mikroişlemci yazılımı

Yüksel TARHAN

UDK: 621.3.049.771.14

ÖZET

Mikroişlemcilerin kullanım ve uygulama alanları giderek genişlerken, mikroişlemcili devre öğeleri hızla gelişmekte ve karmaşıklaşmaktadır. Bu yazıda, mikroişlemciler yazılım açısından ele alınmakta, genel programlama özellikleri, programlama dilleri ve işletim sistemleri kısaca incelenmektedir. Geliştirme ve deneme sistemleri açıklanmaktadır.

SUMMARY

As the areas of application of microprocessors groıv larger, circuit elements become more sophisticated and complex. in this article microprocessor softıuare and general programming characteristics are explained. Program testing and development tools are given briefly.

Yan iletken teknolojisinde son yıllarda sağlanan gelişmeler mühendislik açısından çok önemli bir dizi devre öğesinin ortaya çıkmasına neden oldu. Bu devre öğeleri içinde tasarım, uygulama ve geliştirme açısından üzerinde en çok söz edileni hiç kuşkusuz mikroişlemcilerdir.

Mikroişlemci, bilgi saklama ve giriş/çıkış birimleri yardımıyla, bazı çevre koşullarını sezme, yorumlama ve denetleme işlemlerini yapan elektronik bir devredir. Bu işlemlerin niteliği ve yapılma biçimi, bellekte saklanan programlarca belirlenir. Programlar, uygulamanın öngördüğü işlevleri sağlamakta, öte yandan, donanım birimlerinin desteğiyle, çevre birimlerinin güvenilir ve verimli biçimde çalışmasını sağlamaktadır. Mini ve büyük bilgisayarlarla, belirli bir uygulamada yer alan mikroişlemci ve yardımcı donanıma bir bütün olarak bu açıdan bakıldığında, kavramsal olarak arada büyük bir fark olmadığı söylenebilir.

Uygulama açısından durum biraz daha değişiktir. Laboratuar araçlarında, elektrikli ev eşyalarında, taşıtlarda ve benzeri diğer basit denetim uygulamalarında mini veya büyük bilgisayarlardan ekonomik biçimde yararlanma olanağı yoktur. Büyük çapta hesaplama işlemlerine gerek göstermeyen, boyut ve elektrik enerjisi açısından zorunlu olarak sınırlandırılmış uygulamaların tasarımında mik-

Yüksel Tarhan, HÜ

roişlemciler kaçınılmaz olarak yer alırken, minibilgisayar kullanımı büyük çaplı veya çok **hızlı** hesaplama işlemlerine yönelik, fazla sayıda kullanıcıya hizmet veren uygulama alanlarına doğru kaymaktadır.

Mikroişlemciler kendi aralarında hız, besleme gerilimi, komut kümesi gibi özellikler bakımından guruplanmakla **birlikte,** yer aldıkları uygulamalar kabaca ikiye ayrılabilir.

Göreli olarak durağan, basit, giriş ve çıkışları belirlenmiş denetim işlevlerini sağlamak için mikroişlemcinin atanmış biçimde kullanıldığı uygulamalarda yer alan dizgelere denetleyiciCconfroüer), daha genel amaçlı, değişik programların işletilmesine olanak veren dizgelere mikrobilgisayar *(microcomputer)* **adı veriliyor. Her** iki **guruptaki donanımın istenilen** biçimde iş görmesi için **uygun programların hazırlanması, işletimle ilgili belge, yöntem** ve **verilerin düzenlenmesi, kısaca, yazılım adı verilen öğenin oluşturulması gerekmektedir.**

