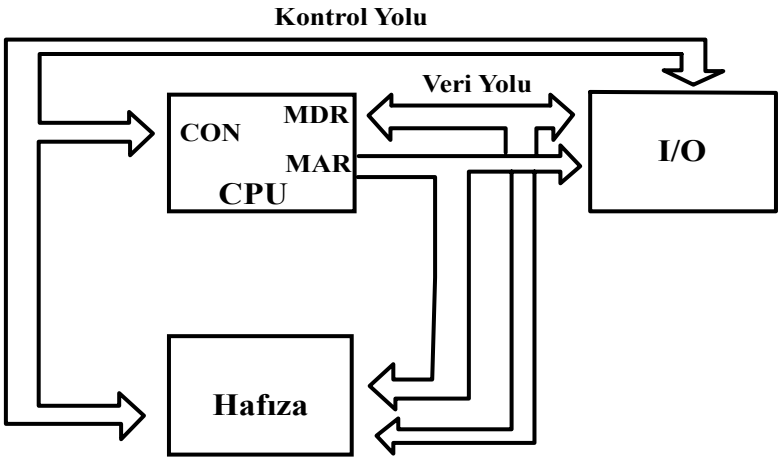
Şekil 1.2: Merkezi İşlem Biriminin içyapısı

Hafıza Veri Saklayıcısı, CPU' dan hafızaya veya giriş-çıkış arabirimine giden veya buralardan CPU ya gelen verilerin alınıp verildiği bir port olarak görev yapar. Bu nedenle veri yoluna bağlıdır.

MDR üzerinden CPU ya gelen veriler Akümülatöre alınır. Bu veri üzerinden yapılacak işlem ise ALU tarafından yürütülür.

Hafıza Adres Saklayıcısı, CPU dan hafızaya veya giriş çıkış arabirimine giden veya buralardan gelen verilerin gideceği veya geldiği yerin adresini belirtmekle görevlidir. MAR adres yoluna bağlıdır.

Kontrol birimi, tüm bilgisayarın yönetimi ile ilgili bilgileri toplamak ve üretmekle görevlidir. Veri yolunda verilerin akış yönünün belirlenmesi, adreslerin ve verilerin zamanında yollara yerleştirilmesi işlemleri kontrol biriminin görevleri arasındadır. Üç temel bilgisayar elemanının veri yolu,adres yolu ve kontrol yolu üzerinden bağlantıları Şekil 1.3'de verilmiştir.



Şekil 1.3: Bilgisayarın üç temel birimi arasındaki yollar

Hafıza, kısaca, birbirinin aynı ve çok sayıda hafıza birimlerinin üst üste yığılmasıyla oluşmaktadır. Hafıza birimlerine Hafıza Gözü denilir. Her hafıza gözünde ise göze denilen hafıza birimleri vardır. Bir hafıza içinde, tüm hafıza gözlerinde aynı sayıda hafıza gözesi bulunur. Çok sayıdaki hafıza gözlerinin her bir gözesinin CPU ya doğrudan bağlanması düşünülemez. Bunun yerine, aynı sırada aynı sırada bulunan gözeler birbirine bağlanmış ve bir sıra göze için, veri yolunda bir hat atanmıştır. Böylece, veri yolundaki hat sayısı bir hafıza gözündeki göze sayısı kadardır. Aynı hatlara bağlı hafıza gözelerinden istenen birine erişebilmek, ancak o göze ilişkin adresin belirlenmesi ile sağlanır. Bu adres belirtmesi işlemi de adres yolu üzerinden gerçeklenmektedir. Bilgisayar yapısında, bir anda tek bir hafıza gözüne erişilebilmesi yöntem olarak benimsenmiştir.

Giriş-çıkış elemanları da birer hafıza gözü gibi düşünülebilir. Aralarındaki fark, bu gözlerin birer ucunun bilgisayarın dış dünyasına açılmasıdır.

8 bitlik mikroişlemcilerin çoğunda veri yolu 8, adres yolu 16 hattan oluşmaktadır. Kontrol yolundaki hatların sayısı, CPU nun yapısına göre değişmektedir.

Daha öncede belirtildiği gibi, bilgisayarı oluşturan üç temel birim ileride geniş biçimde ele alınacaktır.

1.2 HAFIZA (MEMORY)

Bilgisayarın çalışmasına yön verecek programın ve programında üzerinde çalışacağı verileri saklamak için, hafıza kullanılmaktadır. Bu bölümde, hafızanın yapısı, türleri ve hafıza tasarımı konuları ele alınacaktır..

Hafıza teknolojisi de, bilgisayar teknolojisindeki gelişmelere paralel olarak zaman içinde gelişmiştir. Bilgisayarın ilk dönemlerinde, mekanik çarklar ve delikli kartlar

hafıza olarak kullanılmıştır. Elektronik bilgisayarlarda, önceleri röleler ve daha sonraları çekirdek hafızalar kullanılmıştır. Yakın zamanda ise kullanılan hafızalar yarı iletken teknolojisine dayanmaktadır.

Çekirdek hafızalar, simit biçimindeki manyetik halkaların içinden geçirilen akımla kutuplanması ve daha sonra, yardımcı sargılarla,kutuplanma yönünün öğrenilmesi ilkesine göre çalışmaktadır. 2-3 milimetre çaplı çekirdek hafıza elemanlarından onbinlercesini, 10x10 cm dolaylarındaki bir yüzeye yerleştirme,yüksek el becerisi gerektirmekteydi. Bu nedenle çekirdek hafızalar çok yüksek fiyatlarla satılmaktaydı.

Bir çekirdek hafızanın fiyatı, aynı boyda yarı iletken bir hafızanın fiyatının yaklaşık 2000 katı dolayındadır. Bu önemli fiyat farkı çekirdek hafızaların terk edilmesine neden olmuştur. Çekirdek hafızanın, önemsenecek iyi yönü ise hafızaya yazılmış bilgilerin saklanması için enerji gerektirmemesidir. Bu nedenle bilgisayar kapatılmış olsa bile bilgiler saklı kalabilir.

Yarıiletken teknolojisindeki gelişmelerin sonucu olarak çekirdek hafızalar, yerlerini yarı iletken hafızalara terk etmişlerdir. Teknolojideki gelişmelere paralel olarak,yarıiletken hafızaların kapasiteleri artmakta ve fiyatları düşmektedir. Yarı iletken hafızalar, hafızaya yazılmış olan bilgileri tutabilmek için enerji gerektirirler. Bu nedenle hafızayı besleyen gerilimin kesilmesi, hafızadaki bilgilerin kaybolmasına neden olur. Hafızadaki bilgileri sürekli tutabilmek için çeşitli yöntemler geliştirilmiştir. Yarıiletken hafızalar, bilgileri tutabilme açısından iki guruba ayrılırlar: Sadece okunabilen ve içine bilgi yazılamayan hafızalar (Salt Oku Hafızalar), yazılıp okunabilen hafızalar (Oku/Yaz Hafızalar).

Disk, manyetik bant gibi birimler de birer hafıza elemanı gibi düşünülebilirler. Ancak, bu hafızalar, bilgisayar yapısı içinde CPU tarafından doğrudan adreslenebilen ve okunup yazılabilen hafızalar değildir. Bu nedenle, bu tür hafızalara yan hafızalar diyeceğiz.

1.2.1 SALT OKU HAFIZALAR (READ ONLY MEMORY)

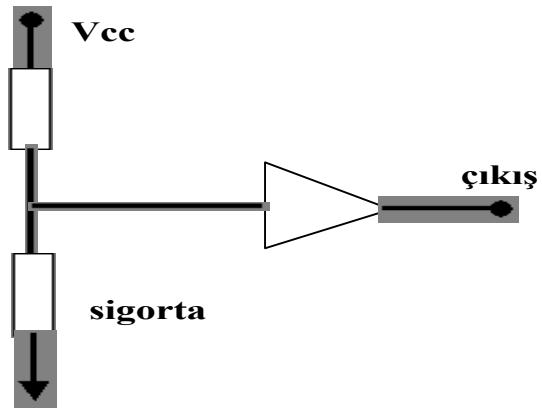
Bilgisayarlarda, sürekli kalması istenen bilgilerin saklanması için kullanılan hafıza türleridir. Özel amaçlı ve sabit programla çalışan bilgisayarların programları ile bilgisayar açıldığında,kullanıcıya hizmet verecek,yol verme programları, bu tür hafızalara yazılmalıdır. Salt oku hafızanın adına bakarak, şu soru akla gelebilir: “Bu hafızalara bilgi yazılamıyor ise bu hafıza içindeki bilgiler nasıl yazıldı ?” salt oku hafıza içine bilgiler özel yöntemlerle ve özel araçlarla yada üretim sırasında yazılabilir. Salt oku hafıza içine bilgi yazmaya, hafızanın programlanması denir. CPU nun bu tür hafızalara doğrudan bilgi yazması olanağı yoktur. Salt oku hafızalar, programlanabilme niteliklerine göre dört guruba ayrılırlar. Bu guruplara ilişkin özellikler şöyledir:

a) Salt Oku Hafıza (ROM)

Bu hafıza türüne bilgi yazılması, hafızanın üretimi sırasında gerçekleşmektedir. Yarıiletken malzemeden hafıza yapılırken kullanılan maskeler, hafızanın içermesi gereken bilgileri oluşturacak biçimde hazırlanır. Sonuçta, hafıza istenen bilgilerle üretilmiş olur. ROM türü hafızaların, üretimi için gerekli maskelerin hazırlanma maliyeti oldukça yüksektir. Bu nedenle aynı programı içeren hafızalardan belli sayının (500 veya daha fazla) üzerinde üretildiği zaman ucuz çözüm elde edilmektedir. ROM ların sadece üretim sırasında programlanabilmeleri ve programlarının daha sonra değiştirilememesi,önemli kısıtlama olarak görülebilir. Ancak çok sayıda ve aynı programla yüklü hafıza gerektiren uygulamalar için,sözgelimi, tek bir program üzerinde çalışan bilgisayarlar için, en ucuz çözüm ROM hafızalarla sağlanmaktadır.

b) Programlanabilir Salt Okunur Hafıza (PROM)

Üretildikleri an bütün gözeleri (en küçük hafıza birimi) 0 veya 1 ile yüklü hafızalardır. Her hafıza gözesi içinde bir sigorta bulunmaktadır. Bu sigortalar, özel bir yöntem ve aygıt aracılığıyla attırılabilir. Bir gözenin sigortasının atmış olması, o gözenin lojik konumunun değişmesi demektir. Sözgelimi tüm gözeleri 0 olan bir hafızanın istenen gözeleri 1 konumuna getirilerek programlanmış olur. Ancak 1 konumuna getirilmiş gözenin tekrar 0 konumuna dönme şansı yoktur. Bir başka deyişle atmış bir sigortanın yenilenmesi olanağı yoktur.



