

mikroişlemciler giriş

Rüyal ERGÜL

UDK: 621.3.049.771.14

ÖZET

Mikroişlemciler konusunu genel olarak işleyen bu makalede, önce mikroişlemcilerin nasıl geliştirildiği incelenmiş ve sonra da mikrobilgisayar iç yapısı anlatılmıştır. Giriş-çıkış ve denetim işlevleri adreslemeye ağırlık vererek açıklanmıştır. Bugünkü mikroişlemci devreler hakkında bilgi verilmiş ve mikroişlemci seçimi, programlanması konusunda karşılaşılabilecek güçlükler tartışılmıştır.

SUMMARY

in this tutorial paper, a review of microprocessor principles and architecture of computers in general are presented. Then, input-output and control functions are briefly discussed with emphasis on addressing techniques. Present day microprocessor chips are introduced and difficulties in selecting chips and implementation of microprocessor based circuits are discussed.

I. İLK MİKROİŞLEMCİLER:

1960 yılları yarıiletken devre elemanları endüstrisinin, bilgisayar ve masa üstü hesaplayıcılar için büyük sayılarda mantık devreleri ürettiği yıllardır. Bütün firmalar hesaplayıcı tasarımlarını kendi özel koşullarını yerine getirecek biçimde KÇT - Küçük çapta tümleşik ve OÇT - Orta çapta tümleşik devreleri kullanarak yapmaktaydılar. Tümleşiklik düzeyinin (*level of integration*) artması, yüksek tasarım ve geliştirme giderlerine karşın, büyük sayıda yonga üretilmesi sonucu düşük fiyatlarla satışı sağlayabilecekti. Bu yönde yapılan çalışmalar sonucu, 1971 senesi Şubat ayında Mostek Firmasıyla işbirliği yapan Japon Busicom firması, 4 işlemleri bir hesaplayıcının gerektirdiği bütün mantık devrelerini üzerinde toplayan bir yonga geliştirilmesini sağladı. Mostek tarafından pazarlanan ve büyük ilgi gören bu yonga o kadar başarılı oldu ki, kısa zamanda endüstri standardı oldu. Bu önemli başarının elde edilmesinde önemli payı olan Busicom'un diğer bir girişimi de Intel firmasıyla bir dizi muhasebe hesaplayıcısı (*Business Calculator*) devreleri geliştirme programıydı. Yerine getirilmesi gereken şey, değişik koşulları sağlayan bir dizi tümleşik devre geliştirmektir. Bu kadar çok sayıda ve değişik gayeler için istenen çeşitli tümleşik devrelerin tasarımı ve geliştirilmesi hiç de kolay ve ekonomik bir yaklaşım değildi. Herşeyden önce, tümleşiklik düzeyinin artması sonucu geliştirilen yeni yongalar başka uy-

Rüyal Ergül, Y.Prof.Dr., ODTÜ

«ulamalarda da kullanılmadığı için bir bakıma uygulama alanlarının daralması sonucunu ortaya çıkartıyordu. Firmaların, bir ölçüde uygulamaya esneklik getirmek için pazarladıkları Programlanır Mantık Devreleri (*Programmable Logic Arrays*) karmaşık tümleşik devrelerin uygulama alanlarını genişletmek için tasarlanmıştı. Gene de, işlevin yazılım yerine donanım tarafından saptanması kullanıcılar, yani hesaplayıcı ve benzeri cihazlar üreten firmalar için en büyük engel oluyordu.

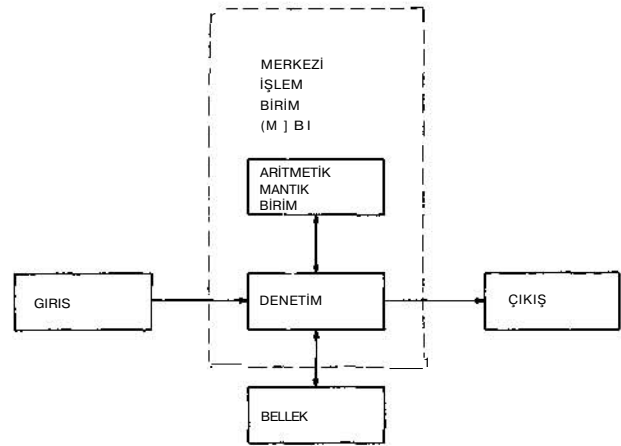
Busicom açısından başarılı olmayan çalışmaların sonucu ortaya çıkan çözüm, Intel firmasının teknik elemanlarından M.E.HOFF Jr. tarafından önerilen çözüm olmuştur. Her özel durum için bir tümleşik devre tasarımı yapmak yerine, özel koşullara göre programlanabilecek bir Salt Oku Bellek-SOB ve bir Rasgele Erişimli Bellek-REB devreleriyle beraber çalışacak güçlü bir Merkezi İşlem Birimi-MİB kullanmak önerilmişti. Böylece, problemi tüm mantık devresini geliştirme problemi olmaktan çıkartıp, sadece SOB üzerinde saklanan bir program geliştirme problemine dönüştürmek istenilen esnekliği sağlayacaktı. Böyle bir sistem bir saklı-program (*stored program*) mikrobilgisayar sistemi oluşturmaktaydı ve kullanıcının isteğine göre geliştirilecek yazılımla özellikleri belli olan dizgeler örneğin hesaplayıcı, yazıcı denetleyicisi (*printer controller*) veya benzeri gibi, çeşitli işlevleri yerine getirebilecek aygıtları gerçekleştirecek tümleşik devreleri geliştirmek kolaylaşıyordu. MİB, REB ve SOB birimlerinin her biri aynı bir yonga üzerinde bulunacaktı. Dolayısıyla, SOB üzerinde saklanacak yazılım SOB yongası üretimi sırasında uygun maskeleyerek gerçekleştirileceğinden, yazılım değişiklikleri dizgenin diğer birimlerini etkilemeyecekti.