Bilgisayar donanımında son yıllarda sağlanan gelişmeler, donanım fiyatlarının her on yılda yaklaşık on kat ucuzlamasını sağlarken, yazılım alanındaki gelişmeler çok daha yavaş ortaya çıkmaktadır. Bu nedenle yazılımın donanıma göre önemi artmaktadır. Yoğun biçimde **bilgisayar kullanan ülkelerdeki bazı uygulamalarda, yazılım giderlerinin donanım giderlerine oranı 10:1 ile gösterilmektedir. Diğer bir deyişle, yazılım için donanıma oranla on kat daha fazla harcama yapılmaktadır. Bu oranın önümüzdeki yıllarda giderek büyüyeceği anlaşılmaktadır. Mikroişlemcili dizgelerde bu durum daha fazla belirginlik kazanmıştır. Birkaç yüz liralık bir harcamayla geliştirilecek bir dizgenin işletilebilmesi için gereken büyükçe bir yazılım harcaması, uygulamanın ekonomik verimliliğini ortadan kaldırabilmektedir. Ote yandan yazılımın donanıma bağlı özellikleri, donanım geliştikçe yazılımın bir kısmının değişmesini gerektirmektedir.**

Mikroişlemcili dizgelerin yazılım sorunlarının çözümlenmesinde, bilgisayar yazılımında sağlanan otuz yıllık bir deney ve bilgi birikiminden yararlanılmaktadır. Burada mikroişlemcilerin genel yazılım özelliklerinden bazılarını kısaca sıralayalım.

KOMUT KÜMESİ

Komut kümesi, bir mikroişlemcinin yapabileceği işlemler' belirler. Değişik mikroişlemcilerin komut kümeleri birbirlerine oranla çok büyük değişiklikler göstermekle birlikte, tümünde bulunan ortak komut türleri aşağıdaki gibi sınıflandırılabilir.

Aritmetik ve mantıksal **her mikroişlemcinin toplama, çıkarma, sayaç kaydırma, VE, YADA evirme gibi işlemleri yapan komutları vardır.**

Veri aktarma: Mikroişlemcinin sayaçları arasında, veya bellekle sayaç arastnda bilgi aktarma için kullanılan komutlardır. Bazı mikroişlemcilerde bulunan, işlemlerin en son saklanan değerler üzerinde yapılmasını sağlayan istif ***(stack)*** işleme komutları da bu türdendir-

Sınama: Durum belirteçlerinin değerlerine bağlı olarak ***(sıfır, eksi veya artı sayaç değerleri, toplamada taşma vb)*** program içinde belirli bir konuma atlamayı sağlayan komutlardır.

Arftrma-Efesittme: Sayaçların veya bellekteki bir alanın içindeki değeri bir artıran veya eksilten, elde edilen sonuca göre durum belirteçlerini düzenleyen komutlardır.

Program Denetimi: Programıp içinde istenilen konuma atlanmasını sağlayan, altyordam çağırma, geri dönme, kesilme gibi işlemleri yaptıran komutlardır.

Bayrakların ***Denetimi:*** Bayrakların ***(flag)*** değerlerini düzenleyen komutlardır.

Giriş/Çikış İşlemleri: Giriş/çıkış işlemlerinin yapılmasını denetleyen komutlardır. Birçok mikroişlemcide bu işlemler, giriş/çıkış adresi olarak tanımlanmış bir konum üzerinde normal komuttan kullanarak yapılır, bazılarında özel komutlar bulunabilir.

Bir mikroişlemcinin kullanışlı olması çok sayıda komutu işletebilmesine bağlı değildir. Dünyanın en çok satılan minibilgisayarlarından biri olan PDP-8'in yalnızca sekiz temel komuttan oluşan bir komut kümesi vardı. Çok geniş bir komut kümesinin verimli kullanım sağlayacak biçimde öğrenilmesi, daha küçük ancak anlamlı bir komut kümesine oranla daha zor olabilir. Programlanabilme açısından bir mikroişlemcinin uygunluğu yeterli bir komut kümesinin yanısıra, adresleme olanaklarına da bağlıdır.

ADRESLEME BİÇİMLERİ

Bir mikroişlemcinin sağladığı değişik adresleme biçimleri, mikroişlemcinin gücünü ve program hazırlama kolaylığını belirleyen çok önemli bir özelliktir. Karmaşık yükleyiciler gerektirmeden, yeri değiştirebilen ***(rebcatable)*** programlar hazırlayabilmek için göreli ***(relative)*** adresleme olanağı gereklidir. Eğer mikroişlemcinin büyük boyutlu diziler üzerinde işlem yapması gerekiyorsa dolaylı ve dizinli adresleme yapılabilmesi istenir.