Şekil 1.4: Bir PROM gözesi benzetimi

Bir PROM gözesi şekil 1.4'de gösterilmiştir. PROM 'lar üretildikleri ilk yıllarda, programlanabilme özellikleri nedeniyle özellikleri nedeniyle üstün sayılmışlardır. Ancak daha sonra üretilmeye başlayan silinebilen PROM lar yani EPROM lar PROM ları gözden düşürmüştür. PROM lar, bipolar teknolojisi ile üretildikleri için, hızlı çalışmaktadırlar. Bu nedenle, günümüzde hafıza olarak değil,daha çok adres çözücü olarak kullanılmaktadırlar.

c) Silinebilir Programlanabilir Salt Oku Hafıza

EPROM: EPROM'lar üretildiklerinde tüm hafıza gözeleri 1 konumundadır. 1 konumunda olan gözelerden istenenler, özel yöntemler ve aygıtlarla 0 konumuna geçirilebilir. Mor ötesi ışığın yarıiletken üzerine düşürülmesi ve belli bir süre tutulması sonunda tüm gözeler 1 konumuna gelirler. Bu amaçla EPROM un gövdesi üzerinde bir pencere bulunmaktadır. Aydınlatmanın tek bir göze için yapılamamasından dolayı, tek bir göz veya gözenin 1 konumuna getirilmesi olanağı yoktur. Ancak tüm hafıza silinebilir.

EPROMun silinmesi ve programlanma sayısı konusunda verilmiş bir sınırlama olmamakla beraber, uygulamada belli bir sayıdan fazla programlamanın olanaksız olduğu gözlenmiştir. EPROMlar silinebilme ve tekrar tekrar programlanabilme özellikleri nedeniyle, araştırma ve geliştirme süresinde çok kullanışlı olabilmektedir. Fiyat olarak ROMlardan daha pahalıdır.

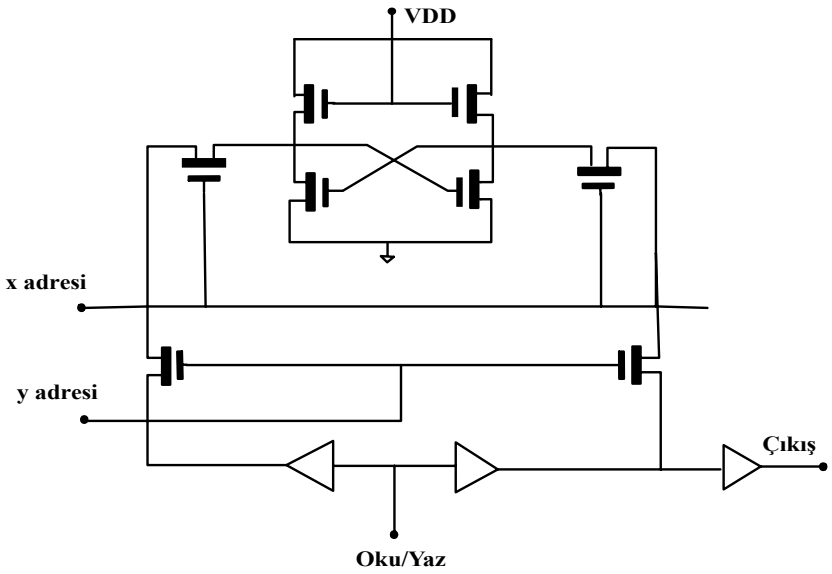
EEPROM: Silinebilir ve programlanabilir hafızaların en gelişmiş olanı elektriksel olarak silinebilen salt oku hafızalardır. Bu hafızalarda, hafıza gözelerine istenen bir değer yazılabilir ve yazılan bu bilgi yeni bir yazmaya kadar kalır. Hafıza gözesine yazılan bu bilgi 0 ve 1 lerden oluşur. Başka bir deyişle, bir göze istendiği an 0 istendiği an 1 konumuna getirilebilir. EEPROM lara veri yazılmasında da özel aygıt ve yöntemlerden yararlanılır. Gerçekte EEPROM un silinmesi, EPROM un silinmesi ile aynı anlamı taşımamaktadır. EEPROM un silinmesinden tüm gözlerin 0 olması anlaşılıyor ise, yapılacak iş tüm hafıza gözlerini 0 ile doldurmaktır. Yada EPROM a benzer bir çözüm isteniyor ise tüm gözler 1 ile doldurulmalıdır. EEPROM ların fiyatları düşmekte ve giderek EPROM ların yerini almaktadır.

1.2.2 OKU/YAZ HAFIZA (RANDOM ACCESS MEMORY - RAM)

Bilgisayar içindeki kullanıcı programının yazılacağı veya verilerin yazılacağı hafıza türü Oku/Yaz hafızadır. Bu tür hafızalar kullandıkları teknik nedeniyle ikiye ayrılırlar. Bunlar statik ve dinamik Oku/Yaz hafızalardır.

a) Statik Oku/Yaz Hafıza

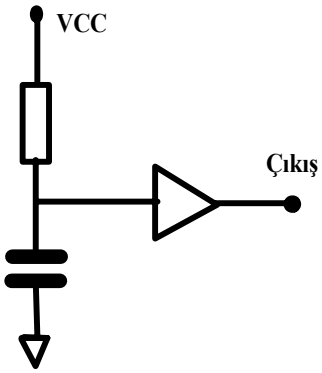
Statik oku/yaz hafızanın her bir gözesi aslında bir flip-flop tur. Bilindiği gibi iki konumlu olan flip-flop girişine uygulanan tetikleme işareti ile 0 veya 1 konumuna getirilebilir. Yeni bir tetikleme gelene kadar eski konumunda kalır. Bir statik hafıza gözesi Şekil 1.5 de verilmiştir. Statik oku/yaz hafızaların harcadıkları güç giderek azalmaktadır. Bunun sonucu olarak içinde pili olan hafızalar üretilebilmektedir. Bu sayede içindeki bilgileri senelerce saklayabilen hafızalar üretilebilir hale gelmiştir. Bir başka kullanım biçimi de, içeriğini saklaması gereken oku/yaz hafızanın yanına bir pilin yerleştirilmesidir.



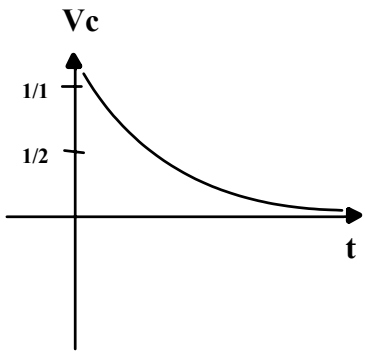
Şekil 1.5: Bir MOS Oku/Yaz hafıza gözesi örneği

b) Dinamik Oku/yaz Hafıza

Dinamik hafıza gözesi temelde bir kapasite ve bir Kuvvetlendiriciden oluşur. Kuvvetlendirici genelde tek bir transistordan oluşur, Şekil 1.6 bir gözeyi 1 konumuna getirmek için kapasitenin doldurulması gerekir. Durum şekil 1.7 de gösterilmiştir. Eğer kapasitenin boşalma karakteristiği bilinir ve kapasite uçlarındaki gerilim, yarısına inmeden okunur ise lojik 1 olarak algılanabilir. Eğer bir gözenin, 1 olduğu algılanır ise kapasite yeniden doldurulur, 0 olduğu algılanır ise doldurulmaz. Böylece hafızadaki bilgilerin saklanması sağlanmış olur. Bu işleme *Dinamik hafızanın tazelenmesi* denir. Üretilen dinamik hafızaların tazelenmesi için hafıza gözelerinin okunması yeterlidir. Bu yalancı okuma işlemi için dinamik hafıza tazeleme devresi kullanılır. Dinamik hafızalar, beraberinde bir tazeleme devresini gerekli kılmakla beraber, çok ucuza üretilebildikleri için geniş kullanım alanı bulmaktadırlar.



Şekil 1.6 Dinamik hafıza gözesi



Şekil 1.7 Kapasite gerilimi

1.3 MERKEZİ İŞLEM BİRİMİ (CPU)

Merkezi işlem birimi (CPU) bilgisayarın temel birimi olarak kabul edilebilir. Giriş bölümünde değinildiği gibi bilgisayar aritmetik ve lojik işlemler yapabilmeli ve ortaya çıkan sonuçlara göre karar verebilmelidir. Bilgisayardan beklenen bu görevler CPU tarafından yerine getirilir. CPU bu görevlerinin dışında, tüm bilgisayarın çalışmasını da düzenler. Bu işlevleri nedeniyle bilgisayarın beyni olarak ta bakılmalıdır.

CPU nun hızı ve yetenekleri, doğrudan bilgisayarın hız ve yeteneğini belirlediği için, CPU nun geliştirilmesi üzerine olan çalışmalar her zaman önemli olmuştur. CPU nun hız ve yeteneklerinin ölçütleri kısaca şöyle açıklanabilir:

HIZ: CPU nun belli zaman içinde yapacağı işlem sayısıdır. Bu sayıya iş yükü de denilmektedir. CPU nun hızını arttırmak için yarıiletken teknolojisinde önemli gelişmeler gözlenmektedir. CPU nun hızını arttırmak için CPU nun donanımı üzerinde de çalışmalar sürmektedir. CPU nun hızı, CPU nun çalışmasını senkronlayan saat frekansına bağlıdır. Bu nedenle, saat frekansının yükselmesi CPU nun hızının artmasına karşılık düşer. Ancak bu sonuç, aynı yapıdaki CPU ler için geçerlidir. Bir başka deyişle, farklı yapıda olan CPU ları içinde saat frekansı yüksek olanın hızı, alçak olandan daha fazladır denemez. Çünkü saat frekansı düşük olan CPU yapısı nedeniyle daha çok işlemi aynı zamanda sonuçlandırabilir.