Bu fikirlerin uygulama alanına konması sonucu ortaya Intel firması tarafından piyasaya sürülen ilk mikroişlemci Intel 4004 ve yardımcı devreleri başlangıçta pek büyük bir ilgiyle karşılanmadı. 4 bit'lik işlem yapan ve P-MOY teknolojisi kullandığı için yavaş olan bu mikroişlemci sonradan iyileştirilen modeli olan 4040a ve 8 bit'lik şekilli olan 8008'e yerini bırakmıştır. Daha sonra piyasaya çıkartılan geliştirilmiş N-MOY teknolojisi kullanılarak yapılan 8080 ile Intel bu alanda gerçek ilgiyi yaratma başarısını göstermiştir.

özet olarak belirtmeye çalıştığımız gibi, muhasebe hesaplayıcısı üreten firmalara tasarım kolaylığı getirmek amacıyla ortaya atılan bir uygulama alanına konması sonucu ortaya çıkan bir ürün olan mikroişlemci, beklenenin çok üzerinde ilgi görerek değişik alanlarda kendisine yer açmış bir tümleşik devre olmuştur. Bugün için uygulamada çok önemli yeri olan mikroişlemci, genel kurallara uygun olarak, çeşitli sorunlarını da barabar getirmiştir. Mikroişlemcinin bir dizge içinde çeşitli denetim, hesaplama ve değerlendirme işlevlerini yerine getiren bir

birim (*component*) olarak kullanılması, daha önce bilgisayar alanında deneyimi bulunmayan devre tasarımcıları tarafından da kullanılmasını gerektirmektedir. Bu da donanım ve yazılımın iyi anlaşılması ve gerekli programlamanın olduğundan basit bir sorun olarak görülmesi gibi uygulamada pek çok zorlukla karşılaşılmasına yol açmaktadır.

Mikroişlemcilerin günümüzdeki durumu, uygulama için uygun seçim, program geliştirme gibi konulara girmeden önce, mikroişlemci sistemleri konusunda oluşmuş yanlış veya eksik bilgileri düzeltmek ve tamamlamak, bu konuda hiç bir şey bilmeyenlerin de yeterli önbilgiyi (*background*) edinmeleri için mikroişlemcilerin iç yapılarından söz etmek gerekmektedir.



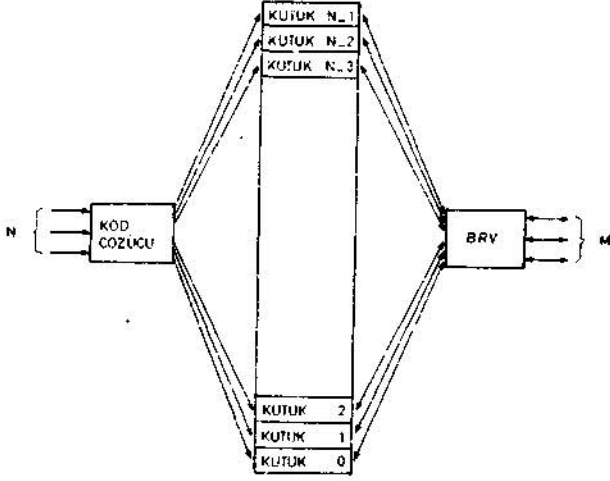
Şekil 1.

II. MİKROİŞLEMCİLERDE İÇ YAPI:

Her bilgisayar Şekil 1 'de gösterilen gerekli 5 temel birimden oluşur. Bir dizgede bu birimlerin tümünün bulunması elbette dizgenin bir bilgisayar olmasını gerektirmez. Örneğin, basit bir cep hesaplayıcısı bile bu beş birimi içerebilir. Fakat bu, bir bilgisayar olarak kabul edilebilmesi için yeterli koşul değildir. Hesaplayıcı bilgisayar yazılımının getirdiği esneklik ve programlanabilirlikten yoksundur. Diğer yünden, bir mikroişlemci dizgesi de bu beş birimi, bir veya birden fazla yonga üzerinde içerecektir. Yapılar arasındaki bu benzerliği gözönüne alarak, bu birimlerin her birini teker teker inceliydim.

Giriş ve çıkış birimleri üzerinden işlem yapılacak verinin alındığı ve işlem gören verinin kullanıcıya döndürüldüğü birimlerdir. Bellek* verinin veya araışlem sonuçlarının daha sonraki kullanım için saklandığı birimdir, bir çeşit yaz-boz tahtası (*scratch-pad*) olarak düşünülebilir.

* Bu yazıda Bellek hem Memory hem de Storage karşılığı olarak kullanılmıştır.



Şekil 2. Rastgele erişilebilir bellek temel yapısı

Aritmetik Mantık Birimi, AMB verinin değişime uğradığı birimdir. AMB girişindeki veri, daha önceden saptanan kurallara uygun olarak değişikliğe uğrar. Bütün bu 4 birim, denetim biriminin gözetimi altında çalışırlar. Denetim birimi bütün işlemlerin ve yer değiştirmelerin hangi sırayla ve hangi birimler arasında yapıldığını sürekli olarak tayin eder. Dolayısıyla mikroişlemcinin tüm özellikleri ve iç yapısı denetim biriminin çalışma şekline bağlıdır.

Mikroişlemci iç yapısı ne olursa olsun, bütün birimler ve alt birimler (*sub-systems*) kütüklerden (*yazmaç-register*)* oluşmaktadır. Mikroişlemci mimarisi, kütükler arasında yapılabilen çeşitli yerdeğiştirmelerin türlerine bağlıdır. Bellek birimi, örneğin, birbirinin aynı pek çok kütüğün bir araya gelmesiyle oluşur. Her kütük tek tek erişilebildiği için, uygulamada belleği adreslenebilen bir kütük demeti olarak düşünebiliriz.