PROGRAMLAMA DİLLERİ

Mikroişlemcinin çalışma biçimi, bütün diğer sayısal aygıtlarda ***olduğu gibi,*** kağıt üzerinde 1 ve 0 ile belirtilen iki değişik gerilim düzeyine dayanıyor. Bütün veri ve komutlar bir noktadan diğerine iletilirken, bellek veya mikroişiemci içinde, bu iki gerilim düzeyi cinsinden gösterilmektedir. Bundan dolayı programcı, program hazırlarken, 1940'lardaki bilgisayar programcısının yaptığı gibi, 1/0 dizilerini ***(veya 8% 16'h karşılıklarım)*** içeren makine dilini kullanmak durumundadır.

Makine dilinin, 1/0 dizilerinden oluşan komutları öğrenme ve hatırlama zorlukları, hata yapma olasılığının yüksekliği, hata bulma ve düzeltme güçlükleri gibi birçok sakıncası, büyük bilgisayar dizgeleri için, yıllarca önce bulunan bir yöntemle, çevirici dili ***(assembly language)*** ile çözümlenmişti. Her komut için akılda tutulması kolay, alfabetik bir kısaltma kullanıldığında gösterim hataları büyük ölçüde azalırken, program hazırlanması kolaylaşıyordu. Çevirici dili ile yazılmış programlar, bilgisayar tarafından, çevirici ***(assembler)*** adı verilen bir başka program tarafından makine diline dönüştürülüyor ve belleğe yerleştiriliyordu. Mikroişlemcili uygulamalarda bu yöntem kavramsal olarak aynen geçerli olmakla birlikte, uygulamada çözümlenmesi gereken sorunlar vardır.

Denetleyicilerin sınırlı yetenekleri ve atanma durumları, çevirici programların belleklerinde bulunmasına olanak vermemekte ve gerektirmemektedir. Bu nedenle, kod dönüşümü, genel amaçlı bilgisayarlarda yapılmaktadır. Hemen her tür bilgisayarda kullanılabilen bir yüksek düzey programlama dilinde ***(FORTRAN gibi)*** hazırlanmış ve derlenmiş olan çevirici program, çevirici dili iie yazılmış olan programı mikroişlemcinin makine diline çevirmektedir. Elde edilen program, kağıt şerit gibi bir ortam üzerinden, veya tuşlarla doğrudan doğruya mikroişlemciye verilmektedir. Bu tüı çeviriciler çapraz çeviricifeross ***assembler)*** adı ile anılmaktadır. Denetleyicilerin belleğinde yer alan programlar genellikle salt oku bellek içinde değişmeyecek biçimde yer aldığından bu yöntem en uygun çözümdür.

Mikrobilgisayarlarda kullanıcının program yazması gerekebildiği için, çeviri işleminin mikrobilgisayar tarafından yapılması, belleğin bir bölümünde yer alan çevirici programın kullanıcı programlarını makine diline çevirmesi istenebilir. Çevirici program salt oku bellek içine yerleştirildiğinde, dizge kapatıldığında silinmeyen, açıldığında otomatik olarak devreye giren yerleşik çevirici ***(resident assembler)*** elde edilir.

Çevirici dilinin, uzmaniaşmamış kullanıcı açısından birçok sakıncası vardır. Programlama süresini kısaltmak, programlama zorluklarının bir bölümünü azaltmak için, konuşma dili ve matematiksel işlemlerin gösterim biçimi gibi günlük yaşamda alışılagelmiş biçimlere yakın kurallardan oluşturulan, bundan dolayı kolaylıkla akılda tutulan ve kullanılan yüksek düzey programlama dilleri

tasarlandı. Bugün birçoğu özel uygulamalar için yararlı olan 200'ü aşkın yüksek düzey programlama dili, nitelikleri farklı sayısız kullanıcının bilgisayarlardan eş düzeyde yararlanmasını sağlamakladır.