Günümüzde de, aynı CPU nun farklı saat frekanslarında çalışan türleri üretilmektedir.

Sözcük Uzunluğu:CPU nun bir anda işleyeceği veri uzunluğu, CPU nun gücünü gösterir. Örneğin 16 bitlik iki sayının toplamını, 16 bit işleyebilen bir CPU i bir adımda yaparken, 8 bitlik bir CPU aynı işlemi yaklaşık 4 veya daha fazla adımda yerine getirir. Her adımı atış hızları aynı bile olsa bu örnekte görüldüğü gibi, 16 bitlik CPU 8 bitlik CPU ya oranla yaklaşık 4 kat daha hızlı çalışacaktır. Bu hız ise sadece, boylarından kaynaklanmaktadır.

Günümüzde, 8 bitlik, 16 bitlik, 32 bitlik ve 64 bitlik mikroişlemciler üretilmektedir.

Komut Kümesi: bir CPU nun yeteneklerinin ölçülmesinde, komut kümesinin çeşitliliği ve komutlarının güçlülüğü de önemli bir etkendir. Komut kümesinin zengin olması, programcıya esneklik ve kolaylık sağlayacağı için, CPU yu güçlü kılar. Ancak bu gerekçeyle, komutların sayısının gereksiz yere arttırıldığı da gözlenmiştir. Nitekim yapılan araştırmalar, komutların belli bir kesiminin sıkça kullanıldığını, bazı komutların ise çok az kullanıldığını göstermiştir. Bu nedenle son yıllarda, komut sayısının arttırılması yerine , komutların yeteneklerinin arttırılması görüşü ağırlık kazanmaktadır. Bu amaçla son yıllarda, Komut Sayısı Azaltılmış (RİSC) CPU lar geliştirilmektedir.

Adresleme Yeteneği: Bir CPU nun en önemli özelliklerinden biri de komut içinde kullanılan adresleme yöntemleridir. Kullanılan adresleme yöntemlerinin zenginliği ve güçlülüğü, bu arada düzgünlüğü de bir CPU için önemli bir ölçüttür.

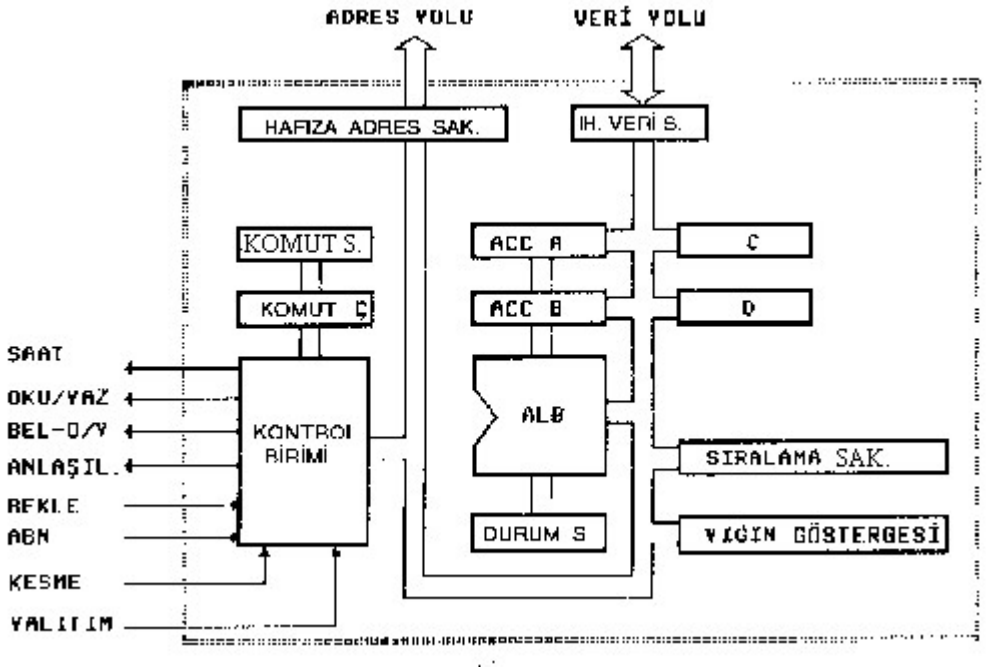
Adresleme Kapasitesi: CPU nun bir başka yeteneği ise, doğrudan adresleyebileceği hafıza boyudur. Adresleme yeteneği, bilgisayarda kullanılacak programın ve saklanacak verilerin boyunu belirlediği için önem taşımaktadır. 8 bitlik mikroişlemcilerde, genellikle adresleme yeteneği 64 K dır. 16 bitlik mikroişlemcilerde bu boy 16 M ya ulaşmıştır.
Her geçen gün mikroişlemcilerin yetenekleri artmaktadır. Bunun için yeni yapıda mikroişlemciler üretilmekte ve eskilerinde daha hızlı çalışanları piyasaya sürülmektedir. Yeni ve gelişmiş mikroişlemcilerin piyasaya sürülmesi, yetenekleri daha kısıtlı bazı mikroişlemcilerin sonu olmamaktadır. Örneğin I8085 ve MC6800 gibi 8 bitlik mikroişlemciler artık standart birer eleman haline gelmiş ve fiyatları daha çok düşmüştür. Mikroişlemci seçiminde akılcı yol, yetenek ve fiyat açısından amaca en uygun olan mikroişlemciyi belirlemektir.

1.3.1 MERKEZİ İŞLEM BİRİMİNİN YAPISI

Merkezi işlem birimi beş temel birimden (MAR, MDR, ACC, ALU ve Kontrol birimi) oluşur. Ancak, daha ayrıntıya inildiğinde, CPU içinde daha fazla birimin olduğu görülür. CPU içinde bulunan alt birimlerin adları aşağıda sunulmuştur:

Hafıza Veri Saklayıcısı	MDR
Hafıza Adres Saklayıcısı	MAR
Aritmetik Lojik Birim	ALU
Akümülatör	ACC
Durum Saklayıcısı	FR
Yardımcı Saklayıcılar	REGS
Program Sayacı	PC
Komut Saklayıcısı	IR
Komut Çözücü	ID
Yığın Göstergesi	SP
Sıralama Saklayıcısı	IR
Kontrol birimi	CON

CPU nun içini ve alt birimler arası ilişkileri Şekil 1.8 de gösterilmiştir. Şekilden görüldüğü gibi CPU içindeki alt birimler arasındaki bilgi alışverişi için İç Yol adını vereceğimiz yol bulunmaktadır.



Şekil 1.8 CPU nun iç yapısı

a) Hafıza Veri Saklayıcısı (MDR)

CPU dan hafızaya veya I/O arabirimine giden veya bu birimden CPU ya gelen verinin yazıldığı yerdir. MDR, CPU içinde iç yol üzerinde ACC, yardımcı saklayıcılar ve komut saklayıcısına bağlıdır. MDR, CPU dışında veri yoluna bağlıdır. MDR nün veri yolu ile bağlantısı üç konumlu yapı yapısındadır. Bu nedenle, gerektiğinde, MDR kendisini veri yolundan yalıtabilir. MDR boyunun, CPU nun sözcük uzunluğuna eşit olması beklenir. Ancak, MDR uzunluğu, CPU sözcük uzunluğunun yarısına eşit olan mikroişlemciler vardır. MC6800 ve I8085 te MDR 8 bittir. Dolayısıyla MDR in boyu CPU sözcük uzunluğuna eşittir. I8088 16 bit sözcük işleyebilmesine karşın MDR 8 bit olarak düzenlenmiştir. Bu nedenle, aktarılacak veriler iki parça olarak MDR de yer alır.

b)Hafıza Adres Saklayıcısı (MAR)

CPU ile hafıza arasında gidip gelen verilerin hafızada hangi göze yazılacağı veya hangi gözden geldiği bu kütüğe yazılan adres ile belirlenir. Giriş-çıkış arabirimlerinin adreslenmesinde de MAR den yararlanılır. MAR iç yol üzerinde PC, SP ve IR ne bağlıdır. MAR çıkışındaki üç konumlu kapılar üzerinden veri yoluna bağlıdır. Bu bağlantının üç konumlu kapılarla yapılması, gerektiğinde, MAR ın adres yolundan yalıtılmasını sağlar.

MAR nün boyu, CPU nun adresleme yeteneği ile belirlenir. Sözgelimi, 8 bitlik mikroişlemcilerde, genellikle, adres yolu 16 bitliktir. Dolayısıyla bu mikroişlemciler için MAR 16 bit uzunluğundadır. 16 ve 32 bit mikroişlemcilerde MAR ın boyu 24 bite kadar çıkmaktadır.

Bazı mikroişlemcilerde, MAR ve MDR ortak hatları kullanırlar. Örneğin I8085 de, 16 hattın tamamı MAR ne bağlı olarak adres yolunu oluştururken, bu hatlardan ilk sekizi , MDR ne de bağlıdır. Dolayısıyla, 16 hattın sekizi, veri yolu olarak da görev yapar. Nitekim, aynı hatların hem veri, hem de adres için kullanılması, zamanda paylaşımı gerektirir. Yani, hatlar zamanın bir kesitinde adres, diğer bir kesitinde veri ile ilgili bilgileri taşırlar.

c) Aritmetik Lojik Birim (ALU)

CPU içinde yapılması gereken aritmetik ve lojik işlemler ALU içinde gerçekleşir. Karşılaştırma ve karar verme işlemleri de bu birim içinde gerçekleşir. ALU nun üzerinde işlem yapacağı verilerden birincisi (birinci işlenen) akümülatörde bulunur. ALU nun yetenekleri, CPU yu doğrudan etkilemektedir. Bu nedenle, ALU nun yapabildiği işlemler, CPU nun yeteneklerini belirleme açısından önem taşır. ALU, ACC ve FR ile doğrudan ilintilidir.