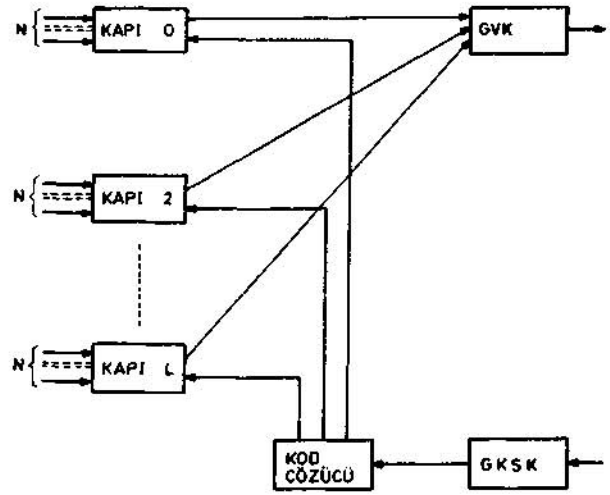
Belirli bir kütüğe erişmek için başka kütüklerden yararlanılacaktır. Adresleme sırasında, bellekteki, bir kütüğün belirleyen adres önce Bellek adres kütüğüne BAK yerleştirilir (Şekil-2) N bit uzunluğunda bir BAK, 2^N kütük adreslemekte kullanılabilir. N bit uzunluğundaki adres, karmaşık bir kütük yapısında olan kod çözücü devre yardımıyla erişilmesi istenen kütükteki verinin okuna süreci içinde BVK-Bellek veri kütüğüne aktarılmasında kullanılır.

Dolayısıyla, genel anlamda bellek biriminin bir kütük dizisi ve iki arabirim kütüğü olan BAK ve BVK'dan oluştuğunu söyleyebiliriz. BVK uzunluğu l betleğin "sözcük u-

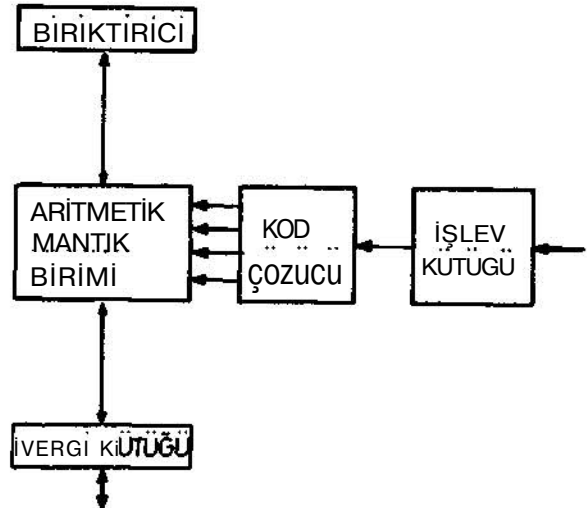
*Nüfus kütüğü örneğinde olduğu gibi, Türkçe'de "kütük" gerçek anlamda "register" karşılığı olarak kullanılmaktadır. Bu nedenle yazıda yazmaç yerine kullanılması uygun görülmüştür.

zunluğu" olarak tanımlanır. BAK ve BVK uzunluklarının aynı olması gerekmediği açıkça bellidir.

Giriş ve çıkış birimleri de bellek birimiyle benzer yapıya sahiptir. Bir mikro işlemci girişi çeşitli kaynaklardan imler kabul edebileceği için birden fazla kapı (*port*) şeklinde organize edilmiştir. Burada kapılar vasıtasıyla (*ki burada her kapının da bir kütük olduğu göz önüne alınmalıdır*) GVK-giriş veri kütüğünün seçilen veri kaynağına erişmesi sağlanır (Şekil 3). Burada hangi kapının kullanılacağı GSKK giriş kapısı seçme kütüğü ve onun kod çözücü devresi yardımıyla saptanır. GVK çıkışı giriş işlemi bitişinde denetim birimine aktarılır. Çıkış birimi de aynı yapıya sahiptir, yalnız bu işlemler ters sırayla uygulanır.



Şekil 3. Giriş kapıları seçme devresi (Çıkışı* kapıları seçme devresi bu devrenin; benzeridir)



Şekil 4.

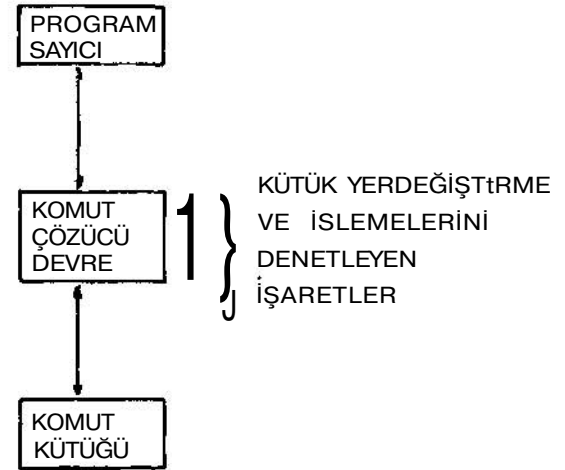
Aritmetik mantık birimi de bir kütükler topluluğudur. Uygun sırayla kütükler arası aktarmalar yaparak istenen işlevi yerine getirir. AMB, toplama çıkarma gibi aritmetik veya VE, YADA gibi manük işlemlerini yapabilmek için iki kütük kullanmak zorundadır. Bunlardan birincisi VK-veri kütüğü diğeri de ara sonuçların yazıldığı ve özel bir isim verilen kütük yani hbirikeç i (*accumulator*) riir. VK yüklenmesi sırasında, «eşzamanlı olarak, veri üzerinde yapılması istenen işlem kodlanmış şekilde İK-ışlev kütüğüne aktarılır. İK çıkışı gene bir uygun kod çözücü devreden geçirilerek AMB'e verilir (Şekil 4). Genel olarak, bilgisayar sözcük uzunluğu hbirikeç i uzunluğu olarak saptanır. Dolayısıyla 8 bitlik sistemler sekiz bit uzunluğunda biriktiricilere sahiptirler.