Mikroişlemci programlarının hazırlanmasında üzerinde durulması gereken etmenlerin başlıcaları programlama kolaylığı ile kod verimliliğidir. Donanımdan bağımsız bir yapıya sahipolan yüksek düzey programlama dilleri algoritmaların biçimlendirilmesi ve uygulanmasına daha elverişli olmakla birlikte, sonuçta elde edilecek kod çoğu zaman çevirici dili ile doğrudan yazılan koda oranla daha az verimlidir. Bir başka deyişle, yüksek düzey programlama dillerinden biri ile yazılan programlar makine diline dönüştürüldüğünde, çevirici dili ile aynı işi yapan programa oranla daha fazla makine komutu içerdiği görülür. Komut sayısı ise, bir programın çalışma hızını belirleyen önemli göstergelerden biridir. Bu sakıncaların giderilmesi için, yüksek düzey programlama dillerinden makine diline dönüşümü sağlayan derleyicilere eniyileştiren *(optimizer)* yordamlar eklenerek sonuçta elde edilen kodun daha kısa ve öz olması sağlanabiliyor.

Yüksek düzey dillerinden biri ile yazılmış mikrobilgisayar programları, çeviricilerde olduğu gibi, genel amaçlı bilgisayarlar yardımıyla, çapraz derleyici *(cross-compiler)* adı verilen programlar tarafından makine düine çevrilebilmektedir. Çapraz derleyiciler, mikrobilgisayarlardan genel amaçlarla yararlanmak isteyen bilgisayar konusunda uzmanlaşmamış kullanıcılara kolaylık sağlamakla birlikte üretilen makine kodunun verimsizliği ve mikrobilgisayara aktarma zorlukları nedeniyle sınırlı biçimde kullanılmaktadır.

Makine diline dönüşümü sağlayan yordamlar, günümüzde pazarlanan mikrobilgisayarların birçoğunda, özellikle hobi mikrobilgisayarlarında, 4-10 K byte'lık salt oku bellek içine yerleştirilmiş biçimde bulunmaktadır.

Derleyici kavramına yaklaşan bir başka program dönüştürme yöntemi, yüksek düzey dili ile yazılmış programların her komutunun, yorumlayıcı *(interpreter)* adını verebileceğimiz bir program yardımıyla, diğer komutlardan ayrı biçimde çözümlenmesi ve gerçek makine diline dönüştürülmeden işletilmesidir. Bu yöntem, her komutu ayn bir birim olarak aldığından uzmanlaşmamış kullanıcıların program hatalarını bulmaları kolaylaşmaktadır.

Mikrobilgisayarlar için geliştirilen derleyici ve yorumlayıcılar, FORTRAN BASIC, APL, Pascal, LISP gibi yüksek düzey dillerinin kullanılmasına olanak vermektedir. Bunlardan BASIC ve APL, etkileşimli *(interactive)* programlama için dünya çapında kullanılan dillerdir, önümüzdeki yıllarda bu dillere yenilerinin katılacağı rahatlıkla söylenebilir.