ALU lardan beklenen işlemler şunlardır:

Aritmetik işlemler: ALU toplama ve çıkarma işlemlerini yerine getirebilmelidir. Çarpma ve bölme işlemlerini her ALU yerine getiremez. Bu açıdan çarpma ve bölme işlemlerini yapabilen ALU lar üstün sayılırlar.

Lojik işlemler: ALU içinde temel VE, VEYA ve YADA işlemleri yerine getirilir.

Karşılaştırma ve karar verme işlemleri: iki büyüklüğün birbirine göre büyüklük, küçüklük veya eşitlik karşılaştırması yapılabilir. Ortaya çıkan duruma göre karar verilir ve sonuç durum saklayıcısına işlenir.
ALU içinde gerçeklenen işlemlerin ayrıntıları, komutların incelendiği bölümde verilecektir.

d) Akümülatör (ACC)

ACC, aritmetik ve lojik işlemlerin yerine getirilmesi sırasında, üzerinde işlem yapılacak verinin bulunduğu yerdir. ALU nun işleyeceği birinci işlenen ve işlem sonunda ortaya çıkan sonuç ACC de yer alır. Akümülatör, bir yerde ALU nun yardımcısıdır. ACC ün iç boyu CPU nun işleyebildiği sözcük uzunluğuna eşit olmalıdır. Dolayısıyla 8 bitlik mikroişlemcilerde ACC 8 bit ve 16 bitlik mikroişlemcilerde ACC 16 bit uzunluktadır. ACC, aritmetik ve lojik işlemlerde birinci işlenenin ve işlem sonucunun bulunması gereken tek yer olması nedeniyle, CPU nun önemli bir alt birimidir. Bu nedenle, bazı mikroişlemcilerde birden fazla ACC bulunmaktadır. Örneğin, MC6800 de ACC A ve ACC B olarak iki ACC bulunmaktadır.

e) Durum Saklayıcısı (FR)

Aritmetik lojik birim tarafından gerçeklenen işlemlerin sonunda ortaya çıkan durumların yazıldığı bir saklayıcıdır. Aslında, ortaya çıkan durumları gösteren bayrakların bir arada bulunduğu yerdir denebilir. Bu nedenle bu kütüğe bayrak saklayıcısı da denmektedir. Durum saklayıcısının içeriği, akümülatörün yüklenmesi durumunda da etkilenir. Durum saklayıcısının içeriği, karar verme işlemlerinde temel alınır. Aritmetik ve lojik işlemlerin sonunda şu durumlar ortaya çıkabilir:

Sıfır olabilir: İşlem sonunda, ACC e aktarılan sayı sıfır olabilir. ACC de bulunan sayının sıfır olması tüm bitlerinin sıfır olması demektir. Bazı mikroişlemciler için *Sıfır Bayrağının* etkilenmesi için, son yapılan işlemin aritmetik yada lojik olması gerekmez. Sıfır bayrağı, ACC deki sayının değerine göre her zaman etkilenir.

Negatif olabilir: İşlem sonunda, ACC e aktarılan sayı negatif olabilir. ACC de bulunan sayının negatif olup olmadığı, yedinci bitin bir olup olmaması ile sınanır. Bazı mikroişlemciler için *Negatif Bayrağının* etkilenmesi için, son yapılan işlemin aritmetik yada lojik olması gerekmez. Negatif bayrağı, ACC deki sayının değerine göre her zaman etkilenir.

Elde oluşabilir: Toplama işleminin sonunda elde biti oluşabilir. Yani toplama sonunda ortaya çıkan sonuç ACC e sığamamakta ve bir bit artmaktadır. *Elde Biti* olarak adlandırılan bu bit, FR içinde E bayrağıyla belirtilir.

Örnek:

```
   1011 1100
 + 1010 0011
 ------------
1   0101 1111
```

Elde Bayrağı

Borç oluşabilir: Çıkarma işleminde çıkan sayının ana sayıdan büyük olması durumunda borçlu kalınır. Bu durum FR içinde bulunan *Borç Bayrağıyla* belirtilir. Genellikle, elde ve borç durumları aynı bayrakla belirtilir. Bayrağın taşıdığı anlam, son işlemden çıkartılabilir.

İşaret bitinin 0 olması, sonucun negatif olduğunu belirtir. Dolayısıyla borçlu kalındığı anlaşılır. Bu durum borç bayrağıyla belirtilir. Doğru sonuç bu ara sonucun 2 ye tümlenmesi ile bulunur ve ACC e bu sonuç yerleştirilir.

Örnek:

```
   0001 0000
 - 1100 0000
   0001 0000
 + 0100 0000    Çıkarılan sayının 2 ye tümleyeni
   0101 0000
```

Yarım Elde veya Borç Oluşabilir: İlk dört bitin toplanması veya çıkarılmasında elde veya borç oluşabilir. İkili onluk sayıların toplanması ve çıkarılması işlemlerinde önemi olan bu bayrağın normal elde ve borç bayrağının etkilenmesi ile aynı biçimde olmaktadır.

Örnek:

```
   0010 0101
 + 0011 0111
        1100    ilk dörtlüğün toplanması ile onucu ortaya çıkar.
```

Bilindiği gibi, ikili-onluk sayı düzeninde bir basamağın değeri 1001 i aşamaz. *Yarım Elde Bayrağı* işte bu durumu belirler.

Örnek:

```
   0010 0101
 - 0011 0111
   0101 1001
        0111   in ikiye tümleyeni
        1101    ortaya çıkan 1101 sonucu, ikili onluk sayı düzeni için tanımsızdır.
```

Bu sonuç birinci basamakların çıkartılması sonunda borçlu kalındığını gösterir. Durum borç bayrağı ile gösterilir.

Taşma oluşabilir: Toplama işlemi sırasında Elde oluşmuş olabilir. Bu durum Elde bayrağı ile gösterilir. Elde bayrağı bir iken yeni bir toplam işlemi yapılırsa, ACC ün boyundan iki bit büyük bir sonuç ortaya çıkabilir. Bu duruma taşma denir ve *Taşma Bayrağı* ile gösterilir.

FR içinde yukarıda tanıtılan bayrakların hepsinin bulunması gerekmez. Bu nedenle bazı mikroişlemcilerde daha az sayıda bayrak bulunabilir. Bazı mikroişlemcilerde ise daha başka durumları göstermek içinde bayraklar bulunabilir. Örneğin bazı Durum Saklayıcılarında Eşlik bayrağı bulunur. ACC de bulunan veri içindeki 1 lerinin tek yada çift sayıda olduğunu belirtir. Örneğin bu bayrağın 1 olması, ACC içinde bulunan sayı içindeki 1 lerin çift sayıda olduğunu gösterir.

Birinci aşama:

```
      1011 1100
+     1010 0011
  1 0101 1111
    ↑
    elde bayrağı
```

İkinci aşama :

```
  1   0101 1111
+     1101 0000
1 0  0010 1111
↑ ↑
  elde bayrağı
  taşma bayrağı
```

Aşağıda durum bayrakları topluca gösterilmiştir:

Elde	E	C
Yarım elde	Y	AC
Negatif	N	S
Sıfır	S	Z
Taşma	T	O

Bu bayrakların Durum Saklayıcısı içinde yerleştirilmesine bir örnek Şekil 1.9 da verilmiştir.

T	S	N	Y	E

Şekil 1.9 Durum saklayıcısının bayrakları

Durum saklayıcısının her bayrağının doğal olarak iki konumu vardır; 0 ve 1. bir bayrağın etkin hali, lojik 1 konumu olarak kabul edilmektedir. Örneğin sıfır bayrağının 1 olması, ACC içeriğinin sıfır olmasına karşı düşer. ACC ün içeriği sıfırdan farklı ise bu bayrak 0 konumunda kalır.

f) Yardımcı Saklayıcılar (REGS)

CPU içindeki işlemlere hız kazandırmak amacıyla, yardımcı saklayıcılar kullanılmaktadır. Bu saklayıcılar, üzerinde sık sık işlem yapılacak işlemler için kullanılır. Yardımcı saklayıcılarda bulunan veriler, MDR üzerinden hafızaya veya giriş-çıkış arabirimine gönderilebilir. Benzer şekilde hafıza ve giriş-çıkış arabirimindeki bilgiler bu saklayıcılara aktarılabilir. Yardımcı saklayıcıların ACC den

tek farkı, aritmetik lojik işlemlerde birinci işlenen yerini alamamalarıdır. Yardımcı saklayıcı kullanılması konusunda Motorola ve İntel tasarımları arasında önemli farklar vardır: Motorola MC6800 ailesinde iki akümülatör bulunmakta, buna karşın yardımcı saklayıcı bulunmamaktadır. İntel 8085 te ise tek akümülatöre ek olarak B, C, D, E, H, L yardımcı saklayıcıları bulunmaktadır. Her iki yönteminde üstün ve eksik olduğu yönler bulunmaktadır. Bazı gelişmiş mikroişlemcilerde, ACC sayısı arttırılmakta ve bu arada yardımcı saklayıcılara bulunmakta böylece, iki yöntemin üstünlüklerinden yararlanılmaktadır.

g) Program Sayacı (PC)

Program sayacı, genellikle, bir sonraki işlenecek komutun bulunduğu hafıza gözünün adresini taşır. Komut bir hafıza gözüne sığmıyor ise, komutun bulunduğu ilk gözün adresini içerir. Komut saklayıcısının çıkışı Şekil 1.8 de görüldüğü gibi MAR ne bağlıdır. PC in boyu, hafıza içindeki her gözü belirtmesi gerektiğinden MAR ın boyuna eşit olmalıdır.

Daha önce değinildiği gibi, komutların boyları, işlevlerine göre değişmektedir. Bu nedenle, peşpeşe gelen program adımlarında, bir sonraki komutun bulunduğu hafıza gözünün adresi her adımda hesaplanır ve bu değer Program Sayacına yazılır.