Şimdiye kadar incelediğimiz birimlerle ilgili çeşitli işlemler, örneğin giriş, çıkış, belleğe yazma, Ibellekten okuma, aritmetik ve mantık işlemleri gibi, belirli bir sıraya göre bir dizi kütük aktarmaları (*register transfers*) yapılarak gerçekleştirilecektir. Denetim birimi bütün diğer birimleri birleştirerek onların belirli bir görevi yerine getirmelerini sağlar. Yalnız denetim biriminin gerekli kütük aktarmalarını yapabilmesi için bu aktarmaların belirlenmesi gerekir. İkili kodlanmış olarak Denetim birimine verilen komutlar (*instructions*) bu aktarmaları belirler. Bir denetim biriminin yerine getirebileceği sonlu sayıda komut vardır ve bunların tümüne birden komut kümesi (*instruction set*) denir.

Programlayıcının seçeceği herhangi bir komut; örneğin veri kütüklerinden birindeki verinin bellekte belirtilen bir yere yazılması gibi, programlayıcının kontrolü altında olmıyan bir dizi iç kütük aktarmaları gerektirecektir. Denetim birimi yönetimi altında yapılan bu kütük aktarmalarına mikrokomut denir. Genellikle denetim birimi, komut kümesindeki her komut için bir veya daha fazla sayıda mikrokomutların yazılı olduğu bir salt oku bellek⁴ kullanır. Bu mikrokomutlar, komutu yerine getirmek için yeterli sayıda kütükler arası aktarmaları gerçekleştirecek iç (*dahili*) imleri içerir. Bilgisayarlarda mikrokomutların yazıldığı SOB değiştirilebiliyorsa, bu türlerine mikroprogramlanabilir özellikle bilgisayar denir. Mikroişlemcilerin pek çoğu sadece üretim yerinde mikroprogramlanır.

Problem çözümü sırasında programlama sonucu ortaya çıkan komut dizisi bir bellekte saklanmalıdır. Günümüzün mikroişlemcilerinin çoğu problem çözümü için gerekli veriyi ve komut dizisini aynı bellek alanında saklarlar. Bu şekilde belleğin daha etkin kullanılması sağlanabilir.

*Bu salt oku bellek, programın yazıldığı ve verinin saklandığı salt oku bellekle karıştırmama Udvır.