İŞLETİM SİSTEMLERİ

işletim sistemi, genel olarak, bir bilgisayar sisteminin çalışmasını denetleyen, kullanıcının program yazma ve işletmesini kolaylaştıran bir program kümesi olarak tanımlanabilir. Büyük bir bilgisayar için bu küme çok sayıda ve kapsamlı programlardan oluşmaktadır. Buna karşın mikroişlemci kullanan basit denetim uygulamaları, sınırlı yetenekleri nedeniyle işletim sistemine gerek göstermemekte, kesilme hata uyarıları, çevre birimlerinin denetimi gibi işlemler ya tümüyle donanım tarafından yada bellekteki küçük bir yordam yardımıyla düzenlenmektedir. Daha genel amaçlı mikrobilgisayarlar için bu geçerli değildir. Çevre birimlerinin belleğe veya mikroişlemciye göndereceği farklı biçim ve içerikteki bilgilerin anlamlandırılması, bilgi kayıplarının önlenmesi, çeşitli programlama ve işletim kolaylıklarının sağlanması için ayrıntılı işletim sistemlerine gerek vardır. Mikrobilgisayarın kullanım amacına, bellek sığasına, donanım ve çevre birimlerinin özelliklerine bağlı olarak değişiklik gösteren işletim sistemi, tuş veya anahtarlar yoluyla kullanıcı tarafından kullanımdan önce yaz oku belleğe yerleştirilebileceği gibi, tümü veya bir bölümü salt oku bellek üzerinde yer alarak, güç verildiği anda mikrobilgisayarın çalışmasını sağlayacak biçimde tasarlanmış olabilir.

Pazarlanan yeni mikrobilgisayarların bir çoğunda, sayaç değerlerinin düzenlenmesi, çevre birimlerinin kullanıma hazır duruma getirilmesi gibi ilk işlemler, dizgeye güç verildiğinde otomatik olarak, bellek üzerinde bulunan küçük bir denetim programı *(monitör)* tarafından yerine getirilmektedir. Bu işleme uyandırma *(bootstrapping)* diyebiliriz. Uyandırma işleminden sonra geniş kapsamlı bir işletim sistemi programı, denetim programının gözetimi altında, kağıt şerit, kaset veya disk gibi bir ortamdan belleğe aktarılabilir.

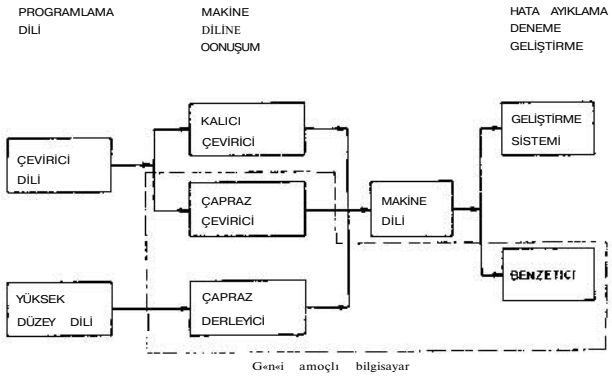
Mikrobilgisayarlarda bugün en çok kullanılan giriş/çıkış birimleri arasında uzakyazıcı *(teletype)* uçlar, katod ışınlı tüpler, satır yazıcı, kağıt şerit, kaset ve küçük disk birimleri bulunmaktadır. Okuma/yazma işlemlerinin büyük bir çoğunluğu, işletim sisteminin hataları önleyici, eşgüdümü sağlayıcı denetimi altında yapılmaktadır. Bununla birlikte, anahtarlı girişlerin kod çözücü vb donanım öğelerince gerçekleştirilmesi de olanaklıdır.

işletim sistemi içinde yer alan programlar, farklı birçok işlemi yerine getirebilmektedir. Programlama dilleri ile ilgili programlardan daha önce söz edilmişti. Metin düzeltici *(tert editör),* kütük düzenleme ve erişim altsistemi, katod ışınlı tüpteki görüntüyü düzenleyen grafik altsistemi, disk ve kaset işletim altsistemi, yükleyici gibi işletim sistemi programları, güvenilir, esnek ve rahat bir iş-

letim sağlamak amacıyla giderek yaygın biçimde kullanılmaktadır.

TASARIM AŞAMASINDA YAZILIM DESTEĞİ

Mikroişlemcili sistemlerin tasarlanmasında yazılım ve donanımın birbirini bütünleme durumu, mikroişlemcilerin atanmış biçimde kullanıldığı denetleyici uygulamalarında ön plana çıkmaktadır. Uygulamanın başarısı, işletilecek programın yetkinliği ile doğrudan ilintilidir. Denetleyicinin yer aldığı dizge genellikle büyük çapta üretileceğinden, üretimden sonra ortaya çıkabilecek bir program hatası veya yetersizliği ekonomik açıdan büyük sorunlar doğuracaktır. Bu durumu önlemek için, tasarım sırasında program hazırlanma ve denenmesinde kolaylık sağlayan, geliştirme dizgesi ve benzetici adı verilen İki yardımcı tasarım öğesi ortaya çık ti (Şekil 1).