Programın akışı bazen alınan kararlar sonucu değiştirilir. Bu durumda, yeni izlenecek program parçasının ilk komutunun bulunduğu adres PC na yazılır.

h) Komut Saklayıcısı (IR)

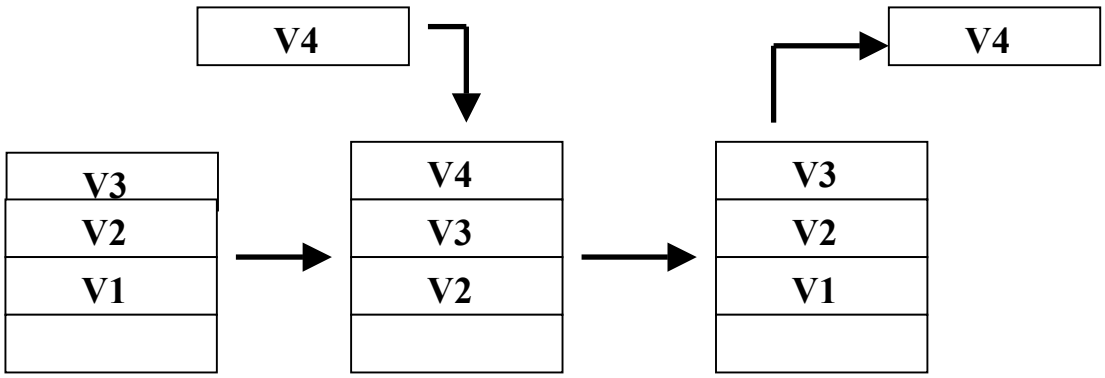
Komut Saklayıcısı, o anda işlenen komutu içerir. Hafızadan okunan bilgi içinde hangisinin komut ve hangisinin işlenen olduğu şöyle anlaşılır: Program Sayacının gösterdiği ilk adreste komut olduğu varsayılır. Bu başlangıç noktasından sonra, her adımda, komutun boyu hesaplanarak, bir sonraki komutun başlangıç noktası belirlenir.

i) Komut Çözücü (ID)

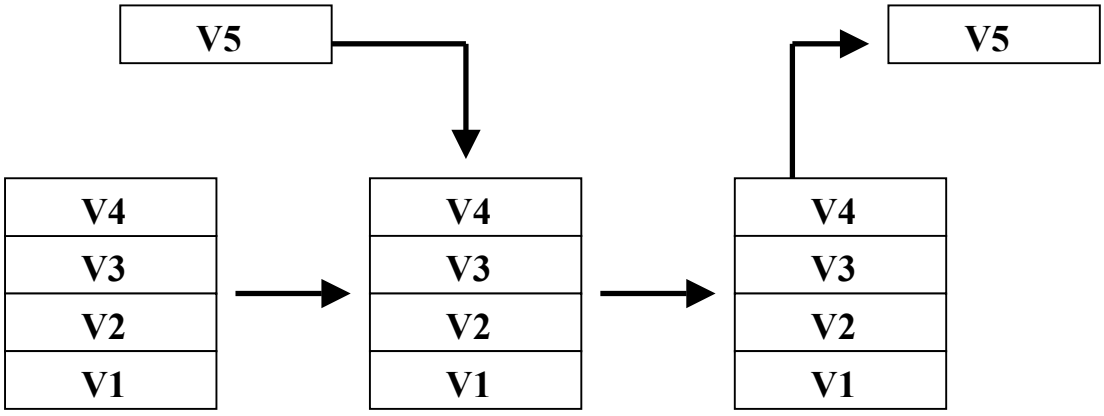
Komut Saklayıcısında bulunan komutun taşıdığı anlamın çözümlendiği yerdir. Komutun çözümlenmesi sonunda, CPU içinde ve dışında yapılacak işler ve bu işlerin yapılacağı yerler belirlenir. Ardından bu işler Kontrol biriminin güdümünde, ALU ve diğer saklayıcı ve ACC ile birlikte yerine getirilir.

j) Yığın Göstergesi (SP)

Yığın Göstergesini tanıtma işlemine yığın kavramını tanıtarak başlamakta yarar vardır. Bilgisayar dilinde verileri üst üste yığmaya ve gerektiğinde, verileri yığından teker teker geri almaya Yığın işlemi denir. Yığının çalışması Şekil 1.10 ve 1.11 de gösterilmiştir.



Şekil 1.10 Yığına veri atılması ve çekilmesi işlemleri

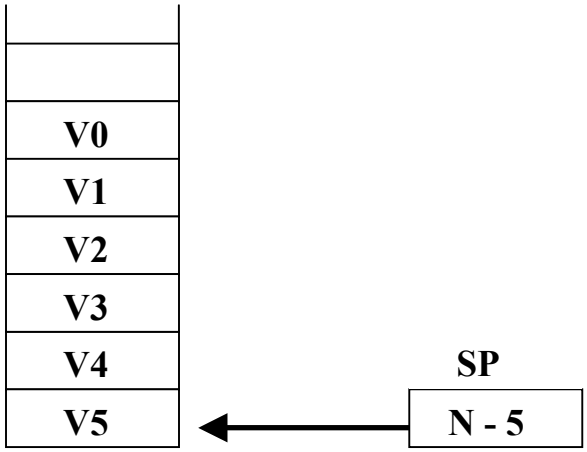


Şekil 1.11 Yığının boyu sınırlı ise veri kaybı olabilir

Şekil 1.10 da örnek olarak ele alınan yığının dört gözü bulunmakta ve bunlar CPU içinde yer almaktadır. Yığının gözlerinden ilk üçüne daha önceden üç veri (V1, V2 ve V3) konmuştur. Yığına V4 verisi yığılmak istendiğinde, yığında daha önceden bulunan veriler birer basamak aşağıya kayarlar. Böylece veri için yer açılmış olur. Yığından veri çekilmeye kalkışıldığında ilk olarak V4 geri alınır. V4 verisinin geri

alınmasının hemen ardından, yığındaki tüm veriler birer basamak yukarı kayarlar. Bu örnekten de anlaşılacağı gibi, yığına son giren veri ilk olarak geri alınmaktadır.

V4 verisinin yığına atılmasının hemen ardından V5 yığına atılmak istenirse, yığında bulunan tüm veriler aşağıya doğru kayarlar. Bu kayma sonucunda V5 için yer açılır. Ancak, yığının dört gözü olması nedeniyle V1 verisinin yazılacağı bir göz kalmaz. V5 verisinin yığına atılmasından sonra yığından bir veri geri çekilmek istenirse, yığının özelliği gereği V5 dışarı alınacak ve hemen ardından yığındaki tüm veriler birer yukarı kayacaklardır. Bir önceki adımda, V1 verisi yığından düştüğü için iki adım önceki durum oluşmayacak ve V1 değeri eski yerine gelemeyecektir.



Şekil 1.12 Yığın göstergesinin çalışma yöntemi

Bilgisayar içinde önemli yeri olan yığın işleminin, bu amaçla ayrılmış olan sınırlı sayıda gözden oluşması yukarıda verilen örnekten anlaşılacağı gibi sakınca yaratmaktadır. Bu sakıncayı ortadan kaldırmak üzere, hafıza içinde yığın kurulması yoluna gidilmiştir. Bu yöntemde, hafıza içinde istenen bir alanda yığın oluşturabilmek için yığın kullanılabilmektedir. Hafıza içinde, yığının nerede kurulduğunu belirtmek için CPU içinde yığın göstergesi kullanılmaktadır. Yığın Göstergesinin nasıl çalıştığı şekil 1.12 de gösterilmiştir. Program çalıştırılmadan önce SP nin değeri bir azalarak, yığına yeni atılacak verinin yerleşeceği adresi gösterir. Yığından bir veri çekilmesi durumunda ise SP nin değeri bir artar.

Hafızada kurulan yığın ile, CPU içinde kurulan yığın arasında işlev açısından fark olmamakla beraber işleyiş açısından önemli farklar vardır. CPU içinde kurulan yığının boyu sınırlıdır. Yığına her yeni veri atıldığında, eski veriler birer adım aşağı kaymaktadır. Veri çekildiğinde ise birer adım yukarı kaymaktadırlar. Hafızada düzenlenen yığında ise, yığındaki elemanların kayması söz konusu değildir. Kayan sadece Yığın Göstergesidir. Bir başka farkta hafıza içerisinde yığının aşağı doğru büyümesidir. Bunun nedeni şöyle açıklanabilir. Bilgisayarda, programlar küçük adreslerden başlayarak ve programın boyu uzadıkça yüksek adreslere erişilir. Bir başka deyişle program küçük adreslerden büyüklere doğru uzanır. Yığının yüksek

adreslerden başlatılarak aşağıya doğru uzaması, veya sarkması, hafızayı en uygun kullanma biçimi olarak kabul edilmektedir.

Yığın işlemi, sıfır adresli komutlar için kullanılmakla beraber, bilgisayarın çalışması içinde kullanılmaktadır. Sözgelimi, alt programlara gidiş ve gelişlerde, saklayıcıların içerikleri ve dönüş adresi yığında saklanmaktadır. Ayrıca kesme işlemlerinde de benzer bilgiler yığına atılmaktadır.

Yığın Göstergesinin boyunun, MAR boyuna eşit olması beklenir. Ancak daha küçük boyda SP kullanan bilgisayarlar da bulunmaktadır.

k) Sıralama Saklayıcısı (IR)

Sıralı verilerin hafızaya yazılması veya hafızada sıralı bulunan verilerin okunması için kullanılır. Bilindiği gibi, matematikte diziler matrisel yapılar önemli ölçüde kullanılmaktadır. Tek boyutlu yada çok boyutlu dizi içindeki elemanları belirtmede indis kullanılması da üzerinde alışılmış bir yöntemdir. Sıralama saklayıcısı bu tür özelliği olan veriler için kullanışlı çözümler sağlar.

IR nün kullanılması ile ilgili şu örnek verilebilir: Hafızada bulunan bir dizinin başlangıç adresi IR ne yüklenir. Ardından bu dizideki n. Veri CPU ya alınmak istenirse, yapılacak tek işlem n. verinin alınacağını söylemektir. IR nün yetenekleri, sıralı adresleme yöntemi içinde tanıtılmıştır.

l) Kontrol birimi (CON)

CPU nun en önemli birimlerinden biri kontrol alt birimidir. Bu birim hem CPU içindeki çalışmayı düzenler, hem de bilgisayar içindeki çalışmayı düzenler. CPU içindeki çalışmayı , Komut Çözücü alt biriminin yönlendirmesi ve diğer alt birimlerin işbirliğiyle düzenler.