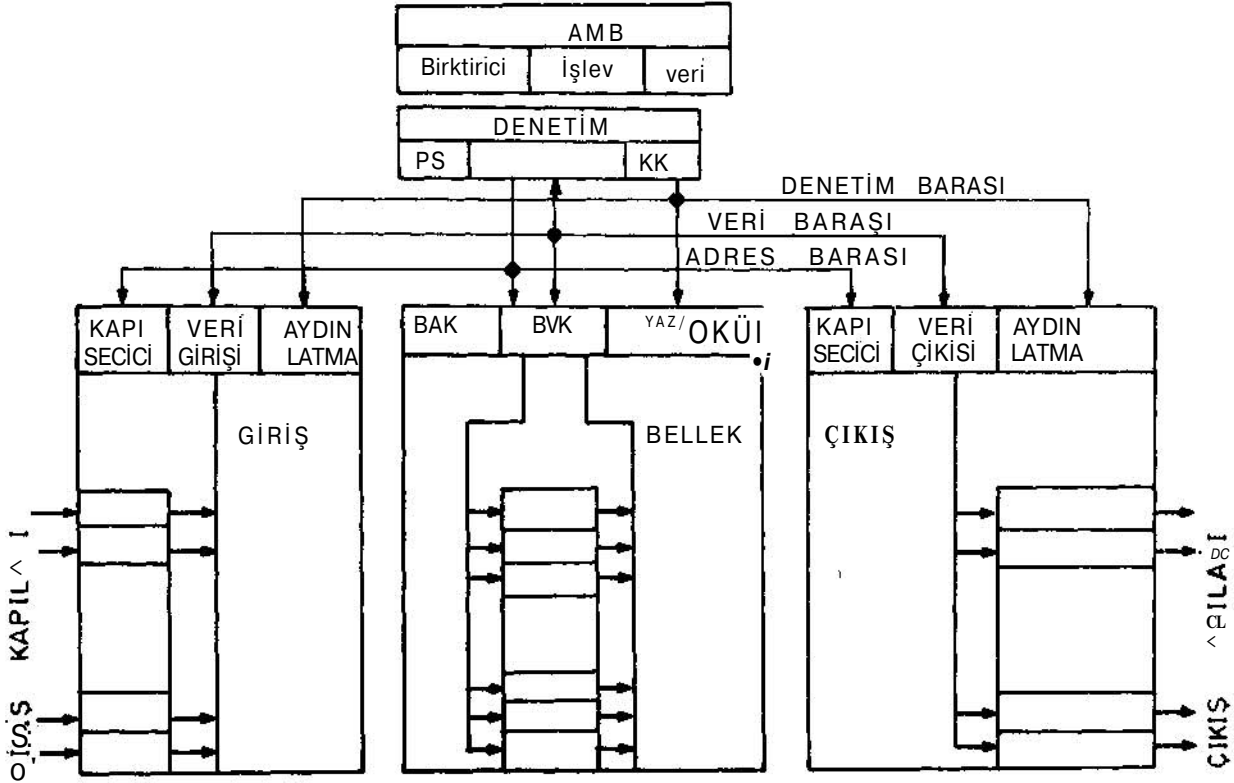


Şekil 5. Denetim birimi temel yapısı

Bellek organizasyonu ne olursa olsun, Şekil 5'de görülen PSK program sayıcı kütüğü tarafından seçilen komutun denetim birimi tarafından yürütülmesi (*execute*) için bu komutun bellekten çağırılması gerekir. Denetim birimi bu çağırma işlemini yapmak için önce PSK işlemi (*content*)'ini bellek adres kütüğüne aktarır. Aktarma işleminin tamamlanmasından sonra da PSK işlemi bir sonraki komutun çağırılmasını sağlamak için bir artırılır. Bellek, seçilen kütük işlemi BVK'ya aktardığı an bu veri BVK'dan KK-komut kütüğüne geçilir.

KK çok sayıda mantık devrelerinden oluşan karmaşık bir kütük olabileceği gibi komuta çözecek (*decode*) bir salt oku bellek adresinin yazılı olduğu başka bir kütük de olabilir. Her iki durumda da sonuç aynıdır: belirli kütükler arasında, komutu gerçekleştirecek biçimde kütük aktarmalarını yapacak bir dizi denetim im'i üretilir. Komut evresi (*eyele*) sonunda yeni komutun okunması ve yürütümü başlatılır. Bu işlem program sonunu belirten komuta tekrarlanır.

Bütün birimlerin bir araya getirilmesiyle elde edilen bir "hypothetical" bilgisayar yapısı Şekil 6'da gösterilmiştir. Burada aşın bir basitleştirme göze çarpmaktadır, örneğin, bir ana saat devresiyle eşzamanlı olarak pek çok denetim im'i üretecek devreler gösterilmiştir, özel işlevleri yerine getirecek pek çok kütük bilgisayar yapısına eklenebilir. Denetim devresinde, örnek olarak, ara sonuçların asıl bellek devresine erişmeden saklanabileceği küçük sayıda yerel kütükler bulunabilir. Bellek erişme zamanı açısından hız ve yerel kütüklerin az sayıda bit'lerden oluşan adres kullanmaları açısından da etkinlik gibi üstünlükler gerçek bir bilgisayarda bu şekilde elde edilebilir.



Şekil 6. Bilgisayar iç yapısı (Kütükler demeti olarak)

Bilgisayar birimleri arasındaki ara-bağlantılar (**interconnections**) genel olarak bara⁴ kavramı göz önünde tutularak yapılır. Mikrobilgisayarların çoğunluğu 3 çeşit bara kullanır. Bunlar adres, veri, denetim bandandır. MİB'nin ürettiği adres ve denetim darbeleri tek yönlü denetim ve adres bara'lan aracılığıyla diğer birimlere iletilir. Halbuki MİB ile diğer birimler arasındaki veri alış-verişi çift yönlü (**bi-directional**) baralar aracılığıyla yapılır. Kuşkusuz, bazı dizgelerin değişik bara yapılan kullanmaları doğaldır. Fakat çift yönlü baraların iki tek yönlü bara olarak veya aynı baranın hem veri hem de adres ve denetim bilgisi iletişiminde kullanılacak şekilde (**ZBC-TDM vasıtasıyla**) tasarlanması her zaman uygun olmayabilir.

III- MİKROİŞLEMCİLERDE ADRESLEME VE KÜTÜK KULLANIMI:

Bir bilgisayarda sözcük boyu bellekte adreslenebilecek birim sayısı birikeç uzunluğu veya bara genişliği olarak alınabilir. Bilgisayar birimleri arasındaki iç iletişim'in (**intercommucation**) önemi dolayısıyla, bu üç büyüklüğün değişik olması durumunda veri bara'sı genişliği genellikle önemli olmaktadır. Dolayısıyla 8 bit'lik mikroişlem-

ci dendiği zaman anlaşılması gereken şey veri barası genişliğinin 8 bit olduğudur. 8 bit'lik dizgelerden örneğin Intel 8008 14 bit'lik BAK, Intel 8080 16 bit'lik BAK kullanılmaktadır.

Buna karşılık Fairchild F8 dizgesinde veri, adres ve denetim bara genişlikleri, sırasıyla, 8, 16 ve 5 bit'dir. Burada denetim bara genişliğiyle komut kümesi büyüklüğü arasında bir ilişki kurulmasına dikkat edilmelidir. Pek çok 8 bitlik dizgelerde 16 veya 24 bit uzunluğunda komutlar kullanılmaktadır.

Gerçek uygulamalarda mikroişlemcilerin aralarındaki farkların iyi anlaşılması için, dizgede hangi kütüklerin olduğu ve kütükler arası veri aktarmalarının nasıl yapıldığının incelenmesi gerekir. Az sayıda kütük kullanan küçük bir komut kümesine sahip ve pek az sayıda kütükler arası işlem yapabilen bir dizge, kolay öğrenilebilir olmasına karşın son derece verimsiz olabilir. Diğer taraftan özel uygulamalar içki pek çok kütüğü olan, çok sayıda komut kabul eden, karmaşık bir dizge başlangıçta zor anlaşılır olmasına rağmen, daha az komutla aynı işi yapabilmesi açısından son derece etkin olabilir.

Bilgisayarlar ve dolayısıyla mikroişlemciler arasında adresleme bakımından da önemli farklılıklar vardır. Yürütümdeki programlar sürekli olarak çeşitli kütüklerde - ki yazılı verilere başvuruyu veya verilerin belleğe yazılı-

ca?

*Ban: "Bu\$" karşılığı olarak ku Uanümüştvr. Bazı çevirilerde "yol" kullanılmışsa da bara Türkçeye yerleşmiş bir terim sayılabileceğinden burada tercih edilmiştir.