Şekil 1. Mikroişlemci programları İşlem evreleri.

Geliştirme sistemi, mikroişlemci yazılımının geliştirilmesinde ve denenmesinde kullanılan tasarlanan dizgenin belirli parçalarına doğrudan doğruya bağlanarak yazılan programların gereken etkinlikte çalışıp çalışmadığını tasarımcı veya programcıya bildiren aygıtlara verilen addır. Bu aygıtlarla programın çalışma çevresi istenilen biçimde düzenlenebilmektedir.

Benzeticiler, bir mikroişlemcinin komut kümesini, çalışma biçim ve ortamını benzetim yoluyla oluşturan, genel amaçlı bilgisayarlar için yazılmış programlardır.

Geliştirme sistemlerinin edinilmesi için oldukça büyük bir harcama yapmak gerekmektedir. Benzeticilerden yararlanmak üzere kiralanacak bilgisayar zamanı ile ek donanım giderleri ise, kısa dönemde az olmakla birlikte, uzun dönemde geliştirme sisteminin giderlerini aşabilir.

Hata ayıklama, deneme ve geliştirme işlemlerinde seçilecek yöntem eldeki olanaklara, istenen işletimsel esneklik düzeyine, kısa ve uzun dönemli giderlere bağlı olarak belirlenmektedir. Geliştirme sistemleri, işletimsel ortama yaklaşan, kullanılan donanım birimleri üzerinde gerçek işlemler yapan bir çözüm olduğundan bazı uygulamalar için uygundur, öte yandan benzeticiler program komutlarını • istenilen düzey ve derinlikte çözümleyebildiklerinden, bellek boyu gibi kısıtlamalar yüzünden bu olanağı sağlamayan geliştirme sistemlerine oranla daha kullanışlıdır.

Programcıya, hazırladığı programı etkileşimli biçimde denetleme olanağı verdiği için geliştirme sistemleri genellikle daha etkin ve hızlı bir çalışma sağlamaktadır.

Mikroişlemci yazılımı, mini veya büyük bilgisayar yazılımının özel bir durumu olarak düşünülebilir. Mikroişlemci ve ek donanım giderlerinin düşük olması, yazılım giderlerinin belirli bir düzeyin altında tutulması sonucunu doğurmaktadır. Mikrobilgisayarlar için bu durum, yetenekli bir dizgenin ilkel sayılabilecek bir yazılım desteği ile işletilmesi anlamına gelmektedir. Hızla gelişen teknoloji, yazılımda standartlaşma çabalarını etkisiz bırakmakta, birçok tasarım için sıfırdan başlayarak yazılım üretmek gerekmektedir.

Mikroişlemciler günlük yaşantımıza girmeye başlamıştır. Bu aygıttan yeterince yararlanabilmek için gereken yazılım geliştirme öğeleri, donanımdan daha hızlı biçimde gerçekleştirilmelidir.

KAYNAKLAR

1. Johnson, G.R., R. Mueller, Automated Generation of Cross-System Software For Microcomputers, Computer, Ocak 1977, s. 23-38.

2. Korn, A.G., A Proposed Method For Simplified Microcomputer Programming, Computer, Ekim 1975, s. 43-52.

3. Krummel, L., G. Schultz, Advances İn Microcomputer Development Systems, Computer, Şubat 1977, s. 13-19.

4. Myers, W. The Need For SoftwareEngineering, Computer, Şubat 1978, s. 12-25.

5. Pantone, J.M., Microprocessor DevelopmentAids: The Timesharing Software Approach, Compcon 78, 1978, s. 331-333.

6. Z80-CPU Technical Manual. Zilog İne, 1976.