Bilgisayarın çalışmalarını düzenlemek için Kontrol biriminin aşağıda sıralanan giriş ve çıkış uçları bulunmaktadır.

GİRİŞLER

Saat	Clock	CLOCK
Albaştan	Reset	RST
Bekle	Wait	WAIT
Kesme	Interrupt	INT
Kesme isteği	Interrupt Request	IRQ
Yalıtım	Hold	HLD

ÇIKIŞLAR

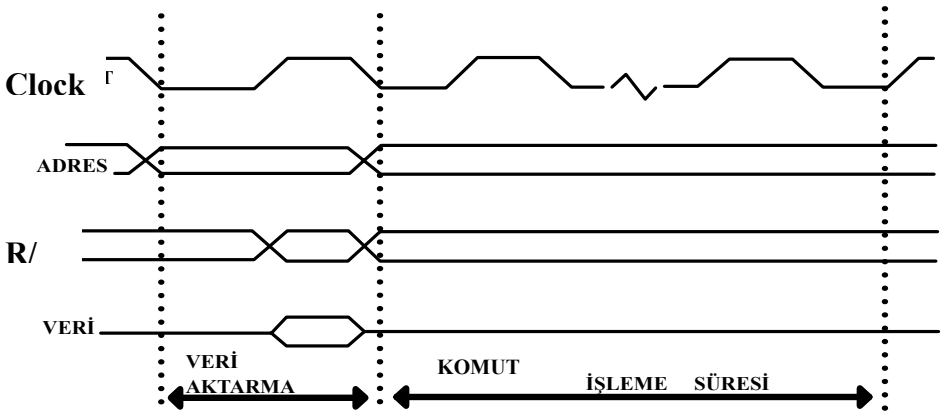
Oku/yaz	Read/Write	R/W
Yalıtıldı	Hold Acknowledge	HLDA

Saat çıkışı	Clock Out	OUT
Hafıza yada Giriş/Çıkış	Memory or Input/Output	M-I/O

GİRİŞLER:

Clock: Bilgisayar içindeki işlemlerin bir sıra izlemesi ve adımların uygun atılması için tüm birimlerin ayak uyduracağı bir saat işareti gereklidir. Bilgisayar içindeki her işlem bu saat işaretine bağlı olarak gerçekleştirilir.

Şekil 1.13 de bir komutun işlenmesi aşamaları, saat işaretine bağlı olarak gösterilmiştir. Şekilden de görüldüğü gibi bir komutun işlenmesinde ilk aşama , komutun hafızadan alınıp CPU ya getirilmesidir. CPU içinde IR ne yerleşen ve daha sonra çözümlemesi yapılan komutun gerektirdiği işlemler, komutun özelliğine göre yapılır. Bu nedenle her komutun süresi birbirinden farklı olacaktır.



Şekil 1.13 Saate bağlı olarak bir komutun işlenmesi

Saat işareti CPU dışında üretildiği gibi, CPU içinde de üretilmektedir. Bu amaçla gerekli devreler CPU içinde gerçeklenmektedir. Bu tür donanımlarda salınım frekansını belirleyecek elemanlar CPU ya dışarıdan bağlanmaktadır. Salınım frekansını kararlı kılmak için genellikle kristal kullanılmaktadır.

Reset: Bu giriş bilgisayarı ilk açıldığı duruma getirmek için kullanılır. Reset edilen bir bilgisayar, bu durumda, izlemesi gereken programın başına gider ve bu programın işletilmesine başlar. Reset girişi, bilgisayar ilk açılma koşullarına döndürülmesi gerektiği durumlarda da kullanılabilir.

Wait: CPU nun bir süre durmasını sağlayan bir giriştir. Bu girişin etkin hale getirilmesi ile CPU nun çalışması durdurulur. Wait girişi etkin olamayan konumuna döndürülene kadar CPU çalışmaz. Bu giriş etkisiz konuma geçince, CPU kaldığı yerden çalışmasın devam eder.

Interrupt (Kesme) : Bu girişin etkin hale gelmesi ile, CPU o anda yürüttüğü programı keser ve daha önce belirlenmiş olan kesme hizmet programını işletmeye başlar. Kesme hizmet programının tamamlanmasından sonra, kesmenin geldiği anda yaptığı işe devam eder.

Interrupt Request (Kesme İsteği): Kesme isteği girişi de kesme girişi gibi çalışmaktadır. Bu girişin tek farkı, bu girişe uygulanan kesme isteğine göre, istendiğinde uyulması veya uyulmamasıdır. Bu amaçla CPU içinde Kesme İsteği karar bayrağı bulunmaktadır. Bu bayrağa verilecek değer ile kesme isteği girişi etkin yada etkisiz kılınabilir.

Hold (Yalıtım) : Yalıtım girişinin etkin hale getirilmesi ile, CPU adres ve veri yollarının çıkışlarını üçüncü konuma getirir. Sonuçta, CPU kendisini adres yolu ve veri yolundan yalıtır. Bu durum aynı veri yollarının birden fazla CPU veya başka akıllı birimler tarafından kullanılmasını sağlamak için gereklidir.

ÇIKIŞLAR:

Read/Write (Oku/yaz) : İki yönlü çalışan veri yolu üzerindeki verilerin akış yönünü belirleyen bir çıkıştır. Bu çıkışın 1 konumu okuma ve 0 konumu yazma durumunu belirler. Okuma yönü hafızadan ve giriş- çıkış arabiriminden CPU ya doğrudur. Yazma yönü ise CPU den hafıza ve giriş- çıkış arabirimine doğrudur.
İntel ve türevleri mikroişlemcilerde Read/write çıkışı oku ve yaz olarak ayrı ayrıdır. Ancak okuma ve yazma kavramında farklılık yoktur.

Hold Acknowledge (Yalıtıldı): Veri yolu ve adres yolunun yalıtıldığını, aynı yolları kullanan diğer birimlere duyurmak için gerekli bir çıkıştır.

Clock out(Saat Çıkışı): Saat osilatör devresi CPU içinde olan mikroişlemcilerde, bilgisayar içindeki diğer birimler için gerekli olan saat işaretini elde etmek için gerekli olan bir çıkıştır. Bu çıkış, ya CPU ya uygulanan yada CPU da üretilen saat frekansında yada bunun bölünmüşü bir frekansta olabilir.

Hafıza yada Giriş/Çıkış arabirimi: Adres yolu üzerinde bulunan adres bilgisinin, hafıza gözünü mü yoksa bir giriş-çıkış arabirimini mi adreslemek için kullanılacağını belirtir.

1.4 MERKEZİ İŞLEM BİRİMİNİN DONANIM ÖZELLİKLERİ

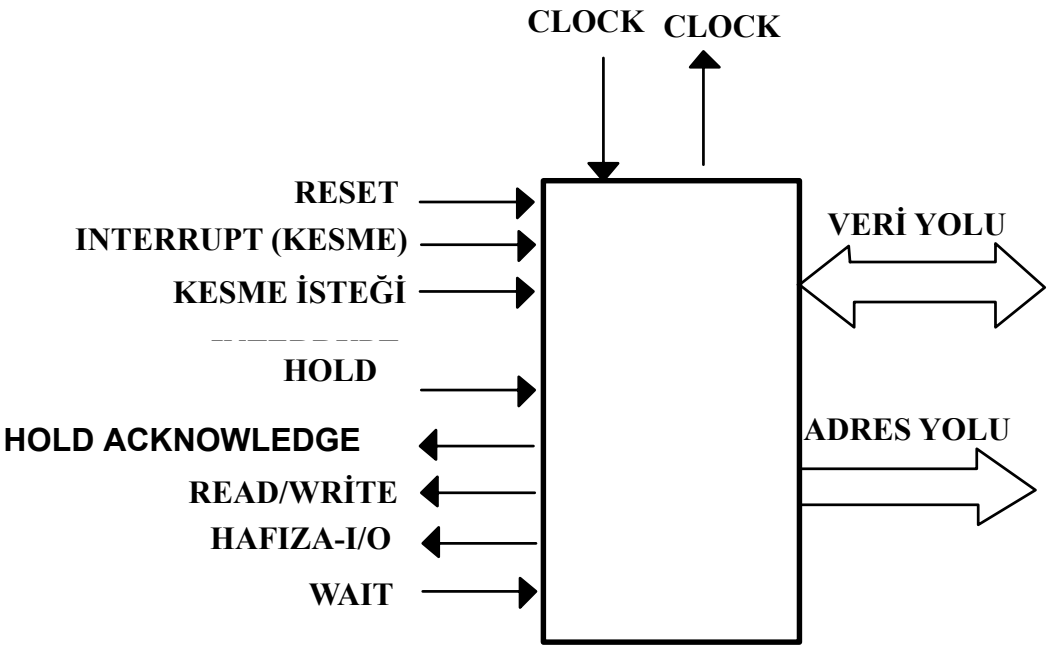
CPU ya dışarıdan bakıldığında şu uçlarının olacağı buraya kadar edinilen bilgilerle söylenebilir:

Veri Yolu	**Data Bus**
Adres yolu	**Address Bus**
Clock	**CLOCK**
Reset	**RST**

Wait	**WAIT**
Interrupt	**INT**
Interrupt Request	**IRQ**
Read/Write	**R/W**
Hold	**HLD**
Clock Out	**OUT**
Hafıza yada Giriş /Çıkış Arabirimi	**M-I/O**

Bu girişlerin dışında besleme uçlarının da olacağı açıktır. CPU nun iki yolu üç konumlu kapılarla tasarlanmıştır. Bu yollara bulunan her bir hat çıkışı bir anda ancak bir TLL yük sürebilecek güçtedir. Yolların dışında kalan diğer çıkışlar, üç konumlu kapı biçiminde olmayabilir ancak sürme güçleri genellikle 1 TLL yüke eşittir.

Görüldüğü gibi, CPU nun çıkışları oldukça güçsüz görülmektedir. Bu nedenle, bilgisayar donanımında giriş akımı az olan devre elemanlarının kullanılması gerekir.