şını gerektirir. Doğrudan adresleme adı verilen yöntemde (Şekil 7-a) bellekteki bir tek kütüğün adreslenmesi için komutun adres kısmı yani komutun ilk veya daha sonraki bir sözcüğü BAK-bellek adres kütüğüne yerleştirilir. Okuma çevrimi sırasında veri BVK-bellek yeri kütüğüne aktarılır.

Doğrudan adresleme basit olmasına karşın uygulamada verimli olmayabilir. Pek çok 8 bit'lik dizgelerde olduğu gibi eğer kullanılan bellek 64 K byte'lik ise, yani adreslenmesi gereken 65536 kütük varsa, bu durumda BAK 16 bit uzunluğunda olmalıdır. Dolayısıyla her belleğe başvuru gerektiren komut aynı zamanda 16 bit'lik bir adresi de beraber üretmelidir. Diğer taraftan, pek çok küçük problemlerde belleğin gerçekten kullanılan kısmı 256 kütük uzunluğunda bir kısmı olabilir. Sonuçta bütün adresleme işlemleri 8 bit'le yapılabileceği yerde her adres için 8'i yarasız 16 bit kullanılacaktır.

Kısaltılmış adresleme kavramı bu tür verimsiz kullanmayı önleyebilir. Şekil 7-b'de görüldüğü gibi, yalnızca anlamlı (**âgnificant**) bitler komutla beraber üretilir, geri kalan bitler ya sabit (**preset**)'dir (**genellikle sıfır**) yada başka bir kütükten okunur. Bu tür özelliği olan sistemlerde kısaltılmış adresleme, normal doğrudan adreslemeye yardımcı olarak kullanılır. Komuttaki bir veya daha çok bit hangi adreslemenin kullanıldığını belirtir. Denetim biriminde gerekli kütük aktarmalarının uygun biçimde yapılmasını sağlar.

Kısaltılmış adreslemenin değişik bir türü Şekil 7-c'de gösterilmiştir. Kütük adresleme diye tanımlıyacağımız bu adreslemeye denetim biriminde bulunan veri kütüklerinin kullanılması sırasında başvurulur. Eğer denetim biriminde 8 kütük varsa bunların adreslenmesinde 16 yerine sadece 3 bit kullanılabilir. Bu da komut'un daha fazla bit kullanarak daha karmaşık kütük işlemlerini gerçekleştirmesini sağlar. Bu şekilde aynı komut içinde iki kütük adresi vererek kütükler arasında veri yer değiştirmesi tek komutla sağlanabilir, örnek olarak da Intel 8080 komutları verilebilir.

Katsayılar ve sabitelerle yapılan işlemlerde bu sayıları doğrudan doğruya komutun bir parçası olarak vermek sık sık yararlı görülebilir. Bu şekilde belleğe yazılan sayıların kullanılmak için çağrılması ve dolayısıyla bir okuma çevrimi kadar süre beklememesi önlenir. Verinin komut içerisinde, komutun bir parçası olarak kullanılması-na bitişik adresleme (**immediate adresâng**) denir.

Çok kullanılan adresleme şekillerinden biri de dolaylı adresleme (**indirect addressing**)'dir. (Şekil 7-e). Karmaşıklık sırasına göre sıralarsak bitişik adresleme belleğe bir kez, doğrudan adresleme iki kez ve dolaylı adresleme ise üç kez başvurmayı gerektirir. Dolayısıyla, komut

istenilen verinin yazılı bulunduğu kütük adresinin bulunduğu kütük adresini belirler.

Açıklanan son adresleme şeklinde, veri adresinin yazıldığı kütük olarak eğer denetim birimi kütüklerinden biri kullanılırsa o zaman doğrudan olmayan kütük adresleme adı verilen değişik şekil elde edilir (Şekil 7-f). Böylece, bellekte bulunan veriye erişmek işlemi kolayca adreslenebilen denetim birimi veri kütükleri yardımıyla yapılmış olur.

Göstergeli adresleme, diğer adresleme yöntemleriyle birleştirilebilen en karışık adresleme yoludur. Gösterge kütüğü adı verilen özel bir kütük içle mi, komutun bir parçası olarak belirlenen adresle özdevimli olarak toplanarak yeni bir "gerçek" adres elde edilir. Gösterge kütüğü işlemi değiştirilerek, komut değişikliğe uğratılmadan belleğin tümünün adreslenmesi yapılabilir. Çeşitli yararlanılan bu tür adresleme mikroişlemci komut kümesinde yoksa, yazılım yardımıyla gerçekleştirilebilir.

IV. MİKROİŞLEMCİLER VE MİKROBİLGİSAYARLAR:

Temel kavramların anlaşıldığını kabul ederek mikroişlemci terimini daha iyi tanımlıyabiliriz. Bugünün mikroişlemcileri gerçekte bir tümleşik devre paketi veya az sayıda paketle gerçekleştirilen bir mikrobilgisayarın AMB-Aritmetik Mantık Birimi ve Denetim birimleridir. Bazı istisnalar dışında, mikroişlemciler tam bir mikrobilgisayar değildir, giriş-çıkış ve bellek birimleri eksiktir. Bu günlerde sayıları hızla artan intel 8045 türünden tek yongalı mikroişlemcilerle "mikrobilgisayar" demek daha doğru olacaktır.

Genel olarak, mikroişlemciler kullanılarak gerçekleştirilen mikrobilgisayarlar birer bilgisayardır. Benzer şekilde bazı işlevlerini yerine getirmek için mikroişlemci kullanan aygıtlar da birer mikrobilgisayar olarak kabul edilebilir. Yalnız, aygıtın diğer birimleriyle kaynaşmış bir parçası olduğu için mikrobilgisayar özelliklerinin çoğunu yitirmiş olur.

Mikrobilgisayar tasarımlarında genellikle 8 veya 16 bit'lik mikroişlemciler kullanılır. Diğer taraftan, tasarımda daha büyük esneklik elde etmek ve daha güçlü mikroişlemciler üretmek için "dilim" (**sUce**) yöntemi kullanılabilir. Dilim mimarisinde MİB kütükleri bir dizi içerisinde uzun kütükler elde etmek için kullanılabilir. Her mikroişlemci 2 veya 4 bit üzerinde işlem yaptığı halde, bunların yanyana bağlanmasıyla ortaya çıkan mikrobilgisayar 2'nin katlan türünden istenen uzunlukta sözcüklerle işlem yapabilir. Tek yongalı türlerine göre daha güçlü ve hızlı olan bu tür devrelerin hem tasarımları daha zordur, hem de maliyetleri yüksektir.

8 bit	Intel 4040/4004 (PMOY) Rockwell *PPS-4,1 (P MOY) American Micro Systems *S 2000 (N-MOY)	National MM5799 (N-MOY) Texas Instruments *TMS 1000/1200 Western Digital Corp *1872 (P-MOY)
8 bit	Electronic Arrays 9002 (N-MOY) Fairchild F8 (N-MOY) General Instruments 8000 Serisi (P-MOY) Intel 8080/85 (N-MOY) MOSTechnology 650 X, 651X (N-MOY) Mostek *3870 (N-MOY) Motorola 6800 (N-MOY) General Instruments *PIC 1650/55/70 Motorola *6801 (N-MOY) National 8060 (N-MOY) National 8060 (SC/MP II)	*MOS Technology 660X (N-MOY) RCA Solid State Div. *CDP 1802 (T-MOY) Rockwell Int. PPS-8,PPS-8/2 (P-MOY) Signetics 2650/A (N-MOY) Western Digital Corp. MCP-1600 (N-MOY) Zilog Z-80, Z8/Z8000 (N-MOY) Intel *8048/8748 (N-MOY) Intel 8041/8741 8035 (N-MOY)
8 bit	Intersil Cİ 6100	
16 bit	Data General Micro Nova (N-MOY) Fairchild Macrologic (Schottky-TTM) General Instruments CP-1600/1610 (N-MOY) National PACE (P-MOY)	National IMP-16 (P-MOY) Fairchild 9440 (I ² L) Intel 3000 (Schottky-TTM) Texas Instruments TMS9900/80/40 (N-MOY) SBP9900(I ² L)
16 bit	Advanced Micro Devices AM 2900 (TTM) Monolithic Mpmories 5701/6701 (Schcttky-TTM) Motorola 10800 (YBM-ECL)	Texas Instruments SBPO400A (IIM-I ² L) Texas Instruments 74S481 (Schottky TTM) Signetics 8X300 (Schottky TTM)

Çizelge I

V- MİKROİŞLEMCİLERİN BUGÜNKÜ DURUMU:

Mikroişlemci üretimi çok geniş boyutlara ulaştığı için, çok sayıda mikroişlemci arasında bir karşılaştırma yapmak, bunların arasından uygulamada doğru seçimi yapmak zorlaşmış bulunmaktadır.

Genel çizgiler içerisinde mikroişlemciler üretim teknolojisi ve mimari yönlerinden sınıflandırılabilirler. Üretim de çoğunlukla MOY teknolojisi kullanılmaktadır, 8 bit'lik çok tutulan Intel 8080, Motorola 6800 gibi modeller N-MOY teknolojisi kullanılmaktadır. Sayılan az da olsa Intersil 6100 gibi bazı mikroişlemciler T-MOY teknolojisi kullanılmaktadır. Çift taşıyıcı (*bipolar*) teknolojisi bit-dilim tekniğine uygun yüksek hızlı ve çoğunlukla mikroprogramlanabilir yongaların üretiminde kullanılmaktadır. Mimari yönden sınıflandırma, tek yongalı ve çok yongalı mikroişlemciler olarak yapılabilir.

Çizelge I bugün üretilen mikroişlemcilerin bu tür bir sınıflandırılmasını göstermektedir.