Şekil 1.14 Merkezi İşlem Biriminin dış görünümü

1.5 ADRESLEME YÖNTEMLERİ

Komut yazımında en önemli konulardan biri, adresleme yöntemidir. Adresleme yöntemi, işlenenin nerede bulunacağını belirtmek için kullanılan yöntemleri kapsar. Bir mikroişlemcinin yeteneklerini belirlerken, kullanabildiği adresleme yöntemlerinin

nitelik ve niceliği göz önüne alınmaktadır. Bir bilgisayarda kullanılan adresleme yöntemlerinin sayıları arttırılabilirse de temel olarak altı tür adresleme yöntemi vardır. Diğer adresleme yöntemleri bu altı adresleme yönteminden türetilmektedir. Bu bölümde altı tür adresleme yöntemi örneklerle tanıtılmıştır. Temel adresleme yöntemleri şunlardır :

İvedi adresleme
Doğal adresleme
Doğrudan adresleme
Dolaylı adresleme
Sıralı adresleme
Bağıl adresleme

1.5.1 İVEDİ ADRESLEME

İvedi adresleme yönteminde, işlenen yerine yazılan bilgi bir adres değildir. İşlenen, üzerinde işlem yapılacak verinin kendisidir. İvedi adreslemeye örnek bir komut şöyle yazılabilir:

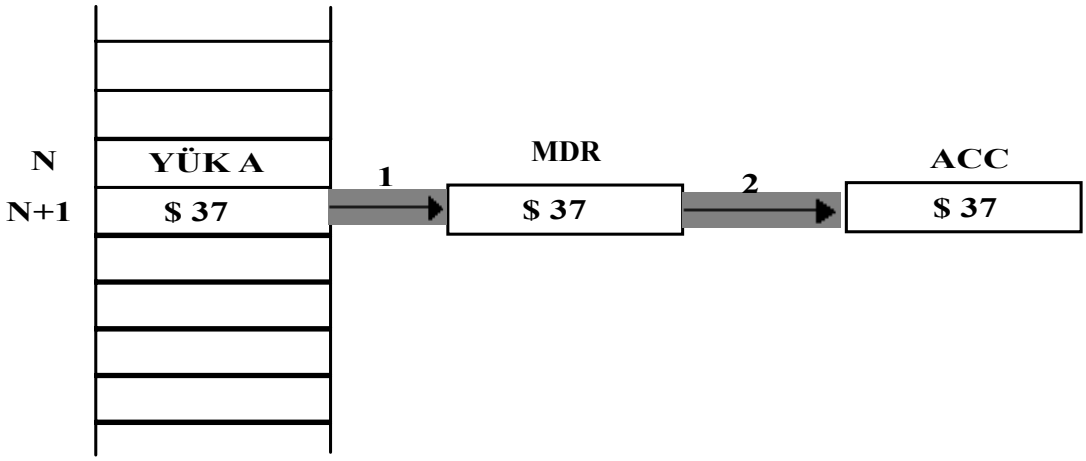
YÜK A, $37

Bu komutla, $37 sayısı doğrudan doğruya akümülatöre yüklenecektir.Bu komut iki alana yazılabilir. 8 bitlik mikroişlemci için bu alanlar iki tane sekizlik, 16 bitlik mikroişlemci için ise iki tane 16 bitlik alan olacaktır. Şekil 1.15 de 8 bitlik mikroişlemciler için örnek verilmiştir.

1. SEKİZLİK	**YÜK A İVEDİ**
2. SEKİZLİK	**$ 37**

Şekil 1.15 8 bitlik mikroişlemcilerde ivedi yükleme kalıbı

İvedi adresleme yöntemi kullanıldığında, komutun hafızaya yazılışı ve komutun işlemesi süreci Şekil 1.16 da gösterilmiştir.



Şekil 1.16 İvedi Yüklemenin hafızada görünümü ve işleyişi

1.5.2 DOĞAL ADRESLEME

Doğal adresleme yönteminde işlenen alanında bulunan bilgi, işlenenin bulunduğu saklayıcısının adıdır. Bu açıklamadan da anlaşılacağı gibi, doğal adresleme yöntemi saklayıcılar üzerinde geçerli olan bir adresleme yöntemidir. Bu nedenle bu adresleme yöntemine, saklayıcı adresleme yöntemi de denilmektedir. Doğal adresleme yöntemine örnek olabilecek bir komut aşağıda verilmiştir.

AKT A,B

Bu komutla B saklayıcısının içeriği A saklayıcısına aktarılacaktır. Doğal adresleme yöntemine göre yazılmış bir komut genellikle tek alan içine sığdırılmaktadır. Bu nedenle hafıza içinde görünümü şekil 1.17 deki gibi olur.

1. SEKİZLİK	AKTAR DOĞAL

Şekil 1.17 Doğal adresleme kalıbı

1.5.3 DOĞRUDAN ADRESLEME

Doğrudan adresleme yönteminde, işlenen yerinde işlenecek verinin bulunduğu yada bulunacağı hafıza gözünün adresi yazılıdır. Komut, verinin hafızadan okunup CPU içine getirilmesini gerektiriyor ise, bu adres, verinin bulunduğu hafıza gözünün adresidir. Eğer komut CPU içindeki bir verinin hafızada bir göze yazılmasını gerektiriyor ise, bu durumda, işlenen alanında bulunan adres, verilerin yazılacağı hafıza gözünün adresini içerir. Bu adresleme yöntemine örnek bir komut şöyledir:

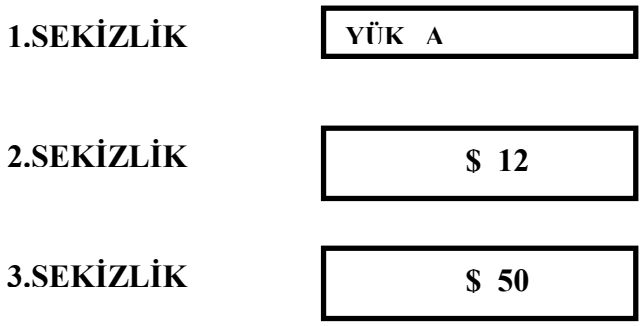
YÜK A,<$1250>

$ 1250 sayılı hafıza gözünün içeriği hafızadan alınıp CPU ya getirilecek ve Akümülatöre yüklenecektir. CPU nde bulunan bir verinin hafızada bir göze yazılması için ise şu komut yazılabilir:

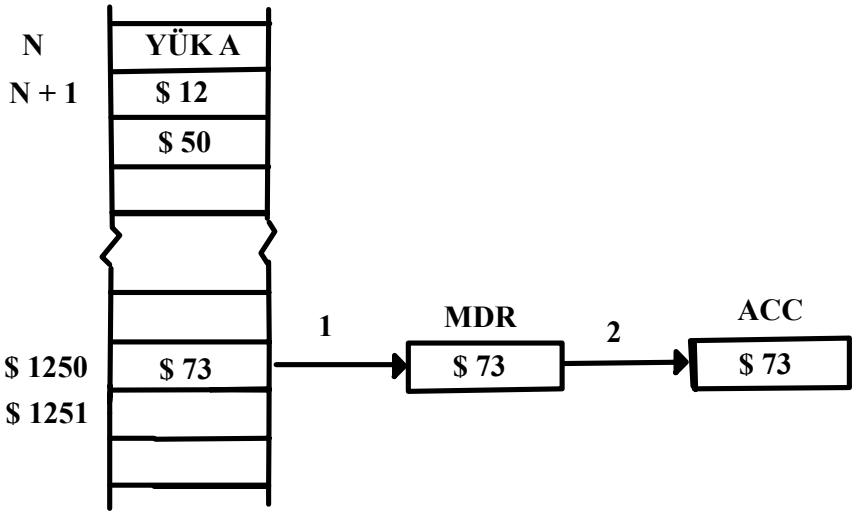
YAZ A, <$1250>

Dikkat edilirse, bir hafıza gözü içeriğinin okunması sürecinde hafıza gözü adresi parantez içine alınmaktadır. Hafızaya veri gönderildiğinde, içeriği gönderilen saklayıcı parantez içine alınmamaktadır. Bu uygulama çoğunluk kullanıcının benimsediği bir yazım yöntemidir.

Yük A, <$1250> komutunun hafızada görünümü ve komutun işleniş adımları sırasıyla şekil 1.18 ve 1.19 da verilmiştir.



Şekil 1.18 Doğrudan adresleme kalıbı



Şekil 1.19 Doğrudan adresleme yöntemi ve işleyişi

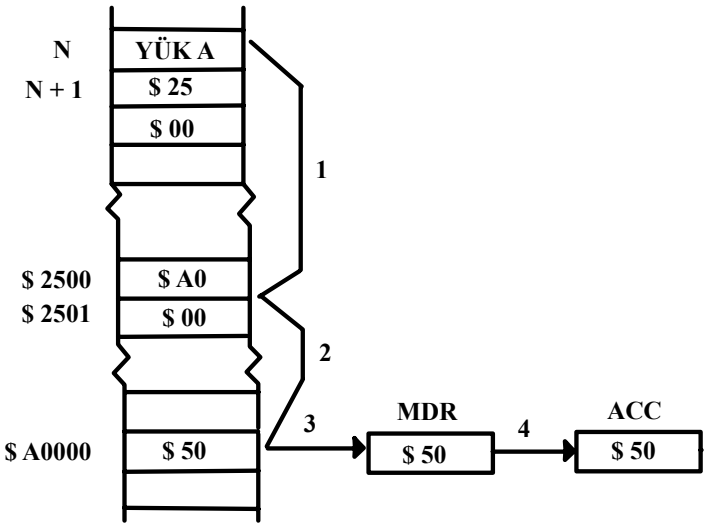
1.5.4 DOLAYLI ADRESLEME

Dolaylı adresleme yönteminde, işlenen yerinde, verinin adresi yerine, bu adresin yazılı olduğu hafıza gözünün yazılı olduğu hafıza gözünün adresi yazılıdır. Dolayısıyla, dolaylı adres içeren bir adres komutunda, ilk aşamada, işlenen yerinde belirtilen adrese gidilir. Bu hafıza gözünde (bu hafıza gözü ve bunu izleyen hafıza gözü adres belirtebilir) bulunan sayı, verinin bulunduğu hafıza gözünün adresidir. Dolayısıyla, ikinci aşamada bu adrese gidilerek veri okunur. Dolaylı adreslemeye örnek olmak üzere aşağıdaki komut yazılmıştır.