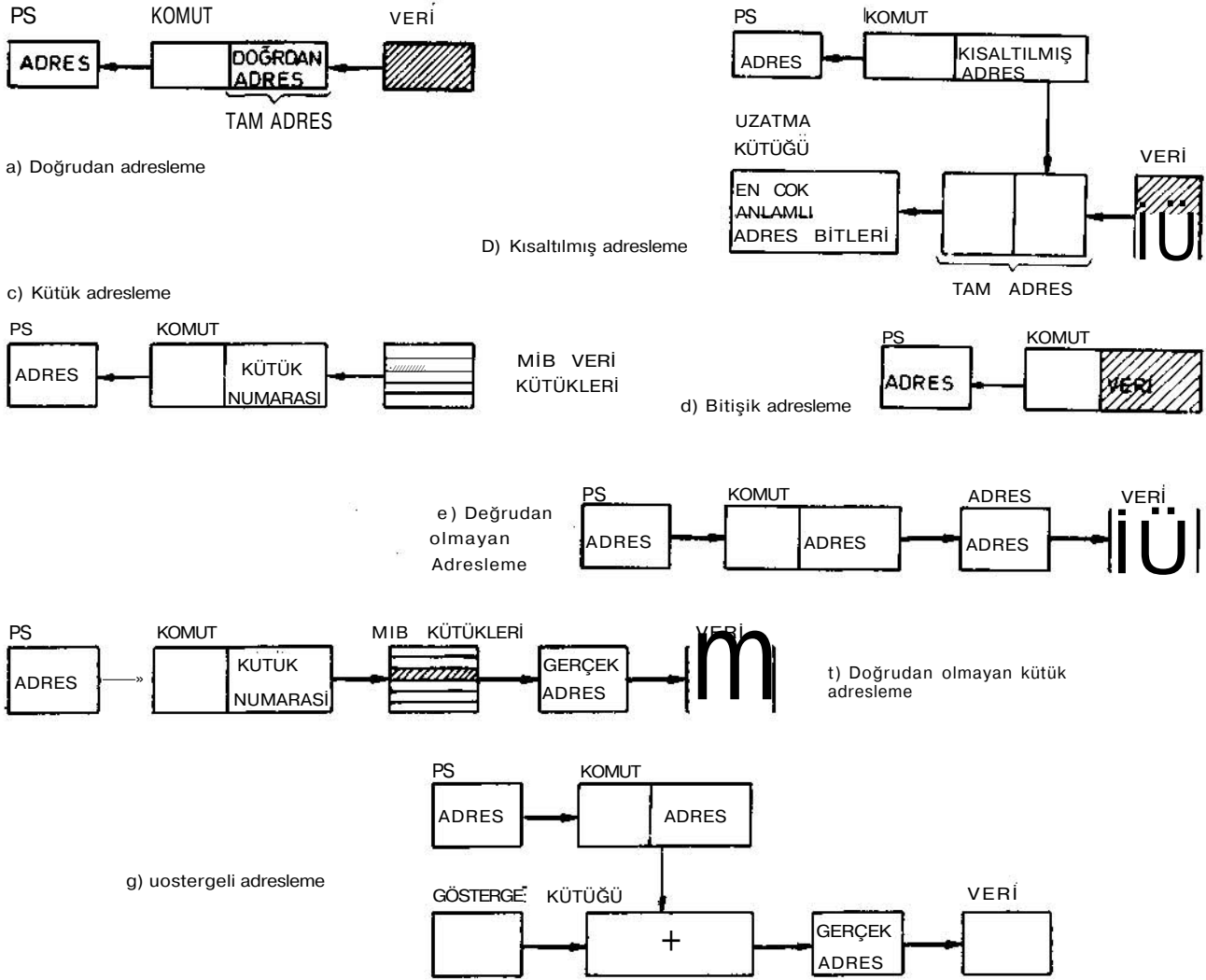
Belirli bir uygulama için mikroişlemci seçimi son derece zor gözükmektedir. Mikroişlemci kullanan dizgelerde donanım sorunlarına ek olarak yazılım geliştirme sorunu da ortaya çıkacaktır. O yüzden sadece yongalar için donanım desteği aranmamalı aynı zamanda yazılım desteği de aranmalıdır.

Donanım açısından bazı kararlar vermek kolay olabilir. Örneğin, tek yongalı mikroişlemciler de SOB, üretim sırasında programlandığı için (*mask programmable*) geliştirme çalışmalarında veya bir kaç binden daha küçük sayıda kullanılacağı yerlerde uygun değildir. Tek yongalı ve programlanabilir SOB'u olan Intel 8748'inde fiyatının 245 dolar olması bu tür mikroişlemcilerin kullanılış alanını daraltmaktadır.

8 bit'lik mikroişlemci kullanılmasına karar verildikten sonra; üreten firmaların çokluğu, iyi yazılım desteği, çeşitli yardımcı devreleri olması ve sistem geliştirme kit'leri olması bakımından büyük bir olasılıkla 8080 seçilecektir veya iyileştirilmiş şekli olan 8085 seçilebilecektir. 8 bit'lik uygulamalarda daha güçlü işlemci istenen durumlarda Zilog Z-80 tercih edilen mikroişlemci olmaktadır. 8080/8085 gibi iyi mimarisi olan ve pek çok firma tarafından üretilen Motorola 6800, 8 bit'lik mikroişlemciler arasında iyi seçim sayılmaktadır.

VI- MİKROİŞLEMCİLERİN PROGRAMLANMASI:

Programlama, istenilen koşulları yerine getirecek komutların bir araya getirilmesi için düzenli bir çalışma gerek-



Şekil 7. Kütük adresleme yöntemleri

tirir. Hiç de kolay olmayan bu işlemin, beş ana öğeden oluştuğu düşünülebilir.

1. Tasarım
2. Kodlama
3. Makine dili ve çevirme
- 4. Test**
5. Hataların düzeltilmesi

Genel olarak, tasarımın yeteri kadar önemle ele alınmadığı durumlarda test ve hataların düzeltilmesi gereksiz yere uzun zaman alacaktır.

Programın yazılması, normal bir problem çözme işleminin daha çok çözümlenmesi gerektirir. Çözümün anlaşılması kadar kullanılacak algoritmanın seçimi de önemlidir.

Programın yazılması bitirildikten sonra kodlanması gerekir. Programın her basamağının, seçilen mikroişlemcinin kabul edebileceği bir dizi komuta çevrilmesine kodlama denmektedir. Programın hatasız olması durumunda, kod-

lama sırasında yapılacak hatalar kolayca bulunabilir ve düzeltilir.

Programın makine diline çevrilmesi, özel bilgisayar programları ve yazılım desteği gerektiren bir işlemdir. Burada özel "**Assembly**" ve "**Compiler**" dilleri kullanılacaktır.

Test evresi geliştirmede zaman alan kısımdır, özellikle tasarım hatalarının bulunması zor hatalardır ve zamanın çoğunu alırlar. Her hata bulunmasında, tasarım, kodlama, **assembly** ve test işlemlerinin tekrarlanması gerekir. Bu basamaklar, program istenilen düzeye erişene kadar tekrarlanır.

KAYNAKLAR

1. Electronic Design News, Kasım 1976.
2. Electronic Design News, Kasım 1977.
3. IEEE Spectrum, Temmuz 1977, S. 18-24.
4. Electronics, Aralık 1977, C. 50, No 25.