YÜK A, (($2500))

$2500 ve 2501 sayılı hafıza gözlerinin içinde yazılı olan adresin belirttiği hafıza gözünün içeriği akümülatöre yüklenecektir. Adreslemenin dolaylı olduğunu belirtmek için işlenen (( )) işaretleri arasına yazılış biçimi Şekil 1.20 de ve komutun işleniş biçimi Şekil 1.21 de gösterilmiştir.

Şekil 1.20 Dolaylı adresleme kalıbı



Şekil 1.21 Dolaylı adresleme yöntemi ve işleyişi

Şekil 1.21 den görüldüğü gibi,komutun işlenmesinde önce, işlenen bölümünde belirtilen $2500 adresine gidilmektedir.ikinci aşamada $2500 ve $2501 sayılı hafıza gözleri birlikte yorumlanarak, verinin $A000 sayılı hafıza gözünde bulunduğu sonucuna varılmaktadır. Bu nedenle $A000 sayılı hafıza gözünün içeriği olan $50 sayısı akümülatöre aktarılmaktadır. Üzerinde işlem yapılan adrese Etkin Adres (EA) denir.

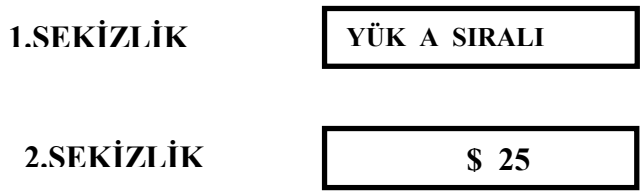
1.5.5 SIRALI ADRESLEME

Dizi yapısında olan veriler hafızada sıralanmış olarak saklanır. Sıralanmış verileri sıra numarasına göre işleme sokmak alışkanlık haline gelmiştir. Bu nedenle CPU içinde sıralama saklayıcısı bulunmaktadır. Dizinin ilk elemanının adresi sıralama saklayıcısına yazılır ve sıralı adresleme yöntemi kullanılırsa, veriler üzerinde işlem yaparken, verinin bulunduğu hafıza gözünün adresini belirtmek yerine, verinin sıra numarasını belirtmek yeterli olur. Sıralı adreslemenin nasıl çalıştığı aşağıdaki örnek üzerinde görülebilir.

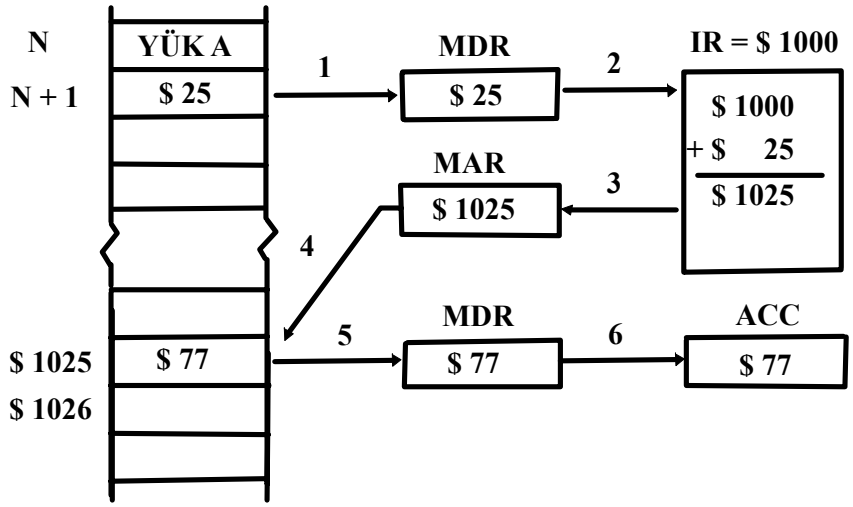
YÜK A, <+$25>

Sıralama saklayıcısının içeriğine $25 eklenerek bulunacak adresin belirttiği hafıza gözünün içeriği akümülatöre yüklenir.

Bu örnekte komut işlenmeden önce IR nın $1000 değerine eşit olduğu varsayılırsa, bu komut işlenirken, ilk aşamada, $1000 sayısına $25 eklenerek $1025 adresi bulunur. İkinci aşamada, veri $1025 sayılı hafıza gözünden okunarak akümülatöre yüklenir. Örnek hafızada görünümü ve hafızada işleniş aşamaları sarasıyla Şekil 1.22 ve 1.23 de verilmiştir.



Sekil 1.22 Sıralı adresleme kalıbı



Şekil 1.23 Sıralı adresleme yöntemi ve işleyişi

Sıralı adresleme yönteminde, IR ne eklenen sayıya, *Sıra Numarası* veya *Kayıklık* denir. 8 bitlik mikroişlemcilerde, eğer IR 16 bit uzunlukta ise kayıklık 8 bit olmaktadır. Kayıklığın yada sıra numarasının işaretsiz bir sayı olması yeğlenmektedir. Bunun sonucu olarak şunlar söylenebilir.8 bitlik mikroişlemcilerde hafızanın istenen bir yerinden başlayarak bir dizi yerleştirilebilir. Dizinin eleman sayısı en çok 256 olabilir. 16 bitlik mikroişlemcilerde IR nın boyu daha uzun olabileceği gibi kayıklık ta daha uzun olabilir. Bazı 8 bit mikroişlemcilerde, IR nın 8 bit ve kayıklığın 16 bit seçilmektedir. Bu yöntemde, dizinin boyu 64 K ya kadar çıkabilmekte ancak dizinin başlangıç adresi $0000 $00FF adresleri arasında sınırlı kalmaktadır.

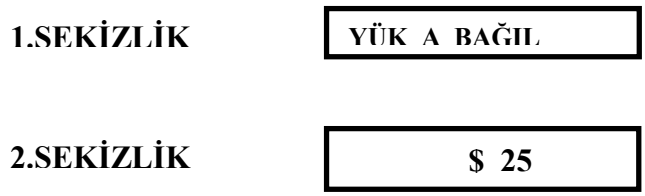
1.5.6 BAĞIL ADRESLEME

İşlenen adresini, bazen, komutun hafızada bulunduğu adrese göre tanımlamak gerekebilir. İşte bu durumlarda Bağıl adresleme yöntemi kullanışlı olmaktadır. Bağıl adresleme yönteminde, işlenen yerinde görülen sayı, üzerinde işlem yapılacak verinin, komutun yazılı olduğu adresten ne kadar uzakta olduğunu belirtir. Örnek bir komut şöyle olabilir:

YÜK A, {$25}

Komutun yazılı olduğu adresten $25 adım ilerideki hafıza gözünün içeriğini akümülatöre yükle.

Bağıl adreslemeyi vurgulamak için işlenen { } işaretleri arasında verilmiştir. Yukarıdaki örnek komutun komut kısmının $B000 adresinde yazılı olduğu varsayılırsa, akümülatöre yüklenecek verinin $B025 sayılı hafıza gözünde bulunacağı ortaya çıkar. Bağıl adresli komut yazım biçimi Şekil 1.24 ve örnek komutun işlem adımları Şekil 1.25 de verilmiştir.



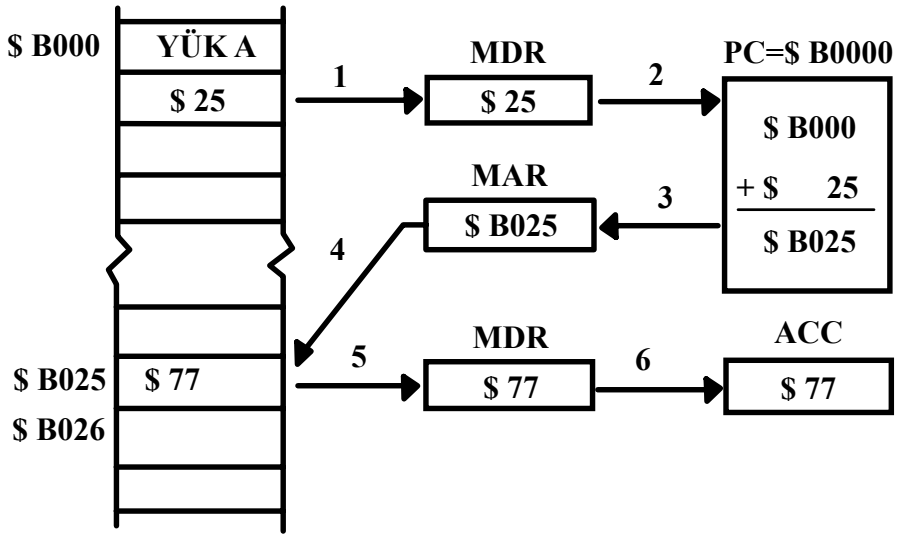
Şekil 1.24 Bağıl adresleme kalıbı

8 bitlik mikroişlemcilerde, bağıl adresleme yöntemi her türlü işlem için kullanılamamaktadır. Hatta, sadece dallanma ve bağlanma işlemlerinde kullanıldığı söylenebilir. Dallanma ve bağlanma işlemleri, program akışını, koşullu veya koşulsuz olarak değiştiren komutlardır. Sözgelimi, sonucun sıfırdan büyük olması halinde program akışını değiştirerek programın $25 adım ilerideki komuttan devam etmesini istersek şöyle yazabiliriz:

DEB $25

Bir evvelki komutta elde edilen sonuç sıfırdan büyük ise programı bir alt satırından devam ettirmek yerine, $25 adım ilerideki komuttan devam ettirir.

Bağıl adreslemedeki işlenene, kayıklık denmektedir. Kayıklık miktarı artı veya eksi olabilir. Yani, ileri doğru adres belirtilebileceği gibi geri doğru adres te belirtilebilir. Bu nedenle Kayıklık işaretli sayı olarak kullanılır. 8 bitlik mikroişlemcilerde, kayıklık için 8 bit ayrılmaktadır. Bu değer, 16 bitlik mikroişlemciler için 16 bit olabilmektedir.



Şekil1.25 Bağıl adresleme yöntemi ve işleyişi